



A highly configurable framework for large-scale thermal building data generation to drive machine learning research

Thomas Krug^{1,2} , Fabian Raisch^{1,3}, Dominik Aimer¹,
Markus Wirnsberger¹, Ferdinand Sigg^{3,4}, Felix Koch^{1,2},
Benjamin Schäfer² and Benjamin Tischler¹

Abstract

Data-driven modeling of building thermal dynamics is emerging as an increasingly important field of research for large-scale intelligent building control. However, research in data-driven modeling using machine learning (ML) techniques requires massive amounts of thermal building data, which is not easily available. Neither empirical public data sets nor existing data generators meet the needs of ML research in terms of data quality and quantity. Moreover, existing data generation approaches typically require expert knowledge in building simulation. To fill this gap, we present a thermal building data generation framework, which we call *BuilDa*. *BuilDa* is designed to produce synthetic data of adequate quality and quantity for ML research. The framework does not require profound building simulation knowledge to generate large volumes of data. *BuilDa* uses a validated single-zone Modelica model representing the entire building, exported as a functional mock-up unit (FMU) and simulated in Python. We demonstrate *BuilDa* by generating data and utilizing it for a transfer learning study involving the fine-tuning of 486 data-driven models.

Keywords

synthetic data generation, transfer learning, building thermal dynamics, data-driven modeling

1. Introduction

In 2021, building operations accounted for 27% of global CO₂ emissions and 30% of the world's energy demand.¹ Intelligent building energy control, such as model predictive control (MPC)^{2,3} or reinforcement learning (RL),^{4,5} can significantly reduce building energy consumption. Usually, these methods rely on a representative model of the building's thermal dynamics. Similarly, such models can act as baseline models for fault detection and diagnosis (FDD) research,⁶ improving energy efficiency. However, building thermal dynamic models often require labor-intensive manual modeling and expert knowledge. Data-driven methods offer a way to overcome this problem, as the models can be learned directly from data.^{2,6,7} Machine learning (ML) techniques are particularly suitable for this purpose, as they allow modeling complex, non-linear building behavior without explicit physical modeling, thereby supporting the widespread adoption of intelligent control strategies. However, depending on the task,

deriving accurate data-driven models often requires long-term data collection over several months to years, which is infeasible for most buildings, especially newly constructed ones.

Transfer learning (TL) as an ML technique addresses the challenge of data scarcity in data-driven models.^{8,9} In TL, a model is pretrained with data from a source building to acquire knowledge. Subsequently, the pretrained model is fine-tuned for a target building using a relatively small amount of data measured in the target.^{8,10,11} Promising developing research areas in the TL field are research in

¹Rosenheim Technical University of Applied Sciences, Germany

²Karlsruhe Institute of Technology, Germany

³Technical University of Munich, Germany

⁴Munich University of Applied Sciences HM, Germany

Corresponding author:

Thomas Krug, Rosenheim Technical University of Applied Sciences,
Hochschulstraße 1, 83024 Rosenheim, Germany.
Email: thomas.krug@th-rosenheim.de

generalized TL models,¹² studies on building similarity (e.g. weather conditions, building size, occupancy patterns)^{13–16} or studies on the adaptiveness of data-driven models to changes in the built environment.¹⁷ Consequently, advancing research in data-driven building thermal dynamic models and TL is essential to achieve a widespread adoption of intelligent building applications and improve their robustness and generalization capabilities. This underlines the need for massive amounts of high-variance data with detailed and accessible metadata, which can be tailored to the research questions.

For this purpose, data sets for ML research should ideally exhibit high variability to cover a wide range of building dynamics, representing numerous buildings, and their environmental and operational conditions. Research data should cover all seasons over multiple years to allow for representative training and test sets, while also providing detailed metadata descriptions. Yet, data with desired properties are rare (see Section 2 “Background”), restricting ML research for building thermal dynamics significantly.

Currently, researchers either need to rely on publicly available data sets or generate synthetic data themselves. Both options have significant limitations: existing data sets lack flexibility and metadata, while synthetic data generation typically requires extensive building simulation expertise. This creates a significant barrier for ML researchers in the building domain, who are thus often constrained in the systematic development and evaluation of data-driven methods for thermal building modeling and control.

To address this gap, we present *Builda*, a highly configurable data generation framework that enables researchers to systematically synthesize large-scale, high-variance building thermal data sets for ML research without requiring simulation expertise. *Builda* provides a validated physical building model and supports parallelized data generation. Users can freely configure the physical and operational parameters governing building thermal dynamics, which influence energy consumption, control performance, and indoor comfort. These include envelope properties, weather conditions, occupancy and internal gains, control strategies, and retrofitting scenarios. In addition, we provide predefined profiles for weather, wall constructions, internal heat gains, window-opening behavior, and control strategies. This work extends the foundations established in previous research.¹⁸

Our contributions are as follows: (1) We provide an extensive framework for batch data generation of synthetic multivariate thermal building time-series data, with a focus on parameter variations. It is highly customizable, physically consistent, usable without building simulation expertise, and suitable for diverse ML research applications. (2) The framework includes a configurable and validated single-zone building simulation model, incorporating extensive domain knowledge and covering a wide range of

possible single-zone building configurations. (3) Moreover, the framework supports the simulation of operational changes and retrofitting scenarios, enabling modifications of building envelope properties during simulation. In addition, we provide predefined load profiles covering occupancy schedules, internal heat gains from occupants and electrical devices, and correlated window-opening behavior, representing five different household types. (4) Finally, we showcase the usage of *Builda* by generating synthetic data for an extended TL study on the influence of building properties on TL performance. In this study, 486 standard ML models are pretrained, fine-tuned, and compared.

Typical application scenarios include TL research, where carefully curated pairs of source and target buildings are required; controller development, where diverse and reproducible training environments are needed; or anomaly and fault detection studies, where parameter changes, internal gains, or control policy variations allow manipulation of system behavior at defined time points. In addition, *Builda* enables systematic analysis of building similarity and the generalization behavior of data-driven models across varying occupancy and internal gains, envelope properties, and retrofit scenarios.

The remainder of this paper is organized as follows. Section 2 “Background” provides an overview of data generation methods and related frameworks. Section 3 “Framework and Functionalities” introduces the *Builda* framework and its features. Section 4 “Demonstration” presents demonstration use cases. Finally, Section 5 “Conclusion and Future Work” concludes the paper.

2. Background

ML research in the building domain often relies on publicly available data sets. These data sets, whether real-world measurements or synthetically generated,^{19–23} typically provide static information that cannot be adapted to different conditions. Since the data is already measured or generated, it represents only a fixed set of buildings and operating scenarios. This lack of flexibility limits detailed analyses of thermal dynamics and constitutes a major drawback of existing data sets. Moreover, public data sources often lack detailed metadata, especially for occupancy, building envelope characteristics, and weather data. Zhang et al.¹⁵ examine how building similarity affects TL performance for cooling load prediction, but could not consider features of the building envelope in detail, due to inaccessible metadata. Li et al.¹⁴ use the Ecobee data set²³ but had to manually infer weather conditions from the nearest weather station, highlighting data aggregation challenges. Similarly, the IDEAL²² data set faces the same limitation, requiring weather data to be extrapolated from nearby weather stations.

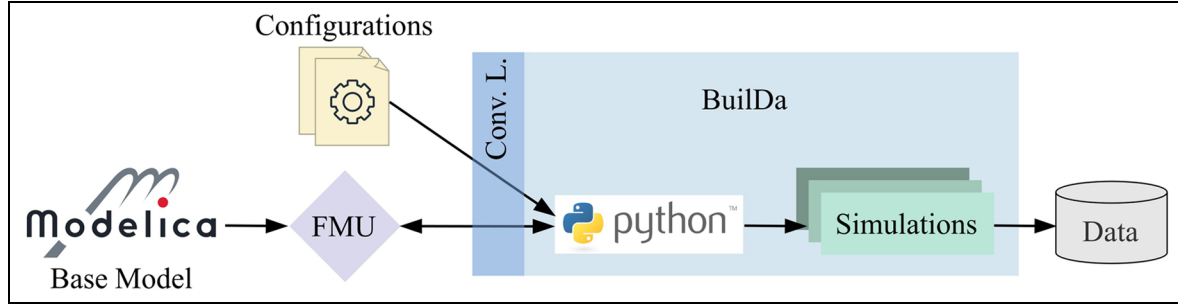


Figure 1. Architecture overview Builda.
Source: Adapted from Krug et al.¹⁸

An alternative approach is the generation of synthetic data using physics-based building models, implemented in simulation environments such as Modelica²⁴ or EnergyPlus.²⁵ While these models theoretically enable the generation of unlimited data, they come with several practical limitations. First, building and configuring such models requires substantial domain expertise and is highly time-intensive. Second, models are typically tailored to a specific building and cannot be easily reused for other building configurations, limiting their adaptability and usefulness for ML research. Tools such as eppy²⁶ partially address this by enabling systematic modifications of EnergyPlus models. However, users must still ensure physical consistency of any modifications made, which continues to demand considerable simulation expertise.

Another approach involves data generators, which integrate building modeling knowledge, making them more accessible to non-experts. Existing solutions range from data augmentation frameworks²⁷ to data generators for synthetic smart meter or electrical data.^{28,29} However, only a few focus on thermal building dynamics, such as Synconn_build,³⁰ SBsim,³¹ or eplusr.³² SBsim is limited as it relies on a finite-difference approximation of building thermodynamics rather than a full physical simulation model and is primarily intended for RL research. Synconn_build uses a physical simulation model, but lacks flexibility in terms of adjustable building parameters, limiting its suitability for ML research. Eplusr offers extensive configurability, but still requires expertise in manipulating EnergyPlus Input Data File (IDF) files.

Other well-known frameworks, such as Sinergym,³³ Energym,³⁴ and the Building Optimization Testing Framework (BOPTEST),³⁵ are primarily designed as building simulation environments for evaluating and benchmarking control strategies such as RL and are not intended for systematic batch data generation. Although time-series data can be extracted from these environments, this typically requires modifications to the framework for parameter variations, orchestration, and data collection.

Furthermore, the set of available building models in these frameworks is generally fixed, with limited support for systematic building parameter variations.

These challenges highlight the need for accessible, customizable, physically consistent, and scalable data generation frameworks for ML research in building thermal dynamics. Builda enables systematic, large-scale data set generation with extensive parameter variation, accessible to researchers without prior building simulation expertise.

3. Framework and Functionalities

3.1. Architecture overview

This chapter provides an overview of the architecture of the Builda framework. As illustrated in Figure 1, the framework consists of two primary components. The first component is a physical Modelica building model (referred to as the base model) that simulates the thermal dynamics of a single-zone building. The second component is a Python-based framework to run multiple simulations with varying parameters and operational schedules. It executes the base model as a functional mock-up unit (FMU) in co-simulation via the FMPy library,³⁶ following the Functional Mock-up Interface (FMI) standard 2.0.4.³⁷ Within the FMU, the CVODE solver³⁸ is used to integrate the system, with a relative error tolerance of 1×10^{-6} and variable step size.

Configuration files enable users to set parameters for the building model, control strategy, simulation, and operational schedules. A key feature of Builda is the converter layer, which simplifies user input by automatically managing parameter dependencies. The converter layer is described in Section 3.4 “Converter Layer.” It is specifically tailored to our FMU. Section 3.2 “Building Model” provides an overview of the base model and the adjustable parameters within Builda, and Section 3.3 “Operational Schedules” describes how parameters and gain profiles can be dynamically changed during operation.

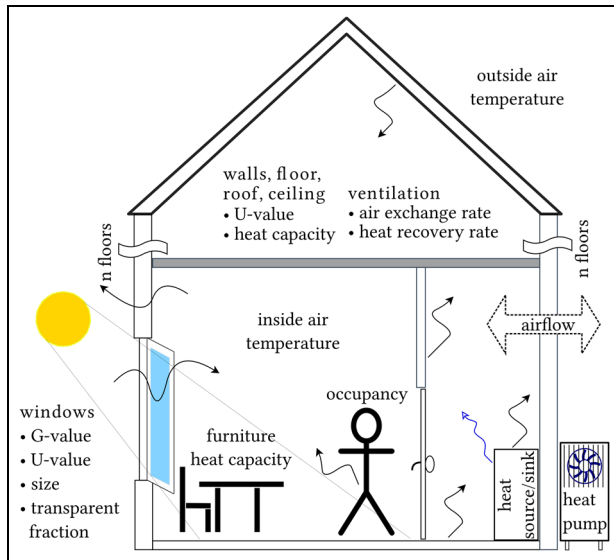


Figure 2. Basic setup of the heating system.
Source: Adapted from Krug et al.¹⁸

3.2. Building model

To ensure model accuracy, the building and its components are modeled in accordance with the methods outlined in VDI 6007 Part 1 for building simulations.³⁹ The entire building is represented as a single thermal zone, encompassing all rooms. This implies that spatial temperature variations within the building are neglected and the indoor air is assumed to be fully mixed. Consequently, a single control unit is used for heating and cooling, and all internal gains, including occupants, are aggregated and assumed to be uniformly distributed within the zone. Figure 2 illustrates the single-zone building model, highlighting its main components and parameters. Heat flows are indicated with arrows (blue for cooling). The model comprises walls, roof, floor, ceiling, windows, and furniture. Heating and cooling are represented by an ideal thermal source with negligible thermal mass, analogous to an electric radiator, which transfers heat directly to the environment. The convective and radiative fractions are defined in the configuration files. Maximum heating power is determined by the converter layer according to DIN 18599 Part 2⁴⁰ (see Section 3.4 “Converter Layer”). The same standard is applied for maximum cooling power, using a simplified approach consistent with the overall model assumptions.

In addition, a simple heat pump model is included to calculate the electrical power required to supply the current heat flow to the zone. The coefficient of performance (COP) is determined from the characteristic curve of the OVUM ACP520 reference heat pump⁴¹ and depends on the outside temperature, supply temperature, and compressor speed.

Solar gains and thermal losses are considered for windows, exterior walls, and the roof, while only thermal losses are accounted for in the floor. Within the zone, heat exchange between walls and between walls and the thermal source is modeled via long-wave radiation and convection. We further assume a uniform indoor air temperature, which is a valid simplification for many building types, particularly residential buildings. This reduces model complexity while still reflecting typical conditions.⁴²

The core model is derived from an example in the Modelica Buildings Library.⁴³ While the core components of the base model, including external weather data input, envelope, and thermal zone, were retained, the model was extended as follows:

- modeling ventilation heat losses due to infiltration, mechanical ventilation, and controllable window opening
- enabling external input data for internal heat gains from occupants and appliances
- integrating active heating and cooling capabilities
- implementing an internal proportional heating controller with configurable setpoints
- providing interfaces for external heating, cooling, and window controllers
- tracking cumulative heating and cooling energy consumption during simulation
- calculating the electrical power consumption of an optional heat pump model

Figure 10 in Appendix 1 provides an overview of the Modelica implementation of our model.

A summary of modifiable parameters is provided in Table 2. Further details are given in the following subsections, where the discussed parameters are highlighted in bold.

3.2.1. Building parameters. These parameters describe the key physical characteristics of a building, including its size, envelope properties, thermal mass, windows, and ventilation, all of which influence its thermal behavior.⁴⁴ They are thus critical drivers of energy consumption, indoor temperature dynamics, and occupant comfort, which motivates their inclusion in the configurable model. The *building size* is defined by floor length, width, and height, and the number of floors. Building components, such as roof, floor, windows, and interior and exterior walls, are modeled as R–C elements. The R-value describes thermal resistance and thus *envelope quality*, while the C-value represents *thermal mass*. Each component is discretized into three R–C nodes, which provide sufficient accuracy for the model.⁴⁵ The achievable accuracy depends on the number of nodes and their distribution within the wall structure. While users

Table 1. Predefined wall profiles.

Name	U-value W/(m ² K)	Heat cap. kJ/(m ² K)	Description
High-hole brick (1980s)	0.83	250	Old-fashioned wall with poor insulation properties, but high inertia
Solid brick	1.61	376	Old-fashioned wall with very poor insulation properties and very high inertia
Concrete + ETICS	0.21	470	Wall with good insulation properties and high inertia
Timber construction	0.15	75	Wall with very good insulation properties and low inertia
High-hole brick (today)	0.23	265	Wall with good insulation properties and high inertia
Drywall	0.56	17.6	Internal wall with insulation and low heat capacity

can manually configure components by specifying the U-value, heat capacity, and the distribution of resistance and capacity within the component, *BuilDa* also provides a convenient option to select from predefined component profiles, representing typical configurations. The available profiles are summarized in Table 1.

The heat capacity of the furniture is considered as part of the internal walls for simplicity, as it is supposed to have a similar impact on the thermal dynamics of the building. For external walls and *windows*, orientation is taken into account to capture direction-dependent solar gains. The window G-value represents the fraction of solar radiation transmitted through the glazing.

For *ventilation*, heat flow (due to ventilation system, infiltration, or window opening) is determined by considering the air flows and the heat recovery rate (0 for natural ventilation or exhaust-air systems, > 0 for ventilation systems with heat recovery). *Window opening* is included in the model. Following the standard VDI 2078,⁴⁶ we have modeled the airflow through a fully opened window based on the indoor and outdoor temperatures, as well as the size and dimensions of the window. During the simulation, the window can be opened according to an external window-opening profile (see Section 3.2.2 “Input Parameters”) or by an external control signal (see Table 2), allowing the user to directly implement customized control logic.

The *heat pump* in the building model can be configured by adjusting the efficiency in relation to the efficiency of the used reference heat pump⁴¹ (see also Section 3.2 “Building Model”). Finally, the steepness of the heat curve, which maps the outside temperature to the required heating supply temperature, can be configured.

3.2.2. Input parameters. In contrast to building parameters, input parameters define the dynamic inputs to the model during simulation, including control strategies, occupancy patterns, window-opening behavior, and weather conditions. *BuilDa* supports two types of *controllers*. The FMU contains an internal heating controller implemented as a proportional controller with night setback, where day and night setpoints can be configured individually. Alternatively, external controllers can be defined for

heating, cooling, and window operation, enabling more flexible and adaptive control strategies. To achieve this level of flexibility, we decoupled the frequency at which the controller updates the actuation signal from the rate at which the simulation retrieves its status and writes data.

Occupancy in *BuilDa* affects *internal heat gains* and, optionally, *window-opening* activity, both represented through annual profiles. A provided Python script can generate these yearly schedules for internal gains (from occupants and electrical devices) and window-opening behavior. The script uses 4-day profiles (workday, Saturday, Sunday, and holiday) with individual activity schedules and optional internal heat gains from electrical devices. Window opening is updated every 5 min based on occupancy, sleep times, and ventilation awareness.

In addition, *BuilDa* includes five predefined profile sets, each consisting of an internal gain profile (including heat gains from occupants and electrical devices) and a corresponding window-opening profile, representing typical German residential household archetypes. Realistic internal heat gains were generated using the *LoadProfileGenerator (LPG)*,⁴⁷ which provides detailed stochastic household demand profiles derived from German time-use statistics and appliance models. It simulates person-specific activities and device usage to produce temporally resolved electricity consumption and occupancy data. The selected household archetypes cover different compositions, age groups, and occupancy patterns, ensuring a realistic variety of internal heat gain behaviors throughout the day (see Table 3).

These profiles offer a diverse representation of occupancy and appliance usage patterns. They serve as stochastic input for the thermal model’s internal heat gain calculation, linking electrical power consumption to the corresponding sensible and latent heat releases. Furthermore, heat emissions from occupants are included to account for the metabolic contribution of human presence.

Weather significantly affects a building’s thermal behavior. In *BuilDa*, different climatic conditions can be simulated by specifying the weather file path in the configuration file. The model imports weather data in the Modelica weather data file format (MOS). Weather files are provided in EnergyPlus Weather format (EPW), which

Table 2. Building simulation parameters.

Category	Parameter	Explanation
Building	Building size	Thermal envelope size (including length, width, number of floors, and floor height)
	Envelope quality	Thermal quality of building envelope (U-value, structure)
	Thermal mass	Heat capacity of external and internal walls, roof, floor, and furniture
	Windows	Size, orientation, and quality of windows (U-value, G-value, and transparent fraction)
	Ventilation	Air exchange rates and heat recovery rate
Input	Heat pump	Heat pump efficiency and steepness of the heat curve
	Controller	Configurable external controller(s) (e.g., for heating, cooling, window opening) or internal controller configuration (setpoints)
	Internal gains	Internal gains profile, calculated from occupancy profile (optionally contains heat gains from electrical devices)
	Window opening	Window-opening profile (influences ventilation heat losses), calculated from occupancy profile
	Weather	Weather data with outside temperature, solar radiation, etc. (main influence on heat energy demand and zone temperature)

Source: Adapted from Krug et al.¹⁸

Table 3. Predefined gain profiles.

Name	Description
CHR07_Single_with_work	Single adult, regular daytime work outside the home. Typical weekday absence during daytime; morning and evening peaks in appliance use. Represents urban single apartments.
CHR01_Couple_both_at_Work	Two adults, both employed. Moderate consumption during weekdays with evening peaks (cooking, laundry, lighting). Represents common dual-income households.
CHR27_Family_both_at_work_2_children	Four-person household (two adults and two school-aged children). Pronounced morning and evening peaks; high weekend variability. Represents family homes in suburban areas.
CHR16_Couple_over_65_years	Elderly couple, retired. Occupied most of the day; more constant internal heat gains with smaller peak loads. Represents older households or senior residences.
CHR52_Student_Flatsharing	Shared flat (three to four young adults) with irregular schedules and variable appliance use. Captures non-synchronous, high evening loads typical for student housing.

is widely used in building performance simulation and available for locations worldwide. These data sets are typically based on standardized typical meteorological years (TMY), including TMYx⁴⁸ and IWECC (International Weather for Energy Calculations) data.⁴⁹

To enable their use in Modelica-based simulations, EPW files are converted into MOS format using an open-source conversion tool from the Modelica Buildings Library.⁴³ For Central Europe, *BUILDa* includes a set of predefined weather files, converted from standardized EPW data sets.

3.3. Operational schedules

During operation, buildings may change due to various factors, such as retrofits of the building envelope or changes in occupancy. This directly affects the thermal dynamic behavior of buildings. *BUILDa* enables users to specify such

modifications by adjusting building envelope parameters or internal gain and window-opening profiles during runtime. For this purpose, a second, optional configuration file is provided. Users can define which parameters or profiles should change at a specific timestamp and whether heating or cooling loads should be recalculated after changes, which is particularly relevant in retrofitting scenarios. At the specified retrofit timestamp, the updated building parameters are applied, and the FMU is reinitialized with consistent default state values. This approach deliberately does not attempt to model the physical retrofit process itself, which would require detailed construction-phase modeling, is infeasible to capture accurately, and is typically not recorded or excluded in the data. Within each simulation segment, the underlying physical model remains energy-consistent and validated. *BUILDa* runs the simulation using the initial configuration until the defined timestamp, applies

Table 4. Role of different converter functions in Builda and their execution order.

Order	Function name	Explanation
1	Link_Resolver	Resolving linked parameters: A parameter value can be linked to another so that it automatically adapts its value. For example, the window-to-wall fraction can be defined once for one orientation, while the other orientations are linked to it. This avoids redundant inputs and keeps the configuration simple.
2	Miscellaneous_Handler	Incorporates functionality that does not fit in other converter functions. For example, translates some user parameter names to Modelica parameter names.
3	Model_Compatibility_Layer	Prevents conflicts in the model by automatically correcting inconsistent user inputs. For instance, to avoid division-by-zero errors in the FMU, parameters are set to values slightly above zero.
4	Zone_Dimensions_Calculator	Calculates the areas of roof, floor, windows, and exterior and interior walls for each orientation, based on the specified zone length, width, floor height, and number of floors.
5	Component_Configurator	Maps the predefined parameter values of wall construction profiles to the respective model parameter values (see Table 1).
6	RC_Distribution_Configurator	Maps predefined parameter values of R and C distribution profiles to the respective model parameter values when this option is enabled.
7	Component_Properties_Calculator	Computes R- and C-value distributions for component RC elements from U-value, heat capacity, element count, and zone dimensions.
8	Nominal_Heating_Power_Calculator	Calculates the nominal heating power of a building, based on building and environmental parameters (DIN 18599-2). ⁴⁰
9	Nominal_Cooling_Power_Calculator	Calculates the nominal cooling power of a building, based on building and environmental parameters (DIN 18599-2).

the specified changes, and then continues with the segment until the next change or the end of the run.

3.4. Converter layer

The converter layer is a key feature of Builda and utilizes the user-defined parameter values to automatically infer missing inputs and resolve parameter dependencies required for simulation. This ensures physical consistency for each run. For instance, the room area and volume are directly related. In the configuration file, only one value needs to be specified, while the dependent value is computed automatically and mapped to the respective Modelica model parameters. Similarly, cooling and heating power can be automatically recalculated for each building configuration for consistent physical behavior. The converter functions are described in Table 4. The functions are executed sequentially for each simulation to handle the correct resolution of the dependencies. The converter layer is specifically tailored to the FMU used in this work. Adapting it to a different FMU with a different set of parameters would require nontrivial adjustments and is beyond the scope of this paper. For users interested in adapting the framework to a different FMU, we provide brief guidance in the GitHub repository.

3.5. Software usage

First, the parameter values need to be specified in the configuration file by the user (see Appendix 1, Table 9 for a list of the major parameters). Builda provides various ways to construct a set of variations of the values for simulation, for example, setting a range of values with a step size or the construction of a Cartesian product. Optionally, the operational schedules can be configured as described in Section 3.3 “Operational Schedules.” Builda iterates through the set of variations and executes the simulations in parallel, utilizing all available CPU cores. The output is written to the file system as a CSV file, and output columns can be specified during configuration. The duration of a complete simulation run with Builda primarily depends on the size of the variation set, the controller’s update sampling time, the FMU state output interval, and the number of available compute cores. For instance, simulating the FMU with 100 variations over 1 year, using an output interval of 900 s, required approximately 1240 s (averaging 12.4 s per simulation) when using the internal controller. In contrast, the same simulation with an external two-point controller, updating the manipulated variable in 900-second steps, took approximately 1600 s (averaging 16 s per simulation). The simulation was executed on an Intel® Core™ i7-6600U CPU (2.60 GHz, 4 cores) running

Table 5. Example ranges for building parameter values.

Category	Component	Min.	Max.	Coverage
U-value (W/m ² /K)	Floor, walls, roof	0.1	1.4	Directly
	Window	0.7	4.3	Directly
Area (m ²)	Floor	62	116	$zone_length * zone_width$
	Walls	120	230	$2 * n_floors * floor_height * (zone_length + zone_width) - A_Windows$
	Roof	86	185	$zone_length * zone_width * f_ARoofToAFloor$
	Window (e.g. south side)	18	42	$floor_height * n_floors * zone_length * f_AWin_south$
Heat capacity (kJ/m ² /K)	Floor	270	500	Directly
	Walls	50	660	Directly
	Roof	35	400	Directly

Source: Adapted from Krug et al.¹⁸

Ubuntu 22.04 with 20 GB of RAM. The increased execution time in the externally controlled FMU is attributed mainly to additional recalculations within the FMU triggered by each controller update. The Builda framework is available on GitHub.⁵⁰

4. Demonstration

4.1. Data showcase

Builda can be configured to cover a wide range of parameter values, as shown in Table 5. These ranges are provided as guidelines to illustrate possible configurations and are consistent with the representative values of the TABULA database,⁵¹ which documents German buildings constructed from 1949 to the present. While the parameter ranges are based on German reference data, TABULA databases also exist for other European residential building typologies. It is important to note that the framework is not limited to TABULA values, and parameter values can be chosen freely. Most parameters can be specified directly by the user, while others are derived by the converter layer. The framework could, in principle, also be applied to represent commercial or industrial buildings. However, due to their typically larger size, approximating such buildings with a single-zone model is likely to be of limited utility.

Figure 3(a) presents the daily mean zone temperatures of five building variations over 1 simulation year, utilizing TMY files contained in Builda and neglecting internal gains. For temperature control, the internal proportional controller with a day setpoint of 22 °C and night setpoint 18 °C is used. As no cooling is applied, the mean temperatures vary significantly depending on parameter combinations and solar radiation. Figure 3(b) shows temperature curves for four variations on a September day, illustrating the effect of different controllers. The upper curves correspond to an internal proportional controller, while the

lower curves represent a two-point controller. Both controllers use a setpoint of 22 °C during the day and 18 °C during the night. While all buildings heat up similarly during the day, the two-point controller generates a distinct pattern by repeatedly switching the thermal source on and off to maintain the setpoint.

The retrofitting capabilities of Builda, as described in Section 3.3 “Operational Schedules,” are demonstrated in Figure 4. The simulation uses weather data of Munich (TMY) and neglects internal heat gains. Here, after 2 days of simulation, the external wall is refurbished, reducing its U-value. Two days later, the windows are replaced with newer ones, which also decrease their U-value. The setpoint temperature for the zone is constantly at 20 °C. The temperature is controlled using a proportional-integral controller. These two retrofitting measures lead to a significant reduction in the required heating power. In addition to external factors such as outside air temperature and solar radiative gains, they have a substantial impact on heating demand.

To further illustrate the impact of the building parameters, we show in Figure 5 how parameters influence the energy consumption (left figure) and the 90th percentile of the indoor air temperature (right figure) of the buildings over 1 year. To compare different climates, we analyzed European locations with varying outdoor temperatures and solar radiation, using the TMY files of Builda. Internal gains are again neglected. For temperature control, the internal proportional controller with a day setpoint of 22 °C and night setpoint 18 °C is used. Higher U-values increase heating energy demand while reducing indoor temperature peaks (90th percentile). Prague exhibits the highest heating energy consumption, followed by London and Zurich, while Zurich shows the highest temperature peaks. Larger zones (120 m²) require more heating energy but tend to have slightly lower temperature peaks than smaller zones (60 m²).

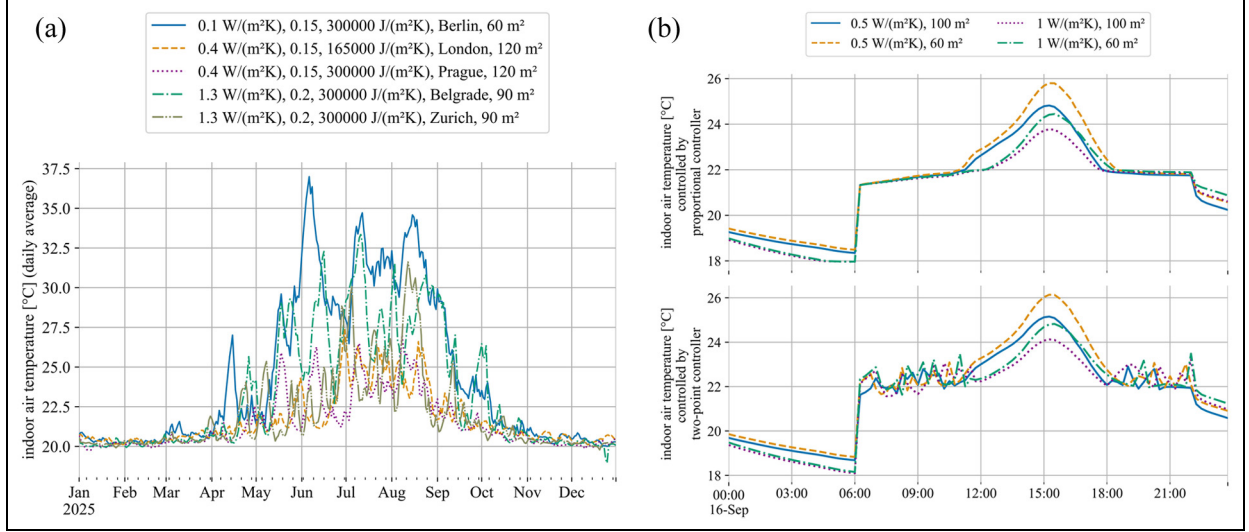


Figure 3. (a) Daily average temperatures over 1 year for various simulations. The parameter variation in the legend is listed as follows: external wall's U-value, window-to-wall fraction, external wall's heat capacity, location, and zone area. (b) Indoor air temperatures over 1 day for different simulated buildings in Munich, using proportional and two-point controllers. Upper figure: proportional controller. Lower figure: two-point controller. The parameter variation in the legend is listed as follows: the external wall's U-value and zone area.

Source: Adapted from Krug et al.¹⁸

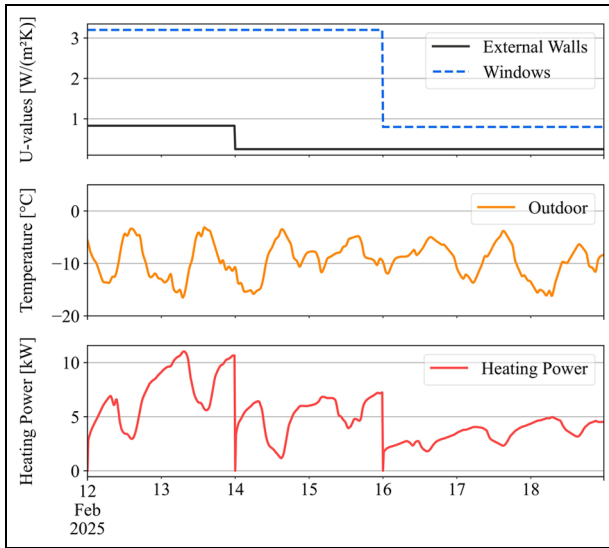


Figure 4. Demonstration of the impact of a retrofit on the heating energy demand by improving the heat insulation of external walls and windows.

These results highlight the inverse relationship between heating demand and peak indoor temperatures; better insulation (lower U-value) reduces heating needs but leads to higher peak temperatures.

The full simulation configurations, including controller setpoints and all other building-defining parameter settings used to generate Figures 3–5, are provided in the Builda repository to ensure full reproducibility.

4.2. Validation

To ensure accuracy of the data, we validated the building model according to the ANSI/ASHRAE 140-2004 standard using the following test cases:⁵² TC600 (low-mass building with heating and cooling) and TC900 (high mass building with heating and cooling), evaluated for annual heating and cooling demand; TC600FF (low-mass building without heating or cooling) and TC900FF (high mass building without heating or cooling), evaluated for minimum, maximum, and mean temperature. The evaluation of the test cases is based on a comparison of the results with simulation outputs from a range of other established simulation tools. A result is considered valid if it falls within the corresponding reference range. The validation results and reference ranges are shown in Table 6.

4.3. TL study

4.3.1. Study introduction. We now present an example of how to use Builda for TL research by pretraining various source models and fine-tuning them on target buildings. We focus on the problem of source building selection, as choosing the right source is crucial for successful TL in building thermal dynamics.^{13,15} In-depth studies on this topic are still rare, as research at the building parameter level is challenging due to limited access to metadata (see Section 2 “Background”) or the need to manually prepare different physical simulation models.¹⁶ We, therefore, use Builda to demonstrate how building parameter variations

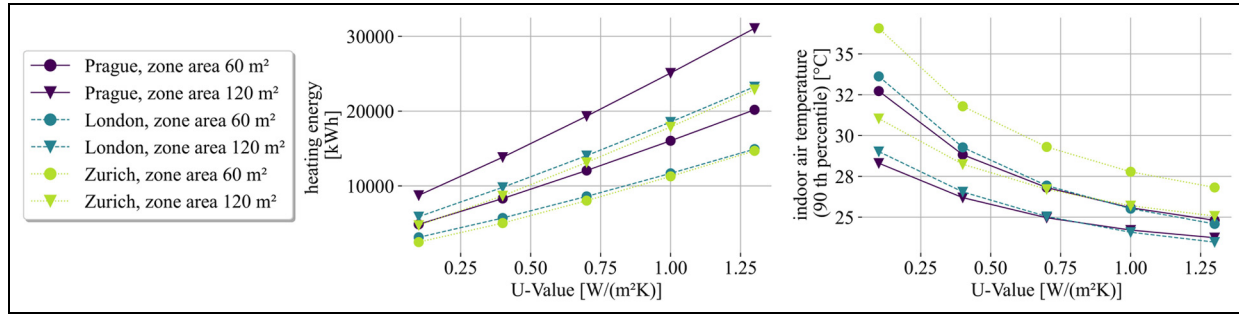


Figure 5. Relation between parameters: U-value, weather, zone area, and the output (left: heating energy, right: the 90th percentile of the indoor air temperature).

Source: Adapted from Krug et al.¹⁸

Table 6. Result of validation with ASHRAE test cases, heating/cooling demand in MWh, and temperatures in °C.

Case	Metric	Results Builda	Min.	Max.	In range
TC600	Heating demand	4.935	4.298	5.709	Yes
	Cooling demand	6.839	6.137	7.964	Yes
TC600FF	Average temperature	25.53	24.6	25.9	Yes
	Minimum temperature	−17.96	−18.8	−15.6	Yes
	Maximum temperature	67.8	64.9	69.5	Yes
TC900	Heating demand	1.847	1.17	1.988	Yes
	Cooling demand	3.173	2.132	3.415	Yes
TC900FF	Average temperature	25.53	24.6	25.9	Yes
	Minimum temperature	−4.36	−4.5	−1.6	Yes
	Maximum temperature	44.07	41.8	44.8	Yes

Table 7. Building parameter values of the simulation study.

Parameter	Source values	$\mathcal{T}^{(H)}$	$\mathcal{T}^{(L)}$
Wall U-value (W/(m²K))	0.10, 0.75, 1.40	0.2	1.3
Wall heat capacity (kJ/(m²K))	50, 250, 450	430	70
Window U-value (W/(m²K))	0.70, 1.90, 3.10	0.9	2.9
Floor area (m²)	60, 90, 120	110	70
Window-to-wall fraction	0.106, 0.128, 0.150	0.11	0.146

of source buildings can influence TL. This study does not aim to comprehensively explore all building parameters or their impact on TL for general conclusions about the building stock. Instead, it serves as an exploratory demonstration of how Builda can be used to systematically analyze TL behavior.

4.3.2. Study design. For the source buildings, we select five different building parameters that we want to vary with Builda. We chose parameters that have a strong

influence on the building's thermal dynamics, namely, wall U-values, wall heat capacity, window U-values, window area, and floor area (see Section 3.2.1 “Building Parameters”). The selected parameter ranges correspond to the ranges of single-family houses depicted in Table 5. Table 7 lists the resulting values. All other simulation parameters were left at their default values, assuming no occupancy (no internal gains), evenly distributed window areas across the north, east, south, and west facades, and a two-story building. We further assumed a monolithic wall structure and used weather data for Munich (TMY). Data were sampled at 15-min intervals, using the internal controller for heating with a lower and upper setpoint of 18°C (night) and 22°C (day), respectively. The corresponding Builda configuration file used in this study, containing all details, is available in the Builda repository.

Note that the window area is defined by the window-to-wall fraction. Assuming buildings with a square floor plan, for a value of 0.106, this results in a minimal window area of 20 m² for the building with a floor area of 60 m², while a value of 0.15 results in a maximal window area of 40 m² for a building with 120 m² floor area, mirroring the TABULA values. Furthermore, the maximum value of the window U-value parameter was chosen to be 3.10, as this

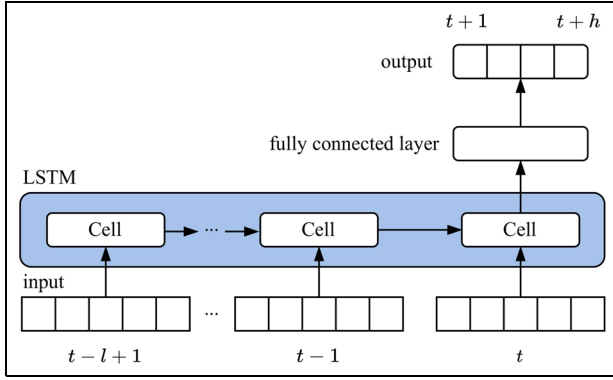


Figure 6. LSTM overview.

Source: Adapted from Raisch et al.¹²

value better represents the characteristics of the current building stock. We employ all possible parameter combinations, getting 243 different source buildings.

For the target buildings, we chose two parameter combinations at the edge of the distribution, representing a building with high thermal inertia $\mathcal{T}^{(H)}$ and low thermal inertia $\mathcal{T}^{(L)}$, respectively. The respective parameter values are listed in Table 7 and are distinct from the source building values. By choosing targets on the edge of the distribution with opposing dynamics, effects on the building parameter level should become more easily apparent. It is important to note that evenly spaced sampling of additional targets in the building parameter domain does not guarantee a uniform coverage of thermal dynamics, due to the inherently nonlinear mapping between physical parameters and resulting thermal behavior. Consequently, this study focuses on two opposing target cases, allowing to more clearly observe how variations in source building parameter values affect TL performance. Except for the varied parameters, all other building parameters are identical between the source and target buildings.

The task we consider is the forecast of the indoor air temperature, depending on current conditions, that is, the weather (direct solar radiation q_{dir} , diffuse solar radiation q_{diff} , outside air temperature T_{out}), indoor air temperature T_{in} , and the control signal of the thermal source u . The input features of our data-driven models are thus: $[q_{\text{dir},t}, q_{\text{diff},t}, T_{\text{out},t}, T_{\text{in},t}, u_t]$ at time t . For the model, we chose an LSTM⁵³ with an additional fully connected layer and a forecast horizon of 1 hour h (the next four steps), similar to the implementation in Raisch et al.¹² For the lookback l of all our models, we chose 1 day, that is, 96 time steps, consistent with related studies on building similarity.¹⁵ A schematic representation of the Long Short-Term Memory (LSTM) architecture is depicted in Figure 6. We use the ADAM⁵⁴ optimizer and the mean squared error as the cost function.

For the experiment, we then follow the workflow described in Figure 7. We generate 1 year of data for the

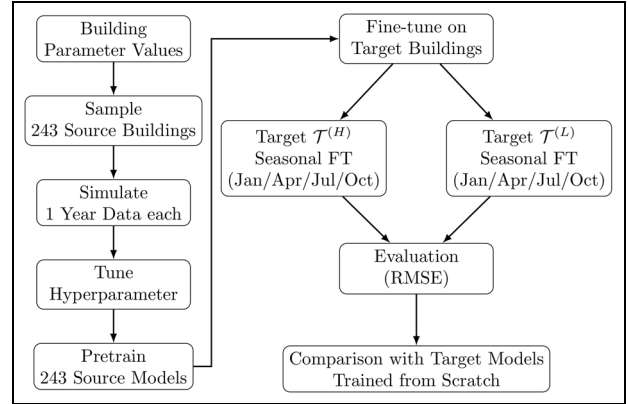


Figure 7. Workflow of the TL experiment. Source buildings are generated by sampling the building parameter values (Table 7) and simulated to produce 1 year of training data each. After hyperparameter tuning, source models are pretrained. These models are then fine-tuned on two target buildings ($\mathcal{T}^{(H)}$ and $\mathcal{T}^{(L)}$) using seasonal fine-tuning (FT). The resulting models are evaluated and compared with models trained from scratch.

Table 8. Hyperparameter search range.

Parameter	Range	Step
Hidden size	[64–128]	16
Layers	2, 3	1
Learning rate	[0.0001–0.0015]	Continuous
Batch size	[32–128]	32

target buildings and for each of the 243 source buildings. We use the source building data to pretrain 243 distinct source models. For each source model, we first perform hyperparameter tuning for the hidden size, the number of layers, the learning rate, and the batch size to get individual LSTM configurations (see Table 8); then, we pretrain the models. Afterward, each source model gets fine-tuned to the two targets, resulting in 486 fine-tuned models. The fine-tuned models inherit the hyperparameters from the pretrained source models. We rely on seasonal fine-tuning introduced by Raisch et al.,¹² as each season (winter, spring, summer, autumn) influences the building dynamics differently, hence influences the fine-tuning success. For each season, we assume limited target data are available, for example, 30 days in January for winter. The remaining data are used for evaluation, and the best model selection is performed on this set. The Root Mean Square Error (RMSE) of the selected model is reported as the test result. This design was deliberately chosen to avoid additional variance from data set splits. The goal is to assess the relative impact of source building parameter variations on TL performance rather than model generalization. The same procedure is applied to April, July, and October, and the

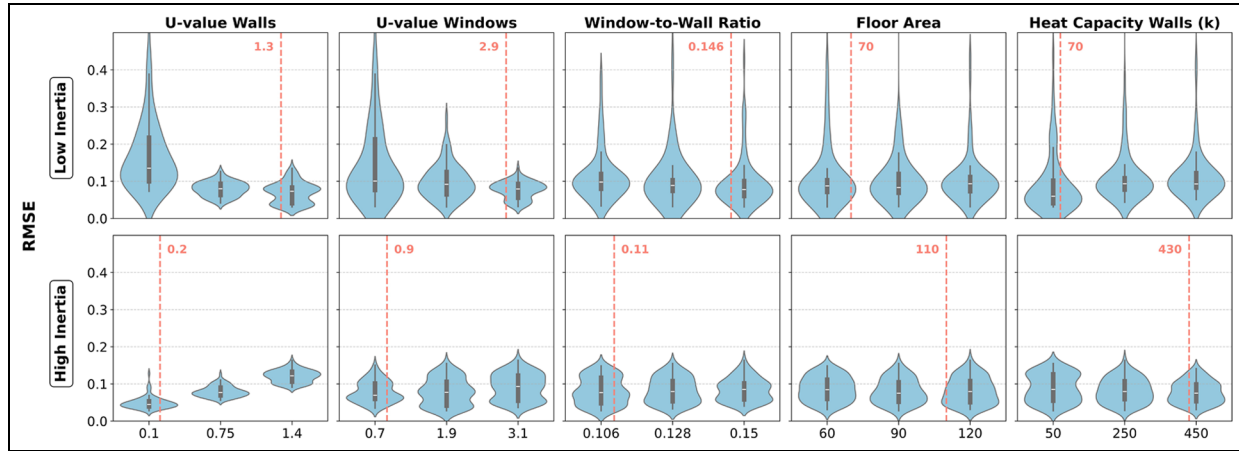


Figure 8. The results of the fine-tuned models for both targets are shown (upper figure: low-inertia target; lower figure: high-inertia target). For each parameter value, the RMSEs of all fine-tuned models exhibiting this value (81 per value) are displayed as violin plots. Dashed lines indicate the actual target values to which the models were fine-tuned.

seasonal results are averaged to provide a robust overall performance estimate. The pretraining and fine-tuning code is available on GitHub.⁵⁵

To compare the prediction performance of the fine-tuned models, we also train target models from scratch. Training from scratch means the data-driven model is trained with the target data (30 days) without having a pretrained model as a starting point. We do this again for each of the four seasons and average the results. We also performed hyperparameter tuning for the two target models (ranges see Table 8). We then compared the 486 fine-tuned models with the two target models trained from scratch.

4.3.3. Study results. Figure 8 shows the averaged RMSE error after fine-tuning (FT) for both targets. To allow an easy comparison of the results and to avoid distortions, the upper tails of the low-inertia violin plot are truncated. The two worst-performing models (RMSEs: 0.572 and 0.678) are included in the data but not fully shown. For the high-inertia target, a clear trend is only visible for the wall U-values. Wall U-values closer to the target improve FT performance. Other parameters show only very minor effects, and the overall spread of the mean RMSE is small, with values below 0.1 except for wall U-values.

For the low-inertia target, clear trends are visible for wall U-values, window U-values, and wall heat capacity. Window size and floor area have minimal impact. The more pronounced trends seem to indicate that for the low-inertia target, similar parameter values are more important for the FT success than for the high-inertia target. Furthermore, the spread of the RMSEs is significantly greater than for the high-inertia target. This indicates that,

for the low-inertia target, FT can still result in poor model performance, even when the parameter values of the source and target buildings are similar. This effect is particularly evident for wall heat capacity, window size, and floor area. This might indicate that FT to a high-inertia target is easier than to a low-inertia target.

Figure 9 shows how parameter relationships affect TL performance for both targets. For the high-inertia target, parameter value combinations with wall U-values close to the target values dominate all other parameter combinations. For example, even if both floor area and wall heat capacity values match the target, they are less influential for TL success than a matching wall U-value. Conversely, mismatched wall U-values cause the highest overall prediction errors, although similar values for the other parameters can still slightly reduce the errors.

For the low-inertia target, the dominant effect of the wall U-values remains visible, although the picture is less clear. When the wall U-value is combined with values close to the target for floor area, wall heat capacity, or window U-value, the fine-tuned models consistently perform very well. Interestingly, the window-to-wall fraction deviates from this trend, as larger window areas yield better performance. A window U-value close to the target also proves important, as shown by the best-performing combination of window U-value and wall heat capacity (average RMSE of 0.059). In comparison, the best-fitting combination of wall U-value and floor area achieves an averaged RMSE of 0.066, despite matching wall U-values. When neither wall nor window U-values match the target values, the worst-performing models occur (averaged RMSE of 0.284). Overall, this could indicate that for low-inertia targets, wall and window U-values are the most important parameters for selecting suitable source buildings.

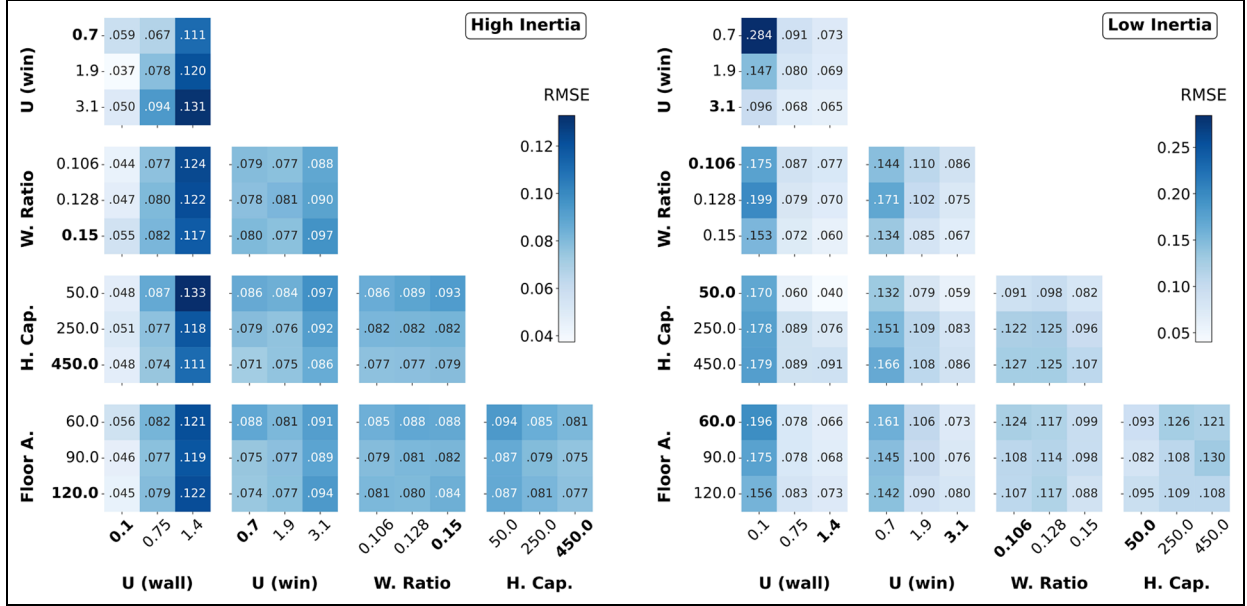


Figure 9. Influence of parameter relations on the fine-tuning success of all models for both targets (left: high-inertia target; right: low-inertia target). Each square shows the averaged RMSE of all models exhibiting the respective two combined values (27 for each square). The values closest to the respective targets are marked in bold.

The RMSE of the models trained from scratch is **0.97** for $T^{(H)}$ and **1.08** for $T^{(L)}$. As can be seen in Figure 8, the TL models consistently achieve lower RMSEs, indicating superior performance of the TL approach for the selected targets. This likely results from the similarities between source and target buildings, as only a fraction of all defining building parameters have been varied, and all buildings are exposed to the same weather.

In summary, the results anecdotally demonstrate that TL models can outperform models trained from scratch when only limited training data is available. Similar parameter values of the source to the target often result in a better model performance after FT. This effect seems to be especially important for the low-inertia target. Furthermore, FT appears easier for the high-inertia target, as indicated by consistently lower RMSEs and fewer outliers. However, it is still difficult to determine which source is best suited for a target, as unfavorable parameter value combinations can cause poor prediction performances, even when some parameter values closely match the target. Thus, similar building parameters might not necessarily imply similar thermal dynamics. Moreover, it remains unclear which building parameters are most critical for selecting an appropriate source building for successful TL. The results of our exemplary experiment suggest that the wall U-value is particularly influential. For the low-inertia target, there is an implication that, next to the wall U-value, the window U-value is important to consider, especially in combination with the former parameter. Although a broad range of

source building thermal dynamics has been covered with 243 different building configurations, only two dynamically opposing target buildings have been considered in the study. Consequently, the presented findings should be interpreted as indicative trends within the investigated experiment design rather than universally valid conclusions. However, the findings clearly underline that comprehensive and in-depth studies on building similarity are very much needed. We want to emphasize that, in addition to the selected parameters in our study, it is important to cover numerous other building parameter configurations, different climatic conditions, occupancy schedules, etc., as well as a larger set of target buildings to get a clear picture. *BuilDa* gives researchers a data generation tool to perform such comprehensive studies.

5. Conclusion and Future Work

We presented *BuilDa*, a highly flexible data generation framework for creating large amounts of high-variance thermal building dynamics time-series data, specifically designed for ML research. We described the variable physical building model and its features in detail, presented the overall architecture of the framework, and emphasized the distinct functionalities of *BuilDa*. The building model ensures physical consistency across all components, providing coherent and realistic thermal behavior. With *BuilDa* and its validated, configurable building model, it becomes easy to simulate a wide range

of building variations without requiring expert knowledge in building simulation. Building parameters, weather scenarios, operating conditions, and control strategies can be individually configured by the user, including the definition of retrofitting scenarios. Finally, we showcased examples of the heterogeneous data BuilDa can generate and demonstrated its use in an exploratory TL study.

One possible research application is the pretraining of generalized transformer architectures for building thermal dynamics forecasting, analogous to recent works on building load or power demand forecasting.^{56,57} Such models can then be fine-tuned for individual buildings and serve as base models for control strategies like MPC. Furthermore, BuilDa now enables comprehensive building similarity studies with a level of detail that is not yet present in the TL community for building thermal dynamics. By allowing fine-grained parameter adjustments, systematic investigations of FT strategies, such as those explored by Chaudhary et al.¹⁶, become possible, but applicable to a broader range of buildings, including residential ones. This also supports deeper analyses of the relationships between building parameters and resulting thermal dynamics. The presented TL study in this paper lays the groundwork for future investigations that will explore these aspects in greater depth.

While we already made our framework available online (see Section 3.5 “Software Usage”), we plan to provide key functionalities of BuilDa as an additional Python package, reducing barriers and allowing seamless integration of the variable building model into other research projects. By wrapping the FMU-interface with Gymnasium,⁵⁸ this will also extend our framework toward RL research, similar to systems such as Sinergym.³³ However, the variability of our physical model allows multiple buildings to be easily configured as RL training environments. In future work, we also plan to provide different base models representing various building types and multi-zone buildings, moving toward a universal and user-friendly ML research platform for building thermal dynamics.

Acknowledgements

The authors would like to express sincere thanks to Timo Germann and David Broos for their assistance with the project programming.

ORCID iD

Thomas Krug  <https://orcid.org/0009-0000-4907-1254>

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This project is funded by the Helmholtz Association’s Initiative and Networking Fund through Helmholtz AI.

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Data availability

The BuilDa framework, including configuration files for all demonstration experiments and the TL study, is available on GitHub.⁵⁰ The code for the TL study, along with usage instructions, is provided in a separate repository on GitHub.⁵⁵

References

1. United Nations Environment Programme. 2022 *global status report for buildings and construction: towards a zero-emission, efficient and resilient buildings and construction sector*. Nairobi, Kenya: United Nations Environment Programme, 2022.
2. Yao Y and 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. <https://doi.org/10.1016/j.buildenv.2021.107952>
3. Drgoňa J, Arroyo J, Cupeiro Figueroa I, et al. All you need to know about model predictive control for buildings. *Annu Rev Contr* 2020; 50: 190–232. <https://doi.org/10.1016/j.arcontrol.2020.09.001>; <https://www.sciencedirect.com/science/article/pii/S1367578820300584>
4. Weinberg D, Wang Q, Timoudas TO, et al. A review of reinforcement learning for controlling building energy systems from a computer science perspective. *Sust Cities Soc* 2023; 89: 104351. <https://doi.org/10.1016/j.scs.2022.104351>; <https://www.sciencedirect.com/science/article/pii/S2210670722006552>
5. Yu L, Qin S, Zhang M, et al. A review of deep reinforcement learning for smart building energy management. *IEEE Internet Things J* 2021; 8(15): 12046–12063.
6. Chen Z, O’Neill Z, Wen J, et al. A review of data-driven fault detection and diagnostics for building hvac systems. *Appl Energ* 2023; 339: 121030.
7. Al Sayed K, Boodi A, Sadeghian Broujeny R, et al. Reinforcement learning for hvac control in intelligent buildings: a technical and conceptual review. *J Build Eng* 2024; 95: 110085. <https://doi.org/10.1016/j.job.2024.110085>; <https://www.sciencedirect.com/science/article/pii/S235271022401653X>
8. Pinto G, Wang Z, Roy A, et al. Transfer learning for smart buildings: a critical review of algorithms, applications, and future perspectives. *Adv Appl Energ* 2022; 5: 100084.
9. Peirelinck T, Kazmi H, Mbuwir BV, et al. Transfer learning in demand response: a review of algorithms for data-efficient modelling and control. *Energy AI* 2022; 7: 100126.
10. Chen Y, Tong Z, Zheng Y, et al. Transfer learning with deep neural networks for model predictive control of hvac and natural ventilation in smart buildings. *J Clean Prod* 2020; 254: 119866.
11. Jiang Z and Lee YM. Deep transfer learning for thermal dynamics modeling in smart buildings. In: *Proceedings of the 2019 IEEE international conference on big data (big data)*, Los Angeles, CA, 9–12 December, pp.2033–2037. IEEE.
12. Raisch F, Krug T, Goebel C, et al. Gentl: a general transfer learning model for building thermal dynamics. In: *Proceedings of the 16th ACM international conference on future and*

- sustainable energy systems (*e-energy'25*), Rotterdam, 17–20 June, pp.322–333. Association for Computing Machinery. <https://doi.org/10.1145/3679240.3734589>
13. Pinto G, Messina R, Li H, et al. Sharing is caring: an extensive analysis of parameter-based transfer learning for the prediction of building thermal dynamics. *Energy Build* 2022; 276: 112530.
 14. Li H, Pinto G, Piscitelli MS, et al. Building thermal dynamics modeling with deep transfer learning using a large residential smart thermostat dataset. *Engineering Applications of Artif Intel* 2024; 130: 107701.
 15. Zhang Q, Niu J, Tian Z, et al. A study on source domain selection for transfer learning-based cross-building cooling load prediction. *Energy Build* 2024; 324: 114856. <https://doi.org/10.1016/j.enbuild.2024.114856>
 16. Chaudhary G, Johra H, Georges L, et al. Transfer learning in building dynamics prediction. *Energy Build* 2025; 330: 115384. <https://doi.org/10.1016/j.enbuild.2025.115384>; <https://www.sciencedirect.com/science/article/pii/S0378778825001148>
 17. Raisch F, Langtry M, Koch F, et al. Adapting to change: a comparison of continual and transfer learning for modeling building thermal dynamics under concept drifts. *Energy Build* 2026; 354: 116868. <https://doi.org/10.1016/j.enbuild.2025.116868>; <https://www.sciencedirect.com/science/article/pii/S0378778825015981>
 18. Krug T, Raisch F, Aimer D, et al. Builda: a thermal building data generation framework for transfer learning. In: *Proceedings of the 2025 annual modeling and simulation conference (ANNSIM)*, pp.1–13, <https://arxiv.org/abs/2508.12703>
 19. Miller C, Kathirgamanathan A, Picchetti B, et al. The building data genome project 2, energy meter data from the ASHRAE great energy predictor III competition. *Sci Data* 2020; 7(1): 368
 20. Miller C and Meggers F. The building data genome project: an open, public data set from non-residential building electrical meters. *Energy Proc* 2017; 122: 439–444.
 21. Li H, Wang Z and Hong T. A synthetic building operation dataset. *Sci Data* 2021; 8(1): 213.
 22. Pullinger M, Kilgour J, Goddard N, et al. The ideal household energy dataset, electricity, gas, contextual sensor data and survey data for 255 UK homes. *Sci Data* 2021; 8(1): 146.
 23. Luo N and Hong T. Ecobee donate your data 1,000 homes in 2017. Technical report, Pacific Northwest National Lab (PNNL), Richland, WA, 2022.
 24. Mattsson SE and Elmqvist H. Modelica: an international effort to design the next generation modeling language. *IFAC Proc Vol* 1997; 30(4): 151–155.
 25. Crawley DB, Lawrie LK, Winkelmann FC, et al. EnergyPlus: creating a new-generation building energy simulation program. *Energy Build* 2001; 33(4): 319–331. [https://doi.org/10.1016/S0378-7788\(00\)00114-6](https://doi.org/10.1016/S0378-7788(00)00114-6); <https://www.sciencedirect.com/science/article/pii/S0378778800001146>
 26. Roth A, Bull J, Criswell S, et al. Scripting frameworks for enhancing energyplus modeling productivity. In: *2018 building performance analysis conference and SimBuild co-organized by ASHRAE and IBPSA-USA, SimBuild conference, volume 8*. Chicago, IL, ASHRAE/IBPSA-USA, pp.312–319, https://publications.ibpsa.org/conference/paper/?id=simbuild2018_C043
 27. Fan C, Chen M, Tang R, et al. A novel deep generative modeling-based data augmentation strategy for improving short-term building energy predictions. *Build Simul* 2022; 15(2): 197–211. <https://doi.org/10.1007/s12273-021-0807-6>
 28. Hong T, Macumber D, Li H, et al. Generation and representation of synthetic smart meter data. *Build Simul* 2020; 13(6): 1205–1220. <https://doi.org/10.1007/s12273-020-0661-y>
 29. Charbonnier F, Morstyn T and McCulloch M. Home electricity data generator (HEDGE): an open-access tool for the generation of electric vehicle, residential demand, and PV generation profiles. *Methodsx* 2024; 12: 102618.
 30. Chaudhary G, Johra H, Georges L, et al. Synconn_build: a python based synthetic dataset generator for testing and validating control-oriented neural networks for building dynamics prediction. *Methodsx* 2023; 11: 102464.
 31. Goldfeder JA and Sipple JA. A lightweight calibrated simulation enabling efficient offline learning for optimal control of real buildings. In: *Proceedings of the 10th ACM international conference on systems for energy-efficient buildings, cities, and transportation (BuildSys'23)*, pp.352–356. Association for Computing Machinery. <https://doi.org/10.1145/3600100.3625682>
 32. Jia H and Chong A. eplusr: a framework for integrating building energy simulation and data-driven analytics. *Energy Build* 2021; 237: 110757.
 33. Jiménez-Raboso J, Campoy-Nieves A, Manjavacas-Lucas A, et al. Sinergym: a building simulation and control framework for training reinforcement learning agents. In: *Proceedings of the 8th ACM international conference on systems for energy-efficient buildings, cities, and transportation (BuildSys'21)*, pp.319–323. Association for Computing Machinery. <https://doi.org/10.1145/3486611.3488729>.
 34. Schamhorst P, Schubnel B, Fernández Bandera C, et al. Energym: a building model library for controller benchmarking. *Appl Sci* 2021; 11(8): 11083518. <https://doi.org/10.3390/app11083518>; <https://www.mdpi.com/2076-3417/11/8/3518>
 35. Blum D, Arroyo J, Huang S, et al. Building optimization testing framework (BOPTTEST) for simulation-based benchmarking of control strategies in buildings. *J Build Perform Simul* 2021; 14(5): 586–610.
 36. Dassault Systèmes. Fmpy, <https://github.com/CATIA-Systems/FMPy> (2023, accessed 28 March 2025).
 37. Blochwitz T, Otter M, Åkesson J, et al. Functional mockup interface 2.0: the standard for tool independent exchange of simulation models. In: *Proceedings of the 9th international Modelica conference*, pp.173–184. The Modelica Association, https://lup.lub.lu.se/search/files/5428900/2972_293.pdf
 38. Cohen SD, Hindmarsh AC and Dubois PF. CVODE, a stiff/nonstiff ODE solver in C. *Comput Phys* 1996; 10(2): 138–143. <https://doi.org/10.1063/1.4822377>; https://pubs.aip.org/aip/cip/article-pdf/10/2/138/11945852/138_1_online.pdf
 39. Verein Deutscher Ingenieure (VDI). Calculation of transient thermal response of rooms and buildings—modelling of rooms. VDI 6007-1:2015-06, 2015.
 40. Deutsches Institut für Normung (DIN). Energy efficiency of buildings: calculation of the net, final and primary energy demand for heating, cooling, ventilation, domestic hot water

- and lighting—part 2: net energy demand for heating and cooling of building zones. DIN 18599-2:2016-10, 2016.
41. OVUM Heat Pumps GmbH. Datenblatt: ACP-serie (ac208p, ac312p, ac417p, ac520p), 2025, https://www.ovum.at/wp-content/uploads/2025/09/250904_ACP_Datenblatt_CI2024_mitMIRA_web_DE.pdf (accessed 10 September 2025).
 42. Dott R, Haller M, Ruschenburg J, et al. The reference framework for system simulations of the IEA SHC task 44 / HPP annex 38 part b: buildings and space heat load, 2014. DOI:10.13140/2.1.2221.4727.
 43. Wetter M, Zuo W, Nouidui TS, et al. Modelica buildings library. *J Build Perform Simul* 2014; 7(4): 253–270. <https://doi.org/10.1080/19401493.2013.765506>
 44. Hensen JL and Lamberts R. *Building performance simulation for design and operation*. Routledge, 2011.
 45. Davies MG. *Building heat transfer*. John Wiley & Sons, 2004.
 46. Verein Deutscher Ingenieure (VDI). Calculation of thermal loads and room temperatures (design cooling load and annual simulation). VDI 2078:2015-06, 2015.
 47. Pflügl F, Lünsdorf O, Remmen P, et al. Loadprofilegenerator: an agent-based behavior simulation for generating residential load profiles. *Energy Build* 2022; 259: 111859. <https://doi.org/10.1016/j.enbuild.2022.111859>; <https://www.loadprofile-generator.de>. Version as used: see <https://github.com/FZJ-IEK3-VSA/LoadProfileGenerator>
 48. Lawrie LK and Crawley DB. Development of global typical meteorological years (TMYX), <http://climate.onebuilding.org> (2022, accessed 28 March 2025).
 49. ASHRAE. *International Weather for Energy Calculations (IWEK weather files) users manual and CD-ROM*. American Society of Heating, Refrigerating and Air-Conditioning Engineers, 2001.
 50. Krug T, Raisch F, Aimer D, et al. A highly configurable framework for large-scale thermal building data generation to drive machine learning research, <https://github.com/fabianraisch/BuilDa> (2026, accessed 15 June 2026).
 51. Loga T, Diefenbach N and Born R. *Deutsche Gebäudetypologie*. Wohnen und Umwelt, 2011.
 52. ASHRAE Standard 140-2004:2004. Standard method of test for the evaluation of building energy analysis computer programs.
 53. Hochreiter S and Schmidhuber J. Long short-term memory. *Neural Comput* 1997; 9(8): 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
 54. Kingma DP and Ba J. Adam: a method for stochastic optimization. Arxiv Preprint 1412.69802014, <https://arxiv.org/abs/1412.6980>
 55. Krug T, Raisch F, Koch F, et al. A highly configurable framework for large-scale thermal building data generation to drive machine learning research, https://github.com/thomaslkrug/BuilDa_TL_Experiment (2026, accessed 15 June 2026).
 56. Hertel M, Beichter M, Heidrich B, et al. Transformer training strategies for forecasting multiple load time series. *Energy Inform* 2023; 6(1): 20. <https://doi.org/10.1186/s42162-023-00278-z>
 57. Gokhale G, Van Gompel J, Claessens B, et al. Transfer learning in transformer-based demand forecasting for home energy management system. In: *Proceedings of the 10th ACM international conference on systems for energy-efficient buildings, cities, and transportation (BuildSys'23)*, pp.458–462. Association for Computing Machinery. <https://doi.org/10.1145/3600100.3626635>
 58. Towers M, Kwiatkowski A, Terry J, et al. Gymnasium: a standard interface for reinforcement learning environments, arXiv preprint arXiv:2407.17032, 2024, <https://arxiv.org/abs/2407.17032>

Author biographies

Thomas Krug, MSc, is a Research Associate at Technical University of Applied Sciences Rosenheim, Germany, and a doctorate candidate at Karlsruhe Institute of Technology, Germany (cooperative doctorate with Technical University of Applied Sciences Rosenheim). His research interests include data-driven modeling of building dynamics, transfer learning, and reinforcement learning for intelligent building control. His email address is thomas.krug@th-rosenheim.de.

Fabian Raisch, MSc, is a Research Associate at the Technical University of Applied Sciences Rosenheim, Germany, and a doctorate candidate at the Technical University of Munich, Germany (cooperative doctorate with Technical University of Applied Sciences Rosenheim). His research interests include transfer learning, data-driven system identification, and control of the building environment. His email address is fabian.raisch@th-rosenheim.de.

Dominik Aimer, MSc, is a Research Associate and doctorate candidate at the Technical University of Applied Sciences, Rosenheim, Germany. His research interests include technical monitoring, IoT and simulation development, physical modeling, energy optimization, and data-driven modeling. His email address is dominik.aimer@th-rosenheim.de.

Markus Wirnsberger, MSc, is a Research Associate and doctorate candidate at the Technical University of Applied Sciences, Rosenheim, Germany. His research focuses on the simulation and monitoring of building HVAC systems to optimize energy consumption and comfort. His email address is markus.wirnsberger@th-rosenheim.de.

Ferdinand Sigg is a Professor at Munich University of Applied Sciences HM, Germany, and a doctorate candidate at the Technical University of Munich, Germany. His research interests include building physics, energy-efficient HVAC systems, and building performance simulation. His email address is ferdinand.sigg@hm.edu.

Felix Koch, MSc, is a Research Associate at Technical University of Applied Sciences Rosenheim, Germany, and a doctorate candidate at Karlsruhe Institute of Technology, Germany (cooperative doctorate with Technical University

of Applied Sciences Rosenheim). His research is centered around foundational models for structured data and generalization across building dynamics. His email address is felix.koch@th-rosenheim.de.

Benjamin Schäfer is an Assistant Professor at the Department of Informatics at the Karlsruhe Institute of Technology (KIT), Germany. He investigates power systems with the help of (explainable) machine learning and stochastic modeling. Before joining KIT, he worked in Göttingen (Germany), Dresden (Germany), London (United Kingdom), Tokyo (Japan), and Ås (Norway),

graduating with a PhD thesis in 2017. His email address is benjamin.schaefer@kit.edu.

Benjamin Tischler is a Professor at the Faculty of Applied Sciences and Humanities at the Technical University of Applied Sciences Rosenheim, Germany. His research interests include machine learning for modeling and optimal control of energy systems and buildings, machine learning operations, and language models. Before joining THRO, he worked in Cologne (Germany) and Munich (Germany), graduating with a PhD thesis in 2015. His email address is benjamin.tischler@th-rosenheim.de.

Appendix I

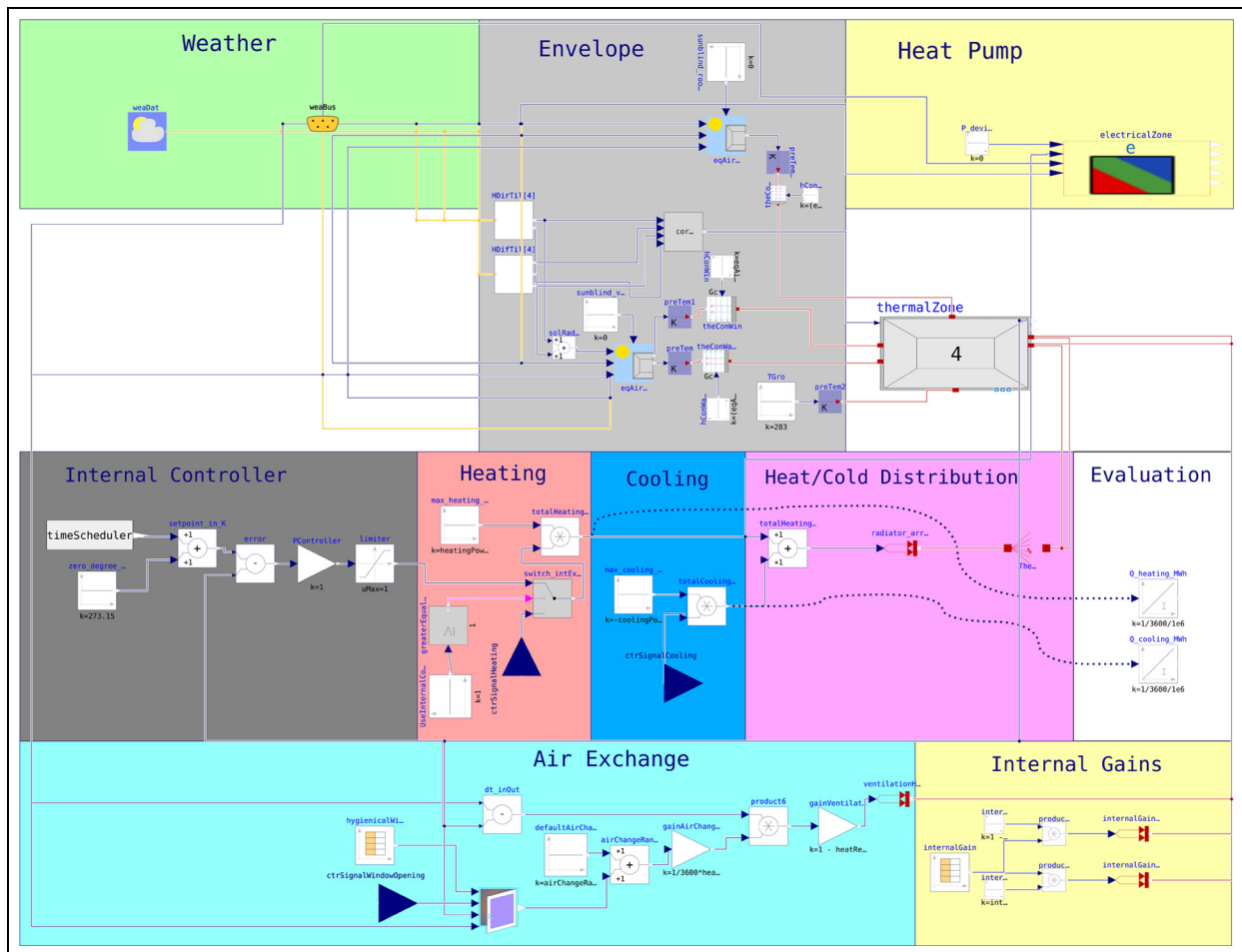


Figure 10. Overview of the implemented model in the Modelica editor with the following sections: Weather, which provides an interface to the external weather file; Envelope, the interface between the thermal zone and the environment; Heat Pump, which calculates the required electric energy for heating using a heat pump; Internal Controller, which controls the heat emission by the heating system based on the setpoint temperature; Heating, which converts the controller output from normalized to absolute heating power and provides an external controller interface; Cooling, which converts the controller output from normalized to absolute cooling power for the external cooling controller; Heat/Cold Distribution, which divides the supplied heating/cooling energy into radiative and convective fractions and forward it to the thermal zone; Evaluation, which calculates the total heating and cooling demand during the simulation; Air Exchange, which calculates heat losses induced by air exchange through infiltration, mechanical ventilation, or window opening, and provides an interface for window-opening profiles or externally generated signals; Internal Gains, which calculates heat gains induced by occupants or electrical devices, defined in an external file.

Table 9. Variable building simulation parameters of BuildDa.

Parameter name	Explication	Unit
zone_length, zone_width	Length (from east to west) and width (from north to south) of the zone	m
n_floors, floor_height	Number and height of floor levels	–, m
fAWin_south (analogously for west, north, east)	Window-to-wall fraction on the southern, Western, northern, and eastern walls	–
fATransToAWindow	Fraction of transparent window area to overall window area	–
fARoofToAFloor	Fraction of roof area to floor area (if inclined roof, this factor is > 1)	–
fAInt	Factor of exterior wall surfaces to interior wall surfaces (both sides)	–
UExt	U-value of the external walls	W/(m ² K)
(analogously for intWall, floor, roof)	Internal wall, floor, and roof	W/(m ² K)
heatCapacity_wall	Heat capacity of the exterior walls related to their area	J/(m ² K)
(analogously for intWall, floor, roof)	Internal wall, floor, and roof	J/(m ² K)
heatCapacity_furniture_per_m2	Specific heat capacity of the furniture related to the floor area	J/(m ² K)
UWin	U-value of the windows	W/(m ² K)
thermalZone.gWin	G-value of the windows	–
weaDat.fileName	Path(s) to weather file(s)	–
internalGain.fileName	Path(s) to internal gain profile file(s)	–
hygienicalWindowOpening.fileName	Path(s) to hygienical window-opening profile file(s)	–
heatRecoveryRate	Heat recovery rate of the ventilation system	–
airChangeRate	Air change rate due to the ventilation system or infiltration	–
roomTempLowerSetpoint	Lower temperature setpoint (for night setback on internal controller)	–
roomTempUpperSetpoint	Upper temperature setpoint (for night setback on internal controller)	–
UseInternalController	Decides if the model internal controller for heating is used	–
extWall_C_distribution	Heat capacity distribution profile for the external wall	–
(analogously for intWall, floor, roof)	Internal wall, floor, and roof	–
extWall_R_distribution	Heat resistance distribution profile for the external wall	–
(analogously for intWall, floor, roof)	Internal wall, floor, and roof	–
internalGainsConvectiveFraction	Fraction of internal gains that are convective	–
heatingConvectiveFraction	Fraction of heating that is convective	–
relative_heatPump_efficiency	Efficiency of the heat pump compared with the reference heat pump	–
heatingCurve_steepness	Steepness of the heat curve; higher values indicate a greater increase in supply temperature relative to the difference between outside and inside temperatures. Typical range: 0.2–1.5	–
#extWall_construction	Parameter that can be used to configure one of the predefined component constructions (see Section 3.2.1 “Building Parameters”)	–

Source: Adapted from Krug et al.¹⁸