Titre: Title:	Building energy model calibration using a surrogate neural network
Auteurs: Authors:	Florent Herbinger, Colin Vandenhof, & Michaël Kummert
Date:	2023
Type:	Article de revue / Article
Référence: Citation:	Herbinger, F., Vandenhof, C., & Kummert, M. (2023). Building energy model calibration using a surrogate neural network. Energy and Buildings, 289, 113057 (15 pages). https://doi.org/10.1016/j.enbuild.2023.113057

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/53703/
Version:	Version finale avant publication / Accepted version Révisé par les pairs / Refereed
Conditions d'utilisation: Terms of Use:	Creative Commons Attribution-Utilisation non commerciale-Pas d'oeuvre dérivée 4.0 International / Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND)

Document publié chez l'éditeur officiel
Document issued by the official publisher

Titre de la revue: Journal Title:	Energy and Buildings (vol. 289)
Maison d'édition: Publisher:	Elsevier
URL officiel: Official URL:	https://doi.org/10.1016/j.enbuild.2023.113057
Mention légale: Legal notice:	

Building Energy Model Calibration Using a Surrogate Neural Network

Florent Herbinger*, Colin Vandenhof, Michaël Kummert

*Corresponding author (florent.herbinger@polymtl.ca)

Department of Mechanical Engineering, Polytechnique Montréal, Montréal, Canada

Abstract

Energy studies of buildings are becoming more widespread as stakeholders strive to improve energy efficiency and reduce carbon emissions. As a result, there is an increased need for novel numerical techniques to automatically calibrate building energy models (BEMs) for these energy studies. In this paper, a new automated building calibration methodology was developed, which uses a surrogate (*i.e.*, meta) multilayer perceptron artificial neural network (ANN) to infer unknown building parameters. Typically, in the field of BEM calibration, surrogate models are used as a time-saving technique. Instead of running the building simulation software during each iteration of an optimization search, a surrogate model can approximate the output of the BEM very quickly, leading to much faster optimization of the unknown building parameters. However, we show that, once trained, the surrogate model itself can be used to find the unknown building parameters. Since the surrogate model is differentiable, gradient descent can be used to find the building parameters that minimize the error between true and predicted metered energy consumption. Previous surrogate modeling approaches for calibrating BEMs map only the unknown building parameters to the BEM's output. As a result, they are blind to the effects of other crucial variables like weather and the building's schedules. On the other hand, our surrogate ANN model accounts for these other predictors of energy consumption. With this advantage, our method was able to outperform a powerful black box optimizer when finding 14 unknown building parameter values in a controlled case study with hourly energy data. In a real metered data case study, our ANN method and the black box optimizer performed similarly on average, but our method had less variance in performance across trials.

Keywords

Building Energy Model Calibration; Building Energy Modeling; Surrogate Modeling; Metamodeling; Emulators; Machine Learning; Artificial Neural Network

1 Introduction

Around 18 % of global carbon emissions originated from residential, commercial, and institutional buildings in 2019 [1]. Therefore, an effective carbon reduction strategy for institutions, cities, and building owners is to improve the energy efficiency of their buildings. However, deciding what aspects of an existing building should be improved requires extensive and detailed studies. Building energy modeling can suggest how the energy consumption and carbon emissions of a building will change in response to retrofits, the addition of renewable energy sources, or other initiatives. Building energy modeling is also very useful for continuous commissioning, where the ongoing performance of a building is assessed to ensure that it is working properly and efficiently. However, the degree of confidence in the results from modeling largely depends on the level of *calibration* of the building energy model (BEM). With a well-calibrated BEM, we can expect that it accurately represents the building we are modeling. Consequently, the energy consumption given by the BEM closely matches the building's actual energy consumption, allowing us to better assess the impact of retrofits, for example.

Calibrating a BEM is a difficult task primarily due to the complexity of modern buildings, resulting in a large number of independent and interdependent attributes affecting energy performance. BEMs therefore need to be either complex with large number of parameters, leading to the *curse of dimensionality* [2], or be reduced to a simpler but less accurate representation. Furthermore, there is a lack of measured data from which to estimate the unknown building parameters. In general, building energy models are *overparametrized* (too many parameters) and *undetermined* (not enough validation points to infer the parameters). Finally, there is no unique calibration solution, leading to the additional challenge of *equifinality* [3], [4].

Increasingly, to find a calibration solution, the vast building parameter space is searched using black box optimization techniques (*e.g.*, genetic algorithms), which create a map between the unknown building parameters and the aggregated error between the BEM's output and the metered data. These black box approaches are often blind to how external and internal variables—like the weather and the building's schedules—directly map to the BEM's energy consumption. During each hour of the year, the weather and building's occupancy, controls, and electricity schedule at that hour (and previous hours) have interactive effects with the unknown building parameters. For example, late at night when the sun is long set, the solar heat gain coefficient of the windows (SHGC) has zero to very little impact on the energy consumption, while during the day, the brighter the sun, the more impact the SHGC has on the energy consumption. Since black box optimizers are often blind to this interaction, they have less data from which to infer the unknown building parameters, leading to poorer results. To include weather and building schedules as inputs, the measured data used for calibration must be on an hourly or sub-hourly time scale. This paper explores a new calibration methodology based on hourly measured data, which uses an ANN surrogate model that takes as input *both* the unknown building parameters and external and internal variables like the weather and the building's schedules. Including this larger breadth of inputs results in faster and more robust calibration.

1.1 Building Energy Models

There are two types of building energy models (BEMs): physics-based and data-driven. Physics-based (*i.e.*, forward) models use physical laws (*e.g.*, heat transfer), along with the building's known physical properties, to predict its behaviour (*e.g.,* energy consumption). Data-driven (*i.e.*, inverse) models work in the other direction, by using data about the building's behaviour to infer its physical properties [5].

Physics-based models are more appropriate for assessing the impacts of retrofits and other energy efficiency or production initiatives since physics-based models (as opposed to data-driven models) can be altered to apply these retrofits. The user can then have an idea of how the building's energy consumption will change from these retrofits. It is therefore very important that the building model be accurate, *i.e.*, *calibrated*. Such models can be calibrated using historical data about the building's behaviour, such as the metered energy consumption or the temperature in one of the building's rooms. The model is first created by fixing the known or safely assumed building parameters, and then the remaining, unknown building parameters of the model are adjusted until the model's behavior aligns most closely with the historical behavior of the real building.

Having a building model with accurate (*i.e.,* calibrated) parameters allows us to simulate potential upgrades to the real building. For example, we can upgrade the windows in our model to quickly obtain a prediction of how the building's energy consumption would be reduced in the real world.

A physical building energy model generally has the following parameters:

- the building shape and dimensions;
- window size, placement, and properties;
- materials and dimensions for the floors, walls, and roof;
- heating and cooling systems, including service hot water;
- equipment and lighting;
- occupant type, density, and schedule; and
- controls, such as schedules and values for setpoint temperatures and service hot water usage.

While some of these parameter values may be known or easy to estimate, other values may be unknown and difficult to obtain, even with access to detailed information about the building. It is these unknown values that must be determined while calibrating the BEM.

1.2 Building Energy Model Calibration

A building energy model (be it single or multiple/urban) is said to be sufficiently calibrated if the error between the historical metered output of the actual building(s) and the simulated output of the model is below a certain threshold [6]. The error is measured at regular intervals (*e.g.*, hourly, monthly) and aggregated over a certain time period (generally a year); building energy simulation software such as EnergyPlus or TRNSYS (TRaNsient SYstem Simulation tool) are typically used to generate the outputs of the model. Table 1 summarizes some of the commonly used thresholds defined by various organizations.

Table 1 : Calibration criteria thresholds for building calibration from various organizations.

Calibration Criteria	ASHRAE Guideline 14 [6]	FEMP [7]	IPMVP [8]
Monthly CVRMSE	15 %	15 %	-
Hourly CVRMSE	30 %	30 %	20 %
Monthly NMBE	±5 %	±5 %	-
Hourly NMBE	±10 %	±10 %	±5 %

One commonly used error is the coefficient of variation of root mean squared error (CVRMSE), which unlike the normalized mean bias error (NMBE), does not suffer from the cancelling of positive and negative bias. Equation (1) and (2) below show the formula for calculating the CVRMSE and NMBE:

$$CVRMSE = \frac{1}{\bar{m}} \sqrt{\frac{\sum_{i=0}^{n} (m_i - s_i)^2}{n - p}} \times 100\ (\%) \tag{1}$$

$$NMBE = \frac{1}{\bar{m}} \frac{\sum_{i=0}^{n} (m_i - s_i)}{n - p} \times 100\ (\%) \tag{2}$$

where m_i is the measured value, s_i is the simulated value, n is the number of measured values, and p is suggested to be equal to 1 [9]. The closer the values of the building parameters in our simulation model are to the actual building parameters (*e.g.*, wall conductance), the lower the NMBE and CVRMSE will be. In industry, the model is typically manually calibrated by a building energy modeler, which can be time-consuming, expensive, and prone to the modeler's biases. It is also heavily reliant on the modeler's expertise, so the accuracy of the calibration is often uncertain [10]. Therefore, numerical optimization techniques are increasingly being used, which attempt to minimize error functions like the CVRMSE by searching the building model's parameter space. The accuracy of the parameters returned by the optimizer depends not only the final CVRMSE error, but also on the quality of the data that was used to perform the optimization. For example, if the metered energy or weather data is inaccurate, the results of the optimization may be poor.

It is difficult and time-consuming to create accurate, calibrated physics-based building models. The major hurdle is that the models are *over-parametrized and under-determined.* There are usually many parameters of the building model that need to be estimated, but relatively few data points that can be used to estimate them. A potential simple fix for these problems is to simplify the BEM by reducing the number of building parameters, but this simplification will almost always increase the calibration error. Managing the trade-off between the complexity of a model and the ability to calibrate all of its parameters can be challenging.

1.3 Surrogate Modeling in BEM Calibration

Since physics-based BEMs are generally required to be quite complex to mimic actual building behavior, calibrating a set of unknown building parameters requires searching through a very large, multi-dimensional space. Due to this curse of dimensionality, it may take thousands or even millions of BEM simulations for naïve optimization methods to find an acceptable solution [11], [12], [13]. Running a BEM simulation in building performance software like EnergyPlus or TRNSYS can take minutes to even hours if computational fluid dynamics (CFD) calculations are involved, leading to calibration routines that can take many days to even weeks. There are three main ways to address these enormous computational requirements [14]:

(1) Supercomputing,

(2) Cloud computing, and

(3) Surrogate or metamodeling.

Supercomputing is expensive and not easily accessible to building energy researchers or practitioners; cloud computing is a cheaper alternative and is increasingly being offered by building performance software; however, executing tens of thousands of simulations can still take days to weeks on the cloud [15], [16]. Surrogate modeling can be implemented locally, on the cloud, or on supercomputers to shorten the computation time of each iteration of an optimization search. Surrogate models (also known as metamodels or emulators) are essentially "a model of model" and are used to approximate a complex function that maps input data to output data, such as a building simulator. Surrogate models are helpful when the underlying function is computationally expensive, or otherwise difficult to run. In BEM calibration, instead of having to execute the computationally expensive building simulation software during each iteration of an optimization search, the trained surrogate model can be fed the unknown parameter values and its output can be quickly retrieved. Another advantage is that one can select a *differentiable* model as the surrogate, which allows gradient-based optimization methods (*e.g.*, gradient descent) to be directly applied.

The construction of a surrogate model starts by collecting training data. Sample points are collected from the input space, and we then record how the underlying function (*e.g.,* simulator) maps these inputs to output data. Next, the surrogate model is trained in a supervised fashion to approximate the function's behavior, through optimization of the model parameters to reduce error over the training set. Techniques for selecting the appropriate surrogate model for a given application is an area of ongoing research.

1.4 Automated BEM Calibration in the Literature

As an alternative to manual calibration, which is time-consuming, sometimes ineffective, and prone to biases, automated calibration methods have been developed that use optimization techniques to find the values of unknown building parameters. Various calibration techniques that are used in automated BEM calibration are listed in Table 2.

Table 2: Techniques used in automated building calibration in the literature.

Techniques used in calibration	**Type of BEM created**
Optimization search algorithms vary the building parameters in a physical model to minimize a function describing the level of calibration of a building (*e.g.* NMBE, CVRMSE) [17], [18]; evolutionary algorithms, specifically genetic and particle swarm optimization algorithms, are among the most popular types [19].	Physics-based
Sensitivity analysis (SA) determines the importance of building parameters on energy consumption through multiple building energy simulations, and parameters with little effect (sensitivity) are ignored [20].	Physics-based
Bayesian inference modeling quantifies the uncertainty of the calibrated parameters by generating a probability distribution over parameter values as opposed to a single estimate [21], [22], [23], [24], [25], [26].	Physics-based or Data-driven
Data disaggregation infers the different individual loads in a building through an aggregate energy signal, originating from a whole-home power monitor for example [27].	Data-driven
Surrogate modeling (*i.e.***,** metamodeling) reduces the computational expense of each iteration in an optimization search by running a statistical approximation of the physical model instead of running the model itself [14], [21], [26], [28], [29], [30].	Data-driven

In the literature, optimization search algorithms work by iteratively running the physics-based BEM with different sets of unknown building parameter values, in order to minimize the error between simulated and metered data [17], [18], [25]. The optimizer iteratively searches the building parameter space until an error threshold is reached, the error is no longer decreasing, or a certain number of simulations have elapsed. The main advantage of this method over surrogate modeling is that there is no additional error introduced in the calculation of the objective function since the output of the BEM itself is used rather than the surrogate. Yet, surrogate models or metamodels are increasingly being used as they greatly reduce the computational expense of calibrating BEMs, and they can approximate BEMs with increasing accuracy.

There are many different types of surrogate models that can be used to approximate a complex function such as the output of a building simulation software. Nagpal et al. [28] developed 27 different surrogate models, either random forests or multilayer perceptron neural networks, to see how well they could estimate the outputs of three different detailed physics-based BEMs—an academic building, a laboratory, and a residence, all on the MIT campus. The random forests were more accurate when there were many unknown building parameters (~28), while the neural networks performed better when there were fewer (~12). The selected surrogate models reduced the simulation time of each iteration of an optimization search from around 30 second to around 0.06 seconds; however, the monthly CVRMSEs for heating and cooling ranged from 6 to 62 % between the case studies, meaning that, overall, the BEMs remained uncalibrated (since the monthly CVRMSE is well above the accepted threshold of 15 %—see Table 1).

Østergård, Jensen, and Maagaard [14] compared a few more different types of surrogate models, namely multiple linear regressions, support vector regressions, multivariate adaptive regression splines (MARS), random forests, Gaussian processes (GP), and artificial neural networks. In one of their tests, they determined how well the different surrogate models could predict the yearly aggregated energy consumption of a generic office building with 14 unknown building parameters, modeled in the BSim software. The GP model performed the best on average across training dataset sizes of 82 to 8192 simulations, followed by ANNs and MARS; however, when the training dataset size was around 2000 simulations and above, the difference between the GP models and the ANNs was negligible.

One way to incorporate surrogate models in building energy model calibration is to replace the physics-based BEM in the evaluation of the objective function that is being optimized. For example, Chen et al. [29] developed multiple linear regression (MLR) and Gaussian process surrogate models that were trained on 50, 100, 200, or 500 EnergyPlus simulations. They then used their trained surrogate models to calibrate a campus building BEM with 8 unknown parameters using an active-set optimizer [31]. Only the unknown building parameters were used as inputs to their surrogate models. They achieved a monthly CVRMSE of 14 % using their GP models, which consistently outperformed their MLR models. The authors found that 50 BEM simulations was sufficient for their calibration procedure, which took around an hour to complete. They showed that their calibration method took a lot less time than the several hours necessary when using an optimization search with a genetic algorithm, but they did not report any performance differences.

Another widely used method in BEM calibration with surrogate modeling is incorporating Bayesian inference [11], [21], [22], [23], [24], [25], [26]. Bayesian modeling/calibration is widely adopted as, unlike deterministic calibration methods like optimization search algorithms, it provides a measure of confidence about the values of each calibrated building parameter. Furthermore, since multiple calibration solutions always exist and the unknown building parameters are not uniquely identifiable, Bayesian calibration offers an opportunity for the modeler to use their judgement in selecting a solution since the uncertainty is quantified. Bayesian calibration requires a large number of building simulations; therefore, a variety of surrogate models have been used to reduce the computation time. Lim and Zhai [32] compared the performance of five surrogate models for Bayesian calibration: multiple linear regressions (MLR), artificial neural networks, support vector machines, multivariate adaptive regression splines, and Gaussian processes (GP). They found that GP surrogate models were the most accurate in estimating the true values of 6 unknown parameters and predicting 17 monthly measured energy consumption of a medium office building. The MLR model was the least accurate, followed by the ANN. However, since the ANN training time using 100 physics-based BEM simulations was only 1.58 seconds, compared to 1223.29 seconds for the GP model, we are led to question whether the tested ANN was large enough (the ANN size is not disclosed in the article). Overall, GP surrogate models have been commonly adopted in the literature, given that they can both accurately approximate the building simulator's output and provide a measure of uncertainty [21], [26], [33].

Yuan et al. [26] trained a GP model on 200 EnergyPlus simulations of a seven-story office building energy model. The GP model only had the 10 unknown building parameters as inputs and had the monthly electricity consumption as the output. The final monthly calibration level over 24 months was ~8%, well-below the ASHRAE threshold of 15%.

GP models suffer more from the curse of dimensionality than other surrogate models, so if the physics-based BEM is complex with a lot of unknown parameters, a surrogate ANN is more appropriate. Cant and Evins [30] used a surrogate two-layer perceptron ANN for their approximate bayesian computation (ABC) calibration study—the number of neurons used is not disclosed. ABC is used when the probabilistic likelihood function, a required function in Bayesian modeling, is challenging or impossible to define. The inputs of the ANN only included the 23 unknown building parameters, like in every other study mentioned, and the outputs were 108 monthly electricity and gas consumptions. The surrogate ANN was trained on 300 EnergyPlus simulations of a fairly complex physics-based BEM following a Latin hypercube sampling distribution. The neural network simply approximated the BEM, and an external Bayesian inference package called PyMC3 was used to determine final probability distributions over building parameter values. The monthly CVRMSE was ~8 % when averaging the three energy outputs for the maximum a posteriori (MAP) calibration results.

As a final note, parallels can be drawn between building energy model calibration and building design optimization; both endeavours attempt to find the set of building parameter values that minimize some objective function. There are a few papers that use surrogate artificial neural networks combined with a genetic algorithm to optimize a building's design, such as the cost of retrofit measures [34], annual energy consumption [34], [35], [36], [37], daylighting autonomy [37], or thermal comfort [34], [36]. In a slightly different approach, Eisenhower et al. [38] built a surrogate model—this time a support vector model—and employed gradient-based optimization with IPOPT (Interior Point OPTimizer) [39], as opposed to the direct search used by genetic algorithms. However, in all of these building design optimization studies, the objective function only considers the ANN output and not a difference between simulated and metered data, which is required for BEM calibration.

1.5 Key Contributions of the Article

In the BEM calibration literature, surrogate models have been used predominately as a time-saving measure by approximating the output of a physics-based BEM [28], [29], [30]. Since a neural network is differentiable, it is possible to not only quickly approximate building simulator outputs, but also to infer inputs from known outputs via gradient descent. To the best of our knowledge, applying this inference method to surrogate ANNs has not been demonstrated in any other study. However, parameter input inference from known outputs has already been demonstrated in at least two papers using surrogate GP models [21], [26], but instead of gradient-descent, matrix inversion is employed. We identify a crucial limitation in this method and many other surrogate BEMs: they fail to include many important external and internal variables that influence energy consumption, in addition to the fixed parameters of the building. Specifically, weather and building schedules data are not incorporated into the models. Since these variable features interact strongly with the fixed building parameters, they greatly affect the energy consumption of a building, and excluding them from a model severely restricts its predictive power. Our study therefore uses a fairly large surrogate neural network (16384 hidden neurons) that includes a much larger breadth of relevant building/weather variables to predict the building's energy consumption. The surrogate model includes the fixed, unknown building parameters *and* variables like the hourly weather and building schedules, thereby resulting in more robust and faster calibration. We tried the exact same surrogate ANN method without these variables, and we found that the performance was much worse. The fact that hourly measured data is used for calibration allows us to incorporate these additional inputs; most surrogate model studies calibrate with monthly data, where the hourly weather and building schedule data cannot be incorporated.

Furthermore, we argue that the CVRMSE or MBE is not a sufficient calibration criterium, as it is blind to the actual physics unfolding in the building. A low CVRMSE does not necessarily indicate that the building model parameter values are close to actual building model parameter values; a value for a building parameter that is too high, such as window conductance, may be partially cancelled out by a value that is too low, such as air infiltration. The novel methodology employs a surrogate neural network that has learned the impact that the building parameter, weather, and building schedules has on the energy output of the model, including secondary impacts such as the impact of wind speed on infiltration which is not nearly as present for window conductance. By learning how all of these variables interact, the neural network can better model the real building physics. In this way, it can find building parameters that not only achieve low CVRMSE but are also likely to match those of the real building.

1.6 Baseline for the Surrogate Neural Network Technique

The results from the novel surrogate neural network method, developed in this paper, are compared to a high-performing black box optimization search algorithm developed by COIN-OR called RBFOpt. COIN-OR, COmputational INfrastructure for Operations Research, is a project that provides open-source mathematical software, including optimization packages like RBFOpt. The RBFOpt algorithm is founded on the radial basis function method (hence the name: RBF), originally proposed by Gutmann in his 2001 paper [40]. The algorithm is a statistical global optimization method, which builds and iteratively improves a surrogate model of the unknown objective function.

The RBFOpt optimization package was chosen as a comparison for two reasons: (1) It is also founded on surrogate modeling, thereby providing an apples-to-apples comparison, and (2) most importantly, it has proven to have the fastest and most robust algorithm for architectural design as demonstrated by Wortmann's benchmarking study [41]; Wortmann compared the performance of eight different algorithms across seven simulation-based problems relating to structural, energy, and daylighting optimization, where the number of variables varied from 4 to 40. The types of algorithms tested were evolutionary algorithms (notably genetic algorithms), other direct search algorithms, and model-based algorithms (RBFOpt). The RBFOpt algorithm had the best overall performance among the 8 algorithms tested.

In the context of this study, RBFOpt's solver operates as follows:

1) Construct a surrogate model that outputs a CVRMSE prediction, given a set of building parameters as input.
    a) Choose a random set of building parameter values,
    b) run the physics-based BEM with these values,
    c) calculate the error between the BEM's simulated energy consumption and the metered energy consumption (the hourly CVRMSE over the year),
    d) repeat steps (a) to (c) several times, keeping the chosen building parameter values and resulting CVRMSE in memory, and
    e) train the surrogate model on this initial data.
2) Use the surrogate model to find the set of building parameters with minimal *predicted* CVRMSE.
3) Run the BEM with these building parameters and calculate the *true* CVRMSE. That is, the error between the metered energy consumption and the energy consumption given by the BEM.
4) Update the surrogate model on this newly acquired data.
5) Repeat steps (2) to (4) until the preallotted number of iterations have elapsed.
6) And finally return the set of building parameters that gave the lowest true CVRMSE.

2 Methodology

2.1 Overview of the Surrogate Neural Network Technique

The surrogate neural network technique uses four steps to find the building parameter values. First, a few thousand simulations of the building model to be calibrated are executed to generate the training and validation datasets. Second, the generated data is organized into two large matrices: the input matrix consisting of building parameter, weather, and schedules (**X**) and the target matrix with the hourly simulated heating and cooling consumption (**Y**). Third, the surrogate neural network is trained in a supervised way to predict the simulated heating and cooling consumption at an hour of the year, given the building parameter, weather, and schedules at that hour. Finally, backpropagation on the neural network is used to infer the building parameter values that minimize the error between the true and predicted energy consumption over the year (given known weather and building schedules).

A python package was coded to run parallelized building simulations with either TRNSYS or EnergyPlus, collect the simulation data, create, train, validate, and test the neural network, and infer the building parameters. In the Results and Discussion (section 3), we first present the effect of including the weather and building schedules on the surrogate ANN calibration method's accuracy. Then, the speed and robustness of the method is compared to the high-performing black box optimization package called RBFOpt.

2.2 Case Study Building Energy Model

To test the surrogate neural network technique, a building energy model was developed for White Hall—a 4265 m^2 academic building on the Cornell University campus in Ithaca, New York (Figures 1 and 2). White Hall makes for an ideal case-study as it is the one of the oldest buildings on campus and has been renovated several times, so the characteristics of the building in its present state are unclear. Finding the properties of the envelope, energy systems, and detailed schedules would require extensive and time-consuming audits, which are not guaranteed to be accurate. On the other hand, the building floor area, number of floors, orientation, basic construction, basic temperature setpoint schedule as well as the basic ventilation system and schedule are known; the window coverage can also be accurately estimated via on-the-ground images. Most importantly, detailed heating, cooling, and electricity power consumption has been collected since 2015. White Hall is heated by a 1st generation district heating system, alimented by a central combined heat and power plant, and cooled by a state-of-the-art district cooling system, using a nearby lake as the cold source. The condensate pounds of steam and the cooling BTUs measured every 15 minutes at the service station in White Hall was integrated to find the hourly heating and cooling energy consumption. Electricity kWh was directly measured by meter at the building every 15 minutes and was also integrated over the hour. There are gaps in the data where the energy consumption is falsely reported to be zero; these hours were not considered in the calibration routines.

White Hall is a relatively heavy building with four floors above ground and a partially underground basement. The basement and first three floors have 26” to 30” (~66 to ~76 cm) limestone walls, and the top floor’s walls and the roof are made of wood.

The rudimentary physics-based building energy model for White Hall was developed in TRNSYS 18, which outputs heating and cooling power consumption for the 8760 hours of the year[1]. Setpoint temperatures follow a fixed schedule, with a setback from 7:00 PM to 4:00 AM. The equipment/lighting schedule follows the metered electricity consumption perfectly, meaning that the hour of the year which had the peak electricity consumption has a schedule factor of 1, and the factors for the other hours vary between 0 and 1 based on this normalized electricity consumption. With this setup, the simulated electricity consumption of our BEM will always be the same as the metered electricity consumption. Consequently, the calibrated model cannot be used to assess energy conservation measures relating to lighting or equipment power. However, this is just one setup; in other cases, the dynamic electricity usage could also be calibrated.

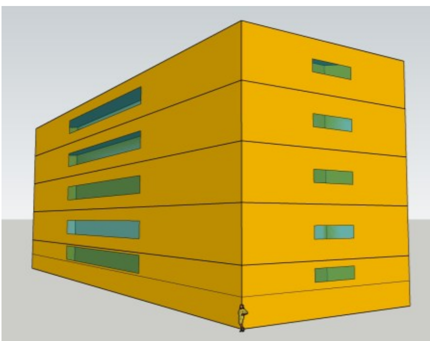

Figure 1: Sketchup model of White Hall with simple, proportional Windows.

Figure 2: Google Street view of White Hall.

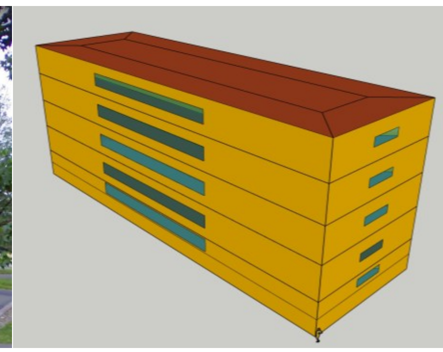

Figure 3: ASHRAE zoning for the White Hall building energy model.

The factors for the occupancy and hot water schedules is based on *operating schedule D* for schools/universities of the 2020 Canadian National Energy Code of Buildings (NECB) [42]. There are three different day types (weekday, Saturday, and Sunday) in *operating schedule D*; however, only the weekday type is used since the weekend day types has zero occupancy throughout the day, which is not reflective of White Hall’s occupancy patterns. Therefore, in the model, the occupancy schedule for a particular day of the year uses three different scales of the weekday type, which roughly follow three categories: (a) academic weekday, (b) academic weekend/summer weekday, and (c) summer weekend, where the schedule is scaled by a factor of 1.0, 0.2, or 0.1 respectively, depending on the maximum electricity consumption during that day. Figure 4 shows the three different categories of days present in the occupancy schedule alongside the metered electricity consumption.

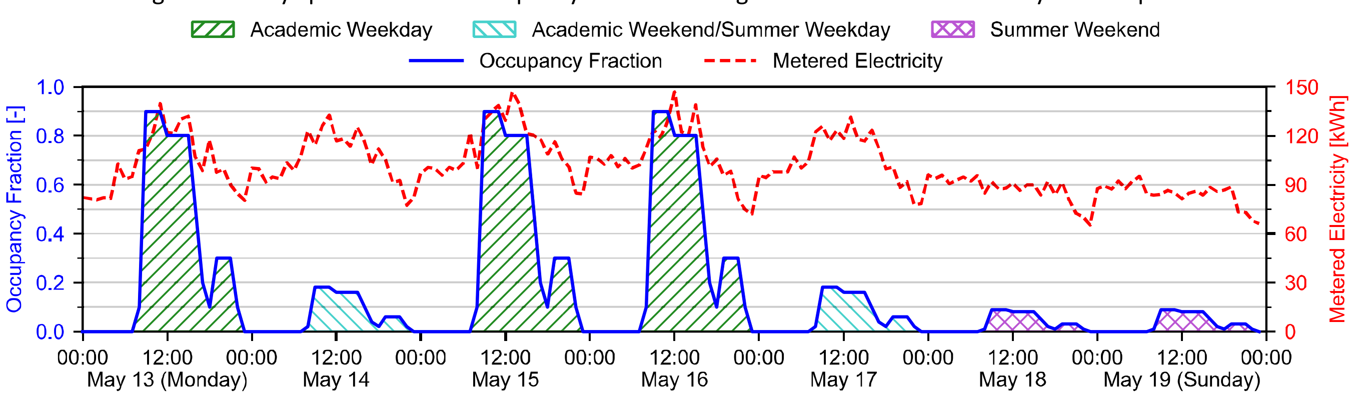

Figure 4: Visualizing the three different day profile occupancy schedules by scaling the NECB’s weekday type, alongside the electricity consumption.

Building loads are calculated and assumed to be met by idealized heating and cooling systems that consider the impact of outside air. The total heating and cooling required by the simplified HVAC system are assumed to be delivered by the district heating and cooling networks; these simulated energy loads are compared to the metered energy data collected at the

[1] As a side note, the Python package that was developed for the methodology is also capable of processing EnergyPlus models.

networks' substations at the White Hall building. An actual meteorological year (AMY) weather file for 2019 was obtained from White Box Technologies [43], which was recorded at the Ithaca Tompkins Airport weather station (~4.8km away from White Hall). Core-perimeter ASHRAE zoning is used for each floor of White Hall (as seen by the roof lines in Figure 3), with each thermal zone having equivalent operating schedules. To summarize, Table 3 shows all the known/assumed external and internal variables of the BEM.

Table 3: Known/assumed metered energy, weather, and schedule variables (hourly time scale).

Category	Known / Assumed Data	Units	Definition
Metered energy	$\dot{Q}_{heat,met}$	kW	Heating energy consumption
	$\dot{Q}_{cool,met}$	kW	Cooling energy consumption
	$\dot{P}_{elec,met}$	kW	Electricity consumption
Weather	T_{out}	°C	Outside air dry bulb temperature
	RH	%	Outside air relative humidity
	u_{wind}	$\mathrm{m \cdot s^{-1}}$	Wind velocity
	θ_{wind}	degrees	Wind direction
	G_{bn}	$\mathrm{W \cdot m^{-2}}$	Hourly direct normal solar irradiation
	G_d	$\mathrm{W \cdot m^{-2}}$	Hourly diffuse solar irradiation
	$G_{bn,wallwin}$	$\mathrm{W \cdot m^{-2}}$	Direct normal solar irradiation on the walls and windows
	$G_{bn,roof}$	$\mathrm{W \cdot m^{-2}}$	Direct normal solar irradiation on the roof
	$G_{d,wallwin}$	$\mathrm{W \cdot m^{-2}}$	Diffuse and reflected solar irradiation on the walls and windows
	$G_{d,roof}$	$\mathrm{W \cdot m^{-2}}$	Diffuse and reflected solar irradiation on the roof
Schedules	f_{occ}	-	Fraction of occupancy. Follows a basic repeating weekday/weekend/holiday schedule
	$f_{d/n}$	-	Either 1 or 0. 1 indicates that the day setpoint temperature is used, while 0 indicates that the night setpoint temperature is used
	f_{shw}	-	Fraction of service hot water. Exactly the same as f_{occ}
	f_{elec}	-	Fraction of equipment/lighting use, which is the metered electricity consumption normalized between 0 and 1

2.3 Controlled and Real Metered Case Studies

The methodology was tested in both a controlled and a real metered case study. In the controlled case study, the controlled energy data is in fact energy data generated by a particular configuration of the White Hall BEM in TRNSYS. The goal of this case study is to test the surrogate neural network technique by looking for this configuration (*i.e.,* parameters values) and determining if the technique is faster and/or more accurate than using the RBFOpt optimizer to search for the parameter values. Although the simulated data are necessarily cleaner than real metered data, it is only in the simulated setting that we know the true value of the building parameters being estimated, which allows us to assess the accuracy of our approach.

For the real metered case study, the actual heating and cooling energy consumption for the 2019 calendar year is used as the "true" energy, with a 20-day period starting on September 27 removed since the reported consumption was unusual and indicative of some fault in the building. Therefore, we are calibrating over a 345-day period.

2.4 Parameters to Calibrate

We manually created different BEM configurations and iteratively tested different sets of unknown building parameters from a total of 25, noting their impact (i.e., sensitivity) on the BEM's simulated energy consumption. Table 4 lists the unknown building parameters that were not ultimately included in the model since they

(1) are almost fully collinear with other parameters,

(2) could become "known" parameters after a brief analysis of the metered energy consumption,

(3) worsen or do not improve the CVRMSE after optimization, and/or

(4) are not sensitive to the energy consumption.

Table 4: The unknown building parameters that were tested but not included in the final model.

Category	Building Parameters that Were not Included	Units	Definition
Envelope Properties	a_{inf}	h^{-1}	a infiltration coefficient in $ACH = a_{inf} + b_{inf} * deltaT + c_{inf} * windSpeed + d_{inf} * windSpeed^2$ (coefficients b_{inf} and d_{inf} are not considered in this model configuration)
	U_{wall}	$\mathrm{W \cdot m^{-2} \cdot K^{-1}}$	Thermal conductance of the exterior walls
	U_{roof}	$\mathrm{W \cdot m^{-2} \cdot K^{-1}}$	Thermal conductance of the roof
	U_{win}	$\mathrm{W \cdot m^{-2} \cdot K^{-1}}$	Thermal conductance of the windows
Setpoint temperatures	$TSet_{on}$	h	Temperature setback (or setup) daily start time
	$TSet_{off}$	h	Temperature setback (or setup) daily stop time
HVAC	fan_{on}	h	HVAC systems daily start time
	fan_{off}	h	HVAC systems daily stop time
	a_{reheat}	-	Factor between 0 and 1 indicating the fraction of reheat in the zones (where 0 indicates no reheat and 1 indicates a reheat amount equal to the difference between the maximum cooling demands and current cooling demands)

The 14 remaining unknown building parameters (Table 5) were selected to be calibrated, since they best represent the building and the building's energy consumption profile is sensitive to them. We did perform two variance-based (*i.e.,* Sobol) sensitivity analyses on these 14 parameters using either the annual heating or cooling consumption as the output; however, the analyses were not very informative since only a few parameters consume most of the total sensitivity, resulting in total sensitivity indices of less than 0.05 for many of the other parameters (0.05 is a threshold that has been commonly adopted to indicate the significance of a parameter). For the annual heating, only two parameters, vDotAirNew and heatLoss, contribute to ~55% of the total sensitivity, while for cooling fElec2Heat contributes to ~70% of the total sensitivity. These parameters are configured in the TRSNYS model in such a way that they have a large impact on the annual consumption, which is not necessarily indicative that they are the only important parameters for calibrating. Our manual testing revealed that the 14 chosen unknown building parameters all improve the BEM's error when included.

Table 5: The 14 unknown building parameters to be calibrated.

Category	Building Parameters to be Calibrated	Units	Definition	Range/ Controlled Value
Envelope Properties	c_{inf}	$\mathrm{s \cdot m^{-1} \cdot h^{-1}}$	c infiltration coefficient in $ACH = a_{inf} + b_{inf} * deltaT + c_{inf} * windSpeed + d_{inf} * windSpeed^2$ (coefficients a_{inf}, b_{inf}, and d_{inf} are not considered in this study)	0 – 0.3 Control: **0.1**
	U_{ins}	$\mathrm{W \cdot m^{-2} \cdot K^{-1}}$	Thermal conductance of the massless insulation layer of the exterior walls and roof	0.01 - 3 Control: **0.4**
Setpoint temperatures	$T_{heat,day}$	°C	Equivalent average heating setpoint temperature of the building during the day	15 – 19 Control: **19**
	$T_{heat,night}$	°C	Equivalent average heating setpoint temperature of the building during the night	15 – 19 Control: **18**
	$T_{cool,day}$	°C	Equivalent average cooling setpoint temperature of the building during the day	19.01 – 23 Control: **23**
	$T_{cool,night}$	°C	Equivalent average cooling setpoint temperature of the building during the night	19.01 – 23 Control: **23**
Service Hot Water	$\dot{v}_{shw}$	$\mathrm{L \cdot min^{-1}}$	Rated service hot water flowrate. The energy owing to water usage is calculated by $\dot{v}_{shw} f_{shw} \rho_{water} c_{p,water} \Delta T_{shw}$	0 – 3 Control: **2**
Occupancy	ρ_{occ}	$\mathrm{Pers \cdot m^{-2}}$	Occupant density (rated value, *i.e.*, value that corresponds to "1" in the schedule. That value is multiplied by a schedule fraction between 0 and 1)	0 – 0.3 Control: **0.05**
Equipment and Lighting	$f_{elec2heat}$	-	Fraction of electricity use that is converted into thermal gains in the building (between 0 and 1)	0.01 – 0.9 Control: **0.7**
Solar	$SHGC$	-	Global solar heat gain coefficient of the windows	0.05 – 0.7 Control: **0.25**
HVAC	$\dot{v}_{air,new}$	$L \cdot s^{-1} \cdot m^{-2}$	Flow rate of fresh outside air per unit floor area	0.2 – 1.5 Control: **0.45**
	$ctrl_{efficacy}$	-	The efficacy fraction of the control of the HVAC system. A value of 1 means that the system is completely shut off during the setback period, while lower fractions mean the system is always partially or fully on (for a value of 0)	0 – 0.7 Control: **0.3**
	$loss_{heat}$	kW	The baseline heating energy consumption level throughout the year, due to heating equipment losses	0 – 40 Control: **20**
	$loss_{cool}$	kW	The baseline cooling energy consumption level throughout the year, due to cooling equipment losses	0 – 10 Control: **10**

Finally, the unknown building parameter ranges are fixed based on previous modeling experience and domain knowledge about similar university buildings. These ranges are used to restrict how small or large the estimates of each building parameter could be. The controlled values were chosen randomly within these ranges, rather than always near the middle of the bounds.

2.5 Step One: Running the Building Energy Model Multiple Times

In the first step of the methodology, many yearly simulations of the White Hall building energy model are executed with a different set of building parameter values for each one. A centre-weighted Latin hypercube sampling distribution is used to sample the building parameter space. The yearly weather and schedules remain the same across all of the yearly simulations. To study the effect of the training data size (*i.e.,* the number of BEM simulations) on the ANN's accuracy, six differently sized training datasets were used to train an ANN (its hyperparameters are listed in section 2.9 Grid Search): N × 8760 rows are randomly sampled from a large dataset of 2600 yearly BEM simulations, where N = [200, 600, 1000, 1400, 1800, 2200]. An additional 200 × 8760 rows are randomly sampled and used as a validation dataset to stop the training of each ANN before they overfit the training dataset (*i.e.*, early stopping). The ANN's validation loss (*i.e.*, error) is recorded every 800 × 8760 rows and, once the slope over the last five validation losses is positive (meaning the loss is no longer decreasing), the training stops and another 200 × 8760 randomly sampled rows are used as a testing dataset to determine the final loss of the ANN.

Figure 5 shows the correlation between the training data size and the ANN's final testing loss, alongside the total computation time. The BEM simulation time to generate the training data as well as the ANN training, validating, and testing time are summed to determine the total computation time.

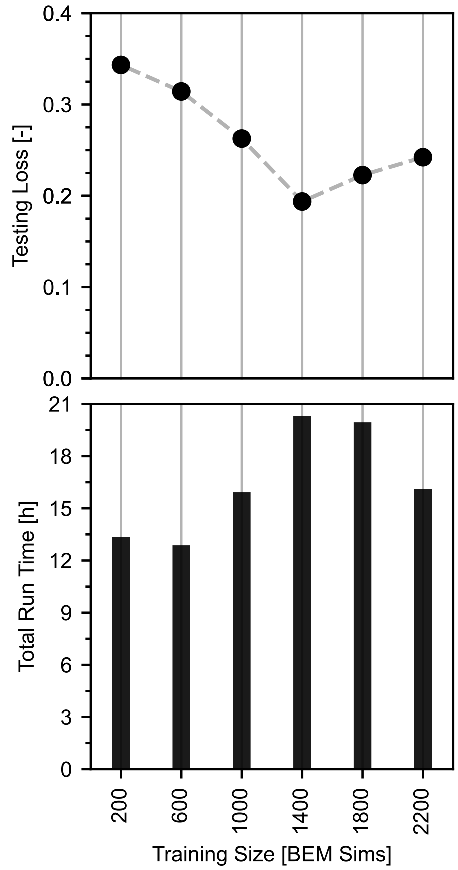

Figure 5: Effect of the training data size on the ANN's testing loss.

It was determined that a number of rows equivalent to 1800 simulations of the physics-based BEM offers enough training data for the neural network, while still being reasonable in the total computation time. 1400 simulations could also have been chosen, but we erred on the side of more training data. Furthermore, there is some variability in the final testing error as sometimes the ANN is underfitted because of the nature of the stopping condition; therefore, in other trial runs, training on 1800 simulations might have a lower testing loss than training on 1400 simulations and vice-versa. Choosing 1800 BEM simulations follows Jones, Schonlau, and Welch's [44] recommendation that the simulation dataset should be at least 10 times larger than the number of inputs (124 in our case) for expensive black box models. Finally, 200 additional BEMs are simulated to contribute to the validation dataset, making the training/validation split 90 % to 10 %.

2.6 Step Two: Collect Generated Data

Having executed 2000 yearly simulations of our BEM (1800 training + 200 validation), we now collect all the generated data into two matrices: input matrix **X** that contains the building parameter, weather, and schedules, and target matrix **Y** that contains the hourly simulated heating and cooling consumption. Each row of matrix **X** and **Y** corresponds to one hour of a yearly simulation. Each yearly simulation samples a new set of building parameters, while assuming identical weather and schedules data over the year; The inputs are standardized by removing the mean and scaling to unit variance, since ANNs tend to perform better on standard normally distributed data.

In addition to the dry bulb temperature, relative humidity, wind speed, and wind direction, four different solar irradiation variables were chosen for the weather input features: direct normal/diffuse on the walls/windows and direct normal/diffuse on the roof. These irradiation variables were chosen over the direct normal solar irradiation and diffuse horizontal irradiation since they are more directly linked to the energy consumption in the building and are therefore more relevant to the ANN model.

The weather and schedules data include not only the current hour, but also a number of previous hours. This “rolling window” of weather and schedules ensures that the neural network retains some of the dynamic, time-delayed behavior of building energy physics. For example, the hourly heating energy consumption is not only a function of what the outside temperature or building occupancy was at that hour, but also what the temperature and occupancy were during previous hours of the day. Figure 6 shows the effect between the ANN’s testing loss and the size of the rolling window for the weather and schedule inputs. The training, validation, and testing process is the same as the one described in section 2.5, and the ANN hyperparameters are the same as the ones chosen after the grid search (section 2.9).

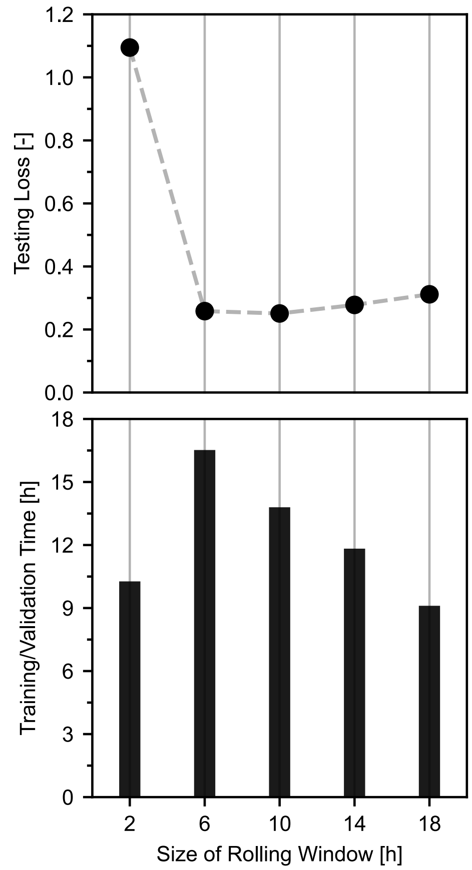

Figure 6: Effect of the size of the rolling window on the ANN’s testing loss.

A rolling window of 10 hours (*i.e.,* current hour + 9 previous hours) was chosen as it had the lowest loss of all rolling windows tested, while still having a reasonable ANN training/validation run time. It may appear counter-intuitive that larger rolling windows have higher ANN losses, but additional inputs that are mostly irrelevant to the building’s energy consumption can result in overfitting because of increased ANN complexity.

In summary, the list below covers all the data in matrix **X**:

Unknown building parameters

1. c_{inf}
2. U_{ins}
3. $T_{setheat,day}$
4. $T_{setback,night}$
5. $T_{setcool,day}$
6. $T_{setup,night}$
7. $\dot{v}_{shw}$
8. ρ_{occ}
9. $f_{elec2heat}$
10. $SHGC$
11. $\dot{v}_{air,new}$
12. $ctrl_{efficacy}$
13. $loss_{heat}$
14. $loss_{cool}$

Weather data

15. $T_{out,t}$
16. RH_t
17. $u_{wind,t}$
18. $\theta_{wind,t}$
19. $G_{bn,wallwin,t}$
20. $G_{bn,roof,t}$
21. $G_{d,wallwin,t}$
22. $G_{d,roof,t}$
23. 9 additional past hours for each weather variable
24. …
94. $G_{d,roof,t-9}$

Schedules

95. $f_{occ,t}$
96. $f_{elec,t}$
97. $f_{d/n,t}$
98. 9 additional past hours for each schedule variable
99. …
124. $f_{d/n,t-9}$

2.7 Step Three: Train and Validate the Surrogate Neural Network

The multilayer perceptron neural network was created using the PyTorch Python library, which allows the neural network to be trained on a GPU, thereby greatly reducing training time. The ANN has an input layer with 124 features—the 14 unknown building parameters as well as a rolling-window of eight meteorological variables and three schedule variables over the last ten hours (see section 2.6). The output layer is size 2—representing the predicted heating and cooling energy consumption for that hour. The optimization algorithm used was Adam, which has proven to be a very good performer [45]. The hyperparameters of *learn rate* (step size in direction that minimizes the loss), *batch size* (number of samples per update of model parameters), *number of hidden layers*, and *size of hidden layers* were determined through a grid search (section 2.9).

Of the 17.52 million rows of data, 90 % are used for training and 10 % for validation. A high proportion is given to training, since in these trials, minimizing variance in the training is more important than minimizing the model performance statistic variance (*i.e.,* the validation error variance). The neural network is iteratively fed batches of rows from the training dataset (~15 million rows) and every ~2 million rows, the mean squared error of the ANN is validated, and the slope is calculated over the last five validation losses (i.e., errors). If the slope becomes positive (meaning the validation loss is no longer decreasing), then the training stops. The frequency of the validation checks during training was reduced from ~7 million rows (see section 2.5) to ~2 million rows to greatly decrease training time. This reduction results in the ANN slightly underfitting the training data, since the slope over five more frequently checked validation losses is more likely to be positive due to the stochasticity of the optimization. However, the final combined heating/cooling CVRMSE with checks every ~7 million rows (~19 hours of training) was effectively the same as with ~2 million rows (~5 hours of training time).

2.8 Step Four: Infer Unknown Building Parameters

The final step of the methodology consists of inferring the unknown building parameters, given known hourly controlled/metered energy consumption from the 8760 hours of the year. With the model parameters frozen, the mean squared error between predicted and true energy consumption is computed. Backpropagation is then used to compute the gradient of the error with respect to a randomly initialized set of building parameter inputs. The building parameters are iteratively modified in the direction of the gradient to minimize the error. A batch size of 256 appeared to offer the best compromise between inference loss and run time. A batch size of 8760 would be needed to compute the error over all hours of the year; however, batches of this size made network updates far too slow.

The Python algorithm processes the batches repeatedly until the slope over the last 10 passes is greater than zero (*i.e.*, the loss is no longer decreasing). The algorithm then returns the set of unknown building parameter inputs that best predicts the 8760 hourly heating and cooling consumptions. The weather and schedules data inputs are fixed during the backpropagation since these values are known. Furthermore, the building parameter estimates are clamped so that they remain in the ranges given in Table 5.

2.9 Grid Search

The neural network hyperparameters of learning rate, batch size, number of hidden layers, and size of the hidden layers can have a substantial impact on the final loss (*i.e.,* error) of a trained neural network; therefore, a grid search was implemented that tested how the ANN's final loss is impacted by three different learning rates (5×10^{-5}, 1×10^{-4}, 5×10^{-4}), five different batch sizes (32, 64, 128, 256, 512), eight different number of layers (2, 3, 4, 5, 6, 8, 16, 32), and six different layer sizes (256, 512, 1028, 4096, 8192, 16384). The total number of combinations between these four hyperparameters would be 720, which would take an enormous amount of time to complete; therefore, the hyperparameter-space was sampled manually by choosing some combinations, observing their loss, and testing new combinations until a hyperparameter set was reached that gave the seemingly best compromise between loss and run time. In the end, a total of 29 different hyperparameter sets were tested. The chosen sets include ANN architectures that either have (1) many neurons in each layer but few layers and (2) few neurons in each layer but many layers. It was not possible to test ANNs with many neurons and many layers due to the limited memory capacity of the 8GB RTX2060 Super graphics card. However, the results in Figure 7 show increasing loss as the number of hidden neurons exceed 2×8192.

There is some variability in the validation loss (and run time) when training an ANN on the same set of hyperparameters. We found through repetition of the same learn rates and batch sizes for a particular ANN size that the loss was not particularly dependent on batch sizes between 32 and 512 (we ended up choosing 64), while a learn rate of 1×10^{-4} was a good compromise between 5×10^{-5} and 5×10^{-4}. For the ANN size, 2 hidden layers and 8192 neurons in each layer proved to be the best compromise between loss and run time. This hyperparameter configuration was chosen for the subsequent case studies. Figure 7 shows this chosen configuration in context with other configurations that have a learn rate of 1×10^{-4} and a batch size of 64 (the results for the other learn rates and batch sizes are not shown for brevity)—note that the proportions of the shaded markers in the figure are not to scale and are just to show the relative difference between the structures of the tested networks.

It makes sense that two hidden layers work well, since the building parameters are related to the energy consumption of the building in a largely linear manner. If the transformation from the inputs to the outputs of an ANN were highly complex and non-linear, then we would expect that a larger number of hidden layers would be required to model this relationship.

When comparing the total run time between the RBFOpt method and the surrogate ANN method, the grid search of the ANN hyperparameters (learn rate, batch size, number of hidden layers, and hidden layer size) is not considered, since it represents a second order optimization—an optimization of the optimizer—and only the first order optimization—the optimization of the objective function—is compared.

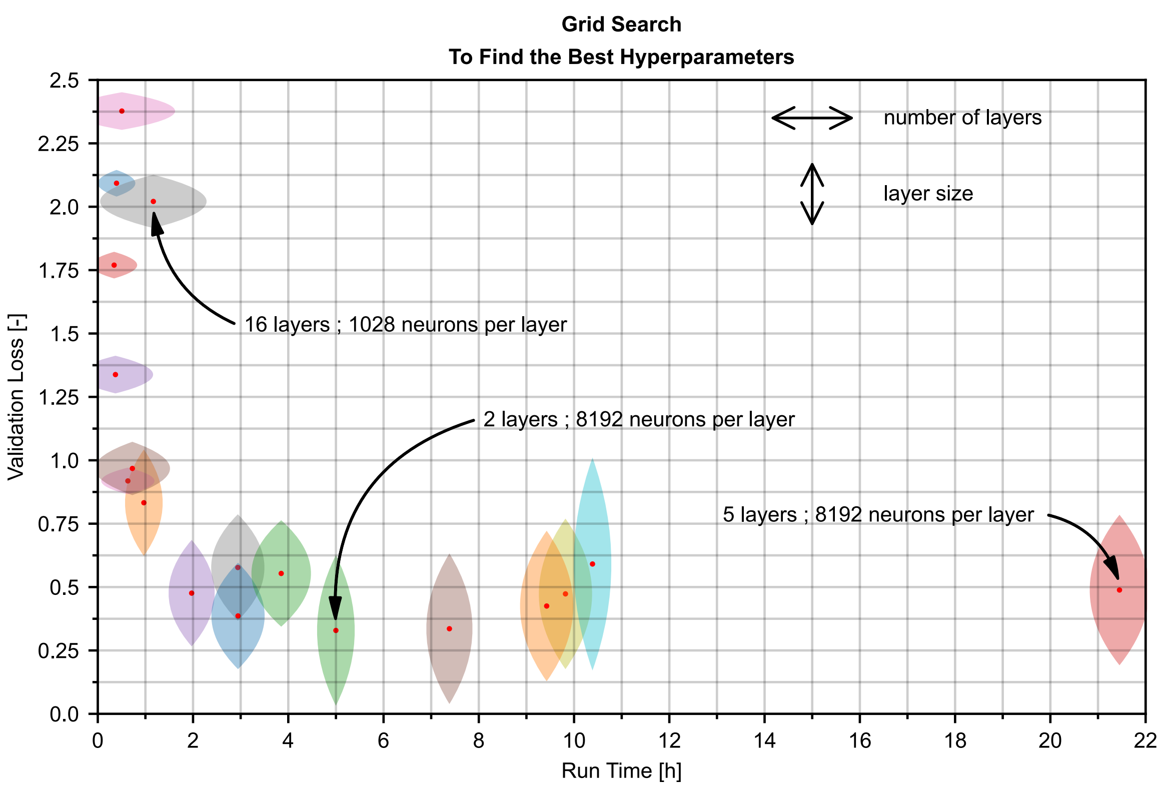

Figure 7: Visualizing the grid search and showing the chosen hyperparameter configuration in context with the other configurations (note that the proportions of the markers in the figure are not to scale and are just to show the relative difference between the structures of the tested networks).

2.10 Comparing the Surrogate ANN Method with RBFOpt, a High-Performing black box Optimizer

As stated in the introduction, the calibration methodology developed in this paper was compared to RBFOpt, a powerful black box optimizer that also builds and iterates a surrogate model of the objective function. The objective function that RBFOpt is tasked with minimizing is the average of the hourly heating and cooling CVRMSE, shown in Equation (3) below.

$$CVRMSE_{comb} = \frac{1}{2} * \left(\frac{\sqrt{\frac{\sum_{i=1}^{8760}\left(heat_{met,i} - heat_{sim,i}\right)^2}{8760}}}{\frac{\sum_{i=1}^{8760} heat_{met,i}}{8760}} + \frac{\sqrt{\frac{\sum_{i=1}^{8760}\left(cool_{met,i} - cool_{sim,i}\right)^2}{8760}}}{\frac{\sum_{i=1}^{8760} cool_{met,i}}{8760}} \right) \tag{3}$$

It is unlikely that either calibration method finds a globally optimal solution on any given run. Furthermore, both RBFOpt and surrogate ANN methods are non-deterministic and will almost always return different solutions from run-to-run. Only comparing a single optimization run between the RBFOpt and surrogate ANN methods would not consider the inherent uncertainty that exists in building energy model calibration. Therefore, five optimization trials were performed for each method, and the optimized values of the unknown building parameters at the end of each trial, along with the final CVRSME, were compared in both the controlled (known values for the parameters) and real (actual metered data) case studies.

The RBFOpt trials consisted of the following steps:

1. Create RBFOpt optimization model.
2. Iterate the solver in parallel (32 threads), where each iterations involves
   - 2.1. creating a new folder to place the new TRNSYS files,
   - 2.2. generating the new TRNSYS files with the unknown building parameter values chosen by the solver,
   - 2.3. running an 8760-hour TRNSYS simulation of the BEM,
   - 2.4. calculating the combined hourly CVRMSE with the results file, and
   - 2.5. updating the solver's objective function with the calculated CVRMSE.
3. When 2000 iterations have elapsed, record best building parameter values to a file.
4. Repeat Steps 1 to 3 for each trial.

(Refer to section 1.6 to understand how RBFOpt's solver works).

The surrogate ANN calibration trials followed the steps below:

1. Run the building energy model 2000 times.
2. Collect the generated energy consumption data along with building parameters, weather, and schedules data.
3. Create, train, and validate an ANN with the chosen hyperparameters from the grid search.
4. Infer the unknown building parameters and record them to a file, and
5. Repeat steps 3 and 4 for each trial.

For the ANN method, the first two steps above—running the building energy model many times and collecting the generated data—only had to be completed a single time. The data could then be reused for all subsequent trials. Note that the building parameter inference step (Step 4) only needed to run once per trial, since it was observed that all inference runs with a given ANN produced identical results.

3 Results and Discussion

3.1 The Impact of Including Variable Inputs in the Neural Network

First, we would like to demonstrate the effect on calibration accuracy when the weather and building schedule variables are included in the surrogate ANN. Two different surrogate ANNs were compared: (1) an ANN with only the 14 unknown building parameters as inputs and (2) the ANN previously discussed with 124 total inputs, covering weather, building schedules, and the unknown building parameters. Another grid search was performed to find the best hyperparameters for the ANN without variables, and we found that a learn rate 1×10^{-4}, a batch size of 64 and one hidden layer with 512 neurons had the lowest loss. The controlled case study was used for the comparison so that we can assess how close the ANNs' building parameter estimates are to the actual values.

Figure 8(a) shows the calibrated parameter values over 5 trials for the ANN without variables (orange pentagons) and with variables (blue cross). When including the weather and building schedule variables in addition to the unknown building parameters, the ANN finds parameter values very close to the actual values (green lines), resulting in a combined heating and cooling CVRMSEs of between 0.5 and 1 % (Figure 8(b)). When only the unknown building parameters are included as inputs, the surrogate ANN performs quite poorly, finding parameter values much further away from the actual values, resulting in combined CVRMSEs between 32 and 33 %. Finally, considering that the 14-input ANN without weather and building schedule variables has a lot less relevant information to process than the full 124-input ANN, it is logical that the computation run times are much shorter.

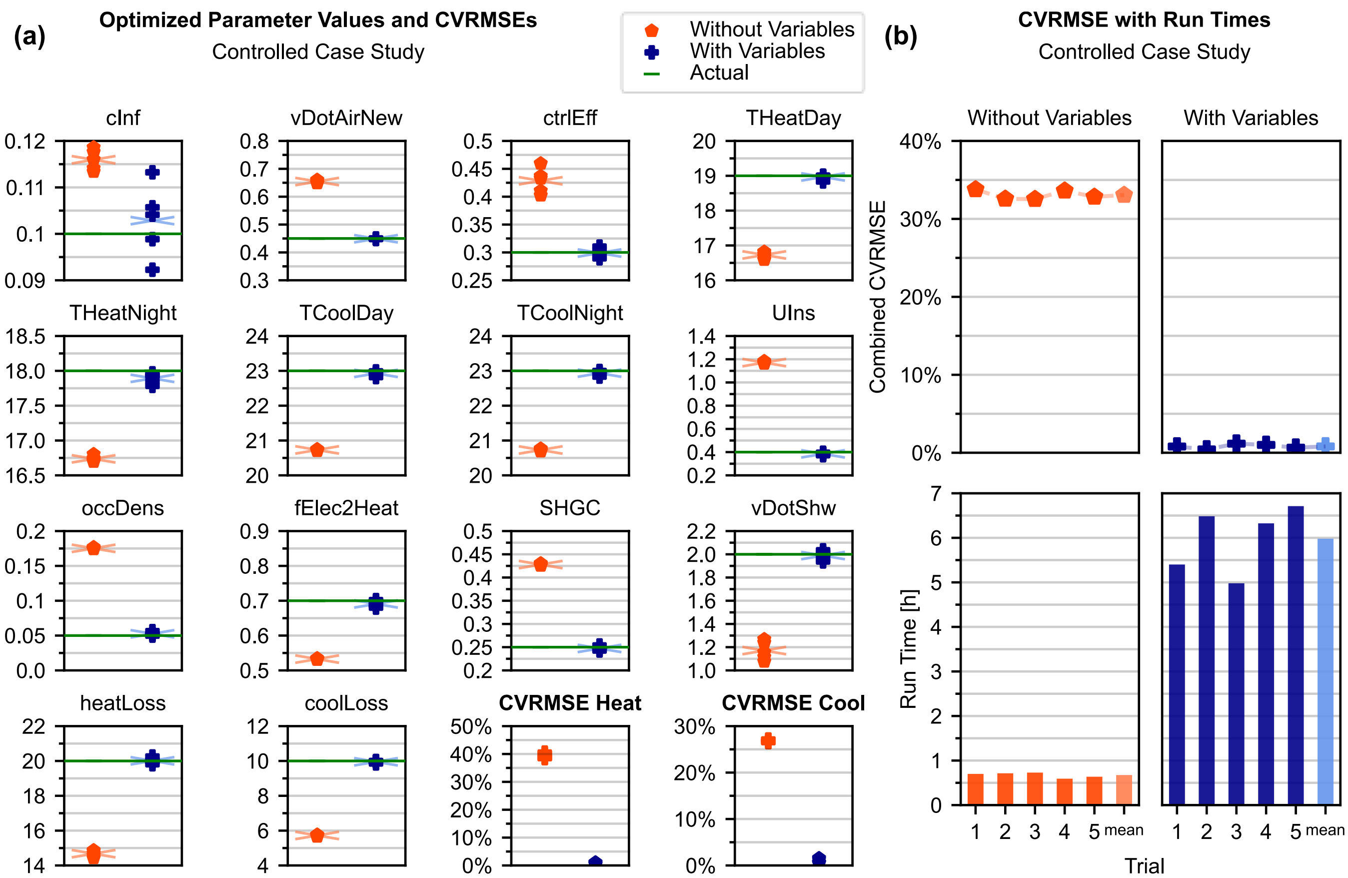

Figure 8: Comparing **(a)** the optimized parameter values (points) with mean (cross) and **(b)** the computation run times between the trials of the surrogate neural without and with variables, in the controlled case study.

3.2 Comparing the RBFOpt Optimizer with the Neural Network on Equal Data Usage

To compare the RBFOpt and the surrogate neural network methods on roughly equal grounds, we decided to compare the methods' performance given equivalent data usage or, in other words, an equivalent number of iterations of the building energy model. We can expect continued speed improvements for numerical optimizers, either through hardware or software, as they are important in many fields, while we can expect slower speed improvements in BEM engineering software as they are more niche and rely on fast CPU cores, which are seeing increasingly smaller speed improvements [46]. Therefore, the computation time associated with simulating law-based BEMs will continue to make up a large portion of the total computation time of automatic BEM calibration methods.

When considering an equivalent number of BEM iterations, the total run times for the ANN trials are higher than for the RBFOpt trials, since the training and inference steps for the ANN is more computationally demanding than the training and inference steps for the RBFOpt model—while the run time associated with iterating the BEM is identical between both methods. However, given a much more powerful GPU (such as an Nvidia Quadro RTX card), the training and inference time for the ANN method would be greatly reduced and the total run times would be more similar. And more importantly, our building model is relatively simple (the HVAC systems, in particular, are quite simple) and only takes about 30 seconds to run on our system (see section 3.3 for computer specifications)—if more complex building models are to be calibrated, the total computational time between the two methods would be closer in this comparison.

3.2.1 Controlled Case Study

Figure 9 below shows the results of the five trials. The surrogate ANN method outperforms the RBFOpt black box optimizer with CVRSMEs consistently around or below 1 %, while the RBFOpt optimizer has CVRMSEs between 3 and 5 % (Figure 9(b)). When observing the spread and accuracy of the optimized "unknown" building parameter values, the surrogate ANN technique found values much closer to the actual parameter values compared to the RBFOpt method (Figure 9(a)). The surrogate ANN method found local minima that are close to each other and, most importantly, that are closer to the global minimum compared to the RBFOpt method—a clear advantage for the ANN method. Through the ~6 hours of training, the ANN has learned the complex map between the energy consumption of the BEM and the unknown building parameters and variable inputs, like the weather. For example, the ANN might account for how the air infiltration through the building envelope and the wind speed interact to contribute to the heating or cooling consumption of the building. The RBFOpt algorithm does not have the wind speed as an input, so it is unable to model this interaction. With this distinct advantage, it is understandable that the building parameter values inferred from the ANN are closer to the true values than those found by RBFOpt.

Averaging the results of the trials does not favor the RBFOpt method over the ANN method. The combined CVRMSE when averaging the optimized parameter values over the five trials is ~0.37 % for the ANN method and ~2.7 % for the RBFOpt method. The CVRMSE from averaging the trials is improved for the RBFOpt method, but the ANN method is still better when considering any single trial, and the method is even better when also averaging over five trials.

Finally, as seen in Figure 9(b), there does not seem to be much correlation between the computation run time and the final CVRMSE value for the surrogate ANN method (nor for the RBFOpt method for that matter). The lack of correlation certainly stems from the inherent uncertainty in BEM calibration, where optimizers can get more-or-less "lucky" from one trial to the next.

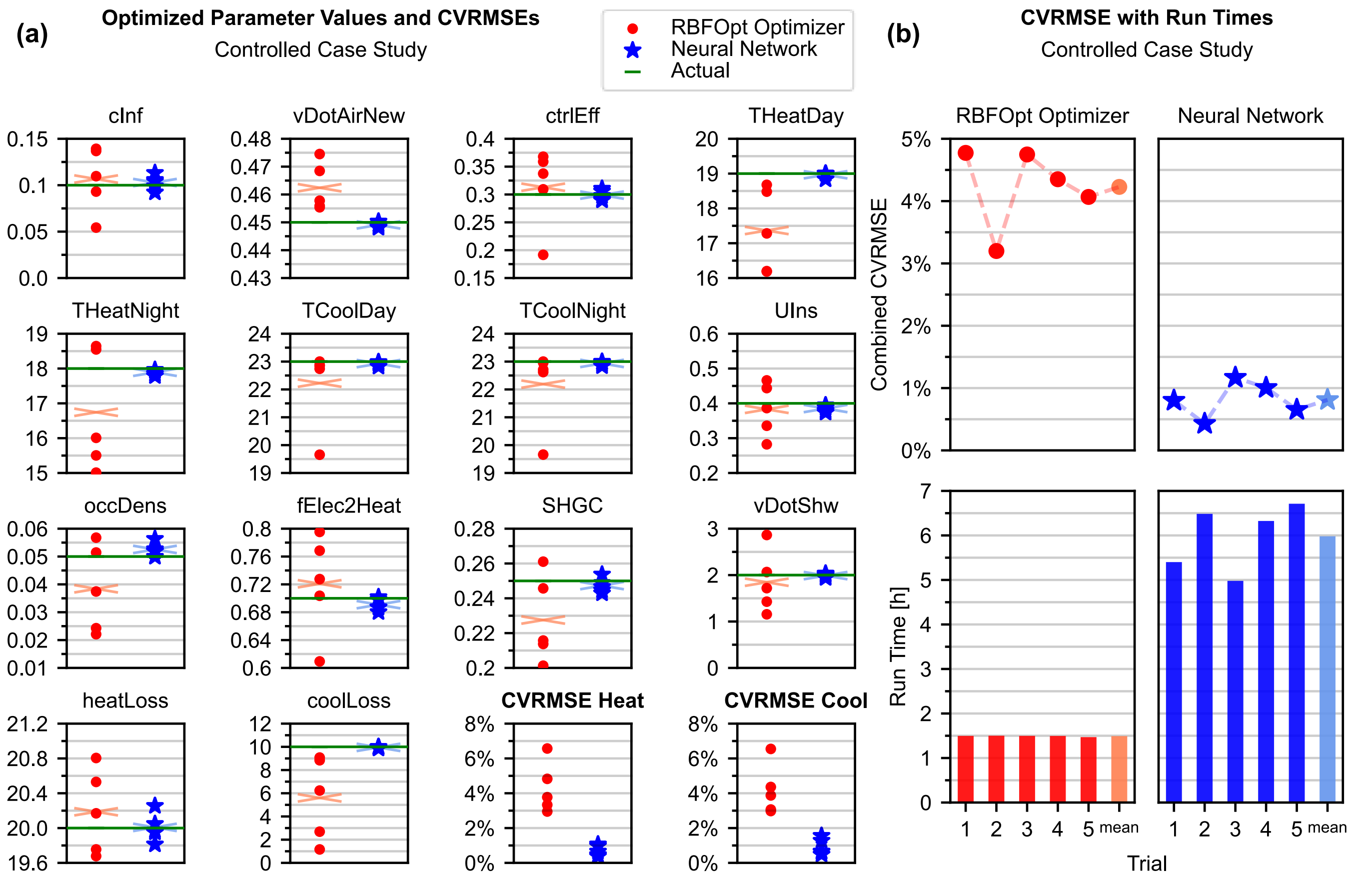

Figure 9: Comparing **(a)** the optimized parameter values (points) with mean (cross) and **(b)** the computation run times between the trials of the RBFOpt optimizer and the surrogate neural network in the controlled case study.

3.2.2 Real Metered Case Study

This case study uses real metered energy data during 2019 at White Hall (the case study building at Cornell University). The ANN method only slightly outperforms the RBFOpt optimizer, with an averaged combined CVRMSE of 22.34 % versus 22.52 % for the RBFOpt method. What is immediately noticeable in Figure 10(a) is that, again, the results of the ANN method have lower variance than those of the RBFOpt method. Yet, since the combined CVRMSEs for all ten trials are within 0.21 percentage points of one another, we are led to conclude that the optimizers were hampered more by the BEM's configuration than their ability to optimize. In other words, the optimizers were ill-constrained by the BEM and were not able to lower the combined CVRMSE any lower than ~22 %. The poorly configured building energy model is reflected in the fact that some optimized parameter values are lower than expected; we posit that the optimizers in both the surrogate ANN and RBFOpt methods lowered these parameter values as much as they could to lower the CVRMSE before getting stuck in a minimum.

We can still venture guesses as to why some of the building parameter values are lower than expected. The solar heat gain coefficient of the windows is optimized to be around 0.1 in both methods, which is quite low. Then again, the window-to-wall ratio is quite high, and there are blinds behind all the windows, so it is possible that the blinds were mostly closed in the building throughout the year. The four setpoint temperatures also appear to be low. Perhaps since the setpoint temperatures in the building are room-specific and occupancy-controlled, the equivalent mean temperature setpoint in the whole building is lower than the individual room setpoint temperatures due to lower occupancy. With respect to occupancy, the maximum occupancy density is nearly zero in both methods, which is not realistic since there are indeed people in the building. It is well known that occupant behavior is very difficult to measure accurately and is one of the chief reasons for the discrepancy between simulated and measured data in BEM calibration [47]. The metered electricity consumption was used to inform the simplified occupancy schedule used in the simulations; however, it is most certainly still a gross estimation of the actual occupancy schedule, thus the optimizers deemed that the occupancy in the building was actually increasing the BEM's error and decided to zero-out the occupancy density parameter. Lastly, the optimized fElec2Heat value (the fraction of electricity that is transferred to internal heat gains) is also lower than expected, with values between 0.2 and 0.4; these low values could be explained by (1) high electricity consumption from equipment and/or lights located outside the building or (2) the majority of heat gains from equipment and lights inside the building being ejected outside the building through the ventilation system or through exfiltration. Despite some of these unexpected building parameter values, the mean combined CVRMSEs after 5 trials for both the surrogate ANN and RBFOpt methods are below the ASHRAE Guideline 14 recommendations of 30 % [6].

These unrealistic parameter values probably cannot be used in the context of a building retrofit or energy conservation measure (ECM) study. Since these unrealistic parameter values very likely stem from a poorly configured (*i.e.*, overly simplified) building energy model, the BEM itself would have to be remodeled in a detailed way, following extensive building audits, and then calibrated again in order to do a proper ECM study.

To conclude, when considering an equivalent number of BEM iterations, the performance is almost identical between both methods; however, the ANN method's results are more consistent, giving it a clear advantage. It appears that only a single calibration trial is required for the ANN method to achieve optimal performance, while the RBFOpt method requires several trials. Finally, just like in the controlled case study, there does not seem to be a correlation between the ANN or RBFOpt trial times and the final CVRMSE values.

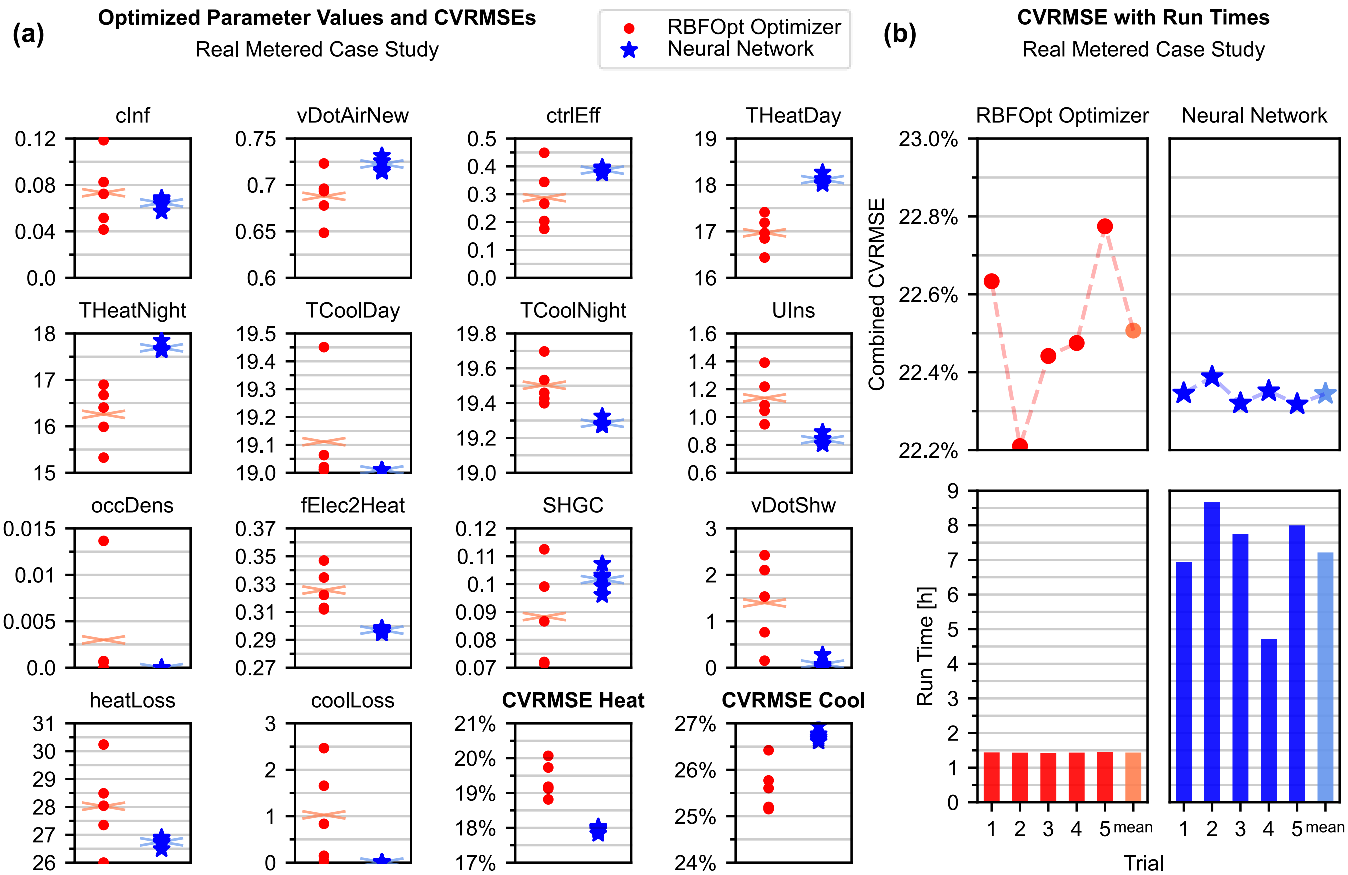

Figure 10: Comparing **(a)** the optimized parameter values (points) with mean (cross) and **(b)** the computation run times between the RBFOpt optimizer and the surrogate neural network trials in the real metered case study.

3.3 Comparing Methods with Roughly Equal Computation Time

Another set of five trials for both the RBFOpt method and the ANN method were performed, but this time, we tried to match the computation time of each trial. It must be stated that comparing computation run time between the RBFOpt and the surrogate ANN methods is not an apples-to-apples comparison, since the neural network is trained and validated on a GPU with the CPU providing auxiliary calculations, while RBFOpt relies exclusively on the CPU. The CPU used is the 16-core AMD Ryzen 9 5950X (retail price of US$ 799 at launch on November 5, 2020), while the GPU used is the Nvidia RTX2060 Super with 2176 CUDA Cores (retail price of US$ 399 at launch on July 9, 2019). Finally, 128GB of RAM at 3200 MT/s (colloquially known as Mhz) was available. Comparing CPU performance to GPU performance is difficult, since a CPU has relatively few processing cores, which are very fast, while the GPU has many more processing cores, which are slower.

To roughly match the computation time, 10,000 iterations of the BEM were executed for the RBFOpt method trials (up from 2000), while 2000 iterations were executed again for the ANN method trials. In the controlled case study, the average computation time ended up being ~7.3 hours for the RBFOpt trials and ~6.8 hours for the ANN trials, while in the real metered case study, the average computation time was ~7.4 hours and ~6.6 hours, respectively. The results of this comparison are remarkably similar to the comparison of equivalent data usage (*i.e.*, equivalent number of BEM iterations). For the controlled case study, the averaged combined CVRMSE for the RBFOpt method only improves by about 0.5 percentage points when running 10,000 iterations instead of 2000, and for the real metered case study, the improvement is only about 0.2 percentage points. For both case studies, the average combined CVRMSE is nearly identical for the ANN method compared to the previous comparison, which is logical given that the same number of BEM iterations (2000) are executed in both comparisons and that the variance of the ANN results is low.

3.4 Limitations of the Study

There are two primary limitations to the study: (1) the BEM does not appear to be very well-calibrated in the real metered case study, despite an hourly CVRSME well below the recommended ASHRAE guidelines; and (2) the hyperparameters chosen thanks to the grid search might not be generalizable to other building energy models.

In the real metered case study, some of the optimized values of the unknown building parameters are unexpectedly low in both the surrogate neural network and RBFOpt methods, namely the solar heat gain coefficient of the windows and the setpoint temperatures. Given that the optimized values between both methods are relatively close, we theorize that both methods were more hampered by the quality of the building energy model or metered data than their ability to optimize. The optimizers in both the surrogate ANN and RBFOpt methods lowered these parameter values as much as they could to lower the CVRMSE before getting stuck in a minimum. It is important to note that a lot of time and effort was placed on improving the BEM by testing different HVAC systems and controls as well as the inclusion or exclusion of different unknown building parameters. However, we would have had to perform extensive and detailed building audits to create a building energy model with more complex and accurate HVAC systems and controls, which would somewhat defeat the purpose of automated calibration. Yet, despite the unexpectedly low calibrated parameter values, the final hourly CVRMSE is around 22 % in our chosen configuration—well below the recommended 30 % by ASHRAE Guideline 14 (see Table 1)—which leads us to question the usefulness of these widely-accepted calibration thresholds.

The neural network hyperparameters that were chosen after a grid search are a learn rate of 1×10^{-4}, a batch size of 64, 2 hidden layers, and 8192 neurons per hidden layer. We do not know if these hyperparameters are generalizable to other building energy models, especially BEMs with more unknown building parameters, more schedules, or more complex HVAC systems. One could argue that the grid search should be included in the computation run times of the surrogate ANN method trials, since the hyperparameters greatly impact the calibration performance on the ANN. However, we theorize that the same hyperparameters could be reusable for other BEMs, since the relationship between building parameters, weather, and energy consumption does not differ dramatically from building to building.

4 Conclusion

A novel method was developed that uses an artificial neural network (ANN) and surrogate modeling (*i.e.,* metamodeling) to calibrate building energy models (BEMs). Typically, surrogate models have only been used in BEM calibration to speed-up the evaluations performed by an external optimizer, such as a genetic algorithm. In our method, the surrogate neural network model is trained and then used itself to calibrate the unknown building parameters through gradient descent. The surrogate model in our study considers a much larger breadth of relevant data to predict the output of the BEM than is typical included. Surrogate models in the literature only include the unknown building parameters as inputs, but our surrogate model also includes relevant variable data, namely the weather and building schedules. Our surrogate model can better predict the energy output of the BEM since it can account for how variables like outside temperature affect energy consumption, and the interactive effects between these variables with the fixed features of the building. Other surrogate models, such as RBFOpt, are blind to the effect of these variables.

The novel surrogate neural network BEM calibration method was compared to a powerful black box optimizer called RBFOpt, which has proven to be the fastest algorithm in a benchmarking study of various building simulation tasks [41]. RBFOpt also uses surrogate modeling as its optimization technique, making the comparison more apples-to-apples. To evaluate the two methods, five trials were performed in both a controlled case study and a real metered case study. In the controlled case study, the "unknown" building parameter values were in fact preselected and inputted into the case study BEM, and the resulting simulated energy consumption of this BEM configuration was used as the "true" energy consumption; the calibration methods would then attempt to match the BEM's energy profile to this true energy profile. This controlled case study allowed us to determine the accuracy of the calibration methods in optimizing the building parameter values. In the real metered case study, the true energy consumption was the hourly heating and cooling consumption of White Hall, an old building on the Cornell University campus in New York State, during the 2019 calendar year. This real metered case study allowed us to determine the speed and robustness of the two calibration methods given real building performance data.

The ANN method outperforms the RBFOpt optimizer when calibrating the "unknown" building parameter values in the controlled case study, finding local minima much closer to the global minimum. The ANN method consistently found combined heating and cooling hourly CVRMSEs between 0.5 – 1 % in all 5 trials, while the RBFOpt optimizer found CVRMEs between 3 – 5 % throughout its 5 trials.

In the real metered case study, the ANN method consistently finds a low CVRMSE, while the CVRMSEs found by the RBFOpt method fluctuate more. The calibration performance is still similar between both methods with hourly CVRMSEs between 22.2 – 22.8 % across all 10 trials; however, the consistency of the ANN method is a clear advantage because both methods are non-deterministic and for most traditional calibration methods, the calibration level of the BEM depends on the number of trials performed. We can assume based on the results of this article that only a single calibration trial needs to be performed with the ANN method, while multiple runs are needed for the RBFOpt method to ensure a consistent calibration level.

Although the final hourly CVRMSE after running either calibration method is below 30 % (the widely accepted threshold) in the real metered case study, not all calibrated building parameter values appear to be realistic. Notably, the solar heat gain coefficient of the windows, maximum occupancy density, and setpoint temperatures are lower than expected. This discrepancy most likely stems from a poorly configured building energy model, which indicates that more effort is required to match the model's HVAC systems and controls to the actual building. However, this modeling step inherently requires extensive audits of the building, which in a way, defeats the purpose of automated BEM calibration. In conclusion, with a simulated data case study where the BEM is essentially a perfect model of the building (*i.e.*, no modeling error), our surrogate ANN method outperforms the RBFOpt optimizer. This result suggests that the gap in performance in the real-world case study is about errors in the in the modeling step, not in the calibration method.

5 Future Research Directions

We invite the building calibration community to develop methods that mix modeling and calibration. Currently, to calibrate well, automated calibration methods require complex models that are manually configured to closely match the real building. Going forward, automated calibration should extend beyond tweaking continuous parameter values in a fixed model configuration. It must have the ability to introduce various model configurations that the optimizer can select from or even create. In addition, more importance should be placed on the values of the calibrated building parameters than the final calibration criteria, since a low CVRMSE or MBE does not necessarily imply that reasonable building parameter values have been found. We need better criteria for evaluating the "goodness" of a set of building parameters.

6 Declaration of Competing Interests

We declare that we have no conflicts of interest; this work is not inappropriately influenced (biased) by financial and/or personal relationships with other people or organizations.

7 Acknowledgements

We would like to thank Dr. Timur Dogan from Cornell University for the inception of the study and facilitating access to the detailed measured data of White Hall. We would also like to thank the National Science and Engineering Research Council (NSERC) of Canada, the Arbour Foundation, the Fondation et alumni de Polytechnique Montréal, and Hydro-Québec for providing funding in the form of scholarships to the first author.

8 References

[1] World Resources Institute, "World Greenhouse Gas Emissions: 2019," World Resources Institute. Accessed: Jan. 11, 2023. [Online]. Available: https://www.wri.org/data/world-greenhouse-gas-emissions-2019

[2] R. E. Bellman, *Dynamic programming*. Princeton, N.J., USA: Princeton University Press, 1957.

[3] R. Troncoso, "A hybrid monitoring-modeling procedure for analyzing the performance of large central chilling plants," in *Proceedings of Building Simulation*, 1997, pp. 421–428.

[4] W. L. Carroll and R. J. Hitchcock, "Tuning simulated building descriptions to match actual utility data: methods and implementation," *ASHRAE Trans.-Am. Soc. Heat. Refrig. Airconditioning Engin*, vol. 99, no. 2, pp. 928–934, 1993.

[5] D. Coakley, P. Raftery, and M. Keane, "A review of methods to match building energy simulation models to measured data," *Renew. Sustain. Energy Rev.*, vol. 37, pp. 123–141, Sep. 2014, doi: 10.1016/j.rser.2014.05.007.

[6] ASHRAE, *Guideline 14-2014, Measurement of Energy and Demand Savings*. Atlanta, GA, USA: American Society of Heating, Refrigerating and Air-conditioning Engineers, 2014.

[7] U.S. Department of Energy, "M&V Guidelines: Measurement and Verification for Performance-Based Contracts Version 4.0," Nov. 2015.

[8] J. Cowan, "International Performance Measurement and Verification Protocol: Concepts and Options for Determining Energy and Water Savings - Vol 1," U.S. Department of Energy, Washington, DC, USA, Mar. 2002.

[9] T. A. Reddy, I. Maor, S. Jian, and C. Panjapornporn, "Procedures for Reconciling Computer-Calculated Results with Measured Energy Data," American Society of Heating, Refrigerating and Air-Conditioning Engineers, Atlanta, GA, USA, Technical Report, 2006.

[10] T. E. Bou-Saada and J. S. Haberl, “An improved procedure for developing calibrated hourly simulation models,” in *Proceedings of the 5th International IBPSA Conference*, Montréal, Canada, 1995.

[11] R. Evins, “A review of computational optimisation methods applied to sustainable building design,” *Renew. Sustain. Energy Rev.*, vol. 22, pp. 230–245, Jun. 2013, doi: 10.1016/j.rser.2013.02.004.

[12] V. Machairas, A. Tsangrassoulis, and K. Axarli, “Algorithms for optimization of building design: A review,” *Renew. Sustain. Energy Rev.*, vol. 31, pp. 101–112, Mar. 2014, doi: 10.1016/j.rser.2013.11.036.

[13] T. Østergård, S. E. Maagaard, R. L. Jensen, and C. E. MOE, “Thermal comfort in residential buildings by the millions-early design support from stochastic simulations,” in *12th REHVA World Congress CLIMA*, 2016, p. 11. Accessed: Oct. 12, 2022. [Online]. Available: https://www.researchgate.net/profile/Torben-Ostergard-2/publication/303522926_Thermal_Comfort_in_Residential_Buildings_by_the_Millions_-Early_Design_Support_from_Stochastic_Simulations/links/5746b41908ae9f741b431dc0/Thermal-Comfort-in-Residential-Buildings-by-the-Millions-Early-Design-Support-from-Stochastic-Simulations.pdf

[14] T. Østergård, R. L. Jensen, and S. E. Maagaard, “A comparison of six metamodeling techniques applied to building performance simulations,” *Appl. Energy*, vol. 211, pp. 89–103, Feb. 2018, doi: 10.1016/j.apenergy.2017.10.102.

[15] E. Naboni, Y. Zhang, A. Maccarini, E. Hirsch, and D. Lezzi, “Extending the use of parametric simulation in practice through a cloud based online service,” in *IBPSA Italy-conference of international building performance simulation association, Bozen, Italy*, 2013. Accessed: Oct. 12, 2022. [Online]. Available: https://www.researchgate.net/profile/Alessandro-Maccarini/publication/259087411_Extending_the_use_of_parametric_simulation_in_practice_trough_a_cloud_based_online_service/links/54b7cdf50cf28faced607e10/Extending-the-use-of-parametric-simulation-in-practice-trough-a-cloud-based-online-service.pdf

[16] J. Yang, “Convergence and uncertainty analyses in Monte-Carlo based sensitivity analysis,” *Environ. Model. Softw.*, vol. 26, no. 4, pp. 444–457, Apr. 2011, doi: 10.1016/j.envsoft.2010.10.007.

[17] S. Asadi, E. Mostavi, D. Boussaa, and M. Indraganti, “Building energy model calibration using automated optimization-based algorithm,” *Energy Build.*, vol. 198, pp. 106–114, Sep. 2019, doi: 10.1016/J.ENBUILD.2019.06.001.

[18] J. Sun and T. A. Reddy, “Calibration of Building Energy Simulation Programs Using the Analytic Optimization Approach (RP-1051),” *HVACR Res.*, vol. 12, no. 1, pp. 177–196, Jan. 2006, doi: 10.1080/10789669.2006.10391173.

[19] A. Chong, Y. Gu, and H. Jia, “Calibrating building energy simulation models: A review of the basics to guide future work,” *Energy Build.*, vol. 253, p. 111533, Dec. 2021, doi: 10.1016/j.enbuild.2021.111533.

[20] B. Eisenhower, Z. O’Neill, V. A. Fonoberov, and I. Mezić, “Uncertainty and sensitivity decomposition of building energy models,” *J. Build. Perform. Simul.*, vol. 5, no. 3, pp. 171–184, May 2012, doi: 10.1080/19401493.2010.549964.

[21] M. Manfren, N. Aste, and R. Moshksar, “Calibration and uncertainty analysis for computer models – A meta-model based approach for integrated building energy simulation,” *Appl. Energy*, vol. 103, pp. 627–641, Mar. 2013, doi: 10.1016/j.apenergy.2012.10.031.

[22] A. T. Booth, R. Choudhary, and D. J. Spiegelhalter, “A hierarchical Bayesian framework for calibrating micro-level models with macro-level data,” *J. Build. Perform. Simul.*, vol. 6, no. 4, pp. 293–318, Jul. 2013, doi: 10.1080/19401493.2012.723750.

[23] S. Wang, X. Sun, and U. Lall, “A hierarchical Bayesian regression model for predicting summer residential electricity demand across the U.S.A.,” *Energy*, vol. 140, pp. 601–611, Dec. 2017, doi: 10.1016/j.energy.2017.08.076.

[24] Y.-J. Kim, K.-U. Ahn, and C.-S. Park, “Decision making of HVAC system using Bayesian Markov chain Monte Carlo method,” *Energy Build.*, vol. 72, pp. 112–121, Apr. 2014, doi: 10.1016/j.enbuild.2013.12.039.

[25] M. H. Kristensen, R. E. Hedegaard, and S. Petersen, “Hierarchical calibration of archetypes for urban building energy modeling,” *Energy Build.*, vol. 175, pp. 219–234, Sep. 2018, doi: 10.1016/j.enbuild.2018.07.030.

[26] J. Yuan, V. Nian, B. Su, and Q. Meng, “A simultaneous calibration and parameter ranking method for building energy models,” *Appl. Energy*, vol. 206, pp. 657–666, Nov. 2017, doi: 10.1016/j.apenergy.2017.08.220.

[27] J. Z. Kolter and M. J. Johnson, “REDD: A public data set for energy disaggregation research,” in *Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*, Citeseer, 2011, pp. 59–62. Accessed: Nov. 09, 2020. [Online].

Available:
https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=d85a51e2978f4563ee74bf9a09d3219e03799819

[28] S. Nagpal, C. Mueller, A. Aijazi, and C. F. Reinhart, “A methodology for auto-calibrating urban building energy models using surrogate modeling techniques,” *J. Build. Perform. Simul.*, vol. 12, no. 1, pp. 1–16, Jan. 2019, doi: 10.1080/19401493.2018.1457722.

[29] J. Chen, X. Gao, Y. Hu, Z. Zeng, and Y. Liu, “A meta-model-based optimization approach for fast and reliable calibration of building energy models,” *Energy*, vol. 188, p. 116046, Dec. 2019, doi: 10.1016/j.energy.2019.116046.

[30] K. Cant and R. Evins, “Improved calibration of building models using approximate Bayesian calibration and neural networks,” *J. Build. Perform. Simul.*, vol. 0, no. 0, pp. 1–17, Oct. 2022, doi: 10.1080/19401493.2022.2137236.

[31] M. J. D. Powell, “Variable Metric Methods for Constrained Optimization,” in *Mathematical Programming The State of the Art: Bonn 1982*, A. Bachem, B. Korte, and M. Grötschel, Eds., Berlin, Heidelberg: Springer, 1983, pp. 288–311. doi: 10.1007/978-3-642-68874-4_12.

[32] H. Lim and Z. J. Zhai, “Comprehensive evaluation of the influence of meta-models on Bayesian calibration,” *Energy Build.*, vol. 155, pp. 66–75, Nov. 2017, doi: 10.1016/j.enbuild.2017.09.009.

[33] M. Riddle and R. T. Muehleisen, “A guide to Bayesian calibration of building energy models,” in *Building simulation conference*, 2014.

[34] E. Asadi, M. G. da Silva, C. H. Antunes, L. Dias, and L. Glicksman, “Multi-objective optimization for building retrofit: A model using genetic algorithm and artificial neural network and an application,” *Energy Build.*, vol. 81, pp. 444–456, Oct. 2014, doi: 10.1016/j.enbuild.2014.06.009.

[35] S. L. Wong, K. K. W. Wan, and T. N. T. Lam, “Artificial neural networks for energy analysis of office buildings with daylighting,” *Appl. Energy*, vol. 87, no. 2, pp. 551–557, Feb. 2010, doi: 10.1016/j.apenergy.2009.06.028.

[36] L. Magnier and F. Haghighat, “Multiobjective optimization of building design using TRNSYS simulations, genetic algorithm, and Artificial Neural Network,” *Build. Environ.*, vol. 45, no. 3, pp. 739–746, Mar. 2010, doi: 10.1016/j.buildenv.2009.08.016.

[37] E. E. Aydın, O. Dursun, I. Chatzikonstantinou, and B. Ekici, “Optimisation of energy consumption and daylighting using building performance surrogate model,” p. 11.

[38] B. Eisenhower, Z. O’Neill, S. Narayanan, V. A. Fonoberov, and I. Mezić, “A methodology for meta-model based optimization in building energy models,” *Energy Build.*, vol. 47, pp. 292–301, Apr. 2012, doi: 10.1016/j.enbuild.2011.12.001.

[39] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Math. Program.*, vol. 106, no. 1, pp. 25–57, Mar. 2006, doi: 10.1007/s10107-004-0559-y.

[40] H. M. Gutmann, “A Radial Basis Function Method for Global Optimization,” *J. Glob. Optim.*, vol. 19, pp. 201–227, 2001, doi: https://doi.org/10.1023/A:1011255519438.

[41] T. Wortmann, “Genetic evolution vs. function approximation: Benchmarking algorithms for architectural design optimization,” *J. Comput. Des. Eng.*, vol. 6, no. 3, pp. 414–428, Jul. 2019, doi: 10.1016/j.jcde.2018.09.001.

[42] Canadian Commission on Building and Fire Codes, “National Energy Code of Canada for Buildings: 2020,” National Research Council of Canada, 978-0-660-37921–0, Mar. 2022. doi: 10.4224/rr3q-hm83.

[43] J. Huang, “White Box Technologies Weather Data.” Accessed: Dec. 22, 2022. [Online]. Available: http://weather.whiteboxtechnologies.com/

[44] D. R. Jones, M. Schonlau, and W. J. Welch, “Efficient Global Optimization of Expensive Black-Box Functions,” *J. Glob. Optim.*, vol. 13, no. 4, pp. 455–492, Dec. 1998, doi: 10.1023/A:1008306431147.

[45] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” presented at the 3rd International Conference for Learning Representations, San Diego, CA, USA, 2015. doi: https://doi.org/10.48550/arXiv.1412.6980.

[46] T. N. Theis and H.-S. P. Wong, “The End of Moore’s Law: A New Beginning for Information Technology,” *Comput. Sci. Eng.*, vol. 19, no. 2, pp. 41–50, 2017, doi: 10.1109/MCSE.2017.29.

[47] E. Azar and C. C. Menassa, “A comprehensive analysis of the impact of occupancy parameters in energy simulation of office buildings,” *Energy Build.*, vol. 55, pp. 841–853, Dec. 2012, doi: 10.1016/j.enbuild.2012.10.002.