Energy and AI 22 (2025) 100634

Contents lists available at ScienceDirect

Energy and AI

journal homepage: www.elsevier.com/locate/egyai

ScaleONet: Scalable and control-oriented modeling of building cluster thermal dynamics using deep operator networks — A practical case study for a Belgian district☆

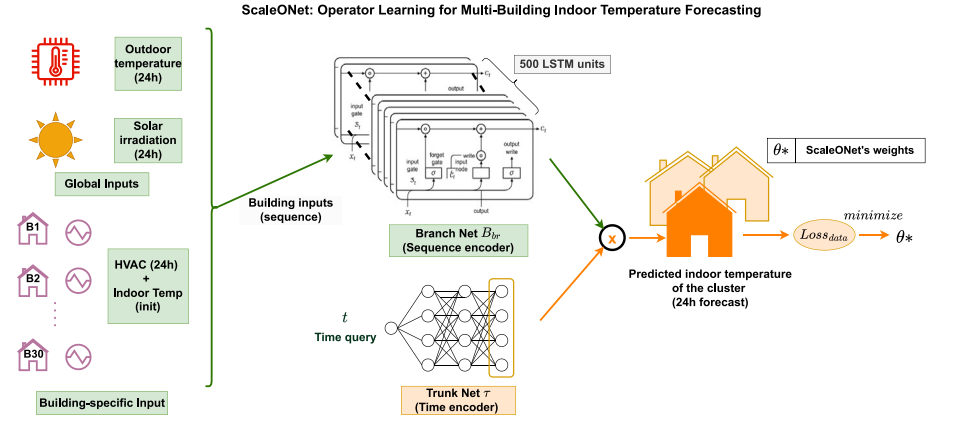
Muhammad Hafeez Saeed [a,b,*], Maomao Hu [c], Hussain Kazmi [a,b], Geert Deconinck [a,b]

[a] Department of Electrical Engineering (ELECTA-ESAT), KU Leuven, Belgium
[b] EnergyVille, Thor Park, Genk, Belgium
[c] Department of the Built Environment, College of Design and Engineering, National University of Singapore (NUS), Singapore

HIGHLIGHTS

- First DeepONet surrogate for day-ahead thermal forecasting in diverse buildings.
- Scalable branch–trunk model generalizes from 1 to 30 buildings with 50% lower RMSE.
- Predictive accuracy stays robust across varied heat-loss rates and data resolutions.
- Millisecond inference (4 ms/sample for 30 buildings) enables real-time control.
- Interpretable error patterns reveal physically consistent feature influences.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Thermal dynamics
Building clusters
Deep operator networks (deepONets)
Control-oriented modeling
Day-ahead forecasting

ABSTRACT

Delivering energy flexibility at the district scale entails coordinating control actions across many buildings to shape aggregate demand; this coordination depends on training and deploying control policies and optimization routines, which in turn require predictive models that can be queried efficiently over large building clusters. However, conventional physics-based simulators are computationally prohibitive for large-scale control training, and simple data-driven surrogates often lack the generalization needed for heterogeneous clusters. This paper introduces *ScaleONet*, a deep operator network framework designed for scalable, control-oriented modeling of building-cluster thermal dynamics. ScaleONet leverages the DeepONet paradigm to decouple and share learning across buildings: an LSTM-based branch network encodes outdoor climate and individual HVAC control signals, while a multilayer perceptron (MLP)-based trunk network embeds prediction timestamps, enabling fast predictions for growing clusters with negligible extra cost for each additional building or timestep. To the authors' knowledge, this is the first operator-learning method tailored to indoor air temperature forecasting in heterogeneous building clusters. Validation on thirty Belgian buildings (GenkNet) simulated in Dymola shows that, although a non-operator-learning LSTM baseline slightly outperforms ScaleONet for single-building cases, its error grows monotonically with cluster size. In contrast, ScaleONet's

☆ This article is part of a Special issue entitled: 'AI for Predictive Analysis' published in Energy and AI.

* Corresponding author at: Department of Electrical Engineering (ELECTA-ESAT), KU Leuven, Belgium.
E-mail address: muhammadhafeez.saeed@kuleuven.be (M.H. Saeed).

https://doi.org/10.1016/j.egyai.2025.100634
Received 20 August 2025; Received in revised form 7 October 2025; Accepted 17 October 2025
Available online 28 October 2025
2666-5468/© 2025 Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

median per-building-per-day RMSE decreases from 0.59 °C at three buildings to 0.53 °C at ten and 0.47 °C at thirty, compared to 0.95 °C for the LSTM at thirty buildings — a 51% reduction in prediction error. Error analysis across envelope heat-loss coefficients ($UA_{building}$) further reveals that while the LSTM's RMSE increases for high-UA structures, ScaleONet maintains uniformly low error. With millisecond-scale inference (approximately 4 ms per sample for thirty buildings), ScaleONet is well suited for large-scale reinforcement learning, receding-horizon optimization, and real-time model predictive control.

1. Introduction

The global pledge under the Paris Agreement [1] to limit the rise in average global temperature to well below 2 °C above pre-industrial levels has driven many countries to adopt decarbonization strategies. In pursuit of this target, numerous nations have pledged to achieve net-zero emissions in the energy sector by 2050 [2], and the European Union has enshrined similar objectives in its climate action framework [3]. Buildings account for a substantial share of global energy-related emissions — approximately 30% of final energy consumption and 26% of CO_2 emissions [4]. To remain on a net-zero trajectory by mid-century, it is estimated that 20% of existing buildings worldwide must meet zero-carbon-ready standards by 2030 [5]. In this context, the rapid deployment of rooftop photovoltaics and other distributed energy resources — such as batteries, heat pumps, and electric vehicles — has emerged as a cornerstone of building decarbonization and electrification [6]. However, the resulting increase in electrified load places significant stress on distribution networks, owing to the variability of solar generation (the "duck curve") and rising household electricity consumption and peak loads [7–9]. Consequently, demand-side energy flexibility mechanisms are required to reshape residential load profiles, support grid stability, preserve occupant comfort, and deliver economic incentives to end users [10].

Energy flexibility control methods can be classified into two broad categories: model-free and model-based approaches. Model-free techniques, including rule-based control and model-free reinforcement learning [11,12], rely exclusively on historical or real-time measurements without incorporating an explicit representation of building dynamics. Rule-based control typically encodes pre-defined heuristics or operational schedules derived from expert knowledge or empirical observations, offering simplicity and ease of deployment but limited adaptability to unseen operating conditions [13]. Model-free reinforcement learning, in contrast, adapts control policies directly from interactions with the environment or from recorded trajectories, enabling the handling of complex, non-linear relationships, albeit with high data requirements and potentially long convergence times [14].

In contrast, model-based frameworks integrate a dynamic model of the building into the decision-making process, employing strategies such as model predictive control (MPC) [15] and model-based reinforcement learning (MBRL) [16]. MPC optimizes control inputs over a receding time horizon by repeatedly solving a constrained optimization problem, benefiting from accurate predictions of system evolution but requiring computationally tractable models and solvers [17]. MBRL similarly leverages a predictive model of the building to guide policy learning, but combines it with reinforcement learning to improve adaptability and sample efficiency compared to purely model-free methods. In both cases, the incorporation of building dynamics enables anticipatory decision-making and the explicit handling of operational constraints [16,18]. The present research study focuses on model-based control.

Traditional white-box simulators remain the most accurate approach for simulating the thermal dynamics of building clusters; however, their computational expense inhibits district-level modeling and real-time control of large, heterogeneous ensembles [19,20]. Data-driven surrogates of districts such as the Aggregate Thermal Dynamics Model (ATDM) approximate the ensemble as a linear, time-invariant system under constant ventilation, but these surrogates omit detailed non-linear thermal dynamics within buildings and the operational characteristics of HVAC systems, limiting their applicability in precise control frameworks [21]. Purely data-driven approaches typically require one neural network per building, which undermines both scalability and practical deployment in extensive clusters [22].

Related efforts have employed a two-state Resistance–Capacitance (RC) network (gray-box model), a linear physics-informed representation whose parameters are calibrated from data to approximate building thermal dynamics, to represent a thousand-building cluster alongside a three-layer hierarchical controller for ancillary-service bidding [23]. However, the linear RC approximation constrains the fidelity of flexibility characterization in complex, heterogeneous ensembles. Accordingly, scalable methods capable of capturing nonlinear thermal dynamics over multi-step horizons while maintaining computational tractability are required. In another work, Dyna-PINN, a model-based physics-informed reinforcement learning approach has been presented to control heating system of a single building facing low diversity-data regimes, a practical issue in building control domain [24]. However, scaling Dyna-PINN to large heterogeneous cluster of buildings can be expensive as it involves training a physics-informed neural network (PINN) based model of each building, which also needs identification of two state RC model parameter, so logistically and computationally this process can be expensive.

In all the mentioned cases above, the challenge is a building-level model is needed for accurate thermal dynamics modeling, and the number of required models scale linearly with number of buildings for model-based control, which makes its application limited on the scale of tens and hundreds of buildings at a district-level. Therefore, a new kind of model that is representative of multi-buildings thermal dynamics evolution by using individual HVAC signal and ambient conditions is needed for mass rollout of model-based control.

Deep Operator Networks (DeepONets), as introduced by Lu et al. [25], constitute a paradigm shift from conventional neural networks, which learn mappings between finite-dimensional inputs and outputs, to operator learning frameworks that approximate mappings between infinite-dimensional function spaces. A DeepONet decomposes the operator approximation into two components: a *branch* network that encodes a sampled input function (e.g., time-varying external temperature or occupancy trajectories) into a finite representation, and a *trunk* network that embeds the query location or parameter (e.g., prediction timestamp or spatial coordinate); their outputs are combined, typically via a dot product, to produce the operator's evaluation [26].

By directly learning operators, DeepONets generalize across diverse input functions and boundary or initial condition profiles without retraining for each new scenario. Here, diverse input functions refer to varying system forcings — such as external temperature trajectories or occupancy schedules — while boundary or initial condition profiles denote different starting thermal states or imposed heat flux patterns. This capability has been demonstrated in the efficient solution of partial differential equations, where DeepONets eliminate the need to re-solve for each configuration and thus reduce computational expense compared to traditional numerical methods [27]. Applications include stress distribution in small-strain plastic deformation under variable loads and material properties [28], physics-informed modeling of lithium-ion battery electrochemistry [29], heat conduction with parameterized heat sources [30], and multiscale bubble growth dynamics that bridge microscale stochastic and continuum regimes via a composite DeepONet framework [31].

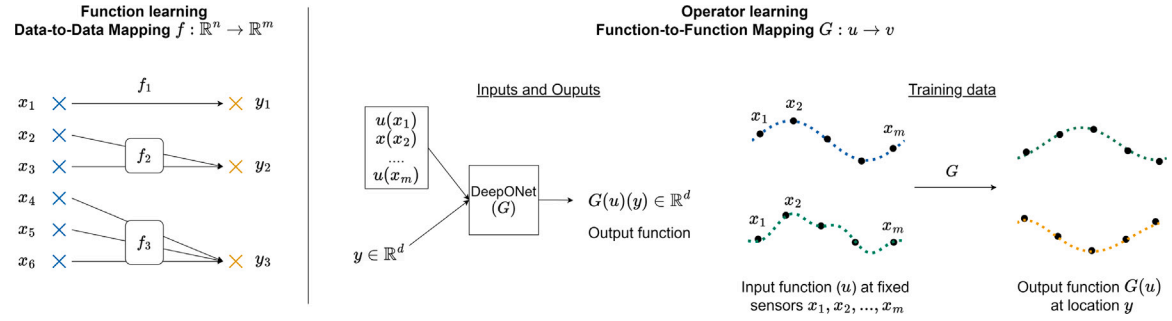


Fig. 1. Schematic illustration of the difference between learning a function and an operator [25]

Deep operator networks (DeepONets) have been deployed as fast surrogates for PDE-governed systems across mechanics and related domains; for example, Geom-DeepONet extends trunk encodings to variable 3D geometries and reports substantial speed-ups over finite-element solvers while maintaining accuracy [32]. Yet district-scale thermal modeling of heterogeneous building clusters with DeepONets remains largely unexplored. Prior causal-DeepONet work targets single-building, short-horizon forecasting of room's thermal dynamics (up to four steps) and adopts a branch-per-input design whose cost scales with the number of input samples, limiting scalability to long horizons and multi-building settings [33]. Related studies underscore operator-learning's promise in other contexts — physics-informed DeepONets have demonstrated real-time static structural response prediction at mesh resolution [34], and DeepONet has been used as a surrogate to approximate MPC control actions in an industrial refrigeration plant, achieving near-MPC performance with reduced online computation [35] — but they do not address district-scale, heterogeneous building-cluster thermal forecasting.

In this study, ScaleONet is introduced as the first scalable, control-oriented surrogate framework for modeling cluster-level thermal dynamics using DeepONets. By embedding the nonlinear heat-transfer behavior of a heterogeneous ensemble of thirty buildings — modeled on a real district in Genk, Belgium — within a single operator, ScaleONet delivers accuracy comparable to high-fidelity Dymola simulations. It supports multi-step state estimation essential for model-based control and reinforcement learning, and preserves predictive accuracy when both input functions and output trajectories are represented with low temporal resolutions, whereas conventional Long Short-Term Memory (LSTM)-based surrogates degrade under these conditions [36]. This study builds upon our previous MLP-based DeepONet surrogate for three-building clusters using low-fidelity RC model data [37].

The main contributions of this paper are:

- The first DeepONet-based surrogate for day-ahead, multi-step thermal forecasting in heterogeneous building clusters.
- Validation of the surrogate's fidelity through direct comparison with detailed white-box Dymola simulations on thirty Belgian buildings (GenkNet).
- A scalable branch–trunk architecture that generalizes linearly from single buildings to clusters of up to thirty, yielding a 51.1% reduction in median per-building RMSE versus an LSTM at the largest scale.
- Validation of robustness across a spectrum of envelope heat-loss coefficients ($UA_{building}$) and across branch/trunk resolutions, confirming consistent accuracy under varying thermal characteristics and data granularities.
- Millisecond-scale inference performance (approximately 4 ms per day-ahead sample for thirty buildings) versus 590 s per sample in Dymola (approximately 10 min), enabling real-time planning workflows such as policy-gradient reinforcement learning and receding-horizon optimization.

The paper is organized into the following sections: Methodology, Experimental Setup, Results, Discussion, and Conclusion.

2. Methodology

2.1. Artificial neural networks

Artificial neural networks (ANNs) are computational models inspired by biological neural systems [38]. By adjusting their parameters through data-driven training, ANNs can approximate complex linear and nonlinear mappings between input and output variables, enabling applications such as time-series forecasting and classification [39]. Common architectures include feed-forward networks, radial basis function networks, Kohonen self-organizing maps, and recurrent networks — among which long short-term memory (LSTM) models are particularly effective for sequential data. In this work, an LSTM serves as the baseline model; its architecture and operation are described in the following subsections. Owing to their capacity to capture intricate system dynamics, ANNs are widely employed in deep learning and in reinforcement learning frameworks for optimal control [40].

2.2. DeepONets: Deep operator network

Deep operator networks (DeepONets) were first introduced by Lu et al. [25]. Unlike standard neural networks that learn a single function mapping finite-dimensional input vectors to outputs, DeepONets learn an *operator*: a mapping that takes an entire input function as input and returns an output function. For example, given a time-varying external temperature trajectory (an input function), the operator produces the corresponding indoor temperature evolution (an output function) without retraining for each new trajectory. The architecture comprises two subnetworks: a branch net that encodes the input function sampled at m discrete points into a finite representation, and a trunk net that embeds the query location or parameter (e.g., the time at which the output function is evaluated); their outputs are combined — commonly via a dot product — to yield the operator's evaluation at that query.

To make this distinction more tangible, Fig. 1 illustrates the difference between conventional function learning (data-to-data mapping) and operator learning (function-to-function mapping) [41]. This schematic helps clarify how DeepONets generalize entire trajectories rather than fitting pointwise mappings, which is central to their effectiveness in dynamical system modeling.

Building on this framework, ScaleONet learns the operator governing the thermal dynamics of heterogeneous building clusters, delivering fast, physics-informed multi-step forecasts for applications such as thermal resilience analysis, model-based reinforcement learning, and receding-horizon studies. The ScaleONet architecture is detailed in the following subsection.

2.3. ScaleONet

ScaleONet is a Deep Operator Network that maps known 24 h input trajectories — exogenous signals (outdoor air temperature, solar irradiance) and a given HVAC control schedule — into day-ahead indoor-temperature forecasts for a cluster of B buildings. It assumes the

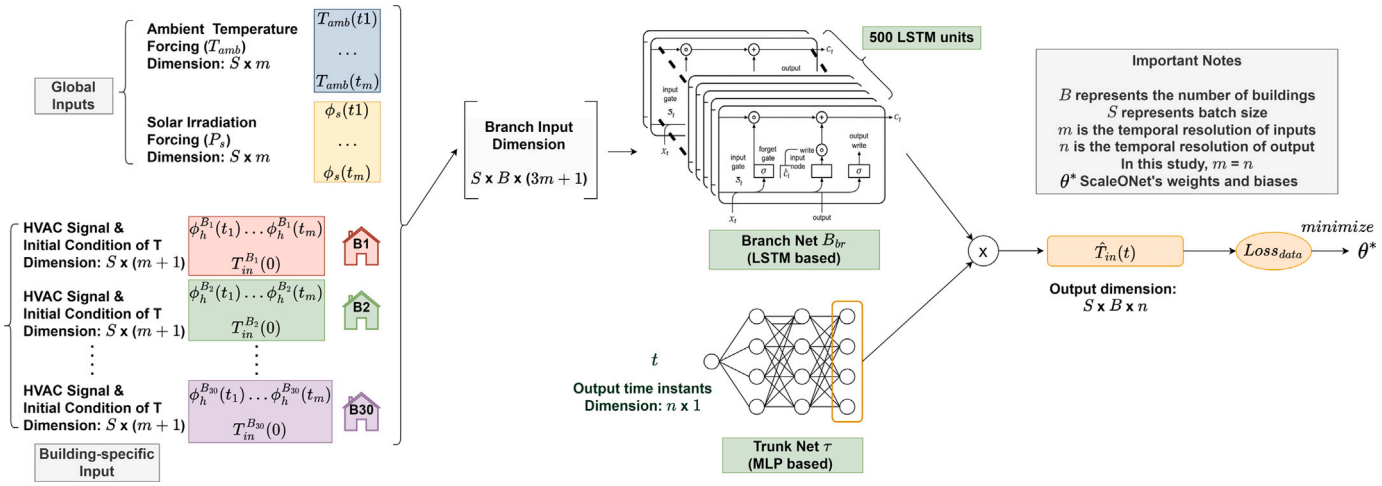


Fig. 2. Schematic of working of ScaleONet.

HVAC schedule and exogenous inputs are provided over the prediction horizon and focuses on indoor-temperature forecasting. The schematic is shown in Fig. 2. Inputs are identical to those used in the LSTM baseline.

2.3.1. Notation

- S: batch size (number of samples per update)
- B: number of buildings
- m: input samples per day; here $m = 50$ (28.8 min sampling over 24 h)
- n: forecast instants per day; here $n = 50$ (same sampling as inputs, so $n = m$)
- p: latent dimension, i.e., the size of the learned feature representation in the intermediate space where the branch and trunk outputs are combined; this is a non-physical hyperparameter set to $p = 500$

2.3.2. Branch network (LSTM)

The branch network extracts per-building *latent features*, meaning non-physical, learned representations generated by the model rather than directly measurable quantities, from time-series inputs, capturing global trends and building-specific dynamics. These features are projected into a latent space of dimension p (defined in Section 2.3.1), matching the trunk network output dimension for subsequent inner-product combination in the DeepONet architecture. It processes:

- Global input functions (shared across buildings), sampled at the same m time steps: outdoor air temperature and solar irradiance.
- Building-specific input functions, sampled at the same m time steps: HVAC power schedule, plus the initial indoor air temperature (a single scalar per building). The inclusion of the initial indoor temperature yields a per-building input length of $m + 1$.

Let

$$X_{\text{global}} \in \mathbb{R}^{S\times m\times 2}, \qquad X_{\text{building}} \in \mathbb{R}^{S\times B\times(m+1)}.$$

Flatten and tile the global inputs across buildings, then concatenate with per-building inputs:

$$X_{\text{global,flat}} = \text{reshape}\big(X_{\text{global}}, (S, 2\ m)\big),$$
$$X_{\text{global,tile}} = \text{tile}\big(X_{\text{global,flat}}, (1, B, 1)\big) \in \mathbb{R}^{S\times B\times 2m},$$
$$U = \big[X_{\text{global,tile}},\ X_{\text{building}}\big] \in \mathbb{R}^{S\times B\times(3m+1)}.$$

An LSTM with hidden size $p = 500$ (same as the latent dimension above) is applied per building (shared weights), treating each $U_{:,b,:}$ as a single-step sequence with feature length $3m + 1$:

$$B_{br} = \text{LSTM}(U) \in \mathbb{R}^{S\times B\times 500}. \tag{1}$$

Here, the LSTM acts as a nonlinear encoder of full input functions rather than as an autoregressive temporal model. This ensures consistency with the operator-learning setting, where complete input trajectories are provided at once.

2.3.3. Trunk network (MLP with cyclical encoding)

The trunk network encodes each forecast timestamp — treated as a coordinate in the time domain — into a latent vector. Because these inputs represent evaluation points rather than a sequence with temporal dependencies, and can be irregularly spaced, a feedforward network or Multi-Layer Perceptron (MLP) is sufficient. First, the raw timestamps are normalized and mapped to a cyclic representation to capture daily periodicity. A single time grid is used for all samples:

$$t \in \mathbb{R}^n \quad \big(\text{forecast times for } n \text{ steps}\big),$$

Normalize to $[0, 1]$ and apply a cyclic map:

$$t_j^{\text{norm}} = \frac{t_j - \min_{j'} t_{j'}}{\max_{j'} t_{j'} - \min_{j'} t_{j'}}, \quad \theta_j = 2\pi\, t_j^{\text{norm}}, \quad t_{\text{enc},j} = \big[\sin(\theta_j), \cos(\theta_j)\big],$$

so

$$t_{\text{enc}} \in \mathbb{R}^{n\times 2}.$$

An MLP then projects these two-dimensional cyclic encodings into the latent space of dimension $p = 500$ (a non-physical, learned feature dimension as defined in Section 2.3.1):

$$\tau = \text{MLP}(t_{\text{enc}}) \in \mathbb{R}^{n\times 500}. \tag{2}$$

2.3.4. Operator fusion and training

The final prediction is obtained by fusing the per-building latent features from the branch network with the time-step latent features from the trunk network. Specifically, for each sample s, building b, and forecast instant j, the predicted indoor air temperature $\hat{T}_{\text{in},s,b,j}$ is computed as the inner product over the shared latent dimension $p = 500$:

$$\hat{T}_{\text{in},\,s,b,j} = \sum_{\ell=1}^{500} B_{br,\,s,b,\ell}\ \tau_{s,j,\ell}. \tag{3}$$

Here, $B_{br} \in \mathbb{R}^{S\times B\times p}$ contains the building-specific embeddings produced by the LSTM branch, and $\tau \in \mathbb{R}^{S\times n\times p}$ contains the time-step embeddings from the MLP trunk. This yields the predicted indoor air temperature $\hat{T}_{\text{in}} \in \mathbb{R}^{S\times B\times n}$.

All network parameters θ (weights and biases of both branch and trunk) are learned jointly by minimizing the mean squared error between the predictions and the ground-truth indoor air temperatures,

obtained from Dymola simulations of the GenkNet building models described in Section 3.1.

$$\mathcal{L}(\theta) = \frac{1}{S \cdot B \cdot n} \sum_{s=1}^{S} \sum_{b=1}^{B} \sum_{j=1}^{n} \left(\hat{T}_{\text{in},s,b,j} - T_{\text{in},s,b,j}\right)^2. \tag{4}$$

A complete description of the training loop and parameter updates is provided in Algorithm 1.

Algorithm 1 Training ScaleONet

Require: Batch $\{X_{\text{global}}, X_{\text{building}}, T_{\text{in}}\}$ with shapes $(S, m, 2)$, $(S, B, m{+}1)$, (S, B, n)
Require: Time grid $T_{\text{raw}} \in \mathbb{R}^n$
Require: BranchNet: LSTM(hidden size p=500); TrunkNet: MLP(output dim p=500)
Require: Optimizer $\mathcal{O}$, learning rate α
for epoch = 1 **to** N_{epochs} **do**
  // Trunk: encode times (shared for all s)
  $t^{\text{norm}} \leftarrow (T_{\text{raw}} - \min T_{\text{raw}})/(\max T_{\text{raw}} - \min T_{\text{raw}})$
  $\varphi \leftarrow 2\pi\, t^{\text{norm}}$
  $T_{\text{enc}} \leftarrow [\sin\varphi,\ \cos\varphi] \in \mathbb{R}^{n\times 2}$
  $\tau \leftarrow \text{TrunkNet}(T_{\text{enc}}) \in \mathbb{R}^{n\times 500}$
  // Branch: prepare inputs and encode
  $X_{\text{global,flat}} \leftarrow \text{reshape}(X_{\text{global}}, (S, 2m))$
  $X_{\text{global,tile}} \leftarrow \text{tile}(X_{\text{global,flat}}, (1, B, 1)) \in \mathbb{R}^{S\times B\times 2m}$
  $U \leftarrow \text{concat}(X_{\text{global,tile}}, X_{\text{building}}) \in \mathbb{R}^{S\times B\times(3m+1)}$
  $B_{\text{latent}} \leftarrow \text{BranchNet}(U) \in \mathbb{R}^{S\times B\times 500}$
  // Fuse and predict (broadcast τ over S)
  $\hat{T}_{\text{in}} \leftarrow \text{einsum}('sbl, jl \to sbj',\ B_{\text{latent}},\ \tau) \in \mathbb{R}^{S\times B\times n}$
  // Loss and update
  $\mathcal{L} \leftarrow \frac{1}{SBn} \|\hat{T}_{\text{in}} - T_{\text{in}}\|^2$
  $\theta \leftarrow \mathcal{O}(\theta, \nabla_\theta \mathcal{L}, \alpha)$
end for

2.4. Baselines

2.4.1. Persistence model

The persistence model serves as a naïve baseline to control for dataset-specific artifacts and to ensure that performance gains of ScaleONet and the LSTM baseline reflect genuine model capabilities rather than data biases. It assumes that the indoor temperature remains constant over the forecast horizon by repeating the initial value at each prediction step.

For day d and building b, let $y_{d,b,0}$ denote the indoor temperature at the first sampled instant, and $n = 50$ the number of uniformly spaced samples per day. The persistence prediction is then

$$\tilde{y}_{d,b,t} = y_{d,b,0}, \quad t = 1, \ldots, n. \tag{5}$$

The RMSE of this baseline is defined in Section 3.3 and is used to compute the normalized RMSE, providing a relative error metric that accounts for day-to-day and building-specific variability and enables fair comparison of surrogate performance.

2.4.2. RC model baseline

A simplified two-state resistance–capacitance (RC) model is employed as a physics-informed baseline. The model consists of an indoor air node and an envelope node, coupled by thermal resistances, with outdoor temperature and solar irradiation as exogenous inputs and HVAC heat flow as a controllable input (Fig. 3). Model parameters are tuned for each building cluster by minimizing the mean squared error between simulated and reference indoor air temperatures on the training set. This baseline provides a parsimonious gray-box comparison against data-driven approaches.

2.4.3. LSTM baseline

This baseline applies a single-layer LSTM followed by a linear projection to forecast, for each building, the next n steps. It uses the same inputs as the branch network: global (outdoor air temperature, solar irradiance) and building-specific (HVAC schedule, initial indoor temperature). Here, the LSTM acts as a nonlinear encoder of the concatenated input functions rather than as an autoregressive sequence model. This ensures that the baseline consumes the same functional inputs as ScaleONet, making the comparison consistent across models.

Let S be the mini-batch size, B the number of buildings, m the input function resolution (sequence length), and n the forecast horizon (set to $n = m$). For each sample, construct

$$x \in \mathbb{R}^{(3m+1)B}$$

by concatenating, for each building, the flattened global signals ($2m$), the HVAC sequence (m), and the initial indoor temperature ($+1$). The model consumes the single-step sequence $(1 \times (3m+1)B)$ per sample:

$$h = \text{LSTM}(x), \qquad h \in \mathbb{R}^{500}. \tag{6}$$

A linear map produces a flat vector of length $B \times n$, reshaped to (B, n):

$$\hat{t}_{\text{flat}} = W h + b, \quad \hat{t}_{\text{flat}} \in \mathbb{R}^{Bn}, \qquad \hat{T}_{\text{in}} = \text{reshape}(\hat{t}_{\text{flat}}; B, n). \tag{7}$$

Over a batch, the output is $\hat{T}_{\text{in}} \in \mathbb{R}^{S\times B\times n}$. Training minimizes MSE:

$$\mathcal{L}(\theta) = \frac{1}{S \cdot B \cdot n} \sum_{s=1}^{S} \sum_{b=1}^{B} \sum_{j=1}^{n} \left(\hat{T}_{\text{in},s,b,j} - T_{\text{in},s,b,j}\right)^2. \tag{8}$$

2.4.4. MLP-ScaleONet

To assess the role of the sequence encoder in the branch network, a variant of ScaleONet with a multi-layer perceptron (MLP) branch is implemented. This model retains the same trunk network as the LSTM-based ScaleONet but replaces the LSTM branch with a feedforward MLP of comparable capacity. The MLP-based variant allows isolating the effect of temporal sequence encoding and serves as an ablation study.

3. Experimental setup

3.1. Building models: GenkNet

The experimental study employs thirty single-family dwellings in Genk, Belgium, selected from the GenkNET dataset [42,43] and modeled in Modelica using the IDEAS library [44]. Modelica is an object-oriented, equation-based modeling language for multi-domain physical systems [45], while Dymola is a commercial simulation environment that enables the implementation, compilation, and execution of Modelica models [46]. Each dwelling is equipped with a radiator served by an electric heater for water heating, and a circulating pump for controlling the mass flow rate, governed by a hysteresis controller that maintains indoor temperature between 21 °C and 24 °C (deadband 3 °C). The binary controller activates heating when the zone temperature falls below 21 °C and deactivates it upon exceeding 24 °C.

Each dwelling is represented as a single-zone thermal envelope with parameterized wall, roof, and window areas, conditioned floor area, and nominal heat-loss characteristics using the IDEAS library in Modelica. Each dwelling includes four windows, with one located on each outer wall. Their surface areas, inclinations, and azimuths follow the geometrical specifications in the GenkNET building envelope dataset implemented in Dymola [42]. The heating loop comprises the same components described above. Outdoor temperature and solar irradiation time series drive the envelope heat fluxes, while the pump–radiator assembly captures flow-dependent thermal transfer without additional HVAC control complexity. Fig. 4 shows the Dymola schematic, including the heating element and mass-flow control for radiator operation.

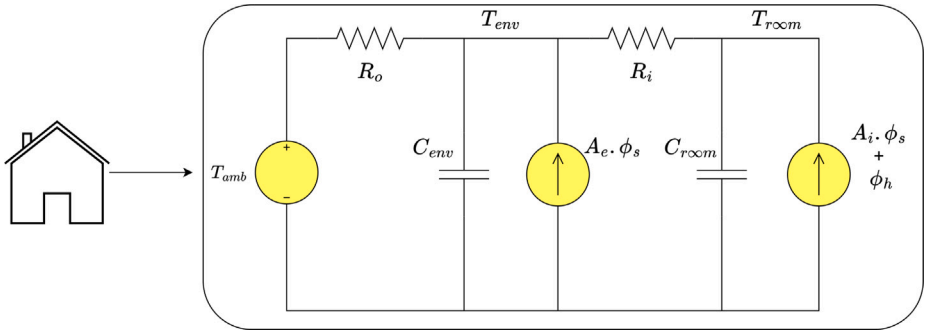


Fig. 3. Two-state resistance–capacitance (2R2C) thermal model with split solar and HVAC gains. The envelope node (T_{env}) and room air node (T_{room}) are coupled through resistances R_o and R_i, with heat capacities C_{env} and C_{room}. Solar irradiation ϕ_s contributes via envelope and indoor gains ($A_e \cdot \phi_s, A_i \cdot \phi_s$), while HVAC input ϕ_h acts directly on the room node. Outdoor air temperature (T_{amb}) provides the boundary condition.

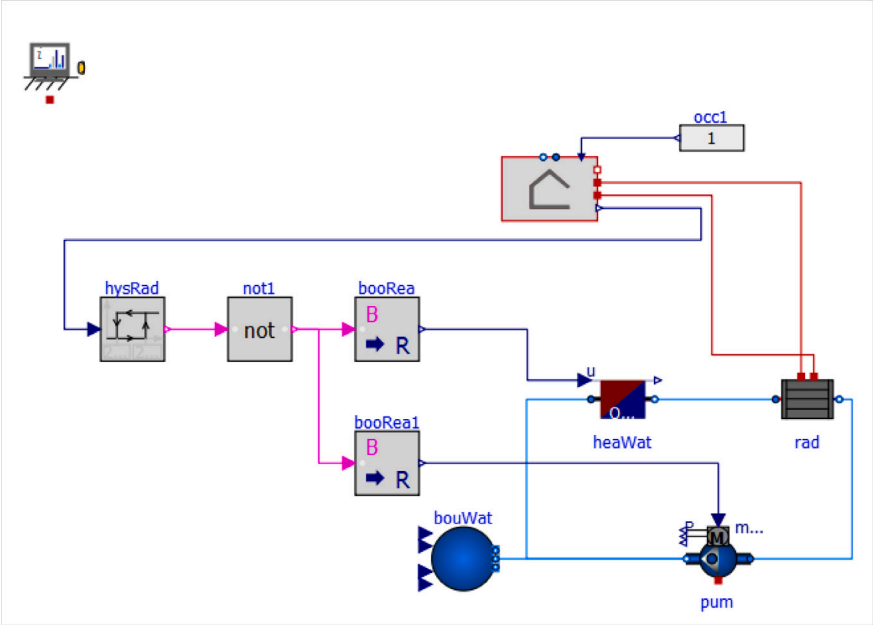


Fig. 4. Single-zone dwelling model from Genk, Belgium, implemented in Dymola (Modelica-based) and governed by a hysteresis controller to maintain indoor temperature between 21 °C and 24 °C. Heating loop abbreviations: `hysRad` — hysteresis controller; `heaWat` — electric water heater; `pum` — circulating pump; `rad` — radiator; `bouWat` — water boundary source. Color code: light blue — cold-water connections; red — hot-water connections; magenta — control signals; dark blue — measurement/data signals. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Summary of building envelope characteristics (N = 30). Floor area is the conditioned floor-plan area; U_{walls}, U_{roof}, $U_{windows}$ are the thermal transmittances (U-values) of the envelope elements; $UA_{building}$ is the overall heat-loss coefficient (sum of U × A for all envelope elements); $A_{env}/V_{building}$ is the external envelope surface area to internal volume ratio [42].

Property	Mean	Std	Min	Max
Floor area [m^2]	89.91	53.57	45.28	239.08
U_{walls} [W/m^2 K]	2.07	0.54	0.60	2.52
U_{roof} [W/m^2 K]	1.88	1.25	0.21	3.11
$U_{windows}$ [W/m^2 K]	3.31	0.86	1.94	4.95
$UA_{building}$ [W/K]	612.71	351.10	230.05	1830.56
$A_{env}/V_{building}$ [m^2/m^3]	0.75	0.11	0.60	1.02

These thirty dwellings, originally constructed circa 2000, exhibit a representative range of floor areas, envelope U-values, and thermal-loss characteristics for the local climate. Envelope properties include opaque U-values (walls and roof), glazed U-values (windows), the overall heat-loss coefficient UA (sum of U × A), and envelope compactness (external surface area to building volume). Table 1 summarizes these metrics in terms of mean, standard deviation, and extrema.

Fig. 5 presents a scatter plot of opaque envelope U-values (walls and roof) versus glazed U-values (windows) for the 30 dwellings, highlighting substantial variability across both parameters. The joint distribution reveals that high wall/roof U-values do not necessarily correspond to high window U-values, indicating diverse insulation characteristics between walls/roof and windows. This heterogeneity is essential for assessing ScaleONet's capability to capture both slow-varying and fast-changing thermal dynamics. Complete dwelling-level characteristics are provided in Table C.6.

3.2. Initial configurations

The forecasting setup used a 28.8-min resolution with a day-ahead (24 h) horizon, where each input and output sequence comprised 50 data points. The considered baselines were an LSTM and a persistence model. Final learning rates for both the DeepONet and LSTM models (10^{-4}) were determined through hyperparameter optimization, as detailed in Appendix A.

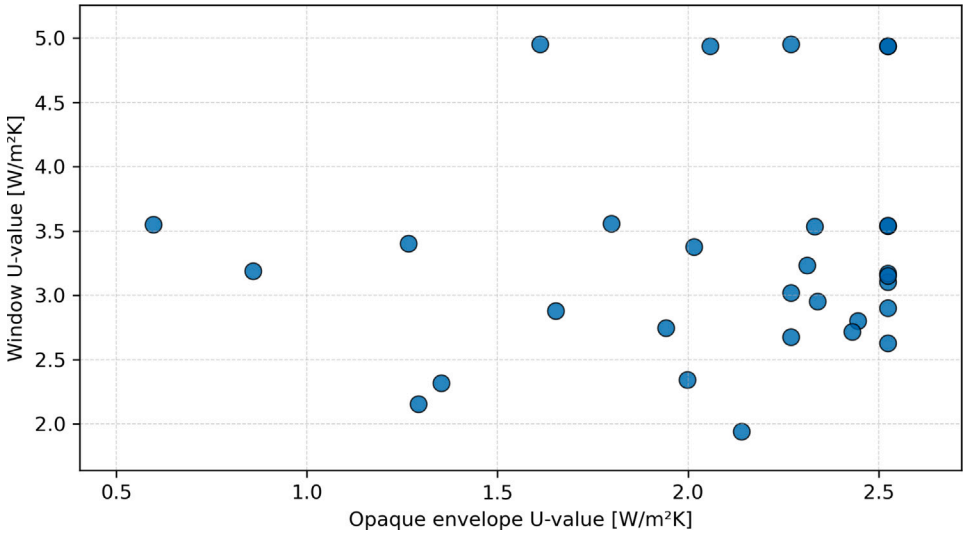


Fig. 5. Scatter plot of opaque envelope U-values (walls and roof) versus glazed U-values (windows) across the 30 dwellings.

3.3. Evaluation criteria and metrics of success

To enable a structured comparison of all methods, this subsection first states the research questions driving the study and then defines the quantitative metrics used to answer them. The following five questions guide the Results section:

1. Can a unified DeepONet surrogate accurately capture and forecast the 24-h thermal dynamics of heterogeneous building clusters under Belgian weather conditions, and how does its performance compare to conventional data-driven and physics-inspired baselines (LSTM, persistence, MLP-ScaleONet, and RC model)?
2. How does the diversity of building characteristics affect surrogate accuracy, and which approaches are most robust across cluster sizes?
3. Does DeepONet provide faster inference than a physics-based simulator such as Dymola with comparable accuracy, enabling its use in model-based reinforcement learning and MPC?
4. How does the temporal resolution of input and output functions influence DeepONet's training efficiency and generalization capability?
5. Which input drivers most strongly influence forecast errors across horizons, and do the surrogates reproduce physically consistent feature–error relationships?

Model accuracy for any surrogate m (ScaleONet or LSTM) is quantified by the per-day, per-building root-mean-square error (RMSE):

$$\mathrm{RMSE}_{d,b}^{(m)} = \sqrt{\frac{1}{T}\sum_{t=1}^{n}\left(y_{d,b,t} - \hat{y}_{d,b,t}^{(m)}\right)^2}, \tag{9}$$

where d indexes test days, b indexes buildings, and $n = 50$ is the number of uniformly spaced samples per day.

The persistence baseline RMSE, denoted by the superscript "pers" for brevity, is defined analogously:

$$\mathrm{RMSE}_{d,b}^{\mathrm{pers}} = \sqrt{\frac{1}{T}\sum_{t=1}^{n}\left(y_{d,b,t} - \tilde{y}_{d,b,t}\right)^2}, \tag{10}$$

with $\tilde{y}_{d,b,t}$ from Section 2.4.1. To quantify each surrogate's performance relative to persistence, the normalized RMSE metric is introduced:

$$E_{d,b}^{(m)} = \frac{\mathrm{RMSE}_{d,b}^{(m)}}{\mathrm{RMSE}_{d,b}^{\mathrm{pers}} + \varepsilon}, \quad \varepsilon = 10^{-8}, \tag{11}$$

where ε is a small constant to prevent division by zero. Distributions of $E_{d,b}^{(m)}$ over all test days and buildings then assess each model's relative accuracy.

All models were trained and tested under identical conditions. One-year simulation data were generated in Dymola using a detailed white-box Modelica building thermal model for thirty representative residential buildings in Genk, Belgium. The simulation was configured with 200,000 output intervals over a one-year period (31,536,000 s), resulting in a sampling interval of approximately 2.63 min (157.68-s). Of the 365 daily profiles, 292 (including 20 for validation) were used for training and 73 for testing. Each profile comprised 552 measurements; for computational efficiency, fifty uniformly spaced samples per profile were used for training and evaluation of all models, including the persistence baseline.

Training was limited to a maximum of 10,000 epochs with early stopping: if the validation loss did not improve over 16 consecutive checks (one check every 25 epochs), training was terminated to prevent overfitting. The validation-check interval and patience were determined via a comprehensive hyperparameter sweep. All models were implemented in TensorFlow, with Python functions compiled into optimized computation graphs via the `@tf.function` mechanism, thereby reducing memory usage and accelerating execution [47].

3.4. Hardware information and hyperparameter tuning results

The experimental study was conducted on a personal computer with an Intel Core i9 processor, an NVIDIA RTX A2000 GPU, and 64 GB of RAM. The software environment consisted of Python 3.8, TensorFlow 2.5, and PyTorch 2.0.1 with CUDA 11.8 support.

The hyperparameter search ranges and the final values adopted for ScaleONet, the LSTM baseline, MLP-ScaleONet, and the RC model are reported in Appendix A, Tables A.4 and A.5.

4. Results

The presented results in this section are aimed at addressing the research questions highlighted in Section 3.3.

4.1. Representative indoor air temperature profiles

To provide context for the surrogate model evaluation, Fig. 6 presents a one-day segment of simulated indoor air temperature tra-

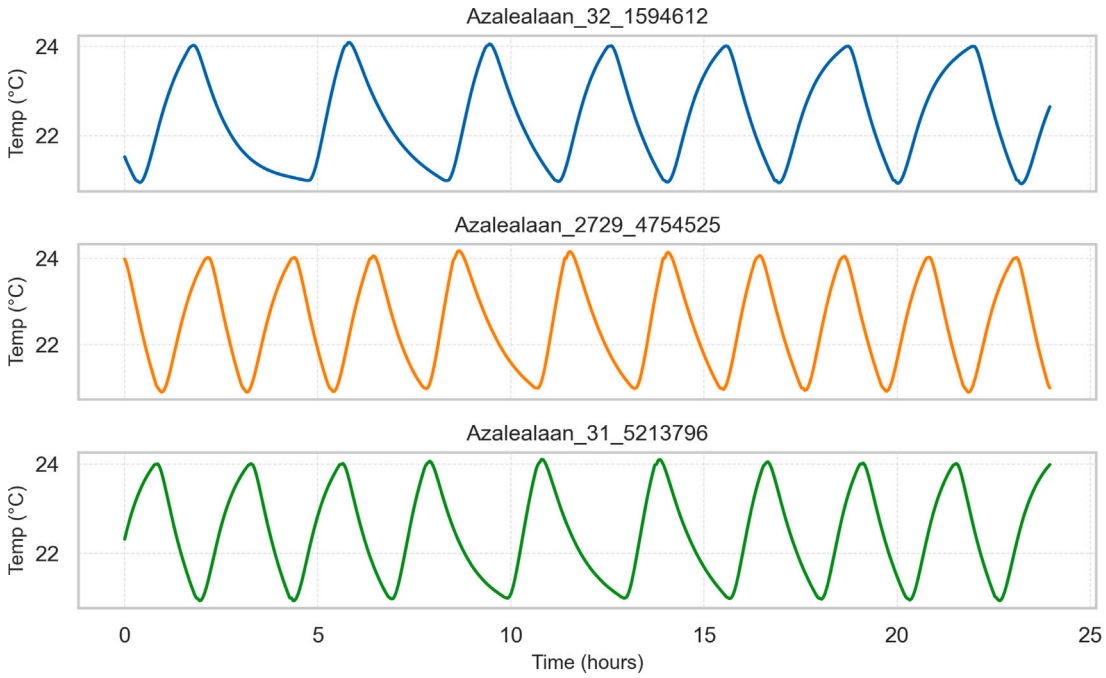


Fig. 6. Ground-truth indoor air temperature profiles over a one-day period for three representative dwellings. All dwellings are subject to hysteresis control with comfort bounds of 21 °C–24 °C, resulting in identical amplitudes but distinct heating cycle frequencies and phases.

jectories for three representative dwellings from the GenkNet dataset (Section 3.1). The dwellings were selected at random to illustrate the typical cyclic behavior under hysteresis control. As all buildings operate with identical thermal comfort bounds of 21 °C–24 °C (deadband 3 °C), the temperature amplitude is fixed. Differences in the profiles arise from variations in cycle frequency, phase, and heating duration, which are influenced by envelope thermal properties, internal gains, and weather exposure. These trajectories correspond to the target outputs of the surrogate model and provide a visual reference for the dynamics it must capture.

4.2. Normalized error analysis across cluster sizes

To control dataset bias — here meaning differences in scale, heterogeneity, or temporal patterns across clusters that could unevenly affect raw error magnitudes — each model's root mean squared error (RMSE) is normalized by the RMSE of the persistence baseline (see Eq. (11)). As shown in Fig. 7, for most cluster sizes and models the interquartile range of the normalized errors $E_{d,b}^{(m)}$ lies below one, indicating consistent improvement over persistence. The RC model forms the main exception: its median normalized RMSE remains above one across all cluster sizes, reflecting the limited ability of a parsimonious gray-box model to reproduce the short-term dynamics captured in the high-fidelity data.

At small aggregation scales (one or three buildings), the LSTM baseline yields the lowest median normalized RMSE (0.268 and 0.383, respectively), outperforming both ScaleONet (0.365 and 0.440) and the MLP-based ScaleONet variant (0.306 and 0.590). This reflects the LSTM's capacity to tightly fit low-dimensional, homogeneous dynamics. However, its error rises monotonically with cluster size — 0.557 at ten buildings and 0.703 at thirty buildings — indicating that an unconstrained sequence-to-sequence mapping struggles with heterogeneity and increased input dimensionality.

In contrast, ScaleONet exhibits an inverted U-shape: its median error peaks at three buildings, then declines to 0.399 at ten buildings and 0.359 at thirty buildings. This pattern indicates that after exposure to moderate heterogeneity, the operator-based architecture generalizes more effectively, leveraging shared structure across buildings. The MLP-based ScaleONet variant consistently underperforms the LSTM and full ScaleONet at larger cluster sizes (0.747 at ten buildings, 0.789 at thirty buildings), confirming the benefit of sequence encoding via an LSTM branch rather than a purely feedforward alternative.

Overall, these results demonstrate that while the LSTM baseline is competitive on very small clusters, operator learning via ScaleONet scales more effectively to heterogeneous building aggregations. The RC model, while interpretable and lightweight, is not competitive in predictive accuracy.

4.3. Grouped Permutation Importance (GPI) analysis

To investigate which input factors drive prediction errors, a Grouped Permutation Importance (GPI) analysis was conducted. GPI quantifies the increase in forecast error (ΔRMSE) when groups of input features are permuted, thereby indicating their relative influence on model accuracy across the forecast horizon.

Fig. 8 compares GPI maps for the LSTM baseline, ScaleONet, the MLP-based variant, and the RC model on a representative three-building cluster. A consistent pattern emerges in ScaleONet: the initial indoor temperature strongly affects short-term forecasts, but its influence decays with horizon length, while outdoor temperature and solar irradiation become increasingly important at longer horizons. HVAC inputs show moderate influence, mainly at intermediate horizons, but are less dominant than weather drivers. This shift reflects the physical reality that initial states dominate short-term dynamics, whereas exogenous weather drivers govern long-term evolution. Such patterns indicate that ScaleONet not only achieves improved predictive accuracy but also captures physically consistent feature–error relationships, providing interpretability beyond purely data-driven models.

The LSTM baseline also shows some horizon-dependent trends, such as stronger importance of initial conditions at shorter horizons and weather drivers at longer horizons, but these effects appear weaker and less clearly structured than in ScaleONet. The MLP-based ScaleONet variant exhibits a different pattern: one of the initial conditions influences the very short horizon, but beyond that the importance becomes noisy and lacks a systematic trend, suggesting a limited ability

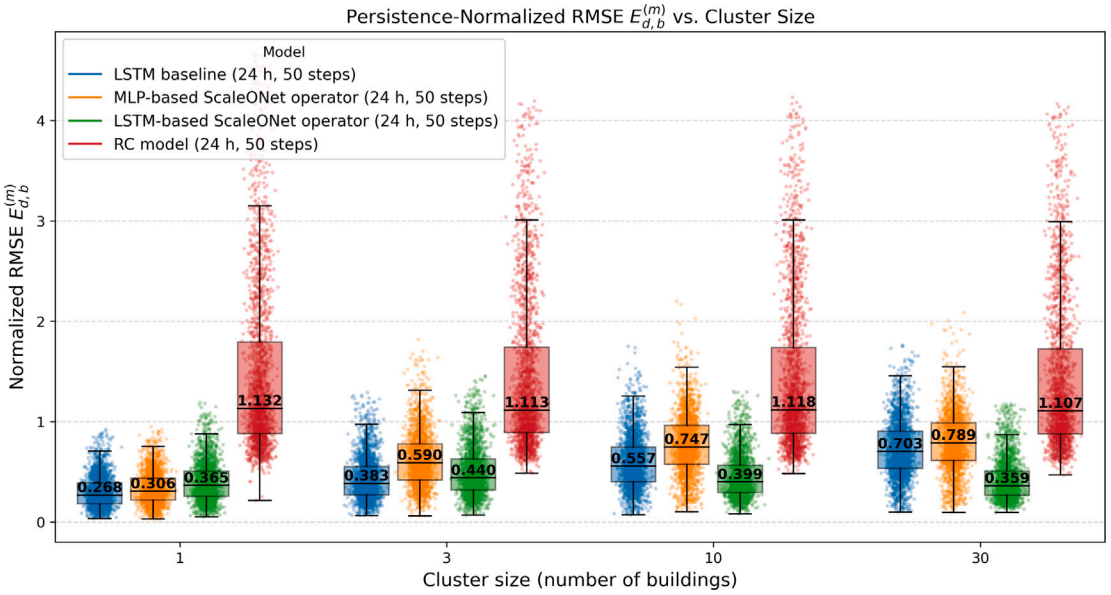


Fig. 7. Persistence-normalized multi-step prediction errors across cluster sizes for all baseline and operator-learning models (LSTM, MLP-ScaleONet, RC model) compared with ScaleONet. Lower values indicate improved performance relative to the persistence baseline.

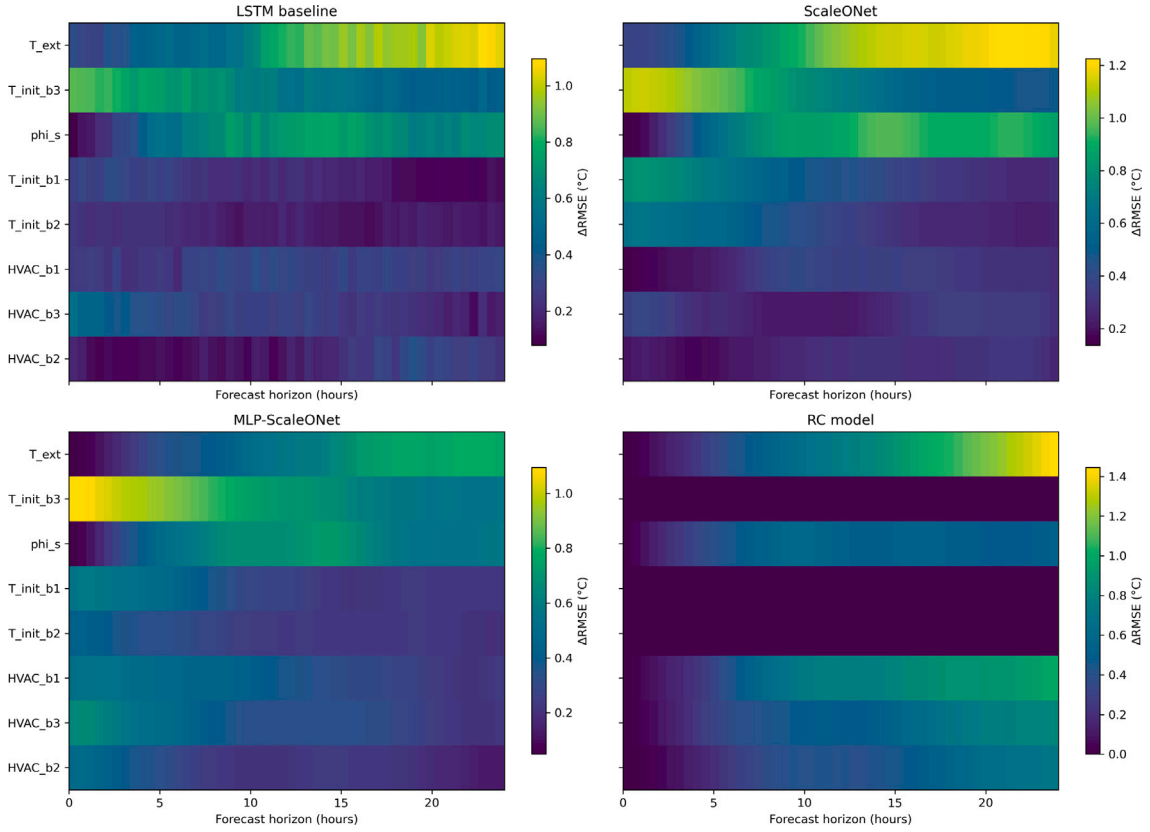


Fig. 8. Grouped Permutation Importance (GPI) maps for a three-building cluster. Each panel shows the increase in forecast error (ΔRMSE) across the horizon when input groups are permuted. ScaleONet exhibits a physically consistent transition from initial-condition dominance at short horizons to weather-driver dominance at longer horizons. The LSTM shows similar but weaker trends, while the MLP-based variant reveals only sporadic influence of initial conditions without a clear horizon-dependent structure. The RC model displays negligible importance of initial conditions, as these are propagated deterministically by its state equations.

to encode dynamics without recurrent sequence processing. In contrast, the RC model shows almost no sensitivity to initial indoor temperature in the GPI maps. This outcome is expected: the RC formulation propagates the provided initial condition deterministically through its state equations, so permuting it within the GPI framework has little measurable effect.

Overall, the analysis highlights that while all models rely on external and building-specific drivers, ScaleONet uniquely reproduces a

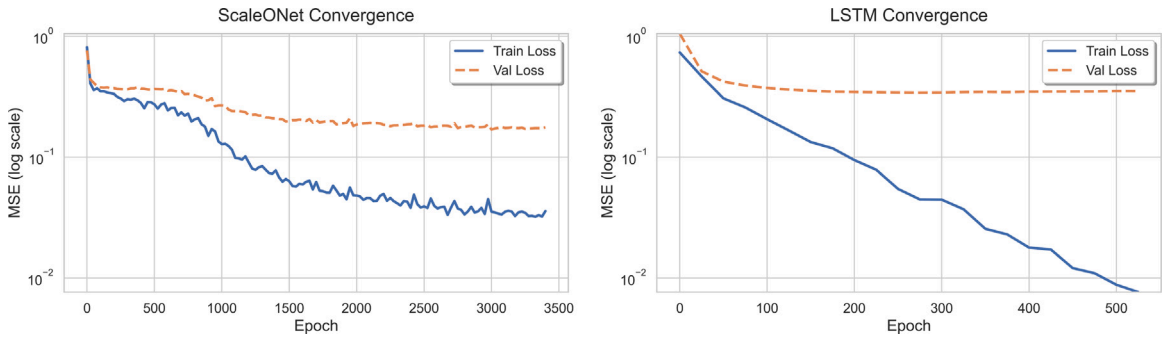


Fig. 9. Epoch-wise training and validation MSE (log scale) for ScaleONet (left) and LSTM (right). While the LSTM converges quickly on the training set but plateaus (and slightly rebounds) on validation, ScaleONet shows slower but more sustained reduction in both train and validation error, indicating better generalization to unseen building configurations.

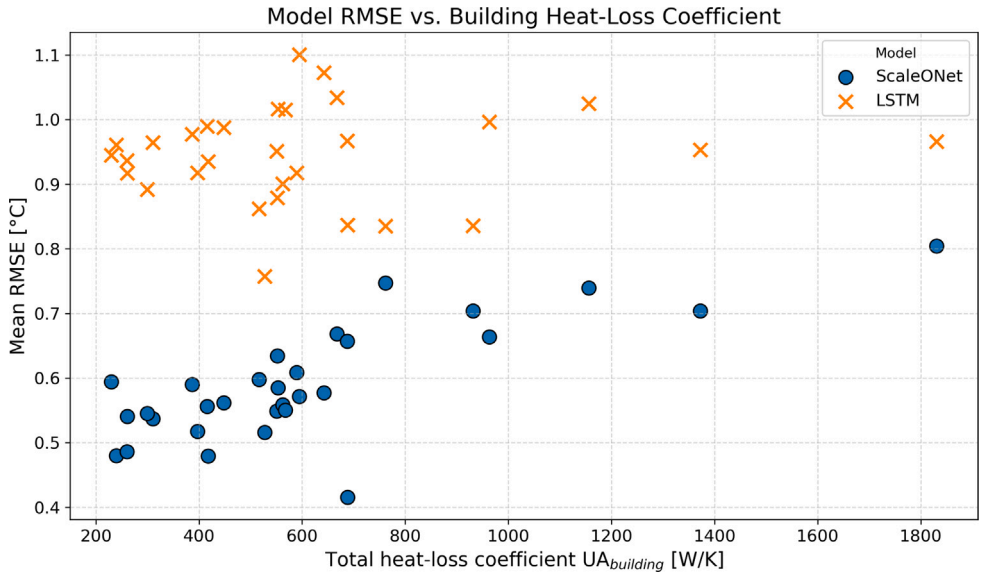


Fig. 10. Mean per-building RMSE vs. total heat-loss coefficient $UA_{building}$ for ScaleONet and the LSTM baseline.

horizon-dependent balance between initial states and exogenous inputs that aligns with physical intuition, reinforcing its advantage as both an accurate and interpretable surrogate. For completeness, the MLP-ScaleONet and RC model baselines requested by the reviewers have also been included in this analysis. However, as their results do not provide additional insights beyond what is captured by the LSTM baseline and ScaleONet, the remainder of the results section focuses on these two models to maintain clarity and brevity.

4.4. Training and validation convergence

Fig. 9 compares the epoch-wise mean squared error (MSE) on training and validation sets for ScaleONet (left) and LSTM (right), plotted on a logarithmic scale. Several observations follow:

- ScaleONet requires a larger number of epochs (around 1200) to drive the training MSE below 10^{-1}, reflecting its higher capacity and the greater difficulty of learning an operator over heterogeneous input functions. The validation MSE decreases more gradually and then plateaus around 1.5×10^{-1}; the persistent gap between training and validation loss stabilizes, suggesting that the model has settled into a consistent generalization behavior on unseen data.
- LSTM converges more rapidly — achieving sub-10^{-1} training MSE in under 200 epochs — owing to its simpler sequence-to-sequence structure and lower parameter count. However, its validation MSE plateaus near 4×10^{-1} and even trends slightly upward with further training, signifying earlier onset of overfitting and reduced ability to generalize across diverse building configurations.

Overall, while the LSTM attains low training error quickly, its validation performance degrades beyond its optimum epoch, whereas ScaleONet's slower but more sustained convergence yields more robust validation accuracy. This behavior corroborates the operator approach's superior generalization in multi-building forecasting tasks.

4.5. Prediction accuracy across total heat-loss coefficient

Fig. 10 presents the mean per-building RMSE as a function of the total heat-loss coefficient $UA_{building}$ [$W K^{-1}$] for both ScaleONet and an

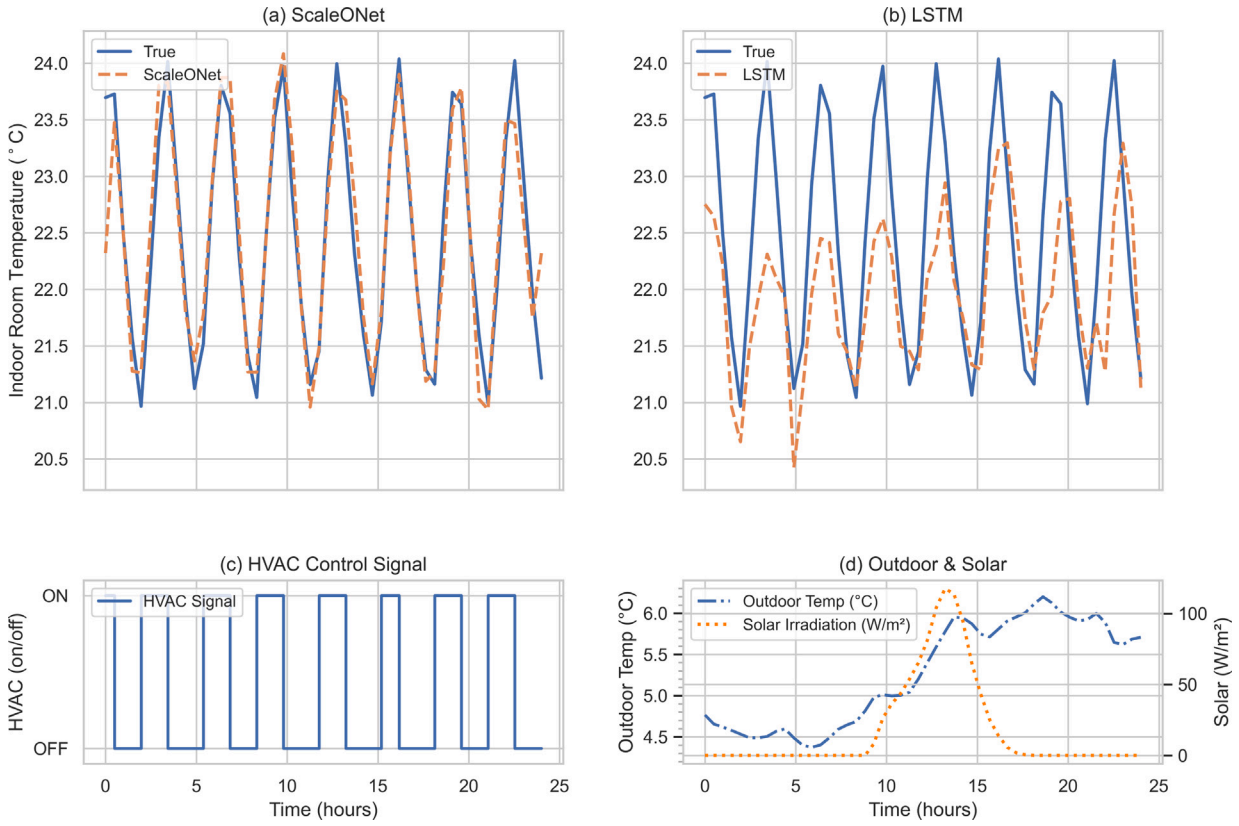


Fig. 11. True vs. predicted indoor temperature for an out-of-sample day in a 30-building cluster model. Panels (a) and (b) show temperature trajectories for a randomly chosen building predicted by ScaleONet and LSTM models, respectively, compared against ground truth. Panel (c) displays the binary HVAC control signal (ON/OFF) for the same building. Panel (d) presents the corresponding outdoor air temperature and solar irradiation profiles for that day, illustrating the environmental drivers influencing the building's thermal dynamics.

LSTM baseline. Prediction error increases with higher $UA_{building}$, reflecting the greater heat exchange and indoor–outdoor temperature coupling in more "leaky" buildings. In contrast, buildings with lower UA values — i.e. more insulated envelopes — exhibit smaller fluctuations and are correspondingly easier to predict.

Across the full UA range, ScaleONet consistently achieves lower RMSE than the LSTM. For instances with $UA_{building} < 600\ W\,K^{-1}$, ScaleONet maintains RMSE below 0.65 °C, while the LSTM shows higher variability and frequently exceeds 0.9 °C. At the high end of the UA spectrum, LSTM errors exceed 1.0 °C for many buildings, whereas ScaleONet remains below approximately 0.8 °C.

Nevertheless, the vertical spread of RMSE values at fixed UA (for example, in the 400–600 $W\,K^{-1}$ range) indicates that envelope's thermal insulation quality alone does not capture all sources of forecasting difficulty. Other factors — such as occupancy schedules, HVAC control strategies and solar gains — are acknowledged as additional influences on prediction accuracy, though they are not further elaborated here.

These results demonstrate that ScaleONet is more robust to variations in envelope tightness and outperforms the LSTM baseline across the spectrum of building heat-loss characteristics.

4.6. Qualitative test performance on a challenging day

To illustrate how the two models behave on an out-of-sample day with rapid temperature swings, Fig. 11 compares their forecasts against the true indoor temperature over a 24 h period. This day was selected because it exhibits several sharp peaks and troughs that are difficult to track.

As seen in Fig. 11, ScaleONet tracks the true trajectory closely — capturing both the timing and magnitude of peaks and valleys. In contrast, the LSTM forecast (dashed) systematically lags and under-/over-shoots at rapid transitions, despite its fast convergence during training. This qualitative result reinforces the validation-set behavior discussed in Section 4.4, and confirms that the operator-learning architecture generalizes more robustly on highly fluctuating test cases. The HVAC on/off cycles in Panel (c) reflect the hysteresis control logic maintaining indoor temperatures within the 21–24 °C deadband, resulting in discrete thermal responses that the models must capture accurately. Furthermore, the pronounced outdoor temperature variability and midday solar peak in Panel (d) impose additional disturbances on the thermal dynamics, making this test case particularly challenging for both models.

4.7. Computational performance

This subsection evaluates the computational performance of the ScaleONet and LSTM surrogates alongside the Dymola simulator to assess their suitability for use in control training and studies. One-day test refers to a day-ahead, multi-step forecast of indoor room temperature. Table 2 shows per-sample inference times for LSTM and ScaleONet versus Dymola across various cluster sizes. Both machine-learning surrogates run in the millisecond range, while Dymola requires seconds to minutes. Training cost remains under 120 ms per epoch for the LSTM, and under 360 ms per epoch for ScaleONet on a 30-building cluster.

The combination of millisecond-scale inference times, linear training scaling, and lower RMSE relative to the LSTM baseline (see Section 4.2) demonstrates ScaleONet's scalability and efficiency for large

Table 2
Per-sample inference (one-day) times for LSTM and ScaleONet surrogates compared with Dymola simulation.

Cluster size	LSTM (ms/sample)	ScaleONet (ms/sample)	Dymola (s/sample)
Single building (1)	0.827	1.076	5.5
3 buildings	0.912	1.318	18.3
10 buildings	1.102	2.080	74.4
30 buildings	1.416	3.985	590

Table 3
Per-building-per-day RMSE statistics for different input–output resolution scenarios.

Scenario	Mean RMSE (°C)	Median RMSE (°C)	Std RMSE (°C)
$m(50) = n(50)$	0.417	0.357	0.207
$m(100) > n(50)$	0.407	0.349	0.207
$m(50) < n(100)$	0.384	0.319	0.211

building clusters. This linear behavior is consistent with the model architecture: the dominant fusion step scales as $O(Bnp)$ in the number of buildings B, forecast horizon n, and latent dimension p (complete analysis included in Appendix B). As a result, training costs increase proportionally with cluster size, whereas inference remains consistently below 4 ms, confirming the suitability of the surrogates for real-time deployment once trained.

4.8. Sensitivity to input functions' temporal resolution

To examine the robustness of ScaleONet under varying temporal resolutions of input functions, the parameters m and n were defined as follows: m denotes the number of each input function samples (branch-network resolution), and n denotes the number of output prediction instants (trunk-network resolution).

In the experiments, m and n were set to either 50 or 100 points. Three resolution scenarios were investigated: (i) $m = n$ ($m = 50$, $n = 50$), (ii) $m > n$ ($m = 100$, $n = 50$), and (iii) $m < n$ ($m = 50$, $n = 100$).

Table 3 presents per-building-per-day RMSE statistics (mean, median, and standard deviation) for each scenario. The small variation in RMSE metrics across scenarios (mean: 0.417–0.384 °C; median: 0.357–0.319 °C; standard deviation: 0.207–0.211 °C) confirms ScaleONet's ability to handle mismatched input and output resolutions without performance degradation. This robustness stems from the DeepONet architecture's interpolation of differing sample resolutions, facilitating accurate modeling under low-resolution data regimes common in building and climatic datasets.

These results indicate that, although LSTM remains adequate for small, homogeneous clusters, ScaleONet provides a more robust and scalable solution for large, heterogeneous ensembles and variable data resolutions. Such capabilities render ScaleONet a promising surrogate for district-level predictive control and large-scale reinforcement learning applications.

5. Discussion

5.1. Summary of key findings

This study evaluated four research questions (RQs), summarized in Section 3.3. Below, the findings have been grouped by related RQs.

RQ 1: Model performance across cluster sizes

RQ 1 asked how four models — ScaleONet, MLP-ScaleONet, an LSTM baseline, and a physics-based RC model — compare as cluster size grows from one to thirty buildings. As shown in Section 4.2, a standard LSTM baseline slightly outperforms ScaleONet on single-building clusters (where operator learning essentially reduces to direct function approximation) and remains competitive on three-building clusters; however, its prediction error increases monotonically with cluster size. MLP-ScaleONet mirrors ScaleONet at small scales and maintains competitive accuracy as cluster size increases, trailing ScaleONet slightly at the largest ensemble. The RC model serves as a physics baseline and exhibits higher median RMSE than the learned surrogates for larger clusters, reflecting its limited capacity to capture inter-building heterogeneity. In thirty-building ensembles, ScaleONet achieves a 49% reduction in median per-building per-day NRMSE relative to the LSTM, demonstrating its superior scalability and generalization. Moreover, Section 4.6 shows that on a challenging test day, ScaleONet outperforms the LSTM in the thirty-building cluster scenario.

RQ 2: Effect of thermal envelope characteristics

To address RQ 2, the prediction accuracy was analyzed as a function of envelope thermal characteristics (refer to Section 4.5). Buildings with higher total heat-loss coefficients $UA_{building}$ exhibited greater RMSE, particularly under the LSTM baseline. In contrast, ScaleONet maintained consistently lower errors across the full $UA_{building}$ range. Error distributions aligned with building difficulty levels, with easy buildings concentrated at low UA and low RMSE, and difficult buildings predominantly at high UA and high RMSE. These results indicate that ScaleONet generalizes well across a spectrum of thermal envelope behaviors, reducing prediction error for both well-insulated and leaky buildings.

RQ 3: Computational scalability

To address RQ 3, the computational performance of all models was evaluated as cluster size increased (refer to Section 4.7). Both ScaleONet and LSTM exhibit linear scaling in training time with cluster size. ScaleONet incurs the highest per-epoch cost, reaching up to 360 ms for thirty buildings, compared 120 ms for the LSTM baseline, while inference times remain below 4 ms across both surrogates.

This linear behavior is consistent with the model architecture: the dominant fusion step scales as $O(Bnp)$ in the number of buildings B, forecast horizon n, and latent dimension p (complete analysis included as Appendix B). As a result, training costs increase proportionally with cluster size, whereas inference remains consistently below 4 ms, confirming the suitability of the surrogates for real-time deployment once trained.

RQ 3: Computational scalability

To address RQ 3, the computational performance of all models was evaluated as cluster size increased (refer to Section 4.7). Both surrogates exhibit approximately linear training-time scaling with cluster size, while maintaining millisecond-scale inference. Although ScaleONet requires moderately higher per-epoch cost than the LSTM baseline, its inference latency remains similarly low, preserving suitability for real-time deployment.

This linear behavior is consistent with the model architecture: the dominant fusion step scales as $O(Bnp)$ in the number of buildings B, forecast horizon n, and latent dimension p (complete analysis included in Appendix B). Hence, training costs grow proportionally with cluster size, whereas inference complexity remains effectively constant, confirming the models' scalability and efficiency once trained.

RQ 4: Sensitivity to data resolution

To address RQ 4, ScaleONet was evaluated under three resolution scenarios: (i) $m = n = 50$, (ii) $m = 100$, $n = 50$, and (iii) $m = 50$, $n = 100$ (refer to Section 4.8). The per-building, per-day RMSE remained approximately 0.4 °C in all cases, confirming that ScaleONet maintains accuracy despite mismatched input and output temporal resolutions. This robustness underscores its suitability for applications with variable sampling rates of input and output functions.

RQ 5: Interpretability through feature–error attribution

RQ 5 asked whether surrogate models not only achieve accurate forecasts but also provide interpretable insights into the drivers of prediction errors. The Grouped Permutation Importance (GPI) analysis (Section 4.3) revealed that ScaleONet exhibits a physically consistent horizon-dependent pattern: initial indoor states dominate short-term forecasts, while outdoor temperature and solar irradiation increasingly govern long-term predictions.

In contrast, the LSTM baseline showed weaker horizon dependence, the MLP-ScaleONet variant produced noisy and inconsistent patterns, and the RC model was largely insensitive to initial condition perturbations. These results indicate that ScaleONet uniquely combines predictive accuracy with interpretability, reinforcing its value as a surrogate that aligns with physical intuition while generalizing to larger clusters.

5.2. Application to model predictive control

This subsection illustrates how ScaleONet can be embedded within a model predictive control (MPC) framework to facilitate real-time optimization for a cluster of thirty buildings. For a cluster of thirty buildings, ScaleONet functions as a high-speed, accurate surrogate model of the ensemble thermal dynamics. At each MPC iteration, candidate heating schedules for all thirty buildings are passed to ScaleONet, which returns 24-h temperature forecasts in a few milliseconds. These rapid multi-step predictions replace computationally intensive detailed simulations, allowing the MPC optimizer to evaluate many control trajectories over a receding horizon in real time. By embedding ScaleONet's forecasts into the MPC cost — which balances aggregate energy consumption, occupant comfort across all thirty dwellings, and grid flexibility constraints — the controller can efficiently compute optimal heating inputs for the entire cluster. This capability makes scalable, horizon-long MPC feasible for large-scale building ensembles in smart-district applications.

5.3. Limitations and future work

This study employed uniform temporal resolution for all input functions (outdoor air temperature, solar irradiation, HVAC control signals). Future work may investigate the impact of differing temporal resolutions across inputs. The evaluation was limited to Belgian residential buildings; hence, the results are most directly transferable to Western European climates with similar weather profiles. Nevertheless, the proposed ScaleONet framework is not tied to Belgian data: as a data-driven operator-learning method, it can in principle be applied to building clusters in any geography provided that representative training data are available (e.g., from simulations or measurements in China, India, or other regions). Validation on such diverse datasets, including multi-zone and commercial buildings, would strengthen the case for international generalizability. Moreover, the present work is based on synthetic training and testing data generated from high-fidelity Dymola simulations. While this allows for controlled benchmarking of the proposed method, it does not capture real-world uncertainties such as sensor noise, missing data, or unmodeled occupant behavior. Addressing these aspects is beyond the scope of the current study but constitutes an important direction for future work, for instance through noise-augmented training, disturbance forecasting/assimilation, or physics-informed regularization. Subsequent research can extend the framework to larger clusters (from tens to hundreds of buildings) and real-time control applications such as model-based reinforcement learning and MPC. A follow-up study by the authors of this work will assess the practical integration of the ScaleONet surrogate within model-based RL control for district-level energy flexibility control. Finally, to delineate the scope of this study, we clarify that stepwise sequence baselines such as seq2seq, GRU, Transformer, or TCN architectures — which predict one step at a time through temporal recurrence or attention mechanisms — fall outside the operator-learning focus adopted here, but represent a promising direction for future comparative analysis.

6. Conclusion

This work evaluated the use of DeepONet-based operator learning (ScaleONet) for modeling thermal dynamics of heterogeneous clusters of single-zone Belgian buildings. Both ScaleONet and the LSTM baseline were implemented as global models, trained on combined data from all buildings within each cluster rather than as individual per-building models. A detailed, white-box physics-based building thermal model was implemented in Dymola for thirty representative residential buildings in Genk, Belgium. All four surrogates — ScaleONet, MLP-ScaleONet, LSTM, and an RC model — were benchmarked on clusters of size one, three, ten, and thirty buildings. The LSTM baseline and MLP-ScaleONet performed comparably to ScaleONet on single- and three-building clusters, but their prediction error increased monotonically with cluster size (median NRMSEs of 0.703 and 0.789, respectively, at 30 buildings). The RC model exhibited the highest errors across all scales (median NRMSE $\approx$ 1.1). In contrast, ScaleONet improved with scale, achieving a median NRMSE of 0.359 for 30 buildings — representing a 49% reduction relative to the LSTM baseline. ScaleONet's inference time remained fast, and its operator-learning architecture exhibited robustness to varying input and output temporal resolutions. These findings demonstrate that DeepONet-based surrogates not only deliver superior accuracy and scalability but also capture horizon-dependent feature–error relationships, offering an interpretable and computationally viable solution for district-level predictive control and reinforcement learning.

CRediT authorship contribution statement

Muhammad Hafeez Saeed: Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Conceptualization. **Maomao Hu:** Writing – review & editing, Supervision, Resources, Methodology. **Hussain Kazmi:** Writing – review & editing, Supervision, Conceptualization. **Geert Deconinck:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the author(s) used ChatGPT (OpenAI) to improve the readability and clarity of the manuscript. After using this tool, the author(s) reviewed and edited the content as needed and take full responsibility for the content of the published article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Muhammad Hafeez Saeed acknowledges support from the Research Foundation — Flanders (FWO) through a PhD fellowship (grant no. 1S66625N) and through a grant for a research stay at the National University of Singapore, Singapore (grant no. V428225N).

This work has also been supported by KU Leuven, Belgium through the TECHPED-C2 project (C24M/21/021), which investigates technically feasible and effective solutions for Positive Energy Districts. Additional support was provided by the National University of Singapore, Singapore through the Start-Up Grant (A-0009876-00-00) and the

Table A.4
Hyperparameter (and parameter) search spaces and finalized values for ScaleONet, MLP-ScaleONet, LSTM, and RC models. Learned models were tuned with Optuna [48] using the TPE sampler. The RC model parameters were identified via bounded robust least squares (soft-ℓ_1).

ScaleONet (search space)	MLP-ScaleONet (search space)	LSTM baseline (candidate ranges)	RC model (search bounds & solver setup)
Branch LSTM hidden units: {128, 256, 500, 1000} Trunk MLP hidden width (per layer): {100, 500, 1000} Shared latent dimension p: {100, 500, 1000} Learning rate: log-uniform $[10^{-5}, 10^{-3}]$ Validation check interval (epochs): {5, 10, …, 50} Early-stopping patience (checks): {2, 4, …, 20}	Branch MLP hidden width (per layer): {128, 256, 500, 1000} Trunk MLP hidden width (per layer): {100, 500, 1000} Shared latent dimension p: {100, 500, 1000} Learning rate: log-uniform $[10^{-5}, 10^{-3}]$ Validation check interval (epochs): {5, 10, …, 50} Early-stopping patience (checks): {2, 4, …, 20}	LSTM hidden units: {128, 256, 500, 1000} Learning rate: log-uniform $[10^{-5}, 10^{-3}]$ Dropout (if used): {0, 0.1, 0.2} Validation check interval (epochs): {5, 10, …, 50} Early-stopping patience (checks): {2, 4, …, 20}	$[C_i, C_e, R_{ie}, R_{eo}, k_{\text{hvac}}, k_{\text{sol}}]$ $C_i \in [10^4, 10^7]$, $C_e \in [10^5, 5\times10^7]$ $R_{ie} \in [0.01, 2.0]$, $R_{eo} \in [0.05, 5.0]$ $k_{\text{hvac}} \in [0.1, 5.0]$, $k_{\text{sol}} \in [0, 0.5]$ *With solar split* (dim = 7): add $k_{\text{sol,air}}, k_{\text{sol,env}} \in [0, 0.5]$ **Solver:** robust `least_squares`, loss=soft-ℓ_1, $f_scale = 0.7$, bounds as above; max_nfev = 3000; xtol = ftol = gtol = 10^{-9}; multi-start restarts = 5 (jittered x_0); optional burn-in removal.

Notes: RC "search space" corresponds to the physical parameter bounds used in the bounded robust least-squares identification over all training days. Finalized values are building-specific optima. Burn-in steps, if used, exclude early transients from the residual.

Table A.5
Finalized hyperparameters and compact architecture summaries (adopted models).

ScaleONet (adopted)	MLP-ScaleONet (adopted)	LSTM baseline (adopted)	RC model (adopted)
Training: LR = 10^{-4} (Adam), batch = 64, max epochs = 10,000; ES: check/25 epochs, patience 16 checks. **Dims:** Branch LSTM units = 500; shared latent $p = 500$. **Arch (top → bottom):** – Branch input: repeat X_{common} across B; concat with X_{bldg} – Branch LSTM(500) → (batch, B, p) – Trunk: time $(\sin, \cos) \in \mathbb{R}^2$ plus absolute time → Dense(100, ReLU) × 4 → Dense(p) – Fusion: inner product over p (einsum) → $Y \in \mathbb{R}^{\text{batch}\times B\times n}$	**Training:** LR = 7×10^{-5} (Adam), batch = 64, max epochs = 10,000; ES: check/25 epochs, patience 16 checks. **Dims:** Branch hidden width = 500; shared latent $p = 1000$. **Arch (top → bottom):** – Branch input: repeat X_{common} across B; concat with X_{bldg} – Branch Dense(500, ReLU) × 4 → Dense(p) reshape → (batch, B, p) – Trunk: time $(\sin, \cos) \in \mathbb{R}^2$ plus absolute time → Dense(100, ReLU) × 4 → Dense(p) – Fusion: inner product over p (einsum) → $Y \in \mathbb{R}^{\text{batch}\times B\times n}$	**Training:** LR = 10^{-4} (Adam), batch = 64, max epochs = 10,000; ES: check/25 epochs, patience 16 checks. **Dims:** LSTM units = 500. **Arch (top → bottom):** – Flatten inputs → reshape (batch, 1, inp_dim) – LSTM(500) – Dense($B\cdot n$) – Reshape → (batch, B, n)	**Identification (per building):** bounded robust least squares. Loss: soft-ℓ_1, $f_{\text{scale}} = 0.7$; bounds as in Table A.4. Multi-start restarts = 5 (jittered x_0); max_nfev = 3000; xtol = ftol = gtol = 10^{-9}; optional burn-in removal. **Parameters:** $\theta_b = [C_i, C_e, R_{ie}, R_{eo}, k_{\text{hvac}}, k_{\text{sol}}]$ (or $+ k_{\text{sol,air}}, k_{\text{sol,env}}$ if split). *Per-building fitted vectors omitted for brevity.*

Notes: B is the number of buildings; $n = 50$ forecast steps. ES = early stopping.

Ministry of Education, Singapore, under the Academic Research Fund Tier 1 (A-8003235-00-00).

Appendix A. Hyperparameters and architectures

See Tables A.4 and A.5.

Appendix B. Computational scalability in Big-O notation

Let B be the number of buildings, n the forecast horizon length, p the shared latent dimension, E the number of training epochs, N the number of training samples (days), h a representative hidden width (e.g., LSTM units), and L_{trunk} the number of hidden layers in the trunk MLP, respectively. We summarize leading-order costs below (ignoring constant factors and lower-order terms):

ScaleONet (LSTM branch + MLP trunk).

$$\text{Train (per epoch)}: \quad O\big(N\,[\,B\,h^2 \;+\; n\,p\,L_{\text{trunk}}\,h \;+\; B\,n\,p\,]\big),$$
$$\text{Inference (per forward)}: \quad O\big(n\,p\,L_{\text{trunk}}\,h \;+\; B\,n\,p\big).$$

Notes: LSTM branch dominates as $O(Bh^2)$ (per sample), the fusion (einsum) is $O(Bnp)$, and the trunk is $O(npL_{\text{trunk}}h)$. Overall linear in B.

LSTM baseline (single-stream + dense head).

$$\text{Train (per epoch)}: \quad O\big(N\,[\,h^2 \;+\; B\,n\,h \;+\; B\,n\,p\,]\big),$$
$$\text{Inference (per forward)}: \quad O\big(B\,n\,h \;+\; B\,n\,p\big).$$

Notes: With a dense head producing $B \times n$ outputs, both head and fusion-like projection scale linearly with B. The recurrent core is $O(h^2)$ per step; the effective cost grows with the input length but the B-dependence is dominated by the output width.

Summary. Across the learned surrogates, the dominant B-dependence arises from (i) branch outputs sized $B{\times}p$ and (ii) fusion $O(Bnp)$, yielding overall $O(B)$ scaling in both training (per epoch) and inference. This matches the empirically observed linear per-epoch timings with cluster size.

Appendix C. Detailed thermal characteristics of all 30 buildings

See Table C.6.

Data availability

Data will be made available on request.

Table C.6
Detailed thermal characteristics of 30 buildings in Genk, Belgium that have been modeled in Dymola. Columns represent: floor area (Area); thermal transmittances of walls, roof, and windows (U-values) that describe envelope insulation quality; overall heat loss coefficient (UA) aggregating all conductive losses; envelope shape factor (A_{env}/V), i.e. ratio of external surface area to enclosed volume that reflects compactness; and window-to-wall ratio (WWR), which characterizes façade transparency and thus influences both heat losses and solar gains.

Name	Area [m^2]	U_{walls} [W/m^2 K]	U_{roof} [W/m^2 K]	U_{win} [W/m^2 K]	UA [W/K]	A_{env}/V [m^2/m^3]	WWR [–]
Anijsstraat_4_1591809	171.52	1.9420	0.4190	2.7460	688.35	0.7887	0.3316
Azalealaan_10_1594731	188.58	2.5244	3.1072	3.5380	1372.49	0.8242	0.1919
Azalealaan_11_1592110	94.60	2.5244	3.1072	3.5423	963.07	0.8712	0.1561
Azalealaan_12_1594732	170.76	2.2697	3.1072	2.6780	1156.32	1.0197	0.1876
Azalealaan_14_4752585	239.08	2.2697	0.7770	3.0200	931.16	0.8180	0.3661
Azalealaan_16_1594730	193.24	2.5244	3.1072	4.9371	1830.56	0.8394	0.2195
Azalealaan_19_4754000	104.56	1.9980	1.3110	2.3444	761.54	0.9418	0.1186
Azalealaan_1_1594381	59.18	0.5970	0.7900	3.5495	230.05	0.7298	0.1325
Azalealaan_20_1594198	177.59	1.3530	0.2110	2.3170	527.69	0.8756	0.2077
Azalealaan_2325_1594028	58.53	2.3120	0.4310	3.2320	310.82	0.6092	0.3870
Azalealaan_25_1594769	57.61	2.5244	3.1072	2.9030	550.56	0.7277	0.0893
Azalealaan_2729_1594027	56.29	1.6114	0.4560	4.9544	260.64	0.6097	0.1669
Azalealaan_2729_4754525	58.15	2.2697	3.1072	4.9526	553.29	0.7235	0.1312
Azalealaan_31_5213796	64.26	2.5244	0.5170	2.6299	589.21	0.8652	0.1377
Azalealaan_32_1594612	61.53	1.6530	1.6110	2.8780	397.03	0.7262	0.1161
Azalealaan_33_5213798	60.03	2.0580	3.1072	4.9384	594.79	0.7080	0.2987
Azalealaan_34_5213799	60.92	2.5244	3.1072	4.9398	667.52	0.7418	0.2723
Azalealaan_35_5213728	57.87	2.3400	2.0550	2.9530	386.63	0.6164	0.2034
Azalealaan_36_1592112	84.89	0.8590	3.1072	3.1900	562.18	0.8772	0.0627
Azalealaan_37_1591726	67.96	2.5244	3.1072	3.1030	687.57	0.7529	0.0977
Azalealaan_3_1592109	60.34	1.2920	0.4440	2.1570	299.36	0.7328	0.1050
Azalealaan_41_1594592	57.35	2.1400	0.3460	1.9400	239.62	0.6049	0.1891
Azalealaan_41_1594774	58.02	1.2660	3.1072	3.4030	417.97	0.7186	0.1111
Azalealaan_43_1594591	58.90	1.7980	0.5230	3.5576	260.45	0.6098	0.1429
Azalealaan_43_1594775	57.76	2.5244	3.1072	3.1700	567.66	0.7245	0.1751
Azalealaan_4648_1594024	45.28	2.4450	3.1072	2.8011	448.12	0.6097	0.2048
Azalealaan_48_1594748	90.44	2.3320	0.3860	3.5369	516.30	0.7383	0.2821
Azalealaan_5052_1594025	56.52	2.0150	2.1800	3.3761	416.03	0.6033	0.1442
Azalealaan_50_1594749	56.82	2.5244	3.1072	3.1530	642.15	0.7213	0.2937
Azalealaan_5456_1594022	68.58	2.4300	0.4530	2.7180	552.06	0.8831	0.0811

References

[1] Paris agreement. 2016, p. 4–18, URL http://data.europa.eu/eli/reg/2018/842/oj.

[2] International Energy Agency (IEA). Net zero roadmap: A global pathway to keep the 1.5 °C goal in reach. 2023, URL https://www.iea.org/reports/net-zero-roadmap-a-global-pathway-to-keep-the-15-0c-goal-in-reach. Licence: CC BY 4.0.

[3] European Commission, Directorate-General for Climate Action. Going climate-neutral by 2050: A strategic long-term vision for a prosperous, modern, competitive and climate-neutral EU economy. 2019, URL https://data.europa.eu/doi/10.2834/02074.

[4] Liang Y, Pan Y, Yuan X, Yang Y, Fu L, Li J, Sun T, Huang Z, Kosonen R. Assessment of operational carbon emission reduction of energy conservation measures for commercial buildings: Model development. Energy Build 2022;268:112189. http://dx.doi.org/10.1016/j.enbuild.2022.112189.

[5] International Energy Agency (IEA). Tracking clean energy progress 2023. 2023, URL https://www.iea.org/reports/tracking-clean-energy-progress-2023. Licence: CC BY 4.0.

[6] Aste N, Del Pero C, Leonforte F, Ounis S, Abdelrahim A. The role of photovoltaic technology in achieving net zero energy target in residential buildings. Energy Strat Rev 2024;55:101543. http://dx.doi.org/10.1016/j.esr.2024.101543.

[7] Langevin J, Harris CB, Satre-Meloy A, Chandra-Putra H, Speake A, Present E, Adhikari R, Wilson EJH, Satchwell AJ. US building energy efficiency and flexibility as an electric grid resource. Joule 2021;5(8):2102–28. http://dx.doi.org/10.1016/j.joule.2021.06.002.

[8] Calero I, Canizares CA, Bhattacharya K, Baldick R. Duck-curve mitigation in power grids with high penetration of PV generation. IEEE Trans Smart Grid 2021;13(1):314–29. http://dx.doi.org/10.1109/TSG.2021.3122398.

[9] Knittel T, Palmer-Wilson K, McPherson M, Wild P, Rowe A. Heating electrification in cold climates: Invest in grid flexibility. Appl Energy 2024;356:122333. http://dx.doi.org/10.1016/j.apenergy.2023.122333.

[10] Söder L, Lund PD, Koduvere H, Bolkesjø TF, Rossebø GH, Rosenlund-Soysal E, Skytte K, Katz J, Blumberga D. A review of demand side flexibility potential in Northern Europe. Renew Sustain Energy Rev 2018;91:654–64. http://dx.doi.org/10.1016/j.rser.2018.03.104.

[11] Zou B, Peng J, Li S, Li Y, Yan J, Yang H. Comparative study of the dynamic programming-based and rule-based operation strategies for grid-connected PV-battery systems of office buildings. Appl Energy 2022;305:117875. http://dx.doi.org/10.1016/j.apenergy.2021.117875.

[12] Dolatabadi A, Abdeltawab H, Mohamed YA-RI. A novel model-free deep reinforcement learning framework for energy management of a PV integrated energy hub. IEEE Trans Power Syst 2023;38(5):4840–52. http://dx.doi.org/10.1109/TPWRS.2022.3212938.

[13] Salpakari J, Lund P. Optimal and rule-based control strategies for energy flexibility in buildings with PV. Appl Energy 2016;161:425–36. http://dx.doi.org/10.1016/j.apenergy.2015.10.036.

[14] Nweye K, Kaspar K, Buscemi G, Fonseca T, Pinto G, Ghose D, Duddukuru S, Pratapa P, Li H, Mohammadi J, Ferreira LL, Hong T, Ouf M, Capozzoli A, Nagy Z. CityLearn v2: energy-flexible, resilient, occupant-centric, and carbon-aware management of grid-interactive communities. J Build Perform Simul 2025;18(1):17–38. http://dx.doi.org/10.1080/19401493.2024.2418813.

[15] Drgoňa J, Arroyo J, Cupeiro Figueroa I, Blum D, Arendt K, Kim D, Ollé EP, Oravec J, Wetter M, Vrabie DL, Helsen L. All you need to know about model predictive control for buildings. Annu Rev Control 2020;50:190–232. http://dx.doi.org/10.1016/j.arcontrol.2020.09.001.

[16] Gao C, Wang D. Comparative study of model-based and model-free reinforcement learning control performance in HVAC systems. J Build Eng 2023;74:106852. http://dx.doi.org/10.1016/j.jobe.2023.106852.

[17] Yao Y, Shekhar DK. State of the art review on model predictive control (MPC) in heating ventilation and air-conditioning (HVAC) field. Build Environ 2021;200:107952. http://dx.doi.org/10.1016/j.buildenv.2021.107952.

[18] Ding X, Cerpa A, Du W. Multi-zone HVAC control with model-based deep reinforcement learning. IEEE Trans Autom Sci Eng 2025;22:4408–26. http://dx.doi.org/10.1109/TASE.2024.3410951.

[19] Gupta J, Chakraborty M. 15 - energy efficiency in buildings. In: Dutta S, Mustansar Hussain C, editors. Sustainable fuel technologies handbook. Academic Press; 2021, p. 457–80. http://dx.doi.org/10.1016/B978-0-12-822989-7.00016-0.

[20] Stoffel P, Berktold M, Gall A, Kiimpel A, Muller D. Comparative study of neural network based and white box model predictive control for a room temperature control application. J Phys: Conf Ser 2021;2042(1):012043. http://dx.doi.org/10.1088/1742-6596/2042/1/012043.

[21] Lu S, Gu W, Ding S, Yao S, Lu H, Yuan X. Data-driven aggregate thermal dynamic model for buildings: A regression approach. IEEE Trans Smart Grid 2022;13(1):227–42. http://dx.doi.org/10.1109/TSG.2021.3101357.

[22] Li S, Li S, Mohebi P, Wang D, Ma MN, Liu G, Wang Z. Field demonstration of model predictive control for chiller sequencing in large-scale commercial buildings. Energy Build 2025;344:116021. http://dx.doi.org/10.1016/j.enbuild.2025.116021.

[23] Amadeh A, Lee ZE, Max Zhang K. Building cluster demand flexibility: An innovative characterization framework and applications at the planning and operational levels. Energy Convers Manage 2023;283:116884. http://dx.doi.org/10.1016/j.enconman.2023.116884.

[24] Saeed MH, Kazmi H, Deconinck G. Dyna-PINN: Physics-informed deep dyna-q reinforcement learning for intelligent control of building heating system in low-diversity training data regimes. Energy Build 2024;324:114879. http://dx.doi.org/10.1016/j.enbuild.2024.114879.

[25] Lu L, Jin P, Pang G, Zhang Z, Karniadakis GE. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nat Mach Intell 2021;3(3). http://dx.doi.org/10.1038/s42256-021-00302-5.

[26] Lin C, Li Z, Lu L, Cai S, Maxey M, Karniadakis GE. Operator learning for predicting multiscale bubble growth dynamics. J Chem Phys 2021;154(10). http://dx.doi.org/10.1063/5.0041203.

[27] Wang S, Perdikaris P. Long-time integration of parametric evolution equations with physics-informed DeepONets. J Comput Phys 2023;475:111855. http://dx.doi.org/10.1016/j.jcp.2022.111855.

[28] Koric S, Viswantah A, Abueidda DW, Sobh NA, Khan K. Deep learning operator network for plastic deformation with variable loads and material properties. Eng Comput 2024;40(2):917–29. http://dx.doi.org/10.1007/s00366-023-01822-x.

[29] Zheng Q, Yin X, Zhang D. State-space modeling for electrochemical performance of Li-ion batteries with physics-informed deep operator networks. J Energy Storage 2023;73:109244. http://dx.doi.org/10.1016/j.est.2023.109244.

[30] Koric S, Abueidda DW. Data-driven and physics-informed deep learning operators for solution of heat conduction equation with parametric heat source. Int J Heat Mass Transfer 2023;203:123809. http://dx.doi.org/10.1016/j.ijheatmasstransfer.2022.123809.

[31] Lu M, Lin C, Maxey M, Karniadakis GE, Li Z. Bridging scales in multiscale bubble growth dynamics with correlated fluctuations using neural operator learning. Int J Multiph Flow 2024;180:104959. http://dx.doi.org/10.1016/j.ijmultiphaseflow.2024.104959.

[32] He J, Koric S, Abueidda D, Najafi A, Jasiuk I. Geom-DeepONet: A point-cloud-based deep operator network for field predictions on 3D parameterized geometries. Comput Methods Appl Mech Engrg 2024;429:117130. http://dx.doi.org/10.1016/j.cma.2024.117130.

[33] Nghiem TX, Nguyen T, Nguyen BT, Nguyen L. Causal deep operator networks for data-driven modeling of dynamical systems. In: 2023 IEEE international conference on systems, man, and cybernetics. SMC, 2023, p. 1136–41. http://dx.doi.org/10.1109/SMC53992.2023.10394294.

[34] Ahmed B, Qiu Y, Abueidda DW, El-Sekelly W, de Soto BG, Abdoun T, Mobasher ME. Physics-informed deep operator networks with stiffness-based loss functions for structural response prediction. Eng Appl Artif Intell 2025;144:110097. http://dx.doi.org/10.1016/j.engappai.2025.110097.

[35] Kim Y. Optimal control strategy for a large-scale refrigeration system in a plant building using DeepONet [Ph.D. thesis], Seoul National University Graduate School; 2025.

[36] Cahyadi, Forshaw M. Hard disk failure prediction on highly imbalanced data using LSTM network. In: 2021 IEEE international conference on big data. big data, 2021, p. 3985–91. http://dx.doi.org/10.1109/BigData52589.2021.9671555.

[37] Saeed MH, Kazmi H, Deconinck G. Physics informed deep-o-nets as universal approximator for modelling district-level building heating systems. In: Proceedings of the 11th ACM international conference on systems for energy-efficient buildings, cities, and transportation. BuildSys '24, Association for Computing Machinery; 2024, p. 246–7. http://dx.doi.org/10.1145/3671127.3698712.

[38] Csáji BC. Approximation with artificial neural networks, master's thesis. Fac Sci Etvs Lornd Univ Hung 2001.

[39] Islam M, Chen G, Jin S. An overview of neural network. Am J Neural Netw. Appl 2019;5(1):7–11. http://dx.doi.org/10.11648/j.ajnna.20190501.12.

[40] Schmidhuber J. Deep learning in neural networks: An overview. Neural Netw 2015;61:85–117. http://dx.doi.org/10.1016/j.neunet.2014.09.003.

[41] Wang S, Wang H, Perdikaris P. Improved architectures and training algorithms for deep operator networks. J Sci Comput 2022;92(2):35. http://dx.doi.org/10.1007/s10915-022-01881-0.

[42] De Jaeger I. On the impact of input data uncertainty on the reliability of urban building energy models [Ph.D. thesis], KU Leuven; 2021.

[43] Vandermeulen A, De Jaeger I, Van Oevelen T, Saelens D, Helsen L. Analysis of building parameter uncertainty in district heating for optimal control of network flexibility. Energies 2020;13(23). http://dx.doi.org/10.3390/en13236220.

[44] Jorissen F, Reynders G, Baetens R, Picard D, Saelens D, Helsen L. Implementation and verification of the IDEAS building energy simulation library. J Build Perform Simul 2018;11(6):669–88. http://dx.doi.org/10.1080/19401493.2018.1428361.

[45] Mattsson SE, Elmqvist H, Otter M. Physical system modeling with modelica. Control Eng Pract 1998;6(4):501–10. http://dx.doi.org/10.1016/S0967-0661(98)00047-1.

[46] Dempsey M. Dymola for multi-engineering modelling and simulation. In: 2006 IEEE vehicle power and propulsion conference. 2006, p. 1–6. http://dx.doi.org/10.1109/VPPC.2006.364294.

[47] TensorFlow Development Team. TensorFlow `tf.function` guide. 2025, https://www.tensorflow.org/guide/function. [Accessed 06 August 2025].

[48] Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. KDD '19, Association for Computing Machinery; 2019, p. 2623–31. http://dx.doi.org/10.1145/3292500.3330701.