THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES
et NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

École doctorale : Electronique, Electrotechnique, Automatique, Traitement du Signal (EEATS)
Spécialité : Génie électrique
Unité de recherche : Laboratoire de Génie Electrique

Apprentissage par renforcement basé sur la connaissance pour la gestion énergétique multi-bâtiments côté demande

Knowledge-Driven Reinforcement Learning for Demand Side Multi-Building Energy Management

Présentée par :

RAM KUMAR Sharath

Direction de thèse :

Benoit DELINCHANT PROFESSEUR DES UNIVERSITES, GRENOBLE INP - UGA	Directeur de thèse
Arvind EASWARAN ASSOCIATE PROFESSOR, NTU SINGAPORE	Co-Directeur de thèse
Rémy RIGO-MARIANI RESEARCH SCIENTIST, CNRS, GRENOBLE INP - UGA	Co-Encadrant

Rapporteurs :

Patrick REIGNER
PROFESSEUR DES UNIVERSITES, GRENOBLE INP - UGA
Rui TAN
ASSOCIATE PROFESSOR, NTU SINGAPORE
Dipti SRINIVASAN
PROFESSOR, NATIONAL UNIVERSITY OF SINGAPORE
Gilles GUERASSIMOFF
PROFESSOR, MINES-PARIS PSL, PARIS

Thèse soutenue publiquement le **12/11/2025** devant le jury composé de :

Patrick REIGNER PROFESSEUR DES UNIVERSITES, GRENOBLE INP - UGA	Président et Rapporteur
Benoit DELINCHANT PROFESSEUR DES UNIVERSITES, GRENOBLE INP - UGA	Directeur de thèse
Rui TAN ASSOCIATE PROFESSOR, NTU SINGAPORE	Rapporteur
Dipti SRINIVASAN PROFESSOR, NATIONAL UNIVERSITY OF SINGAPORE	Rapporteure
Gilles GUERASSIMOFF PROFESSOR, MINES-PARIS PSL, PARIS	Rapporteur

Invités :

Arvind EASWARAN
ASSOCIATE PROFESSOR, NTU SINGAPORE
Rémy RIGO-MARIANI
RESEARCH SCIENTIST, CNRS, GRENOBLE INP - UGA

Titre : Apprentissage par renforcement basé sur la connaissance pour la gestion énergétique multi-bâtiments côté demande

Mots clés : efficacité énergétique, bâtiments intelligents, apprentissage par renforcement, connaissances expertes, gestion multi-bâtiments, flexibilité électrique

Résumé : Les bâtiments représentent une part importante de la consommation et des émissions énergétiques mondiales. Cette thèse explore l'utilisation des connaissances expertes pour surmonter les limites des approches classiques d'apprentissage par renforcement (RL) appliquées à la gestion énergétique des bâtiments. Un cadre de représentation et d'intégration de ces connaissances est proposé, couvrant toutes les étapes du processus RL, de la formulation du problème à la phase de déploiement. À travers plusieurs exemples, il est montré que ces approches « knowledge-informed » permettent d'améliorer l'efficacité de l'apprentissage, de réduire les besoins en données et en infrastructure, de soutenir la participation des occupants et de renforcer la sécurité. L'étude couvre des scénarios allant du bâtiment individuel à l'agrégation multi-bâtiments, démontrant la faisabilité pratique du RL pour la gestion énergétique coordonnée dans des contextes réels.

Title: Knowledge-Driven Reinforcement Learning for Demand Side Multi-Building Energy Management

Keywords: energy efficiency, smart buildings, reinforcement learning, expert knowledge, multi-building management, demand response

Abstract: Buildings account for a significant share of global energy consumption and emissions. This thesis investigates the use of expert knowledge to overcome the limitations of conventional reinforcement learning (RL) approaches applied to building energy management for demand response. A framework for representing and integrating such knowledge is proposed, covering all stages of the RL workflow, from problem formulation to deployment. Through several examples, it is shown that these knowledge-informed strategies improve learning efficiency, reduce data and infrastructure requirements, support occupant participation, and enhance safety. The study spans both single-building and multi-building scenarios, demonstrating the practical feasibility of RL for coordinated energy management under real-world conditions.

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

Disclosure on AI Use:

I hereby declare that OpenAI ChatGPT, version 4, was used to improve the clarity and readability of this thesis. I also declare that I have abided by NTU's Policy on the Use of GenAI in Research, and I have reviewed and edited the content as needed after using this tool, and I take full responsibility for the content of this thesis.

18/12/2025

. . . . . .

Date



. . . . . .

Sharath Ram Kumar

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

18/12/2025

. . . . . .

Date



. . . . . .

Assoc Prof Arvind Easwaran

Authorship Attribution Statement

This thesis contains material from 4 paper(s) published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Chapter 4 includes content from the following publications:

1. S. Ram Kumar, A. Easwaran, B. Delinchant, and R. Rigo-Mariani, "Deep Reinforcement Learning for Coordinated Air-Conditioner Control in Groups of Buildings Using Smart Meter Data", *Engineering Applications of Artificial Intelligence*, Accepted Dec 2025.
2. S. Ram Kumar, A. Easwaran, B. Delinchant, and R. Rigo-Mariani, "Improving Demand Response Programs Using Override Signals with Reinforcement Learning," in Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems, in E-Energy '25. New York, NY, USA: Association for Computing Machinery, June 2025, pp. 603–611. doi: 10.1145/3679240.3734657.

The contributions of the co-authors are as follows:

- A/Prof Arvind Easwaran and Prof Benoit Delinchant provided the initial project direction and supervision.
- I prepared the manuscript drafts, which were revised by all co-authors.
- I co-designed the experiments with guidance from Dr Rémy, developed the codebase, collected the results and analyzed the data to prepare the draft.

Chapter 3 includes content from the following publications:

1. S. R. Kumar, R. Rigo-Mariani, B. Delinchant, and A. Easwaran, "Towards Safe Model-Free Building Energy Management using Masked Reinforcement Learning," in 2023 IEEE PES Innovative Smart Grid Technologies Europe (ISGT EUROPE), IEEE, Oct. 2023, pp. 1–5. doi: 10.1109/ISGTEUROPE56780.2023.10407781.
2. S. R. Kumar, A. Easwaran, B. Delinchant, and R. Rigo-Mariani, "Behavioural cloning based RL agents for district energy management," in Proceedings of the 9th ACM

International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, New York, NY, USA: ACM, Nov. 2022, pp. 466–470. doi: 10.1145/3563357.3566165.

The contributions of the co-authors are as follows:

- A/Prof Arvind Easwaran and Prof Benoit Delinchant provided the initial project direction and supervision.
- I prepared the manuscript drafts, which were revised by all co-authors.
- I co-designed the experiments with guidance from Dr Rémy, developed the codebase, collected the results and analyzed the data to prepare the draft.

18/12/2025

. . . . . .
Date

. . . . . .
Sharath Ram Kumar

Acknowledgments

I would like to express my sincere gratitude to my supervisors, Prof Benoit Delinchant and Assoc Prof Arvind Easwaran, and my co-supervisor, Dr Rémy Rigo-Mariani, for their guidance, encouragement, and insightful feedback throughout the course of this research. Their expertise and support were invaluable to the completion of this thesis.

I am grateful to the members of my thesis committee and examiners for their time, constructive comments, and thoughtful suggestions, which helped improve the quality of this work.
I also acknowledge the support of the Université Grenoble-Alpes and Nanyang Technological University under the joint degree programme, and thank the administrative and technical staff at both institutions for facilitating the research and review process.

My thanks extend to my collaborators and colleagues at the MAGE and SYREL groups at the G2ELab, as well as the HESL at NTU, for their discussions, feedback, and companionship during this journey.
I gratefully acknowledge the financial support provided by the CNRS in France and the NRF in Singapore, through the DesCartes research project at CNRS@CREATE in Singapore, without which this research would not have been possible.

Finally, I would like to thank my family and friends for their constant encouragement, patience, and understanding throughout this endeavour.

This thesis is dedicated to my wife, Shweta, who has been by my side for every step of this journey.

Contents

Acknowledgments i

General introduction 1

1 Buildings and the Energy Transition . . . 1
2 The Potential for Advanced Controls . . . 2
2.a Advanced Control Strategies in Building Energy Management (BEM) . . . 2
2.b Limitations of Advanced Control Strategies . . . 2
3 Summary . . . 4

I Review of Control Techniques for Demand-Side Building Energy Management 5

1 The Role of Building Energy Management (BEM) in Demand Response . 5
2 Types of Controls in Demand-Side Building Energy Management (BEM) 7
2.a Traditional control approaches : Rule-Based Control (RBC) . . . 8
2.b Model-driven control . . . 10
2.c Data-driven control . . . 11
2.d Section Overview . . . 13
3 Hybrid control approaches . . . 14
3.a The Potential for Reinforcement Learning (RL) in the Demand Response (DR) Context . . . 15
3.b Challenges with Reinforcement Learning (RL) for Building Energy Management (BEM) . . . 16
4 Summary . . . 18

II Knowledge-Driven Reinforcement Learning for Building Energy Management 19

1 Knowledge-Informed Reinforcement Learning . . . 19
2 Domain Knowledge in Building Energy Management . . . 20
3 Extended Offline-to-Online Reinforcement Learning (RL) Workflow . . 23
4 Techniques to Incorporate Domain Knowledge in Reinforcement Learning (RL) for Building Energy Management (BEM) . . . 24
4.a Problem Formulation Strategies . . . 26
4.b Reward Function Design . . . 28
4.c Hybrid System Modelling . . . 30
4.d Offline Initialization . . . 32
4.e Asymmetric Actor-Critics . . . 33
4.f Action Masking . . . 35
4.g Online Adaptation . . . 36
5 Comparison to Existing Knowledge-Integrated Frameworks . . . 37
6 Discussion and Summary . . . 38

III Application to a Single-Building Control Problem 39

1 Problem Overview . . . 39
1.a Learning Objectives . . . 40
2 Problem Analysis . . . 41
2.a Relevant Challenges . . . 41
2.b Available Domain Knowledge . . . 42
3 Methodology & Technique Selection . . . 42
3.a Problem Specification and Markov Decision Process (MDP) Design . . . 42
3.b Offline Data Preparation and Policy Initialization . . . 46
3.c Policy Deployment and Online Fine-Tuning . . . 47
3.d Plan of Experiments . . . 47
4 Results & Analysis . . . 49
4.a Numerical Results . . . 49
4.b Metric Trends During Evaluation . . . 52
4.c Impact of D^{init} and D^{upd} . . . 53
5 Summary & Conclusion . . . 55

IV Application to a Multi-Building Coordination Problem **57**
1 Problem Overview . . . 57
1.a System Description . . . 58
1.b Evaluation Metrics . . . 60
2 Solution Approach Under Knowledge-Informed Reinforcement Learning (RL) Paradigm . . . 60
2.a Relevant Challenges . . . 60
2.b Available Domain Knowledge . . . 61
3 Methodology & Technique Selection . . . 61
3.a Problem Formulation Strategies . . . 62
3.b Hybrid System Modelling (Approximate Consumer Model (ACM) Generation) 63
3.c Model-Based Offline Initialization using Asymmetric Actor-Critics . . . 65
3.d Online Adaptation . . . 66
3.e Reward Function Design . . . 67
4 Results & Analysis . . . 68
4.a Baseline and Alternate Strategies for Comparison . . . 69
4.b Accuracy of Approximate Consumer Model (ACM) Predictions . . . 70
4.c Peak Reduction and Effect on Load Profile . . . 71
4.d Impact of Online Adaptation . . . 72
4.e Sensitivity to Discomfort Models . . . 73
4.f Impact of Actor-Critic Architecture and Approximate Consumer Model (ACM) Accuracy . . . 75
4.g Sensitivity to Disaggregation Errors . . . 76
4.h Analysis of the Split-Input Architecture . . . 77
4.i Runtime, Reproducibility and Robustness . . . 81
5 Summary & Conclusion . . . 82

V General conclusion **85**

General conclusion **85**
1 Main Contributions . . . 86
1.a Knowledge-Informed Reinforcement Learning (RL) for Building Energy Management (BEM) . . . 86
1.b Techniques for domain knowledge integration . . . 86
1.c Applications and Key Findings . . . 87

2 Limitations and Future Work 88
2.a Detailed review of existing work under knowledge-informed paradigm 88
2.b Comparison with model-based solution approaches 88
2.c Real-world trials 88
2.d More techniques focused on Demand Response (DR) 89

Bibliography **104**

Appendices **107**

Appendix **107**

A Technical Background **107**
A.1 Deep Reinforcement Learning 107
A.1.a Actor-Critic Reinforcement Learning 108
A.2 Measures of Indoor Thermal Comfort 109

B Appendix to Chapter 4 **111**
B.1 Generation of Appliance Load Dataset 111
B.2 Calculation of Cooling Loads in the Reference System 111

List of Figures

1 Global final energy consumption in buildings by fuel and end-use for Net-Zero Emissions by 2050 (NZE2050). Reproduced from IEA [1] under CC license. . . . 1
2 Potential for advanced control techniques in buildings. Reproduced from IEA [2] under CC license. . . . 3

I.1 Illustration of a typical consumer in Incentive-Based Programs (IBP) in Demand Response (DR). . . . 6
I.2 Interaction between an Aggregator (AG) and consumers in a Demand Response (DR) program. . . . 7
I.3 Main families of controls in Building Energy Management (BEM) contexts, along with their respective input requirements. . . . 8
I.4 Schematic of Rule-Based Control (RBC) approaches . . . 8
I.5 Schematic of model-based approaches . . . 10
I.6 Different paradigms of data-driven control. . . . 12
I.7 Different coordination architectures for multi-building control problems . 13
I.8 Incorporating Rule-Based Control (RBC) and model-based controls into data-driven learning through initialization. . . . 14
I.9 Using data-driven models with optimization-based approaches. . . . 15

II.1 Information flow in Informed Machine Learning, reproduced from [3] under Creative Commons CC-BY license . . . 20
II.2 Steps in Extended RL Workflow . . . 23
II.3 Challenges addressed by different techniques . . . 26
II.4 An example of a Hybrid System Model for a typical building . . . 31
II.5 Different paradigms for offline initialization . . . 32
II.6 Asymmetric actor-critic architectures using a system model . . . 34
II.7 Action masking applied to an Air-Conditioner (AC) control problem . . . 35

III.1 Schematic of Control Problem . . . 40
III.2 Knowledge-Informed Reinforcement Learning (RL) Workflow for Single-Building Problem. . . . 43
III.3 Illustration of ideal operating curve for arbitrage, with an example $P_t^{bess,arb}$ highlighted. . . . 45
III.4 Evaluation metrics under runs (a) with different reward functions, with and without (b) offline initialization, (c) action masking, (d) online adaptation. Red zones denote unacceptable thermal discomfort levels, while yellow is moderately uncomfortable according to [4]. . . . 51
III.5 (a) PMV during occupied hours and (b) electricity bill for each month in the evaluation for different controllers. Reinforcement Learning (RL) runs are based on R^B. . . . 52

III.6 Power flows and temperatures over an example day near the end of the evaluation period. . . . 54

IV.1 System illustration . . . 58
IV.2 Discomfort model used in the simulated testcase . . . 59
IV.3 Aggregated load and ambient temperature over a typical week in the case study with $N = 100$ and $T^{set}_{n,t} = T^{in,ideal}_{n}$ . . . 60
IV.4 Knowledge-Informed Reinforcement Learning (RL) Workflow for Multi-Building Problem. . . . 62
IV.5 Block diagram of an Approximate Consumer Model (ACM). Hat notations represent estimated/unknown quantities. . . . 64
IV.6 Schematic of the proposed Split-Input architecture. Hats represent estimated quantities, bold notations are vectors over L time steps before t. . . . 65
IV.7 Steps involved in the offline and online phases of the approach. . . . 66
IV.8 Visualization of Reward Function Components . . . 67
IV.9 (a) Mean Absolute Error (MAE) of the indoor temperature (b) Mean Absolute Error (MAE) of the Air-Conditioner (AC) power. Both are calculated over the test period under the baseline policy. . . . 70
IV.10 (a) Load duration curves (annotations are the fraction of energy consumed above the highlighted power) (b) Aggregated grid power for a typical day (c) Control signals under each strategy (mean and std deviation across all consumers.) . . . 71
IV.11 Consumer override patterns under different override models with the Split-Input agent. CB-Det represents the deterministic model used to train the agent. . . . 74
IV.12 Violin plots of the aggregate grid power for different strategies (annotations show total energy consumption over the test period) . . . 75
IV.13 Mean training rewards and Explained Variance (Explained Variance (ev)) scores for different agents during training. Explained Variance (ev) is a measure of the accuracy of the critic output (higher values are better, maximum value is 1.0). . . . 78
IV.14 T-Distributed Stochastic Neighbor Embedding (t-SNE) plots of hidden activations of each neural network for different states. Colours indicate whether an override occurred in the previous time step (Yes, No). s is the silhouette score, with higher values indicating a better separation between the labels. . . . 78
IV.15 Visualization of agent policies in Split-Input and Measured-Input architectures for different weather. Subplot x- and y-axes show mean values of $\hat{\mathbf{P}}^{\mathbf{grid}}_{\mathbf{n}}$ and $\sum \mathbf{P}^{\mathbf{grid}}_{\sim\mathbf{n}}$ respectively. Annotations are the fraction of the decision space where $|T^{set,req}_{n,t}| < 0.5$, used here as a measure of policy entropy. . . . 79
IV.16 (a) Avg episode reward in training and (b) Violin plots of the aggregated load for 10 runs of the split-input architecture with different random seeds. Δ is the percentage peak reduction compared to the baseline. . . . 82

V.1 Steps in Extended RL Workflow . . . 86
V.2 Challenges addressed by different techniques . . . 87

A.1 Agent-Environment Interaction in Reinforcement Learning (RL) . . . 108

B.1 4R2C model network [5] . . . 112

B.2 Cooling power and indoor temperatures for different buildings in the reference system 113

List of Abbreviations

AC Air-Conditioner

ACM Approximate Consumer Model

AG Aggregator

AI Artificial Intelligence

ASHRAE American Society of Heating Refrigerating and Air-Conditioning Engineers

BEM Building Energy Management

BESS Battery Energy Storage System

BMS Building Management System

DDPG Deep Deterministic Policy Gradient

DNN Deep Neural Network

DR Demand Response

DRL Deep Reinforcement Learning

EMA Energy Market Authority

EMCS Energy Management and Control System

ev Explained Variance

HVAC Heating Ventilation and Air-Conditioning

IBP Incentive-Based Programs

IEQ Indoor Environmental Quality

IFC Industry Foundation Class

IoT Internet of Things

ISO International Standards Organization

MAE Mean Absolute Error

MDP Markov Decision Process

ML Machine Learning

MPC Model Predictive Control

MSE Mean Squared Error

NILM Non-Intrusive Load Monitoring

NZE2050 Net-Zero Emissions by 2050

PBP Price-Based Programs

PC Pearson Correlation Coefficient

PID Proportional-Integral-Derivative

PLC Programmable Logic Control

PMV Predicted Mean Vote

PPO Proximal Policy Optimization

PV Photovoltaic

RBC Rule-Based Control

RC Resistance-Capacitance

RL Reinforcement Learning

SAC Soft Actor-Critic

SCADA Supervisory Control and Data Acquision

SOC State-of-Charge

t-SNE T-Distributed Stochastic Neighbor Embedding

TD Temporal Difference

TMY Typical Meteorological Year

VAV Variable Air Volume

General introduction

1 Buildings and the Energy Transition

The electricity sector has seen unprecedented change in recent years, driven by the global efforts towards reducing greenhouse gas emissions to avoid the worst impacts of climate change [6]. Despite the growth in renewable electricity sources (such as solar and wind), around 60% of the global electricity production remains fossil-fuel based, and the sector contributes to around 40% of the total greenhouse gas emissions [7]. As the world shifts away from polluting sources of energy through the adoption of technologies such as heat pumps and electric vehicles, there has been a rapid increase in the global demand for electricity in recent years - between 2014 and 2024, there was a 30% increase in the annual production of electrical energy to meet the growing demand [8]. In recent years, rising electricity demand has outpaced the growth of renewable generation capacity, while the increasing share of renewables poses challenges for grids that were designed around centralized production. Together, these trends place significant stress on the grid [8].

A large fraction (over 50%) of the electrical energy is used to power building operations, for end-uses such as heating, cooling, lighting and appliances (and increasingly for the operation of data-centers) - in fact, the building sector is responsible for about 30% of global emissions today [9]. As such, optimizing the operations of energy resources in buildings has the potential to achieve significant energy and emissions reductions. Modern buildings are complex entities comprising of several sub-systems, including Heating Ventilation and Air-Conditioning (HVAC) equipment, local Photovoltaic (PV) production and Battery Energy Storage System (BESS), and several others [10]. The advent of the smart grid has also enabled real-time interaction between utility providers and building management systems, for example through Demand Response (DR) programs based on real-time tariffs [11].

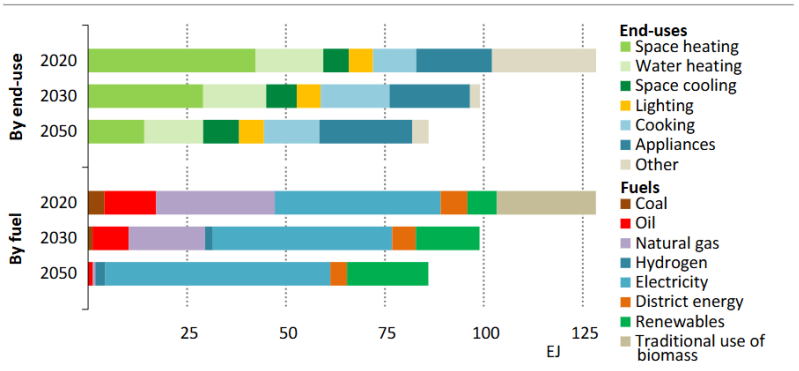

Figure 1: Global final energy consumption in buildings by fuel and end-use for Net-Zero Emissions by 2050 (NZE2050). Reproduced from IEA [1] under CC license.

These trends are illustrated in Fig 1 (adapted from [1]), which depicts global final energy use in buildings by fuel type and end-use, along with the projected pathways needed to meet the goal of Net-Zero Emissions by 2050 (NZE2050). Importantly, these projections assume not only rapid decarbonization of energy supply and widespread electrification of end-uses, but also significant efficiency gains in buildings, leading to an estimated two-thirds reduction in energy demand for space heating and cooling [1]. The optimal management of building sub-systems, taking into account real-time weather and price conditions while ensuring occupant comfort and reliable equipment operation, is a challenging control problem [12]. Meeting these NZE2050 requirements will further intensify stress on the grid due to large-scale electrification, while simultaneously demanding greater operational efficiency within buildings. Addressing both objectives calls for advanced control strategies capable of balancing grid needs with building performance.

2 The Potential for Advanced Controls

2.a Advanced Control Strategies in Building Energy Management (BEM)

A promising path towards reducing emissions from building operations is demand-side management [13], where electricity consumers are incentivized to adjust their consumption patterns to support objectives such as reducing peak demand and improving the integration of renewable energy. Advanced control strategies leverage smart technologies, such as sensors, actuators, Internet of Things (IoT) devices, and Artificial Intelligence (AI), to coordinate energy usage and communicate with the grid for DR[14]. A variety of building systems can be managed in this manner, including HVAC, appliances, lighting, window shades, and on-site generation and storage [2]. In traditional setups, these resources are typically controlled through static schedules or Rule-Based Controls (RBCs), which are easy to implement but often inefficient and unable to adapt to real-time conditions or evolving objectives [15]. By contrast, advanced controls dynamically coordinate and optimize across these systems, delivering more efficient energy use while maintaining safe and reliable operation. Fig 2 (reproduced from [2]) highlights the scope and potential benefits achievable through such controls.

Beyond improving efficiency and reducing energy use, advanced controls take on an even more critical role in today's context of increasing renewable integration into the grid. By moving from purely reactive operation to a planned, proactive approach, they enable predictive strategies such as peak shaving and load shifting [13]. Studies estimate that building flexibility combined with efficiency could reduce annual electricity use by approximately 30%, and cut peak demand by around 20-25%, thus avoiding up to 742 TWh and 181 GW of summer peak capacity in 2030 in the U.S [16]. In this way, advanced control of grid-connected building energy resources can deliver substantial benefits simultaneously to building owners, the electricity grid, and the broader environment.

2.b Limitations of Advanced Control Strategies

Despite their potential, the real-world adoption of advanced control techniques remains very limited [17]. Estimates suggest that fewer than 10% of US buildings are equipped with such capabilities, with penetration rates even lower in most other countries. Several factors contribute to this gap. First, deploying advanced controls often requires significant upfront investment in sensors and automation systems, with limited short-term financial returns for building owners [17]. Second, each building is inherently unique,

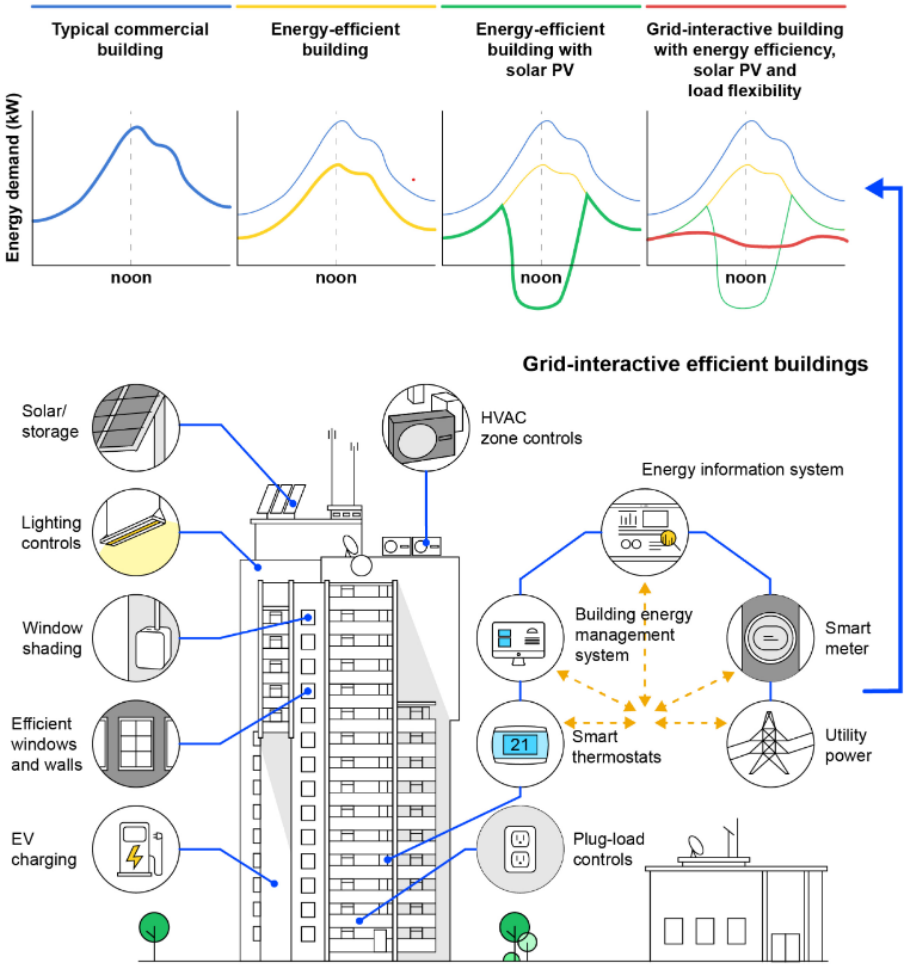

Figure 2: Potential for advanced control techniques in buildings. Reproduced from IEA [2] under CC license.

making the engineering and customization effort substantial and hindering the scalability of solutions across different sites [18]. From the occupant's perspective, concerns about comfort, reliability, and the lack of user interaction options further reduce acceptance [19, 20]. On the technological side, legacy systems are frequently incompatible with advanced controls, and there is a shortage of skilled contractors and service providers to support implementation [21]. Finally, these approaches introduce new risks, such as potential cyberattacks, and can create additional points of failure in building operations [22].

3 Summary

Buildings play a central role in the energy transition, accounting for a major share of electricity use and emissions, and their interaction with the grid is critical for achieving NZE2050 targets. Advanced control strategies, which dynamically coordinate equipment such as HVAC, appliances, storage, and on-site generation, offer a promising means to reduce demand, improve efficiency, and enhance flexibility to support renewable integration. However, despite their potential, adoption remains limited due to high upfront costs, building-specific customization needs, concerns over comfort and reliability, interoperability challenges with legacy systems, and risks such as cybersecurity vulnerabilities. These factors highlight the importance of carefully understanding the building energy management problem, its constraints, and the key issues that must be addressed to enable scalable and effective solutions. The following chapter offers a detailed overview of the problem space and surveys the range of solution approaches proposed in the literature, thereby setting the context for the core focus of this thesis: to study the integration of domain knowledge into a type of data-driven control approach, and its application to the BEM problem.

Chapter I

Review of Control Techniques for Demand-Side Building Energy Management

As illustrated in Fig 2, the BEM problem encompasses the management of multiple components and subsystems within and across buildings, making it inherently broad in scope. Despite this diversity, these problems generally share a common structure: available knowledge and sensor data are leveraged to determine control actions, typically expressed as setpoints. Crucially, these control strategies seek to optimize various metrics, often requiring trade-offs between local objectives - such as energy costs and occupant comfort - and grid-level objectives, like peak load reduction. When extended to multiple buildings, the problem gains an additional layer of complexity, depending on the chosen control architecture, which can range from fully centralized to fully decentralized decision-making.

This chapter reviews the main control types and architectures presented in the literature, outlining their key advantages and disadvantages. The discussion is confined to techniques for managing two critical equipment classes in DR: HVAC systems and BESSs. In line with the scope of this thesis, attention is placed on the technical aspects of control and coordination, and the emphasis is on supervisory control via interfaces such as setpoints, rather than low-level device operation [1].

We start with a brief discussion about demand-side BEM, highlighting the problem scope, as well as the social and technical challenges that are relevant to the space. Subsequently, an overview of existing control approaches is provided, with a specific focus on data-driven controls such as Reinforcement Learning (RL). Finally, we identify the key challenges with such methods in the context of BEM, and motivate the use of domain knowledge as a means to overcome these challenges.

1 The Role of BEM in Demand Response

DR schemes are promising avenues for achieving grid-level targets such as peak load reduction and load shifting [23] by encouraging end-users to alter their consumption patterns. In practice, this can range from sending DR signals that ask users to lower their consumption during specific time windows, to directly controlling end-use resources

[1] We assume the presence of reliable local controllers, such as Proportional-Integral-Derivative (PID), to ensure that equipment can accurately track the assigned setpoints.

such as HVAC systems. There are broadly two types of DR - Incentive-Based Programs (IBP) and Price-Based Programs (PBP) [24]. IBPs directly provide incentives to consumers in exchange for their participation in DR and are typically deployed in industrial contexts. PBPs utilize the price of electricity as a means to modulate the electricity demand with consumers who wish to lower their bills (a paradigm discussed in work published during the course of the thesis [25]). A comprehensive survey of various DR schemes is provided in [26]. While a full review is beyond the scope of this thesis, we present a broad overview and emphasize the key challenges encountered in real-world applications in this section.

While academic interest in the field has grown exponentially in recent years [27], a recent International Energy Agency (IEA) report [1] estimates that the total installed demand response capacity from buildings needs to increase tenfold (from 25 GW to 250 GW between 2020 and 2030) in order to meet global net-zero emissions targets. In this context, there is a need for new DR programs which can address a high level of flexibility (100 kW - few MWs per program [28]), which may be achieved by targeting groups of consumers. PBPs, such as real-time or time-of-use pricing, are by far the most commonly used approach today [23] to achieve such scale, as they require no additional infrastructure. However, their effectiveness has been hindered by poor incentives, low participation rates and negligible response to price variations [29, 30, 31].

As a result, there is increasing interest in scalable IBPs for DR, where consumers' energy resources (e.g., HVAC systems, BESS, or appliances) are directly managed by the DR entity, typically a utility company. While the lack of sensing and control infrastructure has historically hindered the deployment of such approaches [32], the growing availability of smart meters is beginning to address this challenge [33]. Control problems in this paradigm are characterized by the interaction between consumers (with their own energy resources and objectives), and an aggregating entity which seeks to control the consumers' resources for system-level benefits.

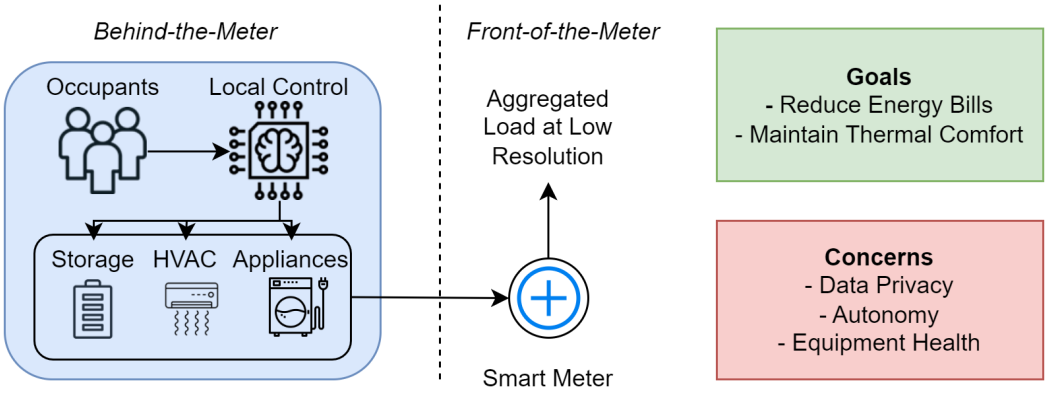

Figure I.1: Illustration of a typical consumer in IBP in DR.

Fig I.1 shows a typical consumer who may be interested to participate in a DR scheme. Consumers have different local energy resources, such as HVAC systems and BESS, and are equipped with smart meters that collect energy consumption data (aggregated at the building level), generally at an hourly resolution. In the absence of the DR scheme, the consumer operates the equipment in the building based on their local objectives, such as reducing energy bills and ensuring indoor thermal comfort (refer A.2 in A.2). In IBP DR schemes, consumers permit an external entity to exercise some control over their equipment - for instance, adjusting HVAC setpoints to lower energy use during peak demand periods [34].

While only a few real-world trials have been reported [35], surveys have shown that potential consumers are reluctant to participate in such DR schemes [36]. This is often due to poor incentive structures, perceived risks of giving up control [37], the need for behind-the-meter access [38, 39], data privacy and security concerns [40], high effort to respond to signals and other personal factors [41]. Other studies suggest that participation rates could improve if consumer autonomy and interaction are encouraged [31, 20], such as allowing signals to be overridden during periods of high discomfort.

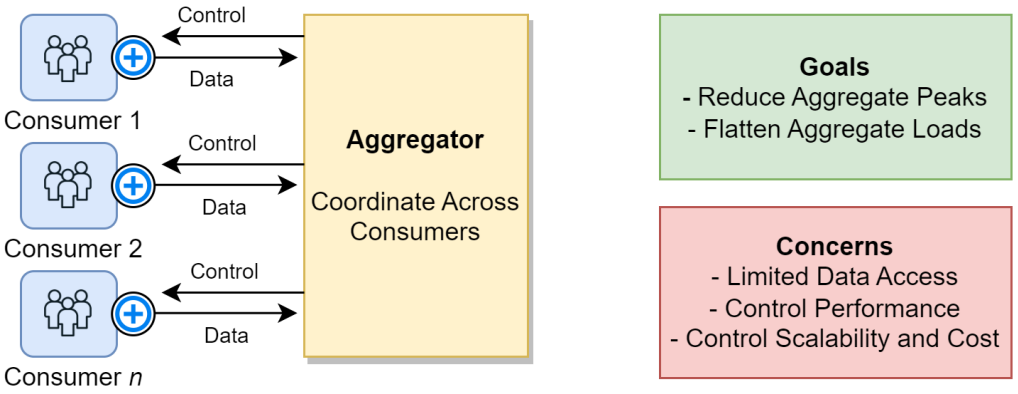

Figure I.2: Interaction between an Aggregator (AG) and consumers in a DR program.

Fig I.2 shows the typical interaction between consumers and an Aggregator in a DR setting, where consumers provide limited data access (for example, smart meter readings) and permit some control of their energy resources. The AG then generates control signals for each participating consumer using the available data, with the goal of achieving system-level objectives such as peak load reduction and load shifting (as illustrated earlier in Fig 2). However, these control signals can conflict with consumers' local objectives, often leading to overrides that reduce the effectiveness of the program [42]. A further complication is that the AG typically lacks access to behind-the-meter data (see Fig I.1) and must rely on building-level smart meter information instead [39]. Additionally, the control approach must be scalable and computationally tractable in order to manage large groups of consumers [40], while requiring minimal additional infrastructure to implement.

The above discussion underscores both the opportunities and the key challenges of implementing IBP in DR, spanning issues at the consumer level as well as at the AG level. This creates a need for control strategies that can jointly address the complex social and technical dimensions present at both ends. The next section reviews the current control landscape, highlighting the main approaches in practice today, and sets the stage for exploring the potential of RL - a data-driven control paradigm - as a means to tackle the problem-level challenges outlined here.

2 Types of Controls in Demand-Side BEM

At the control level, existing approaches can be broadly grouped into three categories: rule-based, model-driven, and data-driven controls. These categories differ in their input requirements, computational complexity, and the trade-offs they offer in terms of strengths and limitations. Fig I.3 illustrates these control families, organized according to their input requirements. Each control type relies on different levels of data and/or

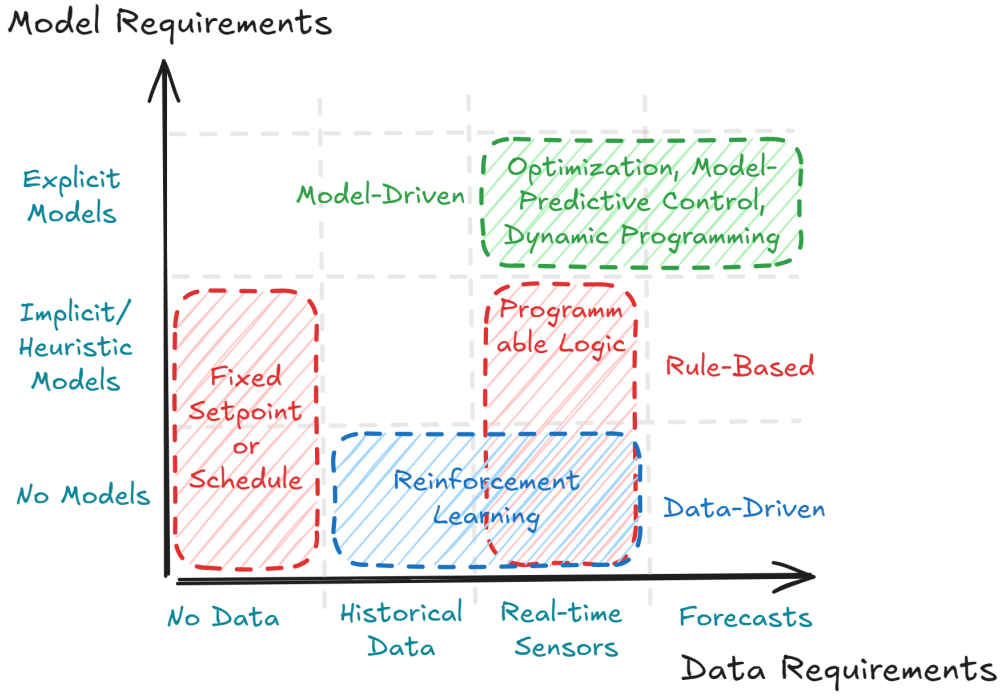

Figure I.3: Main families of controls in BEM contexts, along with their respective input requirements.

modeling effort, with the axes arranged to reflect increasing difficulty in obtaining the corresponding inputs. Some approaches operate without any explicit models, relying only on heuristics or an implicit understanding of the system, whereas others demand detailed mathematical formulations linking inputs, control actions, and outputs. Likewise, along the data dimension, certain methods function without external data, while others require continuous real-time measurements and/or forecasts to operate.

2.a Traditional control approaches : RBC

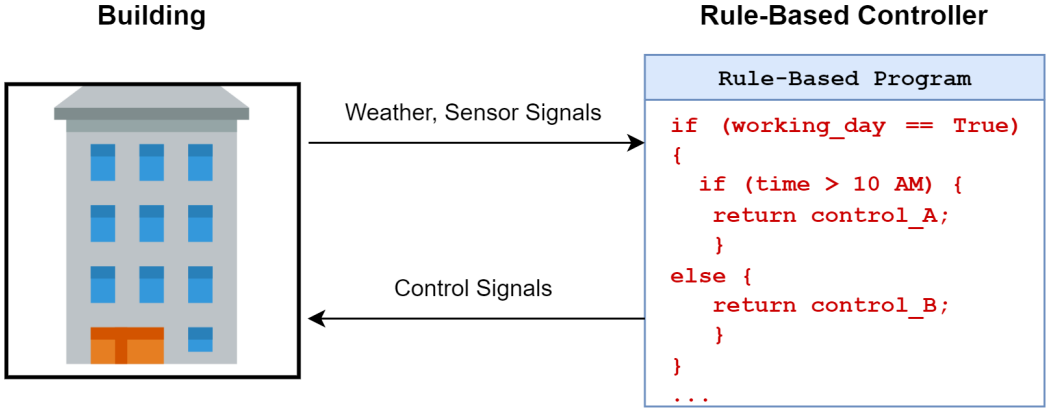

Figure I.4: Schematic of RBC approaches

RBCs remain the most prevalent control strategy in BEM today, largely due to their

simplicity and ease of deployment. Fig I.4 shows the general form of such controls, which are essentially programmed control sequences based on real-time sensor data. They cover a broad spectrum, ranging from fixed setpoint schedules to programmable controls implemented through Programmable Logic Control (PLC) or Supervisory Control and Data Acquision (SCADA) systems. For HVAC applications, many rule-based strategies are formalized in the American Society of Heating Refrigerating and Air-Conditioning Engineers (ASHRAE) Guideline 36 [43], which prescribes a comprehensive set of predefined operation sequences grounded in established system knowledge, with an emphasis on ease of implementation and commissioning. In contrast, for BESSs, no equivalent standardized guideline exists; here, the chosen control mode is typically tailored to the specific system configuration and operational objectives [44]. However, certain patterns, such as maximizing off-grid operation in combination with a PV system, are widely deployed as rule-based strategies for BESSs [45].

In many cases, RBCs strike a favorable balance between simplicity and performance. For instance, [46] compared an advanced Deep Reinforcement Learning (DRL) controller with the operation sequences specified in ASHRAE Guideline 36 for a multi-zone Variable Air Volume (VAV) system in a commercial building, and found that both achieved comparable results in simulation. In a subsequent study, the same authors [47] reported similar findings when benchmarking rule-based controls against optimization-based approaches relying on explicit models. While average performance over the year (in terms of both thermal comfort and energy use) was broadly comparable, the studies highlight that RL and model-based optimization methods offer greater adaptability to seasonal variations. Several studies have shown that simple RBCs, such as fixed schedules or heuristic rules, perform noticeably worse than more advanced control strategies. For example, [48, 49, 50] present direct comparisons between rule-based and advanced controllers in various HVAC scenarios, consistently finding that the latter deliver substantially better performance. It is worth noting, however, that most buildings still rely on these basic RBCs, with only a limited number implementing more sophisticated rule sets such as those outlined in Guideline 36.

These findings underline a central characteristic of RBCs: they deliver reasonable performance for many systems under typical conditions and well-defined objectives, but struggle when external factors such as weather or occupancy vary. Their limitations are particularly evident in BESS and PV applications, where no standardized operational guidelines exist. Recent studies using the CityLearn benchmarking environment [51] show that advanced controllers consistently outperform simple rule-based strategies like schedules. For instance, [52] and [53] report that in dynamic pricing settings, RBCs fail to respond to real-time price signals, resulting in significantly higher energy costs. In this context, there is a large opportunity for advanced control techniques to offer substantial benefits, as discussed subsequently.

The Need for Advanced Control Approaches

Beyond their limited adaptability, a further drawback of RBCs is that they operate in a purely reactive manner - control actions are triggered only after changes in the inputs occur. This makes them ill-suited for applications that require anticipation and forward planning, such as peak load shifting, where actions must often be taken in advance of expected changes in conditions. Advanced control methods address this limitation by incorporating additional sources of information, such as historical data, forecasts, or explicit system models, to generate more optimal strategies that plan over longer horizons and adapt to evolving circumstances. While these approaches typically deliver improved performance, they do so at the expense of increased implementation complexity. A de-

tailed review of advanced control approaches for BEM is provided in [54, 55]. These strategies may be broadly classified into model-driven and data-driven approaches, as discussed below.

2.b Model-driven control

Since buildings and their subsystems are inherently physical, much research has focused on developing computational models that capture the behavior of key components such as thermal zones and storage systems [56]. The dynamics of these systems are also influenced by external factors - such as weather and occupancy - that often exhibit predictable patterns. Accordingly, there is a substantial body of work on forecasting such disturbances over different time horizons [57]. Together, these analytical and differentiable models, combined with forecasts, enable the application of efficient optimization algorithms to building control problems, forming the foundation of model-driven control approaches. This is illustrated in Fig I.5, which highlights how a model and a forecast are integrated in an optimization problem, which calculates the optimal control signals to send to the building.

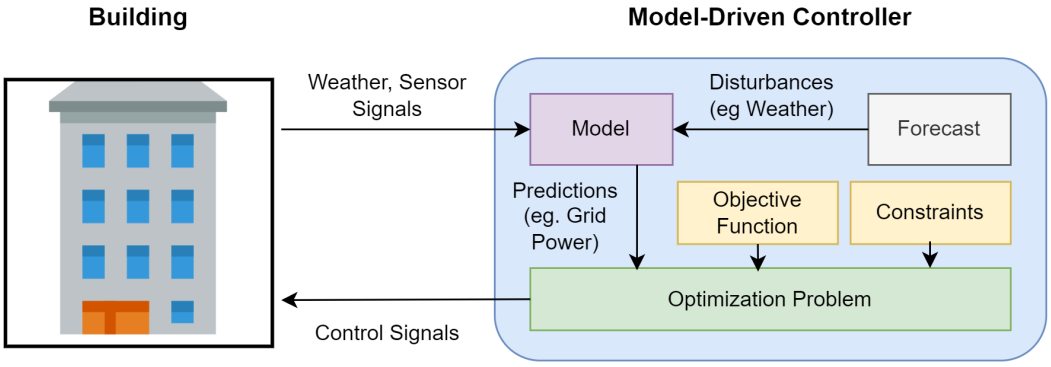

Figure I.5: Schematic of model-based approaches

As outlined in [18], a wide range of optimization formulations have been proposed in BEM, varying with the type of models used, the problem objectives, and the imposed constraints. In general, these methods compute control signals by solving an optimization problem that minimizes a cost function subject to a set of constraints. To ensure tractability, the models are often assumed linear and the cost function convex, enabling the use of efficient solvers. One of the earliest applications of Model Predictive Control (MPC) to HVAC control was presented in [58], which showed improvements in both comfort and energy use for a single-zone building. Similarly, [59] demonstrated MPC in a real-world single-zone lecture theatre, and [60] for an office building, reporting comparable benefits. Beyond HVAC, MPC has also proven effective in managing BESS under dynamic prices and local operating requirements. For instance, [61] proposed an MPC formulation for optimal economic dispatch, while [62] extended this with an aging-aware cost function that accounts for battery health. A further line of research [63, 64, 65] has explored the joint optimization of HVAC and BESS, demonstrating the added value of coordinated control of both systems.

Although extensive academic research and several field demonstrations have highlighted the potential of model-based approaches for building-level DR, their widespread deployment remains limited due to a number of persistent challenges [17]. A primary concern lies in the models themselves: developing and maintaining them requires signif-

icant engineering effort, while the performance of the control solution is highly dependent on model fidelity. The uniqueness of each building exacerbates this issue - models calibrated for one building are rarely transferable to others [66]. Conversely, simplified models with fewer parameters may be applied more broadly but often suffer from reduced accuracy. A further challenge arises from the need to solve an optimization problem at every control step, which imposes a considerable computational burden. This burden grows with problem size and complexity and can hinder integration with existing building management systems [18].

2.c Data-driven control

The data-driven control paradigm relies on operational data from buildings to either learn system models or directly synthesize controllers, thereby avoiding the need for explicit analytical modeling.The dominant approach within this paradigm is Reinforcement Learning, where an agent interacts with the system and improves its performance over time through algorithms that exploit feedback from these interactions. RL is a highly general framework that has been successfully applied to diverse domains, including games [67], robotics [68], and autonomous driving [69]. In recent years, research has focused on DRL, given its promising performance for several challenging control tasks.

Within RL, two broad categories can be distinguished based on how data is used (Fig I.6). In model-free RL, the agent learns a control policy directly from interaction data - for instance, mapping raw sensor measurements in a building directly to control actions. In contrast, model-based RL first learns a predictive model of the system dynamics from data, which is then used for decision-making via optimization methods such as RL, MPC, or evolutionary algorithms. A detailed overview of RL in this problem domain is presented in [15, 55]. The works reviewed in this section focus exclusively on data-driven models that rely on standard function approximation techniques such as neural networks. Approaches that incorporate domain knowledge (for example, thermal physics) to guide the learning of specialized models are discussed in the following section.

A growing body of work has explored the use of RL for control problems in BEM. One of the earliest large-scale successes was reported in [70], where RL was applied to cooling management in a data center, achieving a 40% reduction in energy costs. In the building domain, [71] demonstrated the potential of DRL for HVAC control in simulation, while [72] highlighted the ability of RL agents to learn policies from offline data. More recent studies have focused on advanced DRL algorithms that address some of the limitations of earlier approaches, such as high data requirements and long training times. For example, [73] and [74] applied algorithms such as Soft Actor-Critic (SAC) [75] and Deep Deterministic Policy Gradient (DDPG) [76], which are well-suited for continuous state and action spaces, and reported improvements in both energy efficiency and thermal comfort. Similar benefits have been observed for BESS management. Several works [74, 77, 78], using the CityLearn simulation environment [51], demonstrate that RL-based approaches outperform baselines across metrics such as energy costs, overall consumption, and peak demand reduction in dynamic pricing scenarios. A notable alternate approach is presented in [79], where a learnt model was used in combination with an evolutionary search algorithm to achieve a very high level of performance in the same environment.

As with model-driven approaches, the real-world deployment of data-driven control solutions in BEM remains limited, even though numerous academic studies have demonstrated their potential in simulation [80]. The main factors underlying this limitation - examined in greater detail in Section 3.b - are the long training times and substantial

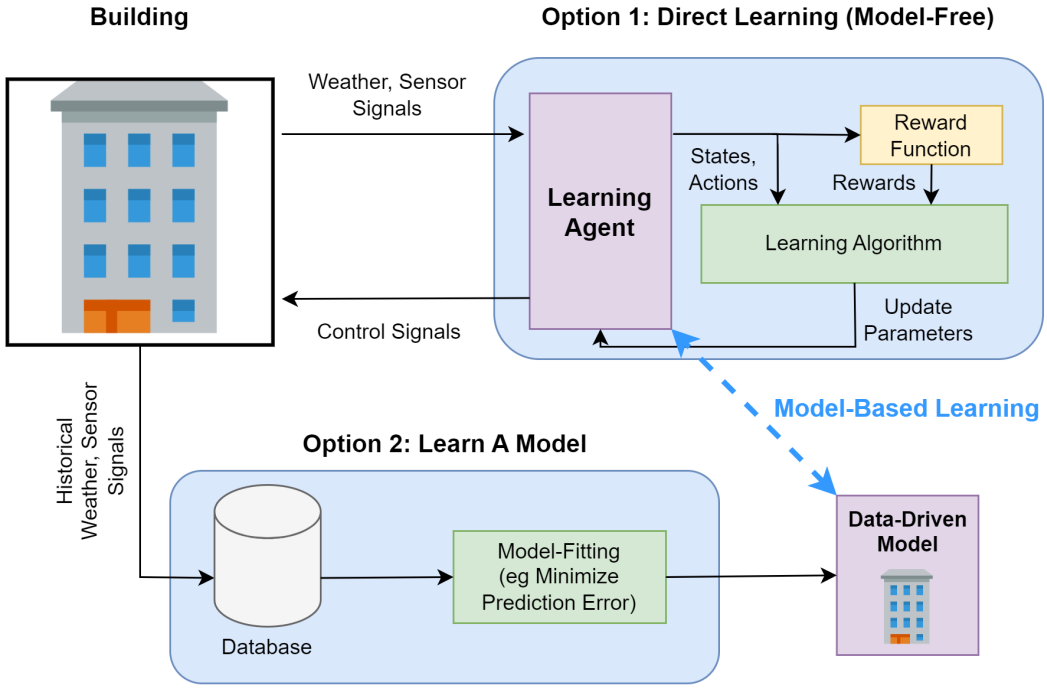

Figure I.6: Different paradigms of data-driven control.

data requirements, as well as the risk of unsafe control actions being issued to the system during training.

Controlling multiple devices and buildings

As the number of devices or buildings under control grows, the complexity of the problem increases, making scalability a critical requirement for DR, where programs often target flexibility at scales of hundreds of kW to several MW [13]. Coordination problems are usually addressed using the same general families of methods, depending on the availability of system data and models. However, advanced control approaches that rely on detailed data or analytical models face significant scalability challenges as system size grows [81, 15]. For example, [82] employ a single centralized RL agent that must process the joint state of the entire system to output a joint action, while [83] formulate multi-consumer control as a single large linear program - in both cases, the authors note the computational complexity that limits the scalability of the solution.

To address scalability, research has focused on adapting existing algorithms to handle larger systems more efficiently. In optimization-based approaches, a common strategy is to decompose the problem into multiple levels and solve it iteratively at each time step. For instance, [84, 73, 85] decompose multi-building HVAC coordination into grid-level and building-level subproblems with distinct objectives. In data-driven methods, parallelization is often pursued through multi-agent schemes. Recent works apply multi-agent RL to coordination problems in BEM [86, 87, 88], while [53] propose a modified framework with a novel information-sharing mechanism.

Beyond computational improvements, the architecture of control algorithms is equally important in BEM, particularly for issues like scalability and data privacy [38]. As shown in Fig I.7, three main arrangements are common: centralized, decentralized, and distributed [89]. Centralized schemes (e.g., [82]) use a single coordinator and achieve near-

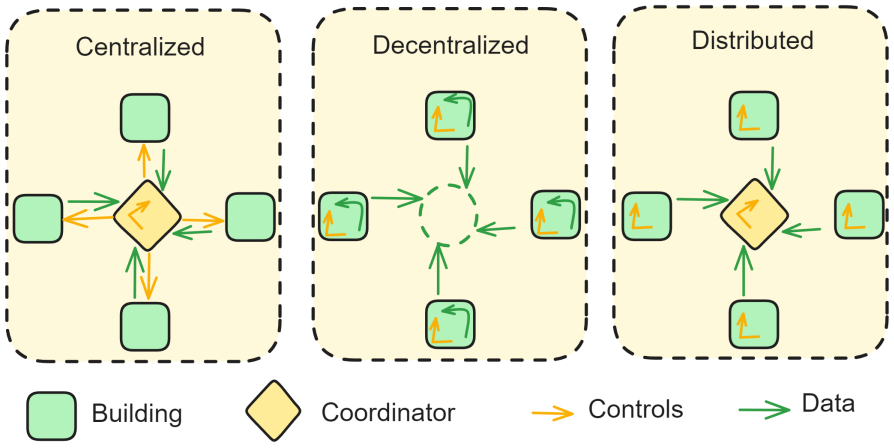

Figure I.7: Different coordination architectures for multi-building control problems

optimality but face scalability and privacy challenges. Decentralized approaches (e.g., [86]) assign independent problems to each entity, improving scalability but risking sub-optimality. Distributed methods (e.g., [84, 73]) split computation across local and central levels with iterative coordination, balancing scalability and optimality at the cost of added complexity.

2.d Section Overview

This section reviewed the main families of control methods applied in BEM, outlining their key characteristics, strengths, and limitations. A summary of these aspects, organized by broad control categories, is provided in Table I.1.

Table I.1: Comparison of different families of control methods in BEM

Criterion	Rule-Based	Model-Driven	Data-Driven
Input Requirements	Minimal	Needs good models and forecasts	Needs large and representative training datasets
Computation Requirements	Very low	High (solving optimization problems)	High (during training), Low (during execution)
Control Optimality	Low-Moderate (acceptable for common scenarios)	High (model and forecast accuracy dependent)	High (after sufficient training)
Adaptability	Low-None	Moderate (reacts to forecasts)	High (learns from interaction and feedback)

Rule-based approaches, such as scheduled setpoints, are the most commonly deployed control mechanism in buildings today due to their simplicity, and provide acceptable performance in many scenarios. However, these techniques often tend to be inefficient due to their inability to adapt to changing conditions (such as weather or occupancy patterns). Model-based approaches, on the other hand, require significant

engineering effort, and are difficult to scale due to the need to develop a separate mathematical model per building. This brings the focus onto data-driven energy management techniques, which use data from sensors and other sources to optimally control different equipment while avoiding the need for detailed models.

3 Hybrid control approaches

Given the limitations of purely model-driven or purely data-driven approaches, recent research has increasingly focused on hybrid methods that integrate elements of both paradigms, with several studies reporting positive simulation results and successful real-world demonstrations. There are a range of techniques proposed in this paradigm, ranging from those that seek to incorporate model-driven insights into a data-driven approach to those that seek to augment models with data-driven insights in an traditional optimization context.

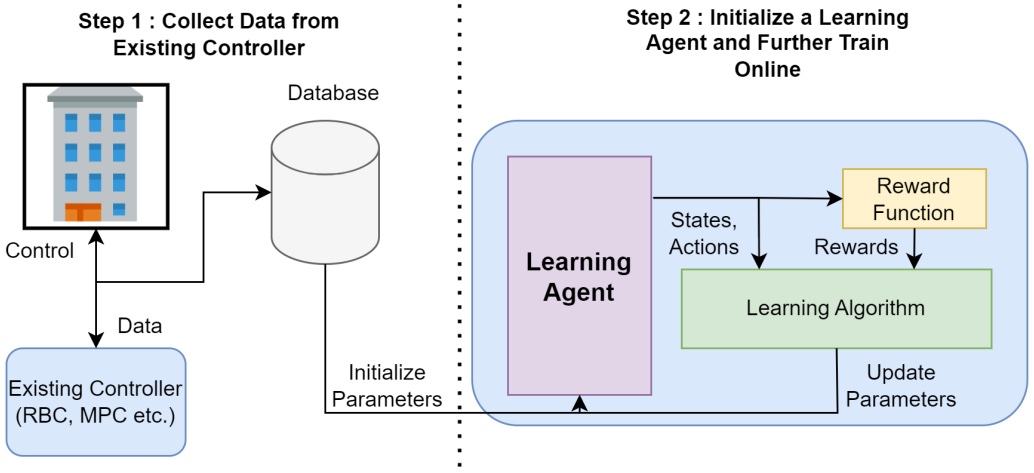

Figure I.8: Incorporating RBC and model-based controls into data-driven learning through initialization.

Fig I.8 illustrates an often-followed approach to integrate rule-based or model-based approaches into data-driven learning paradigms, using an initialization step to bootstrap the learning agent. Here, the interaction between the building and an existing controller is first recorded to create a dataset, from which the parameters of a learning agent are initialized, before transitioning to a data-driven online learning phase [90]. When the initial agent is an RBC, as discussed in our work [91, 92] and others [93, 77, 94, 95, 96], the goal is to preserve the preserve the knowledge embedded in the (potentially sub-optimal) RBC while enabling online adaptation and gradual performance improvements through techniques such as RL. In cases where an initial model-based controller (such as an MPC) is available, the focus is on approximating (imitating) this controller using neural networks in order to eliminate the computational burden of solving an optimization problem at every time step [97, 98].

The model-based learning paradigm shown in Fig I.6 permits another avenue to incorporate models into learning-driven workflows, by parameterizing the model in specific ways based on domain knowledge - a notable example is [99], where a physics-based simulation model fit to telemetry data from an office building is used for RL training. Similarly, [100] encode spatial information about the layout of an office building

through graph neural networks, which are subsequently used to train an RL controller. Models have also been embedded within RL frameworks to provide safety guarantees - [101, 102, 103, 104] describe the use of analytical models to train and safely deploy a DRL controller for HVAC management problems. In general, the above approaches can be extended to any physics-driven building modelling technique discussed in detail in [105, 106] along with suitable model-free optimization techniques such as RL [107].

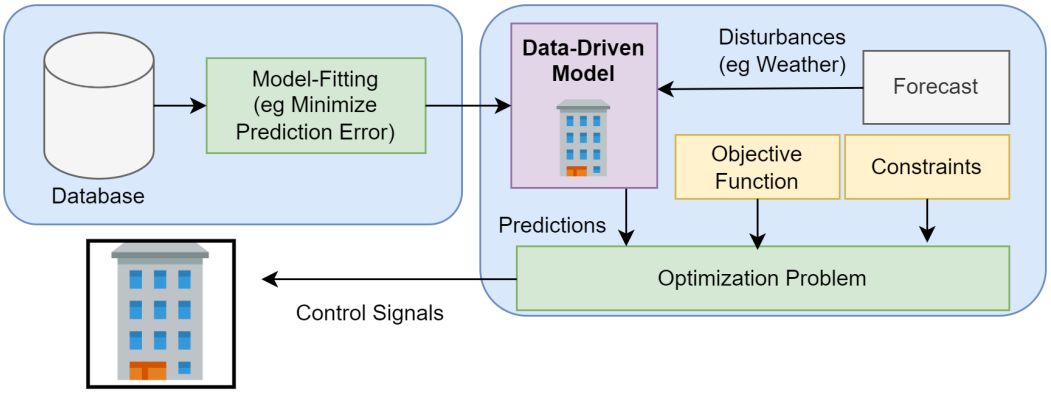

Figure I.9: Using data-driven models with optimization-based approaches.

A complementary approach is illustrated in Fig I.9, in which the models used in an optimization-based approach are based on data from the system. In this context, the structure of the model is critical, with particular emphasis on linearized dynamics that are well-suited for MPC-based formulations. A key line of research focuses on reduced-order models, such as grey-box approaches [108], which have been explored in several works [109, 110, 97]. Along similar lines, [111] presents a real-world demonstration of an MPC controller based on a linear model learned from operational data from an apartment building. Recently, [112] proposed an MPC applied to Deep Neural Network (DNN)-based models for a similar problem. A different approach is presented in [113], where the value-based RL is combined with a one-step MPC in an HVAC control problem. Finally, in contrast to the majority of works that estimate model parameters using accuracy-based metrics, [114] employs a differentiable MPC framework [115] to directly learn parameters that optimize the control objective for an Air-Conditioner (AC) control problem.

3.a The Potential for RL in the DR Context

As discussed above, DR program designers (such as utility companies) often face control problems that must operate at large scale while relying on limited consumer data. Among the available control approaches, RL-based techniques provide a distinct set of advantages that make them particularly well-suited to these problems [15, 116], as summarized below. Additionally, A.1 in A.2 provides a mathematical overview of RL and actor-critic methods.

1. **Ease of Development** : RL controllers are nominally model-free, and can learn directly through interaction with a system.
2. **Adaptability and Performance** : Controllers adapt to changing conditions in real-time, and can offer high performance after sufficient training.
3. **Transferability** : RL controllers can learn in complex environments, and can be designed for transfer across different buildings with minimal fine-tuning [77].

4. **Ease of Deployment** : At deployment, RL policies directly map system states to control actions with minimal computational overheads.

5. **Flexible Knowledge Integration** : A wide range of inputs (including sensor signals, consumer feedback, and other contextual information) can be directly included in the input space without additional modelling effort.

Although numerous studies have explored the use of RL in this context, its real-world deployment remains limited [17]. We conclude this chapter by outlining the main factors behind this low adoption, and in the following chapter propose leveraging domain knowledge as a means to address these challenges.

3.b Challenges with RL for BEM

The previous section reviewed common control methods for resources such as HVAC and BESSs. Among data-driven approaches, Reinforcement Learning is particularly relevant, as it learns control policies through direct interaction with a system [107]. While RL has shown success in domains like games and robotics, its application to building energy management remains limited. In practice, many RL methods are difficult to deploy and often fail to outperform simple heuristics, due to several key challenges [117, 15]:

- RL algorithms often require large datasets (or equivalently, long deployment times) covering years of operational data to learn feasible policies.
- RL approaches are difficult to scale, and solutions remain limited in scope to tens of buildings at best.
- The control policies are generally represented by deep neural networks, which are black-box in nature and are not interpretable.
- There is a risk of unsafe control actions, especially during the initial learning stage, which may damage the equipment under control.

As such, directly deploying standard RL methods in real-world systems like buildings is often impractical. Despite the progress in algorithms and methods, numerous challenges still hinder the effective application of data-driven control. Since this thesis explores how domain knowledge can help overcome these obstacles and enable practical deployment, it is important to clearly define the challenges at hand. To that end, we draw on the comprehensive analysis in [15], which outlines key challenges specific to applying RL in BEM. Each challenge is presented along with measurable metrics, allowing us to later assess whether the knowledge-informed methods proposed in this thesis lead to meaningful improvements.

Learning Challenges

Several learning challenges limit the direct application of RL in building control. These include sample inefficiency and unstable initial behavior, which make online training unsafe and unacceptable in real settings; high state-space complexity, especially in large or heterogeneous buildings, which complicates model generalization and transferability; reward shaping difficulties, where imbalanced cost terms (e.g., thermal comfort dominating energy cost) can misguide learning; delayed feedback, which obscures the long-term effects of actions; and limited transferability of trained agents across different buildings due to differences in state and action representations. Addressing these issues requires

careful design choices and the integration of domain knowledge throughout the RL workflow.

Measurement Metrics

1. **Time to Deployment** (*days*): Total time from from problem specification to policy deployment, including time for data collection or model creation. Lower is better.
2. **Solution Complexity** (*scaling rate*): The dependence of the number of states, actions and model parameters on the number of units to be controlled. Lower is better.

Infrastructure

Many buildings face infrastructural limitations that hinder the deployment of RL-based control. First, most existing Energy Management and Control Systems (EMCSs) are vendor-locked and do not allow external real-time control inputs, making integration of RL agents challenging. Second, the available sensing infrastructure often lacks sufficient spatial and temporal resolution. Critical variables, such as internal heat gains or plug loads, may not be measured at all, and the data that is available may be aggregated or recorded at too coarse a timescale to be useful for learning or control. These issues limit both the development and real-time execution of RL policies.

Measurement Metrics

1. **Required Data Resolution** (*frequency*): The resolution needed for any time-series data used in the control. Lower is better.
2. **Existing Sensor Coverage** (*fraction*): The fraction of required inputs that are covered by existing sensors. Higher is better.

Costs & Benefits

Deploying RL-based building controls involves both costs and benefits that must be carefully weighed. On the cost side, there are expenses associated with installing additional sensors, integrating systems, and training personnel to support and maintain the RL solution. These can pose significant barriers to adoption, particularly in existing buildings. On the benefit side, while energy savings and cost reductions are often highlighted, RL controllers can also deliver non-energy benefits such as reduced greenhouse gas emissions, lower peak electricity demand, and improved Indoor Environmental Quality (IEQ). These broader benefits are often overlooked due to the difficulty in quantifying them, but they represent a critical part of the value proposition for RL-based control.

Measurement Metrics

1. **Lifecycle Profit** (*$*): The total profit potential of the solution over its lifecycle, considering installation and maintenance costs. Higher is better.
2. **Lifecycle Emissions** (*kgCO2e*): The total CO2 emissions caused by the use of the solution over its lifecycle. Lower is better.
3. **Alternate Benefits** (*various*): Measures of alternate benefits such as peak electricity demand (in kWh) or indoor environment quality (fraction of comfortable occupants) - these are context-dependent.

Safety, Security & Trust

RL controllers must prioritize safety, security, and trust to be viable in real-world building applications. Safety involves ensuring that agent actions do not cause equipment failures or create unsafe or uncomfortable indoor conditions. To mitigate risks, fallback mechanisms should allow for seamless reversion to default or rule-based controllers in case of faults. Security must also be addressed, ensuring that the RL system cannot be compromised through cyberattacks. To foster trust, RL decisions should be interpretable, allowing building operators to understand and justify the actions taken. This transparency is critical for operational acceptance and long-term adoption of RL-based control systems.

Measurement Metrics

1. **Safety Violations** (*count*): The number of times the control led to constraint violations (eg, out of temperature/state-of-charge bounds). Lower is better.
2. **Explainability Scores** (*various*): Several methods exist in the domain of explainable AI[118]. Higher is better.

4 Summary

Chapter Summary: Review of Control Techniques

- Modern buildings account for around 30% of global greenhouse gas emissions. Efficient energy management can significantly reduce these emissions and energy bills, in addition to providing grid-level services through DR.
- Optimal management of energy resources in buildings is a complex problem involving several sub-systems, and requires advanced control strategies. Existing approaches have limitations such as the need for explicit mathematical models.
- Advances in sensor infrastructure (eg, smart meters) have resulted in a lower barrier to the development and deployment of data-driven techniques. Reinforcement Learning is a promising data-driven approach for control, but it is not directly suitable for real-world deployments.
- While some innovations exist to address the limitations of RL, they are not sufficient to enable widespread deployments in safety-critical contexts such as energy systems.

Chapter II

Knowledge-Driven Reinforcement Learning for Building Energy Management

1 Knowledge-Informed Reinforcement Learning

In recent years, research has focused on overcoming the limitations of RL by adopting a range of strategies, as discussed briefly in the previous chapter. Despite these innovations, the use of RL in safety-critical contexts such as energy systems has remained very limited. This thesis investigates how domain knowledge can be integrated into the RL pipeline to overcome key challenges in real-world deployments. In the context of energy systems, there exists a rich body of such knowledge, ranging from control heuristics designed by experts, to detailed physical models of building and equipment behaviour. This existing knowledge offers a complementary axis of innovation to address the aforementioned limitations of RL. This thesis specifically focuses on the problem of scalable building energy management, and seeks to answer the following research question:

Research question addressed in this thesis

How can domain knowledge be systematically incorporated into Reinforcement Learning workflows to enable real-world deployments for building energy management, addressing complex multi-objective problems while remaining scalable and safe?

In this chapter, we introduce the proposed framework, including a taxonomy of domain knowledge relevant to BEM, an extended RL pipeline suited to domain knowledge integration, and a set of techniques developed to incorporate this knowledge into different stages of this pipeline. The rest of the chapter is organized as follows:

- Section 2 outlines the taxonomy adopted in this thesis to classify different types of domain knowledge.
- Section 3 presents an extended RL pipeline for domain knowledge integration and introduces a minimal example problem for illustration.
- Section 4 introduces several techniques to leverage different types of domain knowledge, along with a brief illustration of its application.

The following chapters apply these techniques to two practical case studies: (i) energy management in a single grid-connected building equipped with PV, BESS, and AC systems, and (ii) district-level demand response aimed at reducing peak loads through coordinated AC control. These examples highlight how a common set of techniques can be adapted to meet different objectives - the first emphasizes robust control performance and safety under complex conditions, while the second expands the focus to include scalability, consumer privacy, and overall controller effectiveness.

2 Domain Knowledge in Building Energy Management

We define **domain knowledge** as **any problem-specific entity that can inform, constrain, or guide the design and training of a Reinforcement Learning agent, beyond what is available through generic trial-and-error interaction alone**. For example, this could include knowledge of system physics (in the form of differential equations), or the knowledge embedded in heuristic control approaches. The term "**Domain Knowledge-Informed Reinforcement Learning**" is used here to refer to **approaches that explicitly make use of domain knowledge at one or more parts of the RL pipeline**.

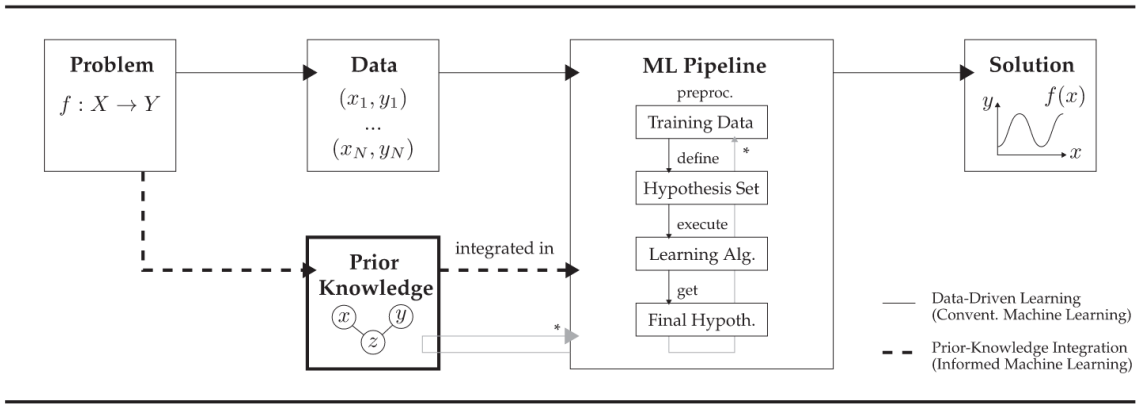

Figure II.1: Information flow in Informed Machine Learning, reproduced from [3] under Creative Commons CC-BY license

To build a taxonomy of the different types of domain knowledge present in the BEM problem, we start by viewing Knowledge-Informed RL as a specific type of Informed Machine Learning [3]. Under this paradigm, prior knowledge is seen as an independent source of information in a machine learning pipeline, and is to be explicitly integrated. This is illustrated in Fig II.1, where information gathered through numerical operations on a dataset are seen as distinct from that gained through the use of domain knowledge. We also adopt the taxonomy proposed in [3] (and its extension to RL for robotics applications [119]), where domain knowledge originates from one of three distinct sources, and may be represented in many different forms. The three knowledge sources identified are:

1. **Scientific Knowledge** : This is formalized knowledge, which typically comes from scientific fields such as mathematics, physics or engineering.
2. **World Knowledge** : This refers to either formalized or intuitive knowledge, typically held by a large group of people, such as general knowledge and facts.
3. **Expert Knowledge** : This is knowledge held by a few experienced professionals, which represents lessons learnt from real-world practice in a domain and is typically informal in nature.

It should be noted that the knowledge sources described above are abstract in nature, and may apply to any problem domain. For a specific problem, domain knowledge manifests as a set of **representations**, which originate from a combination of different knowledge sources. A representation provides a form that embodies the knowledge - for example, abstract scientific knowledge of the thermal physics of a system may be represented as a set of differential equations. It is these problem-dependent representations that are ultimately embedded into the Machine Learning (ML) workflow - as such, they are the primary concern for practitioners aiming to embed knowledge into learning systems. Dedicated research streams exist which address the integration of specific knowledge representations into ML workflows - for example, Physics-Informed ML [120] focuses on incorporating physics knowledge. In contrast, Informed ML is more flexible, allowing a broader range of knowledge representations.

In practice, a wide range of such representations may be available, such as architectural drawings, building codes, operation schedules and semantic models. While an exhaustive list of these representations is beyond the scope of this thesis, an illustrative subset is used here to facilitate a structured discussion on the use of domain knowledge for RL applications. As such, we propose that domain knowledge in the field of building energy management may be classified into the following representations:

1. **Building Physics** - Physical properties and thermodynamic behavior of the building envelope and systems, including material thermal conductivity, heat transfer dynamics, and HVAC system responses.

2. **Semantic Models** - Structured descriptions of building components, spatial relationships, and system topology, typically encoded in formats such as Industry Foundation Class (IFC) or Brick schemas [121]; may also include architectural drawings and zoning layouts.

3. **Computational Models** - Simulation-based or data-driven models used for predicting thermal loads, indoor conditions, or system behavior, including Resistance-Capacitance (RC) models, EnergyPlus simulations, and surrogate models.

4. **System Constraints** - Inherent physical, operational, and safety limits of building systems, including actuator bounds, ramp rates, equipment cycling constraints, and thermal setpoint ranges that must be respected to ensure reliable and safe operation. Also includes constraints imposed through policy and regulations, such as building codes.

5. **Existing Operational Strategies** - Expert knowledge describing typical equipment operation, rule-based/heuristic controls, occupancy schedules, and maintenance routines.

6. **User Interaction and Feedback** - Data reflecting user comfort preferences, control overrides, thermal complaints, or qualitative feedback gathered through surveys or sensor-driven inference.

The above classification focuses specifically on domain knowledge representations that are relevant from a control perspective. Table II.1 outlines these in terms of the sources of knowledge (following the taxonomical structure in [3]), and shows a few specific examples of each representation. Scientific and world knowledge are generally widely available, whereas expert knowledge is often harder to acquire. Nonetheless, many of the representations discussed here depend to some extent on expert input. This reliance is one of the key limitations of knowledge-informed RL.

Table II.1: An overview of domain knowledge representations in Building Energy Management

Knowledge Representation	**Knowledge Source**			**Examples**
	Scientific	*World*	*Expert*	
Building Physics	Y	N	N	Heat Transfer Dynamics; Physical Properties; HVAC Operation Principles
Semantic Models	N	Y	Y	BIM, IFC or Brick Schema; Architectural Drawings; Zoning Layouts
Computational Models	Y	N	Y	White-Box models (EnergyPlus, Modelica, TRNSYS); Reduced-Order Models (RC models, surrogates)
System Constraints	N	Y	Y	Equipment ramp rates, cycling constraints, actuator bounds; Operational guidelines; Policy/regulatory directives
Existing Operational Strategies	N	N	Y	Setpoint schedules; Expert-designed control policies (ASHRAE Guideline-36); Heuristic/rule-based controls
User Interaction and Feedback	N	Y	N	User comfort preferences; discomfort tolerance and overrides; user feedback (surveys, inferred from sensors)

3 Extended Offline-to-Online RL Workflow

Traditional RL methods rely on online interaction with the environment, gradually refining the policy through trial and error. However, such demands are impractical and potentially unsafe for real-world, safety-critical systems like energy infrastructure [116]. As a result, a more viable alternative is to first train effective policies offline using simulators or historical data, and then deploy them in the real environment with mechanisms for ongoing adaptation. This approach, known as "offline-to-online RL" [122], is the one considered in this thesis.

In the RL literature, the offline-to-online learning pipeline typically begins after the problem has already been framed as an Markov Decision Process (MDP)[123]. The focus is then on selecting algorithms that make the most effective use of available offline data to learn a policy, and on addressing errors that arise during the transition to online deployment [122] - particularly those caused by discrepancies between the offline dataset (or simulator) and the real-world system [124].

In domains like BEM, tasks are not always naturally suited for an RL-based formulation. This opens opportunities to embed domain knowledge earlier, even before defining the problem as an MDP, where careful formulation can prevent downstream issues. Domain expertise is equally important after training, to ensure that learned behaviors remain realistic and aligned with system dynamics. Accordingly, this thesis adopts an extended offline-to-online RL workflow that incorporates domain knowledge both prior to MDP formulation and after policy training.

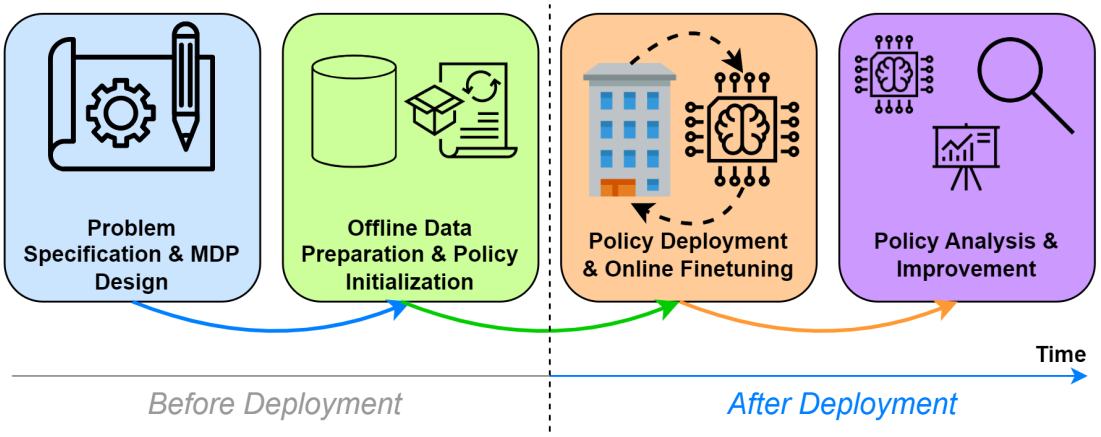

Figure II.2: Steps in Extended RL Workflow

With this in mind, we define an Extended Offline-to-Online RL Workflow consisting of the following steps, as outlined in Fig II.2. Each stage of this pipeline presents an opportunity to integrate domain knowledge - a topic that will be explored in detail throughout the remainder of the chapter.

1. **Problem Specification and MDP Design**: This step involves identifying the control task, defining the objective, and formulating the MDP - including states, actions, transitions, and rewards.

2. **Offline Data Preparation and Policy Initialization**: Here, available offline data (e.g., logs, historical trajectories, or simulation data) is processed for use in training. Depending on the data quality and coverage, various techniques are used to derive a reasonable initial policy.

3. **Policy Deployment and Online Fine-Tuning**: The initial policy is deployed in the target environment with mechanisms for safe and gradual online improvement. This stage must carefully manage the trade-off between exploration and exploitation, especially in real-world systems with strict operational constraints.

4. **Policy Analysis and Improvement**: The post-training phase where a learned policy is examined in depth to assess its quality, robustness, and alignment with real-world expectations. Insights from this analysis can then guide targeted improvements to the policy or even inform revisions to the problem formulation itself.

4 Techniques to Incorporate Domain Knowledge in RL for BEM

This section covers a set of techniques which enable the use of different types of domain knowledge at different stages of the RL workflow. A technique requires one or more groups of domain knowledge, and may be used at different stages in the pipeline. Each technique improves the RL solution w.r.t one or more of the challenges identified in 3.b, and is illustrated through its application in a simplified example problem. Several techniques may be utilized in combination to improve different aspects of the RL solution, as long as the relevant domain knowledge inputs are available.

In this work, the term technique is used broadly to reflect interventions across the entire RL workflow, not just algorithmic tweaks. The most significant role of domain knowledge often arises early, during problem formulation - through choices in state and action spaces, reward design, or model architecture - which ultimately shape what the agent learns. Beyond this, domain expertise can guide model initialization, policy training, and environment design to improve efficiency, safety, and generalization. It also remains crucial after training, where post-hoc analyses help interpret, validate, and refine learned behaviors. Taken together, the techniques discussed here represent informed modifications to the full RL pipeline, tailored to real-world deployment in building energy management under data limitations, safety requirements, and multi-agent complexity.

Overview of Techniques

Table II.2 offers a high-level overview linking each technique (presented in detail subsequently) to the types of domain knowledge it relies on. Its purpose is to help readers navigate the information presented in the chapter, rather than to serve as a definitive or exhaustive reference. The table is designed to answer two practical questions. First, for users who already possess certain forms of domain knowledge, it shows which techniques can be used to incorporate that knowledge into their RL workflow. Second, for those encountering specific challenges during the development of RL-based solutions, it helps identify which techniques are applicable and what types of domain knowledge are required. These two use cases reflect common entry points for domain experts exploring the integration of RL in building energy management.

Table II.2: Knowledge Representations Associated with Different Techniques

Techniques	***Building Physics***	***Semantic Models***	***Computational Models***	***System Constraints***	***Existing Operational Strategies***	***User Interaction & Feedback***
Problem Formulation Strategies	N	Y	N	Y	N	Y
Reward Function Design	Y	N	N	Y	N	Y
Hybrid System Modelling	Y	Y	Y	N	Y	Y
Offline Initialization	N	N	Y	N	Y	N
Asymmetric Actor-Critics	N	N	Y	Y	N	N
Action Masking	N	N	N	Y	N	N
Online Adaptation	N	N	N	Y	N	Y

Each technique may also address multiple challenges outlined in Section 3.b. Fig II.3 visualizes the relationships between techniques and the challenges they help mitigate. The height of each bar on the left (for techniques) indicates how broadly applicable the technique is - the taller the bar, the more challenges it addresses. Similarly, the height of each bar on the right (for challenges) reflects how much attention that challenge receives, based on the number of techniques that target it. Additionally, the colors of the connecting lines provide key information: they indicate the stage of the RL pipeline at which each technique is applied. From the perspective of a given challenge, this helps identify the most relevant stage in the pipeline for applying domain knowledge to effectively address it.

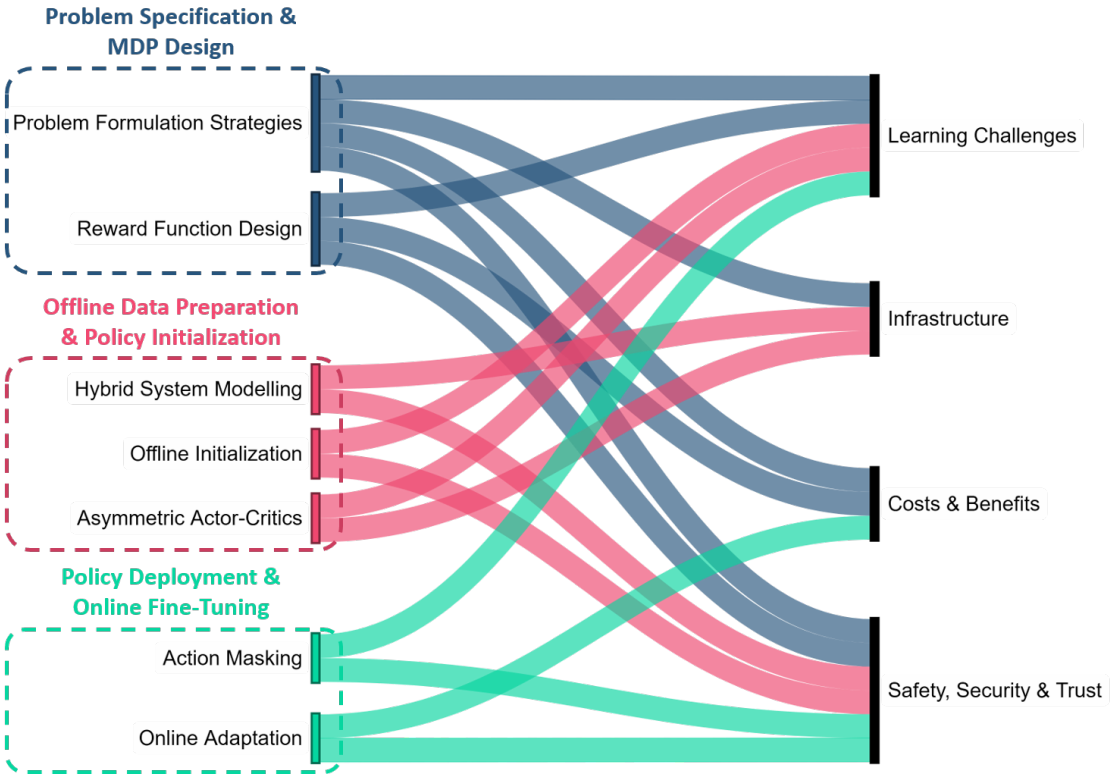

Figure II.3: Challenges addressed by different techniques

The overview highlights how the presented techniques serve as a bridge, linking various forms of domain knowledge to different stages of the RL workflow, with the goal of addressing key challenges in applying RL to BEM. It becomes evident that certain types of domain knowledge are particularly influential - for instance, a strong understanding of real-world system constraints enables the application of multiple techniques. Similarly, some techniques prove especially valuable due to the range of challenges they help overcome; for example, thoughtful problem formulation can mitigate several issues at once. From the challenges' standpoint as well, clear patterns emerge - greater attention is devoted to issues like learning difficulties and safety, mirroring broader priorities in the research community.

4.a Problem Formulation Strategies

The term **Problem Formulation Strategies** serves here as an umbrella for a range of techniques that use domain knowledge to structure the learning problem effectively.

Specifically, the following aspects are considered here:

1. **State Space Design** : The design of the RL policy's state space directly impacts which information signals must be accessible during online deployment. Thoughtful state space formulation can reduce reliance on certain inputs (such as forecasts) or enable the use of alternative knowledge sources, like model-based predictions.

2. **Action Space Design** : The structure of the RL policy's action space defines the range and granularity of control decisions the agent can make during deployment. Careful design of the action space can simplify the learning problem, reduce unsafe or infeasible actions, and align the agent's behavior with real-world operational constraints (such as discrete actuator settings, user overrides, or equipment limitations).

3. **Training and Deployment Architecture** : Architectural decisions, such as whether to adopt centralized or decentralized control, play a key role in determining the practical viability of a solution. In multi-building scenarios, these choices influence the system's scalability, flexibility, and its ability to respect constraints like data privacy.

It should be noted that reward function design and the choice of RL algorithm are not included in the discussion above. Reward function design is addressed in the following section, as it is typically guided by knowledge of the problem's objectives and constraints. In contrast, the design of the state and action spaces is more closely tied to the availability of inputs and the scope of control during deployment. Additionally, once a specific MDP formulation and solution architecture are defined, selecting an appropriate RL algorithm is largely a well-established topic within computer science [125], with limited opportunity for incorporating domain knowledge from other fields.

4.a-i Suitable Domain Knowledge Representations

- **System Constraints**: Formulate problems to avoid private data use; choose interpretable models if explainability is required.

- **Semantic Models**: Use schemas (e.g., Brick) or floorplans to identify relevant sensors and reduce model complexity.

- **User Interaction & Feedback**: Design for occupant comfort and overrides; include feedback signals and ensure graceful handling of interruptions.

4.a-ii Challenges Addressed

- **Learning Challenges**: Domain knowledge simplifies state/action spaces, speeding training and improving generalization [100].

- **Infrastructure**: Align with realistic sensing/actuation limits; minimize computational requirements [97].

- **Costs & Benefits**: Reduce sensor and controller needs [126]; increase acceptance by considering comfort and agency [42].

- **Safety, Security & Trust**: Embed constraints to avoid unsafe policies [102]; allow human overrides [92] (our work); use interpretable models to build trust [127].

Checklist : Informed Problem Design for BEM

To support a structured approach to incorporating domain knowledge during problem formulation, the key insights discussed above are distilled into the checklist below. This serves as a practical guide for domain experts seeking to apply RL, offering a systematic set of considerations to guide their design choices.

Checklist : Informed Problem Design for BEM

- State Space Design
  - ☐ Have I selected only the signals that are realistically available during deployment?
  - ☐ Can any unavailable or noisy signals be replaced with alternate inputs or engineered features?
  - ☐ Have I used semantic or architectural information (e.g., floorplans, equipment schemas) to focus on the most relevant states?
  - ☐ Do the states reflect user interactions (e.g., overrides, feedback) if applicable?
- Action Space Design
  - ☐ Does the action space match the capabilities and limitations of the physical system (e.g., discrete settings, delays)?
  - ☐ Are unsafe or infeasible actions naturally excluded?
  - ☐ Have I considered including actions to accommodate human agency?
- Training and Deployment Architecture
  - ☐ Is centralized, decentralized, or hierarchical control more suitable for my system?
  - ☐ Have I considered constraints like data privacy, communication latency, or hardware limitations?
  - ☐ Will the architecture scale to larger deployments or more zones?

4.b Reward Function Design

The reward function in RL plays a critical role in determining both learning efficiency and the real-world effectiveness of the resulting policy. In principle, RL allows for highly flexible reward formulations - for example, rewards need not be convex or differentiable with respect to policy parameters, and may even be sparse in nature [107]. This flexibility offers a powerful mechanism to embed domain knowledge into the learning process and guide the agent toward desirable behaviors. It should be noted that this approach is intended to complement, rather than replace, established numerical techniques and best practices such as reward shaping and scaling. A detailed overview of such numerical methods for optimizing reward functions is presented in [128].

4.b-i Suitable Domain Knowledge Representations

- **Building Physics**: Use thermodynamic knowledge (e.g., pre-cooling, thermal inertia) and comfort models (e.g., Predicted Mean Vote (PMV)) to guide learning.
- **System Constraints**: Encode operational limits and equipment-friendly behavior (e.g., penalties for excessive switching).
- **User Interaction & Feedback**: Incorporate override signals or feedback to align policies with user comfort and preferences.

4.b-ii Challenges Addressed

- **Learning Challenges**: Physics-informed rewards accelerate learning and mitigate delayed feedback effects [129, 102].
- **Costs & Benefits**: Rewards can balance energy, emissions, comfort, and equipment longevity [61] for holistic optimization [51].
- **Safety, Security & Trust**: Penalties prevent unsafe actions [104]; user feedback improves trust and responsiveness [92] (our work).

Illustrative Example: Knowledge-Informed Reward Design

To make these concepts more concrete, consider a simplified control task for an AC system, where the objective is to maintain thermal comfort while minimizing electricity usage. A typical reward function might only penalize energy consumption and temperature deviations from a fixed setpoint, as shown in Eq II.1.

$$r_t^{basic} = -\lambda_1 \cdot |T_t^{in} - T_t^{set}| - \lambda_2 \cdot P_t^{cool} \tag{II.1}$$

Here, T_t^{in} is the indoor temperature and T_t^{set} is the setpoint at time t, P_t^{cool} is the cooling power used, and λ_1 and λ_2 are positive scaling coefficients. While this may result in a feasible control policy, it ignores several domain-relevant aspects such as human thermal perception, equipment wear-and-tear, and system dynamics. A **knowledge-informed** reward function augments this baseline with additional terms based on domain expertise, as shown in Eq II.2, where the domain-informed terms are highlighted in blue.

$$r_t^{informed} = -\lambda_1 \cdot |T_t^{in} - T_t^{set}| - \lambda_2 \cdot P_t^{cool} - \lambda_3 \cdot |\text{PMV}_t| - \lambda_4 \cdot |\Delta u_t| \tag{II.2}$$

Here, PMV_t is the Predicted Mean Vote, a measure of occupant comfort [130], and Δu_t is the change in the control signal (ie, the action) taken by the agent across subsequent timesteps. The PMV term complements the penalization for a deviation from the setpoint and attempts to keep the indoor conditions at a comfortable level considering multiple factors, while the penalty on large changes in the control signal encourage smoother policies. These additional terms represent the use of domain knowledge at the reward formulation stage that enables the agent to learn more human-centered, efficient and robust policies.

These additions may introduce some complexity - for instance, determining appropriate values for the weights (λ_1 to λ_4) is not always straightforward. However, this complexity also presents an opportunity to apply domain expertise. A knowledgeable practitioner can estimate typical value ranges for each term and select weights that bring them onto comparable scales. Moreover, they can design meaningful trade-offs that better reflect the intended control objectives.

4.c Hybrid System Modelling

Unlike traditional ML, which learns from a fixed dataset, RL requires direct interaction with a system and often millions of iterations to converge. Since this is impractical in real-world systems, agents are typically trained in simulators that model the dynamics and allow unrestricted exploration. In BEM, such simulators (more commonly referred to as **system models**) are widely used and take the form of the computational models described in Section 2. These models span a broad spectrum of fidelity. For instance, high-fidelity tools like EnergyPlus [131] offer detailed representations of building systems but demand significant engineering effort and are computationally intensive, making them less suited for RL training. In contrast, reduced-order models [132] prioritize speed and simplicity by capturing only the essential building physics, but risk oversimplification, potentially leading to a significant simulation-to-reality gap [124] and suboptimal learned policies.

Given these limitations, it is often more effective to **develop a dedicated simulator** tailored to the needs of RL training. This opens up a valuable opportunity to embed domain knowledge into the design process. A well-crafted simulator should aim to:

- Accurately capture the key phenomena that influence the outcomes of RL-based control.
- Represent different system aspects at an appropriate level of detail, balancing complexity and simulation speed.
- Integrate available data to model measurable factors while using domain knowledge to approximate those that are hard to observe directly.

We refer to this process of designing simulators that combine both domain knowledge and data-driven insights as **Hybrid System Modelling**. Unlike traditional system modelling for optimization, which often requires differentiable and analytically tractable formulations, RL-oriented models benefit from a key flexibility: they do not need to be differentiable. This allows the focus to shift toward accurately capturing real-world behaviors without being restricted to specific model forms (e.g., linear or convex structures). However, since RL algorithms typically require millions of interactions, computational efficiency remains essential.

This flexibility also enables the inclusion of complex or often-neglected aspects, such as human behavior and preferences, which are difficult to integrate into conventional optimization models. Furthermore, by appropriately parameterizing the simulator, it becomes possible to enhance policy robustness using domain randomization - a technique where model parameters are varied during training to help the agent generalize to a broader range of real-world conditions [133].

4.c-i Suitable Domain Knowledge Representations

- **Building Physics**: Fundamental laws (e.g., heat transfer, equipment dynamics such as battery chemistry) ground the model in real-world constraints.
- **Semantic Models**: Spatial and architectural information (e.g., zones, connectivity) supports reduced-order or graph-based model construction.
- **Computational Models**: Existing simulation modules (e.g., HVAC, battery) can be reused and extended for flexible hybrid modelling.
- **Operational Strategies**: Occupancy schedules and control rules provide realistic load profiles and system usage patterns.

- **User Interaction & Feedback**: Occupant preferences, overrides, and comfort behaviors capture human-centric dynamics.

4.c-ii Challenges Addressed

- **Infrastructure**: Domain knowledge allows approximation of hidden variables (e.g., internal gains [134]) without additional sensing or instrumentation.
- **Safety, Security & Trust**: Embedding constraints and dynamics ensures safer, more realistic policies [102, 104] and prevents exploitation of simulator artifacts [124].

Illustrative Example: Hybrid System Modelling for Buildings

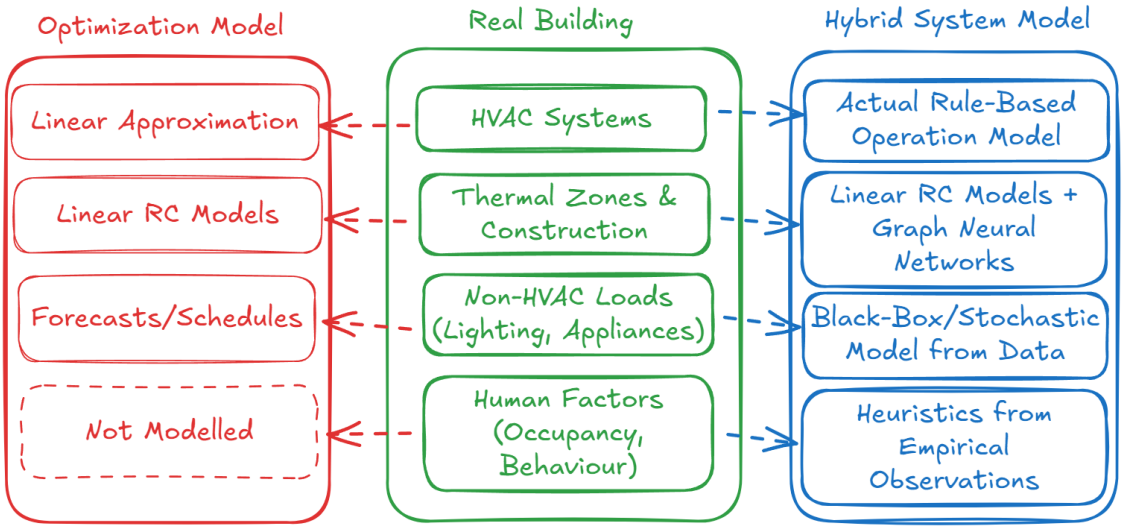

Figure II.4: An example of a Hybrid System Model for a typical building

Fig II.4 illustrates how hybrid system modelling can bridge the gap between idealized optimization models and the complexity of real-world building systems. Such systems comprise diverse and interacting subsystems - HVAC equipment operating under rule-based logic, thermally heterogeneous zones, stochastic loads, and human-driven factors that are difficult to measure or model precisely. A hybrid system model integrates these layers using a mix of methods: actual rule-based logic to emulate HVAC control, graph neural networks to enhance RC-based thermal modelling [135], black-box or stochastic models to represent uncertain non-HVAC dynamics, and heuristic rules derived from empirical observations to capture human factors. This combination provides a richer and more actionable representation of the building environment, enabling learning-based controllers to be trained under more realistic dynamics while still being deployable under limited sensing conditions.

It is important to note that while hybrid system models offer a more representative depiction of real building dynamics, the simpler, analytical models commonly used in optimization formulations have distinct advantages. These models are typically low in complexity and allow the use of highly efficient solution methods beyond reinforcement learning. Moreover, many building phenomena, such as thermal dynamics, can often be captured with sufficient accuracy using simple linear models. Therefore, the added complexity of a hybrid model should be justified by the needs of the application. In cases

where human factors or nonlinear interactions are not critical, a more elaborate model may be unnecessary.

4.d Offline Initialization

Once data is collected (either from a simulator or from the real system), a practical way to reduce learning time is to initialize the agent offline, enabling it to develop a reasonably effective policy before any real-world deployment. This process is referred to as **Offline Initialization** in this thesis. Depending on the availability of a simulator, offline initialization strategies may be broadly categorized into two types as shown in Fig II.5.

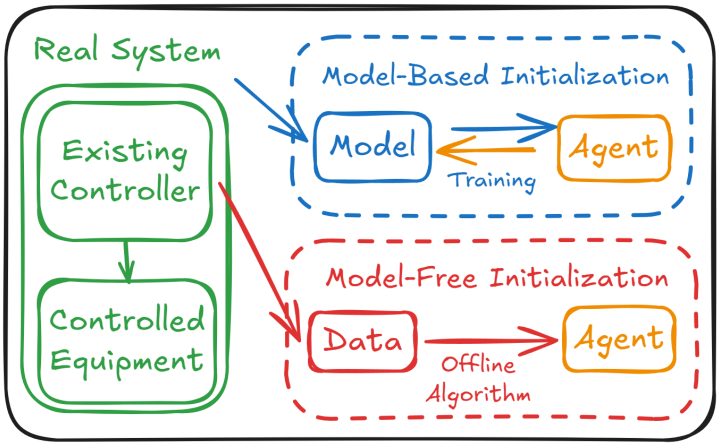

Figure II.5: Different paradigms for offline initialization

1. **Model-Free Offline RL**: This approach assumes access to operational data collected under an existing control policy. The goal is to extract a control policy directly from this data without any further interaction with the system. Several offline RL algorithms have been developed for this purpose [123]. A simple example is Behavioral Cloning, which frames the problem as supervised learning, ie, training the agent to replicate the actions observed in the dataset. More sophisticated methods, such as CalQL, incorporate a reward function into the learning process, enabling the agent to go beyond imitation and learn a more informed policy.

2. **Model-Based Offline RL**: In this setting, the agent interacts with a simulator to learn a policy prior to deployment. Ideally, the simulator is constructed following the hybrid modelling principles discussed in Section 4.c. The primary challenge here lies in developing a simulator that captures the key real-world dynamics with sufficient fidelity to enable effective policy transfer. This approach is a natural extension of Hybrid System Modelling, with the specific objective of facilitating offline training for RL agents.

Regardless of the paradigm followed, the offline initialization process can be improved through the integration of certain types of domain knowledge, as discussed subsequently. A systemic issue with these approaches is that they strongly bias the agent's policy towards those considered optimal in the offline setting. If the existing control policy (from which data was collected) is not optimal, or the simulator is not realistic, this bias can limit the effectiveness of the agent in the real system.

4.d-i Suitable Domain Knowledge Representations

- **Computational Models**: Simulators enable model-based pretraining, providing near-optimal policies when fidelity is high, but risking bias when fidelity is low.
- **Operational Strategies**: Data from existing control policies guides offline initialization. Broader datasets from multiple strategies improve generalization and reduce bias.

4.d-ii Challenges Addressed

- **Learning Challenges**: Offline initialization accelerates training, avoids early-stage instability, and reduces the need for real-time interaction [92] (our work).
- **Safety, Security & Trust**: Pretraining prevents unsafe exploratory actions and allows policy validation before deployment [72].

Illustrative Example: Behavioural Cloning

The application of an offline initialization strategy, called "behavioural cloning" [123], to a coordination problem in BEM is described in detail in our previous work [91]. The problem involves the control of BESS systems for a system of 5 buildings (each with their own non-controllable loads) in response to dynamic electricity tariffs [51]. At every time step, the controller receives data from each building (loads, PV, battery State-of-Charge (SOC)), along with the global context (time of year/day, weather, electricity prices), and sends the desired charge/discharge level for each BESS.

To bootstrap the learning agent, a schedule-based RBC was first deployed in the system, and its operational data collected. Then, the DNN representing the policy of a DRL agent was initialized offline to mimic the RBC prior to deployment, and allowed to train further through direct interaction. The results in [91] highlight a 3.8% cost reduction compared to the RBC, and a 20% cost reduction compared to an RL agent without the initialization step.

4.e Asymmetric Actor-Critics

In Actor-Critic RL algorithms, the agent is composed of two distinct components: the actor, which selects actions based on the current state, and the critic, which evaluates the quality of those actions by estimating a value function. A key characteristic of this architecture is that the critic is only used during training to guide the learning of the actor. During deployment, only the actor is retained and used to make decisions. This separation presents a unique opportunity to incorporate domain knowledge during training without affecting real-time operation. Specifically, one can provide additional inputs such as model-derived estimates, unobservable internal states, or perfect future forecasts - to the critic only, while keeping the actor restricted to inputs that will be available during deployment. These additional inputs enable the critic to make more accurate value estimates, which in turn yield better gradient signals for updating the actor's policy (when using DNNs to represent actors and critics).

Architectures that leverage this idea, where the actor and critic are trained using different input spaces, are referred to as Asymmetric Actor-Critic architectures [68, 136]. By enriching the critic with privileged or model-based information during training, these architectures enable the actor to learn a higher-quality policy than would be possible using observable inputs alone. The approach also allows domain knowledge to directly inform the learning process at the policy level.

4.e-i Suitable Domain Knowledge Representations

- **Computational Models** (*required*): Simulators provide hidden variables to the critic, improving value estimates and actor training.
- **System Constraints**: Actor uses only real-world observable signals, while the critic accesses richer simulator-based information.

4.e-ii Challenges Addressed

- **Learning Challenges**: Extra critic information accelerates convergence and improves policy quality under delayed feedback and complex dynamics [68].
- **Infrastructure**: Keeps real-world sensing simple by limiting actor inputs, while still leveraging richer simulator data during training.

Illustrative Example: Asymmetric Actor-Critics for AC Control

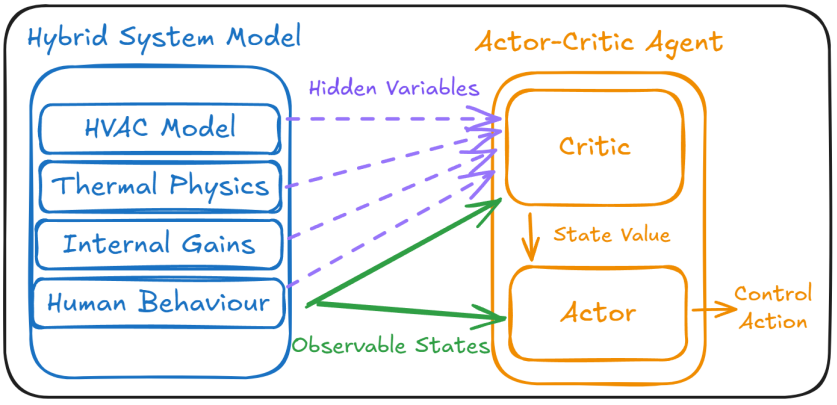

Figure II.6: Asymmetric actor-critic architectures using a system model

Fig II.6 illustrates an asymmetric actor-critic RL architecture applied to a building control problem. The environment is represented using a hybrid system model that integrates various components influencing building dynamics, as discussed in the previous section. Within this setup, certain variables are observable through typical sensing infrastructure in real buildings. These include indoor air temperature, outdoor weather conditions (such as temperature and humidity), HVAC control signals or energy usage, and possibly coarse indicators of occupancy. These observable states form the input to the actor network, which is intentionally restricted to data that would realistically be available during deployment.

However, many influential variables are hidden or difficult to measure in practice. Examples include the internal thermal state of building materials (e.g., heat stored in walls or floors), latent heat gains from human activity or plug loads, and fine-grained human decision-making. These variables are available only within the simulation and are not accessible during real-world operation. During training, the critic network is granted access to these hidden states, allowing it to more accurately estimate value functions and guide the actor's policy updates. By leveraging high-fidelity simulations during training but enforcing realistic observability at deployment, asymmetric actor-critic architectures strike a balance between learning performance and practical feasibility. This makes them especially well-suited to applications in building energy management, where full state observability is often infeasible.

4.f Action Masking

Domain knowledge can be incorporated into the control policy as simple rules that ensure safety by imposing state-dependent constraints on the action space. These constraints are implemented using action masks, which dynamically restrict the agent's available actions based on the current state. While action masking is commonly used to reduce problem complexity in games [67], here it serves to embed expert knowledge, guiding the agent away from unsafe or undesirable behaviors during both training and deployment. This approach requires no formal mathematical models - rules can be encoded as logic or decision trees - and integrates seamlessly with standard neural network training, as masks can be applied directly within the agent without disrupting backpropagation [137].

It is worth noting that when an analytical model of the system is available, more rigorous constraint-enforcement techniques can be employed. For instance, methods such as Action Projection [138] utilize these models to adjust the actions proposed by the RL agent, projecting them into a predefined "safe" set by solving a quadratic program at each timestep. These techniques can also be interpreted as mechanisms for incorporating domain knowledge into the learning process. However, their applicability is limited by the requirement for differentiable analytical system models - an important consideration already discussed in Section 4.c.

4.f-i Suitable Domain Knowledge Representations

- **Building Physics**: Use rule-based constraints (e.g., prevent heating when temperature is already high) to ensure physically plausible actions.
- **System Constraints**: Encode operational and regulatory limits (e.g., battery SOC bounds, HVAC off-hours) through action masks.

4.f-ii Challenges Addressed

- **Learning Challenges**: Action masks narrow exploration to safe regions, improving sample efficiency and speeding convergence [139] (our work).
- **Safety, Security & Trust**: Enforces safety and compliance in both training and deployment, improving reliability and user trust [139] (our work).

Illustration: Action Masking Implementation

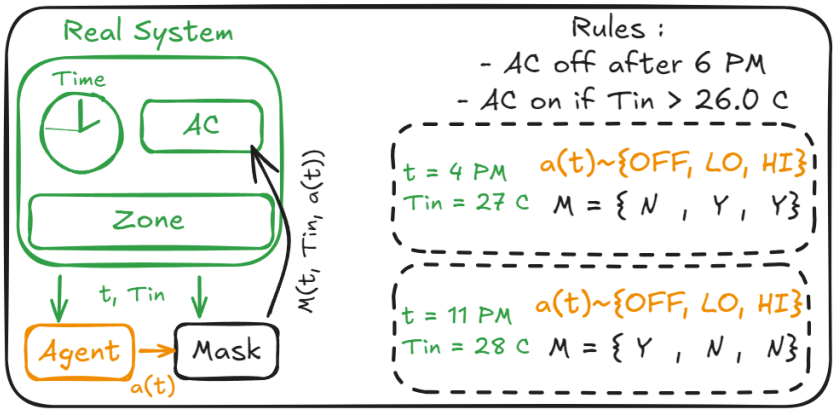

Figure II.7: Action masking applied to an AC control problem

Fig II.7 shows an action mask applied to a simple AC control task, where the controller is required to follow certain rules throughout its operation. At every time step t, the state of the system is composed of the current time and the indoor temperature T_t^{in}. The action space is discrete with 3 choices - switch off the AC (OFF), operate at low power (LO), or operate at high power (HI). The RL agent chooses a control action a_t based on these inputs and the permitted actions defined by the masking operation $M(\cdot)$. In the example, M is implemented as a binary mask that allows (Y) or bans (N) each action, which is dynamically calculated based on the state of the system. M disallows those actions that would violate the rules (based on an implicit understanding of the problem dynamics), thus ensuring that they are respected at all times.

4.g Online Adaptation

As previously discussed, directly training an agent on a real-world system is typically impractical for control tasks like BEM. Even with extensive pre-training using realistic simulators or historical data, and incorporating as much domain knowledge as possible, the deployed agent may still perform sub-optimally in practice. This is primarily due to two key reasons. First, there is always a mismatch between simulation and reality. Agents trained in simulation often fail to account for unmodeled phenomena or real-world variability, leading to degraded performance when deployed. Second, real buildings are inherently dynamic systems. Over time, they undergo changes such as retrofits, shifting occupancy patterns, evolving weather conditions, or equipment degradation. As a result, a policy that performs well at one point may become ineffective later.

To address these challenges, it is natural to allow the agent to adapt online - continuously learning and updating its behavior based on real-time data from the actual system. This process, referred to in the thesis as Online Adaptation, also creates a valuable opportunity to incorporate domain knowledge directly from real-world signals during the agent's learning process. In particular, this is achieved through the use of dedicated algorithms and alternate reward formulations during the online learning stage.

The process of deploying a policy trained offline onto a real system, followed by its refinement through online learning, is commonly referred to in the RL literature as Offline-to-Online RL [122]. In practice, this transition is known to pose several challenges. One such issue is catastrophic forgetting, where the agent's performance drops sharply upon switching to online learning and requires extensive real-world interactions to recover. These difficulties often arise due to distributional shifts in the state space caused by the gap between simulation and reality. While this thesis does not attempt to solve such RL-specific challenges - which remain an active area of research - it instead focuses on how existing Offline-to-Online RL frameworks can serve as a mechanism for injecting domain knowledge during the online adaptation phase of the control pipeline.

4.g-i Suitable Domain Knowledge Representations

- **System Constraints**: Fine-tune policies in the real world to correct violations that were not captured in simulation.
- **User Interaction & Feedback**: Use simple feedback signals (e.g., frequent overrides) to adapt policies toward occupant preferences.

4.g-ii Challenges Addressed

- **Costs & Benefits**: Online adaptation improves efficiency and enables personalization to occupant-specific preferences [92] (our work).

- **Safety, Security & Trust**: Ensures constraints are respected after deployment and builds trust through human feedback [92] (our work).

Illustrative Example: Online Adaptation

Our earlier work [92] applied online adaptation to enhance the acceptability of a DR scheme controlling AC setpoints for 100 consumers. In the absence of indoor temperature sensing infrastructure, a binary feedback signal, indicating whether a consumer override occurred in the previous step, was used. This signal was incorporated as a penalty term in the reward function of a DRL agent during online interaction. As further detailed in Chapter IV, the results demonstrate that this approach effectively reduces the frequency of overrides.

5 Comparison to Existing Knowledge-Integrated Frameworks

Section 3 in Chapter I provided an overview of the main pathways for integrating knowledge-driven and data-driven techniques for control applications, with a focus on demand-side building energy management. In this section, we compare the proposed Knowledge-Informed RL paradigm with existing approaches in the literature, highlighting key conceptual and methodological differences. We mainly highlight two features of Knowledge-Informed RL as below, to facilitate the comparison.

Knowledge-Informed RL is Domain Specific

The proposed approach seeks to leverage all forms of domain knowledge relevant to a given problem and provides techniques tailored to address domain-specific challenges. As a result, Knowledge-Informed RL is inherently domain-specific, meaning that techniques effective in one field (eg, BEM) may not directly translate to another (eg, robotics [119]).

In contrast, alternative approaches such as Physics-Informed RL [140] emphasize one class of knowledge representations (ie, physics) that are shared across domains, enabling broader transferability. However, real-world systems involve multiple interacting processes, of which physics typically captures only a subset. Knowledge-Informed RL can therefore incorporate insights from Physics-Informed RL as needed, while remaining more flexible in its knowledge representations but narrower in its domain-specific focus.

Focus on RL for Optimization

The proposed framework focuses on RL as the optimization paradigm and examines how domain knowledge can address challenges that hinder its real-world adoption. This is complementary to the approaches reviewed in Chapter I (Fig I.9), which enhance other optimization methods, such as linear programming, through data-driven techniques. While this thesis is limited to RL, analogous frameworks may be developed for model-based optimization workflows that integrate data and domain knowledge, representing a natural direction for future work.

6 Discussion and Summary

In summary, this chapter presents the motivation for learning-based approaches in BEM, with a particular focus on the potential of RL to address the practical challenges faced by practitioners. It outlines the key limitations of applying RL in this context and identifies domain knowledge as a powerful tool for overcoming many of these issues. This forms the basis of the paradigm of **Knowledge-Informed RL for BEM**, anchored by a critical discussion on the meaning and role of domain knowledge in this field. A range of techniques is introduced to demonstrate how different types of domain knowledge can be systematically integrated into various stages of the RL workflow. Each technique is suited to specific knowledge contexts and targets one or more identified challenges. The chapter concludes with a high-level overview of the entire framework - linking knowledge types, techniques, and challenges - to provide a clear and cohesive understanding of how domain knowledge can be leveraged to strengthen RL-based control in building energy management.

It is important to acknowledge that several simplifications and abstractions were made in this chapter to present the techniques and challenges in a unified framework. Some techniques are inherently broader in scope and therefore capable of addressing a wider range of challenges. Similarly, certain challenges (particularly those related to learning) encompass diverse and complex issues that have entire subfields dedicated to them within the RL community. Despite these caveats, the real strength of the framework lies in its emphasis on combining techniques and knowledge representations in a context-aware manner. **No single technique can effectively address all challenges or leverage every form of domain knowledge.** Instead, the recommended approach is to select and apply a tailored subset of techniques based on the specific problem context and control objectives. The following chapters demonstrate how this can be done through two realistic case studies, each focused on a different set of challenges commonly encountered in BEM.

Chapter III

Application on a Single-Building Control Problem

This chapter presents the application of the proposed techniques to a single-building control problem. Following a detailed description of the control scenario, the problem is analyzed through the lens of knowledge-informed RL, and different techniques are studied in this context. The work discussed in this chapter builds on published studies on offline initialization [91] and action masking [139] conducted as part of this thesis.

1 Problem Overview

The control task is to manage Air-Conditioner and Battery Energy Storage System operation in a simulated office building to maintain indoor thermal comfort and minimize energy costs. The building is equipped with a grid-connected PV system and operates under a real-time electricity pricing contract, allowing electricity to be bought or sold at prevailing market rates. At each timestep, the controller selects the indoor temperature setpoint and battery power, which are sent to the existing Building Management System. Downstream systems within the BMS then actuate the requested setpoint and battery power levels. The overall control objective is cost-driven, while ensuring thermal comfort satisfaction.

Fig III.1 provides an overview of the system components. At each timestep t (with a 10-minute control interval), the agent issues control signals P_t^{bess} and T_t^{set} to operate the BESS and AC systems, respectively. We consider only cooling in this study, and assume that the building is located in a tropical region without heating loads. An external price signal λ_t specifies the cost of consuming electrical energy in the next timestep. The agent is billed for the net load $P_t^{grid} \cdot \Delta t$ at every timestep - in this formulation, any energy exported to the grid is also remunerated at the prevailing price λ_t. The power balance equation in Eq III.1 connects the various power flows at each time step. Here, P_t^{load} is the building demand (due to lights, appliances and AC systems), and P_t^{pv} is the PV generation at time t. Positive values of P_t^{grid} denote imports from the grid, and positive P_t^{bess} denotes that the battery is charging (consuming energy).

$$P_t^{grid} + P_t^{pv} = P_t^{load} + P_t^{bess} \tag{III.1}$$

The building is equipped with sensors and metering devices that record indoor temperature and humidity (T_t^{in}, RH_t^{in}), outdoor conditions (ambient temperature T_t^{amb} and relative humidity RH_t^{amb}), battery status (SOC_t^{bess}), electrical loads, and PV generation.

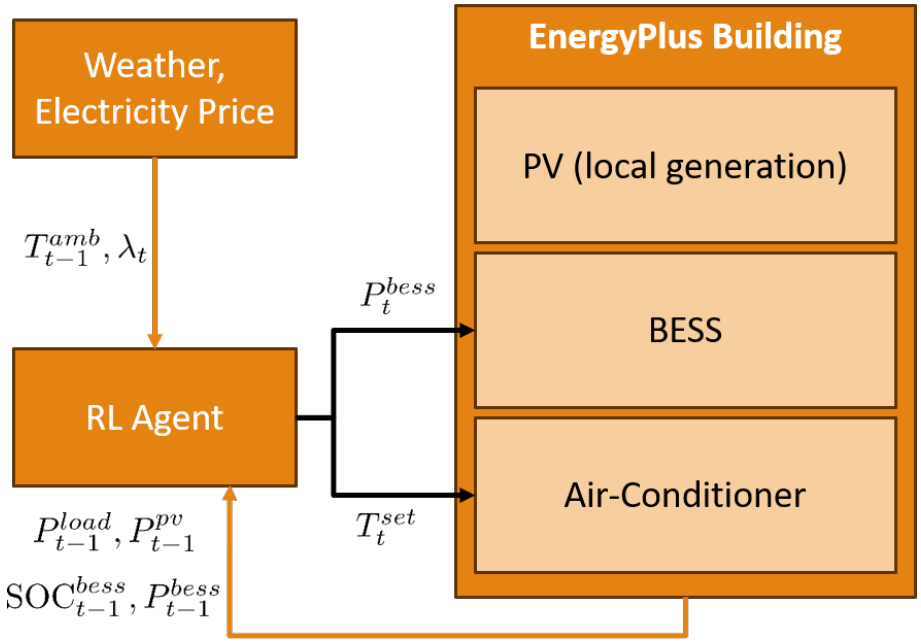

Figure III.1: Schematic of Control Problem

The controller is able to access the latest sensor readings (ie, those from time $t-1$) to decide the control action.

As noted by [141], this problem formulation is particularly relevant today, as advanced control strategies that can respond to real-time price and weather signals remain a key challenge for PV-integrated BESS systems. The building is simulated in EnergyPlus [131], a high-fidelity platform capable of accurately modeling construction features, AC systems, PV panels, BESS operation (including inverter losses), and internal heat gains. The simulation is based on a standard reference building within EnergyPlus, and the tool handles aspects such as AC sizing based on a provided weather file. For the purposes of this study, the simulation is treated as the real building, and no separate computational model is available.

1.a Learning Objectives

The objective is to jointly optimize battery operation and temperature setpoints to leverage the real-time tariff structure while ensuring safe operation. The key trade-off is between the energy bill (denoted as $\sum_t b_t$, where b_t is the step cost, $\lambda_t \cdot P^{grid}_t \cdot \Delta t$, at time t), and the thermal comfort of the occupants during working hours. Table III.1 lists the evaluation metrics used to assess controller performance. Occupant thermal comfort is quantified using the PMV [142], which accounts for factors such as radiant temperature, air speed, clothing, and occupant activity to produce a score ranging from -3 (too cold) to +3 (too hot). According to international thermal comfort standards, such as ASHRAE Standard 55 [4], indoor PMV should be maintained between -0.5 and 0.5 to ensure acceptable comfort levels.

1.a-i Evaluation Period and Safety Rules

The controller's performance is evaluated over a full year. We assume that no prior operational data from the building is available - as such, the evaluation includes all phases, such as data collection and online model training, within the assessment period. This

Table III.1: Metrics (Over Test Period)

Metrics	Equation
Electricity Bill	$\sum_t b_t = \sum_t P_t^{grid} \cdot \lambda_t$
Thermal Comfort	$\frac{1}{T} \sum_t \lvert \text{PMV} \rvert_t$

provides a realistic measure of performance and is particularly important for RL, where the initial deployment time and the costs incurred during learning can substantially offset any benefits realized later in operation [143].

Throughout the evaluation period, controllers are required to respect certain safety rules outlined by the equipment owners. These rules are aimed at minimizing equipment damage and preventing unacceptable levels of discomfort to the occupants in the building. The rules here, adapted from our earlier work [139], are:

1. The AC must be switched off outside of working hours.
2. The indoor temperature must remain between 20°C-27°C during working hours.
3. The battery SOC (a measure of the charge available in the battery relative to the maximum charge) must remain within 20% to 90% at all times.

Another key constraint is that the BMS only accepts discrete control inputs (for example, [144] describes a BESS management interface based on discrete operating modes rather than continuous control signals). For T_t^{set}, there are four allowable setpoints between 22 ^{o}C and 25 ^{o}C, plus an ON/OFF flag. For P_t^{bess}, there are five discrete charging levels, five discharging levels, and one idle level.

1.a-ii Existing Rule-Based Controller

An existing rule-based controller, designed for a static electricity pricing scheme, charges the battery using surplus PV generation during midday and discharges it to meet demand when local generation is insufficient. Indoor temperature setpoints are determined using pre-existing schedules in this case. The operation of the RBC may be summarized as shown in Algorithm 1. As discussed in a recent comparative study of BESS controllers [45], this corresponds to the widely used "Self-Consumption Maximization" strategy, which was found to perform close to a model-based optimization approach in the comparison.

2 Problem Analysis

This section analyzes the problem statement through the lens of Knowledge-Informed RL. We first identify the relevant challenges and available domain knowledge representations, and then use these insights to select a set of suitable techniques, following the discussion in the previous chapter.

2.a Relevant Challenges

1. **Learning Challenges**: The controller must learn online and outperform the baseline within one year[1], despite limited exploration, continuous state spaces, and large action spaces.

[1]Since there is no separate train and test period

Algorithm 1 Existing Rule-Based Controller Algorithm

Require: AC Setpoint Schedule $T_t^{set,ref}$, Battery Maximum Charge/Discharge Power $P_t^{bess,max}$

```
while True do
    T_t^set ← T_t^set,ref
    if P_t^pv >= P_t^load and SOC_t^bess < 0.9 then
        P_t^bess ← min(P_t^bess,max, P_t^load − P_t^pv)        ▷ Charge to Store Excess PV
    else if P_t^pv < P_t^load and SOC_t^bess > 0.2 then
        P_t^bess ← max(−P_t^bess,max, P_t^load − P_t^pv)       ▷ Discharge to Meet Excess Load
    else
        P_t^bess ← 0
    end if
    t ← t + 1
end while
```

2. **Costs & Benefits**: The approach should deliver net economic savings while ensuring occupant comfort and safe equipment operation.

3. **Safety, Security & Trust**: Exploration may lead to unsafe or undesirable actions; policies must respect comfort, equipment limits, and safety rules.

2.b Available Domain Knowledge

1. **Building Physics**: Well-known thermal and battery models can guide safe operation, even without detailed AC dynamics.

2. **System Constraints**: Controllers must respect predefined rules and BMS limitations, which restrict allowable actions.

3. **Existing Strategies**: The baseline controller offers a safe but suboptimal policy that can be leveraged to speed up learning.

3 Methodology & Technique Selection

The solution strategy in this work follows the extended RL workflow described in Section 3, applying different techniques at each stage to effectively incorporate domain knowledge and enhance the solution. Fig III.2 outlines how domain knowledge is incorporated in each section of this workflow, and the implementation details are provided in the rest of the section.

3.a Problem Specification and MDP Design

In the first stage of the workflow, the goal is to formalize the control task as an RL problem by framing an MDP consisting of states, actions and rewards. Two techniques are employed at this stage to incorporate domain knowledge - Problem Formulation Strategies, and Reward Function Design.

3.a-i Problem Formulation Strategies

The state and action spaces of the RL agent are defined according to the available sensor data and the equipment under control. Since the BMS accepts only discrete control

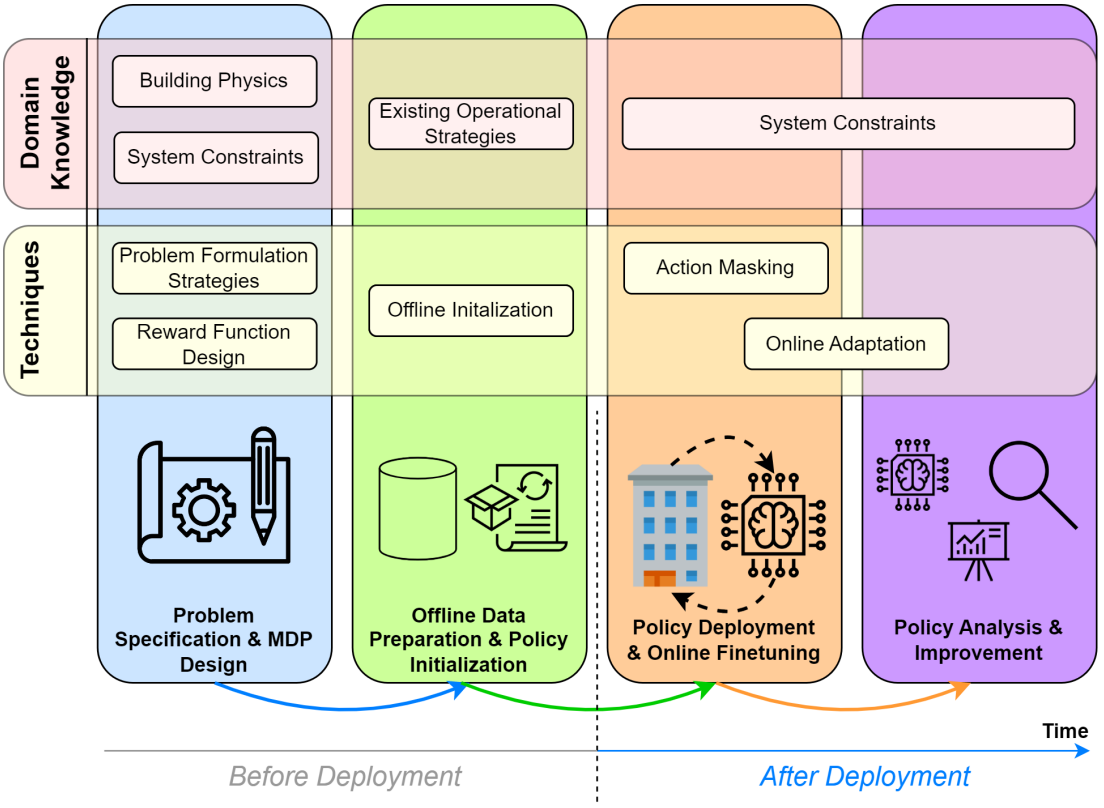

Figure III.2: Knowledge-Informed RL Workflow for Single-Building Problem.

inputs, this constraint can be naturally addressed by using a discrete action space. Additionally, the charge and discharge limits ($P_t^{bess,max}$) are enforced by setting the maximum values in the discretized action space accordingly. The state and action spaces of the MDP thus formulated are shown in Tables III.2 and III.3 respectively. It should be noted that the AC OFF action is implemented as setting $T_t^{set} = 33.0°\text{C}$, and $P_t^{bess} = 0.0$ kW puts the battery into the idle mode.

Table III.2: State Space of the MDP

State	Notation
Time	t
Weather	T_{t-1}^{amb}, RH_{t-1}^{out}
Grid Price	λ_t
Load	P_{t-1}^{load}
PV Generation	P_{t-1}^{pv}
Battery Power	P_{t-1}^{bess}
Battery SOC	SOC_{t-1}^{bess}
Indoor Conditions	T_{t-1}^{in}, RH_{t-1}^{in}

Table III.3: Action Space of the MDP

Action	Notation
Indoor Setpoint	$T_t^{set} \in \{22, 23, 24, 25, 33\}$
Battery Operation	$P_t^{bess} \in \{-12.5\text{kW to } 12.5\text{kW (11 levels)}\}$

The above formulation results in a continuous state space **S** (with infinitely many states), and a discrete action space **A** with 55 distinct actions arising from different setpoints and battery power combinations.

3.a-ii Reward Function Design

Since the problem involves a trade-off between two metrics, a common approach is to combine the different terms (e.g., thermal comfort, monetary cost) using sum [87, 74, 145] or product [139] (our work) operations, thereby capturing their combined significance in the reward function. Here, selecting appropriate scaling constants is challenging, as the quantities being combined often differ in both units and magnitude [146].

In this context, domain knowledge can be used in the scaling and normalization process, to bring different quantities into a common, comparable range. We first define a scaling function $f^{scale}(x, x^-, x^+) : \mathbb{R}^3 \to [-1, 0]$ as shown in Eq III.2.

$$f^{scale}(x, x^-, x^+) = \text{clip}_{[-1,0]}[\frac{x - x^+}{x^+ - x^-}] \tag{III.2}$$

f^{scale} linearly maps x from the interval $[x^-, x^+]$ to $[-1, 0]$ and clips values outside this range. Here, x represents a quantity of interest in physical units, x^- a highly undesirable value, and x^+ a highly desirable value. For example, if x denotes P_t^{grid} (in kW), a domain expert may set $x^- = 20$ kW (ie, import) and $x^+ = -20$ (ie, export) kW to denote that importing from the grid is undesirable.

We study two reward function formulations (with different types of knowledge integration) in this study, denoted as R^A and R^B, where $R^i : S \times A \to \mathbb{R}$ for $i \in \{A, B\}$. R^A (shown in Eq III.3) is based on the step bill, $b_t = P_t^{grid} \cdot \lambda_t \cdot \Delta t$ (cents) , and the indoor temperature T_t^{in}, which is used as a proxy for thermal comfort (as in [145]). Here, C_1 and

C_2 are scaling constants - in all experiments, they are set to 0.5 to convey equal importance to both reward terms. $\mathbf{I}^{occupied} \in \{0, 1\}$ is the indicator function denoting whether the building is occupied. The limit terms passed to the f^{scale} function in each term have physical meanings, and can be easily designed by a domain expert - here, they convey that a step cost of 50 cents, or a 3°C deviation from a known comfortable temperature (24 ^{o}C), are highly undesirable.

$$R^A(b_t, T_t^{in}) = C_1 \cdot f^{scale}(b_t, 50, -50) + \mathbf{I}^{occupied} \cdot C_2 \cdot f^{scale}(|T_t^{in} - 24.0|, 3, 0) \quad \text{(III.3)}$$

R^B adopts an alternative approach in which specific behaviors, known to be effective strategies, are explicitly promoted. In a dynamic pricing setting, a common method to improve profits is to use the battery for arbitrage, i.e., to charge during low-price periods and discharge during high-price periods [39]. To quantify the arbitrage opportunity, a domain expert first defines lower and upper threshold prices, λ^- and λ^+. When $\lambda_t = \lambda^-$ (λ^+), the ideal arbitrage action (denoted as $P_t^{bess,arb}$) is to charge (discharge) at the maximum possible power, $P^{bess,max}$. $P_t^{bess,arb}$ may be linearly interpolated in the range $\lambda_t = (\lambda^-, \lambda^+)$.

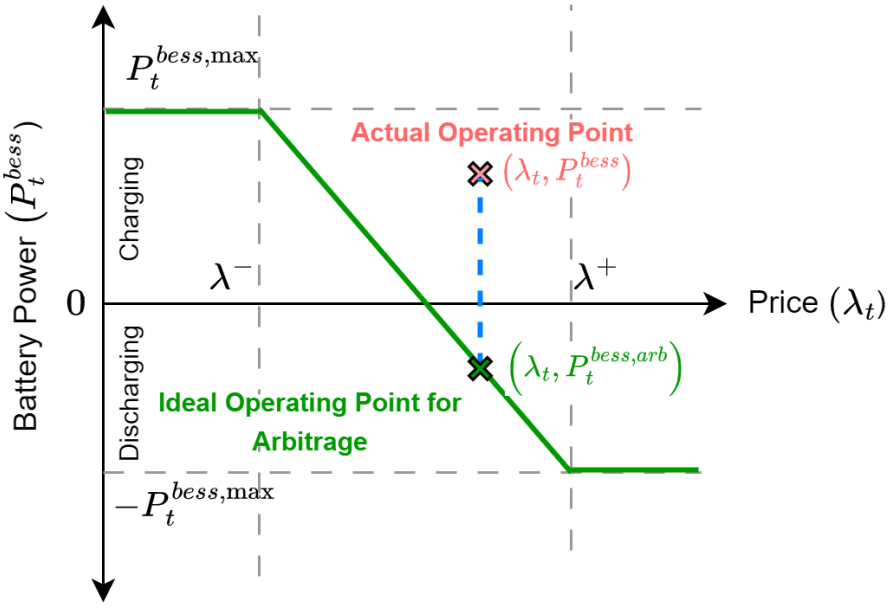

Figure III.3: Illustration of ideal operating curve for arbitrage, with an example $P_t^{bess,arb}$ highlighted.

Fig III.3 visualizes this ideal operating curve, and highlights an example value of $P_t^{bess,arb}$. An arbitrage cost, measured in kW, may thus be formulated using the distance between P_t^{bess} and $P_t^{bess,arb}$. Such a penalty term can be incorporated into R^B as shown in Eq III.4. Here, an additional point of difference from R^A is the use of the PMV as a measure of thermal discomfort, rather than using only T_t^{in} as a proxy. The PMV here is estimated based on T_t^{in} and RH_t^{in}, assuming fixed values for other terms such as air velocity, rate of metabolism and clothing level. As before, $C_1 - C_3$ are scaling constants set to equal values.

$$\begin{aligned} R^B(b_t, T_t^{in}, \mathrm{RH}_t^{in}, \lambda_t, P_t^{bess}) = C_1 \cdot f^{scale}(b_t, 50, -50) \\ + \mathbf{I}^{occupied} \cdot C_2 \cdot f^{scale}(|\mathrm{PMV}|, 1, 0) \\ + C_3 \cdot f^{scale}(|P_t^{bess} - P_t^{bess,arb}|, 5, 0) \end{aligned} \tag{III.4}$$

Both reward formulations illustrate how domain knowledge can enhance standard reward design in RL. Since all reward functions inherently reflect problem objectives, they already embed domain knowledge to some extent [107]. The use of f^{scale} makes normalization [128] intuitive for domain experts by relying on clear physical limits. Moreover, as shown in R^B, rewards can be explicitly shaped to promote behaviors known to be beneficial.

3.b Offline Data Preparation and Policy Initialization

Since RL relies on trial and error, agents typically start with a random policy that is refined over time based on rewards. This is not feasible here because the agent controls real-world equipment, and erratic actions could cause damage. Therefore, the agent must be initialized before its first interaction with the building system.

3.b-i Offline Initialization

Although no historical data is available for offline training, the existing controller can be operated for an initial period during evaluation to collect a dataset, which can then be used to initialize the RL agent. This data-driven offline initialization [123] serves to preserve the knowledge embedded in the pre-existing operating strategy. In this work, the CalQL algorithm [147] is employed to train an RL controller using data gathered from running the RBC in the system for D^{init} days. The offline initialization process is outlined in Algorithm 2.

Algorithm 2 Steps for Offline Initialization

Require: Existing Controller $\pi^{existing}$, Data Collection Period D^{init} days, Reward Function R
Initialize dataset $\mathcal{D} \leftarrow \emptyset$
for $t \leftarrow 1$ to D^{init} days **do**
  Observe current state s_t from building sensors
  Select action $a_t \leftarrow \pi^{existing}(s_t)$
  Apply a_t to the building system via the BMS
  Observe next state s_{t+1} and reward $r_t = R(s_t, a_t, s_{t+1})$
  Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$
end for
Train initial RL policy π_θ using $\mathcal{D}$ with the CalQL algorithm
Deploy π_θ from day $D^{init} + 1$ until the end of the evaluation

The CalQL algorithm leverages the reward function to more accurately evaluate the actions taken by the RBC, meaning the resulting offline policy may even outperform the original controller. Consequently, the choice of D^{init} is critical: a smaller value allows a potentially superior learned controller to operate for a longer period, but may also limit the dataset size, reducing the quality of the offline policy.

3.c Policy Deployment and Online Fine-Tuning

During the online control phase, when the RL agent is actively interacting with the building, both safety and performance can be enhanced using Action Masking and Online Adaptation, as detailed below.

3.c-i Action Masking

Hard operational rules that the controller must follow throughout deployment can be enforced by designing action masks that restrict the set of allowable actions based on the current system state. While some constraints, such as AC operating hours or maximum charge/discharge power, can be addressed during problem formulation, action masking offers a more flexible mechanism for ensuring compliance with these rules. The action mask used in this work is described in Table III.4.

Table III.4: Description of Action Masks

Rule	Condition	Masked (Disallowed) Actions
Indoor temperature < 27°C during working hours	$t \in$ working hours and $T^{in}_{t-1} > 26.5^o$C	$T^{set}_t = 33.0$
Battery SOC always above 20%	SOC^{bess}_t <= 0.21	$P^{bess}_t < 0$ (discharging)
Battery SOC always below 90%	SOC^{bess}_t >= 0.89	$P^{bess}_t > 0$ (charging)

3.c-ii Online Adaptation

Although a functional control policy can be achieved using offline initialization and action masking, the RL agent can potentially improve its performance over time by continuing to learn from interactions with the system during deployment. This can be accomplished by collecting operational data and periodically (ie, every D^{upd} days) updating the agent's parameters with the latest information. To realize these gains, the agent must explore actions not encountered in earlier training, making some level of exploration essential. This, in turn, requires a mechanism to balance exploiting existing knowledge with trying new actions.

As the algorithm used in this work (CalQL) trains a neural network to predict the Q-value of each action for a given state, a degree of stochasticity in action selection is introduced by applying a softmax distribution over the Q-values and sampling from the resulting probability distribution [148]. The level of exploration is controlled by a temperature parameter τ: higher values produce action selections that are nearly random, while lower values make the policy increasingly greedy with respect to the estimated Q-values. This formulation is also compatible with action masking, since the probabilities of masked actions can be manually zeroed out prior to sampling the action. The overall process is outline in Algorithm 3

3.d Plan of Experiments

To systematically evaluate the impact of each technique, experiments were conducted by testing all possible combinations of the techniques, as shown in Table III.5.

Algorithm 3 Online Adaptation with Softmax Action Selection and Action Masking

Require: Initial RL policy π_θ, Update Interval D^{upd} days, Temperature τ, Reward Function R
Initialize dataset $\mathcal{D} \leftarrow \emptyset$
for each day d in deployment period **do**
  for each timestep t in day d **do**
    Observe current state s_t from building sensors
    Obtain Q-values $Q(s_t, a)$ for all actions a using π_θ
    Apply action mask $M(s_t)$ to zero out probabilities of invalid actions
    Compute probabilities $p(a|s_t) = \frac{\exp(Q(s_t,a)/\tau)}{\sum_{a'} \exp(Q(s_t,a')/\tau)}$ for unmasked actions
    Sample action $a_t \sim p(a|s_t)$
    Apply a_t to the building system via the BMS
    Observe next state s_{t+1} and reward $r_t = R(s_t, a_t, s_{t+1})$
    Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$
  end for
  if $d \bmod D^{upd} = 0$ **then**
    Update policy parameters θ using the latest $\mathcal{D}$ with the CalQL algorithm
  end if
end for

Table III.5: Table of experiments for each reward function

Index	Offline Initialization	Action Masking	Online Adaptation	Description
Baselin	N	N	Y	Direct RL for comparison
1	N	Y	Y	#1 + action masks
2	Y	N	N	Offline initialization only
3	Y	N	Y	#3 + continuous learning
4	Y	Y	N	#3 + action masking
5	Y	Y	Y	All techniques used

The baseline for all comparisons is deploying the RBC throughout the evaluation period. Because the RBC is a fixed, rule-based program, none of the proposed techniques can be applied on top of it. Consequently, these combinations are excluded from Table III.5. Additionally, for clarity, any RL-based results displayed in the rest of the chapter correspond to the reward function R^B unless otherwise stated.

4 Results & Analysis

The experiments were conducted over an evaluation period between January and December using a co-simulation approach with the EnergyPlus simulation. Typical Meteorological Year (TMY) weather data for Singapore, collected through the diyepw Python library [149], was used as an input to the simulation. Additionally, wholesale electricity price data at half-hourly resolution was obtained from the Singapore Energy Market Authority (EMA) website.

For RL-based runs, default hyperparameters were used for all algorithms, based on their implementations in the d3rply Python library [150]. The value of τ was gradually reduced from 0.1 to 0.01 over a period of 100 days from the start of the RL policy deployment, and different combinations of D^{init} and D^{upd} were tested. The training process was repeated for each reward function - additionally, for runs with online adaptation, the results are reported for three runs with different random seeds [151].

4.a Numerical Results

Table III.6 shows the metric values achieved by each controller over the deployment period with $D_{init} = 15$ days and $D^{upd} = 15$ days. These values were empirically found to offer the best performance - the impact of different values of these parameters is discussed in more detail in the subsequent section.

Table III.6: Evaluation metrics for different controllers. Here, OI : Offline Initialization, AM : Action Masking, OA : Online Adaptation.

Type	OI	AM	OA	Net Bill			PMV Mean (Std)		
				Seed 1	Seed 2	Seed 3	Seed 1	Seed 2	Seed 3
RBC	N	N	N	1034.25			-0.362 (0.335)		
RL (R^A)	N	N	Y	-947.52	-2392.17	-2103.21	0.071 (0.404)	0.374 (0.69)	0.25 (0.832)
RL (R^A)	N	Y	Y	872.62	852.29	630.67	-0.229 (0.402)	-0.273 (0.417)	-0.229 (0.412)
RL (R^A)	Y	N	N	-2546.24			0.641 (1.049)		
RL (R^A)	Y	N	Y	-7319.64	-7257.12	-7275.75	1.636 (0.93)	1.629 (0.92)	1.622 (0.931)
RL (R^A)	Y	Y	N	933.45			-0.174 (0.442)		
RL (R^A)	Y	Y	Y	119.51	-187.01	-49.65	-0.034 (0.442)	-0.016 (0.439)	-0.023 (0.438)
RL (R^B)	N	N	Y	-1847.57	-1788.63	-2834.57	0.203 (0.674)	0.25 (0.9)	0.364 (0.962)
RL (R^B)	N	Y	Y	732.29	813.33	732.27	-0.23(0.411)	-0.268(0.428)	-0.234 (0.422)
RL (R^B)	Y	N	N	-6702.61			1.745 (0.797)		
RL (R^B)	Y	N	Y	-7591.73	-7593.48	-7461.09	1.772 (0.790)	1.764 (0.785)	1.773 (0.772)
RL (R^B)	Y	Y	N	1065.14			-0.151 (0.457)		
RL (R^B)	Y	Y	Y	-95.16	-292.77	-174.5	-0.008 (0.442)	0.018 (0.424)	0.020 (0.421)

From a numerical standpoint, several combinations of RL agents demonstrate improvements over the baseline, both in reducing electricity costs and enhancing thermal comfort. However, a clear tradeoff emerges between these two objectives: in many cases, the RL agent converges to a local optimum where it sacrifices thermal comfort almost entirely in pursuit of higher cost savings. Each of the techniques affects the final policy in different ways - this is visible in Fig III.4a-d, which plots the results with and without each technique in isolation.

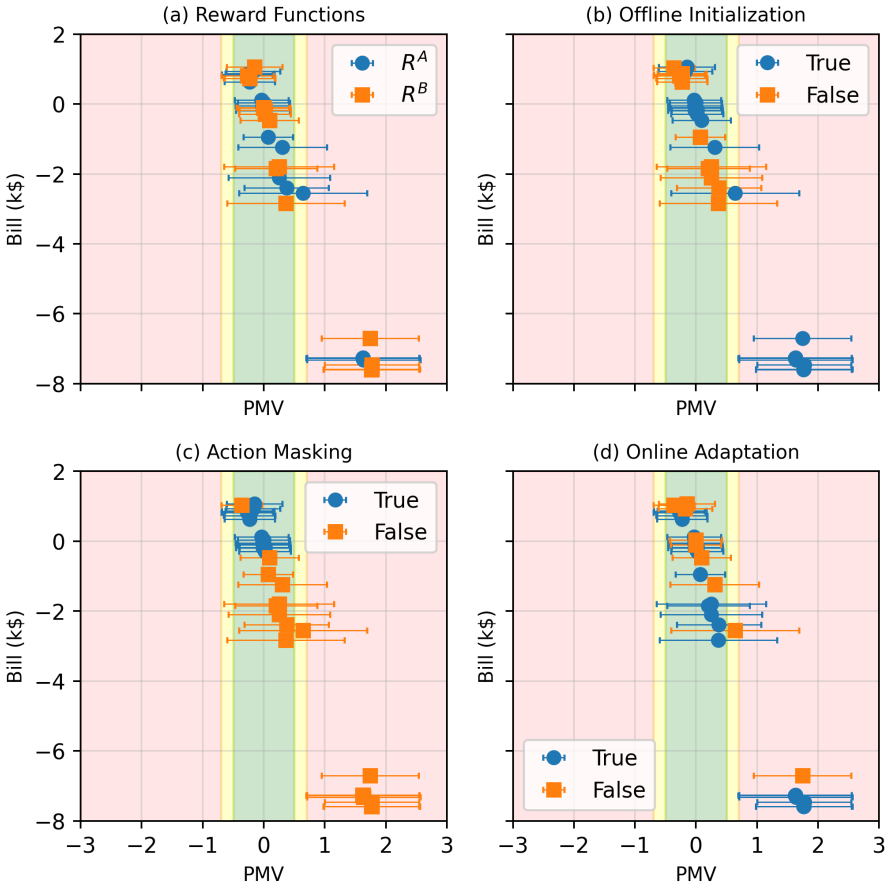

Figure III.4: Evaluation metrics under runs (a) with different reward functions, with and without (b) offline initialization, (c) action masking, (d) online adaptation. Red zones denote unacceptable thermal discomfort levels, while yellow is moderately uncomfortable according to [4].

Fig III.4a shows the metric values in Table III.6 for runs with each reward function. Across reward functions, agents generally achieve comparable performance and constraint satisfaction, with R^B yielding slightly lower bills when feasible PMV ranges are maintained. The limited one-year evaluation window likely restricts exploration, preventing the agent from fully leveraging richer reward formulations. Importantly, both R^A and R^B sometimes produce unsafe controls despite including comfort terms, highlighting the risk of relying solely on soft penalties in online settings where harmful actions may still be explored, especially during the initial phase.

Fig III.4c shows the results for runs with and without the action mask. Action Masking has a clear impact on control safety - by enforcing rules, it ensures that occupant thermal comfort requirements are always met. However, this comes at the cost of higher

bills, as masking forces the agent to select underexplored actions it has not sufficiently encountered during training. This mirrors the findings in our previous work [139], and is consistent with the theoretical analysis of action masking presented in [137].

4.b Metric Trends During Evaluation

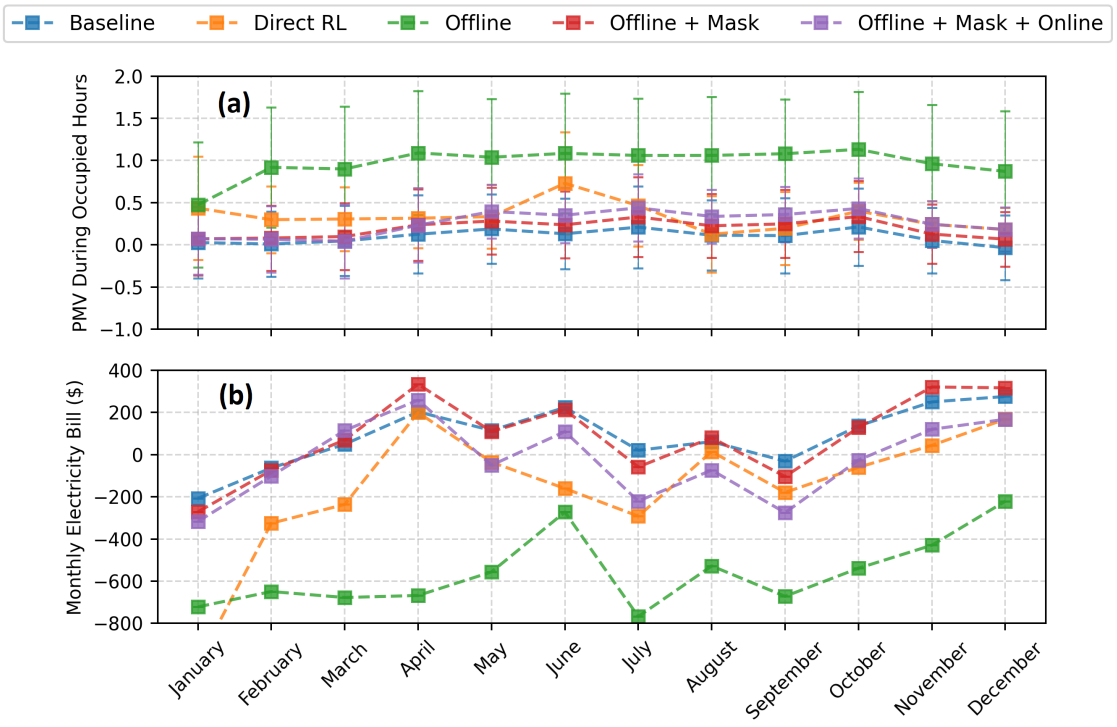

Figure III.5: (a) PMV during occupied hours and (b) electricity bill for each month in the evaluation for different controllers. RL runs are based on R^B.

Fig III.5a and III.5b present the monthly evolution of the metrics over the evaluation period for a subset of controllers discussed earlier. The tradeoff between thermal comfort and energy costs is clearly visible: agents maintaining better comfort (PMV closer to 0) generally incur higher electricity consumption. The results for the direct RL controller highlight the main challenge of such approaches in real-world settings - during the first three months, it produces unacceptable levels of discomfort (PMV > 1.0). This issue becomes more pronounced with offline initialization ($D^{init} = 15$ days) without masking or adaptation, leading to a suboptimal control policy that is neither safe nor responsive to feedback.

The use of an action mask helps the agent maintain higher thermal comfort, ensuring competent performance even during input conditions not seen in training. When the agent is further trained online on the real system, the policy adapts to changing conditions over time, eventually resulting in consistently lower bills. However, this comes at the cost higher bills during the initial exploration phase, which can be offset by later savings. For instance, the online-trained agent incurred the highest bills in March but the lowest from May onwards, while maintaining acceptable comfort levels.

Thus, while each technique individually influences controller performance, combining all of them enables the agent to adapt to changing conditions while consistently adhering to safety rules. The effectiveness of this integrated approach grows over time as the agent

trains on more system data. This is illustrated in Fig III.6, which compares the baseline agent with an RL agent employing the full set of techniques for a single day during the evaluation.

Both controllers begin the day with the SOC at its lower bound and follow a similar strategy during the morning hours. Around 7 AM, the baseline controller charges the battery using excess PV generation. At the same time, a sharp spike in electricity prices occurs. In response, the RL agent chooses to export the surplus generation to the grid, achieving significant savings on the energy bill. As the high-price period extends into the start of the workday, the agent maintains the AC setpoint as high as allowed while adhering to the rules. In contrast, the baseline controller's scheduled setpoints increase energy consumption while reducing thermal comfort, as indicated by the PMV. This pattern continues until the end of the workday, when prices drop and the agent shifts its focus to aggressively charging the BESS. Although the baseline agent earns higher profits toward the end of the day, after PV generation ceases, the RL agent's gains during the earlier sharp price peaks more than compensate for this difference. Over the entire day, the RL agent achieves a net profit nearly equal in magnitude to the baseline's cost, while maintaining a higher level of thermal comfort.

It is important to note that the example above reflects the RL agent's performance near the end of the training period, after multiple training cycles have been completed. Similar benefits are not observed during the early months of the year; nonetheless, Table III.6 shows that the overall training approach provides a net benefit. Fig III.6 also highlights a common issue with discrete RL controllers: the control actions can be erratic, resulting in high-frequency fluctuations in setpoints and power profiles. Although the magnitudes are relatively small in this case, such erratic behavior is undesirable, as it can contribute to equipment wear and tear.

4.c Impact of D^{init} and D^{upd}

Table III.7a and III.7b give the bill and PMV for different values of D_{init} and D_{upd} - here, $D^{init} > 0$ implies that online adaptation was used, and results for multiple runs are reported. Fig III.4b plots the results based on whether offline initialization was used in the experiment, and Fig III.4d plots the results based on whether online adaptation was used. The metrics in Table III.7 do not use action masking, and are based on reward function R^B.

It is clear that these parameters have a significant impact on the performance of the control approach. In the absence of online adaptation (ie, $D^{upd} = 0$), the offline learning performance improves significantly as more data from the system in collected. Transitioning from the RBC to the offline RL control is infeasible when $D^{init} = 15$ or $D^{init} = 45$ days, as the controller completely sacrifices thermal comfort for lower energy bills. However, when trained offline with $D^{init} = 75$ days, it vastly outperforms the RBC on both metrics. This may be because the long-term effects of the comfort term in the reward become apparent only when analyzing a longer period of data. As such, controllers initialized with smaller datasets may learn suboptimal policies that reflect short-term gains.

Due to the randomness used in RL training algorithms [107], online adaptation shows a degree of performance variation across different random seeds. Despite these effects, it is evident that the updation pushes the policy towards a more desirable region - in all cases, the controllers with online adaptation are able to improve thermal comfort conditions. An interesting observation is that the best offline agent (ie, $D^{init} = 75$) does not lead to the overall best performance over the evaluation period. Among controllers that maintain acceptable PMV levels, the best results were achieved with $D^{init} = 15$ days

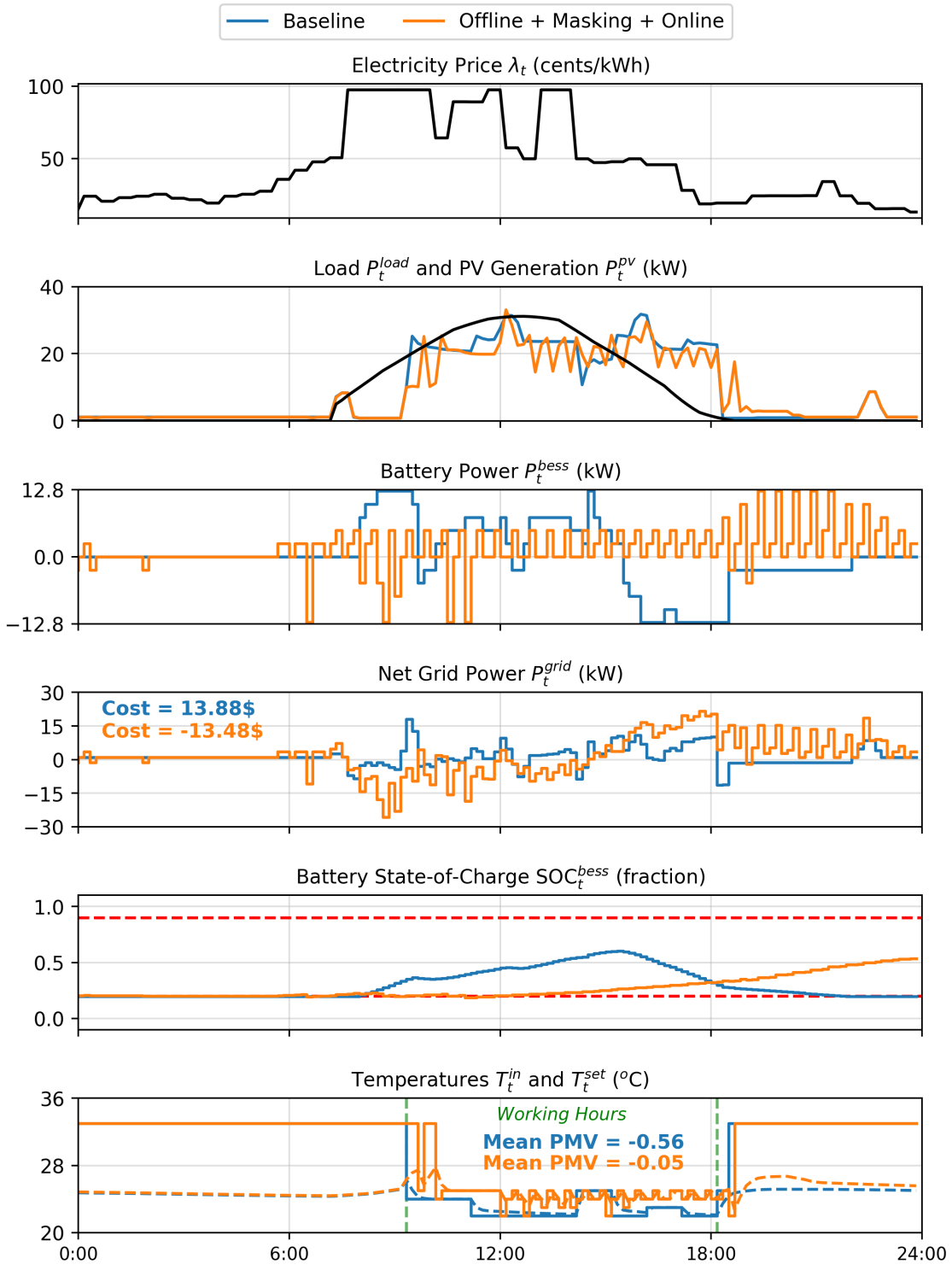

Figure III.6: Power flows and temperatures over an example day near the end of the evaluation period.

Table III.7: Results for different values of D_{init} and D_{upd}.

(a) Bill

	D_{init} = 15	D_{init} = 45	D_{init} = 75
$D_{\textbf{upd}} = 0$	-6702.61	-5398.92	-107.85
$D_{\textbf{upd}} = 7$	603.97, -176.82, 73.48	-94.39, -15.38, 113.49	177.19, 179.40, 393.23
$D_{\textbf{upd}} = 15$	-95.16, -292.77, -174.50	92.36, 99.48, 230.31	453.18, 395.57, 522.30
$D_{\textbf{upd}} = 30$	327.02, 312.59, 211.98	171.90, 198.88, 412.54	574.22, 634.52, 665.64

(b) PMV

	D_{init} = 15	D_{init} = 45	D_{init} = 75
$D_{\textbf{upd}} = 0$	1.745 (0.797)	1.445 (1.029)	-0.003 (0.414)
$D_{\textbf{upd}} = 7$	-0.179 (0.446), 0.020 (0.428), 0.018 (0.423)	-0.037 (0.438), -0.033 (0.428), -0.073 (0.443)	-0.073 (0.435), -0.075 (0.433), -0.134 (0.872)
$D_{\textbf{upd}} = 15$	-0.008 (0.442), 0.018 (0.424), 0.020 (0.421)	-0.054 (0.440), -0.060 (0.436), -0.070 (0.440)	-0.149 (0.449), -0.136 (0.444), -0.149 (0.457)
$D_{\textbf{upd}} = 30$	-0.085 (0.480), -0.096 (0.465), -0.004 (0.428)	-0.102 (0.454), -0.088 (0.436), -0.120 (0.443)	-0.186 (0.979), -0.176 (0.457), -0.223 (0.472)

and $D^{upd} = 15$ days. Another observation is that more frequent updates are better when $D^{init} >= 45$ days; however, with $D^{init} = 15$ days, a higher update frequency leads bigger variations across random seeds.

This suggests that some interdependence exists between D^{init} and D^{upd}, and a balanced set of values need to be chosen. A longer D^{init} allows the controller to gather more data from an existing controller, leading to a better offline policy that captures any long-term effects embedded in the reward function. However, this limits the time available for further improvement through online adaptation. An alternate strategy is to transition to RL at an earlier point (at the risk of an initial suboptimal controller), and allow more time for direct interaction and learning from the system. Finally, lower values of D^{init} and D^{upd} may offer superior performance, but run the risk of long-term instability due to random variations when training.

5 Summary & Conclusion

This chapter examines a representative single-building control problem in BEM through the lens of knowledge-informed RL, demonstrating how the concepts introduced in the previous chapter can be systematically applied to design effective control strategies that adhere to various constraints. Focusing on the joint management of the AC setpoint and battery operation under dynamic electricity tariffs, we show that integrating domain knowledge via appropriate techniques enables a purely data-driven controller to achieve substantial improvements over a baseline policy within a realistic timeframe while maintaining occupant thermal comfort and complying with the equipment owner's safety requirements.

In addition to a careful MDP design based on the principles outlined earlier, the pro-

posed solution leverages three key techniques. First, offline initialization from an existing controller, combined with a tailored reward function, is employed to retain the embedded knowledge of the baseline while enhancing its performance. Second, an action mask is applied to dynamically restrict the agent's action space, thereby enforcing strict safety rules that guarantee occupant comfort and protect equipment at all times. Finally, the agent undergoes continual training during deployment, permitting limited exploration to discover improved policies - albeit at the potential cost of temporary increases in energy bills or reductions in comfort. This approach ensures that the policy evolves over time, steadily improving with experience while remaining safe for real-world operation.

While the above results highlight the potential of data-driven controllers in such settings, several limitations remain. RL policies often exhibit erratic behavior, particularly in the early stages of training, leading to high-frequency fluctuations in control signals that can accelerate equipment wear. Moreover, although improving upon a baseline policy through purely data-driven learning is appealing, RL agents are prone to becoming trapped in local optima and typically require extensive exploration cycles before approaching a true global optimum. Depending on the specific control task, a model-based approach may therefore be more advantageous, as the upfront effort of model design can be offset by greater efficiency and reliability during deployment.

Ongoing Real-World Trials

The methodology presented in this chapter is presently undergoing validation studies through a real-world testbed in Singapore. A lecture theatre at the Nanyang Technological University campus is used for the trial, in which the cooling power of the HVAC system is modulated by an RL agent, with the objective of balancing the power consumption and the thermal comfort of occupants. As the testbed was previously used for a real-world MPC trial [59, 97], the results will provide a direct comparison between RL and MPC. The results are expected to be published in early 2026.

Chapter IV

Application to a Multi-Building Coordination Problem

This chapter presents the application of the proposed techniques to a multi-building AC coordination problem considering both technical and social factors. As in the previous chapter, the problem is introduced and analyzed through the lens of knowledge-informed RL, and a suitable set of techniques is utilized to design a solution. The work discussed in this chapter is based on published studies on coordinated DR using RL [25] and online adaptation [92].

1 Problem Overview

This chapter describes a method to control ACs in a group of 100 buildings to reduce aggregate peak loads while explicitly allowing consumers to override control signals based on their thermal comfort. A centralized architecture is adopted, with an AG that sends individual temperature setpoint signals to each consumer on an hourly basis. It is based on approximate grey-box models of consumers and a novel Split-Input Actor-Critic DRL architecture that enhances the learning performance while limiting online data requirements. With this in mind, the key novelties here are:

1. A solution combining grey-box models and actor-critic DRL is presented, which allows model-based insights to be used in training while remaining data-driven in deployment.
2. A scalable centralized architecture that allows coordination across 100s of buildings while remaining easy to implement.
3. The approach does not require the use of behind-the-meter information or pre-existing models, and is entirely based on smart meter and weather data, thus protecting consumer privacy.
4. Model inaccuracy is explicitly considered, and a domain randomization technique [124] is employed to mitigate the gap between the training models and evaluation system.
5. By allowing consumers to override control signals based on their local conditions, the scheme protects user thermal comfort while encouraging user interaction.

In summary, the main novelty of the methodology is that it is simultaneously scalable, easy to implement, and respects consumer privacy. It is motivated by the need to address

several technology-centric and consumer-centric adoption barriers identified from real-world trials of such systems.

1.a System Description

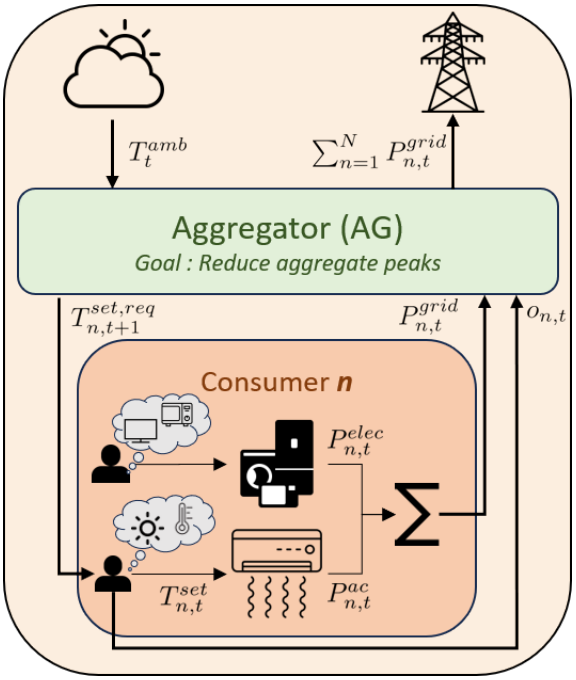

Figure IV.1: System illustration

Fig IV.1 shows a schematic of the system components and interactions. The case study is based on the control of a group of residential consumers ($n \in N$) over time ($t \in T$ at 1-hour resolution here) situated in a tropical region. Each consumer is equipped with a smart meter, and has a cooling load $P_{n,t}^{ac}$ (for the AC system - this is the controllable load[1]) and an electrical load $P_{n,t}^{elec}$ (non-controllable appliance load). The overall consumption is metered and aggregated as $P_{n,t}^{grid}$. Each consumer also has a preferred indoor temperature $T_n^{in,ideal}$. In the default case, the AC setpoint $T_{n,t}^{set} = T_n^{in,ideal} \forall t$.

The group is controlled by an AG, such as a utility company, with the objective of reducing the overall peak load of the system. At every time step t, the AG accesses the latest smart meter readings ($P_{n,t-1}^{grid}$) and the ambient temperature (T_t^{amb}). It uses these inputs to generate dimensionless control signals ($T_{n,t+1}^{set,req} \in [-1, 1]$), indicating an indoor setpoint adjustment from lower to higher values relative to $T_n^{in,ideal}$. Prior to the start of the control, the AG has access to 2 months of smart meter data from all consumers under a static pricing scheme.

Consumers may override the signal arbitrarily at their discretion by communicating an override signal ($o_{n,t} \in \{0, 1\}$) - for example, this could be due to thermal comfort factors. The AG's decision is only based on smart meter, weather and override data - it notably does not have access to any behind-the-meter data, such as indoor temperatures, occupancy or disaggregated AC consumption. At the consumer premises, the value of $T_{n,t}^{set}$ is chosen as shown in Eq IV.1 after accounting for override signals $o_{n,t}$.

[1] Heating loads are not considered since we study a tropical climate.

$$T^{set}_{n,t} = \begin{cases} T^{in,ideal}_{n,t} + T^{set,req}_{n,t} & \text{if } o_{n,t} = 0 \\ T^{in,ideal}_{n,t} & \text{if } o_{n,t} = 1 \end{cases} \tag{IV.1}$$

1.a-i Simulated Override Mechanism

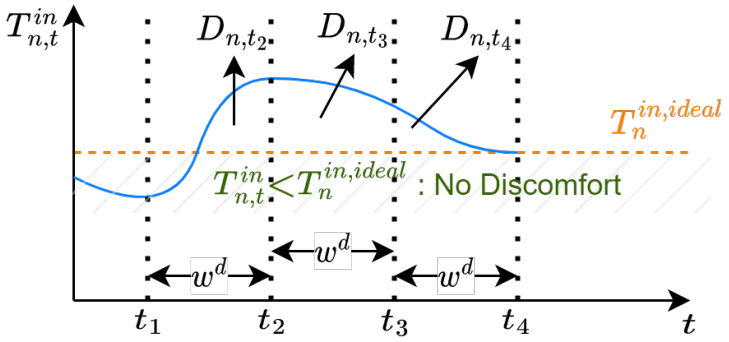

Figure IV.2: Discomfort model used in the simulated testcase

To simulate consumer overrides, a discomfort model based on cumulative deviation from $T^{in,ideal}_{n,t}$ is adopted, following empirical results from [152] and [153]. As illustrated in Fig IV.2, the discomfort $D_{n,t}$ at a given time t is the cumulative temperature deviation above $T^{in,ideal}_{n,t}$ over a window of length w^d before t (ie, the area of $T^{in}_{n,t}$ above $T^{in,ideal}_{n}$ in each window in Fig IV.2). A consumer n has a maximum discomfort capacity represented by D^{max}_n, and triggers an override if $D_{n,t}$ exceeds D^{max}_n and a setpoint increase is requested. The override behaviour is implemented as shown in Eq IV.2.

$$o_{n,t} = \begin{cases} 1 & \text{if } D_{n,t} > D^{max}_n \text{ and } T^{set,req}_{n,t} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{IV.2}$$

1.a-ii Calculation of Cooling and Appliance Loads

Each building contains a thermostatically controlled AC unit to satisfy the cooling demand of its occupants. An AC has a rated maximum power denoted by $P^{ac,max}_n$, and follows the setpoint-driven operation shown in Eq. IV.3. Here, $T^{in}_{n,t}$ is the indoor temperature and T^{db} is a deadband around the setpoint where $T^{in}_{n,t}$ is permitted to float. In the default case, the AC setpoint $T^{set}_{n,t} = T^{in,ideal}_{n} \forall t$.

$$P^{ac}_{n,t} = \begin{cases} P^{ac,max}_n, & T^{in}_{n,t} - T^{set}_{n,t} > T^{db} \\ 0, & T^{in}_{n,t} - T^{set}_{n,t} < T^{db} \\ P^{ac}_{n,t-1}, & \text{otherwise} \end{cases} \tag{IV.3}$$

The evolution of the indoor temperature $T^{in}_{n,t}$ in response to the cooling power and ambient conditions is calculated using a 4R2C thermal model of the building envelope as suggested in [5]. The electrical load $P^{elec}_{n,t}$ is taken from time-series data generated using a load generation tool, "LoadProfileGenerator" [154], and is not controllable. Fig IV.3 shows $\sum_{n \in N} P^{grid}_{n,t}$ and T^{amb}_t over a week in the default case. A detailed description of the implementation is given in B.2.

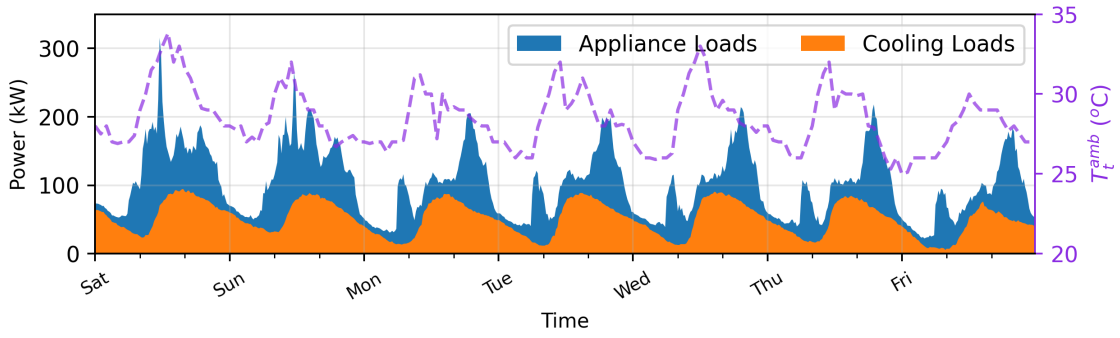

Figure IV.3: Aggregated load and ambient temperature over a typical week in the case study with $N = 100$ and $T_{n,t}^{set} = T_n^{in,ideal}$

1.b Evaluation Metrics

We use the term 'policy' (denoted by π) here to describe any decision-making process adopted by the AG to generate $T_{n,t}^{set,req}$. The performance of a given policy is measured using the metrics given in Table IV.1. Here, E^{peak} measures the energy consumed under peak loads (ie, $G_t > 200$ kW for the simulated testcase), and quantifies the peak reduction capabilities of the agent. o^{tot} is count of the total number of override signals ($o_{n,t} = 1$) received, and is a measure of consumers' satisfaction with the DR program. It should also be noted that a reduction in o^{tot} often coincides with a reduction in E^{peak}, since overrides lead to unexpected load spikes that the AG wishes to avoid. Finally, E^{tot} is the total energy consumption under the policy. A reduction in E^{peak} is prioritized even if it creates a small increase in E^{tot}, since the latter may lead to better utilization of renewable generation depending on the electricity mix [155]. All metrics are evaluated over the entire test duration, and lower values are better when comparing different policies.

Metric	Equation
Energy Under Peak Loads E^{peak} (MWh)	$\sum_{n,t} P_{n,t}^{grid} \Delta t$ if $G_t >$ 200 kW
Number of Overrides o^{tot}	$\sum_{n,t} o_{n,t}$
Total Energy Consumption E^{tot} (MWh)	$\sum_{n,t} P_{n,t}^{grid} \Delta t$

Table IV.1: Description of evaluation metrics

2 Solution Approach Under Knowledge-Informed RL Paradigm

2.a Relevant Challenges

1. **Learning Challenges**: The agent must coordinate across N consumers, leading to larger state and action spaces. It also faces delayed effects due to the thermal inertia of AC-based control, and operates with limited information - typically only smart meter data, without access to indoor sensors.

2. **Infrastructure**: Sensing and control infrastructure is limited, and in the residential DR context, the solution must remain lightweight with minimal infrastructure

requirements. Only aggregated meter data is available at a 15-minute resolution.

3. **Costs & Benefits**: The solution should achieve the coordinated objective of peak load reduction while minimizing consumer discomfort. Additionally, although not detailed here, the override signal can support the design of fair incentive schemes for consumers who contribute more to DR.

4. **Safety, Security & Trust**: Control should consider equipment health but place greater emphasis on consumer feedback and acceptance, ensuring that data privacy and user agency are respected.

2.b Available Domain Knowledge

1. **Building Physics**: The thermal physics of the building envelope and the operation of the residential AC are known - as such, the basic response of the building to different weather patterns and control signals can be predicted.

2. **System Constraints**: The AG's setpoints are defined relative to each consumer's (unknown) preferred setpoint. Crucially, the control must respect the override mechanism, allowing users to revert to their preferred operation whenever they experience discomfort. Additionally, no behind-the-meter inputs are available from the system for the control.

3. **Existing Operational Strategies**: The AC operates under a known hysteresis control sequence (Eq. IV.3), but other parameters (such as maximum cooling power) are unknown. Smart meter data from a static pricing scheme (i.e., without DR) is also available, providing partial insight into typical consumption patterns.

4. **User Interaction & Feedback**: During deployment, the feedback signal $o_{n,t}$ provides a direct indication of user satisfaction with the applied controls.

3 Methodology & Technique Selection

The solution is based on actor-critic DRL algorithms and grey-box building models. At the AG level, historical smart-meter ($P^{grid}_{n,t-}$) and weather (T^{amb}_{t-}) data is first collected from the uncontrolled system (ie, $T^{set}_{n,t} = T^{in,ideal}_{t}$). This data is used to create a set of Approximate Consumer Models (ACMs) for every consumer, which are used inside training environments for an actor-critic DRL agent to learn a policy offline. The trained actor network is finally deployed in the reference system, and the override signals available at that point are used to refine the controller over time. An overview of the steps involved in the end-to-end methodology is shown in Fig IV.4, highlighting the techniques and associated domain knowledge representations.

Assumptions

A major assumption made in the process of generating ACMs (described in the next section) is that the AG is able to disaggregate $P^{grid}_{n,t-}$ into the historical cooling load $P^{ac}_{n,t-}$ and electrical load $P^{elec}_{n,t-}$, using the historical ambient temperature data T^{amb}_{t-} [156]. $P^{ac}_{n,t-}$ and $P^{elec}_{n,t-}$ are not directly observable by the AG as they are behind-the-meter data. While this is a strong assumption, recent works have achieved good performance on this task [156, 157, 158]. These works utilize DNNs and hidden Markov models, and take advantage of the correlation between T^{amb}_{t-} and $P^{ac}_{n,t-}$. Additionally, as discussed in [159],

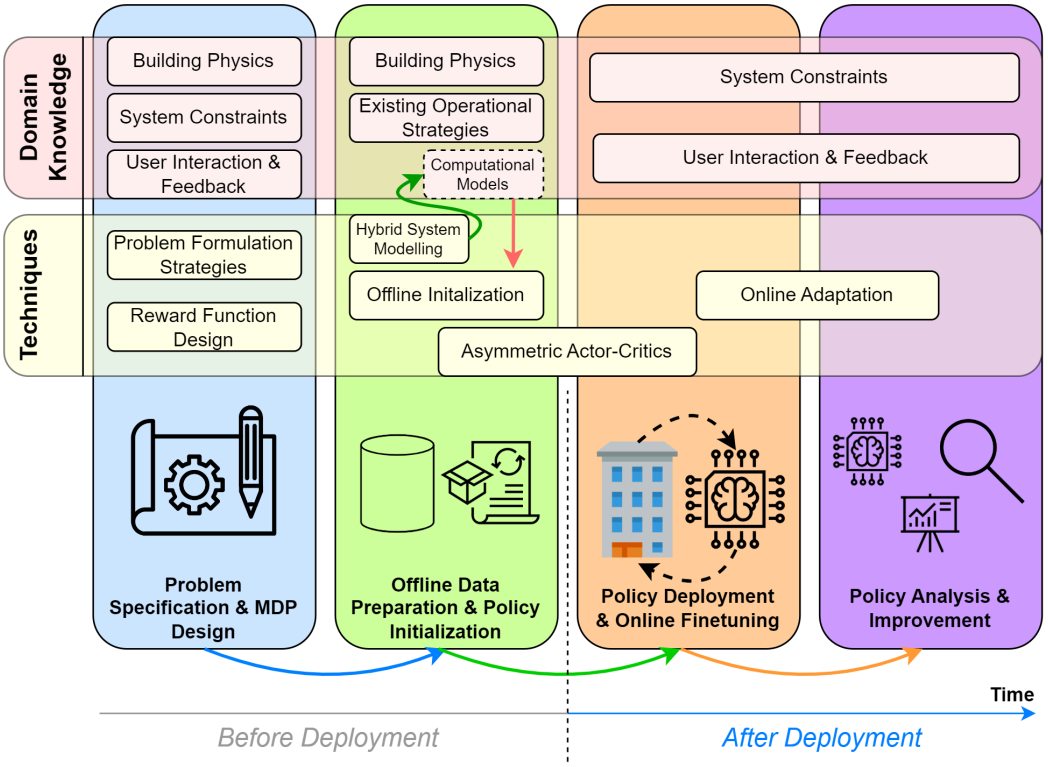

Figure IV.4: Knowledge-Informed RL Workflow for Multi-Building Problem.

the complexity of load disaggregation greatly increases with the number of components (such as appliances) required. Here, only the cooling load is to be separated; further, the decomposition is only performed on historical data, and is not needed in real-time. A second assumption is that all ACs follow the operation described in Eq. IV.3 with a fixed deadband temperature $T^{db} = 0.5°C$. This model is considered valid for residential settings [160].

3.a Problem Formulation Strategies

The RL agent is designed to individually monitor and control each building. The controller uses inputs over a window of length L steps from $t-1$ to $t-L$. For a consumer n, the states $s_{n,t}$ and actions $a_{n,t}$ at every time step t are as follows (bold notation indicates a vector containing L previous values and $\sim n$ denotes all consumers excluding n).

- $s_{n,t} = \{t, \mathbf{T^{amb}}, \mathbf{P^{grid}_n}, \sum \mathbf{P^{grid}_{\sim n}}, \mathbf{T^{in}_n}, \mathbf{o_n}\} \in \mathbb{R}^{5 \cdot L+1}$
- $a_{n,t} = T^{set,req}_{n,t} \in [-1, 1]$

Here, manual feature selection techniques such as in [161] are not used - instead, $s_{n,t}$ is composed of all possible information signals. A single actor network is trained that sequentially accepts $s_{n,t}$ and outputs $a_{n,t} \forall n \in N$ during deployment. This is because RL-based approaches do not scale well if the dimensions of the state and action spaces depend on N (due to the "curse of dimensionality"[162])[2]. Coordination across consumers is achieved through the reward function, which is described in subsequent sections. It should be highlighted that $s_{n,t}$ contains $T^{in}_{n,t}$ which is not observable, and

[2]The collective state of the entire system has $N \cdot (5 \cdot L + 1)$ dimensions along with N actions.

$o_{n,t}$ which is only available during deployment. As such, specific techniques are used to enable decision making using only an observable subset of features, each of which are explained in detail subsequently.

The main benefits of designing the state and action spaces, along with the agent architecture, in this way are:

1. The architecture is lightweight - requiring only a single DNN of fixed size regardless of N - while remaining scalable.

2. The action space naturally enforces the constraint that setpoint changes are relative to each consumer's preferred indoor temperature.

3.b Hybrid System Modelling (ACM Generation)

ACMs are grey-box models of consumers consisting of a thermal component (representing the consumer's building and AC system) and a behavioural component (representing their thermal comfort preferences). An overview of the components and working of an ACM is depicted in Fig IV.5 - the model parameters are shown in blue, and hat notations are used to describe unknown quantities which need to be estimated. ACMs are employed in a recursive manner to generate time-series predictions of the indoor temperature ($\hat{T}^{in}_{n,t}$), AC power ($\hat{P}^{ac}_{n,t}$) and the override status ($\hat{o}_{n,t}$) given a sequence of control signals ($T^{set,req}_{n,t}$) and weather (T^{amb}_{t}).

The thermal comfort model is parameterized by estimates of the ideal indoor temperature $\hat{T}^{in,ideal}_{n}$ and discomfort capacity $\hat{D}^{max}_{n}$. These cannot be extracted from $P^{grid}_{n,t-}$ since historical data was collected with fixed setpoints. As such, the AG assigns typical values for these parameters based on domain knowledge. For a given choice of $\hat{T}^{in,ideal}_{n}$ and $\hat{D}^{max}_{n}$, the behaviour described in Eq IV.2 is incorporated in the model to calculate the estimated setpoint $\hat{T}^{set}_{n,t}$ and the override status $\hat{o}_{n,t}$.

In contrast, the AC parameters $P^{ac,max}_{n}$ and T^{db} (the deadband temperature) are perfectly known to the AG. The former is inferred from $P^{ac}_{n,t-}$ after disaggregation, while the latter is assumed to be a constant throughout the system. The AC model follows the operation in Eq. IV.3 and predicts the AC power $\hat{P}^{ac}_{n,t}$.

A first-order RC thermal network [5] is used to approximate the thermal behaviour of the buildings. This choice is suitable for data-limited contexts such as the present case, and is often found to outperform black-box approaches in comparative studies [110]. The simplified thermal model uses 2 parameters - a resistive element R^{in}_{n} and a capacitive element C^{in}_{n}. It predicts the evolution of the indoor temperature as a function of $\hat{T}^{in}_{n,t-1}$, T^{amb}_{t-1} and $\hat{P}^{ac}_{n,t-1}$ as shown in Eq. IV.4. Here, COP is the coefficient-of-performance of the AC unit and Δt is the time step - in this work, $COP = 3.5$ and $\Delta t = 15$ min.

$$\hat{T}^{in}_{n,t} = \hat{T}^{in}_{n,t-1} + \frac{\Delta t}{C^{in}_{n}} \left(\frac{T^{amb}_{t-1} - \hat{T}^{in}_{n,t-1}}{R^{in}_{n}} - COP \cdot \hat{P}^{ac}_{n,t-1} \right) \tag{IV.4}$$

The approach given in [163] and [164] to create thermal models from meter data is used to find suitable parameter values, as follows. For a given R^{in}_{n} and C^{in}_{n}, the ACM is used to estimate the AC power $\hat{P}^{ac}_{n,t-}$ (this is a time-series prediction with the same length as $P^{ac}_{n,t-}$) for the historical ambient temperature T^{amb}_{t-} and the estimated $\hat{T}^{in,ideal}_{n}$ (with $T^{set,req}_{n,t} = 0$)[3]. R^{in}_{n} and C^{in}_{n} are then fit by minimizing the Mean Squared Error (MSE) between $\hat{P}^{ac}_{n,t-}$ and $P^{ac}_{n,t-}$ as described in Eq. IV.5. The implementation of the differential

[3]The RC parameters are independent of $\hat{D}^{max}_{n}$ since no setpoint deviation is requested here.

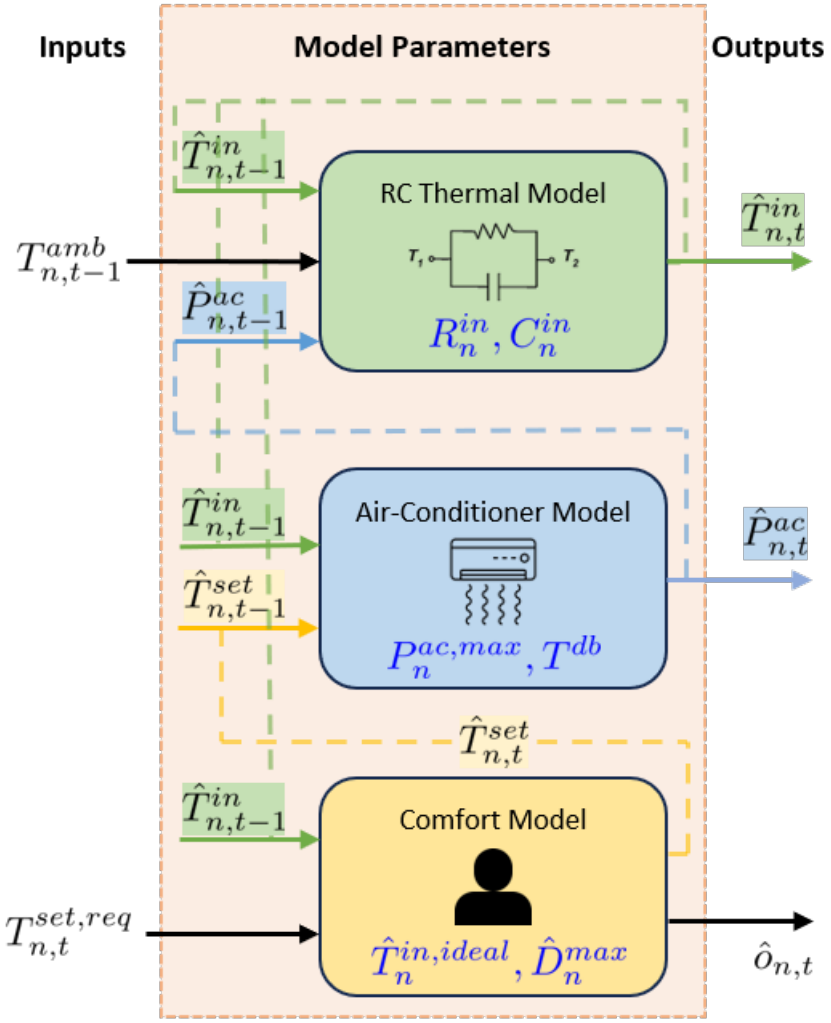

Figure IV.5: Block diagram of an ACM. Hat notations represent estimated/unknown quantities.

evolution algorithm [165], a meta-heuristic function minimization approach, in the scipy library[4] was used in this work.

$$R_n^{in}, C_n^{in} = \arg\min_{R_n^{in}, C_n^{in}} \textbf{MSE}\left(P_{n,t-}^{ac}, \hat{P}_{n,t-}^{ac} | R_n^{in}, C_n^{in}, T_{t-}^{amb}, \hat{T}_n^{in,ideal}\right) \tag{IV.5}$$

It is clear from Eq. IV.5 that a different choice for $\hat{T}_n^{in,ideal}$ would result in different thermal parameters for a given $P_{n,t-}^{ac}$ and T_{t-}^{amb}. In this work, the AG defines a set of M discrete values of $\hat{T}_n^{in,ideal}$, each representing a possible choice of the preferred indoor setpoint. A corresponding set of thermal parameters are generated for each choice. This gives rise to a set of M models for each consumer, and the entire space of approximate models is denoted as $b_{n,m} : \forall n \in N, m \in M$. $b_{n,m}$ and $b_{n,m'}$ have the same AC model but different thermal model parameters if $m \neq m'$. Also, $b_{n,m}$ and $b_{n',m'}$ have no common parameters if $n \neq n'$ irrespective of the values of m and m'.The discomfort capacity D_n^{max} is sampled from a normal distribution reflective of typical values.

Since the $\hat{T}_n^{in,ideal}$ values used by the AG may be very different from the actual $T_n^{in,ideal}$, the ACM predictions may be highly inaccurate. However, since the thermal models are

[4]Version 1.13.1

based on heat physics, they still capture the underlying system dynamics. The set $b_{n,m}$ is embedded into the training environment (described subsequently) for the RL agents with the goal of enhancing the robustness of the agent to differences in consumer behaviour. This process of incorporating uncertainties as random variables within the training environment of an RL agent is termed "Domain Randomization" [124]. It is often used in cases where agents are trained in a simulator before deployment in a real system, to handle the gap between simulation and reality.

3.c Model-Based Offline Initialization using Asymmetric Actor-Critics

$s_{n,t}$ contains behind-the-meter information (such as $T^{in}_{n,t}$ and $o_{n,t}$), which makes it impractical to learn a policy by direct interaction with the real system. Therefore, an offline training environment is created in which the ACMs are used to estimate these inputs. Within the training environment, $s_{n,t}$ is split into $s^{measured}_{n,t} = \{t, \mathbf{T^{amb}}, \hat{\mathbf{P}}^{\mathbf{grid}}_{\mathbf{n}}, \sum \mathbf{P}^{\mathbf{grid}}_{\sim\mathbf{n}}\}$ and $s^{predicted}_{n,t} = \{\hat{\mathbf{T}}^{\mathbf{in}}_{\mathbf{n}}, \hat{\mathbf{o}}_{\mathbf{n}}\}$ (hat notations represent model estimates). Then, an actor-critic architecture is defined in which the actor only uses $s^{measured}_{n,t}$, while the critic uses the full input space [68]. This is called the "Split-Input" architecture in this work, and is shown in Fig IV.6. The advantage is that the agent can learn a better policy with access to $s^{predicted}_{n,t}$ in training while only using observable inputs in deployment.

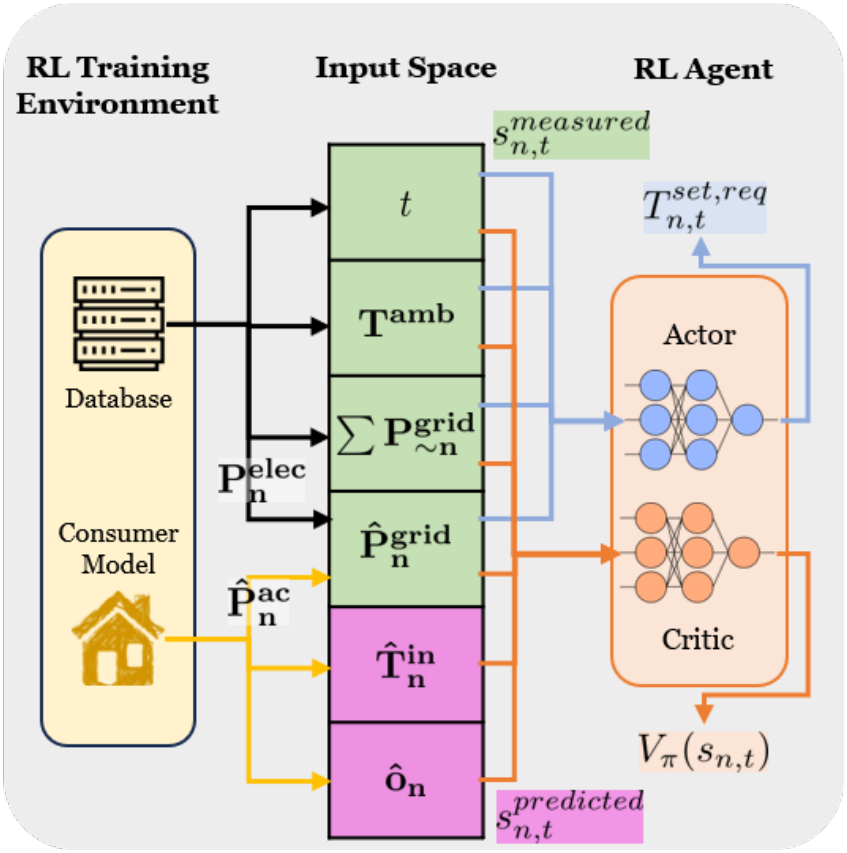

Figure IV.6: Schematic of the proposed Split-Input architecture. Hats represent estimated quantities, bold notations are vectors over L time steps before t.

It should be noted that only one ACM in $b_{n,m}$ is used in the training environment at any point of time, along with a "database" of the disaggregated $P^{grid}_{n,t-}$ and T^{amb}_{t-}. $\sum P^{grid}_{\sim n,t}$

is considered independent of $T^{set,req}_{n,t}$; hence, this is taken as the difference between the aggregate grid load $\sum_{n\in N} P^{grid}_{n,t}$ (from historical data) and $\hat{P}^{grid}_{n,t}$. The DRL agent is trained across a sequence of episodes, each of length 1 day. In each training episode, a random ACM $b_{n,m}$ is selected, and data from a random day is taken from the database. The agent interacts with this environment and uses the states, actions and rewards to update its parameters. By repeating this process over a large number of episodes, the agent learns a policy that generalizes to any element in $b_{n,m}$, and thus becomes robust to different consumer behaviours.

3.d Online Adaptation

The proposed methodology follows a typical training sequence in offline-to-online RL, with an initial offline phase based on a pre-existing policy (π^{init}), and an online phase where data is used to continuously improve the agent. Here, π^{init} is the policy learned through the offline initialization approach with the ACMs, and all subsequent policies are represented by DNNs which are updated using the CalQL algorithm [147].

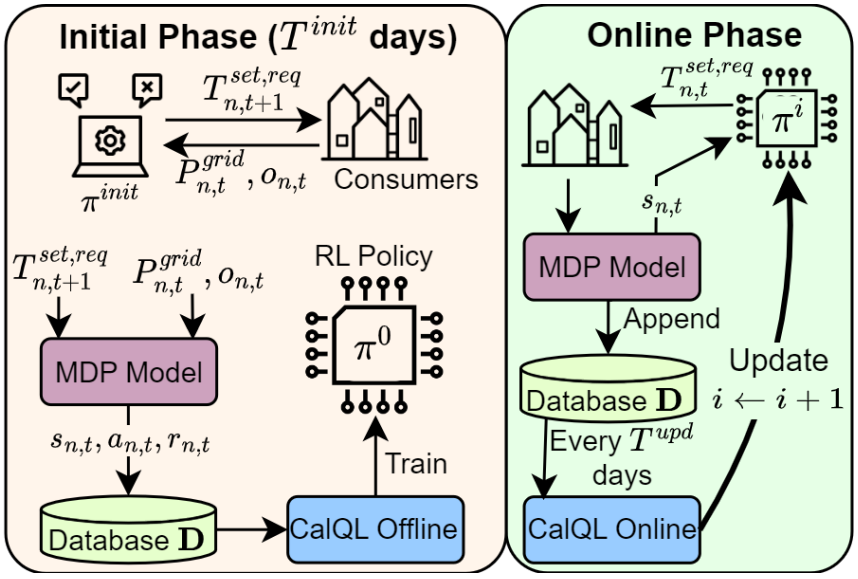

Figure IV.7: Steps involved in the offline and online phases of the approach.

Since an additional input ($o_{n,t}$) becomes available during the online operation, the MDP is reformulated in this phase, in terms of the states and the reward function used to train the agent. The new state includes the override signal, and is formulated as shown below ($s^{online}_{n,t}$). The reward function is discussed in Section 3.e.

- States : $s^{online}_{n,t} = \{\mathbf{T^{amb}}, \mathbf{P}^{grid}_{n}, \sum \mathbf{P}^{grid}_{\sim n}, \mathbf{o}_n\} \in \mathbb{R}^{4\times L}$
- Actions : $a_{n,t} = T^{set,req}_{n,t} \in [-1, 1]$

In the initial phase, π^{init} is deployed in the system for a period of T^{init} days, and data from this period is used to train the initial online policy represented as π^0 (which maps $s^{online}_{n,t}$ to $a_{n,t}$). An iterative online learning process follows - in the i-th iteration, π^i is deployed in the system for T^{upd} days, and the data collected during this period (as well as data from previous policies) is used to train the next policy π^{i+1}. This online update process continues throughout the test duration, such that the number of update iterations depends on the choice of T^{upd}. The final policy at the end of the test duration

is denoted as π^{end}, and the set of online policies is $\pi^{upd} = \{\pi^0, \pi^1, ..., \pi^{end}\}$. Fig IV.7 shows the sequence of operations in both phases.

3.e Reward Function Design

3.e-i During Offline Initialization

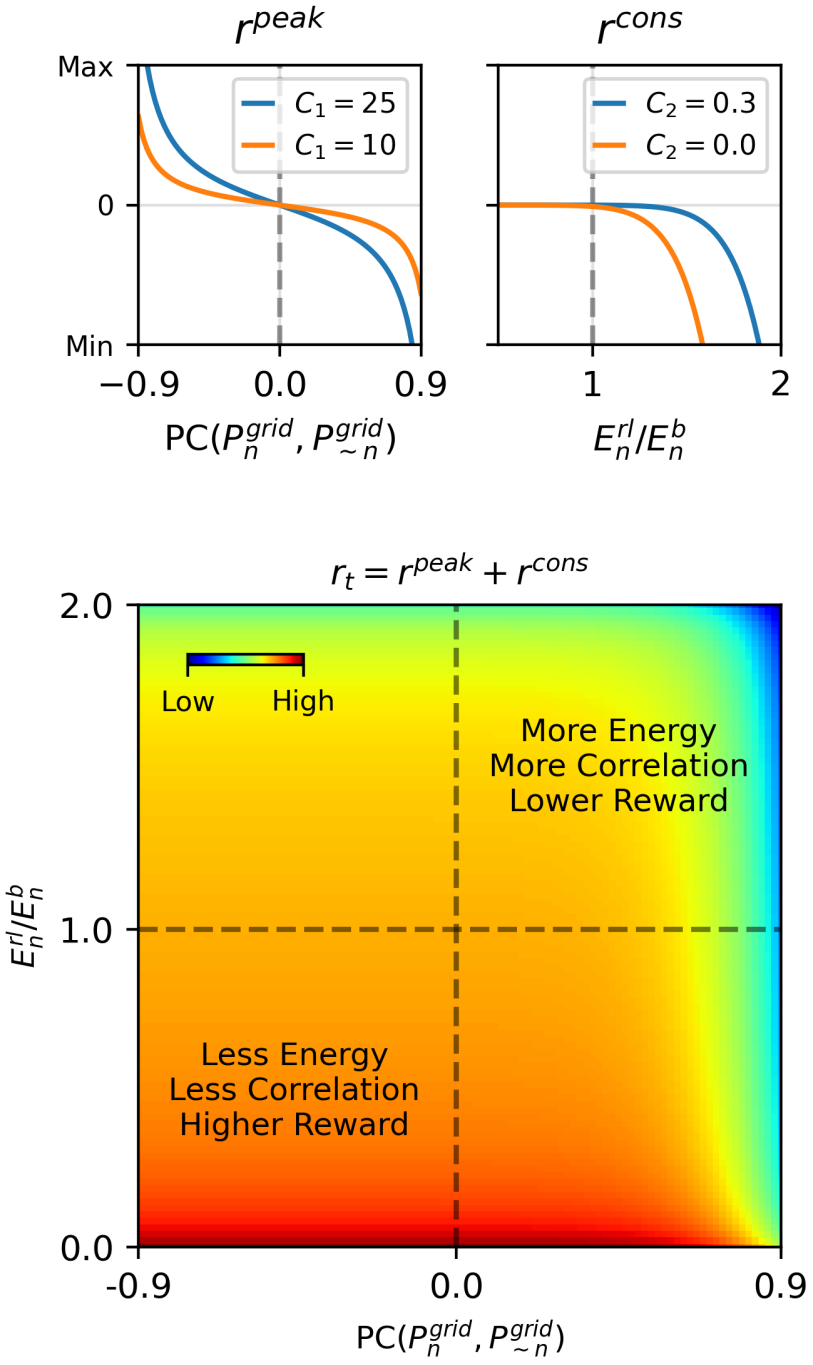

Figure IV.8: Visualization of Reward Function Components

A sparse reward formulation is used to represent the objectives of the AG - ie, the DRL

agent only receives a reward at the end of the episode, after all control actions (ie, $T^{set,req}_{n,t-1}$ to $T^{set,req}_{n,t-24}$) have been taken. The goal of the agent is to reduce the peaks in the aggregate power profile without increasing the total energy consumption compared to the default case (there is a tradeoff between these metrics in such systems, as discussed in [166]). As such, the reward function consists of two terms - a peak reduction term r^{peak}, and a consumption term r^{cons}.

Aggregate peaks in the system occur when $P^{elec}_{n,t}$ and $P^{ac}_{n,t}$ for multiple consumers increase together, which depends on the weather and consumer demographics [167]. We hypothesize that this effect can be quantified by measuring the similarity between $P^{grid}_{n,t-1:t-24}$ and $\sum P^{grid}_{\sim n,t-1:t-24}$ (power profiles of the building and the aggregation over the past 24 hours respectively). Consequently, r^{peak} is designed to reduce the Pearson Correlation Coefficient (PC) [5] (used here as a measure of similarity) between the above terms as shown in Eq. IV.6. The tan function is used here to heavily penalize (reward) very high (low) values of the correlation, and C_1 is a positive scaling constant.

$$r^{peak} = C_1 \tan\left(\frac{-\pi \cdot \mathbf{PC}(P^{grid}_{n,t-1:t-24}, P^{grid}_{\sim n,t-1:t-24})}{2}\right) \tag{IV.6}$$

r^{cons} is used to penalize agents that consume more energy than in the default case. Within each training episode, a baseline energy consumption ($E^b_n = \sum_t \hat{P}^{grid}_{n,t} \cdot \Delta t | T^{set,req}_{n,t} = 0$) is first estimated. r^{cons} is based on the ratio between the agent's consumption $E^{rl}_n (\sum_t \hat{P}^{grid}_{n,t} \cdot \Delta t | T^{set,req}_{n,t} = a_{n,t})$ and E^b_n as shown in Eq. IV.7. Its value is negligible when this ratio is below a scaling constant C_2 and explodes to a high negative value above C_2.

$$r^{cons} = -\left[\frac{E^{rl}_n}{E^b_n} - C_2\right]^{10} \tag{IV.7}$$

$$r_t = \begin{cases} r^{peak} + r^{cons}, & \text{end of training episode} \\ 0, & \text{otherwise} \end{cases} \tag{IV.8}$$

The total reward r_t at each time step is calculated as shown in Eq. IV.8. Fig IV.8 plots the two terms in the reward function, as well as the heatmap of their summation. Since r^{peak} and r^{cons} are normalized, the same formulation and weights can be used for any $b_{n,m}$ that the agent encounters during the training process.

3.e-ii During Online Adaptation

During the online phase, $o_{n,t}$ is provided as an input to the controller, and the reward function is updated to a continuous form that incorporates this term. The revised formulation, shown in Eq.IV.9, adds a penalty based on $\mathbf{o}_n$ alongside the components from Eq.IV.8.

$$r_{n,t} = -\left(\text{PC}\left(\mathbf{P}^{grid}_n, \sum \mathbf{P}^{grid}_{\sim n}\right) + \frac{\text{sum}(\mathbf{o}_n)}{L} + \left[\frac{P^{grid}_{n,t-1}}{\text{mean}(P^{grid}_{n,t})}\right]^2\right) \tag{IV.9}$$

4 Results & Analysis

To evaluate the performance of the agent, a test environment of $N = 100$ consumers was set up as described in the previous sections. The episode duration was fixed to 24 hours (24 time steps), with a simulation time step of 15 minutes (each $T^{set,req}_{n,t}$ is applied for four simulation steps). The AG has access to two months of weather and smart meter

[5] zero-lag cross-correlation coefficient

data, covering the period from March - April, and the test period is 30 days in May. The data was first used to generate the set of ACMs $b_{n,m}$ through the process described in Section 3.b. The estimated ideal indoor temperature, $\hat{T}_n^{in,ideal}$, was discretized into 3 levels (ie, $M = 3$) - {20^{o}C, 23°C, 26°C}, while the discomfort capacity $\hat{D}_n^{max}$ was sampled from a normal distribution $\mathcal{N}(0.5, 0.25)^oCh$. The duration of the input history window was set to $L = 8$. Then, the agents were trained offline for 2 million steps (equivalent to 21000 simulated days or episodes), after which the trained actor network was deployed in the real system for the test phase. To control for the inherent randomness in RL-based solutions, every experiment described in this section was repeated for 3 random seeds, and the results were averaged. TMY weather data from Singapore was used in all experiments, accessed using the diyepw Python library [149]. In this work, a type of actor-critic algorithm called Proximal Policy Optimization (PPO) [168], implemented via the stable-baselines3 library [169][6], was used for its balance of performance and stability. The default hyperparameters are used in all cases, with one notable exception - the discount factor (γ) was set to 0.958 to align with the episode duration of 24 steps[7]. In the reward function, a choice of $C_1 = 25$ and $C_2 = 0.3$ were empirically found to produce performant agents.

To study the impact of the Split-Input architecture shown in Fig IV.6, a comparison was performed between DRL agents provided with different levels of training information in the input space as described in Table IV.2. The default architecture proposed in this paper is the Split-Input architecture, in which only the critic is provided model-estimated inputs. In the Measured-Input case, intermediate model outputs are not used as agent inputs during the training process. Both the above architectures only require $s_{n,t}^{measured}$ during deployment. In contrast, the Full-Input architecture uses both $s_{n,t}^{measured}$ and $s_{n,t}^{predicted}$ both during training (model-estimated) and deployment. Since this requires $T_{n,t}^{in}$ and $o_{n,t}$ to be measured from the real system online, it is not realistic and is only used for comparison. In all three cases, a set of agents were also allowed to train directly on the reference system during the training period, bypassing the intermediate model fitting process. This is equivalent to using perfect models of the consumers for training - however, the consumption and weather data are still different across the training and test phases.

Architecture	Actor Inputs	Critic Inputs
Measured-Input	$s_{n,t}^{measured}$	$s_{n,t}^{measured}$
Split-Input	$s_{n,t}^{measured}$	$s_{n,t}^{measured} \cup s_{n,t}^{predicted}$
Full-Input	$s_{n,t}^{measured} \cup s_{n,t}^{predicted}$	$s_{n,t}^{measured} \cup s_{n,t}^{predicted}$

Table IV.2: Different DRL architectures considered in the study. Only actor inputs are required from the real system during deployment.

4.a Baseline and Alternate Strategies for Comparison

To compare the performance of the Split-Input approach, the following alternate strategies are considered:

1. **Baseline** : The fixed-setpoint strategy in which $T_{n,t}^{set} = T_n^{in,ideal} \forall n, t$ - it describes the situation where no flexibility is requested from the consumers.

2. **Precooling** : This strategy aims to shift and reduce the peaks by sending precooling signals ($T_{n,t}^{set,req} = -1 \forall n \in N$) 2 hours before the expected peak loads, followed by

[6] Version 2.3.2

[7] Here we use the heuristic $\frac{1}{1-\gamma} = 24$, and solve for γ

high setpoint requests during the peak period ($T_{n,t}^{set,req} = 1 \forall n \in N$). From Fig IV.3, it is visible that the peaks occur between 11 AM to 1 PM on weekends and 6 PM to 8 PM on weekdays - as such, precooling signals are sent at 9 AM on weekends and 4 PM on weekdays. The approach is analogous to time-of-use pricing approaches commonly used across the world [11].

3. **Decentralized Split-Input PPO (DSI-PPO)** : A set of N independent PPO agents, where each one is trained for a specific consumer $n \in N$. All agents use the Split-Input architecture, but each agent is trained only on the subset of M ACMs corresponding to its assigned consumer. During deployment, $T_{n,t}^{set}$ is chosen independently for each consumer using their dedicated model. Compared to the nominal Split-Input case (ie, centralized), this approach increases the parameter count and training time by a factor of N.

4. **Centralized PPO (C-PPO)** : A single centralized PPO agent that simultaneously observes the entire state of the system and outputs an N-dimensional action, similar to the approach in [82]. This agent takes only $s_{n,t}^{measured}$ as input, but is trained directly on the real system by looping the training data (ie, ACMs are not used). It represents a hypothetical agent that learns an optimal policy through direct interaction with consumers, and is only presented for comparison purposes[8]. The C-PPO agent uses a bigger DNN (with 4 times the number of parameters of the Split-Input network), and requires a significantly higher training time (~35 times higher than Split-Input).

4.b Accuracy of ACM Predictions

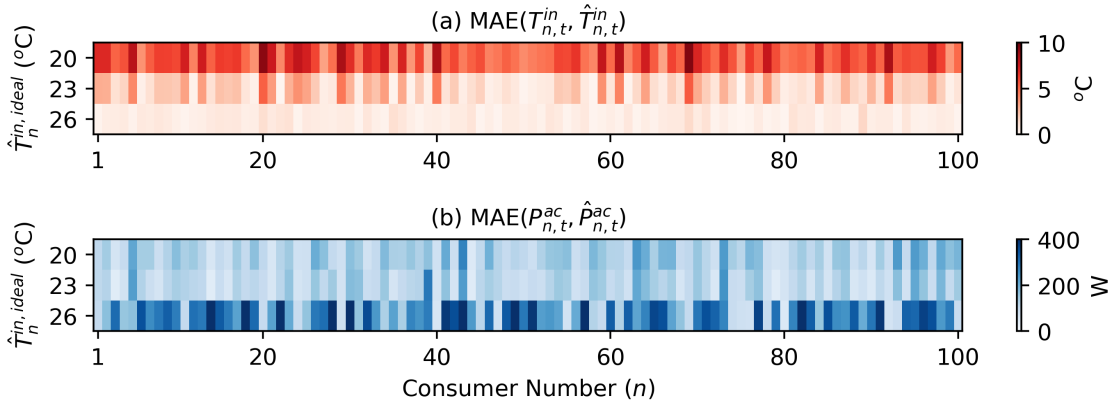

Figure IV.9: (a) Mean Absolute Error (MAE) of the indoor temperature (b) Mean Absolute Error (MAE) of the AC power. Both are calculated over the test period under the baseline policy.

Figs IV.9a and b provide a visualization of the prediction errors of the ACM over the test period, for the indoor temperature and AC power respectively. In both plots, the y-axes show the $M = 3$ values of $\hat{T}_n^{in,ideal}$ used to fit the ACM, while the x-axes are consumer indices n. The relatively high errors in Fig IV.9a reflect the difference between $\hat{T}_n^{in,ideal}$ and $T_n^{in,ideal}$, since the AC keeps the indoor temperature close to the ideal value in the baseline policy. The error is systematically higher for $\hat{T}_n^{in,ideal} = 20°C$ than for $26°C$,

[8] Such an agent is not practical for real-world application due to the long training time (~4.5M steps, ie, just over 500 years).

suggesting that the actual $T_n^{in,ideal}$ is generally closer to the $26°C$. It should be noted that the AG has no way to know this, motivating the use of ACMs to train agents that are robust to any $T_n^{in,ideal}$. The Mean Absolute Error (MAE) in the AC power is considerably lower, typically equivalent to around 15% of $P_n^{ac,max}$[9] for any $\hat{T}_n^{in,ideal}$. This is because the ACM parameters are fit by minimizing this error.

4.c Peak Reduction and Effect on Load Profile

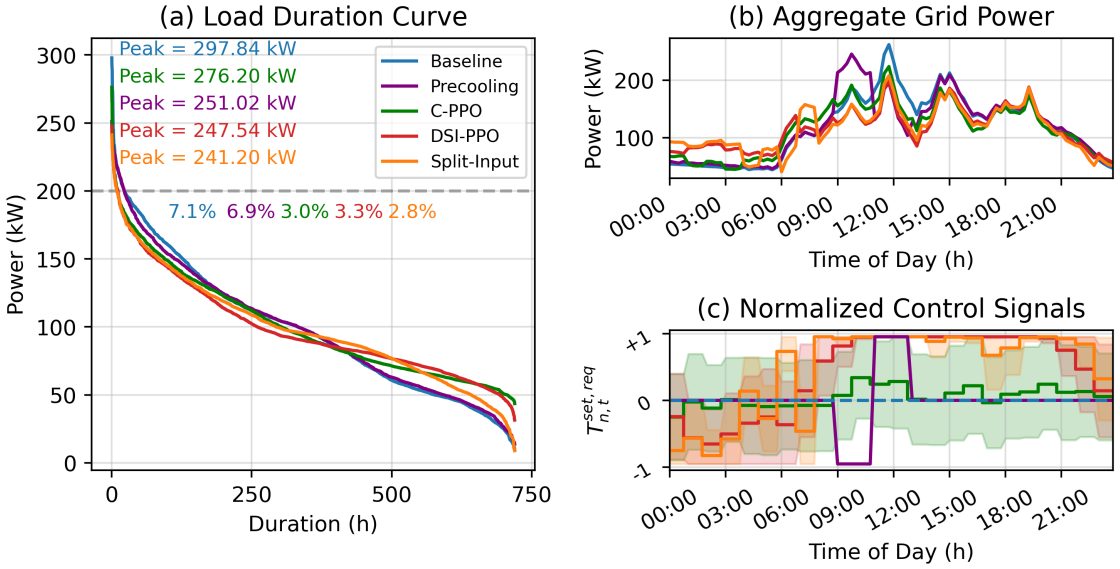

Figure IV.10: (a) Load duration curves (annotations are the fraction of energy consumed above the highlighted power) (b) Aggregated grid power for a typical day (c) Control signals under each strategy (mean and std deviation across all consumers.)

Fig IV.10a compares the load duration curves achieved by different strategies over the test period, with annotations showing the fraction of energy consumed above an aggregated power of 200 kW. All the DRL agents, as well as the precooling strategy, are able to reduce the peak system load, with the Split-Input agent achieving a 20% reduction compared to the baseline. An additional benefit of the DRL control is that a larger fraction of the load is consumed at lower aggregate power. This is of importance since the duration of peak loads directly affects the period of time for which peaking power plants need to be activated, leading to higher costs and emissions [170].

The aggregate grid power ($\sum_n P_{n,t}^{grid}$) for an example weekend day is shown in Fig IV.10b, while the corresponding control signals ($T_{n,t}^{set,req}$) are plotted in Fig IV.10c (the DRL agents generate N signals at every step t). Since the precooling signals are sent at the same time to all consumers, a new peak appears in that window due to the increased AC load. In contrast, by spreading out $T_{n,t}^{set,req}$ over time and across consumers, the DRL controllers are able to flatten the aggregated power profile more effectively. The C-PPO agent exhibits a bigger variance in its action selection, while the Split-Input and DSI-PPO agents learn a more direct strategy of precooling the buildings in the early hours of the morning to request high setpoints later in the day.

Table IV.3 displays the values of different objective metrics under each control strategy. Here, the Peak Energy Consumption is the energy consumed when $\sum_n P_{n,t}^{grid} > 200$ kW,

[9]Mean value of $P_n^{ac,max}$ across consumers in the real system is 1500 W.

Table IV.3: Evaluation Metrics for Different Strategies (Mean and Std for Split-Input)

Strategy	**Peak Energy Cons** (MWh)	**Total Energy Cons** (MWh)	**Override Fraction**
Baseline	5.002	70.723	0.0
Precooling	4.843	70.640	0.019
C-PPO	2.139	72.252	0.104
DSI-PPO	2.303	70.837	0.199
Split-Input	2.051 (0.329)	70.915 (0.339)	0.189 (0.017)

and the Override Fraction shows the fraction of agent control signals that were overridden by consumers. A key observation is that the Split-Input agent achieves a similar level of performance to the C-PPO and DSI-PPO agents, despite having a significantly lower parameter count and training time.

The DRL-based strategies result in a 10-20% higher override fraction compared to the baseline, which indicates that the peak reduction benefit comes at the cost of discomfort for the consumer. In comparison, the precooling strategy leads to much lower discomfort levels, which represents a major advantage. It must be noted that in all cases, the discomfort never exceeds the threshold value of D_n^{max} as the consumer is allowed to override setpoint requests. Certain techniques, such as penalizing high setpoint requests in the reward function, may help to reduce the discomfort caused by the agent's actions.

Incentive schemes in real-world DR programs depend on factors such as the power mix, electricity markets and consumer tariff structures - as such, a detailed discussion is beyond the scope of this work. Given the variation in discomfort encountered, any incentive offered to consumers must ensure that participants are compensated in proportion to their response to the control signals. The override feedback signal ($o_{n,t}$) would serve as a suitable mechanism to measure this response - in its absence, the task of identifying the most responsive consumers is very difficult, and is the subject of open research. Using the weights in the reward function, the proposed approach may be adapted to meet the needs of the AG depending on the context.

4.d Impact of Online Adaptation

To evaluate the effect of online adaptation (Section 3.d), the Split-Input agent was used as π^{init} and deployed with online updates over a 4-month period. For comparison, the same procedure was repeated using the Precooling agent as π^{init}. Table IV.4 reports the evaluation metrics for different policies, with $T^{init} = 15$ days and $T^{upd} = 10$ days, which were found empirically to yield the best results.

In this scenario, given the 120 day test duration, there are 10 update steps, so that $\pi^{end} = \pi^{10}$. The impact of the online training is demonstrated by comparing 3 sets of results for each π^{init} - only π^{init}, $\pi^{init} \cup \pi^{upd}$ and only π^{end}. The first case evaluates the underlying policy without online updates, the second represents the proposed method, and the third is a hypothetical evaluation of the final online policy in the same test system (ie, assuming π^{end} is available at the beginning of the test period, and is applied for 4 months without any update).

The online learning improves the underlying policy in both cases, reducing peak energy consumption by up to 70% and overrides by up to 100%. The marginal increase (2-3%) in the total energy consumption is an acceptable trade-off, since this increase occurs outside of peak periods[10]. The final policy (π^{end}) performs significantly better than

[10]Depending on the electricity mix, this may lead to better utilization of renewables or other clean energy

Initial Policy	Precooling			Split Input		
Metric	Only π^{init}	$\pi^{init} \cup \pi^{upd}$	Only π^{end}	Only π^{init}	$\pi^{init} \cup \pi^{upd}$	Only π^{end}
E^{peak} (MWh)	13.65	6.45 (-52.75%)	4.46 (-67.33%)	7.88	5.04 (-36.04%)	3.65 (-53.68%)
o^{tot} (Count)	5741	1108 (-80.70%)	49 (-99.14%)	55989	31659 (-43.46%)	20348 (-63.65%)
E^{tot} (MWh)	274.36	283.96 (+3.49%)	285.43 (+4.03%)	279.88	284.91 (+1.79%)	282.97 (+1.10%)

Table IV.4: Evaluation metrics obtained under different policies. Values in brackets show percentage change from π^{init}.

the initial (π^{init}), suggesting the potential for cumulative gains over a longer deployment period.

Most importantly, the offline-to-online RL process ($\pi^{init} \cup \pi^{upd}$) also leads to an improvement over the underlying policy. This demonstrates the feasibility and practicality of the approach, since the benefits are obtained over a relatively short duration (4 months) while avoiding pitfalls such as catastrophic forgetting. While the approach retains the knowledge embedded in π^{init}, it also introduces a strong bias, as demonstrated by the differences in evaluation metrics when starting from different policies. A detailed study of this offline-to-online transfer is presented in [92]. The rest of this section discusses different aspects of the model-based learning procedure.

4.e Sensitivity to Discomfort Models

The results in 4.c assume that the override model in the test system follows Eq. IV.2, where overrides occur through a deterministic process based only on thermal comfort. Recent studies suggest that real-world override behaviours are stochastic processes [42], and may dependent on several other factors [41, 171]. To evaluate the robustness of the approach to unseen override models, the Split-Input agent (trained using the comfort-based deterministic override model, denoted as CB-Det) was tested against the following alternate override models.

1. **No Overrides** : The consumers accept all setpoint requests from the agent, without any overrides.

2. **Comfort-Based Stochastic (CB-Sto)** : A stochastic reformulation of the thermal comfort-based override model in Eq. IV.2, with $p(o_{n,t} = 1) = \max(\frac{D_{n,t}}{D^{max}_{n,t}}, 1.0)$. This model is analogous to the indoor temperature-dependent probabilistic model presented in [42].

3. **Multifactor Stochastic (MF-Sto)** : A stochastic decision-tree model in which the leaf nodes represent override probabilities. It is based on the analysis presented in [41], where real-world DR overrides were found to depend on (in order of feature importance) a consumer's level of interaction with their thermostats[11], ambient and indoor temperatures, and the number and intensity of recent setpoint requests.

sources.

[11] Here, consumers were uniformly distributed into 3 interaction levels - low, medium and high

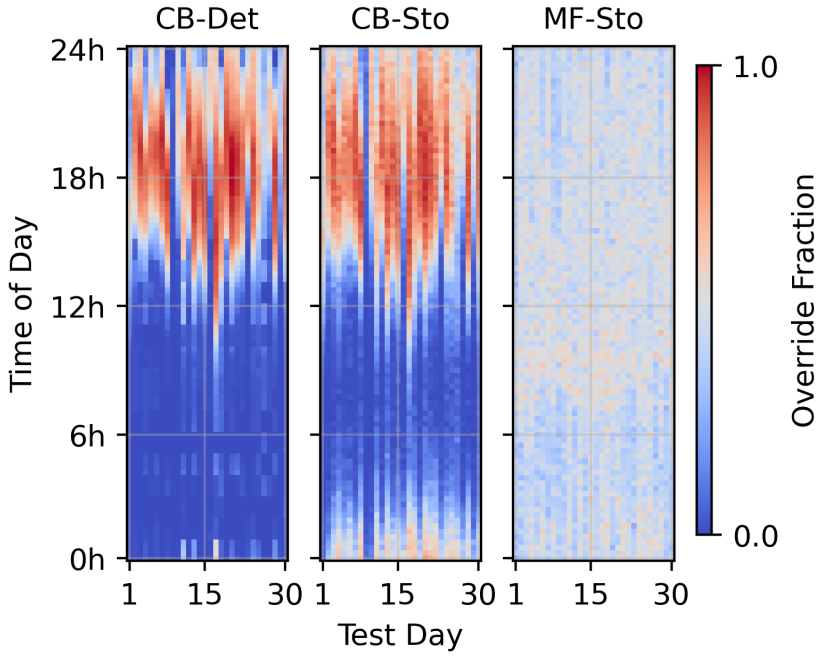

Figure IV.11: Consumer override patterns under different override models with the Split-Input agent. CB-Det represents the deterministic model used to train the agent.

Fig IV.11 shows the fraction of overrides under the Split-Input policy over the test period for different override models. In the thermal comfort-based models, overrides are strongly concentrated during the late afternoon, with almost all the signals being overridden at around 6 PM. In contrast, in the multi-factor model, overrides are almost equally distributed throughout the day, which is a pattern completely unseen during the agent training. It should be noted that the total number of overrides are comparable in all cases.

Table IV.5: Performance Comparison Across Override Models

Override Model	**Peak Energy Cons** (MWh)	**Total Energy Cons** (MWh)	**Override Fraction**
Baseline	5.002	70.723	0.0
No Overrides	0.503 (0.087)	61.664 (1.109)	0.0
CB-Det	2.051 (0.329)	70.915 (0.339)	0.189 (0.017)
CB-Sto	3.040 (0.526)	71.630 (0.245)	0.313 (0.016)
MF-Sto	1.350 (0.172)	68.583 (0.502)	0.447 (0.002)

Table IV.5 shows the mean and standard deviation of evaluation metrics under each override model across multiple runs with different random seeds. In all cases, the DRL agent is able to reduce peak loads (between 20% - 90%) without significantly changing the total energy consumption. This suggests that the learned policy is applicable to stochastic override behaviours that may occur in the real world even if it is trained on a deterministic model. The best metrics occur when overrides are absent in the system - in this study, the presence of overrides degrades the peak reduction performance of the agent by around 30% on average, which is consistent with the findings in [42].

When overrides are present, the benefits of the methodology strongly depend on the

nature of the override decision process. For the comfort-based models used in this work, the stochastic reformulation reduces the agent's effectiveness and leads to higher energy consumption (both peak and total), as well as a bigger override fraction. Surprisingly, a higher fraction of overrides does not necessarily imply a degradation in the objective metrics. In the multifactor override model, the agent achieves significantly better metrics (50% lower peaks) compared to the comfort-based case, despite a higher fraction of overrides (10% higher). This can be attributed to the difference in override distributions resulting from different models - in the comfort-based override models, higher peak loads occur due to the concentrated nature of override signals, as seen in Fig IV.11. In this sense, the multifactor model may be viewed as a stochastic extension of the case without overrides in which around 45% of the agent's actions are overridden (leading to a 20% increase in peak loads.).

4.f Impact of Actor-Critic Architecture and ACM Accuracy

Fig IV.12 shows violin plots of $\sum_n P_{n,t}^{grid}$ for different control strategies over the test period. For DRL-based approaches, the results from using perfect models are plotted alongside the ones based on ACMs, denoted as "Perfect Models" and "ACM" respectively. The choice of architecture has a significant impact on the performance of the agent. Compared to the baseline, the peak reduction performance improves when using the Split-Input architecture, and approaches that of the Full-Input case (where behind-the-meter information is accessible from the real system). This suggests that the proposed architecture represents a good trade-off between performance and input data requirements. Notably, since all the DRL architectures share the same reward function, it is possible to use the technique even when there are different levels of data access for different consumers by deploying multiple agents.

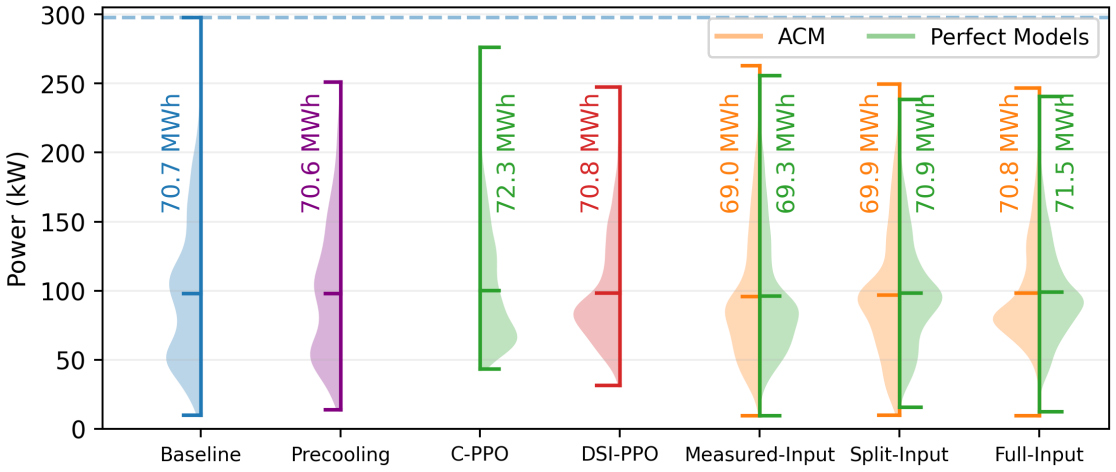

Figure IV.12: Violin plots of the aggregate grid power for different strategies (annotations show total energy consumption over the test period)

Another observation is that under the Measured-Input case, the energy consumption is lower and a higher fraction of the load occurs at lower power levels (eg, below 75 kW). While it may seem like a positive outcome at first glance, this actually happens when agents adopt a conservative strategy of sending high setpoints more often to exploit any available flexibility (discussed subsequently). This may reduce the energy consumption, but it causes a higher level of discomfort to consumers.

The peak reduction performance of the agent improves with the accuracy of the models, with a systematic improvement of around 3% in all cases compared to using approximate models (Fig IV.12). This suggests that the agents are able to learn competent policies using the ACMs despite their inaccuracy, since the problem dynamics are well-captured. Improving the model accuracy can then help to fine-tune the policies to suit specific consumer groups. Since the relative improvement is nearly the same across all RL architectures, it can be inferred that the fine-tuning effect results from accuracy improvements in inputs that are shared among all architectures (in this case, only the load $P^{grid}_{n,t}$).

Another interesting observation is that the performance achieved with perfect models is nearly identical in the Split-Input and Full-Input cases. This implies that by utilizing the actor-critic architecture described in this paper and improving the model accuracy, the need to obtain real measurements of behind-the-meter data during the deployment phase can be avoided. In addition to saving on the economic costs of implementing extra sensing infrastructure, such an approach is also more desirable for the consumer, since their data privacy is respected.

The results also highlight the effectiveness of the domain randomization techniques employed in this work. By training a single Split-Input agent on an ensemble of approximate consumer models - despite their limited accuracy, as shown in Fig IV.9 - the agent is able to learn a generalized policy that captures common behavioral patterns across consumers. This shared training approach enables stronger generalization than the DSI-PPO agent, which relies on separate, personalized agents for each consumer but fails to outperform the Split-Input baseline. Similar findings have been reported in prior work on multi-agent RL (MARL) [172], where shared policy training has proven beneficial.

While domain randomization in this work is used primarily to account for variations in consumer behavior, the same technique can be extended to model a broader range of real-world uncertainties, such as sensor noise, occupancy variation, and communication delays, to enhance the robustness of the trained policies. Such application-dependent extensions have been explored in other domains including robotics [124] and electric vehicle charging [173]. The general framework proposed here (offline training using informed critics on an ensemble of approximate models) is agnostic to model fidelity and can be readily adapted to other domains that require sim-to-real transfer.

4.g Sensitivity to Disaggregation Errors

As discussed in 3, it is assumed here that it is possible to perfectly disaggregate the building and AC loads from historical data. This information is used at two key stages in the methodology - first, during ACM thermal parameter fitting, and second, for the consumer electrical loads used in the training environment. To analyze the impact of imperfect disaggregation on the outcomes achieved using this approach, we systematically introduce noise of different levels [174] with a parameter ϵ, and train Split-Input agents using the noisy data. Here, for the RC parameters, it is used multiplicatively (for eg, $\delta R^{in}_n = R^{in}_n \times \epsilon$) and for the time-series load profiles, it is used additively (sampling from $\mathcal{N}(0, \text{mean}(P^{grid}_n) \times \epsilon)$). The evaluation metrics under different values of ϵ are given in Table IV.6.

All agents were able to learn strategies that reduce peak loads compared to the baseline case, suggesting that the proposed approach is not severely affected by disaggregation errors. Agents that were trained on noisy data perform slightly worse than those trained under perfect disaggregation with 10% higher peak loads and 5% more overrides on average. The architecture and training approach used here, based on grey-box models, encourage agents to learn generalizable policies based on system-level dynamics.

Table IV.6: Effect of Noise Due to Disaggregation Errors

Noise Level	**Peak Energy Cons** (MWh)	**Total Energy Cons** (MWh)	**Override Fraction**
Baseline	5.002	70.723	0.0
No Noise ($\epsilon = 0.0$)	2.051 (0.329)	70.915 (0.339)	0.189 (0.017)
5% ($\epsilon = 0.05$)	2.441 (0.171)	70.256 (0.372)	0.222 (0.028)
27.5% ($\epsilon = 0.275$)	2.433 (0.630)	70.100 (0.332)	0.237 (0.033)
50.0% ($\epsilon = 0.5$)	2.636 (0.149)	70.261 (0.066)	0.236 (0.009)

These experiments demonstrate the robustness of the proposed approach to model errors arising from imperfect disaggregation. In practice, Non-Intrusive Load Monitoring (NILM) errors can originate from multiple sources, including overlapping appliance signatures and limited sensor resolution [175]. This work assumes the existence of a disaggregation algorithm and focuses solely on downstream control performance. Additionally, the disaggregation is not required during the deployment phase since the actor networks operate directly on meter data. As such, a comprehensive evaluation of various disaggregation methods and their impact on agent performance is beyond the scope of this study.

4.h Analysis of the Split-Input Architecture

Fig IV.12 shows that the Split-Input architecture improves evaluation metrics compared to the Measured-Input case despite both using an actor with a reduced input space. To understand how providing additional inputs to the critic enables the training of a better actor network, we study the impact of this paradigm on the training process, the hidden activations of the DNNs, and on the final learnt policy.

4.h-i During Agent Training

Fig IV.13 shows the mean rewards and the Explained Variance (ev) score (of the critic network) for different architectures over the training duration. The ev score, calculated here between the estimated and actual cumulative rewards obtained by the agent, is a numerical measure of the accuracy of the critic network (ev$\in [-1, 1]$, higher values are better). While both architectures have a similar training reward curve, the ev score for the Split-Input architecture is considerably higher, suggesting that the critic in this case is able to learn a better estimate of the true reward function.

In actor-critic RL algorithms, the output of the critic is used to calculate the gradients used to update the actor network [107]. As such, an accurate critic reduces the variance of these gradient steps, guiding the actor towards a more stable policy. It should be noted that the training reward in this case is calculated from an approximate model of the real system. It is possible for two policies to achieve similar rewards in the training environment while performing differently in the actual system.

4.h-ii Visualizing Hidden Activations of the DNNs

Both the actor and critic are parametrized by DNNs in this study (with 2 hidden layers of 64 neurons), and the outputs of their hidden layers encode the key features used by the networks during inference [176]. Fig IV.14 shows a coloured T-Distributed Stochastic Neighbor Embedding (t-SNE) projection [177] of the 64-dimensional hidden layer activations for different DNNs for a sample of 2000 states, equally split between those with

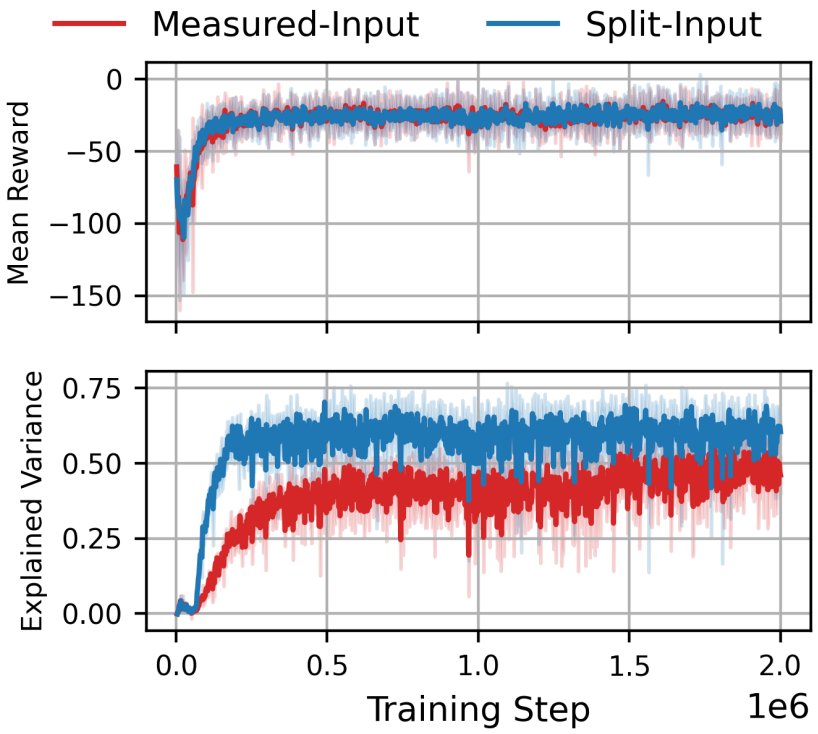

Figure IV.13: Mean training rewards and Explained Variance (ev) scores for different agents during training. ev is a measure of the accuracy of the critic output (higher values are better, maximum value is 1.0).

$o_{n,t-1} = 1$ (orange) and $o_{n,t-1} = 0$ (blue). If the states belonging to each label are well-separated in the projection space (measured by the silhouette score, $s \in [-1, 1]$, where higher values indicate better separation), it is a clear indication that the feature plays an important role in the DNN output. It should be noted that the actor and critic share a common embedding space in the Measured-Input case, since they have the same inputs, and are therefore represented by one panel in Fig IV.14.

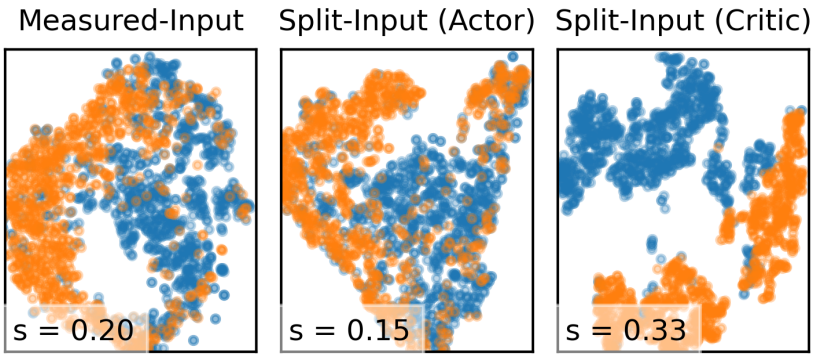

Figure IV.14: t-SNE plots of hidden activations of each neural network for different states. Colours indicate whether an override occurred in the previous time step (Yes, No). s is the silhouette score, with higher values indicating a better separation between the labels.

The Split-Input critic attaches a high importance ($s = 0.33$) to the presence of overrides

when estimating the value function compared to the Measured-Input case ($s = 0.20$), as evidenced by the significantly higher silhouette score for this feature. This indicates that one of the reasons for the improved critic accuracy noted in Fig IV.13 is that the critic is able to better approximate the long-term effects of overrides on the cumulative reward at the end of the training episode. Interestingly, this marginally *reduces* the importance attached by the Split-Input actor to this feature ($s = 0.15$) compared to the Measured-Input actor. This could reflect the fact that while overrides play a critical role in estimating the value function (ie, the critic output), the optimal policy (ie, the actor) should focus on other factors to maximize the achieved reward. The absence of additional inputs thus leads to two effects - the Measured-Input critic output has a higher variance as it is unable to infer override behaviours from limited data, while the actor slightly overestimates their potential impact.

4.h-iii Implications at the Policy Level

Under the Measured-Input and Split-Input paradigms, the agent's decision depends on the building load ($\hat{\mathbf{P}}^{\mathbf{grid}}_{\mathbf{n}}$), the aggregate load of the other buildings ($\sum \mathbf{P}^{\mathbf{grid}}_{\sim\mathbf{n}}$), the weather ($\hat{\mathbf{T}}^{\mathbf{amb}}$) and the time of day ($t$). Fig IV.15 visualizes the policy in each actor-critic architecture (columns) under different weather conditions (rows). Each subplot shows the value of $T^{set,req}_{n,t}$ as a function of the mean $\hat{\mathbf{P}}^{\mathbf{grid}}_{\mathbf{n}}$ and $\sum \mathbf{P}^{\mathbf{grid}}_{\sim\mathbf{n}}$. Numerical annotations in the subplots show the fraction of the decision space where $|T^{set,req}_{n,t}| < 0.5$, used here as a measure of policy entropy.

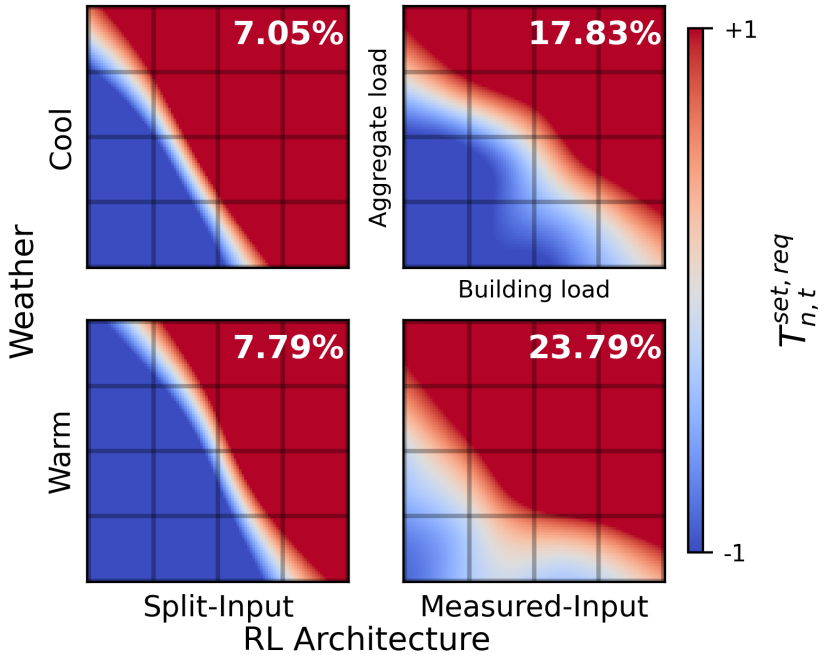

Figure IV.15: Visualization of agent policies in Split-Input and Measured-Input architectures for different weather. Subplot x- and y-axes show mean values of $\hat{\mathbf{P}}^{\mathbf{grid}}_{\mathbf{n}}$ and $\sum \mathbf{P}^{\mathbf{grid}}_{\sim\mathbf{n}}$ respectively. Annotations are the fraction of the decision space where $|T^{set,req}_{n,t}| < 0.5$, used here as a measure of policy entropy.

It is clear that the agents learn a similar mapping in both cases, with higher setpoints being requested as the building load and the aggregate load increase. In the Measured-Input case, a larger fraction of the decision space is assigned to higher setpoints, and

the decision boundary is less sharp compared to the other case (with about 15% higher entropy). This illustrates the conservative policy adopted by the agent in the absence of additional context provided by the model-estimated inputs. By requesting higher setpoints more often, the agent runs the risk of more overrides (and higher consumer discomfort), which affects its peak reduction performance. In contrast, the policy learnt with the Split-Input architecture has a better distribution between actions, as well as a sharper decision boundary. It is clear that the agent is able to use the additional inputs to obtain a clearer understanding of the system dynamics, and adjust its policy to obtain a better outcome.

This is supported by the dichotomy in the agents' response to changing weather signals. In the Measured-Input case, as the ambient temperature increases (downwards in Fig IV.15), the decision boundary moves leftwards, increasing the chance for higher setpoints to be requested. This follows the intuitive notion that cooling loads are proportional to the ambient temperature - as such, the agent tries to reduce this load using higher setpoints. Interestingly, the agent with the Split-Input architecture learns the opposite - hotter weather pushes the decision boundary rightwards, giving a *higher* chance for low setpoints to be requested. This may be because the agent learns that consumers face a higher level of discomfort during hot weather, and have a higher probability of overriding the requested signals. The policy thus allows for more aggressive precooling during hot weather, in the expectation that consumers will be more amenable to higher setpoints later in the day. This results in a trade-off between energy consumption and peak reduction, as visible in Fig IV.12.

4.h-iv Theoretical Justification

While the analysis in this section focuses on empirical metrics, we note that the findings align with the theoretical justification for asymmetric actor-critic architectures presented in [136]. In that work, the authors prove that the addition of more information to the critic improves its accuracy by eliminating a specific source of error (termed "aliasing error"). They also prove that a more accurate critic is theoretically guaranteed to result in a more optimal policy, due to a stronger and more stable gradient signal to update the actor. However, this is only true if the actor has sufficient inputs to take advantage of this signal, failing which the optimality of the policy is limited by the error of projecting critic gradients (calculated using a higher-dimensional input space) into the actor space. Finally, while [136] assumes that the actor and critic use linear models, a separate proof is presented in [178] for the convergence of actor-critic RL with deep neural networks, which are used in the present work.

4.i Runtime, Reproducibility and Robustness

Task	Cumulative	Per Unit	#Units
Generate ACMs	36h55m± 12m15s	11m4s±40s	100 ACMs
Train Split-Input Agent	37m52s±55s	37m52s±55s	1 Agent (11650 parameters)
Train C-PPO Agent	19h57m21s	19h57m21s	1 Agent (43521 parameters)
Offline-to-Online Transfer	19m17s ± 1m1s	19m17s ± 1m1s	1 Agent (6400 parameters)
Perform One Online Update	1m15s ± 2s	1m15s ± 2s	1 Agent (6400 parameters)

Table IV.7: Runtime for different stages

The typical runtime for each step in the solution is tabulated in Table IV.7 for experiments performed on a laptop with an AMD Ryzen 9 5900HS processor and 24.0 GB of RAM. It is evident that the model generation takes the longest cumulative time - however, the runtime may decrease significantly with parallelization across multiple CPUs, which was not done in this study. The negligible runtime of the deployment step ($\sim 500\mu$s), which requires just a forward pass through a deep neural network, demonstrates the scalability of the solution. Since modern deep learning libraries are optimized to process inputs in batches, the marginal cost of performing multiple inferences is very low.

Table IV.7 also highlights a key advantage of the architecture proposed in this paper - the time required to train an agent is 40 times lower than that required for a fully centralized architecture, and requires only a quarter of the parameter count. Additionally, since the input and output dimensions in the C-PPO case depend on N, the inclusion (or removal) of even a single participant from the scheme would necessitate the training of a new agent. The choice of sharing network weights while making decisions using each consumer's local state eliminates this problem, which greatly simplifies real-world deployment. In comparison, the decentralized DSI-PPO approach requires N times the number of parameters and training time compared to the Split-Input case, which may become prohibitively expensive as N increases. Additionally, this approach raises privacy concerns, since consumers might not be willing to allow their data to be used to train and deploy targeted agents.

An important concern with RL based solutions is the issue of reproducibility, due to the inherent variance in most algorithms and stochasticity in the environments. While several recommendations in [179] have been followed in this work to encourage reproducibility, the robustness of the method to random variations can only be quantified empirically. Figs IV.16a and b show the average rewards obtained during training and violin plots of the aggregated load in the test period respectively for 10 repeated runs of the Split-Input architecture experiment, each with different random seeds[12]. The learning process of the agent is relatively stable since the rewards converge to the same high value within the training period irrespective of the seed. The agents also consistently reduce peaks compared to the baseline, with only a 5% deviation between the best and worst results. The policies achieve a similar load distribution with a sharp decrease in the energy consumption above 200 kW. This suggests that the PPO-based training

[12]This changes the order in which models are sampled from $b_{n,m}$ during training, in addition to algorithmic randomness. All runs still use the same data.

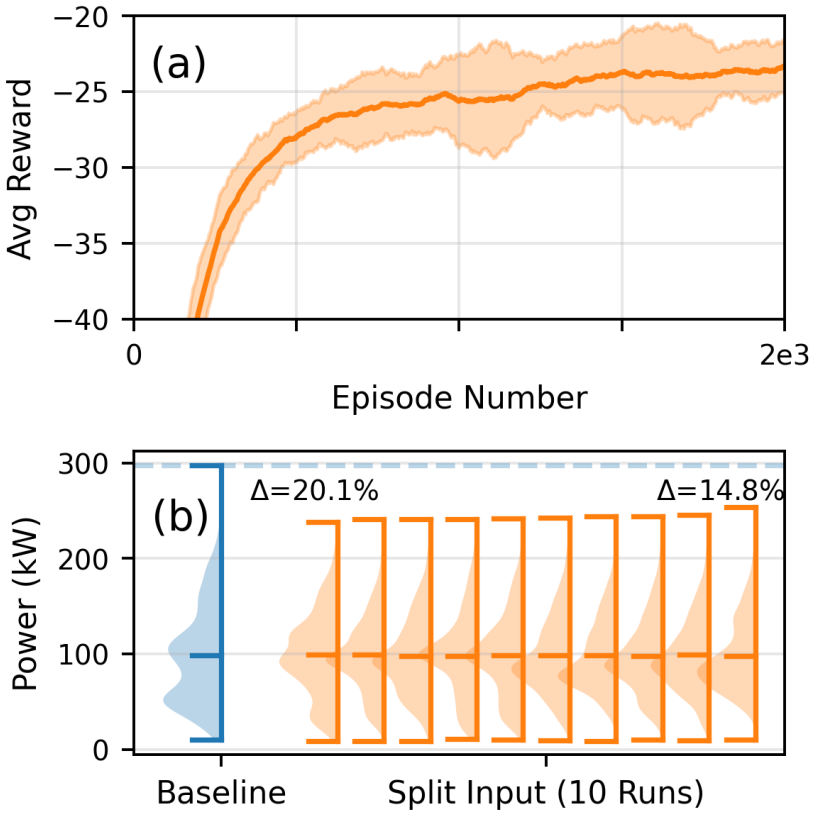

Figure IV.16: (a) Avg episode reward in training and (b) Violin plots of the aggregated load for 10 runs of the split-input architecture with different random seeds. Δ is the percentage peak reduction compared to the baseline.

approach used in this paper is robust to algorithmic variance.

5 Summary & Conclusion

A novel demand response scheme based on the co-ordinated control of air-conditioner setpoints in a group of residential buildings is proposed, based on grey-box modelling and actor-critic reinforcement learning. A key feature is that consumers are allowed to freely override the setpoint signals sent by the acdrl agent based on their thermal comfort, preserving their privacy and promoting interaction. Using a novel Split-Input architecture, the approach is able to achieve a significant peak reduction (of up to 20% from the baseline) in the aggregated electrical consumption, while eliminating the need for "behind-the-meter" information such as indoor temperature signals. An offline-to-online transfer approach was used to further refine the agent through direct interaction with the system, which was found to improve its performance over time. The impact of the Split-Input architecture was investigated by visualizing the agents' policies. The sensitivity of the solution to the accuracy of the models and the quality of the training data was quantified, and the impact of the signals on the consumers' thermal comfort was studied.

The main factors affecting the peak reduction performance are the actor-critic architecture and inputs, while the model accuracy has a fine-tuning effect. The use of inter-

pretable consumer models that respect the temperature dynamics of the system resulted in agents that were robust to different consumer behaviours, and generalized well to the problem. The final policies were more stable and reflective of a deeper understanding of the system despite having limited access to data. The performance of the methodology also depends on external factors, specifically on the distribution of overrides over the day. Adversarial patterns, such as a high concentration of overrides during specific periods of the day, inherently reduce the ability of the agent to reduce peaks. For a given distribution, it was found that stochasticity in the override model further degrades the acdrl agent performance by around 5%. However, the learnt policy remains applicable to different override models, and reduce peak loads by atleast 20% in all cases. Finally, it was found that while the control signals are able to decorrelate the consumption profiles and reduce aggregate peaks, certain consumers faced a significant increase in their discomfort, although within their discomfort capacity.

The results of the study hinge on two key assumptions about the system - the AC operation model and the possibility of load disaggregation. While the same AC model is widely used in literature, and may be sufficient to describe residential cooling systems, the impact of different technologies (such as inverter-based ACs) needs to be studied in future work. Disaggregation of smart-meter data into HVAC and non-HVAC components is also not a trivial problem, and is an open research topic. However, for the particular case here, where historical data of both load and weather are available, along with a known operation model of the AC, recent work has achieved promising results [156, 157, 158]. A noise-sensitivity analysis shows that the disaggregation errors may reduce the agent performance by around 10% - however, the approach still results in a significant peak load reduction compared to the baseline in all cases.

While the present study relies on the use of grey-box models to capture the system physics, other approaches may result in better performance. Future work should focus on alternate model creation strategies, such as Physics-Informed Neural Networks [180], and provide a more detailed analysis of the most important inputs to enhance the performance of the RL agents. Another limitation of the current work is that a detailed comparison is not provided other advanced control algorithms such as Model Predictive Control. While the literature review highlights the difficulty of scaling such techniques to hundreds of consumers, future work should investigate alternate control approaches, and quantify the relative tradeoffs associated with each technique. Finally, while the present work highlights the importance of the override model on agent performance, future work should focus on online adaptation techniques to adapt the RL policy to real-world override signals in a data-driven manner.

Ongoing Real-World Trials

Validation studies for the coordinated control approach presented in this chapter are planned through a testbed at the GreEn-ER building in Grenoble, France [181]. This consists of a set of rooms including a classroom, an exhibition room and six offices covering a total area of about 600 m^2, with dedicated indoor air setpoints for each room. The proposed study seeks to coordinate the heating power usage of the testbed to respond to DR signals using the methodology presented here. Preliminary results from this study are expected to be released in 2026.

Chapter V

General conclusion

In the global effort to reduce greenhouse gas emissions, with targets such as NZE2050, buildings have become a key focus given their contribution of nearly 30% of energy sector emissions [182]. This thesis has addressed the problem of building control for DR flexibility provision, considering both the individual scale (behind-the-meter resources such as HVAC systems and BESSs) and the collective scale (demand response across groups of consumers at the point of common coupling). At each level, advanced control strategies based on RL were developed and evaluated to demonstrate how building resources can be coordinated in a grid-aware manner, thereby contributing to large-scale deployment of flexibility services.

The discussion began with an analysis of existing control approaches (Chapter I, Section 2), comparing their relative strengths and limitations. Broadly, control methods can be categorized into RBCs, model-driven, and data-driven approaches. RBC, while widely used due to their simplicity, are often suboptimal and inflexible. Model-driven controls achieve higher performance but require detailed system models and significant engineering effort. Data-driven methods promise comparable performance without explicit models, yet current approaches face practical challenges that limit real-world adoption. As summarized in Chapter I, Section 3.b and [15], these challenges are broadly classified as into those related to learning, infrastructure, costs & benefits, and safety, security & trust.

A wide range of hybrid approaches have been proposed in recent years, which combine elements from different families of control to address these domain-specific challenges. As discussed in detail in Chapter I, Section 3, these approaches broadly focus on either integrating data-driven models into model-driven workflows, or on incorporating analytical models into data-driven workflows. A range of methods have been proposed, including the use of reduced-order grey-box models in MPC formulations [108, 109, 110], imitation of rule-based or MPC controllers through behavioral cloning [91, 97], and physics-informed or safety-constrained RL techniques [100, 104, 102]. We argue that these approaches can be seen as a form of domain knowledge integration into BEM control approaches.

Research gap in BEM for DR

A review of the literature revealed a key limitation in research on advanced control schemes for DR in BEM. Although academic interest has grown rapidly in recent years [19, 57], real-world adoption remains limited [17] and falls short of what is needed to achieve NZE2050 targets [1]. Consumer surveys indicate that the primary barriers are social and behavioral, with participation influenced by concerns over comfort [20], pri-

vacy [38], and perceived loss of control [31]. In contrast, most existing research emphasized technical improvements to control performance [117]. The work in this thesis addresses this research gap by proposing control schemes that offer high performance while remaining practical to implement, and importantly considers both the technical and social aspects of BEM for DR.

1 Main Contributions

1.a Knowledge-Informed RL for BEM

In this thesis, we formalized the integration of domain knowledge into data-driven control workflows as a means to address the aforementioned challenges. This approach was framed within the "Knowledge-Informed RL" paradigm introduced in Chapter II Section 1, which also presented a taxonomy of domain knowledge types in BEM. As discussed in Chapter II Section 2, this knowledge may take the form of building thermodynamics, simplified computational models, established operational strategies, or human feedback. To support the development of knowledge-informed approaches, we introduced an extended RL workflow that spans the entire process - from problem specification and MDP formulation to post-hoc policy analysis and refinement (Chapter II, Section 3). These steps are illustrated in Fig V.1 (reproduced from Fig II.2).

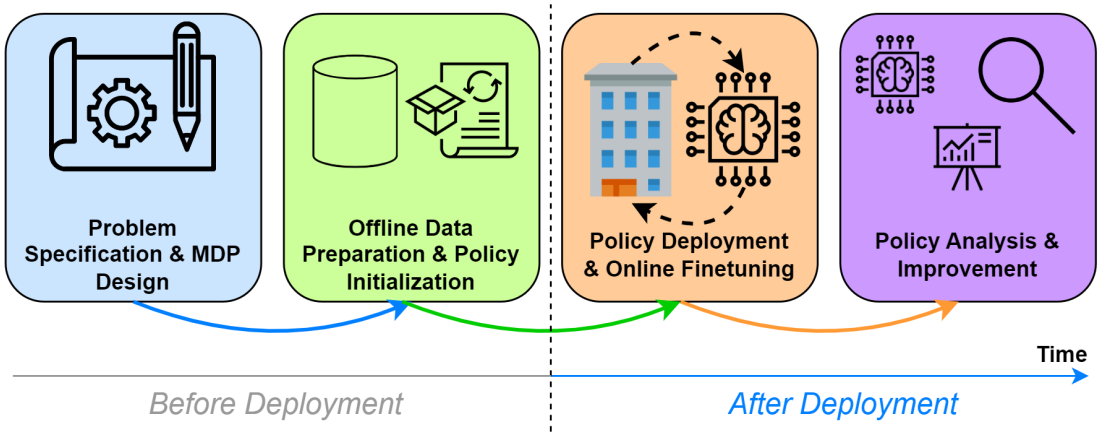

Figure V.1: Steps in Extended RL Workflow

1.b Techniques for domain knowledge integration

To inform and guide the RL process using different knowledge representations, a set of specific techniques were presented in Chapter II, Section 4. These techniques link specific knowledge representations to one or more challenges and can be applied at defined stages of the extended RL workflow. Fig V.2 (reproduced from Fig II.3) illustrates these techniques based on the challenges they address, and on which stage of the RL pipeline they are applicable. Each technique was subsequently described in detail, and illustrative examples were provided based on typical problems in BEM.

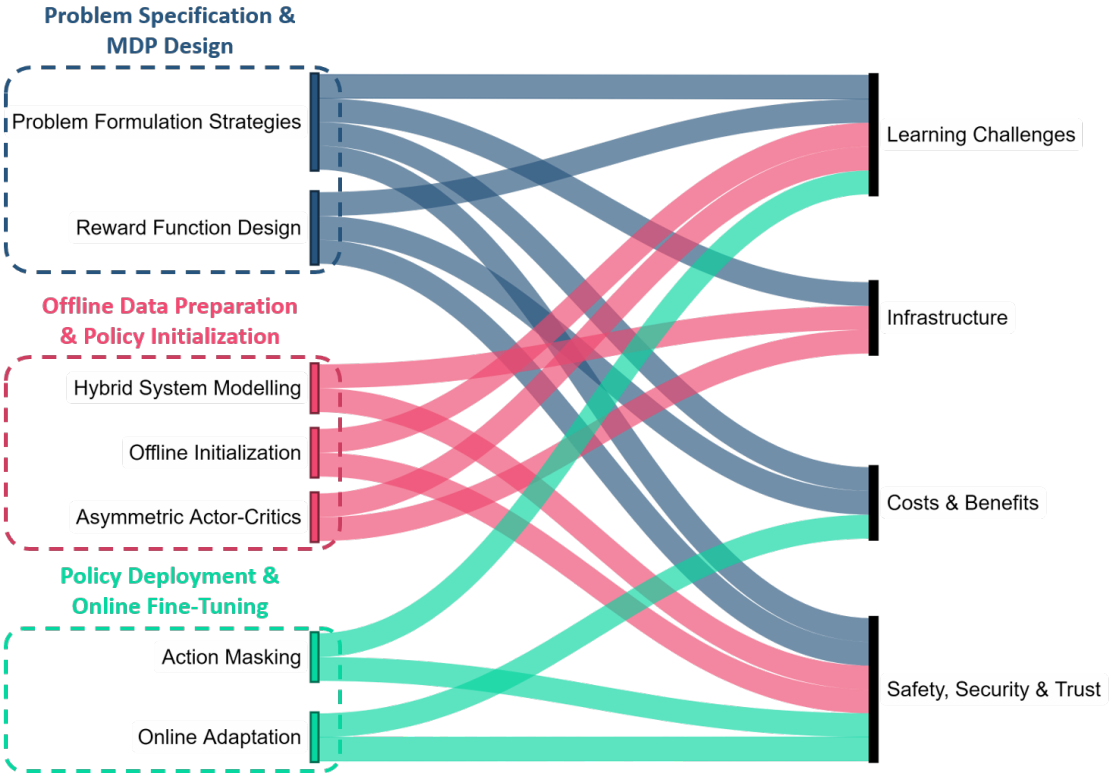

Figure V.2: Challenges addressed by different techniques

1.c Applications and Key Findings

The end-to-end application of the Knowledge-Informed RL paradigm to BEM problems was presented through two representative case studies. Chapter III addressed the design of a price-responsive controller for a single building, where indoor AC setpoints and BESS operating points were optimized under dynamic pricing. Chapter IV examined the generation of DR signals for a group of 100 consumers participating in an AC-based DR program. Together, these applications illustrated the key social and technical challenges in BEM and demonstrated how the proposed paradigm can address them while balancing objectives such as thermal comfort, energy bill reduction and price arbitrage.

At the single-building level, three techniques were evaluated in addition to the knowledge-informed problem formulation and reward design: offline initialization, action masking, and online adaptation. Offline initialization showed a clear trade-off: longer horizons allowed the control agent to internalize long-term reward effects but left limited scope for further improvement, whereas shorter horizons enabled faster adaptation but often produced unstable or suboptimal control. Action masking consistently guaranteed occupant thermal comfort by restricting unsafe actions, but this safety came with higher energy bills, as the agent was forced into underexplored actions it had not sufficiently practiced during training. These outcomes were consistent with our earlier findings [139] and theoretical analyses of masking [137]. Finally, online adaptation allowed the agent to adjust its policy after deployment, stabilizing performance in some cases but also introducing variability depending on the update frequency. In a 1-year simulation study, it was found that using all three in combination (compared to a baseline RBC) lowered the net energy bill from 1034 (ie, net import) to -187 (ie, net export) - a 118% reduction - while also improving comfort, as reflected by the PMV shifting from -0.36 to -0.03.

At the multi-building level, the limited access to behind-the-meter data was addressed

through a novel problem formulation approach, combined with hybrid system modelling, model-driven initialization, and asymmetric actor-critics. In a simulation study, the approach achieved significant peak load reductions (>20%) while preserving consumer comfort and privacy through override capability. Peak reduction performance was primarily influenced by the actor-critic design and input selection, with model accuracy playing a secondary role. The learned policies were robust to varying consumer behaviors and stochastic override patterns, though extreme or concentrated overrides can modestly reduce performance. Limitations include reliance on the AC operation model and the need for load disaggregation, which can affect performance by up to 10%.

2 Limitations and Future Work

Although this thesis examined the potential of Knowledge-Informed RL in BEM and proposes a set of techniques for incorporating domain knowledge, the coverage is not intended to be comprehensive. Many other techniques and approaches exist beyond those discussed here. Accordingly, the following sections outline the main limitations of the present work and suggest directions for future research to advance and refine the concepts introduced.

2.a Detailed review of existing work under knowledge-informed paradigm

As noted in Chapter I Sections 3 and Chapter II Section 4, an increasing share of recent research in RL for BEM has explored the integration of domain knowledge to mitigate key limitations of purely data-driven methods. Beyond the contributions of this thesis [91, 139, 25, 92], several alternative techniques have been proposed in the literature, including graph neural networks [100], safety-constrained RL [104], and physics-informed neural networks [102]. Chapter I discussed these works in the general context of existing approaches to BEM. However, a comprehensive review of these hybrid approaches - evaluating the forms of data and knowledge representations they rely on, and mapping them to the specific challenges identified in Section 3.b - would be valuable for identifying research gaps and guiding future work in DR-oriented applications.

2.b Comparison with model-based solution approaches

The thesis primarily emphasized the practical advantages of the proposed techniques, with numerical results demonstrating their improvements over solutions of comparable deployment complexity. Nonetheless, Chapters III and IV do not investigate model-based solutions to the stated problem formulations, but instead focus on explaining why existing methods in the literature are often impractical for real-world deployment. Additionally, these approaches require specific expertise such as forecasting techniques and mathematical optimization, which are not in the scope of this thesis. A valuable direction for future work would be to benchmark the objective performance of the proposed data-driven approaches against established optimization-based baselines, in order to assess how closely such methods can approximate the performance of model-based strategies.

2.c Real-world trials

As highlighted in [126], there remains a critical need for rigorous real-world trials to validate the practical viability of the control solutions proposed in the literature. This

gap is evident across much of the advanced control research for BMS [17], largely due to the inherent challenges of conducting real-world deployments. While the work presented in this thesis seeks to address these challenges and was framed within the broader context of enhancing DR flexibility, the results are derived from simulation-based studies. Accordingly, future research should prioritize the design and execution of experiments in real buildings to assess the transferability of the simulation findings.

2.d More techniques focused on DR

The Knowledge-Informed RL paradigm introduced in this thesis offers a structured framework for situating future research in the domain and for emphasizing its relevance to the challenges previously identified. Building on this foundation, future work should aim to broaden the library of knowledge representations and techniques, while systematically demonstrating their benefits through appropriate benchmarking studies. These techniques should address other aspects of equilibrium problems in multi-stakeholders environments (such as the DR problem in Chapter IV), including fairness and remuneration schemes for system services.

Bibliography

[1] IEA, "Net Zero by 2050 – Analysis," https://www.iea.org/reports/net-zero-by-2050, May 2021.

[2] ——, "Unlocking the Potential of Distributed Energy Resources – Analysis," https://www.iea.org/reports/unlocking-the-potential-of-distributed-energy-resources, May 2022.

[3] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, M. Walczak, J. Garcke, C. Bauckhage, and J. Schuecker, "Informed Machine Learning – A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 614–633, Jan. 2023.

[4] ASHRAE, "Standard 55 – Thermal Environmental Conditions for Human Occupancy," https://www.ashrae.org/technical-resources/bookstore/standard-55-thermal-environmental-conditions-for-human-occupancy.

[5] H. Harb, N. Boyanov, L. Hernandez, R. Streblow, and D. Müller, "Development and validation of grey-box models for forecasting the thermal response of occupied buildings," *Energy and Buildings*, vol. 117, pp. 199–207, Apr. 2016.

[6] L. Battisti, "Energy, power, and greenhouse gas emissions for future transition scenarios," *Energy Policy*, vol. 179, p. 113626, Aug. 2023.

[7] "Energy Systems," in *Climate Change 2022 - Mitigation of Climate Change*, 1st ed., Intergovernmental Panel On Climate Change (Ipcc), Ed. Cambridge University Press, Aug. 2023, pp. 613–746.

[8] Ember, "Global electricity review 2025," Ember – Global Energy Think-Tank, Tech. Rep., Apr. 2025.

[9] M. Ge, J. Friedrich, and L. Vigna, "Where Do Emissions Come From? 4 Charts Explain Greenhouse Gas Emissions by Sector," 2024.

[10] M. S. Aliero, M. Asif, I. Ghani, M. F. Pasha, and S. R. Jeong, "Systematic Review Analysis on Smart Building: Challenges and Opportunities," *Sustainability*, vol. 14, no. 5, p. 3009, Jan. 2022.

[11] M. L. Nicolson, M. J. Fell, and G. M. Huebner, "Consumer demand for time of use electricity tariffs: A systematized review of the empirical evidence," *Renewable and Sustainable Energy Reviews*, vol. 97, pp. 276–289, Dec. 2018.

[12] R. Eini, L. Linkous, N. Zohrabi, and S. Abdelwahed, "Smart building management system: Performance specifications and design requirements," *Journal of Building Engineering*, vol. 39, p. 102222, Jul. 2021.

[13] A. R. Jordehi, "Optimisation of demand response in electric power systems, a review," *Renewable and Sustainable Energy Reviews*, vol. 103, pp. 308–319, Apr. 2019.

[14] M. Seraj, M. Parvez, O. Khan, and Z. Yahya, "Optimizing smart building energy management systems through industry 4.0: A response surface methodology approach," *Green Technologies and Sustainability*, vol. 2, no. 2, p. 100079, May 2024.

[15] Z. Nagy, G. Henze, S. Dey, J. Arroyo, L. Helsen, X. Zhang, B. Chen, K. Amasyali, K. Kurte, A. Zamzam, H. Zandi, J. Drgoňa, M. Quintana, S. McCullogh, J. Y. Park, H. Li, T. Hong, S. Brandi, G. Pinto, A. Capozzoli, D. Vrabie, M. Bergés, K. Nweye, T. Marzullo, and A. Bernstein, "Ten questions concerning reinforcement learning for building energy management," *Building and Environment*, vol. 241, p. 110435, Aug. 2023.

[16] J. Langevin, C. B. Harris, A. Satre-Meloy, H. Chandra-Putra, A. Speake, E. Present, R. Adhikari, E. J. Wilson, and A. J. Satchwell, "US building energy efficiency and flexibility as an electric grid resource," *Joule*, vol. 5, no. 8, pp. 2102–2128, Aug. 2021.

[17] G. P. Henze, K. J. Kircher, and J. E. Braun, "Why has advanced commercial HVAC control not yet achieved its promise?" *Journal of Building Performance Simulation*, pp. 1–12, Nov. 2024.

[18] J. Drgoňa, J. Arroyo, I. Cupeiro Figueroa, D. Blum, K. Arendt, D. Kim, E. P. Ollé, J. Oravec, M. Wetter, D. L. Vrabie, and L. Helsen, "All you need to know about model predictive control for buildings," *Annual Reviews in Control*, vol. 50, pp. 190–232, Jan. 2020.

[19] Y. Yuan, C. Song, L. Gao, K. Zeng, and Y. Chen, "A review of current research on occupant-centric control for improving comfort and energy efficiency," *Building Simulation*, vol. 17, no. 10, pp. 1675–1692, Oct. 2024.

[20] T. Broberg and L. Persson, "Is our everyday comfort for sale? Preferences for demand management on the electricity market," *Energy Economics*, vol. 54, pp. 24–32, Feb. 2016.

[21] H. Fowle, "Skilled labour shortages effects smart homes and buildings," https://www.iotinsider.com/smart-home/skilled-labour-shortages-effects-smart-homes-and-buildings/, May 2023.

[22] Inc. Dragos, "Cybersecurity for building automation systems (BAS): Securing the future," https://www.dragos.com/blog/cybersecurity-building-automation-systems/, Sep. 2023.

[23] D. Stanelyte, N. Radziukyniene, and V. Radziukynas, "Overview of Demand-Response Services: A Review," *Energies*, vol. 15, no. 5, p. 1659, Jan. 2022.

[24] S. Mohajeryami, I. N. Moghaddam, M. Doostan, B. Vatani, and P. Schwarz, "A novel economic model for price-based demand response," *Electric Power Systems Research*, vol. 135, pp. 1–9, Jun. 2016.

[25] S. R. Kumar, A. Easwaran, B. Delinchant, and R. Rigo-Mariani, "Real-time Retail Electricity Pricing Using Offline Reinforcement Learning," in *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems*. New York, NY, USA: Association for Computing Machinery, 2024, pp. 454–458.

[26] L. Toderean, T. Cioara, I. Anghel, E. Sarmas, V. Michalakopoulos, and V. Marinakis, "Demand response optimization for smart grid integrated buildings: Review of technology enablers landscape and innovation challenges," *Energy and Buildings*, vol. 326, p. 115067, Jan. 2025.

[27] G. Benetti, D. Caprino, M. L. Della Vedova, and T. Facchinetti, "Electric load management approaches for peak load reduction: A systematic literature review and state of the art," *Sustainable Cities and Society*, vol. 20, pp. 124–141, Jan. 2016.

[28] C. Goldenberg, M. Dyson, and H. Masters, "Demand flexibility: The key to enabling a low-cost, low-carbon grid," Rocky Mountain Institute, Tech. Rep., Feb. 2018.

[29] N. Fabra, D. Rapson, M. Reguant, and J. Wang, "Estimating the Elasticity to Real-Time Pricing: Evidence from the Spanish Electricity Market," *AEA Papers and Proceedings*, vol. 111, pp. 425–429, May 2021.

[30] S. Silva, I. Soares, and C. Pinho, "Electricity demand response to price changes: The Portuguese case taking into account income differences," *Energy Economics*, vol. 65, pp. 335–342, Jun. 2017.

[31] M. Kubli, M. Loock, and R. Wüstenhagen, "The flexible prosumer: Measuring the willingness to co-create distributed flexibility," *Energy Policy*, vol. 114, pp. 540–548, Mar. 2018.

[32] S. Nolan and M. O'Malley, "Challenges and barriers to demand response deployment and evaluation," *Applied Energy*, vol. 152, pp. 1–10, Aug. 2015.

[33] M. Maggiulli, M. S. Piscitelli, and A. Capozzoli, "A Data-Driven Approach to Evaluate the Smart Readiness Indicator for the Functionality "Respond to Users' Needs"," in *Sustainability in Energy and Buildings 2023*. Springer, Singapore, 2024, pp. 589–599.

[34] M. Uddin, M. F. Romlie, M. F. Abdullah, S. Abd Halim, A. H. Abu Bakar, and T. Chia Kwang, "A review on peak load shaving strategies," *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 3323–3332, Feb. 2018.

[35] P. Kohlhepp, H. Harb, H. Wolisz, S. Waczowicz, D. Müller, and V. Hagenmeyer, "Large-scale grid integration of residential thermal energy storages as demand-side flexibility resource: A review of international field studies," *Renewable and Sustainable Energy Reviews*, vol. 101, pp. 527–547, Mar. 2019.

[36] B. Parrish, P. Heptonstall, R. Gross, and B. K. Sovacool, "A systematic review of motivations, enablers and barriers for consumer engagement with residential demand response," *Energy Policy*, vol. 138, p. 111221, Mar. 2020.

[37] A. Nilsson and C. Bartusch, "Empowered or enchained? Exploring consumer perspectives on Direct Load Control," *Energy Policy*, vol. 192, Sep. 2024.

[38] S. Harper, M. Mehrnezhad, and J. Mace, "User Privacy Concerns in Commercial Smart Buildings," *Journal of Computer Security*, vol. 30, no. 3, pp. 465–497, Jan. 2022.

[39] M. Rezaeimozafar, R. F. D. Monaghan, E. Barrett, and M. Duffy, "A review of behind-the-meter energy storage systems in smart grids," *Renewable and Sustainable Energy Reviews*, vol. 164, p. 112573, Aug. 2022.

[40] X. Huo, H. Huang, K. R. Davis, H. V. Poor, and M. Liu, "A review of scalable and privacy-preserving multi-agent frameworks for distributed energy resources," *Advances in Applied Energy*, vol. 17, p. 100205, Mar. 2025.

[41] L. Sarran, H. B. Gunay, W. O'Brien, C. A. Hviid, and C. Rode, "A data-driven study of thermostat overrides during demand response events," *Energy Policy*, vol. 153, p. 112290, Jun. 2021.

[42] K. Kaspar, K. Nweye, G. Buscemi, A. Capozzoli, Z. Nagy, G. Pinto, U. Eicker, and M. M. Ouf, "Effects of occupant thermostat preferences and override behavior on residential demand response in CityLearn," *Energy and Buildings*, p. 114830, Sep. 2024.

[43] ASHRAE, "Guideline 36-2024: High-Performance Sequences of Operation for HVAC Systems," https://store.accuristech.com/ashrae/standards/guideline-36-2024-high-performance-sequences-of-operation-for-hvac-systems-includes-supporting-files?product_id=2926782&trk=public_post_reshare-text, 2024.

[44] R. S. Alshareef and H. M. Maghrabie, "Building-integrated photovoltaics with energy storage systems – A comprehensive review," *Journal of Energy Storage*, vol. 116, p. 115916, Apr. 2025.

[45] D. Azuatalam, K. Paridari, Y. Ma, M. Förstl, A. C. Chapman, and G. Verbič, "Energy management of small-scale PV-battery systems: A systematic review considering practical implementation, computational requirements, quality of input data and battery degradation," *Renewable and Sustainable Energy Reviews*, vol. 112, pp. 555–570, Sep. 2019.

[46] X. Lu, Y. Fu, S. Xu, Q. Zhu, and Z. Oâ, "Comparison Study Of High-performance Rule-based HVAC Control With Deep Reinforcement Learning-based Control In A Multi-zone VAV System," 2022.

[47] X. Lu, Y. Fu, and Z. O'Neill, "Benchmarking high performance HVAC Rule-Based controls with advanced intelligent Controllers: A case study in a Multi-Zone system in Modelica," *Energy and Buildings*, vol. 284, p. 112854, Apr. 2023.

[48] N. S. Raman, A. M. Devraj, P. Barooah, and S. P. Meyn, "Reinforcement Learning for Control of Building HVAC Systems," in *2020 American Control Conference (ACC)*, Jul. 2020, pp. 2326–2332.

[49] Y. Fu, S. Xu, Q. Zhu, Z. O'Neill, and V. Adetola, "How good are learning-based control v.s. model-based control for load shifting? Investigations on a single zone building energy system," *Energy*, vol. 273, p. 127073, Jun. 2023.

[50] D. Wang, W. Zheng, Z. Wang, Y. Wang, X. Pang, and W. Wang, "Comparison of reinforcement learning and model predictive control for building energy system optimization," *Applied Thermal Engineering*, vol. 228, p. 120430, Jun. 2023.

[51] K. Nweye, K. Kaspar, G. Buscemi, T. Fonseca, G. Pinto, D. Ghose, S. Duddukuru, P. Pratapa, H. Li, J. Mohammadi, L. L. Ferreira, T. Hong, M. Ouf, A. Capozzoli, and Z. Nagy, "CityLearn v2: Energy-flexible, resilient, occupant-centric, and carbon-aware management of grid-interactive communities," *Journal of Building Performance Simulation*, pp. 1–22, Nov. 2024.

[52] G. Pinto, M. S. Piscitelli, J. R. Vázquez-Canteli, Z. Nagy, and A. Capozzoli, "Coordinated energy management for a cluster of buildings through deep reinforcement learning," *Energy*, vol. 229, Aug. 2021.

[53] J. R. Vazquez-Canteli, G. Henze, and Z. Nagy, "MARLISA: Multi-Agent Reinforcement Learning with Iterative Sequential Action Selection for Load Shaping of Grid-Interactive Connected Buildings," in *BuildSys 2020 - Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. Association for Computing Machinery, Inc, Nov. 2020, pp. 170–179.

[54] Z. Liu, X. Zhang, Y. Sun, and Y. Zhou, "Advanced controls on energy reliability, flexibility and occupant-centric control for smart and energy-efficient buildings," *Energy and Buildings*, vol. 297, p. 113436, Oct. 2023.

[55] H. Zhang, S. Seal, D. Wu, F. Bouffard, and B. Boulet, "Building Energy Management With Reinforcement Learning and Model Predictive Control: A Survey," *IEEE Access*, vol. 10, pp. 27 853–27 862, 2022.

[56] J. Allegrini, K. Orehounig, G. Mavromatidis, F. Ruesch, V. Dorer, and R. Evins, "A review of modelling approaches and tools for the simulation of district-scale energy systems," *Renewable and Sustainable Energy Reviews*, vol. 52, pp. 1391–1404, Dec. 2015.

[57] H. Raj, S. Jaiswal, and A. Maheshwari, "A review of energy management strategy and forecasting techniques in multi microgrids," *International Journal of Ambient Energy*, vol. 46, no. 1, p. 2507808, Dec. 2025.

[58] F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and M. Morari, "Use of model predictive control and weather forecasts for energy efficient building climate control," *Energy and Buildings*, vol. 45, pp. 15–27, Feb. 2012.

[59] S. Yang, M. P. Wan, B. F. Ng, S. Dubey, G. P. Henze, W. Chen, and K. Baskaran, "Experimental study of model predictive control for an air-conditioning system with dedicated outdoor air system," *Applied Energy*, vol. 257, p. 113920, Jan. 2020.

[60] J. Drgoňa, D. Picard, and L. Helsen, "Cloud-based implementation of white-box model predictive control for a GEOTABS office building: A field test demonstration," *Journal of Process Control*, vol. 88, pp. 63–77, Apr. 2020.

[61] A. Nottrott, J. Kleissl, and B. Washom, "Energy dispatch schedule optimization and cost benefit analysis for grid-connected, photovoltaic-battery storage systems," *Renewable Energy*, vol. 55, pp. 230–240, Jul. 2013.

[62] J. Cai, H. Zhang, and X. Jin, "Aging-aware predictive control of PV-battery assets in buildings," *Applied Energy*, vol. 236, pp. 478–488, Feb. 2019.

[63] C. R. Touretzky and M. Baldea, "Integrating scheduling and control for economic MPC of buildings with energy storage," *Journal of Process Control*, vol. 24, no. 8, pp. 1292–1300, Aug. 2014.

[64] C. Qian, N. He, Z. Cheng, R. Li, and L. Yang, "Optimal scheduling of building energy system with integrated virtual energy storage based on multi-task model predictive control," *Journal of Building Engineering*, vol. 98, p. 111185, Dec. 2024.

[65] D. T. Vedullapalli, R. Hadidi, and B. Schroeder, "Combined HVAC and Battery Scheduling for Demand Response in a Building," *IEEE Transactions on Industry Applications*, vol. 55, no. 6, pp. 7008–7014, Nov. 2019.

[66] L. Chamari, S. Walker, E. Petrova, and P. Pauwels, "Towards portable model predictive control-based applications for demand side management in buildings," *Energy and Buildings*, vol. 347, p. 116257, Nov. 2025.

[67] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[68] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric Actor Critic for Image-Based Robot Learning," Oct. 2017.

[69] A. Likmeta, A. M. Metelli, A. Tirinzoni, R. Giol, M. Restelli, and D. Romano, "Combining reinforcement learning with rule-based controllers for transparent and general decision-making in autonomous driving," *Robotics and Autonomous Systems*, vol. 131, p. 103568, Sep. 2020.

[70] Google, "DeepMind AI Reduces Google Data Centre Cooling Bill by 40%," https://deepmind.google/discover/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-by-40/, Jul. 2016.

[71] T. Wei, Y. Wang, and Q. Zhu, "Deep reinforcement learning for building HVAC control," in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, Jun. 2017, pp. 1–6.

[72] F. Ruelens, B. J. Claessens, S. Vandael, B. De Schutter, R. Babuska, and R. Belmans, "Residential Demand Response of Thermostatically Controlled Loads Using Batch Reinforcement Learning," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2149–2159, Sep. 2017.

[73] X. Kou, Y. Du, F. Li, H. Pulgar-Painemal, H. Zandi, J. Dong, and M. M. Olama, "Model-based and data-driven HVAC control strategies for residential demand response," *IEEE Open Access Journal of Power and Energy*, vol. 8, pp. 186–197, 2021.

[74] G. Pinto, D. Deltetto, and A. Capozzoli, "Data-driven district energy management with surrogate models and deep reinforcement learning," *Applied Energy*, vol. 304, Dec. 2021.

[75] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," Aug. 2018.

[76] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," Jul. 2019.

[77] K. Nweye, Z. Nagy, B. Liu, and P. Stone, "Offline training of multi-agent reinforcement agents for grid-interactive buildings control," in *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, ser. E-Energy '22. New York, NY, USA: Association for Computing Machinery, Jun. 2022, pp. 442–443.

[78] G. Palma, L. Guiducci, M. Stentati, A. Rizzo, and S. Paoletti, "Reinforcement Learning for Energy Community Management: A European-Scale Study," *Energies*, vol. 17, no. 5, p. 1249, Mar. 2024.

[79] V. Khattar and M. Jin, "Winning the CityLearn Challenge: Adaptive Optimization with Evolutionary Search under Trajectory-based Guidance," Dec. 2022.

[80] A. J. Khabbazi, E. N. Pergantis, L. D. Reyes Premer, P. Papageorgiou, A. H. Lee, J. E. Braun, G. P. Henze, and K. J. Kircher, "Lessons learned from field demonstrations of model predictive control and reinforcement learning for residential and commercial HVAC: A review," *Applied Energy*, vol. 399, p. 126459, Dec. 2025.

[81] M. Killian and M. Kozek, "Ten questions concerning model predictive control for energy efficient buildings," *Building and Environment*, vol. 105, pp. 403–412, Aug. 2016.

[82] Y. Du, H. Zandi, O. Kotevska, K. Kurte, J. Munk, K. Amasyali, E. Mckee, and F. Li, "Intelligent multi-zone residential HVAC control strategy based on deep reinforcement learning," *Applied Energy*, vol. 281, p. 116117, Jan. 2021.

[83] O. Erdinc, A. Tascikaraoglu, N. G. Paterakis, Y. Eren, and J. Catalao, "End-user comfort oriented day-ahead planning for responsive residential HVAC demand aggregation considering weather forecasts," in *2017 IEEE Power & Energy Society General Meeting*, Jul. 2017, pp. 1–1.

[84] G. Lankeshwara, R. Sharma, R. Yan, and T. K. Saha, "A hierarchical control scheme for residential air-conditioning loads to provide real-time market services under uncertainties," *Energy*, vol. 250, p. 123796, Jul. 2022.

[85] N. Lefebure, M. Khosravi, M. Hudoba de Badyn, F. Bünning, J. Lygeros, C. Jones, and R. S. Smith, "Distributed model predictive control of buildings and energy hubs," *Energy and Buildings*, vol. 259, p. 111806, Mar. 2022.

[86] Y. Xia, X. Wang, X. Yin, W. Bo, L. Wang, S. Li, and K. Li, "Federated Accelerated Deep Reinforcement Learning for Multi-Zone HVAC Control in Commercial Buildings," *IEEE Transactions on Smart Grid*, vol. 16, no. 3, pp. 2599–2610, May 2025.

[87] W. Xue, N. Jia, and M. Zhao, "Multi-agent deep reinforcement learning based HVAC control for multi-zone buildings considering zone-energy-allocation optimization," *Energy and Buildings*, vol. 329, p. 115241, Feb. 2025.

[88] V. Mai, P. Maisonneuve, T. Zhang, H. Nekoei, L. Paull, and A. Lesage-Landry, "Multi-agent reinforcement learning for fast-timescale demand response of residential loads," *Machine Learning*, vol. 113, no. 8, pp. 5203–5234, Aug. 2024.

[89] B. Celik, R. Roche, S. Suryanarayanan, D. Bouquain, and A. Miraoui, "Electric energy management in residential areas through coordination of multiple smart homes," *Renewable and Sustainable Energy Reviews*, vol. 80, pp. 260–275, 2017.

[90] T. Xie, N. Jiang, H. Wang, C. Xiong, and Y. Bai, "Policy Finetuning: Bridging Sample-Efficient Offline and Online Reinforcement Learning," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 27 395–27 407.

[91] S. R. Kumar, A. Easwaran, B. Delinchant, and R. Rigo-Mariani, "Behavioural cloning based RL agents for district energy management," in *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. New York, NY, USA: ACM, Nov. 2022, pp. 466–470.

[92] S. Ram Kumar, A. Easwaran, B. Delinchant, and R. Rigo-Mariani, "Improving Demand Response Programs Using Override Signals with Reinforcement Learning," in *Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems*, ser. E-Energy '25. New York, NY, USA: Association for Computing Machinery, Jun. 2025, pp. 603–611.

[93] G. Gokhale, J. Van Gompel, B. Claessens, and C. Develder, "Transfer Learning in Transformer-Based Demand Forecasting For Home Energy Management System," in *Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. New York, NY, USA: ACM, Nov. 2023, pp. 458–462.

[94] C. Blad, S. Bøgh, and C. S. Kallesøe, "Data-driven Offline Reinforcement Learning for HVAC-systems," *Energy*, vol. 261, Dec. 2022.

[95] S. Dey, T. Marzullo, and G. Henze, "Inverse reinforcement learning control for building energy management," *Energy and Buildings*, vol. 286, p. 112941, May 2023.

[96] H.-Y. Liu, B. Balaji, S. Gao, R. Gupta, and D. Hong, "Safe HVAC Control via Batch Reinforcement Learning," in *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*, May 2022, pp. 181–192.

[97] S. Yang, M. P. Wan, W. Chen, B. F. Ng, and S. Dubey, "Model predictive control with adaptive machine-learning-based model for building energy efficiency and comfort optimization," *Applied Energy*, vol. 271, p. 115147, Aug. 2020.

[98] M. Liu, M. Guo, Y. Fu, Z. O'Neill, and Y. Gao, "Expert-guided imitation learning for energy management: Evaluating GAIL's performance in building control applications," *Applied Energy*, vol. 372, Oct. 2024.

[99] J. Goldfeder and J. Sipple, "A Lightweight Calibrated Simulation Enabling Efficient Offline Learning for Optimal Control of Real Buildings," Oct. 2023.

[100] S. Nagarathinam and A. Vasan, "PhyGICS – A Physics-informed Graph Neural Network-based Intelligent HVAC Controller for Open-plan Spaces," in *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems*, ser. E-Energy '24. New York, NY, USA: Association for Computing Machinery, May 2024, pp. 203–214.

[101] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa, "Safe Exploration in Continuous Action Spaces," Jan. 2018.

[102] R. Wang, Z. Cao, X. Zhou, Y. Wen, and R. Tan, "Green Data Center Cooling Control via Physics-guided Safe Reinforcement Learning," *ACM Trans. Cyber-Phys. Syst.*, vol. 8, no. 2, pp. 19:1–19:26, May 2024.

[103] X. Wang, P. Wang, R. Huang, X. Zhu, J. Arroyo, and N. Li, "Safe deep reinforcement learning for building energy management," *Applied Energy*, vol. 377, p. 124328, Jan. 2025.

[104] G. Ceusters, L. R. Camargo, R. Franke, A. Nowé, and M. Messagie, "Safe reinforcement learning for multi-energy management systems with known constraint functions," *Energy and AI*, vol. 12, Apr. 2023.

[105] L. Von Krannichfeldt, K. Orehounig, and O. Fink, "Combining physics-based and data-driven modeling for building energy systems," *Applied Energy*, vol. 391, p. 125853, Aug. 2025.

[106] Y. Wu, B. Sicard, and S. A. Gadsden, "Physics-informed machine learning: A comprehensive review on applications in anomaly detection and condition monitoring," *Expert Systems with Applications*, vol. 255, p. 124678, Dec. 2024.

[107] R. S. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, second edition ed., ser. Adaptive Computation and Machine Learning. Cambridge, Massachusetts London, England: The MIT Press, 2020.

[108] Y. Li, Z. O'Neill, L. Zhang, J. Chen, P. Im, and J. DeGraw, "Grey-box modeling and application for building energy simulations - A critical review," *Renewable and Sustainable Energy Reviews*, vol. 146, p. 111174, Aug. 2021.

[109] J. Joe, P. Im, B. Cui, and J. Dong, "Model-based predictive control of multi-zone commercial building with a lumped building modelling approach," *Energy*, vol. 263, p. 125494, Jan. 2023.

[110] A. Talib, S. Park, P. Im, and J. Joe, "Grey-box and ANN-based building models for multistep-ahead prediction of indoor temperature to implement model predictive control," *Engineering Applications of Artificial Intelligence*, vol. 126, p. 107115, Nov. 2023.

[111] F. Bünning, B. Huber, P. Heer, A. Aboudonia, and J. Lygeros, "Experimental demonstration of data predictive control for energy optimization and thermal comfort in buildings," *Energy and Buildings*, vol. 211, Mar. 2020.

[112] Y. Sun, F. Haghighat, and B. C. M. Fung, "Fairness-aware data-driven-based model predictive controller: A study on thermal energy storage in a residential building," *Journal of Energy Storage*, vol. 87, p. 111402, May 2024.

[113] J. Arroyo, C. Manna, F. Spiessens, and L. Helsen, "Reinforced model predictive control (RL-MPC) for building energy management," *Applied Energy*, vol. 309, Mar. 2022.

[114] B. Chen, Z. Cai, and M. Bergés, "Gnu-RL: A precocial reinforcement learning solution for building HVAC control using a differentiable MPC policy," in *BuildSys 2019 - Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. Association for Computing Machinery, Inc, Nov. 2019, pp. 316–325.

[115] B. Amos, I. D. J. Rodriguez, J. Sacks, B. Boots, and J. Z. Kolter, "Differentiable MPC for End-to-end Planning and Control," Oct. 2018.

[116] J. R. Vázquez-Canteli and Z. Nagy, "Reinforcement learning for demand response: A review of algorithms and modeling techniques," *Applied Energy*, vol. 235, pp. 1072–1089, Feb. 2019.

[117] D. Weinberg, Q. Wang, T. O. Timoudas, and C. Fischione, "A Review of Reinforcement Learning for Controlling Building Energy Systems From a Computer Science Perspective," *Sustainable Cities and Society*, vol. 89, p. 104351, Feb. 2023.

[118] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman, "Measures for explainable AI: Explanation goodness, user satisfaction, mental models, curiosity, trust, and human-AI performance," *Frontiers in Computer Science*, vol. 5, Feb. 2023.

[119] J. Eßer, N. Bach, C. Jestel, O. Urbann, and S. Kerner, "Guided Reinforcement Learning: A Review and Evaluation for Efficient and Effective Real-World Robotics [Survey]," *IEEE Robotics & Automation Magazine*, vol. 30, no. 2, pp. 67–85, Jun. 2023.

[120] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, Jun. 2021.

[121] Z. Qiang, S. Hands, K. Taylor, S. Sethuvenkatraman, D. Hugo, P. Ghiasnezhad Omran, M. Perera, and A. Haller, "A systematic comparison and evaluation of building ontologies for deploying data-driven analytics in smart buildings," *Energy and Buildings*, vol. 292, p. 113054, Aug. 2023.

[122] Y. Luo, J. Kay, E. Grefenstette, and M. P. Deisenroth, "Finetuning from Offline Reinforcement Learning: Challenges, Trade-offs and Practical Solutions," Mar. 2023.

[123] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems," Nov. 2020.

[124] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: A Survey," Sep. 2020.

[125] F. Bongratz, V. Golkov, L. Mautner, L. D. Libera, F. Heetmeyer, F. Czaja, J. Rodemann, and D. Cremers, "How to Choose a Reinforcement-Learning Algorithm," Jul. 2024.

[126] K. Nweye, B. Liu, P. Stone, and Z. Nagy, "Real-world challenges for multi-agent reinforcement learning in grid-interactive buildings," *Energy and AI*, vol. 10, p. 100202, Nov. 2022.

[127] D. Vos and S. Verwer, "Optimizing Interpretable Decision Tree Policies for Reinforcement Learning," Aug. 2024.

[128] S. Ibrahim, M. Mostafa, A. Jnadi, H. Salloum, and P. Osinenko, "Comprehensive Overview of Reward Engineering and Shaping in Advancing Reinforcement Learning Applications," *IEEE Access*, vol. 12, pp. 175 473–175 500, 2024.

[129] L. Di Natale, B. Svetozarevic, P. Heer, and C. N. Jones, "Physically Consistent Neural Networks for building thermal modeling: Theory and analysis," Dec. 2021.

[130] D. Enescu, "A review of thermal comfort models and indicators for indoor environments," *Renewable and Sustainable Energy Reviews*, vol. 79, pp. 1353–1379, Nov. 2017.

[131] "EnergyPlus™," National Renewable Energy Laboratory (NREL), Golden, CO (United States); Lawrence Berkeley National Laboratory (LBNL), Berkeley, CA (United States), Tech. Rep. EnergyPlus™ (e+); 005462MLTPL00, Sep. 2017.

[132] Z. Wang and Y. Chen, "Data-driven modeling of building thermal dynamics: Methodology and state of the art," *Energy and Buildings*, vol. 203, p. 109405, Nov. 2019.

[133] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 23–30.

[134] X. Lin, Z. Tian, W. Song, Y. Lu, J. Niu, Q. Sun, and Y. Wang, "Grey-box modeling for thermal dynamics of buildings under the presence of unmeasured internal heat gains," *Energy and Buildings*, vol. 314, p. 114229, Jul. 2024.

[135] X. Zhan, X. Zhu, P. Cheng, X. Hu, Z. He, H. Geng, J. Leng, H. Zheng, C. Liu, T. Hong, Y. Liang, Y. Liu, and F. Zhao, "Data Center Cooling System Optimization Using Offline Reinforcement Learning," in *The Thirteenth International Conference on Learning Representations*, Oct. 2024.

[136] G. Lambrechts, D. Ernst, and A. Mahajan, "A Theoretical Justification for Asymmetric Actor-Critic Algorithms," in *Forty-Second International Conference on Machine Learning*, Jun. 2025.

[137] S. Huang and S. Ontañón, "A Closer Look at Invalid Action Masking in Policy Gradient Algorithms," Jun. 2020.

[138] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-End Safe Reinforcement Learning through Barrier Functions for Safety-Critical Continuous Control Tasks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 3387–3395, Jul. 2019.

[139] S. R. Kumar, R. Rigo-Mariani, B. Delinchant, and A. Easwaran, "Towards Safe Model-Free Building Energy Management using Masked Reinforcement Learning," in *2023 IEEE PES Innovative Smart Grid Technologies Europe (ISGT EUROPE)*. IEEE, Oct. 2023, pp. 1–5.

[140] C. Banerjee, K. Nguyen, C. Fookes, and M. Raissi, "A Survey on Physics Informed Reinforcement Learning: Review and Open Problems," 2023.

[141] A. Siddiqui and M. Asim, "Advancement, challenges and solutions of PV integrated battery energy storage systems: A review," *Journal of Energy Storage*, vol. 132, p. 117996, Oct. 2025.

[142] P. Fanger, *Thermal Comfort: Analysis and Applications in Environmental Engineering*. Danish Technical Press, 1970.

[143] D. Jang, L. Spangher, S. Nadarajah, and C. Spanos, "Deep reinforcement learning with planning guardrails for building energy demand response," *Energy and AI*, vol. 11, p. 100204, Jan. 2023.

[144] Movicom, "BMS Main X 2.x - Configuration manual (ENG).pdf," https://movicomelectric.com/wa-data/public/site/doc/BMS%20Main%20X%202.x%20-%20Configuration%20manual%20(ENG).pdf#page=27.60.

[145] C. Gao and D. Wang, "Comparative study of model-based and model-free reinforcement learning control performance in HVAC systems," *Journal of Building Engineering*, vol. 74, p. 106852, Sep. 2023.

[146] A. Bukharin, Y. Li, P. He, and T. Zhao, "Deep Reinforcement Learning from Hierarchical Preference Design," Jun. 2024.

[147] M. Nakamoto, Y. Zhai, A. Singh, M. S. Mark, Y. Ma, C. Finn, A. Kumar, and S. Levine, "Cal-QL: Calibrated Offline RL Pre-Training for Efficient Online Fine-Tuning," in *Thirty-Seventh Conference on Neural Information Processing Systems*, Nov. 2023.

[148] Z. Song, R. E. Parr, and L. Carin, "Revisiting the Softmax Bellman Operator: New Benefits and New Perspective," May 2019.

[149] A. Smith, B. Stürmer, T. Thurber, and C. Vernon, "Diyepw: A Python package for Do-It-Yourself EnergyPlus weather file generation," *Journal of Open Source Software*, vol. 6, no. 64, p. 3313, Aug. 2021.

[150] T. Seno and M. Imai, "D3rlpy: An Offline Deep Reinforcement Learning Library," Dec. 2022.

[151] M. Andrychowicz, A. Raichuk, P. Stańczyk, M. Orsini, S. Girgin, R. Marinier, L. Hussenot, M. Geist, O. Pietquin, M. Michalski, S. Gelly, and O. Bachem, "What Matters In On-Policy Reinforcement Learning? A Large-Scale Empirical Study," Jun. 2020.

[152] J. Ryu, J. Kim, W. Hong, and R. De Dear, "Quantifying householder tolerance of thermal discomfort before turning on air-conditioner," *Energy and Buildings*, vol. 211, p. 109797, Mar. 2020.

[153] Y. Jian, J. Liu, Z. Pei, and J. Chen, "Occupants' tolerance of thermal discomfort before turning on air conditioning in summer and the effects of age and gender," *Journal of Building Engineering*, vol. 50, p. 104099, Jun. 2022.

[154] N. Pflugradt, P. Stenzel, L. Kotzur, and D. Stolten, "LoadProfileGenerator: An Agent-Based BehaviorSimulation for Generating Residential Load Profiles," *Journal of Open Source Software*, vol. 7, no. 71, p. 3574, Mar. 2022.

[155] G. Hoogsteen, A. Molderink, J. L. Hurink, Gerard. J. Smit, F. Schuring, and B. K. Liandon, "Impact of peak electricity demand in distribution grids: A stress test," in *2015 IEEE Eindhoven PowerTech*, Jun. 2015, pp. 1–6.

[156] M. Liang, Y. Meng, N. Lu, D. Lubkeman, and A. Kling, "HVAC load Disaggregation using Low-resolution Smart Meter Data," in *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. Washington, DC, USA: IEEE, Feb. 2019, pp. 1–5.

[157] H. Liang and J. Ma, "Separation of Residential Space Cooling Usage From Smart Meter Data," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3107–3118, Jul. 2020.

[158] N. Qi, L. Cheng, H. Xu, K. Wu, X. Li, Y. Wang, and R. Liu, "Smart meter data-driven evaluation of operational demand response potential of residential air conditioning loads," *Applied Energy*, vol. 279, p. 115708, Dec. 2020.

[159] H. Cimen, N. Cetinkaya, J. C. Vasquez, and J. M. Guerrero, "A Microgrid Energy Management System Based on Non-Intrusive Load Monitoring via Multitask Learning," *IEEE Transactions on Smart Grid*, vol. 12, no. 2, pp. 977–987, Mar. 2021.

[160] B. Rajasekhar, W. Tushar, C. Lork, Y. Zhou, C. Yuen, N. M. Pindoriya, and K. L. Wood, "A Survey of Computational Intelligence Techniques for Air-Conditioners Energy Management," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 4, pp. 555–570, Aug. 2020.

[161] N. Lei, H. Zhang, R. Li, J. Yu, H. Wang, and Z. Wang, "Physics-informed data-driven modeling approach for commuting-oriented hybrid powertrain optimization," *Energy Conversion and Management*, vol. 299, p. 117814, Jan. 2024.

[162] F. Charbonnier, T. Morstyn, and M. D. McCulloch, "Scalable multi-agent reinforcement learning for distributed control of residential energy flexibility," *Applied Energy*, vol. 314, May 2022.

[163] A. Melillo, R. Durrer, J. Worlitschek, and P. Schütz, "First results of remote building characterisation based on smart meter measurement data," *Energy*, vol. 200, p. 117525, Jun. 2020.

[164] P. Gianniou, C. Reinhart, D. Hsu, A. Heller, and C. Rode, "Estimation of temperature setpoints and heat transfer coefficients among residential buildings in Denmark based on smart meter data," *Building and Environment*, vol. 139, pp. 125–133, Jul. 2018.

[165] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[166] W. J. Cole, J. D. Rhodes, W. Gorman, K. X. Perez, M. E. Webber, and T. F. Edgar, "Community-scale residential air conditioning control for effective grid management," *Applied Energy*, vol. 130, pp. 428–436, Oct. 2014.

[167] H. Fan, I. F. MacGill, and A. B. Sproul, "Statistical analysis of drivers of residential peak electricity demand," *Energy and Buildings*, vol. 141, pp. 205–217, Apr. 2017.

[168] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," Aug. 2017.

[169] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[170] E. Van Orden and U. Simović, "Managing Peak Load in the Past, Present, and Future: Challenges and Opportunities for a Flexible Energy Future," *Climate and Energy*, vol. 39, no. 12, pp. 15–20, 2023.

[171] Z. Khorasani Zadeh, M. Ouf, B. Gunay, B. Delcroix, G. Larochelle Martin, and A. Daoud, "Development of prediction models for thermostat override behavior in direct load control events," *Energy and Buildings*, vol. 301, p. 113707, Dec. 2023.

[172] H. Kazmi, J. Suykens, A. Balint, and J. Driesen, "Multi-agent reinforcement learning for modeling and control of thermostatically controlled loads," *Applied Energy*, vol. 238, pp. 1022–1035, Mar. 2019.

[173] N. Lei, H. Zhang, J. Hu, Z. Hu, and Z. Wang, "Sim-to-real design and development of reinforcement learning-based energy management strategies for fuel cell electric vehicles," *Applied Energy*, vol. 393, p. 126030, Sep. 2025.

[174] H. Zhang, N. Lei, B. Chen, B. Li, R. Li, and Z. Wang, "Modeling and control system optimization for electrified vehicles: A data-driven approach," *Energy*, vol. 310, p. 133196, Nov. 2024.

[175] N. Batra, O. Parson, M. Berges, A. Singh, and A. Rogers, "A comparison of non-intrusive load monitoring methods for commercial and residential buildings," Aug. 2014.

[176] D. A. Pomerleau, "What's hidden in the hidden layers."

[177] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[178] H. Tian, A. Olshevsky, and I. Paschalidis, "Convergence of Actor-Critic with Multi-Layer Neural Networks," in *Thirty-Seventh Conference on Neural Information Processing Systems*, Nov. 2023.

[179] N. A. Lynnerup, L. Nolling, R. Hasle, and J. Hallam, "A Survey on Reproducibility by Evaluating Deep Reinforcement Learning Algorithms on Real-World Robots," in *Proceedings of the Conference on Robot Learning*. PMLR, May 2020, pp. 466–489.

[180] N. Lei, H. Zhang, H. Wang, Z. Hu, H. Chen, J. Hu, and Z. Wang, "Theory-Constrained Neural Network with Modular Interpretability for Fuel Cell Vehicle Modelling," *IEEE Transactions on Vehicular Technology*, pp. 1–13, 2025.

[181] S. Osonuga, S. Shahid, A. Chouman, F. Wurtz, and B. Delinchant, "PREDIS-MHI Thermal Data," Mar. 2025.

[182] IEA, "Electricity Market Report 2023 – Analysis," https://www.iea.org/reports/electricity-market-report-2023, Feb. 2023.

[183] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3, pp. 229–256, May 1992.

[184] K. H. Yang and C. H. Su, "An approach to building energy savings using the PMV index," *Building and Environment*, vol. 32, no. 1, pp. 25–30, Jan. 1997.

[185] ISO, "ISO 7730:2005," https://www.iso.org/standard/39155.html.

[186] R. Yao, S. Zhang, C. Du, M. Schweiker, S. Hodder, B. W. Olesen, J. Toftum, F. Romana d'Ambrosio, H. Gebhardt, S. Zhou, F. Yuan, and B. Li, "Evolution and performance analysis of adaptive thermal comfort models – A comprehensive literature review," *Building and Environment*, vol. 217, p. 109020, Jun. 2022.

[187] C. John, C. Vallianos, J. Candanedo, and A. Athienitis, "Estimating time constants for over 10,000 residential buildings in North America: Towards a statistical characterization of thermal dynamics," in *Healthy, Intelligent and Resilient Buildings and Urban Environments*. International Association of Building Physics (IABP), 2018, pp. 1383–1388.

Appendices

Appendix A

Technical Background

A.1 Deep Reinforcement Learning

Reinforcement learning is a machine learning paradigm where an agent learns to make decisions by interacting with an environment [107]. The agent receives feedback in the form of rewards based on the actions it takes, and the goal is to maximize the cumulative reward over time. DRL extends this concept by leveraging deep neural networks to approximate complex functions, allowing RL to be applied to high-dimensional state and action spaces [67].

RL problems are typically formalized as MDPs, which are characterized by a set of states S, actions A, a reward function $R : S \times A \times S \to \mathbb{R}$, a transition probability function $\tau : S \times S \times A \to [0, 1]$ and a discount factor $\gamma \in (0, 1)$. MDPs are episodic sequential processes divided into a set of discrete steps. At each step t, an agent observes the state of the system $s_t \in S$ and takes an action $a_t \in A$. Upon applying a_t, the system transitions to a new state $s_{t+1} \sim \tau(\cdot|s_t, a_t)$, and the agent obtains an instantaneous reward $r_t \sim R(s_t, a_t, s_{t+1})$. The goal of RL is to find the policy $\pi : S \to A$ that maximizes the expected cumulative sum of rewards G_t, as shown in Eq A.1.

$$G_t = (\sum_{k=0}^{\infty} \gamma^k r_{t+k}) \tag{A.1}$$

There are broadly two types of RL algorithms - value-based and policy-based. In value-based methods, the focus is on learning a value function $V_\pi : S \to \mathbb{R}$ (or an action-value function $q_\pi : S \times A \to \mathbb{R}$) which estimates the expected return from a given state $s \in S$ when following a policy π. These are defined as shown in Eqs A.2 and A.3 respectively.

$$V_\pi(s) = \mathbb{E}_\pi[G_t|s_t = s] \tag{A.2}$$

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t|s_t = s, a_t = a] \tag{A.3}$$

By interacting with the environment, empirical measures of these quantities are obtained, using which the estimated functions are updated. The policy is then derived from the value function, commonly using an ϵ-greedy approach[1] to trade-off between exploration and exploitation. After many iterations, the policy so derived converges to the optimal policy.

[1] That is, with probability $\epsilon < 1.0$, a random action is selected to encourage exploration; otherwise, the action with the highest estimated value is chosen (greedy action).

On the other hand, policy-based methods directly optimize the policy π by adjusting its parameters θ to maximize the expected return based on the gradient (ie, $\nabla_\theta(\mathbb{E}_\pi[G_t])$). Several algorithms exist which describe the calculation of this gradient, under the assumption that the policy gradient $\nabla_\theta(\pi)$ can be calculated. The parameters are iteratively updated through gradient ascent, as described in Eq A.4 (where α is a positive constant representing the learning rate).

$$\theta_{k+1} = \theta_k + \alpha\nabla_\theta(\mathbb{E}_\pi[G_t])|\theta_k \tag{A.4}$$

The key idea in DRL is to represent one or both of the policy and the value functions as deep neural networks. The parameters of these networks can then be updated using algorithms such as Deep-Q Learning [67] or REINFORCE [183], generally using deep learning libraries to automatically calculate the necessary gradients. In recent years, DRL has been used to solve complex control problems in robotics, video games and energy systems.

A.1.a Actor-Critic Reinforcement Learning

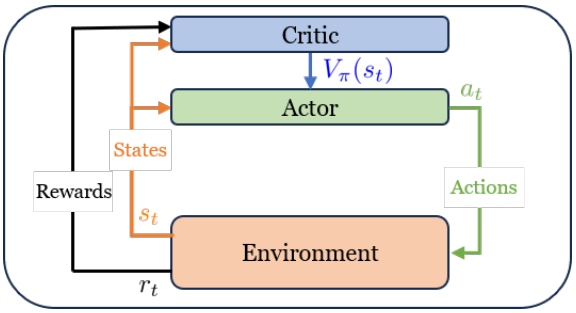

Figure A.1: Agent-Environment Interaction in RL

Actor-critic RL architectures are a class of algorithms which combine both policy-based and value-based learning. There are two components - an actor, which represents the policy π, and a critic, which represents the value function V_π or q_π, as show in Fig A.1. The critic provides feedback to the actor by calculating the Temporal Difference (TD) error δ_t (Eq A.5), which is used to update the policy parameters. The parameters of the critic (ϕ) are generally updated in a supervised manner based on the error between the estimated and observed values of $V_\pi(s_t)$.

$$\delta_t = r_t + \gamma V_\pi(s_{t+1}) - V_\pi(s_t) \tag{A.5}$$

The actor update subsequently incorporates the TD error as shown in Eq A.6. The main idea is that a high value of δ_t represents a state whose value is underestimated; as such, the actor is encouraged to take more actions that would push the system towards this state.

$$\theta_{k+1} = \theta_k + \alpha\delta_t\nabla_\theta(\mathbb{E}_\pi[G_t])|\theta_k \tag{A.6}$$

Actor-critic methods are particularly useful in environments with continuous or high-dimensional action spaces, where purely value-based methods like Q-learning become impractical. Unlike policy gradient methods that suffer from high variance, the critic reduces this variance by providing a more informed update direction. This makes actor-critic architectures especially effective in complex control problems - such as building

energy management - where policies must be fine-tuned based on nuanced state information and where stability during learning is critical.

A.2 Measures of Indoor Thermal Comfort

Indoor thermal comfort is a central consideration in building design and environmental control, aiming to maintain conditions where occupants feel neither too hot nor too cold. Accurately assessing comfort is essential for creating healthy, productive indoor environments and for optimizing building energy use. Thermal comfort is inherently subjective, depending on environmental and personal factors such as air temperature, humidity, air velocity, radiant heat, clothing insulation, and metabolic rate [130].

Many of these variables are difficult to measure directly, so mean indoor temperature is often used as a proxy, with assumed values (e.g., 24°C) considered comfortable for most occupants. However, studies have shown that such approximations can lead to discomfort and unnecessary energy consumption [184]. A widely used metric for evaluating comfort holistically is the PMV, proposed by Fanger in 1970 based on empirical studies and controlled experiments [142]. The PMV predicts a comfort score from -3 (too cold) to +3 (too hot), based on the metabolic rate M (W/m^2), mechanical work done W (W/m^2) and the heat loss from the body to the environment, H (W/m^2):

$$\text{PMV} = (0.303 \times e^{-0.036M} + 0.028) \times (M - W - H) \tag{A.7}$$

The calculation of M, W, and H incorporates all the factors listed above, and is described in detail in [142]. PMV is the recommended metric for indoor comfort in international standards, including International Standards Organization (ISO) 7730 [185] and ASHRAE Standard 55 [4], which suggest maintaining a PMV between -0.5 and 0.5 for comfortable conditions. Several alternatives to PMV have also been proposed in the literature to model other effects, and address its limitations. For example, adaptive comfort models [186] emphasize the role of occupant adaptation to changing outdoor climates, and are used in aforementioned standards for naturally ventilated buildings.

Appendix B

Appendix to Chapter 4

B.1 Generation of Appliance Load Dataset

The appliance loads are pre-calculated using the synthetic load generation software 'LoadProfileGenerator' [154]. The tool performs bottom-up simulations of household activities based on a desire-driven behavioural simulation using a psychological model. It is capable of generating detailed time-series data based on inputs such as household composition, occupants, appliances and type of day. The load profiles in this study were generated using the templates provided in the tool, which are reflective of residents in Germany (it is assumed that residential appliance load profiles are similar in tropical countries such as Singapore). The synthetic profiles cover a period of 1 year at a resolution of 15 minutes, for a group of 100 households. An overview of the composition of the dataset is given in Table B.1.

Household Template	Count	Load Statistics (kW)			Energy Consumption Distribution (%)		
		Avg	Std	Peak	12 AM - 8 AM	8 AM - 4 PM	4 PM - 12 AM
CHR05 2 Working Parents, 3 Children	5	0.71	1.08	9.58	17	34	49
CHR27 2 Working Parents, 2 Children	10	0.57	0.95	12.68	15	32	53
CHR03 2 Working Parents, 1 Child	30	0.54	0.84	8.94	19	30	51
CHR15 2 Working Parents, 2 Children, 2 Seniors	9	1.22	1.42	13.08	16	50	33
CHR01 2 Working Parents, No Children	10	0.49	0.74	12.31	21	33	46
CHR02 2 Working Parents, No Children	11	0.35	0.66	8.25	20	36	44
CHR22 1 Working Parent, 1 Child	7	0.21	0.35	5.52	25	33	42
CHR06 1 Unemployed Adult	4	0.28	0.35	7.57	23	41	35
CHR29 1 Working Adult	7	0.22	0.4	5.65	32	25	43
CHR31 1 Retired Adult	7	0.29	0.39	4.88	31	31	37
Total	100	51.24	40.00	264.23	19	36	45

Table B.1: Composition of simulated households in LoadProfileGenerator

The distribution of household types is representative of the population statistics released by the Singapore Department of Statistics[1]. It should be noted that the loads within a given household template are not identical due to the stochastic nature of the simulation; instead, they may share consumption patterns.

B.2 Calculation of Cooling Loads in the Reference System

The cooling loads of a consumer are calculated through a combination of a thermal model of the building, an air conditioner model and a thermal comfort model. The AC is

[1] https://tablebuilder.singstat.gov.sg/table/TS/M810651

modelled as a thermostatically controlled device with a single speed compressor, that operates as described in Section 1.a-ii. The thermal comfort of the consumer is represented using a discomfort capacity model as explained in Section **??**. The maximum discomfort values were randomly set between 0.025^oCh and 0.65^oCh, with a mean and standard deviation of 0.24 ± 0.14^oCh. This matches the values found empirically in [152] for a group of residents in Daegu, Korea. The preferred indoor temperatures were sampled from a normal distribution, ie, $T_n^{in,ideal} \sim \mathcal{N}(24, 2)^oC$ which is reflective of typical preferences in tropical climates.

Since the study requires the dynamic simulation of the temperature evolution of a large group of buildings, existing tools such as EnergyPlus are not feasible due to time and resource constraints. As such, a 4R2C thermal equivalent circuit is used to calculate the heat dynamics of the building, with the structure shown in Fig B.1. The model and its implementation follow the explanation in [5], who evaluated different RC thermal models based on their ability to accurately capture the dynamics of a real building. Once the model parameters and $T_n^{in,ideal}$ are fixed, $P_n^{ac,max}$ is set sufficiently high to meet the cooling demand on the hottest day (highest average T^{amb}) in the weather data.

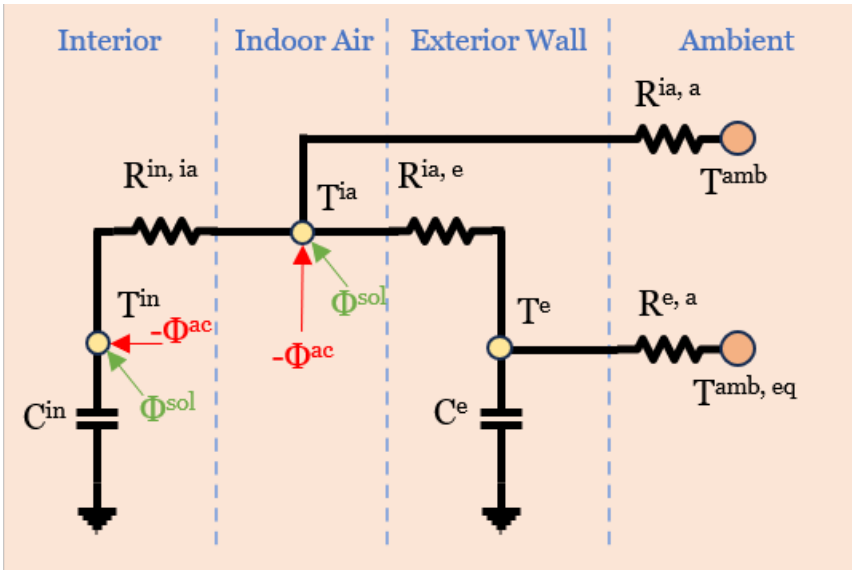

Figure B.1: 4R2C model network [5]

The model considers the impact of ambient temperature T^{amb}, solar irradiation Q^{irr} and cooling power Φ^{ac} on the indoor temperature T^{in} and the external wall temperature T^e. Notably, the layer of indoor air adjacent to the wall is treated as a separate node with a temperature T^{ia}, and the effect of Q^{irr} on the external wall is captured by an equivalent temperature $T^{amb,eq}$. The physical characteristics of the wall are represented by $R^{e,a}$ and C^e - in this work, concrete walls with a thickness of 150mm and a height of 2.6m are used to calculate typical values for these parameters. The effects of ventilation and poor insulation are captured by the shunt resistance $R^{in,a}$. The thermal capacitance of the indoor zone, C^{in}, captures the effects of objects inside the building, such as floors and furnishings. The heat transfer between T^{ia} and T^{in} is modelled as a convective process, moderated by the parameter $R^{in,ia}$. The parameter values used for the buildings are sampled from the ranges given in Table B.2, which reflect plausible ranges based on the interpretations mentioned above.

As a validation of the used parameters, the time constant τ_n of the final thermal networks were calculated using the methodology in [187], in which a study of 10,000

	C^{in} (Wh/K)	C^{e} (Wh/K)	$R^{in,ia}$ (W/K)	$R^{ia,a}$ (W/K)	$R^{ia,e}$ (W/K)	$R^{e,a}$ (W/K)
Mean	4000	4000	0.006	0.015	0.0009	0.001
Std	1000	1000	0.003	0.003	0.0005	0.0003
Min	1000	1000	0.0005	0.0005	0.0001	0.0005
Max	100000	500000	0.05	0.05	0.01	0.05

Table B.2: Parameter ranges for 4R2C models

real buildings in North America is performed. It was found that the distribution of τ_n obtained from the models closely match the values reported from the study, with a peak between $\tau^n = 20h$ and $\tau^n = 40h$. Fig B.2 depicts the temperature evolution of a few example buildings (for the same setpoint and AC power), clearly showing the impact of the different thermal model parameters.

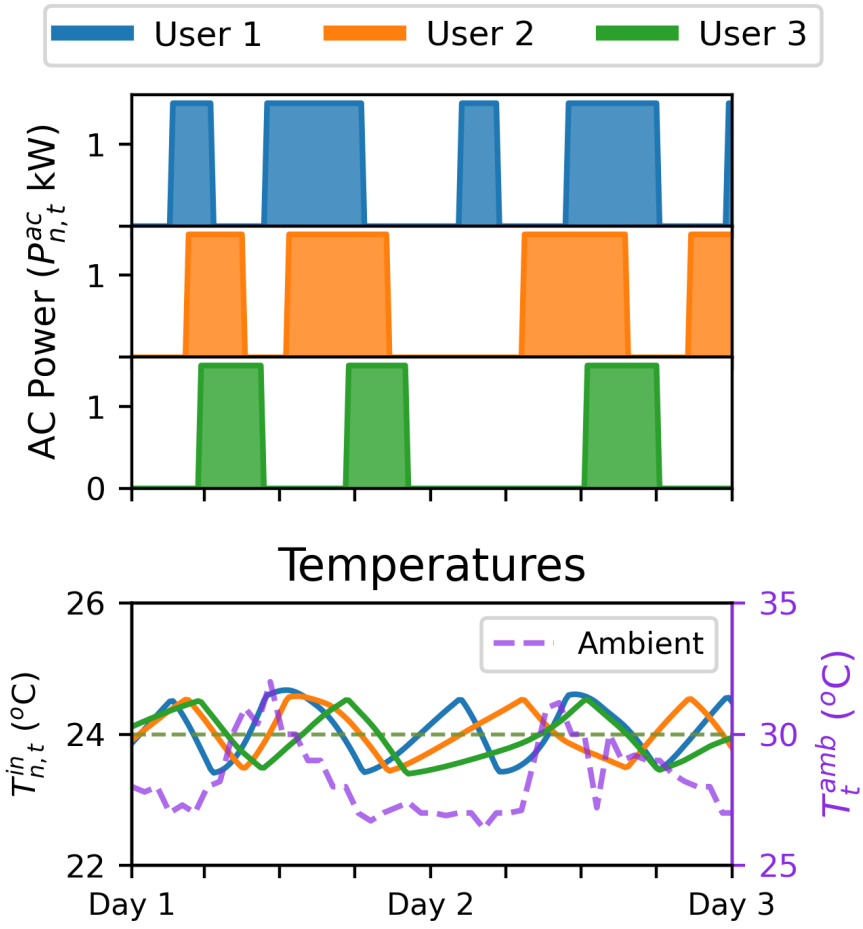

Figure B.2: Cooling power and indoor temperatures for different buildings in the reference system

Publications

List of Publications

1. S. Ram Kumar, A. Easwaran, B. Delinchant, and R. Rigo-Mariani, "Deep Reinforcement Learning for Coordinated Air-Conditioner Control in Groups of Buildings Using Smart Meter Data," *Journal Submission*, Under Review, Jul. 2025.

    - Journal paper regarding the multi-building co-ordination problem in Chapter IV.
    - Some aspects of the paper are also present in Chapter I, related to behind-the-meter and front-of-meter control.

2. S. Ram Kumar, A. Easwaran, B. Delinchant, and R. Rigo-Mariani, "Improving Demand Response Programs Using Override Signals with Reinforcement Learning," in *Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems (e-Energy '25)*, New York, NY, USA: Association for Computing Machinery, Jun. 2025, pp. 603–611. doi: 10.1145/3679240.3734657.

    - Conference paper discussing offline initialization and online adaptation in the multi-building context, based on override signals.
    - A subset of the results are included in Chapter IV.

3. S. R. Kumar, A. Easwaran, B. Delinchant, and R. Rigo-Mariani, "Real-time Retail Electricity Pricing Using Offline Reinforcement Learning," in *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems (e-Energy '24)*, New York, NY, USA: Association for Computing Machinery, 2024, pp. 454–458. doi: 10.1145/3632775.3661964.

    - Conference paper discussing a general framework for RL in multi-building problems based on consumer models.
    - Some aspects of this work were developed into the journal submission above, and discussed in Chapter IV.

4. S. R. Kumar, R. Rigo-Mariani, B. Delinchant, and A. Easwaran, "Towards Safe Model-Free Building Energy Management using Masked Reinforcement Learning," in *2023 IEEE PES Innovative Smart Grid Technologies Europe (ISGT EUROPE)*, IEEE, Oct. 2023, pp. 1–5. doi: 10.1109/ISGTEUROPE56780.2023.10407781.

    - Conference paper describing the use of action masking for a single-building control problem.
    - The action masking experiments in Chapter III were based on this work.

5. S. R. Kumar, A. Easwaran, B. Delinchant, and R. Rigo-Mariani, "Behavioural Cloning Based RL Agents for District Energy Management," in *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, New York, NY, USA: Association for Computing Machinery, Nov. 2022, pp. 466–470. doi: 10.1145/3563357.3566165.

    - Conference paper describing offline initialization for a multi-building control problem based on cloning from an RBC.
    - The general approach is used in the offline initialization step in Chapter III.