MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Calibrating building simulation models using multi-source datasets and meta-learned Bayesian optimization

Zhan, Sicheng; Wichern, Gordon; Laughman, Christopher R.; Chong, Adrian; Chakrabarty, Ankush

TR2022-072 July 20, 2022

Abstract

Reliable building simulation models are key to optimizing building performance and reducing greenhouse gas emissions. Informed decision making requires simulation models to be accurate, extrapolatable, and interpretable, all of which require calibrating model simulations to ground truth. Complicated building dynamics and highly uncertain exogenous disturbances make the model calibration process challenging and expensive; hence, a scalable and efficient calibration approach is needed to enable actual application. Current automatic calibration algorithms do not leverage data collected from multiple sources: for example, data obtained from previous calibration tasks on other buildings. In this paper, we employ probabilistic deep learning to meta-learn a distribution using multi-source data acquired during previous calibration. Subsequently, the meta-learned Bayesian optimizer accelerates calibration of new, unseen tasks. The few-shot (that is, requiring few model simulations) nature of the proposed algorithm is demonstrated on a Modelica library of residential buildings validated by the United States Department of Energy (USDoE). The proposed algorithm is compared against classical Bayesian optimization-based calibration, and it is shown that ANP significantly sped up the calibration procedure: the optimal model parameters are identified with 40-60% less simulations compared to the baseline.

Energy and Buildings 2022

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Calibrating building simulation models using multi-source datasets and meta-learned Bayesian optimization

Sicheng Zhan[a,2], Gordon Wichern[b], Christopher Laughman[b], Adrian Chong[a], Ankush Chakrabarty[b,1]

[a] *Department of the Built Environment, National University of Singapore, Singapore.*
[b] *Mitsubishi Electric Research Laboratories, Cambridge, MA, United States*

Abstract

Reliable building simulation models are key to optimizing building performance and reducing greenhouse gas emissions. Informed decision making requires simulation models to be accurate, extrapolatable, and interpretable, all of which require calibrating model simulations to ground truth. Complicated building dynamics and highly uncertain exogenous disturbances make the model calibration process challenging and expensive; hence, a scalable and efficient calibration approach is needed to enable actual application. Current automatic calibration algorithms do not leverage data collected from multiple sources: for example, data obtained from previous calibration tasks on other buildings. In this paper, we employ probabilistic deep learning to meta-learn a distribution using multi-source data acquired during previous calibration. Subsequently, the meta-learned Bayesian optimizer accelerates calibration of new, unseen tasks. The few-shot (that is, requiring few model simulations) nature of the proposed algorithm is demonstrated on a Modelica library of residential buildings validated by the United States Department of Energy (USDoE). The proposed algorithm is compared against classical Bayesian optimization-based calibration, and it is shown that ANP significantly sped up the calibration procedure: the optimal model parameters are identified with 40-60% less simulations compared to the baseline.

Keywords: Meta learning, Deep learning, Parameter estimation, Probabilistic machine learning, Bayesian methods, Digital twin

1. Introduction

Efforts to reduce societal energy consumption and mitigate drivers of climate change have continued to focus on improving building energy efficiency because of their significant energy consumption; in the United States, the building sector alone covers around 40% of total national energy consumption and almost 20% of greenhouse gas emissions [1]. To address this growing concern, strict guidelines for constructing new buildings have been proposed both at the government[3] and scientific level. In fact, ASHRAE and the United States Department of Energy (US DOE) have identified that building energy modeling (BEM) is a key technology for enabling energy-efficiency in buildings [2]. These BEMs are typically designed to predict the energy dynamics of buildings during various modes of operation via numerical simulations. Since numerical simulations are cheaper than performing experiments, one can use these models to quickly ascertain how the building system behaves under different heating/cooling equipment, uncertainties due to weather conditions or occupants, and retrofitting; we refer to [3] for a detailed discussion on the benefits of BEMs. Consequently, BEMs play a major role in the design of advanced control and estimation algorithms for regulating building thermal dynamics despite load variations [4, 5], in equipping building digital twins with monitoring capabilities [6, 7], and in self-optimization for energy reduction [8, 9] and personal conditioning [10], to name a few applications.

To reiterate, simulation BEMs are essential to curtailing energy expenditure, but for simulation-based optimization to be effective, one must ensure that the simulation model accurately reflects reality. Simulation models can range from white-box (where the models are constructed based on a physical understanding of building dynamics) to black-box (where the models are constructed using function approximators like neural networks directly from data) and various hybridizations of these categories. Regardless of the modeling framework, these simulation models contain a large number of model parameters that need to be chosen so that the simulated outputs are accurate with respect to real building measurements [11]. These model parameters are typically representative of climates outside the building over multiple time and area scales, material properties of the building envelope, and attempts to model uncertainties such as equipment degradation, power grid fluctuations, and occupant behavior. In

[1] Email: chakrabarty@merl.com. Phone: (+1) 617-758-6175.

[2] This research was completed during S. Zhan's internship at MERL.

[3] See, for example, California guidelines on net-zero energy by 2030: https://www.cpuc.ca.gov/zne/

fact, a recent study showed that it is not uncommon for a typical building to contain over 3000 parameters to be tuned [12], of which hundreds may require automatic tuning rather than tuning manually with expert knowledge.

Roughly, automatic calibration is a non-user-driven and mathematical process in which an objective function is designed to represent the discrepancy between measured and simulated outputs from a simulation model and this objective is optimized systematically and without manual intervention via numerical methods. With recent availability of efficient numerical solvers and advanced computing resources, automatic calibration can often drastically outperform manual calibration performance both in terms of convergence rates and calibration frequency [13]. Automatic calibration methods have been generally classified into: deterministic or Bayesian, where deterministic automatic calibration has been reported to result in calibrated parameters that are far-removed from their true values, since these methods do not take uncertainty into account [3]. Conversely, Bayesian algorithms such as Markov chain Monte Carlo (MCMC) methods provide a suite of tools for estimating simulation model parameters while concurrently quantifying different uncertainty sources [14]. Concretely, Bayesian calibration attempts to fit a probability distribution on the parameters of the simulation model that best explains the observed data: unfortunately, constructing a probability distribution often requires a large number of model simulations, although advanced Monte-carlo methods have been proposed recently to reduce the simulation overhead such as Hamiltonian Monte-Carlo (HMC) and No U-Turn sampling (NUTS) [15]. Since building simulation models are often slow to simulate due to multi-scale dynamics and noise, a widely adopted approach is to employ simplified (often, oversimplified) meta-models of the building dynamics for quick simulation [16–18]. Rather than using meta-models that may compromise prediction quality, a combination of deterministic and Bayesian approaches has been proposed recently [19], wherein the idea is to use Bayesian optimization to search for optimal parameter combinations with few simulations of the high-quality building simulation model, and modeling the uncertainty associated with the building directly in the calibration objective function. This completely avoids the need to construct a meta-model from the building parameters to the building dynamics, replacing it with a much simpler map from the building parameters to the building calibration cost.

Despite these advances in building model calibration, an open research question is how to extract high-quality information from multi-source building data, that is, data collected on sensors from multiple buildings over the country, and indeed, globally. This is an imminent opportunity made possible due to the advancement of sensor technology and the emergence of connected sensing in buildings, often referred to as the building internet-of-things (IoT) [20]. In this paper, we take a step towards answering the question of how to systematically parse these big multi-source building datasets and extract information for fast and efficient model calibration. The value of multi-source (buildings with different geometries, constructions, locations, etc.) data for developing black-box energy prediction or forecasting models has been recognized. Transfer learning was integrated with neural networks to leverage the knowledge gained from similar buildings for energy prediction [21, 22]. Meta learning was used to recommend the most suitable machine learning model [23, 24]. However, current calibration methodologies only used data obtained from the target building considering its uniqueness [25]. Meanwhile, each optimization-based or sampling-based building calibration task produces a dataset of building parameter to objective function values. These multi-source datasets are often archived but not used to calibrate simulation models for a new building. Ignoring these highly-relevant, often abundant, archived datasets and performing calibration 'from scratch' for each new building presents a missed opportunity. Meta-learning attempts to mimic a human's 'learning to learn' process by training deep learners to estimate distributions of calibration-relevant quantities from previously seen calibration tasks to improve the calibration performance of new tasks [26]. It has been applied in many scenarios where it is extremely slow to estimate parameters from scratch, such as hyper-parameter optimization of deep networks [27]. In the case of calibrating building simulation models, a meta-learning based method has the potential to distill the knowledge of general building physics from the dataset generated from multiple unique buildings.

We demonstrate, for the first time, that data obtained during calibration of related, albeit non-identical, buildings contains useful information about general building dynamics that can significantly accelerate the model calibration of new, unseen target buildings by the use of meta-learning via a class of deep probabilistic neural networks called attentive neural processes [28]. Our **major contributions** in this work include: (i) we propose an automated calibration framework that integrates probabilistic meta-learning to enable Bayesian optimization-based calibration with very few model simulations; (ii) we propose the use of deep probabilistic networks to efficiently learn from large, multi-source calibration datasets; (iii) we construct a Modelica-based high-fidelity multi-source building dataset with operational equipment, internal heat loads, and realistic weather variations; and, (iv) we illustrate the effectiveness of meta-learning in speeding up calibration against conventional BO-based calibration by over 40%.

The rest of the paper is organized as follows. Section 2 recaps the basic concept and formulations of digital twin calibration for buildings and the use of Bayesian Optimization. Section 3 introduces the meta-learning calibration framework using attentive neural processes (ANPs), which are a class of probabilistic deep learning architectures. Section 4 describes the experiment configured to manifest the superiority of the proposed framework. The experiment results are presented in Section 5. Lastly, in Section 6, we

discuss the practical issues and points out the opportunities for future development.

2. Preliminaries

2.1. Data-driven optimization for model calibration

We denote by

$$y_{0:T} = \mathcal{M}_T(\theta) \tag{1}$$

a forward simulation model of the building and HVAC dynamics. The model is parameterized by the parameter vector $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$, and the admissible search space of parameters Θ is assumed to be known. For instance, Θ could denote a set of upper and lower bounds on parameters, obtained from physics or domain expertise. The output vector $y_{0:T} \in \mathbb{R}^{n_y \times T}$ contains all measured quantities available from the building system over a time interval of interest, say $[0, T]$.

The reason we consider the abstract model $\mathcal{M}_T(\theta)$ rather than choosing a model with a specific structure is because our approach is agnostic to selection of the model structure. We assume that $\mathcal{M}_T(\theta)$ is any simulation model that can be simulated on the time interval $[0, T]$, where the model is defined by a set of parameters θ. The forward simulation using $\mathcal{M}_T(\theta)$ generates a time-series of simulated outputs that we denote

$$y_{0:T} := \begin{bmatrix} y_0 & y_1 & \cdots & y_t & \cdots & y_T \end{bmatrix},$$

where at each time $t \in [0, T]$, the output y_t is a column vector of size n_y.

Since $\mathcal{M}_T(\theta)$ has no specific structure, it follows that our proposed calibration method is applicable for parameter estimation on a wide range of dynamical models. For example, consider the well-known building simulation model studied in [13, 29], which is given by

$$y_t = \chi_1(x_t, \theta_1) + \chi_2(x_t, \theta_2) + \chi_3(x_t, \theta_3), \tag{2}$$

where χ_1 denotes the energy prediction, χ_2 is the model discrepancy, χ_3 is the observation error, and θ_1, θ_2, θ_3 are the parameters defining each model component. One can recursively simulate this model from $t = 0$ to $t = T$ and obtain $y_{0:T}$. Therefore, the recursive update using (2) from $t = 0$ to $t = T$ can be written abstractly as the model $\mathcal{M}_T(\theta)$, parameterized by $\theta = [\theta_1, \theta_2, \theta_3]$.

For simulation-based model calibration (equivalently, estimation of model parameters using model simulations in a data-driven manner), we assume that we have available to us some ground-truth measured data $y^\star_{0:T}$ from the building under consideration that can be used to estimate the best parameters for the model $\mathcal{M}_T(\theta)$. Concretely, our objective is to obtain the optimal set of parameters $\theta^\star$ such that the modeling error $y^\star_{0:T} - \mathcal{M}_T(\theta^\star)$ is minimized, according to a user-defined model fit metric. Accordingly, we define the optimization problem to find the optimal parameters

$$\theta^\star = \arg\min_{\theta \in \Theta} J(y^\star_{0:T}, \mathcal{M}_T(\theta)). \tag{3}$$

While the designer is free to select any modeling error function J in (3), we select

$$J := \mathsf{MSE}(y^\star_{0:T}, \mathcal{M}_T(\theta)) = \sum_{t=0}^{T} (y^\star_t - y_t)^\top W (y^\star_t - y_t), \tag{4}$$

where W is a $n_y \times n_y$ positive-definite matrix that is used to assign importance or scale the output errors. Since the cost function is likely to be non-convex, high-dimensional, and analytical gradients are unavailable due to the unmodeled nature of the map from θ to J, gradient-based methods are unreliable as they frequently yield poor quality locally optimal solutions. Conversely, population-based methods like genetic algorithms or intelligent swarms exhibit extremely slow convergence because each model simulation is slow for large buildings, especially if also simulating HVAC dynamics. Thus, performing a large number of model simulations while searching for parameters becomes impractical, so we resort to a Bayesian optimization framework for calibration that has recently been demonstrated to be sample-efficient even for complex building energy models [19].

2.2. Classical Bayesian optimization

In high-dimensional parameter search spaces, the number of samples required to obtain near-optimal solutions to (3) can be large unless the sampling is done intelligently. The classical Bayesian optimization (BO) algorithm provides a sample-efficient way to search for optima in Θ by iterating through three steps that balance exploration and exploitation [30]:

1. Probabilistic regression methods are used to approximate the mapping from the parameter space to the calibration-cost function J. By learning a probabilistic surrogate model of the calibration cost, one can quantify the uncertainty associated with the calibration cost on Θ.
2. The statistics associated with the probabilistic surrogate cost can be used to generate subsequent search directions towards sub-regions of Θ which are most likely to contain the global solution that minimizes the cost.
3. After a new sample is acquired in the promising subregion, the probabilistic model is retrained through Bayes rule, thus incorporating new information and refining its predictions.

Gaussian processes (GP) are the prevailing probabilistic surrogate model of choice in BO due to the existence of a closed-form model update expression as well as a closed-form objective to tune it. GP functions are characterized by a mean function $\mu(\theta)$ and a kernelized covariance function $\mathcal{K}(\theta, \theta')$. The accuracy of the predicted mean and variance are strongly linked to the choice of kernel, along with the kernel parameter values, such as length-scales and variances. While many kernel functions are available,

the Matérn 3/2 function is (empirically) found to provide a good approximation of calibration-cost functions [19]. Among a variety of methods to optimize these kernel parameters, the most common one involves maximizing the log-marginal likelihood [31, Chapter 2].

The exploration-exploitation trade-off in BO methods is performed via an acquisition function $\mathcal{A}(\cdot)$. The acquisition function uses the predictive distribution given by the GP to compute the expected utility of performing an evaluation of the objective at each θ sampled from the parameter space. The next point at which the objective has to be evaluated is given by

$$\theta_{N_\theta+1} = \arg\max \mathcal{A}(\theta).$$

A commonly used acquisition function is the expected improvement (EI) function [30], given by

$$\mathcal{A}_{\mathsf{EI}}(\theta) = (J(\theta^+) - \mu(\theta) - \epsilon)\Phi(Z) + \sigma(\theta)\phi(Z), \quad (5)$$

where

$$Z = \frac{(J(\theta^+) - \mu(\theta) - \epsilon)}{\sigma(\theta)}$$

and $\Phi(\cdot)$ denotes the cumulative distribution function of a zero-mean unit-variance normal distribution, $\phi(\cdot)$ denotes the probability density function of a zero-mean unit-variance normal distribution, $\mu(\theta)$ and $\sigma(\theta)$ are the predicted mean and standard deviation at θ based on the GP surrogate cost, and $J(\theta^+)$ is the best cost encountered so far. Additionally, the parameter $\epsilon = 0.01$ is used to encourage exploration. It is common practice to maximize this acquisition function by extracting random samples on Θ, evaluating $\mathcal{A}$ at every sample, and selecting the sample that maximizes $\mathcal{A}$.

After a suitable number of iterations N_θ, the GP regressor is expected to learn the underlying function J and the best solution obtained thus far by the acquisition function is denoted the best set of parameters for the model. The selection of N_θ is a design decision: it is usually informed by practical considerations such as the total number of simulations achievable within a practical time budget. Note that N_θ is the number of model simulations, and has no relation to the size of the measured data obtained from the building. Thus, even if N_θ is large for BO convergence, this indicates that a large number of model simulations were required for calibration, not that a lot of measured data was needed for calibration.

3. Meta-Learning for Data-Efficient Calibration

3.1. Calibration from multi-source data

In the previous section, we have considered the scenario when the calibration task is performed using data from a single source. That is, the calibration cost is learned using optimization trajectory data obtained from only the building that is to be calibrated: herein, we will refer to the building to be calibrated as the *query building*. In this section, we will consider the 'multi-source' calibration setting, where the calibration cost function of the query building is learned using a combination of: (i) a very limited set of optimization trajectory data from the query building, and (ii) a significantly larger set of optimization data from related, but not necessarily identical, *source buildings*. This scenario emulates a practical scenario where calibration has been performed on a large number of building models in the past and that data has been archived, and therefore, can contribute to the calibration of a new, unseen (query) building model, with very few simulations of the new building model. Since very few simulations are needed from the query building model, this is referred to as a 'few-shot' calibration method.

Concretely, suppose that N_S is the number of source buildings whose simulation models have been calibrated in the past. Let $s \in \{1, 2, \cdots, N_S\}$. For the s-th source building, suppose the measured outputs are denoted by $y^{\star,s}_{0:T}$ and the corresponding source building simulation model is given by $\mathcal{M}^s_T(\theta)$. We assume that for all source building models, the set of parameters θ being calibrated and the admissible set of parameters Θ are the same. Note, however, that the best parameters found during calibration, $\theta^{\star,s}$, could be different for each source building model, indicating that the source buildings can all exhibit different dynamics. We assume that each source calibration task has been completed in the past; this involved searching over Θ using any calibration algorithm of choice to obtain *optimization trajectories* $\mathcal{S}^s := \{(\theta^s_k, J^s_k)\}_{k=0}^{N^s_{BO}}$, where N^s_{BO} is the number of model simulations performed using $\mathcal{M}^s_T$ during calibration. The intuition is that if the calibration procedure on the source building models was done properly, then the space Θ for each source building model has been well-explored, and a solution close to the true parameters has been found. Consequently, while calibrating the query building, we can avoid wasting resources exploring large portions of Θ and focus in on promising sub-regions of Θ that are likely to possess good parameter candidates for the query building, and therefore, reduce the number of model simulations required from the query building, thereby making our calibration mechanism efficient. Note that few-shot optimization will make the calibration procedure sample-efficient, and is not related to data-efficiency: that is, we do not assume we can control how much measured data we have available from each source building. In fact, this method can be implemented without requiring access to any measurements from the source buildings. We only require optimization trajectories, i.e., parameters and cost values, from every source calibration task. This collection of multi-source optimization trajectory data forms the training set $\mathcal{S} := \bigcup_{s=1}^{N_S} \mathcal{S}^s$ for a learning algorithm that will learn trends from a family of calibration cost functions. These learned trends, along with a few model simulations, will enable the learner to quickly adapt and estimate the true calibration cost of new, unseen query building models.

We provide a simple example to illustrate our terminology and provide some further intuition:

Example 1. *An example of source buildings are buildings that have similar geometries (e.g. townhouses) but are located at different geographical areas (such as New York and Boston). As discussed in the previous paragraph, the parameters θ to be calibrated are the same across all buildings, e.g., the emissivity coefficient of the roof. The bounds on these parameters are also the same for all buildings, which implies Θ is identical for all source tasks. The source tasks are assumed to have been completed in the past, and optimization trajectories are available for them, say with $N_\theta^1 = 2000$ model simulations for the New York townhouse, and $N_\theta^2 = 1000$ for the Boston townhouse. However, the materials used for constructing the houses are different, so the two roof emissivity coefficients are not equal. Thus, the source dataset comprise different optimization trajectories $\mathcal{S}^1$ and $\mathcal{S}^2$ where different parts of Θ have likely been explored during calibration. Now, suppose the query task is a townhouse in Chicago, for which we have only a few initial model simulations, say $N_{BO}^q = 10$. A learning algorithm can be designed to learn from the $2000 + 1000 = 3000$ data points obtained from the multi-source dataset (New York and Boston), and use the information gleaned from those buildings, along with the 10 model simulations of the query building model, to generate a good estimate of the calibration cost function of the Chicago query building. If an initial estimate of the cost function is good over Θ, then sample-efficient methods like Bayesian optimization are expected to converge in a few-shot manner i.e., with very few additional model simulations.*

3.2. Attentive neural processes (ANPs) for few-shot Bayesian optimization

In this paper, we employ attentive neural processes (ANPs) for learning from multi-source building optimization trajectories. ANPs are probabilistic deep neural networks [28] that possess some properties beneficial to this multi-source calibration problem. First, ANP training is performed using mini-batches, where the batches can contain data obtained from different sources. This enables the ANP to learn underlying trends about calibration cost functions from the multi-source dataset that are likely to remain relevant for the unseen query calibration task. Second, the ANP leverages context and latent encodings that can be used to easily adapt its encoded knowledge from the multi-source dataset to estimate a calibration cost for the query task with very limited data from the query task itself. Specifically, even with a very limited number of model simulations of the query building, one can form a *context set*

$$\mathcal{S}_\mathcal{C}^q := (\theta_\mathcal{C}^q, J_\mathcal{C}^q) := \{(\theta_k^q, J_k^q)\}_{k=1}^{N_{BO}^q}, \tag{6}$$

with which the ANP can provide a good estimate of the overall calibration cost landscape for the query building. Finally, the ANP generates probabilistic estimates, which can subsequently be leveraged to construct acquisition functions (such as those used in classical BO) to execute few-shot Bayesian optimization.

3.2.1. Training and inference

The ANP is a probabilistic deep neural network that estimates a conditional distribution on the calibration cost function, given by

$$p_{\mathsf{ANP}}(J_\mathcal{T}|\theta_\mathcal{T}, \theta_\mathcal{C}, J_\mathcal{C}), \tag{7}$$

where the conditioning input arguments are a set of context parameters $\theta_\mathcal{C}$, a set of context cost values $J_\mathcal{C}$, and a set of target parameters $\theta_\mathcal{T}$. The target points $\theta_\mathcal{T} \in \Theta$ are the locations where we wish to evaluate the conditional distribution (7). The role of the context points is to generate distributions of the cost function conditioned upon the information contained in the parameter-cost pairs $(\theta_\mathcal{C}, J_\mathcal{C})$. For example, if the context points were from the s-th source building, the ANP would generate a distribution close to the true calibration cost of the s-th source building at the target points in Θ. Similarly, if the context points are the limited parameter-cost pairs from the query building, then the ANP would generate a conditional distribution likely to emulate the query building's true cost function. Clearly, the more context points we provide, the more accurate the ANP estimates will be.

Figure 1 shows an overview of the data required for training and inference for an ANP. For training (see upper subplot), we denote $\mathcal{S}_\mathcal{C}^s$ and $\mathcal{S}_\mathcal{T}^s$ as context and target sets drawn from the s-th source task and $n_\mathcal{C}$, $n_\mathcal{T}$ as the number of context and target points, respectively. These are formed by partitioning a given source task $\mathcal{S}^s \subset \mathcal{S}$ into randomly selected context and target sets, $\mathcal{S}_\mathcal{C}^s$ and $\mathcal{S}_\mathcal{T}^s$, respectively. These sets may not necessarily be disjoint. The parameters of the target set $\theta_\mathcal{T}^s$ along with the context set $(\theta_\mathcal{C}^s, J_\mathcal{C}^s)$ can be used to infer a conditional distribution by the ANP. Under the assumption that this distribution is approximable by a Gaussian distribution, the output of the ANP is described by $\mathcal{N}(\mu_\mathcal{T}^s, \sigma_\mathcal{T}^s)$. Since the training is supervised, labels of the true cost values $J_\mathcal{T}^s$ at the target locations are known, and therefore, a loss function to be optimized such as an evidence lower-bound (ELBO), can be computed. For training, stochastic gradient descent variants are used with batching of context and target sets. By computing gradients based on random selections of context and target sets across all tasks in the multi-source dataset, we enforce that the ANP learns a family of distributions conditioned by context sets over the entire multi-source training set $\mathcal{S}$.

The inference procedure is shown in the lower subplot of Figure 1. Here, the context set consists of the limited set of optimization trajectory data obtained from the query building described in (6). The target parameters are a dense grid on Θ where we wish to evaluate the query building calibration cost. The ANP at inference has

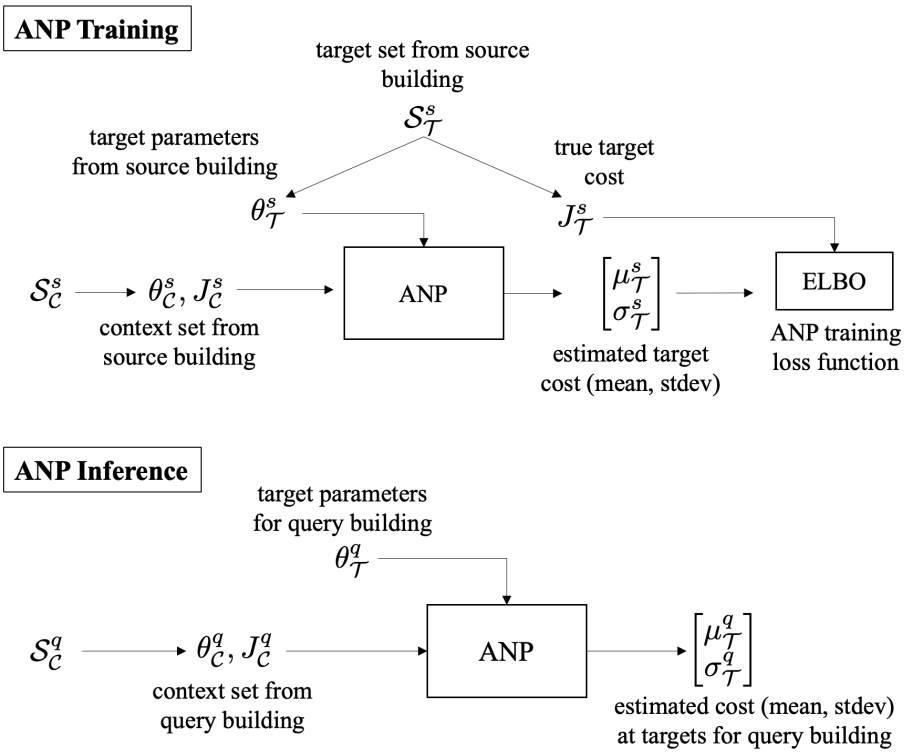


Figure 1: Schematic diagram of ANP training and inference.

been trained, so it contains embedded information from the multi-source dataset $\mathcal{S}$. Using a combination of the context points $\mathcal{S}^q_{\mathcal{C}}$ and the target parameters for the query building $\theta^q_{\mathcal{T}}$, the trained ANP can infer a conditional distribution $\mathcal{N}(\mu^q_{\mathcal{T}}, \sigma^q_{\mathcal{T}})$ that is likely to contain the true calibration cost function of the query building.

3.2.2. Implementation specifics

Figure 2 displays the ANP architecture used in this work. Internally, we factorize the conditional distribution (7) as

$$p_{\mathsf{ANP}}(J_{\mathcal{T}}|\theta_{\mathcal{T}}, \mathcal{S}^q_{\mathcal{C}}) := \int p_{\mathsf{ANP}}(J_{\mathcal{T}}|\theta_{\mathcal{T}}, \mathcal{R}^{q,\mathsf{det}}, z)\, p_{\mathsf{proxy}}(z|\mathcal{S}^q_{\mathcal{C}})\, \mathrm{d}z. \tag{8}$$

where z is a global latent variable z with a proxy prior distribution $p_{\mathsf{proxy}}(z|\mathcal{S}^q_{\mathcal{C}})$ that generates different stochastic process realizations thereby incorporating uncertainty into the predictions of target function values $J^q_{\mathcal{T}}$ despite being provided a fixed context set.

Here,

$$\mathcal{R}^{q,\mathsf{det}} = [\mathcal{R}^{q,\mathsf{det}}_1, \cdots, \mathcal{R}^{q,\mathsf{det}}_{N^q_{BO}}],$$

and each

$$\mathcal{R}^{q,\mathsf{det}}_k := \mathsf{Enc}^{\mathsf{det}}([\theta^q_k, J^q_k])$$

is the output from the deterministic encoder $\mathsf{Enc}^{\mathsf{det}}$. In the deterministic path, the ANP aggregates using a cross-attention mechanism, where each target query attends to the context points θ^q to generate the representation $\mathcal{R}^{\times}$. In particular, to generate an encoding $\mathcal{R}^{\times}_{\ell}$ for a single target point $\theta_{\mathcal{T},\ell} \in \theta_{\mathcal{T}}$, we use the multi-head attention (MHA) function [32] for the cross-attention mechanism, given by

$$\mathcal{R}^{\times}_{\ell} = \mathsf{MHA}(\theta_{\mathcal{T},\ell}, \theta^q_{\mathcal{C}}, \mathcal{R}^{q,\mathsf{det}}) = \omega_0[\mathsf{head}_1, \ \ldots, \ \mathsf{head}_{N_h}], \tag{9a}$$

where

$$\mathsf{head}_i = \mathsf{dotProductAttention}(\eta_1, \eta_2, \eta_3), \tag{9b}$$

$$\mathsf{dotProductAttention}(\eta_1, \eta_2, \eta_3) = \mathsf{softmax}\left(\frac{\eta_1 \eta_2^{\top}}{\sqrt{n_\theta}}\right)\eta_3, \tag{9c}$$

and

$$\eta_1 := \omega_1 \theta_{\mathcal{T},\ell}, \qquad \eta_2 := \omega_2 \theta^q_{\mathcal{C}}, \qquad \eta_3 := \omega_3 \mathcal{R}^{q,\mathsf{det}}. \tag{9d}$$

Here, $\omega_{0:3}$ in (9d) are attention weight matrices that are part of the trainable set of parameters for the ANP, and N_h is the number of attention heads in the multihead attention function MHA described in (9a). Each attention head (9b) is defined by the scaled dot product attention function, denoted $\mathsf{dotProductAttention}$ described in (9c). Recall that n_θ is the dimension of θ. Note that $\mathcal{R}^{\times}$ is a matrix whose columns are $\mathcal{R}^{\times}_{\ell}$.

Intuitively, the cross-attention operation generates a representation of the entire context set unique to each target point, such that the context points that are most relevant to the target point are given more importance in the representation. The $\mathsf{dotProductAttention}$ function consists

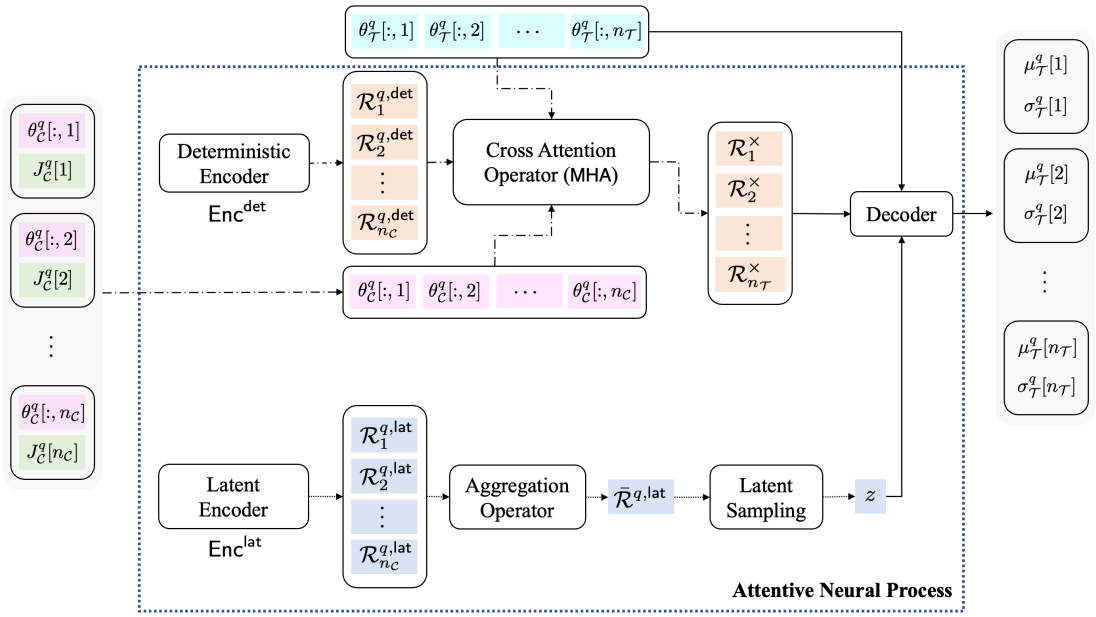


Figure 2: Architecture of the ANP. This figure shows a typical ANP pipeline for inference, with inputs and outputs kept outside the dotted rectangle. The context set parameters and costs are described by the subscript $\mathcal{C}$ and the target set parameters and costs by the subscript $\mathcal{T}$ for the query building. The total number of points in the context and targets sets are denoted $n_\mathcal{C}$ and $n_\mathcal{T}$, respectively. The deterministic path of the ANP is shown using dash-dot arrows, and the latent path with dotted arrows. The outputs of the ANP consist of a collection of means ($\mu^q_\mathcal{T}$) and standard deviations ($\sigma^q_\mathcal{T}$) for each target parameter $\theta^q_\mathcal{T}$. We use $[:,k]$ and $[k]$ to represent the k-th column or element of the corresponding quantity, respectively.

of (i) a softmax function which computes weights based on the importance of each context point, where importance is measured by the similarity of the context point to the target point, and (ii) a weighted combination of the outputs of the deterministic encoder based on these assigned similarity weights. Each such weighted encoding contributes to a single head, which typically learns to attend to a subregion in Θ. The multihead operation combines the individual heads in order to attend to multiple subregions of Θ concurrently.

Along with the deterministic path, the ANP has a latent path. The latent path has a latent encoder $\mathsf{Enc}^{\mathsf{lat}}$ which transforms the context points to

$$\mathcal{R}^{q,\mathsf{lat}}_k := \mathsf{Enc}^{\mathsf{lat}}([\theta^q_k, J^q_k])$$

similar to the deterministic encoder, from which we get

$$\mathcal{R}^{q,\mathsf{lat}} = [\mathcal{R}^{q,\mathsf{lat}}_1, \cdots, \mathcal{R}^{q,\mathsf{lat}}_{N^q_{BO}}].$$

A mean aggregation operator is employed in the latent path to form an aggregated variable $\bar{\mathcal{R}}^{q,\mathsf{lat}}$ from $\mathcal{R}^{q,\mathsf{lat}}$, which is invariant to the ordering of the context points. Subsequently, the aggregated variable $\bar{\mathcal{R}}^{q,\mathsf{lat}}$ is used for latent sampling as in a variational autoencoder (VAE) [33] to obtain a realization of the global latent z. The decoder arm of the ANP combines the outputs of the deterministic and latent paths to generate the conditional distribution $p_{\mathsf{ANP}}(J_\mathcal{T}|\theta_\mathcal{T}, \mathcal{R}^\times, z)$. This distribution is parameterized by a mean and variance, which constitute the outputs of the ANP.

To make the implementation tractable, we enforce that each point in the target set is derived from conditionally independent Gaussian distributions, and that the proxy distribution p_{proxy} is a multivariate Gaussian with a diagonal covariance matrix. This enables the use of the reparametrization trick [33] and we train the ANP to maximize the evidence-lower bound (ELBO)

$$\mathsf{E}\big[\log p_{\mathsf{ANP}}(J_\mathcal{T}|\theta_\mathcal{T}, \mathcal{R}^\times, z)\big] - \mathsf{KL}\left[p_{\mathsf{proxy}}(z|\mathcal{S}^s_\mathcal{T})||p_{\mathsf{proxy}}(z|\mathcal{S}^s_\mathcal{C})\right]$$

for randomly selected $\mathcal{S}^s_\mathcal{C}$ and $\mathcal{S}^s_\mathcal{T}$ within $\mathcal{S}$. Maximizing the expectation term $\mathsf{E}(\cdot)$ ensures good fitting properties of the ANP to the given data, while minimizing (maximizing the negative of) the KL divergence embeds the intuition that the targets and contexts arise from the same family of stochastic processes. The original ANP formulation [28] uses self-attention in both the latent and deterministic encoders, however, the complexity of the ANP with both self-attention and cross-attention is $\mathbf{O}\left(n_\mathcal{C}(n_\mathcal{C} + n_\mathcal{T})\right)$. Empirically, we observed that only using cross-attention does not deteriorate performance while resulting in a reduced complexity of approximately $\mathbf{O}\left(n_\mathcal{C} n_\mathcal{T}\right)$, which is beneficial because $n_\mathcal{T}$ is fixed, but $n_\mathcal{C}$ grows with BO iterations.

3.2.3. Few-shot Bayesian optimization with ANP

Since the output of the ANP is a Gaussian distribution by design, we can use an ANP instead of using GP as in classical BO (see §2.2). We refer to ANP-based Bayesian optimization as ANP-BO. ANP-BO has a few distinct advantages over GP-BO: first, it can scale well to large multi-source datasets, which GP cannot; it can generate more

varied inference distributions than a GP; and finally, it can enable BO to be performed in high-dimensional parameter spaces.

Having learned from $\mathcal{S}$, ANP infers the target objective function with a few context points in Θ. Instead of retraining the traditional surrogate model every iteration, the ANP-BO procedure is:

1. A large number of target points $\theta_{\mathcal{T}}^q$ are randomly sampled from the parameter space Θ. ANP is used to estimate the corresponding distribution of objective values given context points $(\theta_{\mathcal{C}}^q, J_{\mathcal{C}}^q)$.
2. The predicted distribution is used to evaluate a given acquisition function $\mathcal{A}(\theta)$ and identify the sub-regions of Θ where the global solution $\theta^\star$ most likely exist.
3. A new sample is acquired in the promising sub-region and evaluated in the simulation. The obtained (θ, J) pair is appended to the context set in the next iteration. ANP need not be retrained.

These steps are repeated until a stopping criterion is met. Leveraging the information from the relevant source tasks, ANP-BO is expected to converge faster than a GP-BO where the GP is trained from scratch.

4. Experimental Setup for Multi-Source Calibration

4.1. Building simulation model library for multi-source calibration

In this section, we describe how we design a testbed for multi-source calibration. Figure 3 illustrates both the testbed and the calibration process, including the overall multi-source dataset generation, meta learning via ANP, and calibration via ANP-BO. In the dataset generation and ANP training stage, we constructed 60 multi-source building simulation models, of which 48 were used to generate optimization trajectories for training the ANP. In the calibration performance validation stage, we then used the remaining 12 models to test the performance of the ANP-BO calibration method and to compare the ANP-BO with a classical GP-BO calibration method to demonstrate the data efficiency of ANP-BO during the calibration procedure. The library of simulation models was constructed using the Modelica[4] language because the models can be used for multiple purposes: for example, they can be used in a standalone configuration to study the hygrothermal dynamics of the building envelope, or they can be interconnected with dynamic models of HVAC and renewable energy sources to simulate and analyze the overall behavior of the coupled system. An additional benefit of the Modelica models is that they can be compiled into standalone executable binaries for running simulations and seamlessly integrated into Python machine learning tool chains via the Functional Mockup Interface (FMI)[5].

The Modelica building envelope models are based on the well-studied US DOE Residential Prototype Buildings Library (RPBL) [34]. The RPBL comprises a group of EnergyPlus[6] building simulation models for similar single-family houses located across different climate zones in the US, and correspond to standards described in the latest International Energy Conservation Code (2018 IECC). The building models all have similar dimensions, consisting of a conditioned two-story living unit and an unconditioned attic with inclined roofs (see Figure 4a). The floor of the living unit is exposed to one of the four foundation types: slab, crawl space, heated basement, and unheated basement. Each of these four types of building geometries is located in the 15 typical climate zones in the US, and many of the building parameters are adjusted according to the climate zone. Parameter variations include changes in the thickness of the insulation layers, the conductivity of the windows, and the effective leakage area. As a result, there are a total of $4 \times 15 = 60$ simulation models with myriad thermal behavior. Figure 4b displays the large variation reflected in 5-day trajectories of the living unit temperature of the 60 models with the HVAC systems deactivated during the first week of January, where up to 40°C of difference can be observed. The differences visible in this plot can largely be attributed to the variation in ambient conditions, as the climate zones include data that ranges from Hawaii to Alaska.

Several measures were applied to both the Modelica and EnergyPlus models to ensure consistency between them. First, the default Surface Convection Algorithm `DOE-2` in EnergyPlus was replaced with the more precise `TARP` (Thermal Analysis Research Program) that calculates the convective heat transfer coefficient with temperature difference and wind speed. Next, year-long simulations were conducted for the EnergyPlus models to generate signals of internal heat gains, which were fed into the Modelica models as boundary conditions. Lastly, the thermostats were disabled to enable the prediction of the free-floating temperatures and thereby validate the building-side model dynamics. Figure 5 illustrates the model outputs of both the EnergyPlus and Modelica models for a simulation of a residential house with a crawlspace located in Memphis, TN (climate zone 3A) for January 1-5 for the purposes of comparison. The minor discrepancies between the two models are caused by the use of different solvers and underlying calculation methods, such as steady state approximations used in the EnergyPlus models.

[4]Modelica (CITE) is an open-source component-oriented, equation-based language for modeling multiphysical systems.

[5]The Functional Mock-up Interface (FMI) (CITE) is an open standard that defines a container and an interface to exchange dynamic models between simulation environments.

[6]EnergyPlus is a console-based whole building energy simulation program that engineers, architects, and researchers use to model energy consumption and the indoor environment in buildings.

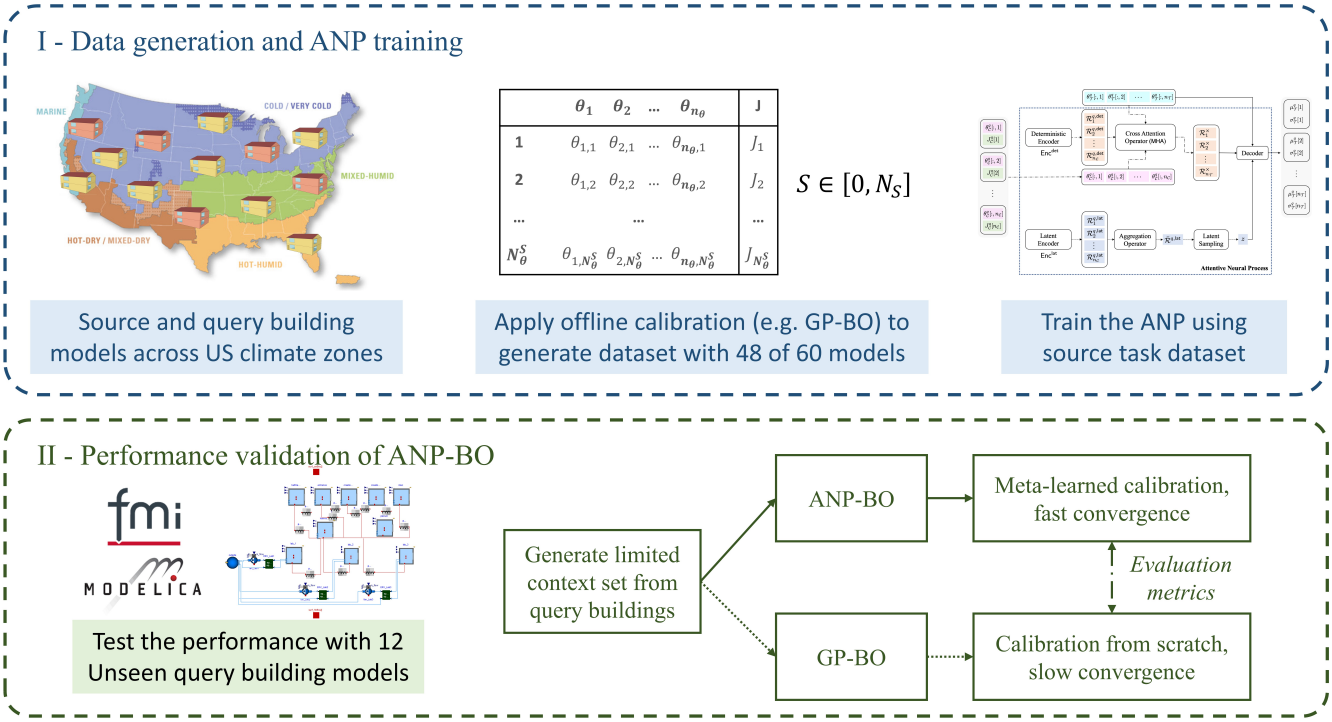
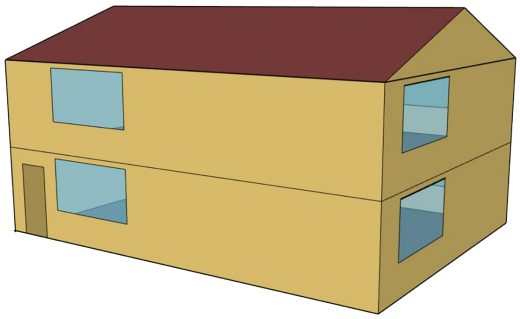
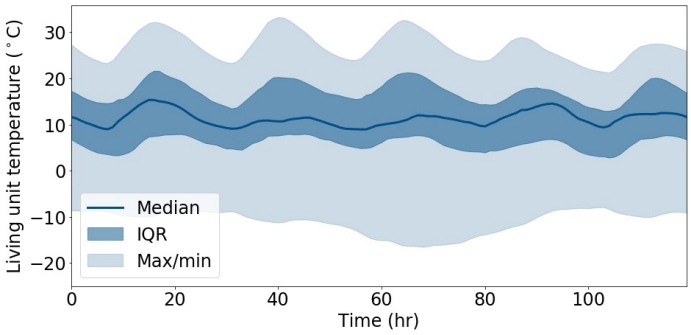


Figure 3: Experimental setup for multi-source calibration via ANP-BO.



(a) Geometry of the building model rendered by SketchUp.



(b) Median, IQR, and extrema of temperatures of the 60 buildings.

Figure 4: Simulation models built for the calibration experiment have similar geometry but myriad thermal behavior.

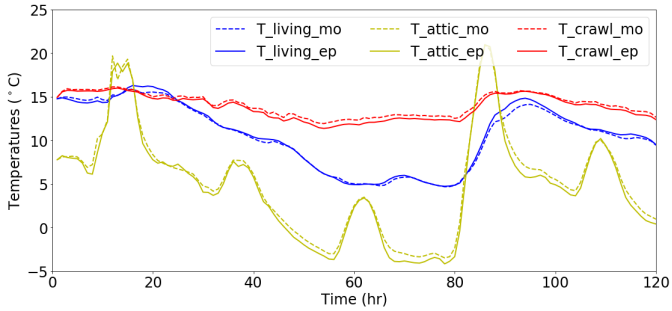


Figure 5: Free-floating temperature predictions (living unit, attic, and crawlspace) of a pair of Modelica and Energyplus models.

4.2. Configurations of the calibration tasks

The calibration task for each source building model is to identify the true IECC parameters of the model. Each source building model is assumed available during calibration, and ranges on the parameters Θ is known, although the true parameter values are unknown. The measurements available from each source building is considered to be a combination of 6 outputs, acquired hourly for 4 days, resulting in 96 time samples. Therefore, $y_{0:T} \in \mathbb{R}^{6\times 96}$. The 6 measurements are: room and attic temperatures (°C), room relative humidity (%), and HVAC consumption (fan, heating, and cooling, kW). Five parameters $\theta \in \Theta \subset \mathbb{R}^5$ chosen to be sensitive to these outputs are calibrated for each source building. These parameters are specifically selected to be varying to (different degrees)

amongst the source and query buildings, and include: (i) external roof solar emissivity (small variation), (ii) room effective infiltration leakage area (medium variation), (iii) fan efficiency (medium variation), (iv) nominal COP (coefficient of performance, high variation), and (v) window thermal conductivity (high variation). Table 1 summarizes the admissible parameter space Θ that covers all the true values, and this admissible parameter space is constant for all calibration tasks.

Table 1: Admissible parameter space applied across all tasks.

Parameter	lower bound	upper bound
Roof emissivity	0.6	0.9
Leakage area	150	750
Fan efficiency	0.3	0.7
Nominal COP	3	5
Window conductivity	0.1	0.3

Calibrating multiple parameters simultaneously may cause identifiability issues, where different combinations of parameters could achieve similar magnitudes of prediction errors. To eliminate this issue, the four-day simulation comprises two days of unconditioned free-floating and two days with HVAC activated. The first two days help identify the building parameters with decoupled dynamics, and the latter two account for coupled dynamics and contribute to HVAC calibration. The BO objective function was the exponential of negative MSE, which has a theoretical optimal value of 1 when $y^{\star}_{0:T} = \mathcal{M}_T(\theta^{\star})$ and MSE is zero. While the exponential objective function is applied in this study, the algorithm can incorporate other forms such as the logarithm function. HVAC power outputs were weighted by 10 to compensate for the smaller absolute values. Formally, the calibration task involves solving the optimization problem

$$\theta^{\star} = \arg\max_{\theta \in \Theta} \ \exp(-\mathsf{MSE}(y^{\star}_{0:T}, \mathcal{M}_T(\theta))) \qquad (10)$$

Since the cost converges to 1 as $\mathsf{MSE} \to 0$, we consider that the building model has successfully been calibrated when the cost (10) exceeds 0.95.

4.3. Source dataset generation and ANP training

We randomly selected 48 out of the 60 houses and applied classical GP-based BO (GP-BO) for calibration to generate source task data. For each of these 48 tasks, GP-BO included 300 random samples for initialization and 100 optimization iterations. Over the iterations, the acquisition function $\mathcal{A}(\theta)$ was evaluated by the EI acquisition function, defined earlier in (5) The resulting training dataset is described by

$$\mathcal{S} := \cup_{k=1}^{48} \{(\theta_t^k, J_t^k)\}_{t=1}^{400}$$

where θ_t^k denotes the calibrated parameters and J_t^k is the corresponding objective function value.

In Figure 6, we attempt to visualize a subset of 15 randomly selected calibration cost functions from the multi-source dataset $\mathcal{S}$. In order to visualize the 5-dimensional parameter space, we used kernel principal components analysis (PCA) to obtain a reduced 2-dimensional space of principal components; we visualize by regressing over this projected 2-D space. Noticeable variations across tasks can be observed, reflected in the positions of global optima, the number of local optima, function steepness, and etc. These variations can be attributed to three factors: 1) the true parameter values according to IECC; 2) the exogenous disturbances (outdoor weather) of different climate zones; 3) the boundary conditions varied by the foundation types.

We use $\mathcal{S}$ to train the ANP. To emulate situations of different levels of source data abundancy, two models were trained respectively with the entire training set $\mathcal{S}$ (ANP100) and 50% of randomly-selected sources tasks (ANP50). We use Adam [35] for training the ANP over 20000 iterations with a batch size of 32, and four source tasks were randomly selected for the ANP validation set. In each mini-batch, context points and target points were randomly sampled from the source tasks to maximize the ELBO loss. The deterministic encoder, latent encoder, and decoder were all configured to have three hidden layers and 256 neurons in each layer, while the cross attention has 16 heads. GP and ANP were implemented using `GPyTorch`[7] and `PyTorch`[8], interacting with the FMUs using `FMPy`[9].

4.4. Calibration and performance test

The 12 buildings not included in the training dataset generation were used to test the calibration performance of ANP-BO and benchmark against GP-BO. The calibration configurations (parameters to be calibrated, and measured outputs of the building simulation models) were the same as those considered in the source tasks. Since the ANP infers the query building cost function based on limited context points, the initial samples are desired to be sparse for better contextualization, especially in high-dimensional parameter space. Hence, Latin hypercube sampling (LHS) [36] was adopted instead of uniform random sampling to improve coverage of the samples over Θ, and to avoid clustering of points in small subregions.

The same random samples were then used to initialize ANP-BO and GP-BO (see Figure 3). For the query building, both ANP-BO and GP-BO were allowed 100 iterations for calibration and the algorithms were prematurely terminated if the cost exceeded 0.95. The calibration results, convergence rate, and the total number of simulations were compared. To evaluate robustness, the calibration experiments were conducted multiple times with the number of initial samples set to 30, 50, and 100.

[7] https://gpytorch.ai/
[8] https://pytorch.org/
[9] https://github.com/CATIA-Systems/FMPy

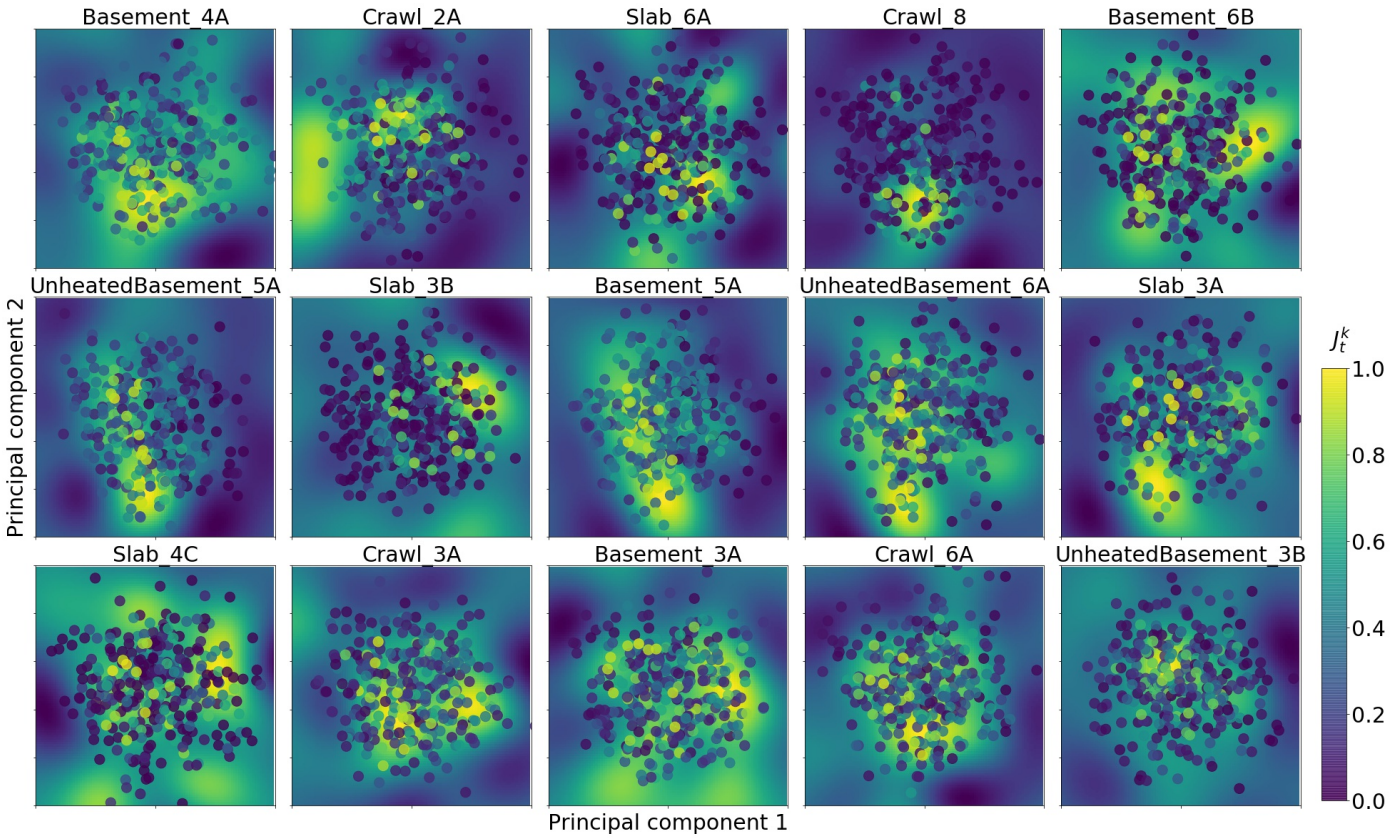


Figure 6: Visualization of 2 principal components of calibration cost functions obtained from multi-source training set.

5. Experiment results

5.1. Data efficiency in calibration

Figure 7 summarizes the distributions of total number of simulations N_{total} consumed by alternative models for the calibration to converge. Most test cases (over 95%) converged within dozens of optimization iterations using the three surrogate models (ANP100, ANP50, and GP). However, several did not reach the 0.95 threshold before the preset limit (150 iterations), which are counted as $N_{initial} + 150$ in the box plots.

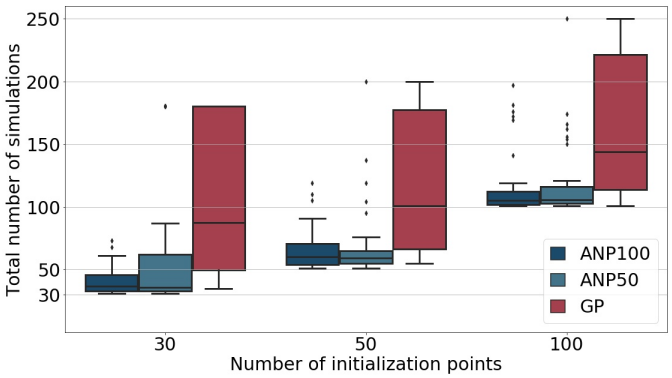


Figure 7: Number of simulations consumed to meet the convergence criterion with different surrogate models in BO.

For the classical GP-BO, 30 initial samples are usually insufficient to guarantee convergence, and $N_{initial}$ of 50 and 100 performed similarly, requiring about 50 iterations on average. It is conspicuous that ANP100 and ANP50 found the solutions much faster than the baseline with all three $N_{initial}$. The average N_{total} using these two models were close, while the variance of ANP100 was smaller for $N_{initial} = 30/100$ but larger for 50. On one hand, this indicates that 50% of source tasks were sufficient to effectively facilitate the calibration of unseen query tasks and the additional 50% caused only marginal improvement. On the other hand, this reveals the importance of initial context points, which is further investigated in the next subsection.

Figure 8 illustrates the optimization process with ANP50-BO and GP-BO by plotting the highest score achieved so far over BO iterations across all test cases. The plots of ANP100 were omitted for better legibility as they mostly overlaps ANP50. It can be seen that the convergence rate of ANP-BO drastically increased once random exploration stopped and optimization iterations started. The effect was more significant with 30 initial points. In contrast, GP-BO progressed only slightly faster than the random initialization stage in all cases.

5.2. ANP prediction results

To explain the calibration results, we compared the inference quality of the ANP variants and the GP with a

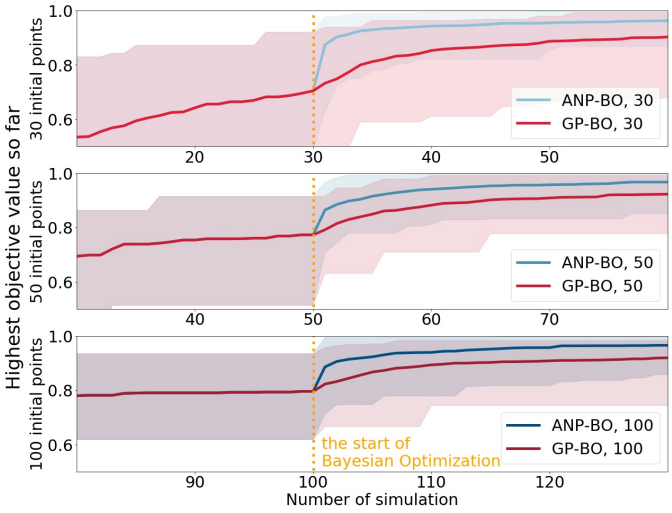


Figure 8: Highest objective value so far over iterations of info-medium ANP-BO and GP-BO (solid lines for means and shadings for interquartile ranges).

varying number of context points. Figure 9 compares the calibration cost function predictions for an unseen query building. In the figure, each row corresponds to one learning algorithm and each column shows the prediction results at 40 target points $\mathcal{T}$, given the same context points (training data for GP). The first 20 target points are randomly sampled from the entire parameter space Θ in order to provide insight into how well the ANP/GP fits the query calibration cost over all of Θ. The final 20 target points are deliberately chosen to be in a small neighborhood of the true parameters for this query building. Note that this is only for comparing the prediction quality of the learners, and we do not assume knowledge of the true parameters during calibration. The goodness of fit is evaluated using $\mathcal{L}_2$ distance as well as the predictive interval coverage probability (PICP), where

$$\mathcal{L}_2 = \sqrt{\frac{\sum_{\theta\in\mathcal{T}} \left(\mu(\theta) - J(\theta)\right)^2}{n_{\mathcal{T}}^q}}$$

and

$$\text{PICP} = \frac{\#\{\theta \in \mathcal{T} \mid J(\theta) \in 99\%\text{CI}\}}{n_{\mathcal{T}}^q}.$$

Here, $\mu(\theta)$ is the predicted mean, $J(\theta)$ is the actual reward evaluated by simulation, $99\%\text{CI} = \mu(\theta) \pm 2.58\sigma(\theta)$ is the 99% confidence interval predicted by the models with the standard deviation $\sigma(\theta)$, and $n_{\mathcal{T}}^q$ is the total number of tested target points.

The predictive performance of surrogate models aligns with the calibration results. From the figure, we observe that the ANP100 consistently outperforms the GP over 30, 50, and 100 context points, based on the PICP score. However, the GP, owing to its non-parametric nature, exhibits a lower $\mathcal{L}_2$ cost with increasing number of context points. However, this did not help the calibration procedure as we have seen in the previous subsection, because the true underlying cost function is not contained within its 99% confidence interval. Since the ANP does contain the true calibration cost within its confidence interval, the acquisition function in a BO-based method can exploit this for more efficient calibration. Additionally, few-shot calibration is expected to perform well on limited data settings, so the efficacy of ANP with 30 context points is more important than with larger context points.

The higher PICP values for the ANP100 compared with the GP are due to ANP having been trained on a multi-source dataset: unlike the GP, the ANP learns that the true calibration cost function distribution is not necessarily tight around the predictive mean. Therefore, it generates wider uncertainty bounds around this mean function based on the uncertainty learned from the training set. This trend of high PICP is also observed on the other 11 query buildings.

As expected, reducing the training dataset resulted in a decrease in predictive quality of the ANP50. However, its calibration performance did not deteriorate significantly, and this is because the trends of objective function were well-captured, which the BO can utilize and locate the optimal θ. The fact that, even with limited data, ANP outperforms GP during calibration, is evident from Figure 7.

It is also observed that the predictive performance decreased and then increased as more context points were taken. This corresponds to the calibration results and is due to the sparsity of context points in high-dimensional parameter space. Due to the curse of dimensionality and the fact that objective functions are usually steep in a small region around the global optimum, the ratio of points outside of this region keeps increasing when the total number of LHS points increases but remains sparse. Consequently, less attention is applied to the points close to the optimum. Most of the outside points come with small J and tend to flatten the inferred objective function. This explains the ANP's slightly worse performance with 50 initial context points. The phenomenon ceases when number of context points keeps increasing and becomes dense enough, after which more points provide better context for function inference. The better performance with 100 context points serves as an illustration, and the improving trend was continued with even more context points.

5.3. Calibrated models

As the parameter values are different across test cases, the calibrations are summarized in figure 10 as the resulting ratios $\mathcal{R} = \theta^*/\theta^{true}$ (calibrated parameters over the ground truth). Using alternative surrogate models observed no significant difference in this comparison. All the five parameters were well calibrated in most cases, indicated by the ratios being close to 1. However, a number of points fall beyond $[0.9, 1.1]$, especially for the two HVAC system parameters. Most of these undesirable points are from climate zone 5A and 8 as marked by star. Houses

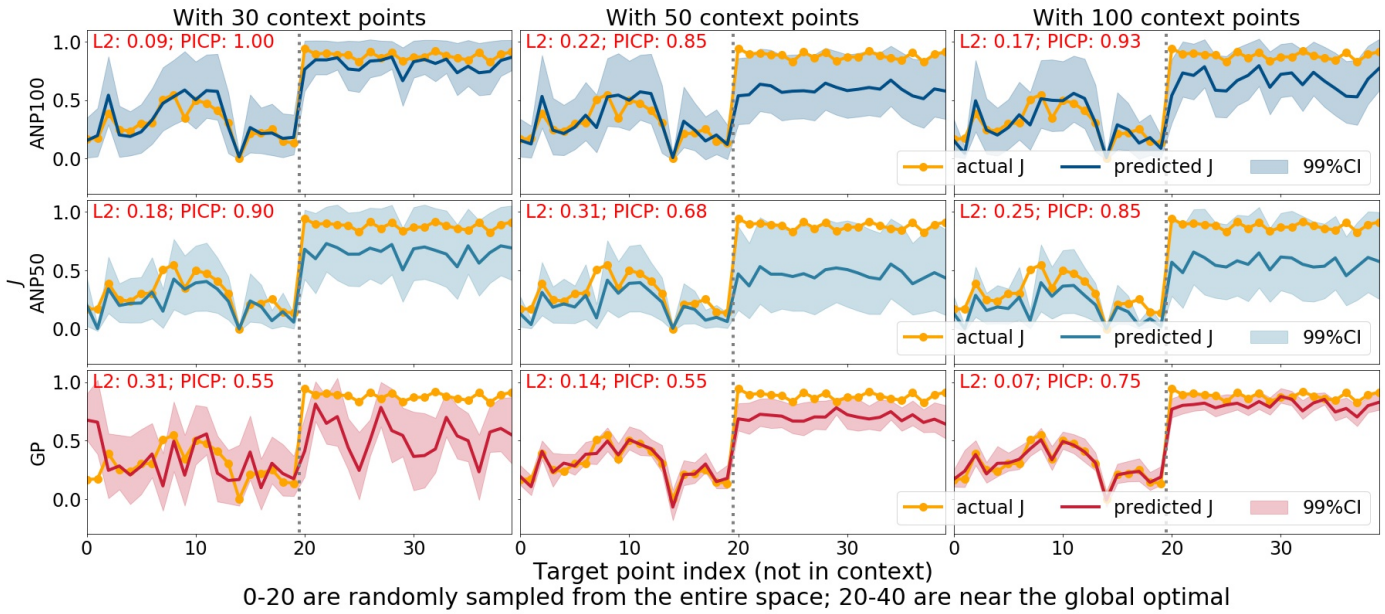


Figure 9: Target point prediction results of ANPs and GP with 30, 50, and 100 LHS context points for one test case.

in these two climate zones had smaller HVAC load during the simulation period. The energy consumption was therefore overwhelmed by other outputs, and the low part load ratio made the parameters insensitive. Consequently, the optimizations could not identify the true parameter values.

Models with calibrated parameters close to the ground truth are nearly identical to the true model, and therefore extrapolated almost perfectly outside of the calibration period as expected. As a worst case scenario, figure 11 validates a calibrated model from climate zone 5A, where $\mathcal{R} = [1.00, 0.96, 1.05, 1.12, 1.24]$. For better robustness, the testing period was two month away from the calibration period, and the operation scheme was changed to 3-day conditioning followed by 1-day unconditioned free-floating. Because of the accurate building thermal parameters, the indoor condition outputs in the first subplot perfectly matches the ground truth. Some deviations can be noticed in the HVAC power outputs, especially in the fan power. Yet, the resulting CVRMSE are 22.2% (cooling), 14.9% (heating), and 24.4% (fan), kept lower than the 30% requirement on hourly predictions as per the ASHRAE guideline 14 [37].

6. Discussion

The virtual testbed in the experiment involves houses with similar geometries but various boundary conditions, yielding relevant but diversified objective functions. Thus, the promising results demonstrate the meta-learner's capability of effectively learning a vast family of objective functions and fastening model calibrations. Given a few context points from a query task, the distribution of objective functions can be narrowed to the related region and accordingly forward the optimization.

Meanwhile, it is recognized that a quick and successful calibration relies on the query task being similar to some of the source tasks. If too few tasks are included in the training data, the probability of an unseen query task being similar is lowered, and the calibration performance could be deteriorated. Therefore, it is beneficial to incorporate more heterogeneous source tasks in practice. Calibration history of any newly-encountered building, not restricted to BO-based, can be appended to the training dataset as source tasks. ANP can be regularly retrained and thereby become more powerful.

Apart from the similarity to source tasks, another important factor is to configure a well-posed calibration problem that has one unique solution. There are many interrelated parameters in building simulation models to be calibrated. It is almost destined to have different combinations of parameters that can produce close results for one output, especially when HVAC systems and building geometries are calibrated together. For example, the error in HVAC power can be eliminated by adjusting either the building load or the system efficiency. Therefore, it is necessary to deal with the parameter identifiability carefully [38]. To this end, we accounted for multiple outputs and combined different operation schemes when defining the calibration problem.

A well-posed calibration problem also require the multiple outputs to be of similar importance in the objective function. Therefore, proper weights should be applied for outputs at different scales. We weighted the HVAC power outputs by 10 as their absolute values are smaller than the temperature and humidity measurements. This weighting strategy helped find the unique solution for most buildings, but not for buildings in climate zone 5A and 8 as elucidated

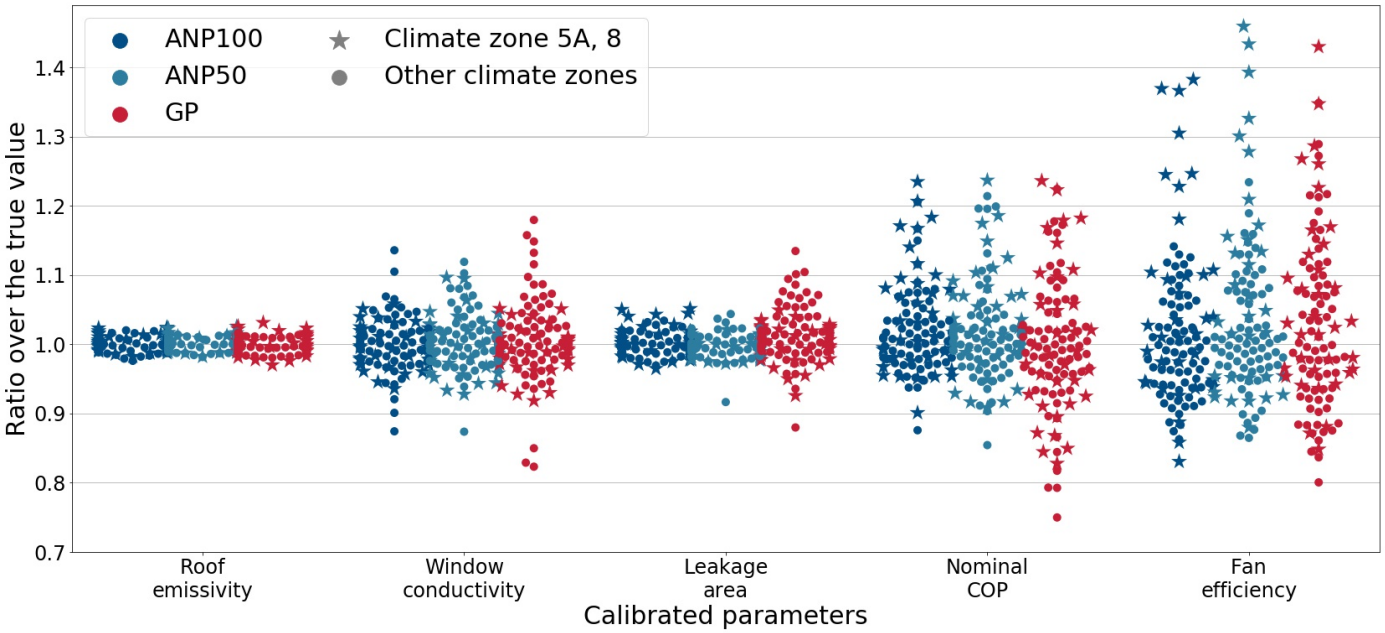


Figure 10: Ratios of calibrated parameters over corresponding true values ($\mathcal{R} = \theta^*/\theta^{true}$) for all test cases. Note that climate zones 5A and 8 are significantly different from the source data.

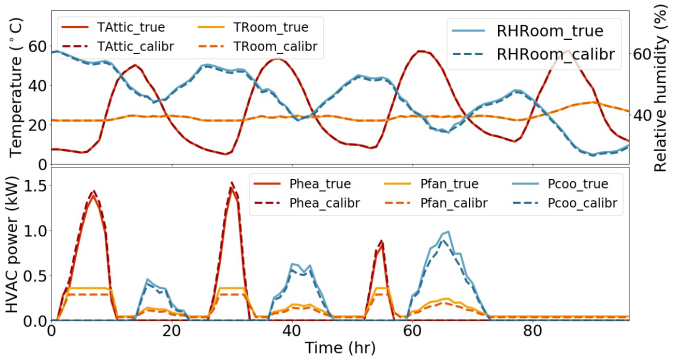


Figure 11: Simulation output comparison of the house with conditioned basement located in climate zone 5A.

in figure 10 and 11. The solution would be to further adjust the weights for these target tasks. Increased weights of 30 were tested for three times in the same test case in figure 11 and observed significant improvement. With 30 initial context points, the optimization still converged within 10 iterations in all three test runs. The mean of calibrated parameters present $\mathcal{R} = [0.98, 0.99, 0.97, 1.08, 0.99]$, and the average testing CVRMSE are 13.8% (cooling), 5.6% (heating), and 2.5% (fan). Thus, slightly changed objective functions can be handled by ANP-BO without retraining.

6.1. Opportunities for future research

Calibrating simulation models for actual buildings instead of the virtual testbed involves more uncertainties, leading to two issues for further research. First, the efficacy of the proposed algorithm is to be tested on more heterogeneous buildings. Also, while the admissible parameter space Θ is consistent across all tasks in the experiments, larger variability is expected in practice. The difference can be reflected in the user-defined parameter ranges and the types of parameters. Theoretically, ANP can naturally incorporate various parameter ranges from different source tasks. Yet, the applicability across different parameter spaces is to be investigated. A larger number of source tasks may be needed.

In addition to building model calibration, the idea of meta-learning from multi-source data may be applied for other purposes in buildings. Since the core is to learn the distribution of objective functions from related tasks, the potential lies in many optimization-based applications such as design and operation.

7. Conclusions

Building simulation model calibration is critical for improving building energy efficiency. Current approaches are typically computational expensive and therefore lacks scalability. This paper address this challenge by proposing a meta-learned Bayesian Optimization framework for building digital twin calibration based on ANP. The concept is demonstrated using an open-source US DOE-validated library of residential building models across different climate zones and with different construction types. The benchmarking results show that ANP outperformed the baseline GP in inferring the objective functions with limited data and thereby improve the data efficiency of BO-based calibration. Key factors for success are pinpointed through the comprehensive evaluation of calibration results. This research provides a promising approach of obtaining re-

liable building simulation models, which promotes many scalable applications for building energy conservation.

References

[1] M. Sofos, J. T. Langevin, M. Deru, E. Gupta, K. S. Benne, D. Blum, et al., Innovations in sensors and controls for building energy management: Research and development opportunities report for emerging technologies, Tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States) (2020).

[2] A. Handbook, Fundamentals, ashrae–american society of heating, Ventilating and Air-Conditioning Engineers (2017).

[3] D. Hou, I. Hassan, L. Wang, Review on building energy model calibration by bayesian inference, Renewable and Sustainable Energy Reviews 143 (2021) 110930.

[4] S. Huang, Y. Lin, V. Chinde, X. Ma, J. Lian, Simulation-based performance evaluation of model predictive control for building energy systems, Applied Energy 281 (2021) 116027.

[5] A. Kathirgamanathan, M. De Rosa, E. Mangina, D. P. Finn, Data-driven predictive control for unlocking building energy flexibility: A review, Renewable and Sustainable Energy Reviews 135 (2021) 110120.

[6] H. H. Hosamo, P. R. Svennevig, K. Svidt, D. Han, H. K. Nielsen, A digital twin predictive maintenance framework of air handling units based on automatic fault detection and diagnostics, Energy and Buildings 261 (2022) 111988.

[7] G. P. Lydon, S. Caranovic, I. Hischier, A. Schlueter, Coupled simulation of thermally active building systems to support a digital twin, Energy and Buildings 202 (2019) 109298.

[8] A. Chakrabarty, C. Danielson, S. A. Bortoff, C. R. Laughman, Accelerating self-optimization control of refrigerant cycles with bayesian optimization and adaptive moment estimation, Applied Thermal Engineering 197 (2021) 117335.

[9] K. A. Barber, M. Krarti, A review of optimization based tools for design and control of building energy systems, Renewable and Sustainable Energy Reviews 160 (2022) 112359.

[10] H. Metzmacher, M. Syndicus, A. Warthmann, C. van Treeck, Exploratory comparison of control algorithms and machine learning as regulators for a personalized climatization system, Energy and Buildings 255 (2022) 111653.

[11] A. Chong, K. P. Lam, M. Pozzi, J. Yang, Bayesian calibration of building energy models with large datasets, Energy and Buildings 154 (2017) 343–355.

[12] G. Chaudhary, J. New, J. Sanyal, P. Im, Z. O'Neill, V. Garg, Evaluation of "autotune" calibration against manual calibration of building energy models, Applied energy 182 (2016) 115–134.

[13] A. Chong, W. Xu, S. Chao, N.-T. Ngo, Continuous-time bayesian calibration of energy models using BIM and energy data, Energy and Buildings 194 (2019) 177–190.

[14] A. Chong, K. Menberg, Guidelines for the bayesian calibration of building energy models, Energy and Buildings 174 (2018) 527–547.

[15] A. Chong, G. Augenbroe, D. Yan, Occupancy data at different spatial resolutions: Building energy performance and model calibration, Applied Energy 286 (2021) 116492.

[16] M. Manfren, N. Aste, R. Moshksar, Calibration and uncertainty analysis for computer models–a meta-model based approach for integrated building energy simulation, Applied energy 103 (2013) 627–641.

[17] H. Lim, Z. J. Zhai, Comprehensive evaluation of the influence of meta-models on Bayesian calibration, Energy and Buildings 155 (2017) 66–75.

[18] J. Chen, X. Gao, Y. Hu, Z. Zeng, Y. Liu, A meta-model-based optimization approach for fast and reliable calibration of building energy models, Energy 188 (2019) 116046.

[19] A. Chakrabarty, E. Maddalena, H. Qiao, C. Laughman, Scalable Bayesian optimization for model calibration: Case study on coupled building and HVAC dynamics, Energy and Buildings (2021) 111460.

[20] K. Lawal, H. N. Rafsanjani, Trends, benefits, risks, and challenges of IoT implementation in residential and commercial buildings, Energy and Built Environment (2021).

[21] Y. Chen, Z. Tong, Y. Zheng, H. Samuelson, L. Norford, Transfer learning with deep neural networks for model predictive control of hvac and natural ventilation in smart buildings, Journal of Cleaner Production 254 (2020) 119866.

[22] A. Li, F. Xiao, C. Fan, M. Hu, Development of an ann-based building energy model for information-poor buildings using transfer learning, in: Building Simulation, Vol. 14, Springer, 2021, pp. 89–101.

[23] C. Cui, T. Wu, M. Hu, J. D. Weir, X. Li, Short-term building energy model recommendation system: A meta-learning approach, Applied energy 172 (2016) 251–263.

[24] W. Li, G. Gong, H. Fan, P. Peng, L. Chun, Meta-learning strategy based on user preferences and a machine recommendation system for real-time cooling load and cop forecasting, Applied Energy 270 (2020) 115144.

[25] A. Chong, Y. Gu, H. Jia, Calibrating building energy simulation models: A review of the basics to guide future work, Energy and Buildings 253 (2021) 111533.

[26] T. M. Hospedales, A. Antoniou, P. Micaelli, A. J. Storkey, Meta-learning in neural networks: A survey, IEEE transactions on pattern analysis and machine intelligence (2020).

[27] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1126–1135.

[28] H. Kim, A. Mnih, J. Schwarz, M. Garnelo, A. Eslami, D. Rosenbaum, O. Vinyals, Y. W. Teh, Attentive neural processes, in: International Conference on Learning Representations, 2019.

[29] Y. Heo, et al., Evaluation of calibration efficacy under different levels of uncertainty, Journal of Building Performance Simulation 8 (3) (2015) 135–144.

[30] J. Snoek, H. Larochelle, R. P. Adams, Practical bayesian optimization of machine learning algorithms, NeurIPS 25 (2012) 2951–2959.

[31] C. K. Williams, C. E. Rasmussen, Gaussian Processes For Machine Learning, Vol. 2, MIT press Cambridge, MA, 2006.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[33] D. P. Kingma, M. Welling, An introduction to variational autoencoders, arXiv preprint arXiv:1906.02691 (2019).

[34] B. E. C. P. DOE, Residential prototype building models (2021).

[35] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[36] J. C. Helton, F. J. Davis, Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems, Reliability Engineering & System Safety 81 (1) (2003) 23–69.

[37] ASHRAE, Guideline 14, measurement of energy and demand savings, American Society of Heating, Ventilating, and Air Conditioning Engineers, Atlanta, Georgia (2014).

[38] D. H. Yi, D. W. Kim, C. S. Park, Parameter identifiability in bayesian inference for building energy models, Energy and Buildings 198 (2019) 318–328.

Symbol	Meaning
Building Modeling	
$\mathbb{R}$	Set of real numbers
$\mathbb{N}$	Set of natural numbers
T	Total simulation time
θ	Parameters to be calibrated
$\theta^\star$	Optimal parameter vector
Θ	Search space of admissible parameters
n_θ	Number of calibrated parameters
n_y	Number of measurable outputs
$y_{0:T}$	Simulated output vector on time interval $[0, T]$
$y^\star_{0:T}$	True measured outputs on time interval $[0, T]$
$\mathcal{M}_T(\theta)$	Forward model for simulation parameterized by θ
$\mathcal{M}_T(\theta^\star)$	Optimally parameterized forward model for simulation
J	Calibration cost function
W	Weight matrix for calibration cost function
MSE	Mean squared error function for calibration
$\#\{A\}$	Cardinality of a set A
Bayesian Optimization	
$\mathcal{K}$	Kernel function used in Gaussian process (GP) regression
μ	Mean function used in GP regression
σ	Standard deviation function used in GP regression
$\mathcal{A}$	Acquisition function for Bayesian optimization
$\mathcal{A}_{\mathsf{EI}}$	Expected improvement acquisition function
N_θ	Number of Bayesian optimization iterations
Meta Learning from Multi-Source Data	
s	Index of source tasks
$\mathcal{M}^s_T$	Simulation model of s-th source building
$\theta^{s,\star}$	Best parameter found for s-th source task
N^s_{BO}	Number of model simulations for s-th source task
$\mathcal{S}^s$	Optimization trajectory collected from s-th source task
$\mathcal{S}$	Dataset from all source tasks for training ANP
$\mathcal{S}^q$	Limited optimization trajectory from query task
N^q_{BO}	Limited number of model simulations on query task
$\mathcal{M}^q_T$	Simulation model of query building
$\theta^{q,\star}$	Best parameter found for query task
Attentive Neural Processes	
p_{ANP}	Conditional distribution induced by ANP
p_{proxy}	Proxy distribution for training ANP by variational methods
z	ANP global latent variable
$\mathcal{N}(\mu, \sigma)$	Gaussian density function with mean μ and variance σ^2
$\mathcal{S}^s_\mathcal{C}$ / $\mathcal{S}^s_\mathcal{T}$	General context/target sets drawn from $\mathcal{S}^s$ during ANP training
$n_\mathcal{C}$ / $n_\mathcal{T}$	Number of context/target points
$\theta^q_\mathcal{C}$	Context set of parameters from query building
$J^q_\mathcal{C}$	Context set of calibration cost values from query building
$\theta_\mathcal{T}$	Target set of parameters where ANP will estimate cost
$J_\mathcal{T}$	Target set of calibration cost to be inferred by ANP
$\mathsf{Enc}^{\mathsf{det}}$	Deterministic encoder
$\mathsf{Enc}^{\mathsf{lat}}$	Latent encoder
$\mathcal{R}^{q,\mathsf{det}}_k$	k-th output of deterministic encoder
$\mathcal{R}^{q,\mathsf{lat}}_k$	k-th output of latent encoder
$\bar{\mathcal{R}}^{q,\mathsf{lat}}$	Aggregated latent encoder output
$\mathcal{R}^\times$	Output matrix of cross-attention in deterministic path
E	Expectation operator
KL	Kullback-Liebler divergence operator

Table 2: List of mathematical symbols.

Acronym	Full Form
ANP	attentive neural process
ANP-BO	attentive neural process-based Bayesian optimization
BO	Bayesian optimization
CI	confidence interval
CoP	coefficient of performance
CVRMSE	coefficient of variation root-mean-squared error
E+/EPlus	Energy Plus
ELBO	evidence-based lower bound
FMU/FMI	functional mockup unit/interface
GP	Gaussian process
GP-BO	Gaussian process-based Bayesian optimization
HVAC	Heating, ventilation, and cooling
IECC	International Energy Conservation Code
IQR	interquartile range
KL	Kullback-Liebler
LHS	Latin hypercube sampling
MAE	mean absolute error
MCMC	Markov-chain Monte-Carlo
MHA	multi-head attention
MSE	Mean-squared error
PCA	principal components analysis
PICP	predictive interval coverage probability
RMSE	root-mean-squared error
RPBL	Residential Prototype Buildings Library
SGD	stochastic gradient descent
TARP	Thermal Analysis Research Program
US DOE	United States Department of Energy
VAE	variational autoencoder

Table 3: List of acronyms in alphabetical order.