Physics Informed Neural Networks for Control Oriented Thermal Modeling of Buildings

Gargya Gokhale[a,*], Bert Claessens[b], Chris Develder[a]

[a] *IDLab, Ghent University – imec, Belgium*
[b] *Centrica Business Solutions, Belgium*

Abstract

Buildings constitute more than 40% of total primary energy consumption worldwide and are bound to play an important role in the energy transition process. To unlock their potential, we need sophisticated controllers that can understand the underlying non-linear thermal dynamics of buildings, consider user comfort constraints and produce optimal control actions. A crucial challenge for developing such controllers is obtaining an accurate control-oriented model of a building. To address this challenge, we present a novel, data-driven modeling approach using physics informed neural networks. With this, we aim to combine the strengths of two prominent modeling frameworks: the interpretability of building physics models and the expressive power of neural networks. Specifically, we use measured data and prior information about building parameters to realize a neural network model that is guided by building physics and can model the temporal evolution of room temperature, power consumption as well as the hidden state, i.e., the temperature of building thermal mass. The main research contributions of this work are: (1) we propose two new variants of physics informed neural network architectures for the task of control-oriented thermal modeling of buildings, (2) we show that training these architectures is data-efficient, requiring less training data compared to conventional, non-physics informed neural networks, and (3) we show that these architectures achieve more accurate predictions than conventional neural networks for longer prediction horizons (as needed for effective control). We test the prediction performance of the proposed architectures using both simulated and real-word data to demonstrate (2) and (3) and argue that the proposed physics informed neural network architectures can be used for control-oriented modeling.

Keywords: Physics-informed neural networks, control-oriented modeling, thermal building models, deep learning

1. Introduction

According to the recent IPCC report on climate change, global temperature is expected to reach the 1.5°C threshold in the next decades [1]. In the fight against climate change, the energy and power sector is going through numerous changes such as phasing out of coal-based generation, the addition of renewable energy sources and decentralization of generation and storage units. Concurrently, there is a growing need for efficient and flexible energy consumption that can accommodate the energy generated by intermittent renewable energy sources such as wind and solar power [2]. An important sector for providing this energy efficiency and flexibility is the building sector. As of 2016, buildings accounted for 40% of total primary energy consumption worldwide and around 55% of total electricity consumption in the EU [3]. With these numbers expected to rise over the years, efficient control of building energy consumption will play a crucial role in the energy transition process.

Significant research has been carried out in the context of control algorithms for energy management in buildings, ranging from simple Rule-based Controllers to advanced controllers like Model Predictive Control (MPC) and Reinforcement Learning (RL) [4]. In MPC, a physical model of the system is used to anticipate the future behavior of the sys-

*Corresponding Author
Email address: gargya.gokhale@ugent.be (Gargya Gokhale)

tem and optimize its performance [5]. This enables MPC-based controllers to be sample efficient and produce interpretable control decisions. However, the accuracy of MPC is closely related to the fidelity of the model, which is often difficult to obtain for real-world scenarios [6]. Contrary to this, data-driven controllers like RL, work directly with past interactions between the system, without the need for explicit physics knowledge. Although these RL-based controllers have shown promising results, they present a black-box solution that requires large amounts of training data. Additionally, in previous work such as [7], RL controller was trained using a physics model-based simulator to ensure that training data obtained was sufficiently diverse and to avoid taking harmful exploration actions.

This makes obtaining accurate building models a crucial requirement for developing better control algorithms. A variety of modeling techniques have been studied previously and are broadly classified into physics models (white box, grey box) and data-driven models (black box) [8]. The physics models involve solving a system of partial differential equations based on the underlying physical laws, commonly achieved using numeric solvers such as EnergyPlus, Modelica, as presented in, e.g., [9, 10]. The use of such models however has been limited in the control domain, primarily due to the high computational cost associated with solving the underlying system of partial differential equations [9]. Alternatively, a lumped parameter model using resistive and capacitive networks is used for control-oriented modeling. With this framework, different thermal components in a building are modeled using an RC network and simplified to obtain a lower order model that is easier to solve. However, even with these approximations, the models obtained are highly specific and require significant modeling effort as demonstrated in [11].

Data-driven models circumvent these modeling challenges by relying completely on obtained data. Previously, techniques such as ARIMA, Genetic Algorithms, Neural Networks, etc., have been studied and have shown good modeling capabilities [8, 12]. Yet, as discussed in [8], these techniques have their own challenges in the form of huge training data requirement and lack of interpretability.

To get the best of both these worlds, we propose to incorporate self-learning, physics guided models with model-based reinforcement learning algorithms to develop interpretable control agents in a data-driven manner. As a first building block, we propose to work with Physics Informed Neural Network architectures to learn physically relevant control-oriented models of real-world systems. This is achieved by explicitly providing information related to the underlying physics of the system to a deep neural network during the training procedure.

The main contributions of this paper can be summarized as:

(1) We propose two new variants of physics informed neural network architectures (Section 3) for control-oriented modeling of thermal behavior of a building and validate their accuracy using simulated data (Section 5.1).

(2) Based on real-world cold storage data (Section 4) we show that these physics informed neural networks perform better than conventional, non-physics informed neural networks at predictions for longer time horizons (Section 5.3).

(3) We further show that training these physics informed neural network architectures is a data-efficient process, requiring less training data than conventional, non-physics informed neural networks (Section 5.2).

While we acknowledge that the general concept of using physics informed neural networks models in itself is not new (as indicated through the literature review in the subsequent Section 2), the specific physics informed neural network model we have designed (incorporating basic constraints based on a simple RC model) is. Through aforementioned experiments we demonstrate our model's feasibility and practical applicability, based on experiments using real-world data.

2. Related Work

This section presents a non-exhaustive review of previous work related to our paper. We specifically focus on (i) Building Control and Modeling Algorithms and (ii) Physics Informed Neural Networks.

2.1. Building Control and Modeling

Extensive research has been carried out previously in this domain, with work such as [4, 13] presenting exhaustive reviews of different control algorithms. MPC has emerged as an established control technique with works such as [14, 15, 16] presenting case studies for practical implementations in

real-world buildings. These works show that MPC-based control strategies can lead to cost savings of about 20% compared to the existing rule-based control algorithms. An MPC strategy involves a physical model of the system and a set of constraints to formulate a receding horizon optimization problem that is solved at every time step to obtain optimum control actions [5]. Authors in [15, 16] utilize a grey-box RC model for the buildings. This leads to a bilinear building model and results in a non-linear optimization problem that can still be solved with reasonable accuracy using a sequence of linear programs [16]. Solving this optimization problem is computationally expensive and can limit the practical applicability of MPC controllers. Besides this, MPC controllers are expensive to obtain as indicated in [16], whose authors conducted a cost-benefit analysis of using MPC-based control strategies in real-world buildings. They concluded that, while MPCs can lead to a decrease in operating costs, the investment costs are much higher, primarily due to higher costs associated with the modeling of buildings, thus prohibiting widespread commercial application. Further, these building models are seldom scalable and need to be developed for individual buildings. E.g., in [11], authors discuss the model identification process for a real-world building and present a procedure for estimating building parameters. This procedure leads to models with accurate multi-step temperature predictions (0.3°C prediction error). However, this identification process involves solving a quadratic program to obtain good initial estimates of building parameters, followed by solving a multi-step prediction optimization problem to obtain the final model. This highlights the high computational requirements for obtaining a good building model and the lack of scalability. Additionally, the modeling approaches presented above approximate the non-linear thermal dynamics of the building using a first-order Euler discretization. As presented in [17], such explicit approximations can lead to inaccurate buildings models especially in scenarios involving low data sampling rates. This leads to approximate models which can be susceptible to errors and lead to biased control actions, as discussed in [9, 18].

Data-driven control techniques circumvent aforementioned shortcomings of MPC by completely relying on collected data as presented in [19]. Owing to the recent success of works such as [20], Reinforcement learning-based controllers are gaining importance in developing controllers for buildings [21]. RL-based controllers are self-learning controllers that use data collected from past interactions between the system and the controller to learn the dynamics of the system and achieve a pre-defined objective [22]. Works such as [7, 23] have studied RL-based controllers in the context of building control and show that such RL controllers can lead to $5 - 12\%$ energy savings compared to the existing rule-based controllers. Additionally, [24] compares the performance of MPC and RL controllers to show that RL controllers are able to outperform a linear MPC-based controller for two different test scenarios. Though these works indicate promising results for RL-based controllers, they also highlight existing challenges in real-world deployment of RL. These include large training data requirement, lack of interpretability, need for safe explorations, etc. [25]. E.g., in [23], the authors use one year of data for training the RL controller using random explorations. Similarly, in [7], the authors use 2 months of temperature data for obtaining a training data size equivalent to 3000 simulated trajectories. This data intensive nature and need for significant exploration represents a common challenge faced by RL-based control strategies. Hybrid control approaches have been studied to mitigate some of these problems by combining domain knowledge with these RL controllers [18, 26]. E.g., in [26], the authors present a hybrid control strategy by merging model-free and model-based control strategies. They propose an aggregate-and-dispatch control framework for a cluster of water heaters in which an MPC controller calculates energy set-points for the cluster and the dispatch is carried out based on a fitted Q-iteration RL strategy.

We present a different approach, where instead of using a model of the system directly, we focus on learning this model using the available data and then using it in a model-based RL approach. While different techniques for this control-oriented modeling problem have been studied previously, these techniques were focused on creating convex, linear (or bi-linear), time invariant models compatible with MPC formulation and available optimization solvers [27, 28]. In contrast, our objective in this work is to learn a low dimensional, latent space dynamics model of the system to use it in RL, where these latent representations can be used to learn optimum control policies as demonstrated in [29, 30]. Concretely, we propose using Physics Informed Neural Network architectures [31].

2.2. Physics Informed Neural Network Architectures

As introduced in [31], Physics informed neural networks represent a novel class of neural network architectures where prior knowledge about the system is encoded explicitly in the architecture. This work is similar to [32], where inductive biases based on the underlying physics laws are coded directly into the network. Several works have built upon this idea and have shown promising results in obtaining approximate solutions for difficult physics problems such as two body mechanics [33], heat transfer [34]. In [31, 34], the encoded physics knowledge is strictly enforced on the predictions of the neural network and assumes the availability of complete physics. Differing slightly from this approach, in [33], the authors enforce partially known physics and learn remaining physics parameters using the available data. These approaches show that trained models are better at extrapolating and require fewer training samples. Consistent with these works, we propose using physics informed neural networks for modeling the thermal behavior of a building. This is an emerging domain in building energy modeling and control, with previous works such as [35, 36, 37] presenting different approaches for providing prior physics knowledge to conventional black-box algorithms. In [35], the authors present a Physics-informed ARMAX method for modeling buildings and using these models with MPCs. The results indicate good performance of MPCs based on this model, including significant training sample efficiency attributed to the prior knowledge that is provided to the physics-informed ARMAX models. In [36], the authors present a physically consistent neural network architecture comprising of a physics-informed module in parallel to a black box module. With this approach, the authors show good prediction performance for longer prediction horizons as compared to a grey-box model. In [37], the authors present a physics-constrained deep learning model for control oriented model of a commercial building. The authors propose a structured recurrent neural dynamics model that models the non-linear thermal dynamics of the building using individual linear neural blocks with constrained eigenvalues. The authors show that such an architecture is data efficient, requires less training time and gives state-of-the-art prediction results for the case of large office buildings.

Differing from the works presented above, we encode the physics directly by using neural network outputs to calculate additional physics-based losses. Additionally, we propose to extract low-dimensional latent representations which correspond to the hidden states of the system and use prior physics knowledge to guide them towards a physically relevant space. This ensures that the obtained latent representations are disentangled, as opposed to the unsupervised learning cases discussed in [38, 39]. Once trained, these physics informed neural network models can be used with model-based RL algorithms such as MuZero [29], Dreamer [30] to obtain optimum control actions for building control.

3. Mathematical Modeling

With a focus on obtaining control-oriented models, we first model the thermal behavior of the household as a discrete-time Markov Decision Process (MDP), a commonly used sequential modeling framework [22]. Subsequently, we train our physics-informed neural networks to predict the 'next state' given the 'current state' and 'action'. This modeling approach has been discussed in this section along with the formulation of Physics informed neural networks.

3.1. Problem Formulation

A Markov Decision Process (MDP) is a commonly used framework to model sequential decision making problems [22]. An MDP consists of 4 main components: state space ($\mathbf{X}$), action space ($\mathbf{U}$), state transition function (f) and reward function (ρ). In a fully observable setting, the transition function $f\colon \mathbf{X} \times \mathbf{U} \times \mathbf{W} \rightarrow \mathbf{X}$ represents the true mapping at time step i, between the current state ($\mathbf{x}_i$), the current action ($\mathbf{u}_i$), an exogenous parameter ($\mathbf{w}_i$) and the next state ($\mathbf{x}_{i+1}$) of the system and is given as:

$$\mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i), \tag{1}$$

Here, $\mathbf{w}_i$ represents the stochasticity in the system and is assumed as an independent random variable. The transition function (f) represents the true dynamics of the system and to obtain a control-oriented model, it is necessary to approximate this transition function. For data-driven methods, this reduces the problem into a supervised learning

problem with the objective of estimating the transition function using a labeled set of state transitions $\mathcal{F} = \{(\mathbf{x}_1, \mathbf{u}_1, \mathbf{w}_1, \mathbf{x}_2), \ldots, (\mathbf{x}_N, \mathbf{u}_N, \mathbf{w}_N, \mathbf{x}_{N+1})\}$. E.g., a neural network with parameters θ can be trained using Stochastic Gradient Descent (SGD) to solve the following optimization problem:

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_{i+1} - f_\theta(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i))^2. \tag{2}$$

With this conventional approach, the neural network learns to estimate the transition function by fully relying on the training data-set, without explicitly learning the underlying physical relationships. It should be noted that Eq. (2) represents the scenario for a fully observable system, where complete state information is known and can be used to obtain predictions for the next states. However, a real-world system; such as a thermal model for a building; is generally partially observable where some state parameters cannot be measured or obtained directly. In such cases, using Eq. (2) directly is not useful as the observed states lack complete information. To mitigate this, [22] presents different approaches, one of which involves engineering new high-dimensional features based on the observed states. E.g., in case of a thermal model for a building, the observed state can include measurements of room temperature or actual power consumption, whereas a hidden state parameter can be the temperature of building thermal mass (e.g., walls, furniture etc.) which is difficult to measure or estimate accurately. Accordingly, to compensate for this missing state parameter, a sequence of past room temperature measurements can be used in lieu of a single room temperature measurement and this corresponds to an engineered feature for mitigating partial observability of this system.

For such partially observable MDPs, the state space($\mathbf{X}$) consists of an observable component ($\mathbf{X}^{\text{obs}}$) and a feature engineered component ($\mathbf{X}^{\text{f}}$) such that $\mathbf{X} = \mathbf{X}^{\text{obs}} \times \mathbf{X}^{\text{f}}$. With this high dimensional state representation, a neural network can be trained to estimate the next observable state ($\mathbf{x}^{\text{obs}}_{k+1}$) given state, action and other exogenous inputs, thus modifying the optimization problem in Eq. (2) as:

$$\begin{aligned} \hat{\mathbf{x}}^{\text{obs}}_{i+1} &= f_\theta(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i), \\ \min_{\theta} \frac{1}{N} \sum_{i=1}^{N} &(\mathbf{x}^{\text{obs}}_{i+1} - f_\theta(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i))^2. \end{aligned} \tag{3}$$

Aside from this data-driven approach, for problems where the physics of the system are known apriori, the system dynamics can be approximated using a system of ordinary or partial differential differential equations that relate the observable states ($\mathbf{x}^{\text{obs}}_i$), actions ($\mathbf{u}_i$), other exogenous factors ($\mathbf{w}_i$), hidden state parameters ($\mathbf{z}_i$) and system parameters (Ω). This is represented using a generic differential operator ($\mathcal{D}_\Omega$) as:

$$\mathcal{D}_\Omega(\mathbf{x}_i, \mathbf{u}_i, \mathbf{z}_i, \mathbf{z}_{i+1}, \mathbf{w}_i) = 0. \tag{4}$$

In the following section, we present the physics informed neural network architectures, which combine Eq. (3)–(4) to obtain control-oriented models for systems where the underlying physics are known a priori.

3.2. Physics Informed Neural Network Architectures

Consistent with previous works such as [31, 33], we explicitly encode the underlying physics of the system in a standard neural network architecture and then train it on the data collected. Assuming a partially observable setting, the network is trained to predict the next observable state ($\mathbf{x}^{\text{obs}}_{i+1}$) and a latent representation ($\mathbf{z}_i$) using a high dimensional state input ($\mathbf{x}_i$), action ($\mathbf{u}_i$) and exogenous information ($\mathbf{w}_i$). This is done by setting up a constrained optimization problem based on Eq. (3) as:

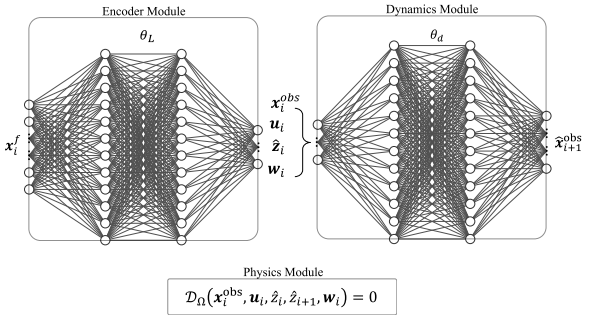
$$\begin{aligned} \min_{\theta,\Omega} \ & \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{x}^{\text{obs}}_{i+1} - \hat{\mathbf{x}}^{\text{obs}}_{i+1} \right)^2 \\ \text{s.t.} \ & \mathcal{D}_\Omega(\mathbf{x}_i, \mathbf{u}_i, \hat{\mathbf{z}}_i, \hat{\mathbf{z}}_{i+1}, \mathbf{w}_i) = 0, \\ \forall \ & (\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i, \mathbf{x}^{\text{obs}}_{i+1}) \in \mathcal{F}. \end{aligned} \tag{5}$$

To solve this optimization problem, we define a loss function composed of two terms; $\mathcal{L}_{\text{reg}}$ represents the mean squared error loss for regression and $\mathcal{L}_{\text{phys}}$ represents the physics-based loss that makes the network adhere to the underlying physics. It should be noted that for real-world scenarios the dynamics of the system is influenced by exogenous parameters/ noise and does not exactly satisfy Eq. (4). To manage this, we replace the strict equality by a least-squared error term as shown in the loss term formulation in Eq. (6).

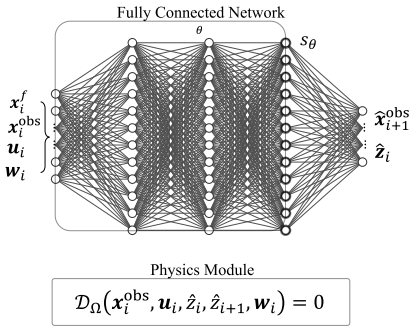
$$\begin{aligned} \text{Loss} &= \mathcal{L}_{\text{reg}} + \lambda \, \mathcal{L}_{\text{phys}} \\ \mathcal{L}_{\text{reg}} &= \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{x}^{\text{obs}}_{i+1} - \hat{\mathbf{x}}^{\text{obs}}_{i+1} \right)^2 \\ \mathcal{L}_{\text{phys}} &= \frac{1}{N} \sum_{i=1}^{N} \left(\mathcal{D}_\Omega(\mathbf{x}_i, \mathbf{u}_i, \hat{\mathbf{z}}_i, \hat{\mathbf{z}}_{i+1}, \mathbf{w}_i) \right)^2 \end{aligned} \tag{6}$$

The influence of the physics-based loss term is regulated using λ. The optimization problem defined in Eq. (5) can then be solved by using stochastic gradient descent to minimize the loss defined in Eq. (6).

Based on this formulation, we propose two variants of Physics Informed Neural Networks as shown in Fig. 1. The proposed networks have different architectures but are both trained based on the methodology described in Eq. (5)–(6). Figure 1(a)



(a) PhysNet



(b) PhysReg MLP

Figure 1: Physics Informed Neural Network architectures. $\mathbf{x}_i^{\mathrm{f}}$ represents a high dimensional feature engineered state component of the system, whereas $\mathbf{x}_i^{\mathrm{obs}}$ represents the observable components of the input sample i. $\mathbf{z}_i$ represents a low-dimensional latent representation of the system.

presents an architecture comprising two modules, an Encoder and a Dynamics module. The encoder module is parameterized by θ_{L} and the dynamics module is parameterized by θ_{d}. The encoder module creates a bottleneck and encodes the high dimensional, feature engineered component of state inputs $(\mathbf{x}_i^{\mathrm{f}})$ into a low dimensional latent representation $(\hat{\mathbf{z}}_i)$. This latent representation along with observable state information $(\mathbf{x}_i^{\mathrm{obs}})$, action (u_i) and other exogenous information $(\mathbf{w}_i)$ are then used by the dynamics module of the network to predict the next observable state $(\hat{\mathbf{x}}_{i+1}^{\mathrm{obs}})$ of the system. Thus, a forward pass of this network can be expressed as:

$$\begin{aligned} \hat{\mathbf{z}}_i &= g_{\theta_L}(\mathbf{x}_i^{\mathrm{f}}), \\ \hat{\mathbf{x}}_{i+1}^{\mathrm{obs}} &= h_{\theta_d}(\hat{\mathbf{z}}_i, \mathbf{x}_i^{\mathrm{obs}}, u_i, \mathbf{w}_i) \end{aligned} \tag{7}$$

With this architecture, the prediction for next observable state depends on, among other parameters, the prediction of the latent representation $(\hat{\mathbf{z}}_i)$. This ensures that the encoded representation obtained from this network contains information regarding the dynamics of the system.

Differing slightly from this approach, Fig. 1(b) presents a conventional fully connected neural network architecture where physics knowledge is incorporated based on Eq. (5)–(6). The inputs of this architecture comprise of the full state representation $(\mathbf{x}_i = (\mathbf{x}_i^{\mathrm{f}}, \mathbf{x}_i^{\mathrm{obs}}))$, action and exogenous information. With these inputs, the network predicts the next observable state and a latent representation simultaneously. Thus, there is explicit parameter sharing between these predictions and these shared parameters are represented by θ. The output layer uses an identity activation function and hence the outputs $(\hat{\mathbf{x}}_{i+1}^{\mathrm{obs}}$ and $\hat{\mathbf{z}}_i)$ are linear combinations of output of the last hidden layer of the network (s_θ). Thus, a forward pass of this network can be formulated as:

$$\begin{aligned} \hat{\mathbf{z}}_i &= \mathbf{g}_1 s_\theta(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i) + \mathbf{g}_2, \\ \hat{\mathbf{x}}_{i+1}^{\mathrm{obs}} &= \mathbf{h}_1 s_\theta(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i) + \mathbf{h}_2, \end{aligned} \tag{8}$$

where $\mathbf{g}_1$, $\mathbf{g}_2$, $\mathbf{h}_1$ and $\mathbf{h}_2$ are matrices of appropriate dimensions. With this parameter sharing, the obtained latent representation does not contribute in obtaining the predictions for the next observable state. Consequently, this latent representation may not contain sufficient information about the dynamics of the system. However, due to the physics, the latent representation is physically relevant and represents the hidden parameters of the system. Hence, in this case, the physics module acts as a regularization term guiding the network to learn the dynamics of the system and some latent representations simultaneously.

4. Experimental Setup

In this section, we apply the general methodology introduced in Section 3 for the case of thermal modeling of a building. We will detail the thermal building model used, the type of experiments performed and the configurations of physics informed

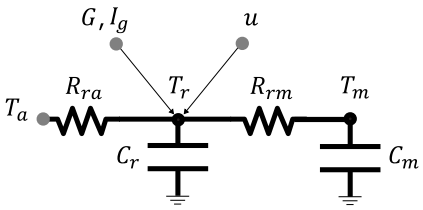


Figure 2: The resistance-capacitance network thermal model of the building

neural network architectures used. We compare the prediction accuracy of both architectures against a similar conventional neural network and assess whether the proposed architectures can be used for control applications.

4.1. Thermal Model of a Building

A simplified scenario is considered with a single room (or zone) that is heated using a heat source. To model this scenario, we adopt a grey box modeling approach using the 2R2C network model [40], illustrated in Fig. 2 and with the following state-space formulation:

$$\begin{bmatrix}\dot{T}_r\\ \dot{T}_m\end{bmatrix} = \begin{bmatrix}-\left(\frac{1}{C_r R_{ra}}+\frac{1}{C_r R_{rm}}\right) & \frac{1}{C_r R_{rm}}\\ \frac{1}{C_m R_{rm}} & -\frac{1}{C_m R_{rm}}\end{bmatrix}\cdot\begin{bmatrix}T_r\\ T_m\end{bmatrix} + \begin{bmatrix}b\\ 0\end{bmatrix}\cdot u + \begin{bmatrix}\frac{\alpha}{C_r} & \frac{\beta}{C_r} & \frac{1}{C_r R_{ra}}\\ \frac{1-\alpha}{C_m} & \frac{1-\beta}{C_m} & 0\end{bmatrix}\cdot\begin{bmatrix}G\\ I_g\\ T_a\end{bmatrix}. \tag{9}$$

Here, T_r, T_m and T_a are the room temperature, temperature of building's thermal mass and outside temperature respectively, G, I_g represent solar irradiance and internal heat gains, and R_i, C_j correspond to heat transfer parameters of the building. The room temperature T_r is an observable state of the system that can be measured. Contrarily, T_m is a hidden state of the system which cannot be measured directly and, in most cases, is extremely difficult to estimate. This modeling approach hence leads to a partially observable model of the building. Additionally, a low-level back-up controller is assumed which ensures that the room temperature remains within a predefined set of limits based on the comfort of the user. The action of this back-up controller affects the actual power consumption (u_i^{phys}) which is modeled as:

$$u_i^{\text{phys}} = \begin{cases} 0 & : T_{r,i} > T_r^{\max} \\ u_i & : T_r^{\min} \le T_{r,i} \le T_r^{\max} \\ u^{\max} & : T_{r,i} < T_r^{\min} \end{cases} \tag{10}$$

This backup controller ensures the comfort of the user and its actions leads to a difference between the power demanded (u_i) and the actual power consumed (u_i^{phys}). To solve Eq. (9)–(10), an accurate estimate of hidden state (T_m) is required along with accurate measurements related to exogenous quantities like G and I_g. Since in practice precise estimates, measurements are difficult to obtain, we will eventually get only an approximate solution. Further, the building parameters like conductivity of different walls change over time due to deterioration and lead to model bias. Hence modeling a household directly using Eq. (9)–(10) is a difficult and expensive process and can lead to biased and sub-optimal control policies.

4.2. Physics Informed Neural Network Configurations

The state-space model defined in Eq. (9) is a continuous time model. To map this model as an MDP, we discretize it based on the frequency of choosing an action (u). For our case study, we set this frequency to one action every 30 minutes and assume that the inside room temperature and power consumed by the heating source are monitored over this fixed time interval (Δt). The objective of the physics informed neural network model is to predict the room temperature and the power consumed for subsequent time steps.

Both architectures shown in Fig. 1 were used and prior physics information was given by discretizing Eq. (9). This state-space model leads to a partially observable system. To mitigate this, input in the form of a sequence of past k observable states and actions ($\mathbf{x}_i^{\text{f}}$) along with observable state, actions and other exogenous information in the form of time of day (t) and outside air temperature is used. With these inputs, the networks predict the room temperature, power consumption and estimate the temperature of building thermal mass (T_m). The resulting state and action definitions are summarized in Table 1.

The parameter k, referred to as 'depth', controls the amount of information given to the neural network. It is important to note that the observed states for time step i consist of the room

Table 1: State and Action Definitions for time step i

Symbol	Physical Meaning
$\mathbf{x}_i^f$	$\{(T_{r,i-k},\ldots,T_{r,i-1}),$ $(u_{i-k-1}^{\text{phys}},\ldots,u_{i-2}^{\text{phys}})\}$
$\mathbf{x}_i^{\text{obs}}$	$(T_{r,i}, u_{i-1}^{\text{phys}})$
$\mathbf{w}_i$	$(t_i, T_{a,i})$
$\mathbf{z}_i$	$T_{m,i}$
u_i	Controller Power

temperature at this time step $(T_{r,i})$ and the actual power that was consumed during the last time step (u_{i-1}^{phys}). Prior physics knowledge is provided to these architectures by directly plugging in Eq. (9) in the form of

$$\begin{bmatrix}\dot{T}_r\\ \dot{T}_m\end{bmatrix} = \begin{bmatrix}-a_{11} & a_{12}\\ a_{21} & -a_{22}\end{bmatrix}\cdot\begin{bmatrix}T_r\\ T_m\end{bmatrix} + \begin{bmatrix}b\\ 0\end{bmatrix}\cdot u + \begin{bmatrix}c_{11} & c_{12} & c_{13}\\ c_{21} & c_{23} & 0\end{bmatrix}\cdot\begin{bmatrix}0\\ 0\\ T_a.\end{bmatrix} \tag{11}$$

The parameters a_i, b, c_j are building specific parameters and initialized based on the building EPC values and further tuned during the training phase. This ensures that in the absence of accurate values of these parameters, the model can be initialized with approximate values. Moreover, these values can be tuned over time, thus taking into account any natural variations. It should be noted that this prior knowledge can be provided based on any other model of choice. However, as we focus on obtaining control-oriented models, a 2R2C modeling approach was chosen, which has been used previously in developing MPC-based control strategies [14, 40]. With this information setting, the different components of the loss function defined in Eq. (6) can be formulated as:

$$\begin{aligned}\mathcal{L}_{\text{reg}} &= \frac{1}{N}\sum_{i=1}^{N}(T_{r,i}-\hat{T}_{r,i})^2 \\ &\quad + \frac{1}{N}\sum_{i=1}^{N}(u_i^{\text{phys}}-\hat{u}_i^{\text{phys}})^2, \\ \mathcal{L}_{\text{phys}} &= \frac{1}{N}\sum_{i=1}^{N}(T_{m,i}^{\mathcal{M}}-\hat{T}_{m,i})^2,\end{aligned} \tag{12}$$

The hidden state $(T_{m,i}^{\mathcal{M}})$ represents the physics module output and is computed by first estimating $\dot{T}_{r,i}$ and then using Eq. (11) to obtain $T_{m,i}^{\mathcal{M}}$ as shown in Eq. (13).

$$\begin{aligned}\dot{T}_{r,i} &= \frac{T_{r,i+1}-T_{r,i}}{\Delta t}\\ T_{m,i}^{\mathcal{M}} &= \frac{1}{a_{12}}(\dot{T}_{r,i}+a_{11}\hat{T}_{r,i}-b\hat{u}_i^{\text{phys}}-c_{13}T_{a,i})\end{aligned} \tag{13}$$

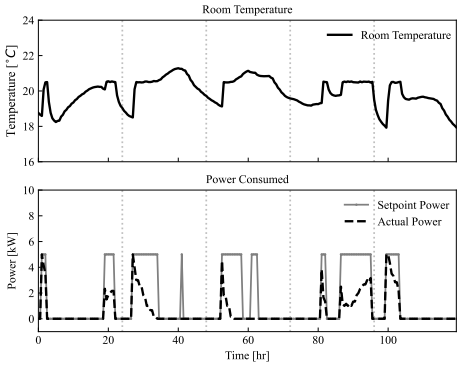
Here, $\hat{T}_{r,i}$ is obtained as an output by using input sample $i-1$ and $\hat{u}_i^{\text{phys}}$ is obtained by using input sample i. The target value for the hidden state is explicitly dependent on the predictions of the room temperature and the power consumed. The loss functions defined in Eq. (12) guides the outputs of the networks towards physically relevant values.

4.3. Training Data

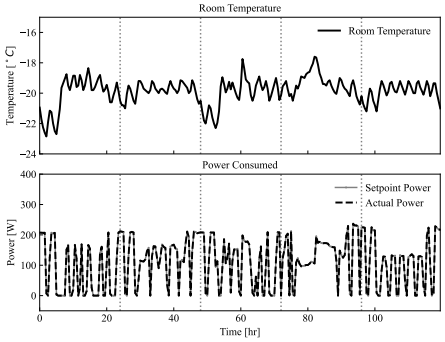
We considered two different scenarios for obtaining training data for these architectures: (1) a *simulated* single household environment, and (2) *real-world cold storage* data. The simulated scenario has been specifically designed to assess the capacity of the proposed physics informed neural network architectures to estimate the hidden state T_m of a building. After establishing this, later experiments focused on the real-world data scenario and assessed the performance of our proposed architectures for different configurations. Both scenarios involved observations related to room temperature, actual power consumption, control actions and outside air temperature. The frequency of these measurements was set to 1 measurement per 30 minutes. A training data-set equivalent to 120 days of such measurements was generated/collected. Similarly, a test data-set was generated equivalent to 5 days that were not a part of the training set. Each day corresponds to 48 input samples and the test sets for both scenarios used are shown in Fig. 3.

4.3.1. Simulated Data

In this scenario, Eq. (9)–(10) were solved, as discussed in [40]. A discretization step of 1 minute was assumed and a control action was taken every 30 minutes. For every minute, a first order approximation of Eq. (9) was solved to obtain the room temperature, hidden state and actual power consumption. To further simplify the scenario, we ignored the effects of solar irradiance and internal heat gain. The simulation was initialised by setting $T_r = T_m = 17°\text{C}$. Further, control action u was chosen randomly and did not follow any active control logic. Figure 3(a) shows a subset of data generated in this scenario.



(a) Simulated data for a household



(b) Real-world cold storage data

Figure 3: Test data-sets used for both data scenarios, (a) Simulated Data, (2) Real-world scenario.

.

4.3.2. Real-world Data

This scenario involved data obtained from a cold storage. This scenario is more complex than the simulated data generated as it involved actions taken by an active control strategy and also included solar irradiance, internal heat gains, etc. The influence of these exogenous factors was recorded only indirectly, via the room temperature measurements, due to absence of sensors to directly measure them. Figure 3(b) shows a subset of data corresponding to this scenario. In this cold storage case, no back-up controller was used and hence the actual power consumed is identical as the control setpoints.

4.4. Parameter Tuning for Physics Informed Neural Network Architectures

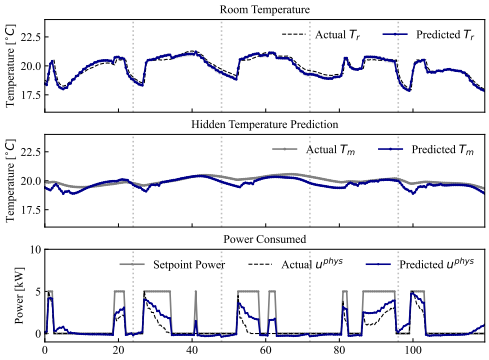
Besides different data scenarios, we also analyze the impact of different parameter configurations for both architectures. The input given to both architectures involves a sequence of past room temperatures and control actions. The length of this sequence, referred to as 'depth', determines the amount of past information available to the model and is an important parameter in the architecture. This information, to a certain level compensates for missing information like solar irradiance or internal heat gains, helping the model to better estimate the hidden state of the building (T_m). Further, the network sizes and hyperparameters (learning rate, type of optimizers) were tuned for a base case of setting $\lambda = 0$ for both architectures. This ensured that the network size and representative power was not constrained by the physics-based regularization, and we can observe supplementary gains in performance after tuning λ. The set of hyperparameters selected for both these architectures are listed in Appendix A. The hyperparameter values were obtained by minimizing the mean absolute error in predicted temperature as a performance metric. For each of these configurations, we train 20 seeded models and the results are expressed using the mean (and standard deviation) of these 20 models. Because of the low training sample regime, training multiple models ensures that we obtain a distribution of performance values, thus mitigating the effects of possible outliers due to under-fitting. Additionally, Appendix B includes information regarding the training time required for each configuration and details the hardware setup used.

5. Results and Discussions

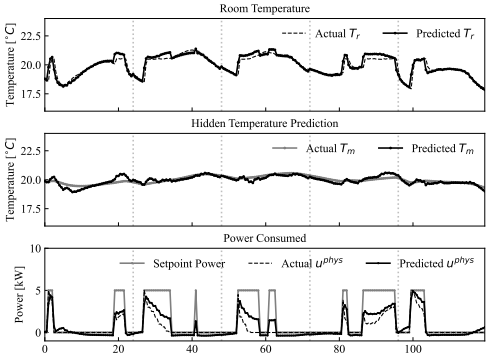
Three different experiments were performed to test our proposed PhysNet and PhysReg MLP architectures (Fig. 1) and assess their performance as a control-oriented model.

5.1. Architecture Validation

The aim of our first experiment was to validate the performance of the proposed physics informed neural network architectures in determining the quality of the hidden state estimates. For this purpose, simulated data was used for training and validation. The validation data-set, shown in Fig. 3(a), contains 240 samples for which we computed the Mean Absolute Error (MAE) of predicted room temperature (T_r) and predicted hidden state (T_m). A fixed training size of 120 days (5,760 samples) was used along with a fixed depth value of 8. Figure 4 shows the predictions of both architectures.



(a) PhysNet Architecture



(b) PhysReg MLP Architecture

Figure 4: Prediction results for the two physics informed neural network architectures on simulated data scenario

Table 2: Comparison of MAEs for room temperature (T_r) and hidden state (T_m) for proposed architectures

	MLP	PhysReg MLP	PhysNet
T_r	$0.209°C$	$0.197°C$	$0.226°C$
T_m	$1.413°C$	$0.385°C$	$0.436°C$

We note that for both cases, the room temperature and action predictions follow the actual values closely, indicating a good prediction performance. Additionally, the estimates of T_m track the actual values of hidden states, thus demonstrating the effectiveness of our proposed architectures for the given prediction task. Table 2 shows the MAE values for room temperature (T_r) and hidden state (T_m) predictions for this experiment.

Table 2 presents error values in °C. A conventional MLP with the same hyperparameters as the PhysReg model was used to benchmark the performance of PhysNet and PhysReg MLP architectures. For all three networks, the mean errors are less than 0.25°C, indicating a good performance. Comparing the architectures, the PhysReg model performs the best with an absolute error of 0.197°C. However, there is a significant difference between mean errors for the hidden state, where conventional MLPs cannot estimate this state due to lack of target values, thus performing poorly in this metric. The physics informed neural network architectures perform $60-70\%$ better than the conventional MLPs with an absolute error of less than 0.5°C. These results demonstrate that PhysNet and PhysReg MLP architectures can be used effectively to predict room temperature and hidden state and hence are more suitable for control oriented thermal modeling of a building.

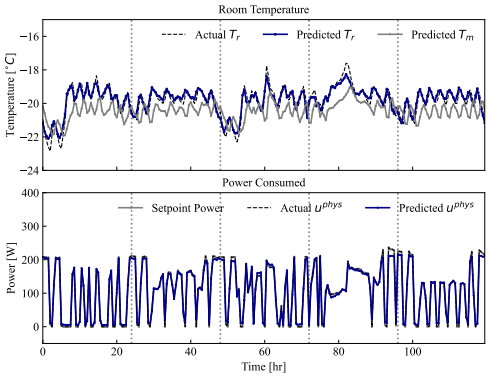
5.1.1. Interpretability

From the results presented in Table 2, it is evident that the proposed physics informed neural network architectures can effectively estimate the hidden state (T_m) while maintaining a good prediction accuracy for the observable state (T_r). It can be observed in Fig. 4 that the estimate of temperature of building thermal mass acts as a thermal inertia quantity, having slower time dynamics compared to the room temperature. This behavior is consistent with our intuition and mimics the actual temperature of thermal mass. Thus, the trained physics-informed models give us additional insights about the behavior of the building by providing accurate estimates of both observable and hidden states of the building system. Such insights can help to better understand the predictions of the neural network and can be leveraged to design interpretable data-driven controllers.

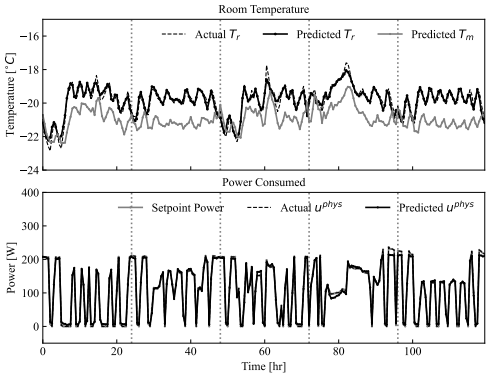
5.1.2. Real-world Data

Following these results, both physics informed neural network architectures were trained on real-world data obtained from a cold storage unit. Similar to the previous case, a training data size of 120 days was used and performance was validated on 5 test days, including a benchmark by a conventional neural network. Figure 5 shows the performance of PhysNet and PhysReg MLP architectures on the real-world data-set.

We note that both architectures accurately predict the room temperature and power consumption values for the 5 test days along with a plausible estimate of the hidden state of the system (T_m). The results shown in Figs. 4–5 and Table 2 validate the



(a) PhysNet Architecture



(b) PhysReg MLP Architecture

Figure 5: Prediction results for physics informed neural network architectures on the real-world data scenario.

performance of the proposed physics informed neural network architectures for the task of modeling the thermal behavior of a building.

5.2. Performance vs. Training Data Size

The second set of experiments analyzed the impact of training data size on the performance of physics informed neural network models. The motivation for using physics informed neural networks was to leverage prior knowledge to train models faster and more efficiently. To validate this, models were trained on real-world training data of varying size, sampled from the main training set. Each model was then tested using MAE as the performance metric on the test data-set of 240 samples (5 days) shown in Fig. 3(b). Two different test configurations were used, depending on the prediction horizon. Our architectures enable one-step ahead prediction. To obtain predictions for longer horizons, a recursive strategy was used, where the model output was fed back to the model as input to

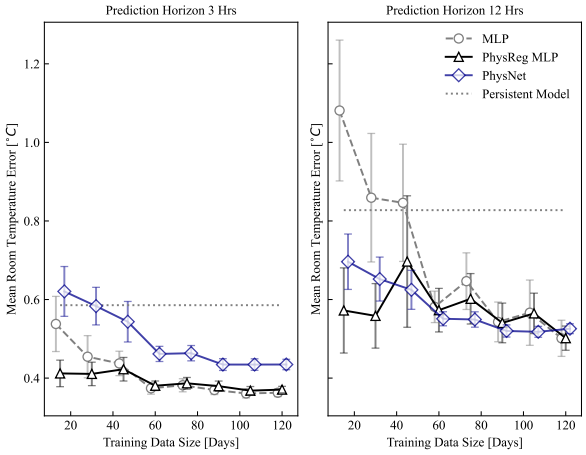


Figure 6: Mean room temperature prediction error for varying training data size. The plots represent mean error of 20 trained models and the error bars represent ± standard deviation. The models used in this experiment are trained and tested on real-world data obtained from a cold storage unit.

generate multi-step forecasts. This strategy mimics a tree search algorithm used in model-based RL techniques like [29]. The two test configurations used a prediction horizon of 3 hours (6 steps) and 12 hours (24 steps). The performance of physics informed neural networks was further compared to a conventional neural network and a persistence forecast model for both these configurations. Figure 6 shows the model performance for different training data sizes for the real-world data scenario.

We note that for a prediction horizon of 12 hours, both PhysNet and PhysReg MLP architectures perform better than the conventional MLP. For smaller training data sizes (15-45 days) the predictions for physics informed neural network architectures attain an MAE that is at least 15% lower than MLP. This difference decreases sharply with increasing training size, where for higher training sizes ($>$ 90 days) the performance of all three architectures is similar. Contrary to this, for a shorter prediction horizon, the conventional MLP outperforms the PhysNet architecture, and performs similarly as the PhysReg MLP architecture. Additionally, in both configurations, all three architectures outperform a persistence forecasting model of similar prediction horizon for most training data sizes. This shows that introducing prior knowledge to the neural network architecture aids the network to learn more efficiently and requires less training data to reach equally good (or better) performance.

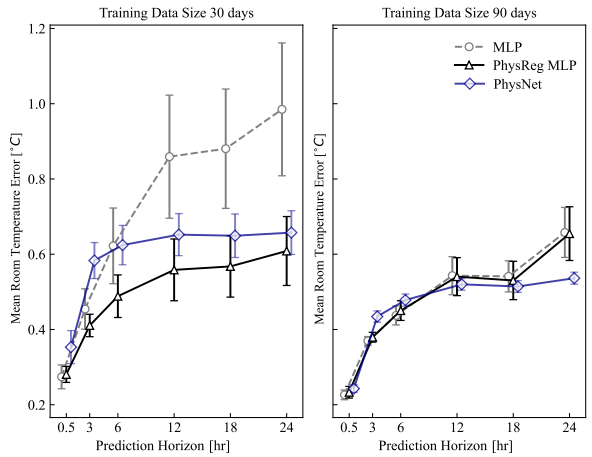


Figure 7: Mean room temperature prediction error for varying prediction horizons.The plots represent the mean error of 20 trained models and the error bars represent ± standard deviation. The models were trained and tested on the real-world cold storage data-set.

5.3. Performance vs. Prediction Horizon Size

From Fig. 6, we note a difference in performance for different prediction horizons. While it is intuitively expected that increasing the prediction horizon will lead to compounding of errors, it is of interest to analyze how this performance degradation evolves for each of the two architectures. This experiment, thus analyzes the performance of physics informed neural networks for varying prediction horizons. Because of their relevance for typical control time frames, prediction horizons of {0.5, 3, 6, 12, 18, 24} hours were selected, with each hour corresponding to 2 prediction steps. To include the impact of training data size, two training configurations of 30 days and 90 days were chosen. Like the previous experiment, real-world data was used with 5 test days as shown in Fig. 3(b). MAE of room temperature predictions was chosen as the performance metric and the performance was again benchmarked using a conventional MLP and a persistence forecast model with prediction horizon of 30 minutes (equalling 1 time step). Figure 7 presents the results obtained for this experiment.

We note that for a large training data (120 days), all three architectures perform similarly in terms of mean values. However, the error bars indicate that PhysNet, PhysReg MLP architectures produce results with a more narrow distribution. This indicates a stable training performance in case of physics informed neural network architectures. For low training sample configurations, the performance of conventional MLP deteriorates rapidly with increase in prediction horizon size, with an error of close to 1°C for the case of 24 hours. While there is a significant decrease in performance for physics informed neural networks, the error margins remain around 0.75°C with a standard deviation of ±0.3°C. This indicates that with less training data, the physics informed neural network models can use prior physics knowledge and lead to trained models that are stable and perform better than conventional MLPs. This is an important feature that can be leveraged in control applications for evaluating longer trajectories in tree searches.

These results demonstrate that introducing prior knowledge into a network leads to better predictions, makes the training process sample efficient and yields models that can be used for developing better control algorithms.

6. Conclusion

This work presented the application of physics informed neural networks for control-oriented thermal modeling of buildings. Our results show that both physics informed neural network architectures perform well for the given task of predicting the room temperature, with a low prediction error (less than 0.25°C). Further experiments confirm that physics informed neural networks are better suited for modeling in case of less training data and longer prediction horizons. This also indicates the robust training performance of PhysNets, PhysReg MLP architectures and their ability to generalize well even with fewer training samples. Additionally, we verify that the physics informed neural network models can estimate hidden states of the building effectively. This is an important feature and can be exploited further in developing control policies. Moreover, our proposed physics informed neural network architectures work with approximate values of model parameters and can incorporate partial knowledge about building physics. With such a setting, models for different buildings can be obtained using the same initial parameters and building physics, making this modeling approach scalable and easy to deploy.

Future Work

Future work will involve two key directions: (i) Developing Control Algorithms, and (ii) Improving PhysNet and PhysReg MLP architectures.

In (i) we will use these architectures in model-based RL algorithms like [29, 30]. The control agent will be capable of learning the model of the building and an optimum control policy simultaneously. Moreover, leveraging the learnt model, the agent can create a schedule for the next hours, making the decision making process interpretable and allowing human supervisory control. For (ii), we aim to improve the architecture by introducing a direct multi-step forecasting capacity rather than the current one-step prediction setting. This will allow the architecture to produce one shot forecasts for a pre-defined prediction window. Other improvements include assessing the performance benefits of using recurrent neural networks in the model architecture and the role of clustering and transfer learning for scalable model deployment.

Funding

This work was supported by the European Union's Horizon 2020 research and innovation programme under the projects BRIGHT (grant agreement no. 957816), RENergetic (grant agreement no. 957845) and BIGG (grant agreement no. 957047).

Appendix A. Hyperparameters for Physics Informed Neural Networks

Both variants of physics informed neural networks were implemented using Pytorch Lightning package, [41]. Table A.3 and Table A.4 present the hyperparamters chosen for these architectures. 20 models were training using the same set of hyperparameters and with different seeds between 1 to 20. Both architectures were trained using a batch size of 2048 and with 75 epochs.

Table A.3: Hyperparameters for PhysReg MLP Architecture

Parameter	Value
Optimizer	Adam
Learning Rate	0.001
Activation Function	tanh
Batch Size	2048
Hidden Layers	2
Neurons per layer	64

These hyperparameters were chosen using a grid search strategy and Mean Absolute Error for predictions on a validation set as the metric. The neural network hyperparameters were tuned first by setting λ equal to 0. After this, the physics informed neural network parameters were tuned.
More information regarding the code can be found on: `https://github.com/GargyaGokhale/PhysNet_Thermal_Models`

Table A.4: Hyperparameters for PhysNet Architecture

Parameter	Value
Optimizer	Adam
Learning Rate	0.001
Activation Function	tanh
Batch Size	2048
Encoder Module (θ_L)	
Hidden Layers	2
Neurons per layer	24
Dynamics Module (θ_d)	
Hidden Layers	1
Neurons per layer	128

Appendix B. Training Time and Hardware Configuration

For results presented in Section 5.1, 20 instances of the neural network were trained using the same set of hyperparameters but different randomly initialised weight and bias values. On training, these 20 instances were used and the mean of their prediction was used. Table B.5 presents the training time required for training these 20 instances for each of the two proposed physics informed neural network variants.
The training was carried out locally on a Dell laptop with Intel(R) Core(TM) i7-10850H CPU, 2.70GHz and 16 GB installed RAM.

Table B.5: Training Time Required

Neural Network Type	Time
PhysReg MLP	3 minutes
PhysNet	4 minutes

References

[1] P. Masson-Delmotte, V.and Zhai, A. Pirani, S. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J. Matthews, T. Maycock, T. Waterfield, O. Yelekçi,

R. Yu, Z. B., IPCC, 2021: Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change (2021).
URL https://www.ipcc.ch/report/ar6/wg1/FullReport

[2] A. Stawska, N. Romero, M. de Weerdt, R. Verzijlbergh, Demand response: For congestion management or for grid balancing?, Energy Policy 148 (2021) 111920.

[3] X. Cao, X. Dai, J. Liu, Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade, Energy and Buildings 128 (2016) 198–213. doi:10.1016/j.enbuild.2016.06.089.
URL http://dx.doi.org/10.1016/j.enbuild.2016.06.089

[4] T. Q. Péan, J. Salom, R. Costa-Castelló, Review of control strategies for improving the energy flexibility provided by heat pump systems in buildings, Journal of Process Control 74 (2019) 35–49. doi:10.1016/j.jprocont.2018.03.006.
URL https://doi.org/10.1016/j.jprocont.2018.03.006

[5] J. Drgoňa, J. Arroyo, I. C. Figueroa, D. Blum, K. Arendt, D. Kim, E. P. Ollé, J. Oravec, M. Wetter, D. L. Vrabie, et al., All you need to know about model predictive control for buildings, Annual Reviews in Control (2020).

[6] J. Cígler, D. Gyalistras, J. Široky, V. Tiet, L. Ferkl, Beyond theory: the challenge of implementing model predictive control in buildings, in: Proceedings of 11th Rehva world congress, Clima, Vol. 250, 2013.

[7] S. Brandi, M. S. Piscitelli, M. Martellacci, A. Capozzoli, Deep reinforcement learning to optimise indoor temperature control and heating energy consumption in buildings, Energy and Buildings 224 (2020) 110225.

[8] A. Foucquier, S. Robert, F. Suard, L. Stéphan, A. Jay, State of the art in building modelling and energy performances prediction: A review, Renewable and Sustainable Energy Reviews 23 (2013) 272–288.

[9] G. Gholamibozanjani, J. Tarragona, A. De Gracia, C. Fernández, L. F. Cabeza, M. M. Farid, Model predictive control strategy applied to different types of building for space heating, Applied energy 231 (2018) 959–971.

[10] D. Perera, D. Winkler, N.-O. Skeie, Multi-floor building heating models in matlab and modelica environments, Applied Energy 171 (2016) 46–57.

[11] E. Žáčeková, Z. Váňa, J. Cigler, Towards the real-life implementation of mpc for an office building: Identification issues, Applied Energy 135 (2014) 53–62.

[12] F. Ferracuti, A. Fonti, L. Ciabattoni, S. Pizzuti, A. Arteconi, L. Helsen, G. Comodi, Data-driven models for short-term thermal behaviour prediction in real buildings, Applied Energy 204 (2017) 1375–1387.

[13] J. Tarragona, A. L. Pisello, C. Fernández, A. de Gracia, L. F. Cabeza, Systematic review on model predictive control strategies applied to active thermal energy storage systems, Renewable and Sustainable Energy Reviews 149 (May) (2021). doi:10.1016/j.rser.2021.111385.

[14] J. Širokỳ, F. Oldewurtel, J. Cigler, S. Prívara, Experimental analysis of model predictive control for an energy efficient building heating system, Applied energy 88 (9) (2011) 3079–3087.

[15] I. Hazyuk, C. Ghiaus, D. Penhouet, Optimal temperature control of intermittently heated buildings using Model Predictive Control: Part I - Building modeling, Building and Environment 51 (2012) 379–387. doi:10.1016/j.buildenv.2011.11.009.

[16] D. Sturzenegger, D. Gyalistras, M. Morari, R. S. Smith, Model predictive climate control of a swiss office building: Implementation, results, and cost–benefit analysis, IEEE Transactions on Control Systems Technology 24 (1) (2015) 1–12.

[17] S. Baldi, S. Yuan, P. Endel, O. Holub, Dual estimation: Constructing building energy models from data sampled at low rate, Applied Energy 169 (2016) 81–92. doi:https://doi.org/10.1016/j.apenergy.2016.02.019.
URL https://www.sciencedirect.com/science/article/pii/S0306261916301428

[18] M. Bhardwaj, S. Choudhury, B. Boots, Blending mpc & value function approximation for efficient reinforcement learning, arXiv preprint arXiv:2012.05909 (2020).

[19] A. Kathirgamanathan, M. De Rosa, E. Mangina, D. P. Finn, Data-driven predictive control for unlocking building energy flexibility: A review, Renewable and Sustainable Energy Reviews 135 (August 2020) (2021) 110120. arXiv:2007.14866, doi:10.1016/j.rser.2020.110120.
URL https://doi.org/10.1016/j.rser.2020.110120

[20] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of go with deep neural networks and tree search, Nature 529 (2016) 484–503.

[21] J. R. Vázquez-Canteli, Z. Nagy, Reinforcement learning for demand response: A review of algorithms and modeling techniques, Applied energy 235 (2019) 1072–1089.

[22] R. Sutton, A. Barto, An introduction to reinforcement learning, Decision Theory Models for Applications in Artificial Intelligence: Concepts and Solutions (2011). doi:10.4018/978-1-60960-165-2.ch004.

[23] L. Yang, Z. Nagy, P. Goffin, A. Schlueter, Reinforcement learning for optimal control of low exergy buildings, Applied Energy 156 (2015) 577–586.

[24] G. Ceusters, R. C. Rodríguez, A. B. García, R. Franke, G. Deconinck, L. Helsen, A. Nowé, M. Messagie, L. R. Camargo, Model-predictive control and reinforcement learning in multi-energy system case studies, arXiv preprint arXiv:2104.09785 (2021).

[25] Z. Wang, T. Hong, Reinforcement learning for building controls: The opportunities and challenges, Applied Energy 269 (2020) 115036.

[26] M. Liu, S. Peeters, D. S. Callaway, B. J. Claessens, Trajectory tracking with an aggregation of domestic hot water heaters: Combining model-based and model-free control in a commercial deployment, IEEE Transactions on Smart Grid 10 (5) (2019) 5686–5695.

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

[28] E. Atam, L. Helsen, Control-oriented thermal modeling of multizone buildings: Methods and issues: Intelligent control of a building system, IEEE Control Systems Magazine 36 (3) (2016) 86–111. doi:10.1109/MCS.2016.2535913.

[29] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, D. Silver, Mastering Atari, Go, chess and shogi by planning with a learned model, Nature 588 (7839) (2020) 604–609. arXiv:1911.08265, doi:10.1038/s41586-020-03051-4.
URL http://dx.doi.org/10.1038/s41586-020-03051-4

[30] D. Hafner, T. Lillicrap, J. Ba, M. Norouzi, Dream to control: Learning behaviors by latent imagination, arXiv preprint arXiv:1912.01603 (2019).

[31] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics 378 (2019) 686–707. doi:10.1016/j.jcp.2018.10.045.
URL https://doi.org/10.1016/j.jcp.2018.10.045

[32] S. Greydanus, M. Dzamba, J. Yosinski, Hamiltonian neural networks, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019.

[33] W. Degroote, S. Van Hoecke, G. Crevecoeur, Physics-Based Neural Network Models for Prediction of Cam-Follower Dynamics Beyond Nominal Operations, IEEE/ASME Transactions on Mechatronics PP (c) (2021) 1. doi:10.1109/TMECH.2021.3101420.

[34] S. Cai, Z. Wang, S. Wang, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks for heat transfer problems, Journal of Heat Transfer 143 (6) (2021) 1–15. doi:10.1115/1.4050542.

[35] F. Bünning, B. Huber, A. Schalbetter, A. Aboudonia, M. Hudoba de Badyn, P. Heer, R. S. Smith, J. Lygeros, Physics-informed linear regression is competitive with two machine learning methods in residential building mpc, Applied Energy 310 (2022) 118491. doi:https://doi.org/10.1016/j.apenergy.2021.118491.
URL https://www.sciencedirect.com/science/article/pii/S0306261921017098

[36] L. Di Natale, B. Svetozarevic, P. Heer, C. N. Jones, Physically consistent neural networks for building thermal modeling: theory and analysis, arXiv preprint arXiv:2112.03212 (2021).

[37] J. Drgoňa, A. R. Tuor, V. Chandan, D. L. Vrabie, Physics-constrained deep learning of multi-zone building thermal dynamics, Energy and Buildings 243 (2021) 110992. doi:https://doi.org/10.1016/j.enbuild.2021.110992.
URL https://www.sciencedirect.com/science/article/pii/S0378778821002760

[38] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (8) (2013) 1798–1828. arXiv:1206.5538, doi:10.1109/TPAMI.2013.50.

[39] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, A. Lerchner, Understanding disentangling in β-vae, arXiv preprint arXiv:1804.03599 (2018).

[40] E. Vrettos, E. C. Kara, J. MacDonald, G. Andersson, D. S. Callaway, Experimental Demonstration of Frequency Regulation by Commercial Buildings-Part I: Modeling and Hierarchical Control Design, IEEE Transactions on Smart Grid 9 (4) (2018) 3213–3223. arXiv:1605.05835, doi:10.1109/TSG.2016.2628897.

[41] W. Falcon, The PyTorch Lightning team, PyTorch Lightning (3 2019). doi:10.5281/zenodo.3828935.