universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Robust Parameter Identification of a Reduced-Order Building Thermal Model Using an Inverse Physics-Informed Neural Network“

verfasst von / submitted by

Noah Sippl BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien / Vienna, 2026

Studienkennzahl lt. Studienblatt / Degree programme code as it appears on the student record sheet:	UA 066 910
Studienrichtung lt. Studienblatt / Degree programme as it appears on the student record sheet:	Masterstudium Computational Science
Betreut von / Supervisor:	Assoz. Prof. Dipl.-Ing. Dr.techn. Sebastian Tschiatschek BSc

Acknowledgements

I would like to thank the AIT Austrian Institute of Technology and my colleagues, Peter Widhalm, Regina Hemm, and Nandor Mihaly, for their expertise, support, and contributions to the project. I would also like to thank my supervisor at University of Vienna, Sebastian Tschiatschek, for his assistance, advice, and valuable feedback.

Declaration

Large language models and other AI-assisted tools were used selectively during the preparation of this thesis for linguistic revision, stylistic refinement, and structural support. No AI-generated content was included without critical evaluation and revision by the author. The author assumes full responsibility for all content, interpretations, arguments, and conclusions presented in this thesis.

Abstract

Renewable energy sources impose new challenges on the electricity grid. Due to their intermittent generation, grid stability hinges on the efficient distribution of available electricity. Buildings consume a significant amount of electricity for space conditioning and, given their thermal mass, can maintain comfortable temperature levels even without active conditioning. Consequently, the thermal mass of buildings can be used as energy storage, allowing space-conditioning demand to be shifted toward periods of high renewable generation. This redistribution is constrained by occupant comfort, requiring accurate and physically consistent long-term indoor temperature predictions to evaluate whether candidate redistributions satisfy comfort constraints. Parameter estimation of reduced-order building thermal models is the standard approach for obtaining representations that enable physically consistent temperature predictions. However, due to the constrained operating conditions observed in buildings, temperature dynamics are weakly excited and key influences overlap in time, making it difficult to infer reliable causal relationships. This poses a major challenge for parameter identification, making the identification process itself the limiting factor for robust indoor temperature predictions. Here we show that parameter identification using an inverse physics-informed neural network (PINN), which jointly estimates thermal parameters and continuous temperature trajectories under physical constraints, improves predictive accuracy across all prediction horizons and reduces variability in predictive accuracy across thermal models identified from different training datasets. Moreover, we show that this approach achieves comparable data efficiency to a gradient-based parameter identification procedure. By decoupling trajectory approximation from parameter estimation, the inverse PINN becomes less sensitive to parameter initialization and more robust to variations in the available training data. Our findings suggest that inverse PINNs provide a promising alternative for parameter identification of reduced-order building thermal models in data-limited settings.

Kurzfassung

Die zunehmende Integration erneuerbarer Energien stellt neue Anforderungen an Energiesysteme. Insbesondere gewinnt die zeitliche Flexibilität des Energieverbrauchs an Bedeutung. Gebäude können hierzu beitragen, indem ihre thermische Trägheit gezielt zur Lastverschiebung genutzt wird. Diese Lastverschiebung wird durch den thermischen Komfort der Bewohner begrenzt und erfordert daher genaue und physikalisch konsistente Vorhersagen der Innenraumtemperatur, um zu bewerten, ob mögliche Lastverschiebungen die Komfortgrenzen einhalten. Einen Ansatz zur physikalisch konsistenten Temperaturvorhersage bildet die Parameteridentifikation thermischer Gebäudemodelle. Dabei werden die vorherrschenden Temperaturdynamiken durch ein vereinfachtes thermisches Modell beschrieben, dessen Parameter im Rahmen der Parameterschätzung aus Daten gelernt werden. Aufgrund der eingeschränkten Betriebsbedingungen in Gebäuden werden Temperaturdynamiken jedoch nur schwach angeregt, während sich wesentliche Einflussgrößen zeitlich überlagern. Dadurch wird es schwierig, verlässliche kausale Zusammenhänge zu identifizieren. Dies stellt eine zentrale Herausforderung für die Parameterschätzung dar und macht den Identifikationsprozess selbst zum limitierenden Faktor für robuste Vorhersagen der Innentemperatur. In dieser Arbeit zeigen wir, dass die Parameterschätzung mithilfe eines inversen physikinformierten neuronalen Netwerks (PINN), welches thermische Parameter und kontinuierliche Temperaturverläufe unter physikalischen Nebenbedingungen gemeinsam schätzt, die Vorhersagegenauigkeit über alle betrachteten Vorhersagehorizonte verbessert und die Variabilität der Vorhersagegenauigkeit zwischen thermischen Modellen reduziert, die auf unterschiedlichen Trainingsdatensätzen identifiziert wurden. Darüber hinaus zeigen wir, dass dieser Ansatz eine vergleichbare Dateneffizienz wie ein gradientenbasiertes Verfahren zur Parameteridentifikation erreicht. Durch die Entkopplung der Temperaturverlaufsapproximation von der Parameterschätzung reagiert das inverse PINN weniger sensitiv auf die Initialisierung der Parameter und ist robuster gegenüber Variationen in den verfügbaren Trainingsdaten. Die Ergebnisse dieser Arbeit legen nahe, dass inverse PINNs eine vielversprechende Methode zur Identifikation der Parameter thermischer Gebäudemodelle in datenlimitierten Szenarien darstellen.

Contents

1 Introduction

The increasing integration of intermittent renewable energy sources is transforming power systems from centrally controlled generation that follows electricity demand to decentralized generation characterized by variable supply [1]. Ensuring grid stability therefore requires greater flexibility on the demand side, allowing electricity consumption to adapt to fluctuations in renewable energy generation. One effective approach to providing this flexibility is to shift the electricity demand of high-consuming devices, such as heat pumps, to periods with high renewable energy availability. This way it is possible to adapt to the fluctuations of renewable energy generation and alleviate shortages, smoothing the mismatch between production and consumption [2, 3].

Buildings represent one of the largest energy consumers globally, accounting for approximately 30% of global energy consumption. In Europe 63% of the residential buildings' energy consumption is used for space heating and cooling [4]. Additionally their inherent ability to store thermal energy allows for the temporal shifting of heating and cooling demand away from peak grid periods and towards times of high renewable energy availability making residential buildings an import part in increasing the demand side flexibility [5, 6].

One way to leverage this flexibility is to shift heating or cooling demand toward periods with high renewable energy availability. This naturally leads to an optimization task in which thermal power trajectories are selected to maximize the alignment between electricity demand for space conditioning and high renewable energy availability while ensuring that indoor temperatures remain within predefined comfort limits. Other common objectives include reducing energy consumption through foresighted space heating and cooling [7], lowering the carbon intensity of electricity consumption [8], and improving grid stability [5]. An example of such a temporal redistribution of thermal energy consumption is illustrated in Figure 1.1.

To respect the constraints (e.g. occupant comfort levels) the optimization requires information about the indoor temperature trajectory induced by a candidate solution [9]. To obtain this information, an exact representation of the building's thermal characteristics could be constructed and used to simulate the temperature dynamics forward in time. However, this approach is highly time-consuming, as it typically requires incorporating detailed information about the building's structure and extensive manual calibration until the simulated temperature dynamics resemble the observed behavior [10].

Another approach is to learn a less complex but representative thermal model from data that captures only the dominant temperature dynamics. Such reduced-order thermal models typically require less time to build, do not need manual calibration and thus are a significant less time consuming way of constructing a thermal model of the buildings' temperature dynamics [11]. However, the challenge of learning a thermal model from

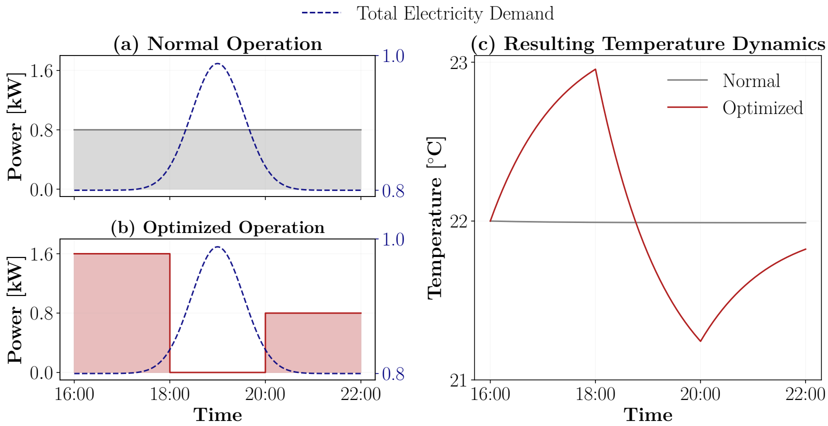


Figure 1.1: Illustrative example of how thermal demand can be shifted away from peak electricity demand periods while maintaining indoor temperatures between $21-23$ °C. **(a)** The normal thermal power levels in gray, while the relative electricity demand (normalized to the daily maximum) is visualized in blue. **(b)** The optimized thermal power supplied to the building. **(c)** The resulting temperature dynamics induced by the different operating strategies from **(a)** & **(b)**.

observational data lies in the nature of the typically available building data. This data shows little variation of the indoor temperature and high cross-correlations among influencing factors [12]. This makes it difficult for purely data-driven methods to find the right dependencies that lead to predictions that generalize across weather conditions and other influencing factors [13].

To solve this, there has been a lot of research on informing data-driven models about the underlying physics of the observed dynamical system. This additional information decreases the number of possible solutions, helping the model identify the correct dependencies. Incorporating prior physical knowledge therefore guides the model toward physically consistent solutions, improving generalization in data-limited settings [14, 15].

Recent work on physics-informed and physics-constrained neural networks for building temperature predictions has focused primarily on improving prediction accuracy while enforcing physical consistency. For example, physically consistent neural networks developed by Di Natale et al. [13] constrain directional sensitivities between heating power and indoor temperature, ensuring that heating has a non-negative influence on future temperatures, but do not enforce the governing heat transfer equations nor guarantee identifiability of physical parameters. Similarly, physics-constrained deep learning approaches, developed by Drgoňa et al. [16] embed structural priors into neural architectures to improve long-horizon prediction performance, while explicitly accepting non-uniqueness of learned parameters. In contrast, classical parameter identification methods yield interpretable thermal models and physically consistent dynamics but suffer from numerically fragile identification procedures [17].

This highlights a gap between predictive deep learning models, which sacrifice physical interpretability of their representation for predictive accuracy, and classical parameter identification methods, which preserve physical consistency but lack robustness during their training process.

In this work, we address this gap by formulating an inverse problem for identifying the parameters of a reduced-order thermal model using an inverse physics-informed neural network (PINN).

During training, the neural network learns a smooth functional approximation of the temperature trajectory while simultaneously adjusting the parameters of the governing heat transfer equations. After identification, the neural network is discarded and predictions are obtained by numerically integrating the identified thermal model.

Although reduced-order models may not achieve highly accurate single-step predictions, they provide stable long-horizon forecasts and preserve a physically consistent representation of the building dynamics. This physical consistency is crucial for flexibility optimization, as the prediction model must respond in a reliable and causal manner to changes in heating and cooling inputs [13].

Compared with existing approaches, the proposed method aims to improve the robustness of the identification procedure while maintaining consistency with the governing physical laws. To evaluate the robustness of the identification procedure, we compare the proposed approach with a classical gradient-based parameter identification method based on a differentiable physics model. In addition to analyzing the identification process, we evaluate the predictive performance of the learned thermal model with respect to different forecasting horizons and data efficiency.

This leads to the following research questions:

1. How does the predictive accuracy of the PINN-based approach compare to that of a classical differentiable physics model in data-limited settings?

2. How does the variability in predictive accuracy of the identified thermal models on different training data compare between the PINN-based approach and a classical differentiable physics model?

3. How do PINN-based and differentiable physics approaches compare in terms of data efficiency?

To address these research questions, we use simulated data generated with the physics-based building simulation software IDA ICE [18]. The obtained dataset consists of indoor temperature measurements, applied thermal power, and weather-related inputs like temperature and solar irradiance.

Predictive accuracy is evaluated by comparing the models under data-limited conditions, using standard error metrics such as the mean absolute error (MAE) on held-out test data. The variability in predictive performance is assessed by analyzing the distribution of prediction errors obtained from models trained on datasets sampled from different periods of the year. Data efficiency is evaluated by comparing model performance under different amounts of training data. These properties are analyzed on unseen test data.

The contributions of this work are summarized as follows:

- We propose an inverse PINN formulation for the identification of reduced-order building thermal models from observational data.
- We perform a systematic comparison between the proposed PINN-based identification approach and a classical differentiable physics model with respect to the predictive accuracy on extended forecasting horizons, variability of the predictive performance across different training data and data efficiency.

To support reproducibility of the presented results, the complete source code used for model training and evaluation is publicly available at `https://github.com/noahongit/robust-thermal-parameter-identification`.

This thesis is further organized as follows. Chapter 2 provides background on the optimization task, forecasting dynamical systems using data-driven learning approaches, physics-informed machine learning, and solving inverse problems using PINNs. Chapter 3 reviews related work on state-of-the-art approaches for building temperature predictions. Chapter 4 formulates the problem statement and research questions, with particular emphasis on the optimization task and our approach of learning the thermal characteristics of buildings from data. Chapter 5 describes the methodological framework adopted in this work, including the identification of a reduced-order building thermal model and the use of PINNs for identifying the parameters of the governing heat transfer equations. In Chapter 6, the proposed approach is evaluated using a simulated building. In particular, predictive accuracy, robustness of the identification procedure, and data efficiency are analyzed under data-limited conditions and compared with the differentiable physics model. Chapter 7 discusses the results, focusing on improvements in predictive accuracy, sensitivity to the training data, and data efficiency of the identification process. Finally, Chapter 8 summarizes the main findings of this thesis and outlines directions for future research.

2 Background

In this chapter, we provide the background to the model-based optimization task, the difficulties of using purely data-driven models to predict physically governed dynamical systems, and how physics-informed approaches can improve predictive performance in this setting. First, we will introduce the model-based optimization. We will discuss how the prediction model improves the potential of the optimization. We will explain the physical principles underlying building thermal models and go over how they simplify the complex temperature dynamics observed in buildings. Finally, we will address how physical knowledge can be integrated into a machine learning framework to improve its prediction accuracy in physically governed dynamical systems.

2.1 Model-Based Optimization

The integration of highly variable renewable energy sources imposes significant challenges to the stability of the electricity grid. One approach to reduce the mismatch between electricity production and consumption is to use the thermal mass of buildings as demand-side flexibility [5]. To this end, buildings are preheated or precooled during periods of high renewable generation, while during periods of low renewable generation, the stored thermal energy sustains the indoor temperature, allowing the building to maintain comfort levels without active heating or cooling [5]. In line with such approaches, our optimization focuses on the timing and amount of electricity used to preheat or precool buildings so that sufficient thermal energy is stored in the building's thermal mass to maintain indoor comfort levels.

Unfortunately, current temperature control strategies primarily rely on rule-based decisions using only current conditions to determine the best action [11]. This reduces the potential of flexibility optimization, as shifting energy demand requires actions that are taken in anticipation of future conditions rather than reacting to the present conditions. Therefore, to improve the potential of flexibility optimization, the control strategy must incorporate indoor temperature forecasts resulting from candidate thermal energy shifts [5, 11].

In Figure 2.1, we compare a normal and foresighted operation regarding their ability to shift thermal energy demand away from periods of high electricity demand. While normal operation turns heating off as electricity demand rises and resumes heating when the lower comfort level (21 °C) is reached, foresighted operation incorporates forecasts of electricity demand and indoor temperature into its decision-making. By leveraging this information, the foresighted operation is able to preheat the building during periods of low electricity demand, thereby reducing thermal energy demand during periods of high electricity demand.

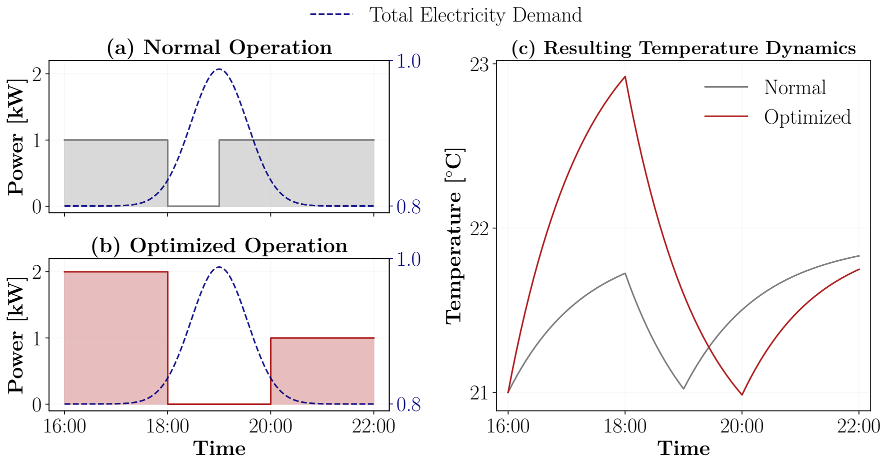


Figure 2.1: Flexibility optimization with respect to the electricity demand and additional predefined comfort constraints of 21 – 23 °C.**(a)** The normal thermal power levels in gray, while the relative electricity demand (normalized by the daily maximum) is visualized in blue. **(b)** The optimized thermal power supplied to the building. **(c)** The resulting temperature dynamics induced by the different operating strategies from **(a)** & **(b)**.

To incorporate temperature predictions into the optimization objective, an accurate representation of the building's thermal dynamics is required [11]. Since every building behaves differently under applied heating or cooling and to influences like outdoor temperature and solar irradiance, the building's thermal dynamics have to be learned from data. Hence, by observing past responses of the building's temperature to control inputs and external influences, a model learns to predict future temperature dynamics. This yields a queryable representation of the buildings, used as the prediction model for the optimization task [13].

2.2 Learning Temperature Dynamics from Data

Deep Learning models produce remarkable results in fields such as natural language processing and image recognition [19–21]. Given their data-driven optimization, they are able to learn the important dependencies for accurate predictions from data and therefore need very little knowledge of the underlying dynamics. In the case of learning temperature dynamics, these approaches do not need any prior information about the building's size or structure. In fact, to achieve highly accurate results, these models build this knowledge through the information contained in the training data. Thus, failing to provide enough information about the underlying system dynamics under different operating conditions or weather inputs, leaves these data-driven models vulnerable to high prediction errors for unseen inputs [22].

Building data come with little information on the observed temperature dynamics. Although there is a lot of data gathered through thermostats and internet of things (IoT) devices, the controlled environment in which the temperature unfolds and the cross-correlation between influencing factors makes it hard for neural networks to learn a reliable representation of the buildings' thermal dynamics that generalizes to unseen operating conditions [12]. As such, building data highlights two challenges for learning thermal models using standard neural networks. First, the data is often limited, either in terms of the available quantity or in terms of the information content about the underlying dynamics. Second, vanilla neural networks are not constrained to learn physically plausible representations. In fact, their optimization procedure is only constrained by the loss function, allowing them to learn any representation that minimizes the given loss [13, 23].

In particular, this means that, given the information-limited scenario in buildings, there can be many different representations (i.e., different weights and biases) that fit the training data equally well. Consequently, all representations achieve the same training loss, but most of them will fail to predict physically consistent temperature trajectories. Intuitively, they fail to generalize reliably because the learned dependencies might only make sense in the limited training data. As a matter of fact, the low information density provided by building data might not be sufficient to constrain the underspecified training procedure of neural networks.

The following example demonstrates this effect. To this end, we introduce a simplified dynamical system representing the indoor temperature under a single influence, the outdoor temperature. Hence, the indoor temperature evolution is solely driven by the outdoor temperature changes. This system can be written as:

$$\frac{dT_{\mathrm{in}}}{dt} = \frac{T_{\mathrm{amb}} - T_{\mathrm{in}}}{C_{\mathrm{in}} R_{\mathrm{total}}} \tag{2.1}$$

To reflect the limited information available in building data, we deliberately sub-sampled the dynamical system to induce ambiguity in the solution. Consequently, we increase the possible trajectories, hence the possible solutions to be found by the neural network, that fit the data.

To learn this simplified dynamical system, we use a fully connected neural network with hyperbolic tangent activation functions. It consists of one hidden layer with 32 neurons and a linear output layer. The network takes time t and outdoor temperature T_{amb} as inputs. The network parameters are initialized using PyTorch's default Kaiming uniform initialization. To account for potential sensitivity to initialization, each experiment is repeated seven times with different random seeds, resulting in different initial weights and biases and thus different starting points in the loss landscape.

Figure 2.2 shows the solutions obtained by the vanilla neural network. Although the network learns solutions that fit the data, the solutions are not consistent with the true dynamics. This indicates that accurate data fitting alone is insufficient when the training data does not fully capture the underlying dynamics.

The prediction model must therefore be constrained to respond to inputs in a physically consistent way, ensuring that its behavior aligns with the governing physics beyond the observed data.

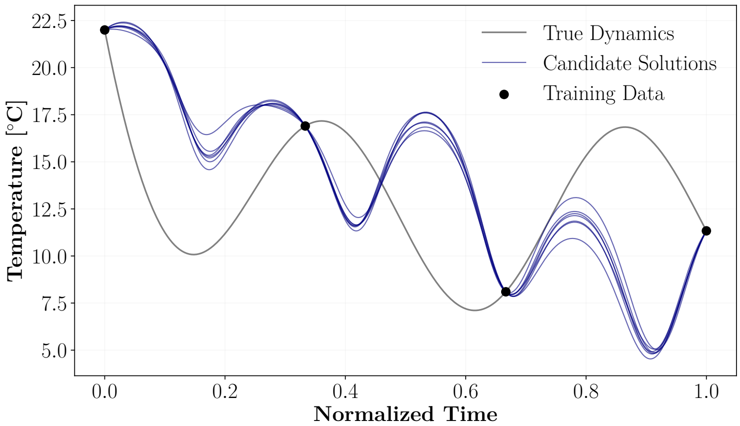


Figure 2.2: Different temperature trajectories found by a fully connected neural network through altering the initialization of the weights and biases between training runs.

However, building data may not provide sufficient information to identify a physically consistent model. To address this limitation, prior physical knowledge must be integrated to compensate for the limited information content in the data. This can be achieved by embedding energy conservation laws into the training procedure [13, 16]. By doing so, the set of admissible solutions is restricted to those that both fit the data and satisfy the governing physical principles [23].

2.3 Building Thermal Dynamics

Embedding physical knowledge of building thermal dynamics into the training process improves the physical consistency of the predictions [23]. Incorporating physical knowledge requires expressing temperature dynamics in terms of fundamental heat transfer mechanisms and enforcing energy conservation constraints.

Heat transfer is the process by which thermal energy moves from a region of higher temperature to a region of lower temperature. A net heat transfer into or out of a system results in a change in its temperature, depending on its thermal capacity. Energy exchange between a building and its surroundings occurs through three fundamental forms of heat transfer: conduction, convection, and radiation [24].

Conduction describes the energy transfer driven through particle interaction. Thereby the energy flows between adjacent particles. It is determined by the conductivity of the material, the energy difference between two surfaces, and the distance between these surfaces. The relationship is described by Fourier's law of heat conduction and is formalized as follows.

$$\dot{Q}_{cond} = kA_s \frac{(T_{\mathrm{s1}} - T_{\mathrm{s2}})}{\Delta x} \qquad [\mathrm{W}] \tag{2.2}$$

In Equation (2.2), k stands for the thermal conductivity of the material, A_{s} defines the area of the surface, and the term $\frac{(T_{\mathrm{s1}}-T_{\mathrm{s2}})}{\Delta x}$ is the temperature difference scaled by the distance Δx between the two surfaces [24].

Convection describes the energy transfer between a surface and a fluid. The energy transfer depends on the temperature difference of the surface and the fluid, as well as the area of the surface. It can be expressed by Newtons's law of cooling.

$$\dot{Q}_{\mathrm{conv}} = h_{\mathrm{conv}} A_s (T_s - T_{\mathrm{surr}}) \qquad [\mathrm{W}]$$

Here, h_{conv} denotes the convective heat transfer coefficient, while A_s and T_s represent the surface area and surface temperature, respectively. The variable T_{surr} denotes the temperature of the surrounding fluid [24].

Radiative heat transfer takes place through electromagnetic waves. Every object with a temperature above absolute zero emits thermal energy through radiation. The radiative energy transfer is described by Stefan-Boltzmann law

$$\dot{Q}_{\mathrm{rad}} = \epsilon \sigma A_s (T_s^4 - T_{\mathrm{surr}}^4), \qquad [\mathrm{W}] \tag{2.3}$$

where ϵ describes the emissivity of the object relative to that of the maximum possible radiation, σ is the Stefan-Boltzmann constant, and T_s and T_{surr} represent the temperatures of the surface and the temperature of its surroundings, respectively [24].

Together, these processes determine the temporal evolution of the indoor temperature and form the physical basis for the main temperature dynamics in buildings.

As discussed temperature changes inside the building are the result from energy flowing from high energy areas towards low energy areas. Energy flowing into the building will increase the temperature, while energy flowing out of the building will decrease its temperature. Using the energy conservation laws, we can now write the heat balance equation:

$$\frac{dE_{\mathrm{B}}}{dt} = \dot{E}_{\mathrm{in}} - \dot{E}_{\mathrm{out}} \qquad [\mathrm{W}] \tag{2.4}$$

$\dot{E}_{\mathrm{in}}$ describes the rate of energy flowing into the building. $\dot{E}_{\mathrm{out}}$ describes the rate of energy flowing out of the building. Both taken together define how the building's energy changes with time $\frac{dE_{\mathrm{B}}}{dt}$. Since the total energy of the building is proportional to its temperature ($E_{\mathrm{B}} = C_{\mathrm{B}} T_{\mathrm{B}}$) we can rewrite equation (2.4) as

$$C_{\mathrm{B}} \frac{dT_{\mathrm{B}}}{dt} = \dot{E}_{\mathrm{in}} - \dot{E}_{\mathrm{out}}, \qquad [\mathrm{W}]$$

where C_{B} is the heat capacity of the building.

Evaluating $\dot{E}_{\mathrm{in}}$ and $\dot{E}_{\mathrm{out}}$ for a specific building requires detailed information about its structure. Since the required level of detail is often unavailable in practice, approximations and simplifying assumptions are introduced to transform the complex thermal dynamics into a reduced-order system that retains the ability to capture the dominant temperature dynamics of the building [25].

2.3.1 Reduced-Order Models

Reduced-order models are a class of models that approximate the temperature dynamics seen in buildings, using a simple approximation of a more complex dynamical system. Hence, by using reduced-order models we are able to construct a simple model of the building that is able to capture the dominant temperature dynamics. It is obtained by introducing simplifying assumptions that reduce the underlying physical complexity.

First, the mean radiant temperature of the interior surfaces is assumed to be approximately equal to the indoor air temperature ($T_\mathrm{s} \approx T_\mathrm{surr} \approx T_\mathrm{in}$), which, under small temperature differences, permits linearization of the nonlinear radiative heat transfer. This linearization is achieved by rewriting the difference of T_s and T_surr from Equation (2.3):

$$T_s^4 - T_\mathrm{surr}^4 = (T_s - T_\mathrm{surr})\left(T_s^3 + T_s^2 T_\mathrm{surr} + T_s T_\mathrm{surr}^2 + T_\mathrm{surr}^3\right)$$

Using the assumption ($T_\mathrm{s} \approx T_\mathrm{surr} \approx T_\mathrm{in}$) we get

$$T_s^3 + T_s^2 T_\mathrm{surr} + T_s T_\mathrm{surr}^2 + T_\mathrm{surr}^3 \approx 4T_\mathrm{in}^3,$$

resulting in the linearized formulation of the radiative heat transfer

$$\dot{Q}_\mathrm{rad} \approx \varepsilon\sigma 4 T_\mathrm{in}^3 A_\mathrm{s}(T_\mathrm{s} - T_\mathrm{surr}).$$

Defining $h_\mathrm{rad} = 4\varepsilon\sigma T_\mathrm{in}^3$ yields:

$$\dot{Q}_\mathrm{rad} \approx h_\mathrm{rad} A_\mathrm{s}(T_\mathrm{s} - T_\mathrm{surr}). \qquad \mathrm{[W]}$$

To obtain a more compact representation, products of physical parameters are grouped and replaced by single effective heat transfer coefficients.

$$\begin{aligned}
\dot{Q}_\mathrm{conv} &= h_\mathrm{conv} A_\mathrm{s}(T_\mathrm{s} - T_\mathrm{surr}) \\
&= \frac{T_\mathrm{s} - T_\mathrm{surr}}{R_\mathrm{conv}} \\
\dot{Q}_\mathrm{rad} &= h_\mathrm{rad} A_\mathrm{s}(T_\mathrm{s} - T_\mathrm{surr}) \\
&= \frac{T_\mathrm{s} - T_\mathrm{surr}}{R_\mathrm{rad}} \\
\dot{Q}_\mathrm{cond} &= k A_s \frac{(T_\mathrm{s1} - T_\mathrm{s2})}{\Delta x} \\
&= \frac{T_\mathrm{s1} - T_\mathrm{s2}}{R_\mathrm{cond}}
\end{aligned} \tag{2.5}$$

Convective and radiative heat transfer act simultaneously at a surface and contribute additively to the net heat flux, such that $\dot{Q}_\mathrm{combined} = \dot{Q}_\mathrm{conv} + \dot{Q}_\mathrm{rad}$. After linearizing the radiative term, both mechanisms can be expressed using effective heat transfer coefficients and combined into a single linear heat transfer expression between the surface s and its surroundings.

$$\dot{Q}_{\text{combined}} = \left(\frac{1}{R_{\text{conv}}} + \frac{1}{R_{\text{rad}}}\right)(T_{\text{s}} - T_{\text{surr}}) = \frac{T_{\text{s}} - T_{\text{surr}}}{R_{\text{s,surr}}}. \tag{2.6}$$

Based on this physical description, a reduced-order model can be derived by capturing the principal heat transfer path: from the indoor air to the interior surface via combined radiative and convective exchange (2.6), through the wall by conduction (2.5), and from the exterior surface to the outdoor environment again via combined radiative and convective exchange (2.6). This process is visualized in Figure 2.3. This provides the foundation for the simplified models used to represent the physical processes underlying the observed temperature dynamics.

$$T_{\text{in}} \longrightarrow R_{\text{conv,in}} \parallel R_{\text{rad,in}} \longrightarrow R_{\text{cond}} \longrightarrow R_{\text{conv,out}} \parallel R_{\text{rad,out}} \longrightarrow T_{\text{amb}}$$

Figure 2.3: The heat transfer between indoor air and the environment. Convective and radiative heat transfer act in parallel at the surfaces, while conduction through the thermal mass acts in series.

The second simplifying assumption concerns the spatial representation of the building. To reduce the dimensionality of the thermal model, the spatially distributed thermal mass of surfaces and internal components is aggregated into a small number of lumped thermal nodes, which define the states of the reduced-order system. Within each node, the temperature is assumed to be spatially uniform, implying that internal temperature gradients and associated thermal resistances are negligible compared to the resistances between adjacent nodes. Under the simplifying assumption that the total thermal mass is lumped into a single indoor temperature node, no thermal capacitances are present between the indoor temperature node and the ambient temperature. Consequently, no energy can be stored within the intermediate layers, and the heat flux through all thermal resistances between the indoor and ambient temperatures is identical:

$$\dot{Q} = \frac{T_{\text{in}} - T_{\text{surf}_{\text{in}}}}{R_{\text{surf,in}}} = \frac{T_{\text{surf}_{\text{in}}} - T_{\text{surf}_{\text{amb}}}}{R_{\text{cond}}} = \frac{T_{\text{surf}_{\text{amb}}} - T_{\text{amb}}}{R_{\text{surf,amb}}}. \tag{2.7}$$

Rewriting the conductive heat transfer expression from Equation (2.7) using $T_{\text{surf}_{\text{in}}} =$

$T_{\text{in}} - \dot{Q}R_{\text{surf,in}}$ and $T_{\text{surf}_{\text{amb}}} = T_{\text{amb}} + \dot{Q}R_{\text{surf,amb}}$ yields:

$$\begin{aligned}
\dot{Q} &= \frac{1}{R_{\text{cond}}}(T_{\text{surf}_{\text{in}}} - T_{\text{surf}_{\text{amb}}}) \\
\dot{Q} &= \frac{1}{R_{\text{cond}}}(T_{\text{in}} - \dot{Q}R_{\text{surf,in}} - T_{\text{amb}} - \dot{Q}R_{\text{surf,amb}}) \\
\dot{Q}(R_{\text{cond}} + R_{\text{surf,in}} + R_{\text{surf,amb}}) &= T_{\text{in}} - T_{\text{amb}} \\
\dot{Q}R_{\text{total}} &= T_{\text{in}} - T_{\text{amb}} \\
\dot{Q} &= \frac{T_{\text{in}} - T_{\text{amb}}}{R_{\text{total}}}. \qquad (2.8)
\end{aligned}$$

This describes the simplified heat transfer from the indoor air temperature node to the ambient air temperature [24]. Using Equation (2.8) we can write a first-order thermal model, approximating the building by a single thermal node which stores and exchanges heat with its surroundings.

$$C_{\text{in}}\frac{dT_{\text{in}}}{dt} = \frac{T_{\text{amb}} - T_{\text{in}}}{R_{\text{total}}}$$

A second order thermal model adds a capacity to the thermal mass. This way the temperature evolution of the indoor air is influenced by the temperature of the thermal mass and wise-versa. Now, instead of having two parameters describing the temperature evolution of the whole building we double the amount of parameters, and get:

$$C_{\text{in}}\frac{dT_{\text{in}}}{dt} = \frac{T_{\text{mass}} - T_{\text{in}}}{R_{\text{im}}} + \frac{T_{\text{amb}} - T_{\text{in}}}{R_{\text{ia}}}, \qquad (2.9)$$

$$C_{\text{mass}}\frac{dT_{\text{mass}}}{dt} = \frac{T_{\text{in}} - T_{\text{mass}}}{R_{\text{im}}}. \qquad (2.10)$$

Equations (2.9) and (2.10) are approximations of the true thermal dynamics of the building. In their current form, they model the natural response of the building to the ambient temperature. However, the building temperature also depends on additional external influences, as well as space heating and cooling. These effects must therefore be included in the model, yielding a variant of the 2R2C model [26]:

$$C_{\text{in}}\frac{dT_{\text{in}}}{dt} = \frac{T_{\text{mass}} - T_{\text{in}}}{R_{\text{im}}} + \frac{T_{\text{amb}} - T_{\text{in}}}{R_{\text{ia}}} + P_{\text{tp}} + P_{\text{sun}} \qquad (2.11)$$

$$C_{\text{mass}}\frac{dT_{\text{mass}}}{dt} = \frac{T_{\text{in}} - T_{\text{mass}}}{R_{\text{im}}} \qquad (2.12)$$

where $P_{\text{tp}} = P_{\text{heat}} - P_{\text{cool}}$ is the net thermal power supplied to the building, and P_{sun} is the thermal power due to solar irradiation.

Given the linearity of Equations (2.11) and (2.12), the system can be expressed in a linear continuous-time state-space representation. The states T_{in} and T_{mass} are stacked into the state vector $\mathbf{x}$, while the inputs P_{tp}, P_{sun}, and T_{amb} are collected in the input vector $\mathbf{u}$.

$$\mathbf{x} = \begin{bmatrix} T_{\text{in}} \\ T_{\text{mass}} \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} P_{\text{tp}} \\ P_{\text{sun}} \\ T_{\text{amb}} \end{bmatrix}$$

$$\dot{\mathbf{x}} = \mathbf{A}\,\mathbf{x} + \mathbf{B}\,\mathbf{u}$$
$$y = \mathbf{C}\,\mathbf{x}$$

As the states are integrated forward in time the state vector $\mathbf{x}$ tracks the temperatures T_{in} and T_{mass}. The matrix $\mathbf{C}$ is called the observability matrix and is multiplied with the state vector to return the observed temperature y. This representation improves readability, enables MPC integration, and provides an predefined algorithm for integrating these differential equations forward in time [11].

To solve this system of ordinary differential equations we split the dynamics into the natural response and the forced response. We get the natural response by the matrix exponential.

$$\mathbf{x}_{\text{nat}}(t) = e^{\mathbf{A}t}\mathbf{x}_0, \qquad e^{\mathbf{A}t} = \sum_{k=0}^{\infty} \frac{(\mathbf{A}t)^k}{k!}$$

Then we calculate the system under the forced response by convolving the natural response with the forcings.

$$\mathbf{x}_{\text{forc}}(t) = \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau$$

Combining both parts yields the solution of the system of differential equations.

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau$$

Enabled by the linear structure of the thermal model, the indoor temperature dynamics admit a closed-form analytical solution. As shown, the state trajectory can be decomposed into a natural response, which captures the intrinsic thermal behavior of the building, and a forced response, which accounts for the influence of external weather and space conditioning inputs. This closed-form solution provides accurate forward simulation and thus enables effective model-based optimization.

2.4 Physics-Informed Learning Approaches

Machine learning methods, and in particular neural networks, have achieved remarkable performance across a wide range of applications. However, the limited information inherent in building data makes purely data-driven methods struggle to find a physically consistent representation of the temperature dynamics [13].

Physics-informed machine learning is an approach to mitigate these issues seen in data-limited, but physically well defined, systems. To reduce the number of possible solutions the neural network is informed about the underlying physics, effectively guiding the training procedure towards physically plausible solutions [14]. Depending on how the physics are incorporated, physics-informed approaches are either soft-constrained or hard-constrained.

In soft-constrained formulations, as used in PINNs, physical laws are enforced through additional loss terms that penalize violations of the governing physics. Consequently, the training procedure is guided towards solutions that obey these physics. Although the network is guided towards physically consistent solutions, inconsistent predictions can still occur in data-limited settings [13].

In hard-constrained formulations, the model structure itself constrains the training procedure to obey the underlying dynamics [12]. Specifically, the set of admissible solutions is given by the set of possible trajectories found by integrating the governing differential equations. Thus, the candidate trajectories differ through changes in the set of parameters used in these equations. This leads to an entirely different training procedure, namely parameter identification. As the name tells, this procedure is estimating parameters of differential equations instead of the weights and biases of neural networks, to fit the integrated trajectory to the observed temperature dynamics [12].

Overall, the key distinction between physics-informed machine learning approaches lies in whether the governing differential equations act as a regularizer to find physical plausible solutions or as the model itself.

2.4.1 Physics-Informed Neural Networks

PINNs have gained attention throughout the last years because they effectively integrate prior physical knowledge into the training process [27]. The idea is that the network is guided towards solutions, that obey the underlying physics and thus improve their generalization to unseen inputs [14].

A PINN can be interpreted as a neural network trained with physics-informed loss functions derived from governing physical equations. In contrast, a standard neural network learns an unconstrained mapping $\mathcal{N}_{\boldsymbol{\theta}}$ from an input space $\mathcal{X}$ to an output space $\mathcal{Y}$, given by

$$\mathcal{N}_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}.$$

In our case, we integrate physical knowledge by introducing a reduced-order thermal model to the training procedure. We consider a general thermal model of the form:

$$\dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big), \qquad \mathbf{x}(0) = \mathbf{x}_0, \qquad t \in [0, T]$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ denotes the thermal state vector, and $\mathbf{u}(t) \in \mathbb{R}^m$ denotes the known influences.

In contrast to the unconstrained approach of a standard neural network, a PINN is guided towards physically plausible solutions. Its goal is to find a specific approximation of the state trajectory $\mathcal{N}(t;\, \boldsymbol{\theta}) \approx \mathbf{x}(t)$ that additionally satisfies:

$$\dot{\mathcal{N}}(t;\, \boldsymbol{\theta}) \approx \dot{\mathbf{x}}(t) \tag{2.13}$$

where $\dot{\mathcal{N}}(t;\, \boldsymbol{\theta})$ denotes the derivative of the network's output with respect to time, and $\dot{\mathbf{x}}(t)$ represents the time derivative of the thermal state vector $\mathbf{x}(t)$. To learn the network's weights and biases such that Equation (2.13) holds, we define the difference of the derivate of the network's prediction and the true derivative as the physics residual:

$$\mathbf{r}(t;\, \boldsymbol{\theta}) = \dot{\mathcal{N}}(t;\, \boldsymbol{\theta}) - \mathbf{f}\big(\mathcal{N}(t;\, \boldsymbol{\theta}), \mathbf{u}(t)\big)$$

where $\mathbf{f}(\mathcal{N}(t;\,\boldsymbol{\theta}), \mathbf{u}(t))$ denotes the right-hand side of the differential equation evaluated using the network's predictions. This residual measures the violation of the energy balance equations at time t, and thus guides the network towards physically plausible solutions [14, 27, 28].

To train the PINN we are looking at a composite loss function, balancing the data fidelity and physical consistency. Consequently, the loss function consists of a data loss:

$$\mathcal{L}_{\text{data}}(\boldsymbol{\theta}) = \frac{1}{N_d} \sum_{i=1}^{N_d} \|\mathbf{C}\mathcal{N}(t_i;\,\boldsymbol{\theta}) - y(t_i)\|^2 \tag{2.14}$$

where we compare the predicted indoor temperature $\mathbf{C}\mathcal{N}(t_i;\,\boldsymbol{\theta})$ with the observed temperature $y(t_i)$.

The physics loss is enforced on pre sampled points t^i_{physics} across the domain. These points are called the collocation or physics points. They differ from the neural network's input t_i at which we observed the target. The physics loss is evaluated at the collocation points, and is defined as:

$$\mathcal{L}_{\text{phys}}(\boldsymbol{\theta}) = \frac{1}{N_r} \sum_{i=1}^{N_r} \left\| \mathbf{r}(t^i_{\text{physics}};\,\boldsymbol{\theta}) \right\|^2 . \tag{2.15}$$

Pulling the data and physics loss together, defined in Equation (2.14) and Equation (2.15), the composite loss function of the PINN can be written as:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{data}}(\boldsymbol{\theta}) + \lambda_{\text{phys}}\,\mathcal{L}_{\text{phys}}(\boldsymbol{\theta})$$

where λ_{phys} controls the relative weight of the physics loss [29].

To demonstrate the improved training characteristics of PINNs, we revisit the experiment from Figure 2.2. We consider the same data-limited scenario using a PINN. As with the vanilla neural network, we employ a fully connected network with hyperbolic tangent activation functions. The network consists of one hidden layer with 32 neurons and a linear output layer. As in the previous experiment, the network parameters are initialized using PyTorch's default initialization. In contrast to the standard neural network, only time t is provided as input, while the indoor temperature T_{in} serves as the target. The outdoor temperature enters the PINN through the physics loss, defined in Equation (2.15). We provide the relationship between the indoor and the outdoor temperature using the thermal model, defined in Equation (2.1). As a results we inform the PINN about the physical plausible trajectories between the sampled data points, helping the training procedure to find solutions that fit the true physical behavior of the dynamical system.

In Figure 2.4 we can see that although we initialized the PINNs weights and biases differently, the training process yields similar solutions that are closer to the true dynamics. This indicates, that the provided regularization works as intended, shrinking the solution space to find solutions that are physically consistent [23].

In summary, PINNs encourage physically consistent solutions within the training data, even in data-limited settings. However, physical consistency on the training data

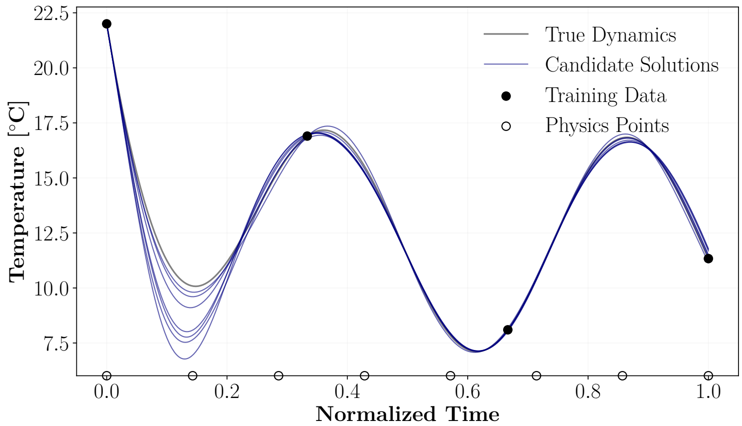


Figure 2.4: The learned mapping of a PINN in a data-limited setting. Its training objective does not only include the data loss using the sampled target T_{in}, it additionally includes samples of the outdoor temperature T_{amb} at the physics points t_{physics} to fill the information gaps between the sampled target. The candidate solutions are created by different initializations of its weights and biases.

does not guarantee physically consistent predictions under out-of-distribution inputs. The generalization limitations inherent to neural networks are not fully resolved by incorporating prior physical knowledge. In the context of building data, where excitation is limited, the training data are unlikely to capture the full range of system behavior [11, 12]. Consequently, the model may encounter unseen input conditions during prediction, potentially leading to unreliable forecasts.

2.4.2 Parameter Identification

While PINNs improve generalization by shrinking the space of possible solutions through physical priors, they do not eliminate out-of-distribution errors. Their performance still depends on the coverage and quality of training data, and they may produce unreliable or physically inconsistent predictions when evaluated on unseen data [30]. Given building data, which is often sparse in both coverage and information density, this limitation motivates models that remain robust to inputs outside the training distribution [12].

Parameter identification addresses this challenge by shifting the learning task. Instead of learning a direct input-output mapping, it estimates unknown parameters of a physically grounded approximation of the dynamical system. Specifically, the system is observed, its dynamics are described by differential equations, and the parameters of the differential equations are adjusted such that simulated trajectories match observed behavior [12]. Consequently, the model structure is fixed by the governing equations, and the remaining degrees of freedom are the parameters of these equations. This defines an inverse problem and reduces the learning approach to parameter estimation, thereby simplifying the training procedure [31].

The training procedure of parameter identification, although similar to that of standard machine learning approaches, differs in the way we calculate the forwards pass. Instead of going through linear transformations and non-linear activation functions to produce a prediction, we numerically integrate the differential equations to get a temperature trajectory in time, thereby producing a candidate solution. Having a candidate solution, we evaluate the error of the integrated trajectory and the observed temperatures. This error is then used to calculate the partial derivatives with respect to the parameters of the differential equations. Like with neural networks these partial derivatives enable parameter updates, that are used to decrease the error of the predictions.

In the following, we visualize the parameter identification process using a simple example. Specifically, we consider the estimation of the time constant $\tau = C_{\text{in}} R_{\text{total}}$ of a first-order building thermal model.

$$\frac{dT(t)}{dt} = \frac{1}{\tau}\left(T_{\text{amb}}(t) - T(t)\right) \tag{2.16}$$

For simplicity, we use the Euler method to integrate the temperature $T(t)$ forward in time. Given an initial temperature T_0 we can update the temperature as:

$$T_{k+1} = T_k + \Delta t f(\mathbf{x}(t_k), \mathbf{u}(t_k);\, \tau_k)$$

where f is the right hand side of our first-order thermal model and τ_k is the current parameter estimate. Using our first-order thermal model from Equation (2.16) the state update step can be written as:

$$T_{k+1} = T_k + \frac{\Delta t}{\tau_k}\left(T_{\text{amb},k} - T_k\right) \tag{2.17}$$

where T_{k+1} depends on the current temperature T_k and the current temperature difference to the outside temperature T_{amb}.

The error of the resulting trajectory can be written as the difference of the true $T_{\text{true},k}$ and the predicted $T_{\text{pred},k}$ temperature at step k. Using this difference we can define loss L as the mean-squared error given an estimate τ_k.

$$L(\tau_k) = \frac{1}{2N} \sum_{k=0}^{N-1} \left(T_{\text{pred},k}(\tau_k) - T_{\text{true},k}\right)^2 \tag{2.18}$$

Here we are summing up all residuals between the observed and predicted temperatures from the initial temperature T_0 to the last observed one T_{N-1}.

To improve the estimate τ, we need the partial derivative of the loss $L(\tau)$ defined in

Equation (2.18) with respect to τ.

$$\begin{aligned}\frac{\partial L(\tau)}{\partial \tau} &= \frac{1}{N}\sum_{k=0}^{N-1}\left(T_{\text{pred},k}(\tau_k) - T_{\text{true},k}\right)\cdot\left(\frac{\partial T_{\text{pred},k}(\tau)}{\partial \tau} - \underbrace{\frac{\partial T_{\text{true},k}}{\partial \tau}}_{=0}\right)\\ &= \frac{1}{N}\sum_{k=0}^{N-1}\left(T_{\text{pred},k}(\tau_k) - T_{\text{true},k}\right)\frac{\partial T_{\text{pred},k}(\tau)}{\partial \tau} \end{aligned} \tag{2.19}$$

Since a tiny change in τ_k will not only change T_k but also T_{k+1} we need to calculate the forward partial differential. This means that we are not only interested in $\frac{\partial T_k}{\partial \tau}$ but also in $\frac{\partial T_{k+1}}{\partial \tau}$.

$$\begin{aligned}\frac{\partial T_{k+1}}{\partial \tau} &= \frac{\partial}{\partial \tau}\left(T_k + \frac{\Delta t}{\tau}\left(T_{\text{amb},k} - T_k\right)\right)\\ &= \frac{\partial T_k}{\partial \tau} + \frac{\partial}{\partial \tau}\left(\frac{\Delta t}{\tau}\left(T_{\text{amb},k} - T_k\right)\right)\\ &= \frac{\partial T_k}{\partial \tau} - \frac{\Delta t}{\tau^2}\left(T_{\text{amb},k} - T_k\right) + \left(\frac{\Delta t}{\tau}\left(0 - \frac{\partial T_k}{\partial \tau}\right)\right)\\ &= \frac{\partial T_k}{\partial \tau} - \frac{\Delta t}{\tau^2}\left(T_{\text{amb},k} - T_k\right) - \frac{\Delta t}{\tau}\frac{\partial T_k}{\partial \tau}\\ &= \left(1 - \frac{\Delta t}{\tau}\right)\frac{\partial T_k}{\partial \tau} - \frac{\Delta t}{\tau^2}\left(T_{\text{amb},k} - T_k\right)\end{aligned} \tag{2.20}$$

Using the derivation from Equation (2.20) and defining the sensitivity s of the temperature T_k with respect to τ as $s_k = \frac{\partial T_k}{\partial \tau}$, we can write the update of the sensitivity s_{k+1} across the Euler steps k.

$$s_{k+1} = \left(1 - \frac{\Delta t}{\tau}\right)s_k - \frac{\Delta t}{\tau^2}\left(T_{\text{amb},k} - T_k\right)$$

Now we can write Equation (2.19) by integrating the sensitivity vector $\mathbf{s}$ across the Euler steps k to calculate the partial derivative of the loss with respect to τ as:

$$\frac{\partial L}{\partial \tau} = \frac{1}{N}\sum_{k=0}^{N-1}\left(T_{\text{pred},k}(\tau_k) - T_{\text{true},k}\right)\cdot s_k,$$

and the update of τ

$$\tau_{\text{new}} = \tau_{\text{old}} - \eta\frac{\partial L}{\partial \tau}, \tag{2.21}$$

where $\eta = 2$ is the learning rate.

As we can see in Figure 2.5 the parameter identification is well-posed. The loss landscape is guiding the parameter estimates directly to the true value τ. No local minima or vanishing gradient directions. Thus, the identification of a first-order thermal model is very stable and converges quickly. But a first-order model without the influencing factors of space conditioning and solar irradiance will likely fail to resemble the the observed

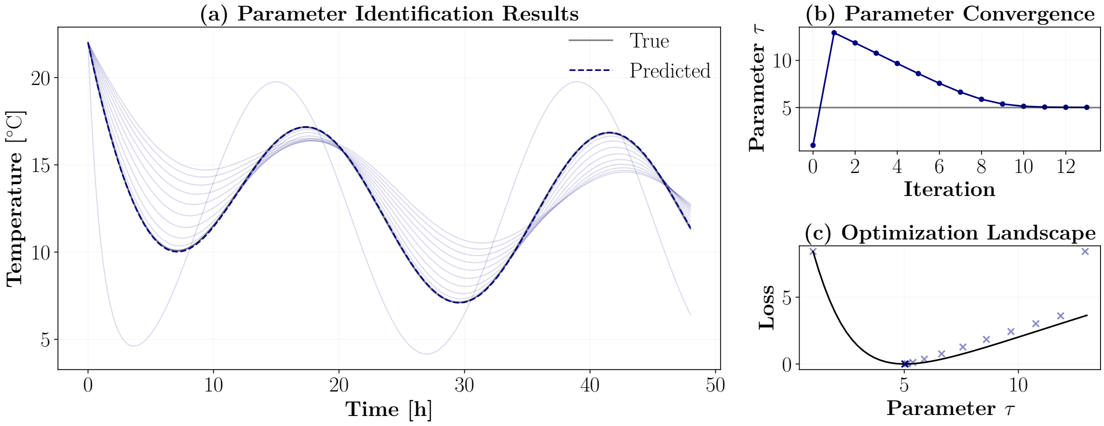


Figure 2.5: Estimation of τ during a parameter identification task mathematically derived through Equations (2.17) to (2.21). **(a)** shows the intermediate trajectories in light blue and the final predicted trajectory produced by the learned parameter τ. **(b)** shows the intermediate parameters in blue and the true parameter in grey. **(c)** shows the loss landscape in black and the loss given by intermediate parameter estimates in blue.

dynamics. Thus, the higher the order of a thermal model, the closer it will be able to resemble the observed dynamics, but the more ill-posed the identification process is going to be [17].

Although structural constraints reduce the space of admissible solutions, parameter identification can remain ill-posed when only limited data are available. In such cases, a unique and reliable set of parameters can not be inferred from the observations. With building data, this ill-posedness is present due to low variability in the measured system behavior, cross-correlations among influencing factors, and unobserved disturbances such as occupant behavior [12]. Hence, multiple parameter configurations may produce the same trajectory within the training data. However, these parameter sets can lead to completely different predictions when evaluated on unseen data.

Overall, parameter identification methods tend to reduce the generalization issues due to the provided structure of the differential equations, but the limited information about the dynamical system makes robust identification the key challenge for reliable predictions [17].

2.4.3 Inverse PINN

Classical parameter identification methods rely on repeated numerical integration of the governing differential equations and the discretization of the time domain. As the simulated trajectory is propagated forward in time, discretization errors and numerical approximation inaccuracies can accumulate, potentially leading to biased gradient estimates and unstable parameter updates. Additionally the limited information in building data, renders robust identification the key challenge of parameter identification methods [17].

Inverse PINNs provide an alternative formulation of the parameter identification. Rather than repeatedly integrating the differential equations to compute gradients with respect to the parameters, inverse PINNs learn a smooth functional approximation of the system trajectory that is regularized by the governing equations. In this setting, the parameters of the differential equations are optimized jointly with the neural network weights. Parameter identification therefore proceeds in parallel with the learning of the trajectory representation. Moreover, the continuous function approximation enables the governing differential equations to be enforced via automatic differentiation across the entire time domain, rather than through discrete-time approximations as in classical identification approaches.

From a conceptual perspective, inverse PINNs address the same parameter identification problem as classical methods. The governing equations, the observations, and the set of unknown parameters $\boldsymbol{\phi}$ remain unchanged. In contrast to standard PINNs, defined in Section 2.4.1, inverse PINNs treat the parameter vector $\boldsymbol{\phi}$ as an additional optimization variable, enabling gradients with respect to all unknown parameters. This leads to different formulation of the loss functions.

Conceptually, we are now looking at a state trajectory $\mathbf{x}(t)$ with unknown thermal characteristics $\boldsymbol{\phi}$.

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t);\, \boldsymbol{\phi}\big), \qquad \mathbf{x}(0) = \mathbf{x}_0, \qquad t \in [0, T]$$

Here $\boldsymbol{\phi} \in \mathbb{R}^d$ represents a d-dimensional vector holding the unknown parameters of the reduced-order thermal model.

As discussed the thermal parameters of the parameter vector $\boldsymbol{\phi}$ are learned in parallel to $\boldsymbol{\theta}$. This is done by adding it to the physics loss, enabling gradients with respect to the unknown parameters of the thermal model. This yields the new loss functions of the inverse PINN. While the data loss

$$\mathcal{L}_{\text{data}}(\boldsymbol{\theta}) = \frac{1}{N_d} \sum_{i=1}^{N_d} \|\mathbf{C}\mathcal{N}(t_i;\, \boldsymbol{\theta}) - y(t_i)\|^2$$

stays the same, the physics residual is parametrized by $\boldsymbol{\phi}$.

$$\mathbf{r}(t;\, \boldsymbol{\theta}, \boldsymbol{\phi}) = \dot{\mathcal{N}}(t; \boldsymbol{\theta}) - \mathbf{f}\big(\mathcal{N}(t;\, \boldsymbol{\theta}), \mathbf{u}(t);\, \boldsymbol{\phi}\big)$$

This yields the physics loss for the inverse PINN:

$$\mathcal{L}_{\text{phys}}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{1}{N_r} \sum_{i=1}^{N_r} \left\| \mathbf{r}(t_r^i; \boldsymbol{\theta}, \boldsymbol{\phi}) \right\|^2.$$

Like with the standard PINN the total loss of an inverse PINN is the sum of both losses:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathcal{L}_{\text{data}}(\boldsymbol{\theta}) + \lambda_{\text{phys}}\, \mathcal{L}_{\text{phys}}(\boldsymbol{\theta}, \boldsymbol{\phi}). \tag{2.22}$$

Minimizing the total loss defined in Equation (2.22) yields a data-driven approximation of the temperature dynamics regularized by a thermal model whose parameters are simultaneously optimized.

To visualize the training dynamics, we consider the estimation of the time constant $\tau = C_{\text{in}}R_{\text{total}}$ of a first-order building thermal model. Hence, we use the same dynamical system as in Figure 2.5, finding an approximation of the indoor temperature trajectory and an estimate of τ in the process. The training procedure of an inverse PINN is visualized in Figure 2.6.

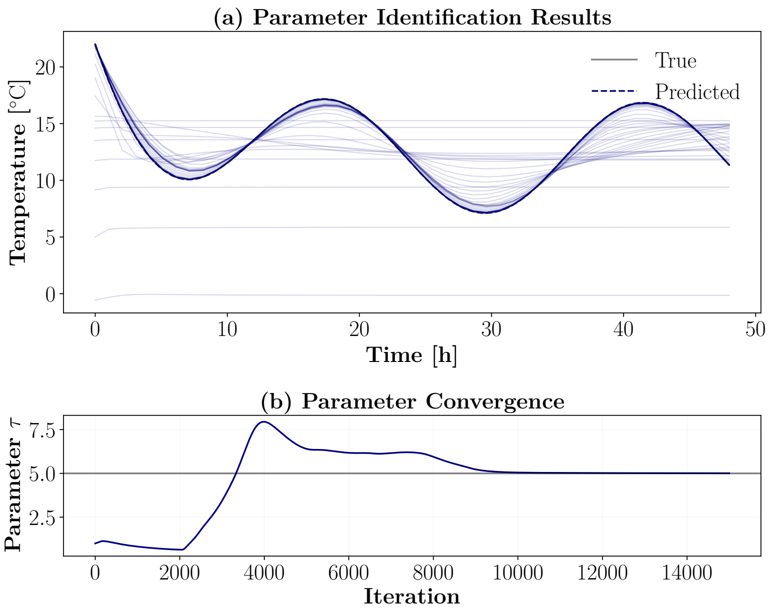


Figure 2.6: Estimation of τ by an inverse PINN minimizing the total loss defined in Equation (2.22). **(a)** shows the intermediate trajectories and the final predicted trajectory in light blue. **(b)** shows the intermediate parameters in blue and the true parameter in grey.

Inverse PINNs represent system trajectories as continuous neural network approximations and compute derivatives via automatic differentiation, thereby avoiding numerical discretization errors. Conceptually, inverse PINNs provide a flexible framework for parameter identification by combining the expressive power of neural networks with governing physical laws. This enables parameter identification from limited data while maintaining a continuous-time representation of the system dynamics [30].

3 Related Work

Developing building thermal models that are capable of forecasting physically consistent temperature dynamics is a challenging and actively researched problem [32–37]. Given the complexity of the dynamics, this research came up with many different approaches that differ substantially in terms of scalability, predictive accuracy, and interpretability. In the following, we review the most relevant modeling paradigms for model-based flexibility optimization and discuss their strengths and limitations with respect to learning control-oriented thermal models from observational data.

Broadly, the research directions can be clustered into three dominant approaches, namely black-box, grey-box, and white-box models. White-box models are complex simulation environments that generate highly interpretable temperature dynamics, but rely on detailed physical knowledge about the building and its structure. Due to the complexity of their simulation environments, these models typically do not support gradient-based calibration. As a result, the calibration of white-box models is computationally intensive and time-consuming, which limits their scalability across multiple buildings [22]. That's why we will focus on the gradient-based approaches, specifically black-box and grey-box models.

3.1 Black-Box Models

Black-box approaches learn predictive models of building thermal dynamics directly from observational data, without relying on an explicit physical description of the building. Instead, these models infer the relationship between influencing variables and indoor temperatures, from data.

Thus, black-box models can be developed rapidly and scaled across large numbers of buildings. This convenience has made black-box models a widely used approach for developing building thermal models [35].

3.1.1 Autoregressive Models

Given the strong influence of past weather conditions and Heating, Ventilation, and Air Conditioning (HVAC) operation on the current and future temperature dynamics, models capable of capturing long-term dependencies are particularly suitable for representing these complex relationships. Autoregressive models, for example, address this by predicting the indoor temperature based on past indoor temperatures and historical exogenous inputs, such as outdoor temperature, HVAC operation, and solar irradiance [38]. The classical ARX model uses a linear mapping from the current and past influences to the output, which makes it computationally inexpensive to estimate and easy to interpret. For this reason, linear ARX models are often used as a baseline when assessing how well the observed thermal dynamics can be captured from data. Non-linear extensions, like the non-linear autoregressive with exogenous inputs (NARX) model, retain the same autoregressive input-output structure but replace the linear mapping with a nonlinear function, for example a neural network [39]. Due to the efficient training of autoregressive models, their interpretability, and their compatibility with control-oriented formulations, these models are a valuable approach to learn building thermal models [40, 41].

In the literature, besides Krüger et al. [42], who used a specialized ARX model for indoor temperature predictions, these linear models are mainly used as a baseline. Given the complex temperature dynamics, the linearity of the ARX model is regarded as too simple to represent the complex non-linear input-output relationships observed in buildings. This is validated by several studies [38, 41, 43] demonstrating that ARX models are outperformed by the NARX model in general, but also specifically in building temperature dynamics, hinting the strong non-linear relationship of the indoor temperature to influencing factors like solar irradiance. Mustafaraj et al. [41] showed that the NARX is more suitable for indoor temperature predictions comparing it to the standard ARX model. Mechaqrane et al. [38] and Afroz et al. [43] trained an NARX model on historical indoor temperature data as well as other exogenous inputs such as weather and HVAC operation. While Mechaqrane et al. [38] compared primarily the performance between the ARX and the NARX approach, Afroz et al. [43] evaluated and optimized the NARX model for long-term temperature predictions and reports high predictive accuracy over longer prediction horizons. Thus, in general the NARX approach seems to improve the predictive accuracy compared to the linear variant and is highlighted as a valuable modeling approach to predict the indoor temperature in buildings.

3.1.2 Recurrent Neural Networks

NARX models use a finite window of past observations to predict the current state of a given system. An extension would be to replace this finite lag with a recurrent architecture. This change yields a new highly effective model class, being the recurrent neural networks (RNN). Instead of relying on a fixed set of past observations, RNNs enable the data-driven approach of retaining the right amount of past observations to explain the future changes [30]. Long short-term memory (LSTM) networks improve the optimization procedure of standard RNNs with gated memory, allowing for longer dependencies affecting the current state of the observed system [44]. This makes LSTMs a widely used approach for predicting indoor temperature changes, because the current temperature depends on both, recent inputs that have short-term effects, and accumulated effects over time. Given sufficient training data, these expressive networks are able to find the statistical correlations and dependencies that lead to robust forecasts of the indoor temperature.

Mtibaa et al. [45] trained an LSTM to predict indoor temperatures of buildings. They evaluated the performance of the LSTM approach and compared it to a standard Multi-Layer Perceptron (MLP) and NARX model. They showed that LSTMs outperform NARX models in terms of predictive accuracy and stability on a prediction horizon ranging from 30 minutes to 6 hours. Although they argue that in their study the LSTM approach achieves the highest predictive accuracy, they acknowledge that as the prediction horizon widens, the prediction accuracy degrades. This highlights the fact that LSTMs can struggle in closed-loop long-range prediction scenarios, where they have to use their past predictions for their next predictions, accumulating errors in the process.

Elmaz et al. [44] focuses on increasing the predictive accuracy of black-box models over long prediction horizons. To achieve this they used a convolutional neural network (CNN) for feature extraction. The extracted features are then provided to the LSTM. They report that the CNN-LSTM outperformed the other black-box variants for all evaluated prediction horizons, being 30 minutes, 60 minutes, and 120 minutes. They argue that the CNN-LSTM's performance gain comes from lower error accumulation when reusing the predictions. Although the CNN seemed to work as intended, they also acknowledge that their approach degrades heavily on long prediction horizons.

Li et al. [36] compared CNN, LSTM, and CNN-LSTM architectures on residential smart-thermostat data for multi-step indoor air-temperature prediction and reported the best performance for CNN-LSTM.

Overall, black-box models seem to be highly accurate for short-term predictions, but their predictive accuracy heavily degrades under closed-loop predictions and long prediction horizons.

Black-box models, LSTMs in particular, have gained attention because they can represent nonlinear thermal dynamics and often achieve strong predictive performance in building time-series forecasting [46]. However, the main challenge for flexibility optimization is not necessarily short-term accuracy, but long-term consistency. This means that while LSTMs are able to produce accurate short-term predictions, several studies show that their performance degrades heavily in settings, where the LSTM is fed its own predictions for extended prediction horizons. Additionally these approaches are often data-hungry, meaning that their expressivity comes with the need for high data availability. If there is not enough data, these models struggle to find robust dependencies that generalize to unseen operating conditions [12]. This challenge that comes with purely data-driven models aligns with several studies, suggesting that specifically building data does not provide enough information for these data-intensive models. Although they might achieve high short-term accuracy, the data leaves these models underspecified, resulting in poor generalization and physically inconsistent temperature predictions [13, 11, 47].

Especially with building data, the unconstrained way of learning a functional mapping of purely data-driven models comes with a cost. In data-limited settings, the information density is not high enough to shrink the solution space to make the learned mapping unique. Consequently, these models potentially have poor generalization capabilities outside the training data [13]. This motivates models that use prior knowledge as inductive bias, helping them to mitigate the effects of information-poor data.

3.2 Grey-Box Models

Compared with purely black-box models, grey-box models embed partial physical structure into the prediction model. In building thermal modeling, this usually means that the model structure is constrained to physically plausible behavior or is informed by simplified heat-transfer relations. Both approaches are desirable for control-oriented applications, where the learned model must remain accurate not only for short-term predictions but also under long-term closed-loop forecasting, in which prediction errors can accumulate over time [13].

In this thesis, we distinguish between two branches of grey-box models. The first branch covers physics-informed neural architectures, which retain a neural or other data-driven component but embed physical priors directly into the model structure or training objective. The second branch represents physics-based models, in which thermal resistances and capacitances provide a reduced-order representation of the building's temperature dynamics and are identified from data. Both branches aim to improve data efficiency and closed-loop prediction reliability compared with purely black-box approaches [16, 17, 48].

3.2.1 Physics-Informed Neural Architectures

Physics-informed neural architectures aim to improve the generalization of purely data-driven predictors by embedding physical knowledge directly into the learning objective. The main idea is to retain the expressive power of neural networks while constraining the learned dynamics to remain physically plausible outside the training data, thereby improving accuracy over long prediction horizons [13, 16].

A representative early example is the physics-constrained recurrent model of Drgoňa et al. [16], which embeds structural priors from traditional building models directly into a neural architecture and complements them with inequality constraints and a stability-oriented parameterization. A more rigorous step is taken by Di Natale et al. [13], who introduce the physically consistent neural network (PCNN) and formally proved physical consistency by design, even on unseen data. In their multi-zone extension, Natale et al. [22] show that this idea can be scaled while retaining state-of-the-art predictive accuracy among physically consistent models.

To the best of the authors' knowledge, Gokhale et al. [48] were among the first to apply an inverse physics-informed neural network approach to building thermal modeling. Their approach focused on next-step prediction of the thermal dynamics. In contrast to the architectural physics constraints proposed by Drgoňa et al. [16] and Di Natale et al. [13], Gokhale et al. [48] used conventional neural network architectures and imposed physical consistency through additional physics-informed loss terms. Furthermore, their approach employed a thermal model with unknown parameters that were jointly estimated during training. They showed that the resulting PINN improved data efficiency and long-horizon predictive performance compared with conventional neural networks.

Overall, physics-informed neural architectures have been successfully applied to building thermal modeling [13, 16, 48]. However, as we have already discussed, the optimization of neural networks tends to be underspecified, making them vulnerable in data-limited settings [13, 47].

3.2.2 Physics-Based Models

Although physics-informed neural models achieve high accuracy even on long prediction horizons, physics-based models are a well-established way to predict temperature dynamics in buildings [17]. The most dominant class of physics-based models are the RC models. By representing the main heat-transfer paths through thermal resistances and heat storage effects through thermal capacitances, they provide a reduced-order model that is physically interpretable, computationally efficient, and directly usable in MPC [31, 33]. This convenience makes RC models highly relevant whenever closed-loop prediction and computational tractability are central design requirements [31, 33, 34, 49].

The main challenge, however, is parameter identification. A recent critical review of low-order grey-box models identifies the initialization and final identification of model parameters as the most difficult and impactful part of the workflow [17]. In particular, the review notes that good initialization and stochastic search is often required to locate a plausible region of the parameter space before local refinement can succeed [17].

A related issue is the stability and identifiability of the estimated parameters. Marty-Jourjon et al. [50] show that RC models suffer from parameter correlations and convergence problems during calibration, such that a model may achieve high predictive accuracy, while using physically implausible parameter estimates. Rouchier et al. [51] further show that repeated training runs do not necessarily return stable parameter values. While stochastic calibration yields more consistent parameter estimates across separate training runs, deterministic calibration is less robust and can lead to less reliable forecasts. Moreover, Yu et al. [52] show that the identified parameters can be strongly influenced by the training data and, depending on preprocessing and model formulation, may even drift toward non-physical values.

In summary, the literature suggests that RC models are not limited primarily by their expressivity, but by the process of parameter identification [17, 50–52].

Although there has been significant progress in the area of physics-informed building modeling, the existing approaches either rely on structurally constrained neural architectures, physics-informed loss functions, or parameter identification. While the structurally constrained neural architectures trade interpretability for accuracy, parameter identification methods yield unstable results due to their challenging identification procedure [13, 17]. This motivates the physics-informed hybrid approach presented in this thesis. While we use the expressivity of the neural network to find functional approximations of temperature trajectories, we estimate the parameters of a reduced-order model and use the obtained thermal model for predictions.

4 Problem Statement & Research Questions

4.1 Problem Setting

Given the intermittent nature of renewable electricity generation, grid stability increasingly depends on the temporal alignment of electricity production and consumption. This alignment can be achieved by shifting electricity demand for space conditioning toward periods of high renewable energy availability. To this end, a control strategy determines thermal power input trajectories, pre-heating or pre-cooling the building, while ensuring that the indoor temperature remains within predefined comfort bounds.

This can be formalized as a constrained optimization problem. Let $P_{\mathrm{tp}}(t)$ denote the thermal power supplied to the building. The objective is to determine a control trajectory that aligns space-conditioning demand with renewable energy generation $P_{\mathrm{res}}(t)$ over a given time horizon $[0, T]$. This can be expressed as

$$\begin{aligned} \min_{P_{\mathrm{tp}}(t)} \quad & \int_0^T \left(P_{\mathrm{tp}}(t) - P_{\mathrm{res}}(t)\right)^2 \, dt \\ \text{s.t.} \quad & T_{\mathrm{in}}(t) \in [T_{\min}, T_{\max}], \\ & \dot{T}_{\mathrm{in}}(t) = f\left(P_{\mathrm{tp}}(t), P_{\mathrm{sun}}(t), T_{\mathrm{amb}}(t)\right), \\ & P_{\mathrm{tp}}(t) \in [0, P_{\max}], \end{aligned}$$

where $T_{\mathrm{in}}(t)$ denotes the indoor temperature, $P_{\mathrm{sun}}(t)$ the solar irradiance, and $T_{\mathrm{amb}}(t)$ the ambient temperature. The function $f(\cdot)$ describes the thermal dynamics of the building.

Maintaining indoor comfort requires a predictive model of the indoor temperature dynamics that is both accurate and physically consistent over the optimization horizon. Thus, the objective is to identify model parameters of a reduced-order thermal model that captures the building's main temperature dynamics, producing physically consistent indoor temperature predictions over extended prediction horizons. To address this we use a parameter identification procedure to identify a thermal model from observational data. The data is generated by a residential building whose thermal dynamics are governed by energy exchange between the indoor air, the building mass, and the environment. Available observations include indoor temperature, applied heating and cooling power, outdoor temperature, and solar irradiance.

The found model is intended to be used within model-based decision-making frameworks. In particular, the model is integrated into model predictive control (MPC) frameworks, where it is repeatedly queried under hypothetical control actions [11]. Additionally, given physical consistent responses, it could serve as a computationally efficient surrogate environment for model-based Reinforcement Learning (RL). Consequently, the model's physical plausibility under hypothetical control actions is critical.

4.2 Assumptions and Scope

This work focuses on buildings whose thermal dynamics can be reasonably approximated by low-order RC models. The structure of the RC model is assumed to be known a priori, while its parameters are unknown and must be identified from data. Consequently, the governing differential equations describing the thermal dynamics are fixed, and the learning task is restricted to parameter estimation rather than system discovery.

The available indoor temperature data are assumed to be limited in amount and variability, reflecting realistic operational conditions. Certain influential factors, such as occupant behavior and internal heat gains, are not directly measured and are treated as unobserved disturbances.

This thesis does not aim to construct high-fidelity building prediction models, nor does it address unobserved disturbances modeling. Instead, the scope is limited to the robustness of the identification procedure under data scarcity and the multi-step consistency of the identified thermal models.

4.3 Challenges and Failure Modes

Identifying thermal model parameters from observational building data is a challenging task. Due to limited excitation, strong correlations among inputs, and unobserved disturbances, multiple parameter configurations may produce similar temperature predictions. This renders the identification problem ill-posed and sensitive to training data, initialization, noise, and numerical errors [17].

Purely data-driven models, while expressive, often suffer from underspecification and poor generalization in such settings. When used for multi-step prediction or embedded in closed-loop optimization loops, small modeling inaccuracies can accumulate over time and lead to physically implausible behavior [13].

These challenges motivate the investigation of alternative identification formulations that combine physical structure with data-driven regularization, thereby improving robustness of the identification procedure.

4.4 Research Questions

This leads to the following research questions:

1. How does the predictive accuracy of the PINN-based approach compare to that of a classical differentiable physics model in data-limited settings?

2. How does the variability in predictive accuracy of the identified thermal models on different training data compare between the PINN-based approach and a classical differentiable physics model?

3. How do PINN-based and differentiable physics approaches compare in terms of data efficiency?

The methods used to answer these questions are presented in the following chapter.

5 Methods

In this chapter we will cover the structure of the residential building used for simulation, the building data for training, the specific implementations of the different approaches, and the different experimental setups and metrics used to answer the research questions, comparing the proposed approach with a state-of-the-art method.

We aim to learn a physically consistent thermal model from observational data. To achieve this we use a physics-based approach. This approach simplifies the observed building using a 2R2C thermal model, as defined in Equation (5.1). This thermal model consists of five unknown thermal parameters that need to be identified from data. To enhance the robustness of this parameter identification process, we utilize a hybrid physics-informed approach.

The hybrid physics-informed approach consists of an inverse PINN for parameter estimation and the identified thermal model for prediction. Hence, the inverse PINN differs from gradient-based parameter identification methods by leveraging the expressivity of neural networks to approximate temperature trajectories while simultaneously guiding these predictions using the thermal model whose parameters are identified. In contrast, gradient-based parameter identification methods find the trajectories by integrating the thermal model forward in time and matching these predictions with the observations. Thus, introducing a neural network for forward simulation eliminates the need for numerical integration and differentiation. Consequently, the inverse PINN approach decouples parameter estimation from numerical integration, offering a potentially more robust method for identifying thermal models from data. To quantify the robustness gains of the inverse PINN approach, we compare its performance with that of a classical parameter identification method implemented using a differentiable state-space model.

Data for this identification are generated using the advanced building simulation framework, IDA ICE. This simulation environment allows us to easily adjust operational settings to test the performance of the identified thermal models under unseen conditions. To compare the two methods, we select twelve distinct seven-day windows from the simulated annual dataset as training data. This yields twelve thermal models, each trained on one of the seven-day windows. Each of these twelve models are used to predict the full year's temperature dynamics given the same building and weather conditions but with different operational settings. The resulting predictions are used to analyze and compare the predictive accuracy, the sensitivity of the predictive performance with respect to the training data, and data efficiency between the differentiable state-space approach and the proposed approach.

This chapter continues as follows. In Section 5.1 we briefly introduce the building used to generate the data. In Section 5.2 we introduce the different approaches used to estimate the unknown thermal parameters, and finally in Section 5.3 and Section 5.4 we discuss the experimental design and evaluation metrics, respectively.

5.1 Reference Building and Data Generation

The training data was simulated using IDA ICE [18], an advanced building energy simulation software. As noted by Ryan et al. [53], IDA ICE is one of the four major building energy simulation programs discussed in the literature. Unlike simplified tools, IDA ICE offers a detailed dynamic representation of thermal behavior, heat pump operation, and temperature changes due to solar gains and occupant-related effects. Hence, IDA ICE enables detailed and realistic temperature simulations, making it well suited for evaluating the research questions of this thesis.

5.1.1 IDA ICE Reference Model

The simulated building represents a modern, well insulated residential building. It consists of a single-zone space conditioning system and has a $100\mathrm{m}^2$ conditioned floor area with an additional unheated attic. The external walls, ground slab, and ceiling towards the attic have U-values of 0.177, 0.159, and $0.184\mathrm{W\,m^{-2}\,K^{-1}}$, respectively, while the attic roof has a U-value of $0.099\mathrm{W\,m^{-2}\,K^{-1}}$. The total window area is $29\mathrm{m}^2$. The windows have a U-value of $0.60\mathrm{W\,m^{-2}\,K^{-1}}$ and a solar heat gain coefficient of 0.49. The building parameters were selected to approximate the thermal characteristics of an average modern residential building, thereby providing a representative reference for indoor temperature dynamics. A floor plan of the described building is shown in Figure 5.1.

A thermostat located in the living area controls the indoor temperature. The indoor temperature conditioning system of the building consists of floor heating and ceiling cooling systems. The heat pump has a brine-source and 500 liter hot and cold storage tanks.

While the occupant comfort levels are set to 20°C in winter and 25°C in summer, the occupant behavior is modeled dynamically through schedules for occupancy, lighting, and appliance-related internal gains. To further improve the realistic operation, window opening is controlled using the measured indoor temperature and CO_2 levels.

The IDA ICE simulation software uses hourly sampled weather data from the "Hohe Warte" weather station in Vienna. This dataset consists of a full weather year which is used to simulate the thermal response of the building [54]. While the weather data is specified hourly, the IDA ICE solver uses an adaptive internal time step. This high-fidelity simulation model serves as a flexible building surrogate, capable of generating realistic temperature dynamics for testing our parameter identification approaches.

5.1.2 Data Gathering & Preprocessing

Data from the IDA ICE simulation are processed in two stages. Initially, we extract key variables from the detailed output files, including indoor air temperature, ambient temperature, applied thermal power, and incident solar irradiance. The solar irradiance, initially derived from the weather data, was refined to improve accuracy by calculating

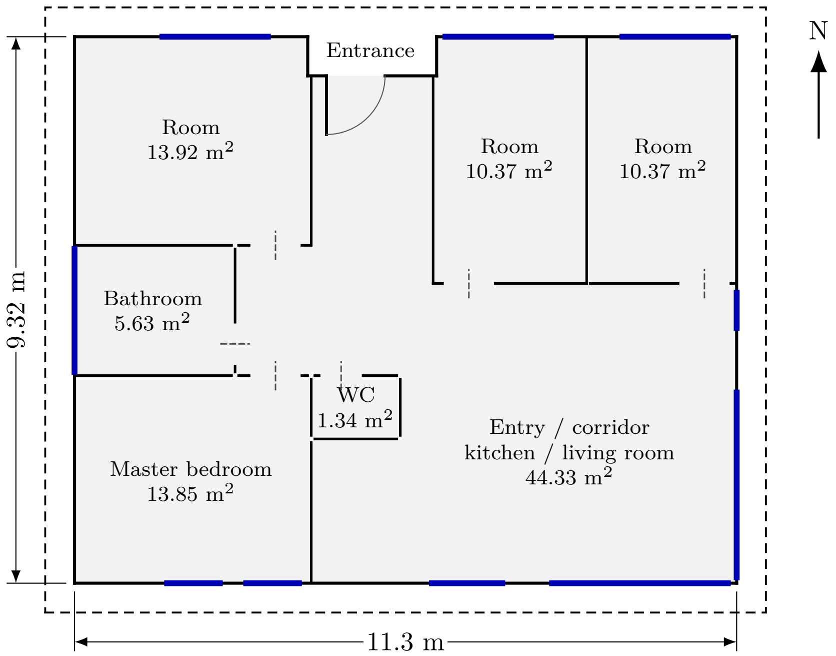


Figure 5.1: The floor plan of the IDA ICE reference building used for data generation. Blue segments denote windows, the dashed outer outline indicates the unheated attic footprint, and dashed interior marks denote doorway openings.

facade-specific irradiance using the `pvlib` Python package [55]. Finally, due to the adaptive time stepping used by IDA ICE to ensure high simulation accuracy, the resulting time series data were resampled to a uniform 15-minute interval using linear interpolation.

In the second step, the preprocessed data was further prepared for model training. The data preparation consists of renaming the input features, removing duplicated timestamps, and applying a threshold to the thermal input, setting small values to zero to remove artifacts that do not represent actual thermal input.

The input features of both models are the ambient temperature T_{amb} [°C], the supplied heating power P_{h} [W], the supplied cooling power P_{c} [W], the solar irradiance incident on south-, north-, east-, and west-facing vertical surfaces, denoted by P_{sS}, P_{sN}, P_{sE}, and P_{sW} [W m^{-2}], respectively, and the indoor air temperature T_{in} [°C]. To obtain a single solar input, these orientation-specific irradiances are combined into an effective solar signal:

$$P_{\text{s}} = \frac{1}{4}\left(P_{\text{sN}} + P_{\text{sE}} + P_{\text{sS}} + P_{\text{sW}}\right).$$

This corresponds to assuming a uniform distribution of the effective solar exposure across the four vertical orientations. Although this aggregation is a simplifying assumption, it provides a more realistic temporal representation of solar influence than a single horizontal irradiance signal, as it reduces the dominance of noon-time radiation and increases the contribution of morning and afternoon solar gains [13]. Small values of heating and cooling power, P_h and P_c, were set to zero to remove artifacts of heat pump operation. Subsequently, the resulting signals were combined into a net thermal power $P_\mathrm{tp} = P_\mathrm{h} - P_\mathrm{c}$. Finally, we added a relative time variable t [s] for model training and visualization purposes.

5.2 Modeling Approaches

As discussed in Section 2.3.1, we are using a RC state-space model to describe the main temperature dynamics. These models can be written as RC-circuits and are named according to their complexity, which is defined by the number of modeled states. This means that a first-order model approximates the full building by aggregating all thermal dynamics and characteristics into one state. A second-order model improves accuracy, while increasing the difficulty of parameter identification, by adding a second state to the thermal model. In this work, we focused on identifying a second-order thermal model describing the thermal dynamics of the building using two resistances and two capacities. This configuration is simple yet complex enough to learn the main temperature dynamics in buildings [26, 48, 51].

The 2R2C model can be formulated by the following equations:

$$
\begin{aligned}
\begin{bmatrix} \dot{T}_\mathrm{in}(t) \\ \dot{T}_\mathrm{m}(t) \end{bmatrix} &= \underbrace{\begin{bmatrix} -\left(\dfrac{1}{C_\mathrm{in}R_\mathrm{im}} + \dfrac{1}{C_\mathrm{in}R_\mathrm{ia}}\right) & \dfrac{1}{C_\mathrm{in}R_\mathrm{im}} \\ \dfrac{1}{C_\mathrm{m}R_\mathrm{im}} & -\dfrac{1}{C_\mathrm{m}R_\mathrm{im}} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} T_\mathrm{in}(t) \\ T_\mathrm{m}(t) \end{bmatrix}}_{\mathbf{x}(t)} \\
&\quad + \underbrace{\begin{bmatrix} \dfrac{1}{C_\mathrm{in}} & \dfrac{\alpha}{C_\mathrm{in}} & \dfrac{1}{C_\mathrm{in}R_\mathrm{ia}} \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} P_\mathrm{tp}(t) \\ P_\mathrm{s}(t) \\ T_\mathrm{amb}(t) \end{bmatrix}}_{\mathbf{u}(t)} \\
y(t) &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} T_\mathrm{in}(t) \\ T_\mathrm{m}(t) \end{bmatrix}
\end{aligned} \tag{5.1}
$$

where $\dot{T}_\mathrm{in}(t)$ and $\dot{T}_\mathrm{m}(t)$ denote the time derivatives of the indoor air and thermal mass temperatures, respectively. The model parameters are collected in the parameter vector

$$\boldsymbol{\phi} = \left[C_\mathrm{in}, C_\mathrm{m}, R_\mathrm{ia}, R_\mathrm{im}, \alpha\right]^\mathrm{T},$$

which includes the indoor air capacity C_in and thermal mass capacity C_m [$\mathrm{J\,K^{-1}}$], the thermal resistances R_im between the indoor air and the thermal mass node and R_ia between the indoor air and the ambient air [$\mathrm{K\,W^{-1}}$], and the coefficient α [$\mathrm{m^2}$], which represents the combined effect of window area and solar heat gain coefficient (SHGC) and scales the solar irradiance to an effective thermal input.

The inputs to the system are represented by the vector $\mathbf{u} \in \mathbb{R}^3$. P_tp [W] is the net thermal power delivered to the building, where heating contributes positively and cooling contributes negatively. P_s denotes the solar irradiance in $\mathrm{W\,m^{-2}}$, and T_amb denotes the ambient temperature in °C. Since we only observe the indoor air temperature we multiply the state vector by the observability matrix $\mathbf{C} \in \mathbb{R}^{1\times 2}$.

For both approaches this second-order state-space model is used while training and for the predictions.

Non-Dimensionalization

To improve the conditioning of the matrices $\mathbf{A}$ and $\mathbf{B}$, we transform the observed system into an equivalent dimensionless system. This is achieved by dividing each input by a characteristic reference scale, thereby placing all inputs on a comparable order of magnitude. It was shown that this preconditioning of the dynamics has positive effects on the optimization procedure [29, 56].

To transform the current system into its dimensionless form, we first subtract a characteristic temperature T_ref from the indoor air temperature, the thermal mass temperature, and the outdoor temperature. Additionally, the temperature trajectories are scaled by a characteristic indoor temperature difference $\Delta T = T_\mathrm{in,\,max} - T_\mathrm{in,\,min}$, corresponding to the comfort bounds. This yields centered indoor temperatures whose changes are of order one. The thermal power and solar irradiance delivered to the building are scaled by its characteristic power levels $P_\mathrm{tp,\,ref}$ and $P_\mathrm{s,\,ref}$, respectively. Time t is scaled by a characteristic time constant t_ref. This yields dimensionless variables of order one.

$$
\begin{aligned}
\theta_\mathrm{in} &= \frac{T_\mathrm{in} - T_\mathrm{ref}}{\Delta T} & \theta_\mathrm{m} &= \frac{T_\mathrm{m} - T_\mathrm{ref}}{\Delta T} & \theta_\mathrm{amb} &= \frac{T_\mathrm{amb} - T_\mathrm{ref}}{\Delta T} \\
\Pi_\mathrm{tp} &= P_\mathrm{tp}/P_\mathrm{tp,\,ref} & \Pi_\mathrm{s} &= P_\mathrm{s}/P_\mathrm{s,\,ref} & \tau &= t/t_\mathrm{ref}
\end{aligned}
$$

Equivalently,

$$
\begin{aligned}
T_\mathrm{in} &= T_\mathrm{ref} + \Delta T\,\theta_\mathrm{in}, & T_\mathrm{m} &= T_\mathrm{ref} + \Delta T\,\theta_\mathrm{m}, & T_\mathrm{amb} &= T_\mathrm{ref} + \Delta T\,\theta_\mathrm{amb}, \\
P_\mathrm{tp} &= P_\mathrm{tp,\,ref}\,\Pi_\mathrm{tp}, & P_\mathrm{s} &= P_\mathrm{s,\,ref}\,\Pi_\mathrm{s}, & t &= t_\mathrm{ref}\,\tau.
\end{aligned}
\tag{5.2}
$$

Using the definitions from Equation (5.2), we can write the dimensionless differential equation describing the change of θ_in with respect to τ.

$$
\begin{aligned}
C_\mathrm{in}\frac{d}{dt}\left(T_\mathrm{ref} + \Delta T\,\theta_\mathrm{in}\right) = \frac{C_\mathrm{in}\Delta T}{t_\mathrm{ref}}\frac{d\theta_\mathrm{in}}{d\tau} &= \frac{1}{R_\mathrm{im}}(T_\mathrm{ref} + \Delta T\theta_\mathrm{m} - T_\mathrm{ref} + \Delta T\theta_\mathrm{in}) \\
&\quad + \frac{1}{R_\mathrm{ia}}(T_\mathrm{ref} + \Delta T\theta_\mathrm{amb} - T_\mathrm{ref} + \Delta T\theta_\mathrm{in}) \\
&\quad + P_\mathrm{tp,\,ref}\,\Pi_\mathrm{tp} + \alpha P_\mathrm{s,\,ref}\,\Pi_\mathrm{s}
\end{aligned}
$$

Rearranging this expression by multiplying by t_ref and dividing by ΔT and C_in gives the differential equation in dimensionless form.

$$
\frac{d\theta_\mathrm{in}}{d\tau} = \frac{t_\mathrm{ref}}{R_\mathrm{im}C_\mathrm{in}}(\theta_\mathrm{m} - \theta_\mathrm{in}) + \frac{t_\mathrm{ref}}{R_\mathrm{ia}C_\mathrm{in}}(\theta_\mathrm{amb} - \theta_\mathrm{in}) + \frac{t_\mathrm{ref}\,P_\mathrm{tp,\,ref}}{C_\mathrm{in}\Delta T}\Pi_\mathrm{tp} + \frac{\alpha\,t_\mathrm{ref}\,P_\mathrm{s,\,ref}}{C_\mathrm{in}\Delta T}\Pi_\mathrm{s}
$$

An analogous derivation holds for θ_{m}. This results in the dimensionless system given by:

$$\frac{d\tilde{\boldsymbol{x}}}{d\tau} = \tilde{\mathbf{A}}\tilde{\boldsymbol{x}} + \tilde{\mathbf{B}}\tilde{\boldsymbol{u}}, \tag{5.3}$$

where $\tilde{\boldsymbol{x}} = [\theta_{\text{in}}, \theta_{\text{m}}]^T$ and $\tilde{\boldsymbol{u}} = [\Pi_{\text{tp}}, \Pi_{\text{s}}, \theta_{\text{amb}}]^T$. The dimensionless matrices $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ are defined as:

$$\tilde{\mathbf{A}} = \begin{bmatrix} -(k_{ia}^{(in)} + k_{im}^{(in)}) & k_{im}^{(in)} \\ k_{im}^{(m)} & -k_{im}^{(m)} \end{bmatrix},$$

and

$$\tilde{\mathbf{B}} = \begin{bmatrix} k_{tp}^{(in)} & k_{s}^{(in)} & k_{ia}^{(in)} \\ 0 & 0 & 0 \end{bmatrix},$$

where

$$k_{im}^{(m)} = \frac{t_{\text{ref}}}{R_{im}C_m}, \qquad k_{im}^{(in)} = \frac{t_{\text{ref}}}{R_{im}C_{in}}, \qquad k_{ia}^{(in)} = \frac{t_{\text{ref}}}{R_{ia}C_{in}},$$
$$k_{tp}^{(in)} = \frac{t_{\text{ref}}\,P_{\text{tp, ref}}}{C_{in}\,\Delta T}, \qquad k_{s}^{(in)} = \frac{\alpha\,t_{\text{ref}}\,P_{\text{s, ref}}}{C_{in}\,\Delta T}.$$

Overall, the non-dimensionalization does not change the underlying physics. Non-dimensionalization changes the numerical conditioning of the matrices $\mathbf{A}$ and $\mathbf{B}$. Consequently, this increases the stability of the optimization problem and typically yields more stable gradients during parameter identification [29, 56].

5.2.1 Differentiable State-Space Model

The differentiable state-space model describes the building thermal dynamics using a reduced-order linear state-space model. It serves as the gradient-based parameter identification approach and describes the evolution of the indoor air temperature and the thermal mass temperature in response to internal and external forcings.

The thermal dynamics are formulated in continuous time and then discretized for simulation on the sampled data. Compactly, the continuous time model defined in Equation (5.1) can be formulated as:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,\mathbf{u}(t), \\ y(t) &= \mathbf{C}\,\mathbf{x}(t), \end{aligned}$$

where $\mathbf{A}$ and $\mathbf{B}$ contain the unknown thermal parameter set $\boldsymbol{\phi}$, $\mathbf{C}$ is multiplied by $\mathbf{x}(t)$ to retrieve the indoor air temperature T_{in}. The state vector $\mathbf{x}(t)$ defines the system's trajectory through state space, characterized by both a natural thermal decay and a forced response. The forced response is defined by the input vector $\mathbf{u}(t)$. The forcings are the net heating power P_{tp}, the solar irradiance P_{s}, and the outdoor air temperature T_{amb}.

Due to the linearity of the state-space model, the integration of the system of ODEs can be done using exact matrix-exponential-based state-transition operators. This leads to more precise dynamics compared to simple explicit integration schemes [51, 57].

State-Space Model Discretization

To propagate the state using sampled data, the continuous-time state-space model must be converted to a discrete-time representation. Under a zero-order hold assumption, that is, assuming the input remains constant over one sampling interval, the discrete-time system can be written as:

$$\mathbf{x}_{k+1} = \mathbf{A}_d \mathbf{x}_k + \mathbf{B}_d \mathbf{u}_k, \tag{5.4}$$

with

$$\mathbf{A}_d = e^{\mathbf{A}\Delta t}, \qquad \mathbf{B}_d = \left(\int_0^{\Delta t} e^{\mathbf{A}s}\, ds \right) \mathbf{B}.$$

The term $\mathbf{A}_d \mathbf{x}_k$ describes the natural evolution of the system over one sampling interval, whereas $\mathbf{B}_d \mathbf{u}_k$ represents the accumulated effect of the input during that interval. Together, both terms define the exact discrete-time dynamics under the zero-order hold assumption.

Given the estimated parameter vector $\boldsymbol{\phi}$, the model integrates the system of ODEs over the full training window. This way, we can compare the predicted indoor air temperature $T_{\mathrm{in,\,k}}^{\mathrm{pred}}$ with the observed indoor air temperature $T_{\mathrm{in,\,k}}^{\mathrm{true}}$ at time step k. The loss is defined as the difference between the predicted and the observed temperatures.

Considering that we want to estimate a thermal model that predicts the main dynamics, we used the Huber loss as our training objective. This loss provides a robust alternative to the purely quadratic loss, putting more weight on the overall trajectory difference compared to the single step error. The Huber loss is quadratic in errors below δ and linear in errors above δ, and can be written as:

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2, & \text{if } |a| < \delta, \\ \delta \left(|a| - \frac{1}{2}\delta \right), & \text{otherwise.} \end{cases} \tag{5.5}$$

In our implementation we set $\delta = 1$.

The final loss of the differentiable state-space model is given by:

$$\mathcal{L}(\boldsymbol{\phi}) = \frac{1}{N} \sum_{k=1}^{N} \ell_{\mathrm{Huber}}\left(T_{\mathrm{in,\,k}}^{\mathrm{pred}}, T_{\mathrm{in,\,k}}^{\mathrm{true}} \right).$$

where N is the number of integration steps. This loss enables the gradient-based optimization with respect to the thermal parameters.

Training Procedure

Starting the training procedure, the system is initialized, yielding an initial condition for the integration. The states, being the indoor air and thermal mass temperature, are both initialized with the initial indoor air temperature $T_{\mathrm{in},0}$. The thermal parameters are initialized with approximate values. The thermal capacities are initialized to $10^6\,\mathrm{J\,K^{-1}}$, and the thermal resistances are initialized to $10^{-1}\,\mathrm{K\,W^{-1}}$. To ensure physically meaningful parameter estimates during gradient-based optimization, all resistances, capacitances, and the solar-gain coefficient are parameterized in log-space and mapped to positive values through an exponential transform.

To ensure a consistent and comparable training procedure across all investigated approaches, parameter estimation was performed in two stages. Following initialization, parameter estimation was performed in two stages. First, Adam was used to move the parameters away from their initial values and into a region of lower loss. Second, L-BFGS was used to refine the estimates locally. In the experiments, the Adam phase was run for 3000 gradient steps, followed by an L-BFGS phase with at most 10^5 iterations. Early stopping in the L-BFGS phase occurred when the optimizer satisfied its internal convergence criteria. That is, when gradients became sufficiently small or when the improvement in the loss between successive iterations became negligible. The complete parameter identification procedure is summarized in Algorithm 1 and discussed in detail below.

Algorithm 1 This pseudo-algorithm explains the steps done to find parameter estimates using the differentiable state-space model.

Construct training windows.
Initialize thermal parameters and map to log-space.
Compute dimensionless inputs and targets.

Stage 1: Global optimization (Adam)
for $k = 1$ to 3×10^3 **do**
    Discretize the dimensionless state-space model
    Roll out the model on the training data
    Compute $\mathcal{L} \leftarrow \frac{1}{N}\sum_{k=1}^{N} \ell_{\text{Huber}}\left(T_{\text{in, k}}^{\text{pred}}, T_{\text{in, k}}^{\text{true}}\right)$
    Update $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ using Adam

Stage 2: Local refinement (L-BFGS)
for $k = 1$ to 1×10^5 **do**
    Discretize the dimensionless state-space model
    Roll out the model on the training data
    Compute $\mathcal{L} \leftarrow \frac{1}{N}\sum_{k=1}^{N} \ell_{\text{Huber}}\left(T_{\text{in, k}}^{\text{pred}}, T_{\text{in, k}}^{\text{true}}\right)$
    Update $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ using L-BFGS

This leads to the following parameter identification procedure. At each optimization step, the current parameters are first mapped to the continuous-time system matrices, discretized, and then used to propagate the state sequentially in time under the observed inputs. This forward simulation yields a predicted indoor-temperature trajectory $T_{\text{in, k}}^{\text{pred}}$, which is compared to the observed trajectory $T_{\text{in, k}}^{\text{true}}$. The discrepancy between the two trajectories defines the training objective. In the implemented model, this objective is a Huber loss, defined in Equation (5.5), evaluated over the full predicted indoor-temperature trajectory. Gradients of this loss with respect to the model parameters are then computed by automatic differentiation and used by the optimizer to update the parameter estimates.

For evaluation, the identified parameters are inserted into the continuous-time state-space model. The resulting system matrices are then discretized, and the discrete-time state-space model is propagated forward in time to generate temperature predictions.

This approach represents a standard deterministic parameter estimation method, using trajectory matching to find the best parameter set to describe the observed temperature dynamics.

5.2.2 Inverse Physics-Informed Neural Network

Similar to the differentiable state-space approach, the inverse PINN approach also uses a reduced-order thermal model to describe the temperature changes observed within the building. Consequently, during training, the inverse PINN also estimates the unknown thermal parameters used in this thermal model. The difference from the differentiable state-space model lies in how the approach is rolling out the initial state of the system. Instead of discretizing the domain for numerical integration, the PINN uses a neural network to approximate the observed temperature trajectory.

$$\mathbf{C}\mathcal{N}(t;\boldsymbol{\theta}) \approx T_{\text{in}}(t)$$

Additionally it imposes the thermal dynamics on this approximation to keep it physically consistent.

$$\dot{\mathcal{N}}(t;\boldsymbol{\theta}) \approx \dot{\mathbf{x}}(t)$$

This means that the neural networks is not only fitting the data, but is also informed about candidate trajectories given by the differential equations $\dot{\mathbf{x}}(t)$ of the thermal model.

Both objectives taken together form the basis of how the PINN chooses the right weights and biases $\boldsymbol{\theta}$ to find an approximation that not only fits the data but also respects the physics. Once the PINN has found an approximation and estimated the parameters of the thermal model, the neural network is discarded and, as with the differentiable state-space approach, the found thermal model is used to generate predictions.

As with the differentiable state-space model, the PINN predicts both the indoor air temperature T_{in} and the thermal mass temperature T_{m}. The trajectory of the indoor air temperature is optimized using the data loss. The data loss measures the difference between the network's predictions of T_{in} and the observed indoor air temperature,

$$\mathcal{L}_{\text{data}}(\boldsymbol{\theta}) = \frac{1}{N_d}\sum_{i=1}^{N_d} \ell_{\text{Huber}}(\mathbf{C}\mathcal{N}(t_i;\boldsymbol{\theta}) - T_{\text{in}}(t_i)) \tag{5.6}$$

where N_{d} is the number of samples in the training data and $\mathbf{C}$ is the observability matrix extracting the indoor air temperature prediction from the output of the neural network.

For the data loss we use the Huber loss function defined in Equation (5.5). In our implementation, we set $\delta = 1$.

Since T_m is a latent state, its trajectory is not optimized by the data loss, but inferred from the predicted indoor temperature using the physics loss. The physics loss compares the derivative of the network $\dot{\mathcal{N}}(t)$ with the derivative described by the thermal model $\dot{\mathbf{x}}(t)$. This difference yields the physics residual $\mathbf{r}(t; \boldsymbol{\theta}, \boldsymbol{\phi})$, which is used to guide the network's predictions towards physically consistent behavior.

$$\mathbf{r}(t; \boldsymbol{\theta}, \boldsymbol{\phi}) = \dot{\mathcal{N}}(t;\, \boldsymbol{\theta}) - \mathbf{A}_{\boldsymbol{\phi}}\, \mathcal{N}(t;\, \boldsymbol{\theta}) + \mathbf{B}_{\boldsymbol{\phi}}\, \mathbf{u}(t)$$

The physics residual is used inside the physics loss during optimization. The physics loss is given by the mean squared error (MSE) of the residuals $\mathbf{r}(t; \boldsymbol{\theta}, \boldsymbol{\phi})$.

$$\mathcal{L}_{\text{phys}}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{1}{N_r} \sum_{i=1}^{N_r} \left\| \mathbf{r}(t_r^i;\, \boldsymbol{\theta}, \boldsymbol{\phi}) \right\|^2 \tag{5.7}$$

The variable N_r represents the number of collocation points, θ contains the parameters of the neural network, and $\boldsymbol{\phi}$ represents the unknown parameters of the thermal model that are updated simultaneously.

Both losses taken together form the composite loss function that is used to estimate the temperature trajectory and the thermal parameters $\boldsymbol{\phi}$.

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathcal{L}_{\text{data}}(\boldsymbol{\theta}) + \lambda_{\text{phys}}\, \mathcal{L}_{\text{phys}}(\boldsymbol{\theta}, \boldsymbol{\phi})$$

This defines the loss function and the general optimization structure of the inverse PINN.

In practice, PINNs are notoriously difficult to train [29]. Therefore, several techniques were employed to improve the training procedure.

Neural Architecture

For the purpose of using an PINN to estimate a smooth approximation of the data, we chose to use a standard MLP as neural architecture, which is the most common architecture used with PINNs [29].

We approximate the state trajectory $\mathbf{x}(t)$ as a function of time t, where the input layer is defined by the time coordinate, i.e., $g^0(t) = t$. The approximation is recursively defined as:

$$\mathbf{x}^{(l)}(t) = \mathbf{W}^{(l)} \cdot g^{(l-1)}(t) + \mathbf{b}^{(l)}, \qquad g^{(l)}(t) = \sigma\left(x_\theta^{(l)}(t)\right), \qquad l = 1, 2, \ldots, L,$$

where $\mathbf{W}^l$ is the weight matrix of the l-th layer. In each layer $g^{(l)}(t)$ is the result of an element-wise activation function acting on the linear transformation $\mathbf{x}^{(l)}(t)$.

The final layer computes the network output. To simplify the PINN training objective, the initial condition is incorporated directly into the network architecture. This is achieved by modifying the final layer using a time-dependent gating mechanism. Instead of directly outputting the network prediction, we scale the final layer output with a function that vanishes near the initial time.

More precisely, let $g^{(L)}(t)$ denote the activation of the last hidden layer. We approximate the state trajectory $\mathbf{x}(t)$ as a function of normalized time $t \in [-1, 1]$. The network output is then defined as

$$\mathbf{x}(t) = \mathbf{x}_0 + \left(1 - \exp\left(-\frac{t+1}{\varepsilon}\right)\right)\left(\mathbf{W}^{(L+1)} \cdot g^{(L)}(t) + \mathbf{b}^{(L+1)}\right),$$

where $\mathbf{x}_0 = [T_{\text{in},0}, T_{\text{in},0}]^T$ denotes the initial state of the system and ε is a small positive constant. The gating term suppresses the learned correction near the initial time $t_0 = -1$, ensuring the network output smoothly recovers the initial condition. As time progresses, the gate quickly approaches one, allowing the network to represent the full solution dynamics.

All weights and biases collectively represent the trainable parameters $\boldsymbol{\theta}$ of the MLP. In our implementation, the hyperbolic tangent (tanh) activation function is used. The global differentiability of the tanh activation function is important for PINNs, since the network must be continuously differentiable with respect to its input t.

Non-dimensionalization

Standardizing the inputs of neural networks improves the optimization procedure of PINNs [56]. Therefore, we used a technique called non-dimensionalization proposed by Wang et al. [29]. Consequently, the dimensionless thermal model defined in Equation (5.3) is used throughout the optimization.

Additionally, we linearly rescale time t to the interval $[-1, 1]$, such that the input is centered around zero. This further improves PINN training, since zero-centered inputs are better aligned with the tanh activation function and help maintain the network in its sensitive, non-saturated state.

Fourier Features and Random Weight Factorization

Another challenge in training PINNs arises from the spectral bias of MLPs [58, 59]. Due to this bias, MLPs tend to learn low-frequency components before high-frequency components. A proven approach to mitigate this limitation is the use of Fourier feature embeddings [58]. These augment the network input with periodic features, enabling the MLP to represent periodic functions more effectively.

Due to the normalization of the time coordinate t discussed in Section 5.2.2, longer temperature trajectories increase the difficulty of PINN optimization. Specifically, since the time coordinate is rescaled to the fixed interval $\tau \in [-1, 1]$, longer temperature trajectories are compressed into the same normalized input domain. As a result, the target function varies more rapidly with respect to τ, effectively increasing the frequencies the network must represent.

To improve optimization for longer temperature trajectories, the input representation is augmented using Fourier features. Instead of sampling random frequencies, however, we use a small set of physically motivated harmonics. Given the periodic nature of indoor temperature dynamics, a base period of 24 h is assumed, from which the first three harmonics are defined as:

$$\omega_k(t) \in \left\{ \cos\left(2\pi \frac{kt}{24}\right), \sin\left(2\pi \frac{kt}{24}\right) \right\}, \qquad k = 1, 2, 3 \tag{5.8}$$

This encoding enriches the input representation with periodic components and enables the network to capture higher-frequency variations more effectively, thereby improving convergence [60].

In Figure 5.2, we compare a standard MLP with an MLP using Fourier feature embeddings. Both use the same fully connected MLP architecture featuring two hidden layers, each with 8 neurons and tanh activation. Parameters were initialized using Xavier-normal initialization and optimized via Adam with a learning rate of 10^{-3}. The difference lies in the input parametrization. The standard MLP uses only the time coordinate t, whereas the MLP using Fourier feature embeddings augments the input with three harmonic Fourier features having periods of 24 hours, 12 hours, and 8 hours. While the

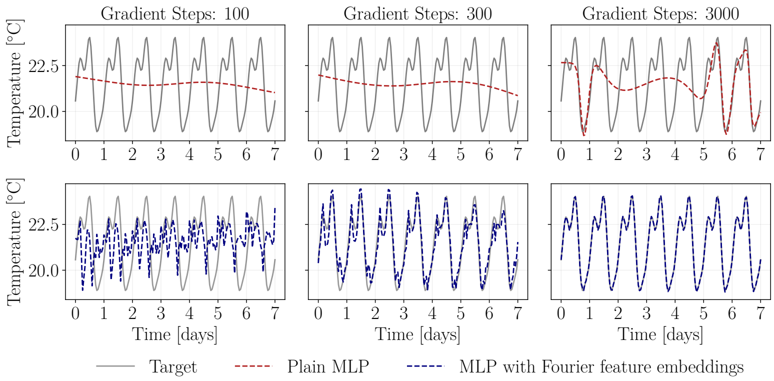


Figure 5.2: Visualization of a standard MLP (top) and a MLP with Fourier feature embeddings (bottom). Both are trained on the same data for 3000 gradient steps.

standard MLP struggles to find a representation to approximate the given data, the MLP using Fourier feature embeddings converges faster and yields a good approximation after 3000 gradient steps.

As a result, the rescaled input feature τ is augmented with the first three harmonics defined in Equation (5.8). Both the original input and the resulting Fourier features are then provided to the MLP, as shown in Figure 5.2.

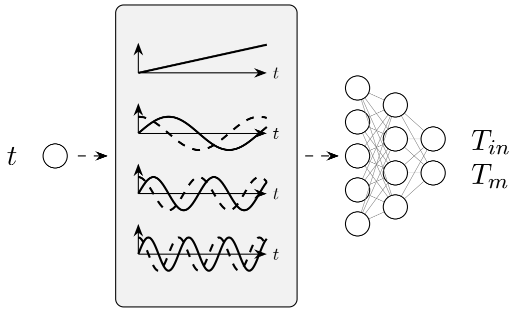


Figure 5.3: Integration of Fourier feature embeddings into the MLP. The input is augmented with Fourier features, consisting of the first three sine (solid) and cosine (dashed) harmonics corresponding to a base period of 24 h.

In addition to Fourier feature embeddings, we used Glorot (Xavier) initialization [61] and random weight factorization (RWF) [62] for all dense layers. Instead of optimizing the weights $\mathbf{W}^{(l)}$ directly, using RWF we factorize the weight matrices

$$\mathbf{W}^{(l)} = \mathrm{diag}\big(\exp(\mathbf{s}^{(l)})\big)\,\mathbf{V}^{(l)}, \qquad \mathbf{s}^{(l)} \sim \mathcal{N}(\mu, \sigma^2\mathbf{I}) \tag{5.9}$$

and then optimize $\{\mathbf{s}^{(l)}, \mathbf{V}^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^{L+1}$. As suggested by Wang et al. [29], we used $\mu = 0.5$ and $\sigma = 0.1$. This factorization has been shown to further mitigate spectral bias and improve the training of PINNs [29].

Training Procedure

To estimate the thermal parameters, we used an MLP consisting of three hidden layers and 128 neurons per layer. All parameters were initialized with Glorot initialization and all weight matrices were reparameterized using RWF, as defined in Equation (5.9). To improve optimization performance we used the described Fourier features to represent periodic behavior over a daily cycle. Thereby we augmented the scalar input feature τ with the first three harmonic fourier features of a 24-hour period. This increases the input dimension from one to seven.

As discussed, the optimization of the neural networks weights and biases and the parameters of the thermal model is performed simultaneously. To ensure physically plausible thermal parameters, we imposed positivity through a log-space parameterization. More precisely, the unknown thermal parameters were optimized in log-space and transformed back to the physical domain via an exponential map during the forward pass.

The total loss is defined as

$$\mathcal{L} = \lambda_{\mathrm{data}}\mathcal{L}_{\mathrm{data}} + \lambda_{\mathrm{phys,\,in}}\mathcal{L}_{\mathrm{phys,\,in}} + \lambda_{\mathrm{phys,\,m}}\mathcal{L}_{\mathrm{phys,\,m}}, \tag{5.10}$$

where $\mathcal{L}_{\mathrm{data}}$ denotes the data mismatch, $\mathcal{L}_{\mathrm{phys,\,in}}$ is the physics residual associated with the observed indoor temperature, and $\mathcal{L}_{\mathrm{phys,\,m}}$ is the physics residual associated with the latent thermal mass temperature. Since these objectives can produce competing gradients, especially during the early stages of training, we introduced weighting factors to balance their contributions. In all experiments, we used

$$\lambda_{\text{data}} = 5, \qquad \lambda_{\text{phys, in}} = 2, \qquad \lambda_{\text{phys, m}} = 4.$$

The loss weights were determined through empirical hyperparameter tuning by evaluating different weighting configurations based on the predictive accuracy of the identified thermal models on the test data. The reported values yielded the best overall performance.

Before training, we generated additional collocation points for evaluating the physics residuals. For each pair of consecutive measurement times, we uniformly sampled four intermediate collocation points. Because measurements were available only every 15 minutes, the input variables at these collocation points were obtained by interpolation. Outdoor temperature T_{amb} and solar gains P_{s} were linearly interpolated, whereas the thermal power input P_{tp} was sampled using zero-order hold interpolation. The complete parameter identification procedure is summarized in Algorithm 2 and discussed in detail below.

Algorithm 2 This pseudo-algorithm explains the steps done to find parameter estimates using the inverse PINN approach.

```
Construct training windows
Initialize trainable parameters θ and φ
Compute dimensionless inputs and targets
Generate collocation points τ_c
Interpolate θ_amb, Π_tp, and Π_s onto τ_c
Set loss weights λ_data ← 5, λ_phys,in ← 2, and λ_phys,m ← 4

Stage 1: Global optimization (Adam)
for k = 1 to 3 × 10^4 do
    Evaluate the PINN on the training data
    Compute L_data, L_phys,in, and L_phys,m using Equations (5.6) and Equation (5.7)
    Compute the total loss L using Equation (5.10)
    Update θ and φ using Adam

Stage 2: Local refinement (L-BFGS)
for k = 1 to 1 × 10^5 do
    Evaluate the PINN on the training data
    Compute L_data, L_phys,in, and L_phys,m using Equations (5.6) and Equation (5.7)
    Compute the total loss L using Equation (5.10)
    Update θ and φ using L-BFGS
```

Training was carried out in two stages. In the first stage, we used the Adam optimizer to obtain a robust initial estimate of both the network weights and the thermal parameters. To avoid instability from poorly initialized physics terms, optimization was started using only the data loss, and the physics losses were gradually introduced afterwards. This strategy first allows the network to learn a smooth approximation of the observations. After the network found a representation, we started to gradually increase the weight of the physics loss terms. This introduced new gradient directions, pushing the parameter estimates towards physically meaningful values [63]. Since the early stage primarily serves to learn a smooth trajectory and explore the loss landscape for plausible parameter values, different learning rates were used for the network parameters and the thermal parameters. The learning rate for the network parameters was set to 10^{-3}, while the learning rate for the thermal parameters was set to 10^{-2}. The Adam stage was run for 3×10^4 iterations.

In the second stage, optimization was continued using L-BFGS in order to refine the solution. The objective function remained unchanged, while both the neural network parameters and the thermal parameters were further optimized. For L-BFGS, we used fixed loss weights of $\lambda_{\text{data}} = 5$, $\lambda_{\text{phys,in}} = 2$, and $\lambda_{\text{phys,mass}} = 4$. The strong Wolfe line search was used with a step size of 0.5. The maximum number of function evaluations per iteration was set to 5×10^4, and the history size was increased to 150 to improve the Hessian approximation. The L-BFGS stage was stopped early either if it reached the maximum iteration number of 10^5 optimization steps, or if the obtained gradient norm was below 10^{-10}, or if the absolute loss change between consecutive iterations was below 10^{-10}.

After convergence, the neural network was discarded and only the estimated physical parameters were retained. These parameters were then used with the discretized state-space model, defined in Equation (5.4), to predict indoor air temperature under varying influences and initial conditions.

5.3 Experimental Design

To address the research questions we designed three experiments that should provide information regarding the predictive accuracy, the sensitivity with respect to training data, and the data efficiency under varying dataset size.

5.3.1 Training Data Selection and Windowing

Training data selection is an important preprocessing step for stable parameter identification. The following procedure selects windows containing sufficient system excitation and temporal variation in the inputs to ensure reliable identifiability of the thermal parameters.

After preprocessing, as described in Section 5.1.2, we iterate over the full year dataset with a stride of one day to extract seven-day sequences. Each of these seven-day sequences is assigned a score. These scores quantitatively describe whether the information inherent in these seven days is enough for parameter estimation. It is based on the temporal variation of the different signals. This is a measure to rate windows higher which have non-zero thermal input data, strongly varying outdoor temperatures, or solar gains, thereby increasing identifiability of the unknown thermal parameters.

After calculating the scores, we sampled from these sequences such that we chose data that has enough information for parameter identification and is diverse enough to represent the full year. Specifically, we assigned a season to each candidate window using its timestamps. Then, using the score of the candidate windows, we selected the three highest-scoring windows per season. To avoid small temporal variation of the selected windows inside one season, we first selected the window with the highest score and then imposed a minimum spacing of one day to the next selected window preventing windows from overlapping. This resulted in twelve training windows in total, having three windows for each season.

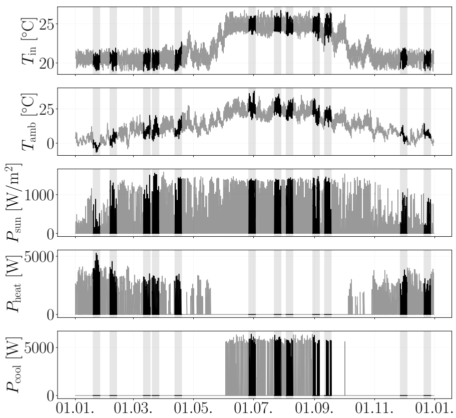


Figure 5.4: Visualization of the selected window for model training across the full year dataset.

For test data, we ran the simulation in IDA ICE a second time on the same building. This time we simulated the building under different thermostat settings. Although we have the same building under the same weather influences, the temperature dynamics inside the building are altered. This is because we change the hysteresis of the temperature control. Instead of using a setpoint of 20.0/25.0 with a hysteresis of 1.0°C we increased it to $20.0/25.0 \pm 1.5$°C. This introduces temperature dynamics unseen in the training data and should therefore show if the trained model is able to generalize to unseen operating conditions, which is of prime interest for maximizing the potential of the flexibility optimization.

5.3.2 Predictive Accuracy

Predictive accuracy of the identified thermal models is an important characteristic for model-based optimization. But instead of focusing on single-step performance, we focus on multi-step consistency. This is because the learned model is used with closed-loop predictions, thereby using its previous predicted state as input. A model that does not capture the driving temperature dynamics will fail in a closed-loop setting, resulting in long-term drifts from the true temperature trajectory [13]. Consequently this would decrease the potential of the flexibility optimization.

To measure the predictive accuracy and compare it between the two approaches, we use the training data selection, described in Section 5.3.1, to extract twelve seven-day sequences from the building simulation data. Each of the extracted seven-day training sequences is used to identify a thermal model. The resulting models are used to predict indoor air temperatures on the unseen test data. The accuracy of these predictions is measured using the MAE, defined in Section 5.4.1. The obtained MAEs from the twelve models form a MAE distribution. From this distribution, we estimate the expected MAE and the standard error, which quantifies the uncertainty in the estimate. Since we are interested in the performance of the thermal models on multiple different prediction horizons, we evaluate the predictive accuracy not only on the day-ahead predictions, but also on three-day, and seven-day prediction horizons.

5.3.3 Sensitivity to Training Data

The sensitivity of the training procedure with respect to the available training data is another important characteristic of a parameter identification approach. Practically, a parameter identification approach can perform very well on certain data, while failing on other. That's why the approaches are compared with respect to their sensitivity to the different training data sequences sampled from different seasons across the year.

Since different parameter combinations can yield similar thermal dynamics, the identified thermal models are evaluated based on the variability of their predictive accuracy rather than solely on the distribution of the estimated parameters. To evaluate the variability of their predictive accuracy, we use the distribution of MAEs obtained from identifying thermal models on the different datasets, as described in Section 5.3.2. We visualize this distribution as boxplots and statistically analyze them for both approaches. In this work, we define an approach to be less sensitive to the training data if it varies less with the training data. Consequently, the higher the standard deviation of the MAE distribution, the higher the sensitivity to the training data. The metric used to quantify the sensitivity is defined in Section 5.4.2.

5.3.4 Data Efficiency

The data efficiency of the approach is important for flexibility optimization, because we want to be able to learn a thermal model from as little data as possible. To compare the data efficiency of both approaches, the procedure described in Section 5.3.2 is used to get an estimate of the expected MAE and its standard error on a specific training data size. As the training data size decreases the degradation of the mean MAE of both approaches is analyzed. Additionally we compare the predictive accuracies of the obtained models across different prediction horizons. Overall, this yields information about how well the parameter identification process utilizes the data at hand and thus describes the data efficiency of the approach [48].

5.4 Evaluation Metrics

In accordance with the research questions, we evaluate both approaches with respect to predictive accuracy, training stability, and data efficiency. Each property is quantified using consistent statistical measures across multiple training runs.

5.4.1 Predictive Accuracy

To measure the predictive accuracy of the resulting thermal models, these models are used to predict indoor temperatures for the next day, the next three days, and the next seven days. As Drgoňa et al. [16] discussed, multi-step consistency is more important for control-oriented thermal models than single-step error. Thus, the predictions are closed-loop, which means that for the next prediction the last prediction is used. This leads to error accumulation and thus a trajectory drift if the main temperature dynamics are not captured by the thermal model.

To quantitatively describe the difference between the observed temperature trajectory $T_{\mathrm{in,\,k}}^{\mathrm{obs}}$ over the different prediction horizons and the predicted temperature trajectory $T_{\mathrm{in,\,k}}^{\mathrm{pred}}$, we use the MAE. The MAE is defined as

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |T_{\mathrm{in,\,k}}^{\mathrm{obs}} - T_{\mathrm{in,\,k}}^{\mathrm{pred}}|$$

where we sum the step-wise absolute differences and divide by the number of steps N to obtain the average absolute error.

We report the MAE as the primary evaluation metric because it measures the average absolute forecast deviation directly and is less sensitive to large individual errors compared to the RMSE. Thus, it is a suitable metric for assessing overall consistency across multi-step forecasts.

Since we are not only interested in a single model performance resulting from a single training data window, we compare the predictive performance across all prediction horizons over all twelve models. This yields a distribution of MAEs which can be used to estimate the expected MAE and the standard error of this estimation. The expected MAE is estimated using the sample mean and is calculated as

$$\overline{\mathrm{MAE}} = \frac{1}{N_{\mathrm{m}}}\sum_{i=1}^{N_{\mathrm{m}}} \mathrm{MAE}_i$$

where $\mathrm{MAE_i}$ is the MAE of the i-th model, and N_m is the number of trained models. The standard error SE of the expected MAE estimate $\overline{\mathrm{MAE}}$ is calculated as

$$\mathrm{SE} = \frac{s}{\sqrt{N_{\mathrm{m}}}}, \quad s = \sqrt{\frac{1}{N_{\mathrm{m}}-1}\sum_{i=1}^{N_{\mathrm{m}}}\left(\mathrm{MAE}_i - \overline{\mathrm{MAE}}\right)^2}$$

where s denotes the sample standard deviation of the MAE values, computed with Bessel's correction. The standard error SE is then obtained by dividing the sample standard deviation s by the square root of the number of models N_{m}.

5.4.2 Sensitivity to Training Data

To analyze the sensitivity of the training procedure across different training data, we compare the predictive accuracy of the resulting models. In particular we are training twelve thermal models on twelve different datasets sampled across the year. This process is described in detail in Section 5.3.1. The resulting models are evaluated with respect to their predictive accuracy on a full year dataset. Each model, trained on one of the twelve datasets, is assigned a predictive accuracy, as described in Section 5.4.1. These predictive accuracies form a distribution, whose mean, median, interquartile range (IQR), quartiles, and outliers are analyzed using boxplots.

Overall the gathered accuracies form a distribution which can be statistically examined to see which approach is less sensitive for estimating thermal models over a full year of building temperature data. With consistent predictive performance across datasets, the approach is considered insensitive to the training data, whereas a higher variability of the predictive performance indicates a higher sensitivity to training data.

Specifically, to quantify sensitivity to the selected training data, this work measures the variability of the obtained predictive accuracies using their standard deviation s.

$$s = \sqrt{\frac{1}{N_\mathrm{m} - 1} \sum_{i=1}^{N_\mathrm{m}} \left(\mathrm{MAE}_i - \overline{\mathrm{MAE}} \right)^2}$$

A lower value of s indicates that the resulting model performance varies less across training windows and is therefore interpreted as less sensitive.

5.4.3 Data Efficiency

To quantitatively describe data efficiency, we use the predictive accuracy of both approaches on different training dataset sizes. As described in Section 5.3.2, the twelve different training datasets are used to train twelve thermal models. To analyze the data efficiency the size of the training data is changed. Specifically we start with twelve training datasets of seven days each and decrease the sequence length to just two days. For each training dataset size this process yields a distribution of predictive accuracies. This distribution is used to calculate the expected predictive accuracy and its uncertainty quantified by the standard error. These measures are plotted against the training dataset size. Additionally we plot the predictive accuracies of the resulting thermal models against the prediction horizon. The calculation of the expected predictive accuracy and the calculation of the standard error is described in Section 5.4.1.

Overall, we evaluate the data efficiency of an approach, by comparing how the predictive accuracy degrades if the training data is reduced [48].

6 Results

In the following we will present results relevant for answering the research questions. In particular we will describe our empirical observations and show how the differentiable state-space model (SSM) and the inverse PINN approach compare in predictive accuracy, in sensitivity to the training data, and in data efficiency.

6.1 Predictive Accuracy of Identified Thermal Models

To evaluate predictive accuracy in data-limited settings, the training data is fixed to seven day windows. These data are used to learn a thermal model of the building, which is then evaluated on unseen test data. The detailed experimental design is described in Section 5.3.2 and the evaluation metrics are defined in Section 5.4.1.

To compare both approaches, their predictive accuracies on the test data are summarized in Table 6.1.

Table 6.1: Comparison of predictive accuracy for prediction horizons of one, three, and seven days ahead. Reported values are mean MAE ± standard error for all evaluated models. Lower values indicate better predictive accuracy. Best performance for each horizon is shown in bold.

	MAE [°C]		
Model	**1 day**	**3 days**	**7 days**
Inverse PINN	$\mathbf{0.467 \pm 0.023}$	$\mathbf{0.615 \pm 0.031}$	$\mathbf{0.761 \pm 0.056}$
Differentiable SSM	0.530 ± 0.044	0.840 ± 0.121	1.329 ± 0.291

Specifically, we report the estimate of the expected MAE with its standard error for both approaches and prediction horizons of one, three, and seven days. Comparing the estimates of the expected MAE, the inverse PINN approach achieves lower MAEs across all prediction horizons. Additionally the standard errors, which quantify the uncertainty in our estimate of the expected MAE, are smaller. The absolute improvements of the obtained $\overline{\text{MAE}}$ by the inverse PINN approach were 0.063, 0.225, and 0.568 for the different prediction horizons. This corresponding to relative improvements of 11.9%, 26.8%, and 42.7%, respectively.

While this demonstrates consistently higher predictive accuracy for the inverse PINN, it also indicates that the predictive performance increase grows with the prediction horizon. Figure 6.1 visualizes the expected MAE as a function of the prediction horizon. Figure 6.1 shows that the inverse PINN not only achieves lower expected MAE, but also

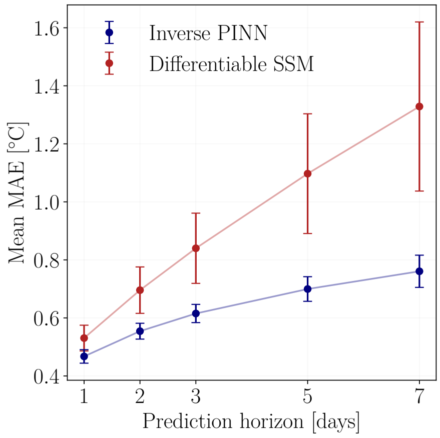


Figure 6.1: Predictive accuracy for closed-loop predictions on increasing prediction horizons. Points show the estimate of the expected MAE, and error bars indicate the standard error of this estimate.

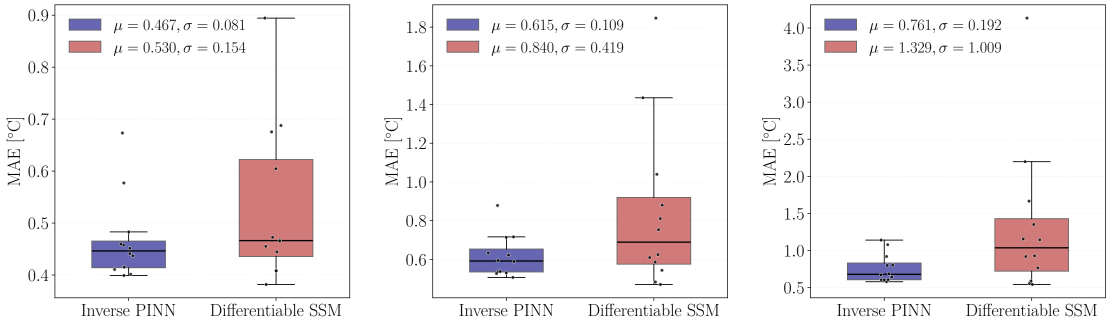
exhibits a slower increase in error with respect to the prediction horizon.

Overall, these results show that the inverse PINN approach is more accurate and provides more reliable predictions in data-limited settings.

6.2 Stability of Parameter Identification

To analyze the stability of parameter identification, that is, the sensitivity of the identified models to the choice of training data, the distribution of MAEs on the test data is examined across models trained on different datasets. The detailed experimental design is described in Section 5.3.4, and the evaluation metrics are defined in Section 5.4.3.

Figure 6.2 shows the distribution of MAEs of the twelve thermal models identified from different seven-day windows across the year and evaluated on unseen test data.



(a) One-day-ahead prediction (b) Three-day-ahead prediction (c) Seven-day-ahead prediction

Figure 6.2: Distribution of MAEs obtained from thermal models identified on different training datasets and evaluated on test data for one-, three-, and seven-day-ahead prediction horizons. Boxplots show the variability of predictive performance for both approaches, with black dots indicating individual models.

The plots in Figure 6.2 indicate that the inverse PINN yields a lower central tendency of the MAE distribution than the differentiable state-space model. This is derived by the smaller IQR and the lower standard deviation, yielded by the models identified using the inverse PINN.

In contrast, the differentiable state-space model shows a larger IQR and standard deviation. As suggested in Section 5.4.2, this indicates a stronger dependence of the predictive accuracy on the training data.

To connect the stability of the predictive accuracy to the distribution of the identified parameters, we additionally report a comparison of the distribution of the identified thermal parameters.

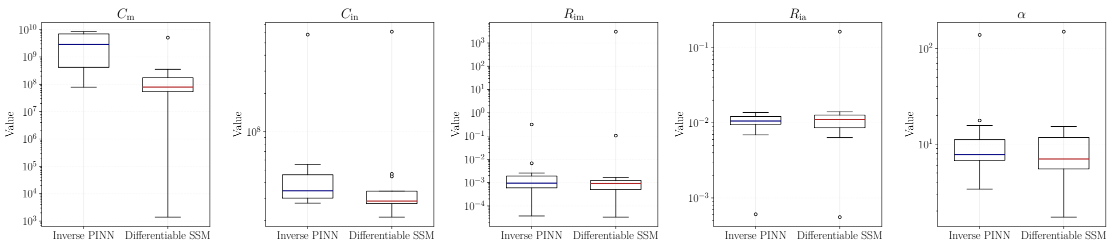


Figure 6.3: Boxplots visualizing the distribution of the obtained RC parameter estimates. The different estimates are generated by identification on twelve different seven day windows across a full year dataset.

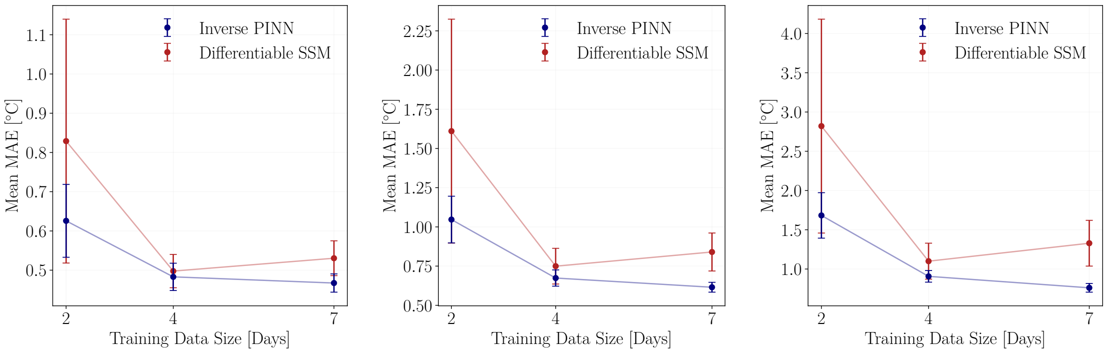
Figure 6.3 shows the identified thermal parameters of the twelve different models, and compares them between both approaches. While the inverse PINN estimates of C_{in}, α, and R_{ia} have lower IQRs, indicating lower uncertainty, the differentiable state-space model has lower IQRs on C_{m} and R_{im}.

Referring to our measure of sensitivity to the training data in Section 5.4.2, these results show that the inverse PINN is less sensitive to the training data, compared to the differentiable state-space model.

6.3 Data Efficiency of Identification Approaches

Data efficiency is evaluated by analyzing how predictive performance degrades as the amount of available training data decreases. The detailed experimental design is described in Section 5.3.4, and the evaluation metrics are defined in Section 5.4.3.

Figure 6.4 visualizes the expected MAE as a function of the training data size for both approaches. The standard error is used to quantify the uncertainty in the estimated mean MAE.

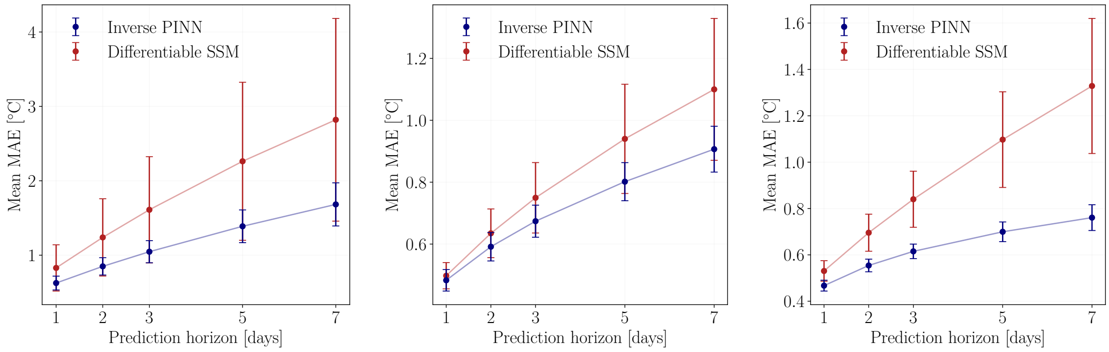


(a) Day-ahead prediction (b) Three-day-ahead prediction (c) Seven-day-ahead prediction

Figure 6.4: Mean MAE as a function of training data size for one-, three-, and seven-day-ahead predictions. Points show the mean MAE over all evaluated models, and error bars indicate the standard errors of the estimates.

The plots in Figure 6.4 show that the inverse PINN approach degrades slower in predictive accuracy as the training data size decreases. As the prediction horizon gets larger, the difference between the inverse PINN and the differentiable state-space approach gets larger.

To further analyze how the predictive accuracy behaves given less training data, we additionally compare the predictive accuracy against the prediction horizon for different training dataset sizes in Figure 6.5.



(a) Two days of training data (b) Four days of training data (c) Seven days of training data

Figure 6.5: Mean MAE as a function of prediction horizon for two, four, and seven days of training data.

The plots in Figure 6.5 show that the inverse PINN approach yields lower mean MAEs consistently across all prediction horizons and training dataset sizes. While the difference of mean MAEs is largest with seven days of training data, it is smallest with four days of training data. Across these plots the difference between the mean MAEs of the two approaches consistently increases as the prediction horizon grows.

Overall, these results show that the inverse PINN achieves lower expected MAE estimates across all training dataset sizes. Nevertheless, the inverse PINN's expected MAE estimate lies within the standard error of the differentiable state-space estimate. This indicates, that the difference is not significant.

7 Discussion

This chapter discusses the results and highlights the main differences between the standard parameter identification approach and the proposed inverse PINN approach. This work investigated the robustness of the identification of thermal models from building data, and compared the identification procedure of an inverse PINN and a differentiable state-space model with respect to their predictive accuracy, their sensitivity to the training data, and their data efficiency.

The results show three main findings. First, thermal models identified using the inverse PINN approach have an increased mean predictive accuracy over all prediction horizons. Their relative improvements range from 11% on day-ahead predictions, to 42% on 7 day-ahead predictions. Second, the standard deviation, measuring the sensitivity of the identification procedure, is consistently smaller for thermal models identified by the inverse PINN approach than that of the differentiable state-space model. The relative decrease of the standard deviation ranges from 47% on day-ahead predictions to 81% on seven day-ahead predictions. Third, the predictive accuracy of the thermal models identified by the inverse PINN approach degrades comparably to the predictive accuracy of the thermal models identified by the differentiable state space model. This indicates that the inverse PINN's data efficiency is similar compared to the gradient-based parameter identification approach.

In the following we will discuss how the results contribute to answering the research questions.

7.1 Predictive Accuracy of Identified Thermal Models

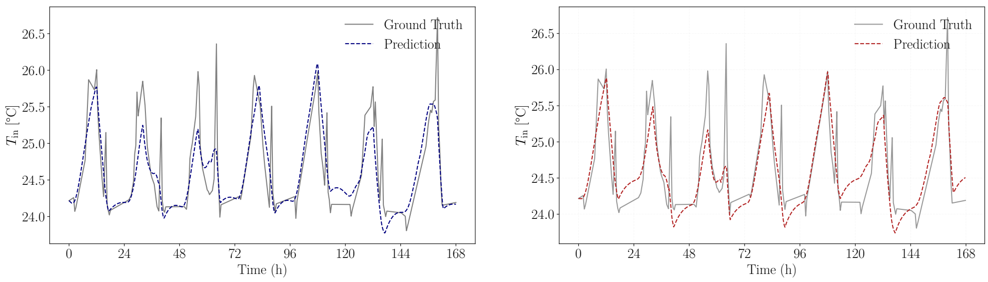
RQ1: How does the predictive accuracy of the PINN-based approach compare to that of a classical differentiable physics model in data-limited settings?

As the results from Section 6.1 indicate, the inverse PINN is able to identify more accurate thermal models from observational building data across all prediction horizons, with the biggest improvements observed on long-term prediction horizons. This indicates that the inverse PINN is more robust in data-limited settings. The following analysis explains the mechanisms underlying this improvement.

Comparison of Training Dynamics

The training dynamics of the inverse PINN and the differentiable state-space model differ mainly in their predictions and training objective.

With the inverse PINN, the temperature is approximated continuously by a forward pass through a neural network, whereas the differentiable state-space model uses a discretized formulation for numerical integration. Considering that the underlying thermal model does not account for disturbances such as occupant behavior or window openings, the observed temperature dynamics are not fully represented. Consequently, the model must identify a trajectory that approximates the data and lies within the set of physically admissible solutions. Figure 7.1a illustrates the temperature dynamics approximated by the inverse PINN, whereas Figure 7.1b presents the solution obtained from the differentiable state-space model.



(a) Inverse PINN.

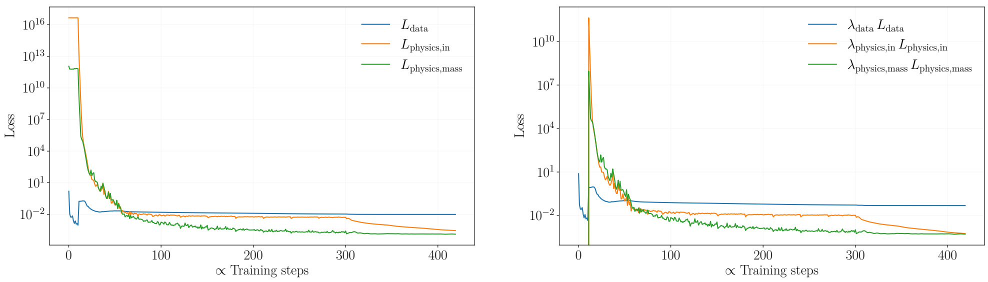
(b) Differentiable SSM.

Figure 7.1: Comparison of the found approximation from which the parameter estimates are derived.

The training objectives differ in the way the physics are incorporated. The differentiable state-space model structurally enforces the physics. This reduces the training objective of the differentiable state-space model to finding a set of thermal parameters that yields temperature trajectories consistent with the true temperature dynamics.

While the differentiable state-space model is constrained to produce trajectories that lie within the admissible solutions, with the inverse PINN we assign weights to the losses, informing the training about the importance of each loss term. In Figure 7.2a we can see the evolution of the different loss terms of the inverse PINN approach against the training steps. Figure 7.2b shows the losses scaled by their weights during training. The initial spike in the physics loss, seen in Figure 7.2b, highlights the training dynamics of the inverse PINN. First, it learns a purely data-driven representation before physics constraints are applied.

This effectively decouples the temperature predictions from parameter estimation. As a result the predictions are not tightly coupled to the parameter estimates. Rather, these estimates guide the predictions to find admissible solutions. This makes the inverse PINN robust to the initialization of the parameter estimates. Consequently, the inverse PINN converges without requiring accurate initialization, whereas the differentiable state-space model fails under such conditions, as shown in Figure 7.3. This suggests that differentiable state-space models are sensitive to their initializations, converging to suboptimal solutions. This is consistent with the review of Serasinghe et al. [17], stating

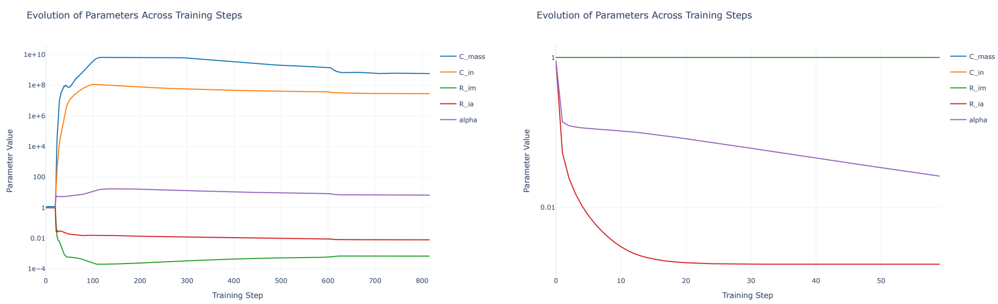


(a) Individual loss terms.

(b) Scaled loss terms.

Figure 7.2: Individual loss terms of the inverse PINN against gradient steps.

that most parameter identification methods require good initial estimates to converge to meaningful solutions. These findings indicate that the differentiable state-space model gets stuck with parameter configurations that do not generalize, whereas the inverse PINN is able to find plausible parameter estimates without relying on good initializations, making it more robust in practice.



(a) Inverse PINN.

(b) Differentiable SSM.

Figure 7.3: Parameter evolution against training steps.

Overall these findings suggest that the inverse PINN improves predictive accuracy through two key mechanisms. First, the smooth data-driven approximation removes numerical errors accumulating through discretization and numerical integration. Second, the staged introduction of the different loss terms, stabilizes the training of the PINN and improves the ability to find parameter estimates that generalize to unseen operating conditions.

7.2 Stability of Parameter Identification

RQ2: How does the variability in predictive accuracy of the identified thermal models on different training data compare between the PINN-based approach and a classical differentiable physics model?

As the results from Section 6.2 indicate, the standard deviation of the resulting MAE distribution produced by the inverse PINN is consistently smaller than the standard deviation of the MAE distribution obtained by the differentiable state-space model. This was shown by analyzing the MAE distribution obtained from thermal models identified on the different training datasets sampled from one year of measurements. Thus, the results provide evidence that the inverse PINN is less sensitive to variations in the training data compared to the differentiable state-space model.

Sensitivity of Identification Procedure

To analyze the sensitivity of the compared identification procedures, we evaluate the thermal models identified on different training datasets. These datasets vary in measurement characteristics such as temperature dynamics, weather influences, as well as heating and cooling effects. Figure 7.4 illustrates the training data sampling procedure.

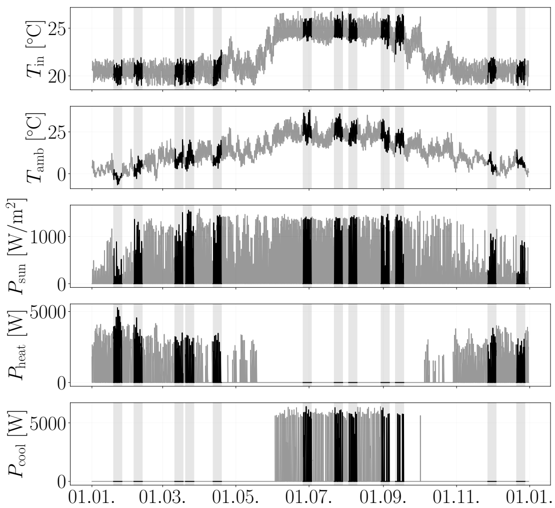
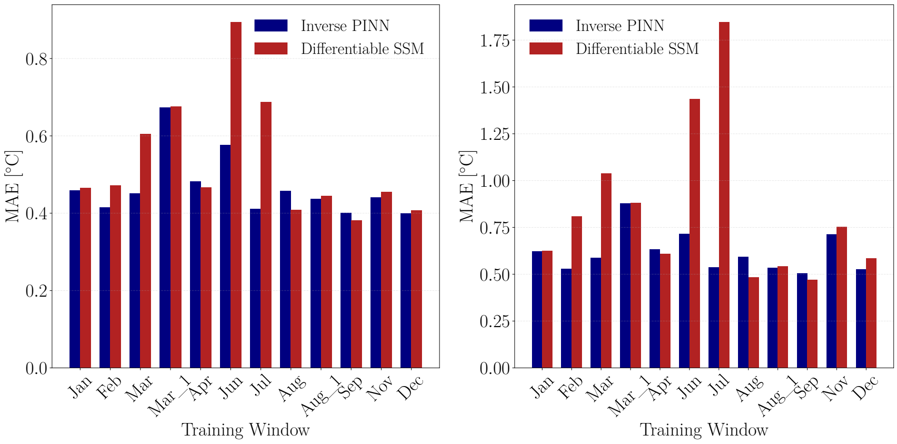


Figure 7.4: Selected windows (grey area) for model training across the full year dataset.

Different datasets yield varying indoor temperature excitations, altering the conditions for parameter identification. To evaluate the resulting difference in the predictive accuracy of the identified thermal models, we visualize the predictive accuracy against each thermal model obtained from a different training window, shown in Figure 7.5. As we can see, while some thermal models identified by the differentiable state-space model achieve a high predictive accuracy over the full year test data, others yield substantially lower predictive accuracy, indicating a strong dependence on the training data. This aligns with the review of Yu et al. [52], arguing that parameter estimates obtained by deterministic identification approaches, like the differentiable state-space model, are significantly influenced by the training data.



(a) Day-ahead performance. (b) Three-day ahead performance.

Figure 7.5: Model performance on test data against the month of the training window.

These findings suggest that the inverse PINN can be less sensitive to to the different parameter identification conditions inherent in the available building data. One possible explanation is the discussed decoupling of the data fitting and parameter estimation. The differentiable state-space approach couples the parameter updates tightly to the numerical integration of the system dynamics. This induces an optimization procedure, which allows only parameter updates that improve the data fit of the numerically integrated temperature trajectory. This constrains the optimization to follow physically admissible trajectories at every step. Inverse PINNs decouple the approximation of the temperature trajectory from the parameter estimation. This yields a less constrained optimization procedure, possibly reducing the sensitivity of the parameter estimation with respect to the training data.

Overall, the results indicate that the inverse PINN identification procedure varies less with the training data and thus yields more stable thermal models across one year of measurements. This robustness is particularly important in real-world applications, where training data is often obtained from environments with limited variability in indoor temperature.

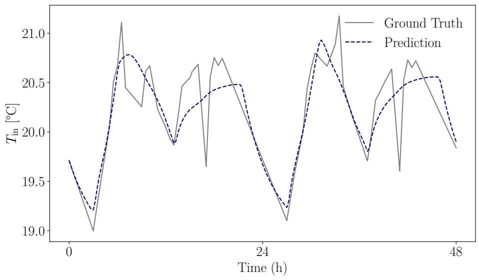
7.3 Data Efficiency of Identification Approaches

RQ3: How do PINN-based and differentiable physics approaches compare in terms of data efficiency?

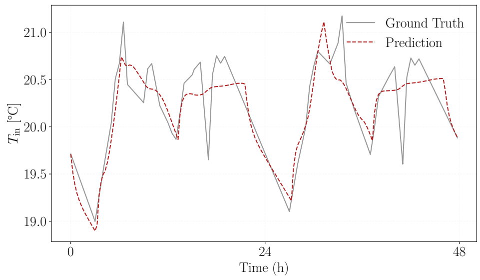
The results from Section 6.3 show that the inverse PINN degrades similarly in performance compared to the differentiable state-space model. This is noteworthy, as classical parameter identification approaches are typically considered data-efficient, while neural network-based methods are often associated with higher data requirements [13].

Degradation of Predictive Accuracy

To better understand their identification performance on limited data, we use a particularly small training window where both approaches struggle to identify a reliable thermal model. On this data, the inverse PINN approach identified a thermal model having a MAE of 1.54 on the full year test data. The differentiable state-space model identified a thermal model having a MAE of 4.22. To get a sense of why their performance diverges, we evaluate their temperature predictions on the training data used for parameter estimation, see Figure 7.6.



(a) Inverse PINN.



(b) Differentiable SSM.

Figure 7.6: Predictions on training data.

The visualized predictions in Figure 7.6 show that the temperature trajectory learned by the inverse PINN is smoother compared to the differentiable state-space model, which potentially improves parameter estimates.

This is consistent with the findings in the previous section, where we derived the improved performance from the decoupling of the temperature approximation and the parameter estimation.

7.4 Contributions

This thesis makes two main contributions. First, it proposes a PINN-based framework for solving the inverse problem of parameter identification for building thermal models. Second, it provides a systematic comparison between the inverse PINN and a gradient-based parameter identification approach, implemented by a differentiable state-space model, with respect to predictive accuracy, sensitivity of the predictive accuracy to training data, and data efficiency.

The findings suggest that inverse PINNs are a promising alternative to classical parameter identification methods for identifying parameters of building thermal models. In particular, the improved predictive accuracy and reduced sensitivity to training data and parameter initialization indicate that the inverse PINN may provide a more robust approach in practical applications. This is particularly relevant in building thermal modeling, where weak excitation conditions often make the parameter identification procedure itself the limiting factor [12, 17].

7.5 Limitations of the Work

It is important to note that this work is subject to several limitations.

The evaluation of both approaches is done on simulated building data. While we tried to keep it as close to real-world operations, real-world building data is more diverse and is influenced by more complex unobserved disturbances. Thus, the findings may not fully generalize to real-world scenarios.

The analysis of the identification procedure was done on a single building configuration. While this brings improvements to the interpretability of the differences within the parameter identification, the differences could be subject to the specific building data and operation.

The simulated building uses a single-zone heating system, allowing the entire structure's temperature to be approximated with one indoor temperature node. More complex system configurations, however, require modeling multiple indoor temperature zones.

The differentiable state-space model serves as a baseline approach for gradient-based parameter identification. In contrast to classical prediction error methods, the model is rolled out over the full temperature trajectory before computing gradients. This formulation enables a direct comparison with the inverse PINN approach, which is likewise trained on trajectory-level consistency. At the same time, more advanced parameter identification methods have been proposed in the building modeling literature and may lead to different comparative results.

Given the loss function of the inverse PINN, the thermal capacity could grow infinitely, minimizing the physics loss. Consequently, we decided to constrain the thermal mass capacity parameter C_m. To minimize interference with the sensitive training procedure of PINNs, the thermal mass capacity was upper-bounded by 1×10^{10}. With the differentiable state-space model the thermal mass capacity parameter stayed unbounded, because its loss function does not directly reward an infinitely growing thermal mass capacity parameter.

Despite these limitations, inverse PINNs were successfully applied in several other fields and represent a promising direction for parameter identification of thermal building models [27].

7.6 Future Research Directions

As discussed, evaluating the inverse PINN on simulated building data and benchmarking it against a classical gradient-based parameter identification approach may not provide a complete picture of its practical improvements for identifying thermal building models. The current implementation of the inverse PINN should be regarded as a baseline, showing the potential of inverse PINNs in the context of building thermal modeling. Hence, there are several improvements to the inverse PINN that could further increase its parameter identification performance.

In its current configuration, the inverse PINN cannot identify thermal models for multi-zone buildings. To extend it to multiple thermal zones, one approach is to couple several inverse PINNs, each representing a single zone, and link them by explicitly modeling the interactions between adjacent zones.

Another improvement could be to learn a building specific window area distribution. As discussed in Section 5.1.2, we decompose the solar irradiance into its directional components incident on the north, east, south, and west facades. Subsequently, these directional irradiance terms are currently aggregated under the assumption that window areas are evenly distributed across all orientations. To improve, one possibility would be to make these aggregation weights trainable, effectively learning the distribution of the window area across the facades.

Finally, an important improvement would be to incorporate disturbances into the thermal model. This would allow the inverse PINN to represent the full observations, rather than having to balance data fit against physically plausible trajectories. Currently, our implementation of the inverse PINN approach only models the observed inputs, leaving a gap between the model and the observations. With PINNs, it is possible to extend the approach by learning a representation of unobserved inputs, such as occupant behavior. Consequently, the inverse PINN would jointly learn the temperature trajectory, infer latent inputs, and estimate the thermal parameters. This separation of the observations into observed and latent components could further improve the stability of the parameter estimates.

These future research directions will help to more extensively evaluate the full potential of inverse PINN in the field building modeling.

8 Conclusion

This thesis is motivated by the redistribution of thermal load by using the thermal mass of buildings as energy storage to support the integration of renewable energy sources into the electricity grid. Effective thermal load redistribution requires prediction models that can accurately and physically consistently forecast indoor temperature dynamics under varying operating conditions, particularly over extended horizons. This ensures that pre-heating or pre-cooling strategies, used to absorb excess electricity, have plausible effects on indoor temperature [5, 13].

In this context, this thesis proposes the use of an inverse PINN for identifying parameters of a thermal model, which can be used to effectively redistribute electricity. Its identification performance is compared to a standard differentiable state-space model with respect to the resulting thermal model's predictive accuracy, the identification procedure's sensitivity to the training data, and the identification procedure's data efficiency.

The results show that using the inverse PINN for parameter identification yields thermal models having a lower prediction error when trained over the course of one year of observations. The most significant improvements were observed in seven-day-ahead predictions, indicating the physical consistency of the found thermal model. Additionally, the inverse PINN's identification shows a decreased dependence on the training data, stabilizing the variability in predictive accuracy. With respect to data efficiency, both approaches yield comparable performance.

These findings suggest that the inverse PINN provides a valuable alternative to classical parameter identification methods. In particular, its training procedure decouples trajectory approximation from parameter estimation, which may reduce sensitivity to training data and parameter initialization. This shifts the identification process from strictly following numerically integrated trajectories to learning data-driven representations that are guided by prior physical knowledge.

However, these results were obtained on a single simulated building, which limits the findings to this specific setting. Future research should evaluate the inverse PINN's potential on real building data and compare it to advanced parameter identification procedures. Hence, it remains to be seen whether the observed potential generalizes to real-world scenarios.

Overall, this thesis demonstrates that inverse PINNs propose an interesting alternative to current parameter identification methods, improving robustness of the identification procedure.

Bibliography

[1] *IEA – International Energy Agency*, https://www.iea.org/reports/tracking-buildings-2020, Oct. 2025.

[2] P. D. Lund, J. Lindgren, J. Mikkola, and J. Salpakari, "Review of energy system flexibility measures to enable high levels of variable renewable electricity", *Renewable and Sustainable Energy Reviews*, vol. 45, pp. 785–807, May 2015, ISSN: 1364-0321. DOI: 10.1016/j.rser.2015.01.057

[3] Z. Luo, J. Peng, J. Cao, R. Yin, B. Zou, Y. Tan, and J. Yan, "Demand Flexibility of Residential Buildings: Definitions, Flexible Loads, and Quantification Methods", *Engineering*, vol. 16, pp. 123–140, Sep. 2022, ISSN: 2095-8099. DOI: 10.1016/j.eng.2022.01.010

[4] Eurostat, *Disaggregated final energy consumption in households - quantities*, 2022. DOI: 10.2908/NRG_D_HHQ

[5] J. Lind, E. Möllerström, H. Averfalk, and F. Ottermo, "Energy flexibility using the thermal mass of residential buildings", *Energy and Buildings*, vol. 301, p. 113 698, Dec. 2023, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2023.113698

[6] A. Kathirgamanathan, M. De Rosa, E. Mangina, and D. P. Finn, "Data-driven predictive control for unlocking building energy flexibility: A review", *Renewable and Sustainable Energy Reviews*, vol. 135, p. 110 120, Jan. 2021, ISSN: 1364-0321. DOI: 10.1016/j.rser.2020.110120

[7] D. Blum, Z. Wang, C. Weyandt, D. Kim, M. Wetter, T. Hong, and M. A. Piette, "Field demonstration and implementation analysis of model predictive control in an office HVAC system", *Applied Energy*, vol. 318, p. 119 104, Jul. 2022, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2022.119104

[8] D. Kim, Z. Wang, J. Brugger, D. Blum, M. Wetter, T. Hong, and M. A. Piette, "Site demonstration and performance evaluation of MPC for a large chiller plant with TES for renewable energy integration and grid decarbonization", *Applied Energy*, vol. 321, p. 119 343, Sep. 2022, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2022.119343

[9] P. Mohebi, W. Zheng, and Z. Wang, "Comparing different parameter identification techniques for optimal control of building energy systems", *Energy and Buildings*, vol. 319, p. 114 563, Sep. 2024, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2024.114563

[10] S. Royer, M. Bressan, S. Thil, and T. Talbert, "Modelling of a multi-zone building and assessment of its thermal behaviour using an energy simulation software", in *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug. 2013, pp. 735–740. DOI: 10.1109/CoASE.2013.6653981

[11] 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, ISSN: 1367-5788. DOI: 10.1016/j.arcontrol.2020.09.001

[12] S. Prívara, J. Cigler, Z. Váňa, F. Oldewurtel, C. Sagerschnig, and E. Žáčeková, "Building modeling as a crucial part for building predictive control", *Energy and Buildings*, vol. 56, pp. 8–22, Jan. 2013, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2012.10.024

[13] L. Di Natale, B. Svetozarevic, P. Heer, and C. N. Jones, "Physically Consistent Neural Networks for building thermal modeling: Theory and analysis", *Applied Energy*, vol. 325, p. 119 806, Nov. 2022, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2022.119806

[14] M. Raissi, P. Perdikaris, and 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*, vol. 378, pp. 686–707, Feb. 2019, ISSN: 0021-9991. DOI: 10.1016/j.jcp.2018.10.045

[15] J. Adler and O. Öktem, "Solving ill-posed inverse problems using iterative deep neural networks", *Inverse Problems*, vol. 33, no. 12, p. 124 007, Dec. 2017, ISSN: 0266-5611, 1361-6420. DOI: 10.1088/1361-6420/aa9581

[16] J. Drgoňa, A. R. Tuor, V. Chandan, and D. L. Vrabie, "Physics-constrained deep learning of multi-zone building thermal dynamics", *Energy and Buildings*, vol. 243, p. 110 992, Jul. 2021, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2021.110992

[17] R. Serasinghe, N. Long, and J. D. Clark, "Parameter identification methods for low-order gray box building energy models: A critical review", *Energy and Buildings*, vol. 311, p. 114 123, May 2024, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2024.114123

[18] *IDA ICE - Simulation Software | EQUA*, https://www.equa.se/en/ida-ice.

[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature14539

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks", *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3065386

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All you Need",

[22] L. D. Natale, B. Svetozarevic, P. Heer, and C. N. Jones, *Towards Scalable Physically Consistent Neural Networks: An Application to Data-driven Multi-zone Thermal Building Models*, Apr. 2023. DOI: 10.48550/arXiv.2212.12380

[23] F. Djeumou, C. Neary, E. Goubault, S. Putot, and U. Topcu, *Neural Networks with Physics-Informed Architectures and Constraints for Dynamical Systems Modeling*, Dec. 2022. DOI: 10.48550/arXiv.2109.06407

[24] Y. A. Çengel, *Heat Transfer: A Practical Approach*, Second Edition. McGraw-Hill, 2003, ISBN: 978-0-07-245893-0.

[25] P. Bacher and H. Madsen, "Identifying suitable models for the heat dynamics of buildings", *Energy and Buildings*, vol. 43, no. 7, pp. 1511–1522, Jul. 2011, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2011.02.005

[26] E. Vrettos, E. C. Kara, J. MacDonald, G. Andersson, and D. S. Callaway, "Experimental Demonstration of Frequency Regulation by Commercial Buildings—Part I: Modeling and Hierarchical Control Design", *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3213–3223, Jul. 2018, ISSN: 1949-3061. DOI: 10.1109/TSG.2016.2628897

[27] S. Cuomo, V. S. di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, *Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next*, Jun. 2022. DOI: 10.48550/arXiv.2201.05624

[28] B. Moseley, "Physics-informed machine learning: From concepts to real-world applications", http://purl.org/dc/dcmitype/Text, University of Oxford, 2022.

[29] S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, *An Expert's Guide to Training Physics-informed Neural Networks*, Aug. 2023. DOI: 10.48550/arXiv.2308.08468

[30] Z. Ma, G. Jiang, Y. Hu, and J. Chen, "A review of physics-informed machine learning for building energy modeling", *Applied Energy*, vol. 381, p. 125 169, Mar. 2025, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2024.125169

[31] 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. 111 174, Aug. 2021, ISSN: 1364-0321. DOI: 10.1016/j.rser.2021.111174

[32] C. Vallianos, J. Candanedo, and A. Athienitis, "Thermal modeling for control applications of 60,000 homes in North America using smart thermostat data", *Energy and Buildings*, vol. 303, p. 113 811, Jan. 2024, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2023.113811

[33] 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. 114 229, Jul. 2024, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2024.114229

[34] P. Klanatsky, F. Veynandt, and C. Heschl, “Grey-box model for model predictive control of buildings”, *Energy and Buildings*, vol. 300, p. 113 624, Dec. 2023, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2023.113624

[35] A. Laukkarinen and J. Vinha, “Long-term prediction of hourly indoor air temperature using machine learning”, *Energy and Buildings*, vol. 325, p. 114 972, Dec. 2024, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2024.114972

[36] H. Li, G. Pinto, M. S. Piscitelli, A. Capozzoli, and T. Hong, “Building thermal dynamics modeling with deep transfer learning using a large residential smart thermostat dataset”, *Engineering Applications of Artificial Intelligence*, vol. 130, p. 107 701, Apr. 2024, ISSN: 0952-1976. DOI: 10.1016/j.engappai.2023.107701

[37] J. Mun, H.-G. Jo, and C. S. Park, “Toward scalable prediction of indoor thermal dynamics: Neural-network-implanted state-space (NNiSS) model”, *Energy and Buildings*, vol. 331, p. 115 359, Mar. 2025, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2025.115359

[38] A. Mechaqrane and M. Zouak, “A comparison of linear and neural network ARX models applied to a prediction of the indoor temperature of a building”, *Neural Computing & Applications*, vol. 13, no. 1, pp. 32–37, Apr. 2004, ISSN: 1433-3058. DOI: 10.1007/s00521-004-0401-8

[39] B. Delcroix, J. L. Ny, M. Bernier, M. Azam, B. Qu, and J.-S. Venne, “Autoregressive neural networks with exogenous variables for indoor temperature prediction in buildings”, *Building Simulation*, vol. 14, no. 1, pp. 165–178, Feb. 2021, ISSN: 1996-8744. DOI: 10.1007/s12273-019-0597-2

[40] H. Siegelmann, B. Horne, and C. Giles, “Computational capabilities of recurrent NARX neural networks”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 27, no. 2, pp. 208–215, Apr. 1997, ISSN: 1941-0492. DOI: 10.1109/3477.558801

[41] G. Mustafaraj, G. Lowry, and J. Chen, “Prediction of room temperature and relative humidity by autoregressive linear and nonlinear neural network models for an open office”, *Energy and Buildings*, vol. 43, no. 6, pp. 1452–1460, Jun. 2011, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2011.02.007

[42] E. L. Krüger, L. Fernandes, G. T. Cardoso, and E. E. Kavamura, “Proposition of a simplified method for predicting hourly indoor temperatures in test cells”, *Ambiente Construído*, vol. 17, pp. 57–70, 2017, ISSN: 1415-8876, 1678-8621. DOI: 10.1590/s1678-86212017000300162

[43] Z. Afroz, T. Urmee, G. M. Shafiullah, and G. Higgins, “Real-time prediction model for indoor temperature in a commercial building”, *Applied Energy*, vol. 231, pp. 29–53, Dec. 2018, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2018.09.052

[44] F. Elmaz, R. Eyckerman, W. Casteels, S. Latré, and P. Hellinckx, “CNN-LSTM architecture for predictive indoor temperature modeling”, *Building and Environment*, vol. 206, p. 108 327, Dec. 2021, ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2021.108327

[45] F. Mtibaa, K.-K. Nguyen, M. Azam, A. Papachristou, J.-S. Venne, and M. Cheriet, "LSTM-based indoor air temperature prediction framework for HVAC systems in smart buildings", *Neural Computing and Applications*, vol. 32, no. 23, pp. 17 569–17 585, Dec. 2020, ISSN: 1433-3058. DOI: 10.1007/s00521-020-04926-3

[46] C. Deb, F. Zhang, J. Yang, S. E. Lee, and K. W. Shah, "A review on time series forecasting techniques for building energy consumption", *Renewable and Sustainable Energy Reviews*, vol. 74, pp. 902–924, Jul. 2017, ISSN: 1364-0321. DOI: 10.1016/j.rser.2017.02.085

[47] A. D'Amour, K. Heller, D. Moldovan, B. Adlam, B. Alipanahi, A. Beutel, C. Chen, J. Deaton, J. Eisenstein, M. D. Hoffman, F. Hormozdiari, N. Houlsby, S. Hou, G. Jerfel, A. Karthikesalingam, M. Lucic, Y. Ma, C. McLean, D. Mincu, A. Mitani, A. Montanari, Z. Nado, V. Natarajan, C. Nielson, T. F. Osborne, R. Raman, K. Ramasamy, R. Sayres, J. Schrouff, M. Seneviratne, S. Sequeira, H. Suresh, V. Veitch, M. Vladymyrov, X. Wang, K. Webster, S. Yadlowsky, T. Yun, X. Zhai, and D. Sculley, *Underspecification Presents Challenges for Credibility in Modern Machine Learning*, Nov. 2020. DOI: 10.48550/arXiv.2011.03395

[48] G. Gokhale, B. Claessens, and C. Develder, "Physics informed neural networks for control oriented thermal modeling of buildings", *Applied Energy*, vol. 314, p. 118 852, May 2022, ISSN: 03062619. DOI: 10.1016/j.apenergy.2022.118852

[49] J. Wang, Y. Jiang, C. Y. Tang, and L. Song, "Development and validation of a second-order thermal network model for residential buildings", *Applied Energy*, vol. 306, p. 118 124, Jan. 2022, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2021.118124

[50] V. Marty-Jourjon, A. Goyal, T. Berthou, and P. Stabat, "Identifiability study of an RC building model based on the standard ISO13790", *Energy and Buildings*, vol. 276, p. 112 446, Dec. 2022, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2022.112446

[51] S. Rouchier, M. Rabouille, and P. Oberlé, "Calibration of simplified building energy models for parameter estimation and forecasting: Stochastic versus deterministic modelling", *Building and Environment*, vol. 134, pp. 181–190, Apr. 2018, ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2018.02.043

[52] X. Yu, L. Georges, and L. Imsland, "Data pre-processing and optimization techniques for stochastic and deterministic low-order grey-box models of residential buildings", *Energy and Buildings*, vol. 236, p. 110 775, Apr. 2021, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2021.110775

[53] E. M. Ryan and T. F. Sanquist, "Validation of building energy modeling tools under idealized and realistic conditions", *Energy and Buildings*, vol. 47, pp. 375–382, Apr. 2012, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2011.12.020

[54] GeoSphere Austria, *Messstationen Stundendaten v2*, Feb. 2024. DOI: 10.60669/9BDM-YQ93

[55] K. S. Anderson, C. W. Hansen, W. F. Holmgren, A. R. Jensen, M. A. Mikofski, and A. Driesse, “Pvlib python: 2023 project update”, *Journal of Open Source Software*, vol. 8, no. 92, p. 5994, Dec. 2023, ISSN: 2475-9066. DOI: 10.21105/joss.05994

[56] S. Xu, Y. Dai, C. Yan, Z. Sun, R. Huang, D. Guo, and G. Yang, “On the preprocessing of physics-informed neural networks: How to better utilize data in fluid mechanics”, *Journal of Computational Physics*, vol. 528, p. 113 837, May 2025, ISSN: 0021-9991. DOI: 10.1016/j.jcp.2025.113837

[57] C. Van Loan, “Computing integrals involving the matrix exponential”, *IEEE Transactions on Automatic Control*, vol. 23, no. 3, pp. 395–404, Jun. 1978, ISSN: 0018-9286. DOI: 10.1109/TAC.1978.1101743

[58] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. A. Hamprecht, Y. Bengio, and A. Courville, *On the Spectral Bias of Neural Networks*, May 2019. DOI: 10.48550/arXiv.1806.08734

[59] Z.-Q. J. Xu, Y. Zhang, T. Luo, Y. Xiao, and Z. Ma, *Frequency Principle: Fourier Analysis Sheds Light on Deep Neural Networks*, https://arxiv.org/abs/1901.06523v7, Jan. 2019. DOI: 10.4208/cicp.OA-2020-0085

[60] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng, *Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains*, Jun. 2020. DOI: 10.48550/arXiv.2006.10739

[61] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks”, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, JMLR Workshop and Conference Proceedings, Mar. 2010, pp. 249–256.

[62] S. Wang, H. Wang, J. H. Seidman, and P. Perdikaris, *Random Weight Factorization Improves the Training of Continuous Neural Representations*, Oct. 2022.

[63] W. Zhou and Y. F. Xu, *Data-Guided Physics-Informed Neural Networks for Solving Inverse Problems in Partial Differential Equations*, Jul. 2024. DOI: 10.48550/arXiv.2407.10836