Contents lists available at ScienceDirect

Energy & Buildings

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

Guidelines for the Bayesian calibration of building energy models[☆]

Adrian Chong[a,*], Kathrin Menberg[b]

[a] School of Design and Environment, Department of Building, National University of Singapore, 4 Architecture Drive 117566, Singapore
[b] Institute of Applied Geosciences, Karlsruhe Institute of Technology, Kaiserstrasse 12, 76131 Karlsruhe, Germany

ARTICLE INFO

Article history:
Received 8 March 2018
Revised 25 May 2018
Accepted 16 June 2018
Available online 4 July 2018

Keywords:
Bayesian calibration
Building energy modeling
Building simulation
Gaussian process
Hamiltonian Monte Carlo

ABSTRACT

This paper provides practical guidelines to the Bayesian calibration of building energy models using the probabilistic programming language Stan. While previous studies showed the applicability of the calibration method to building simulation, its practicality is still impeded by its complexity and the need to specify a whole range of information due to its Bayesian nature. We ease the reader into the practical application of Bayesian calibration to building energy models by providing the corresponding code and user guidelines with this paper.

Using a case study, we demonstrate the application of Kennedy and O'Hagan's (KOH) [1] Bayesian calibration framework to an EnergyPlus whole building energy model. The case study is used to analyze the sensitivity of the posterior distributions to the number of calibration parameters. The study also looks into the influence of prior specification on the resulting (1) posterior distributions; (2) calibrated predictions; and (3) model inadequacy that is revealed by a discrepancy between the observed data and the model predictions. Results from the case study suggest that over-parameterization can result in a significant loss of posterior precision. Additionally, using strong prior information for the calibration parameters may dominate any influence from the data leading to poor posterior inference of the calibration parameters. Lastly, this study shows that it may be misleading to assume that the posteriors of the calibration parameters are representative of their true values and their associated uncertainty simply because the calibrated predictions matches the measured output well.

1. Introduction

Building energy modeling (BEM), also known as building energy simulation, is the use of a physical-based building energy simulation program to model, predict and assess a building's energy consumption for heating, cooling, ventilation, lighting, and plug loads. Originally intended for use during the design phase, BEM is increasingly being used throughout the whole building's life-cycle [2]. The potential applications of BEM include retrofit analysis; supporting recommendations for energy conservation measures (ECMs) [3]; measurement & verification (M&V) [4–6]; as well as operations and controls [2]. In order to ensure model reliability and accuracy, calibration is an integral part of the overall modeling process, providing the modeler with increased confidence about the simulated outcomes. In general, calibration can be thought of as the process of tuning uncertain or unknown model parameters until model predictions matches measured data reasonably well.

Every building is unique and consists of complex systems interacting with one another. Consequently, BEMs typically contain various sources of uncertainty, including specification uncertainty, modeling uncertainty, numerical uncertainty, and scenario uncertainty [7]. Therefore, to achieve greater confidence in the modeling and calibration results, it is necessary that the calibration accounts for different sources of uncertainty. Recent years have seen an increasing application of Bayesian approaches for BEM calibration. This is because of its ability to naturally incorporate prior information (e.g. expert knowledge or information from drawings and specifications) and update this belief or uncertainty at different stages of the modeling procedure using measured data. Table 1 provides a summary of previous studies employing a Bayesian framework for BEM calibration.

In particular, there has been increasing interest in the Bayesian framework proposed by Kennedy and O'Hagan (KOH) [1] because it allows for different sources of uncertainty and accounts for any model inadequacy that is revealed by a discrepancy between the observations and the model predictions. Typically, the objective of model calibration is to minimize the difference between measured data and model predictions by tuning uncertain model parameters. In contrast, the idea behind KOH's Bayesian calibration framework is to model multiple sources of uncertainty, while retaining consistency with measured data.

[☆] Source code available on GitHub (https://github.com/adChong/bc-stan).
[*] Corresponding author.
E-mail address: bdgczma@nus.edu.sg (A. Chong).

https://doi.org/10.1016/j.enbuild.2018.06.028
0378-7788/© 2018 Elsevier B.V. All rights reserved.

Nomenclature

ASHRAE	American society of heating, refrigerating and air-conditioning engineers.
GP	Gaussian process.
GPR	Gaussian process regression.
HMC	Hamiltonian Monte Carlo.
BEM	Building energy model/modeling.
NUTS	No-U-Turn Sampler.
IPMVP	International Performance Measurement and Verification Protocol.
HMC	Hamiltonian Monte Carlo.
MCMC	Markov chain Monte Carlo.
CVRMSE	Coefficient of variation of the root mean squared error.
NMBE	Normalized mean biased error.
LR	Linear regression.
SVR	Support vector regression.
ECM	Energy conservation measure.
M&V	Measurement and verification.
KOH	Kennedy and O'Hagan [1].
$\hat{R}$	Gelman-Rubin statistic.
N	$= n + m$.
D	Dataset.
a, b	Parameters of probability distribution.
$pred$	Predictions.
x	Observable model inputs.
t	Calibration parameters.
p	Number of observable model input.
q	Number of calibration parameters.
n	Number of observed values.
m	Number of simulation data.
y	Observations.
z	$= [y_1, \ldots, y_n, \eta_1, \ldots, \eta_m]$.
f	Field/measured data.
c	Computer simulation data.
β	Correlation parameter.
ρ	$= \exp(-\beta/4)$.
λ	Precision parameter.
η	Simulator.
δ	Model discrepancy/bias/inadequacy.
ϵ	Observation errors.
Σ	Covariance matrix.
σ	Standard deviation.
μ^*	Absolute mean.
μ	Mean.
$\mathbb{R}$	Real numbers.
$\in$	A member of.

Early studies that applied the KOH framework to BEMs focused on its application to retrofit modeling and risk analysis, as well as introduced its ability to account for model inadequacy [13,14,26]. Since then, the framework has been applied to individual buildings [10,18], as well as to building clusters and building stock models at a district scale [9,17]. Additional focus was put on the performance of different emulators or surrogates under a Bayesian calibration framework, which showed that Gaussian process (GP) emulators provided the best accuracy but with the highest computational burden [19,20]. In order to alleviate the high computational costs of a Bayesian calibration framework, studies suggested using Hamiltonian Monte Carlo (HMC) for the Markov Chain Monte Carlo (MCMC) sampling [10,11,23] and using a representative subset of the data for the calibration [10].

However, to date, there are no practical guidelines on the application of KOH Bayesian calibration framework to BEM. Bayesian calibration requires the modeler to specify a whole range of information and inputs, which are not always readily available or intuitively obvious, but yet may have a significant impact on the calibration outcome and thus the accuracy and reliability of the resulting model. In particular, one of the main challenges of Bayesian calibration that has not been addressed in depth is the issue of posterior identifiability and the resulting precision of the posterior distribution of calibration parameters. In other words, different posteriors (with different interpretations about the calibration parameters) could result in similar model performance measures (e.g., different posteriors could produce "calibrated" models that satisfies the thresholds of CVRMSE $\leq 15\%$ and NMBE $\leq 5\%$ set by ASHRAE [5]). Therefore, the objectives of this paper are to:

1. Analyze the sensitivity of the calibration results to two critical user-defined inputs, which includes (1) the number of calibration parameters, and (2) the choice of priors for these calibration parameters.
2. Demonstrate (with the accompanying code) the Bayesian calibration process using an actual building case study that has the typical characteristics for BEM applications.

2. Bayesian calibration

2.1. Statistical formulation

Bayesian calibration is carried out following the statistical formulation proposed by Kennedy and O'Hagan [1]. The proposed approach uses Bayesian inference to model the relationship between field observations y and the output of the computer simulations η, by explicitly accounting for parameter uncertainties, model discrepancy, and observation error (Eq. 1).

$$y(x) = \eta(x, t^*) + \delta(x) + \epsilon \tag{1}$$

Table 1
Summary of publications using a Bayesian framework for BEM calibration.

Author(s)	Year	Calibration resolution	Emulator / Surrogate	Simulation tool	Reference
Booth et al.	2013	Annual	None	SUSDEM [8]	[9]
Chong et al.	2017	Hourly	GPR	TRNSYS; EnergyPlus	[10]
Chong and Lam	2017	Hourly	GPR	EnergyPlus	[11]
Heo et al.	2012, 2013, 2015	Monthly	GPR	ISO 13790 [12]	[13–15]
Kim and Park	2016	Monthly	GPR	EnergyPlus	[16]
Kristensen et al.	2017	Hourly	GPR	ISO 13790 [12]	[17]
Kristensen et al.	2017	Monthly; Weekly; Daily; 6-hourly	GPR	ISO 13790 [12]	[18]
Li et al.	2016	Monthly	GPR; LR	EnergyPlus	[19]
Lim and Zhai	2017	Monthly	GPR; LR; SVR	EnergyPlus	[20]
Manfren et al.	2013	Hourly	GPR	Energy signature model [21]	[22]
Menberg et al.	2017	Hourly	GPR	TRNSYS	[23]
Sokol et al.	2017	Monthly	None	EnergyPlus	[24]
Tian et al.	2016	Monthly	LR	EnergyPlus	[25]

GPR: Gaussian process regression, LR: Linear regression; SVR: Support vector regression.

Here t^* represents the true but unknown values of the calibration parameters t, suggesting that the simulation model is a biased representation of the actual physical building system even in the ideal situation where $t = t^*$. Therefore, a discrepancy term $\delta(x)$ is included to account for this model inadequacy. $\delta(x)$ is also commonly known as model bias or model inadequacy, and it is a structured error term that can assume different values over the range of chosen contour states. Presumably, the separate consideration of parameter and model discrepancy could alleviate the possibility of calibration parameters subsuming variations that might stem from an inadequacy of the simulation model in representing the actual physical system.

2.2. Gaussian process (GP) surrogate/emulator

Using BEM during the iterative calibration process is often computationally expensive. Therefore, to reduce computation costs, a surrogate or emulator is used to map the inputs of the model to the output of interest. We specify GP models because of their convenience, flexibility and fairly broad generality [27,28]. Following Hidgon et al. [27], GP models are used to combine the field observed data D^f and the computer simulation data D^c. Separate GP models are used to emulate the simulator $\eta(x, t)$ and the discrepancy term $\delta(x)$ in Eq. (1). Therefore, the GP models consider two kinds of inputs:

1. $x = x_1, x_2, \ldots, x_p$ denotes the inputs to the simulation model that are observable or measurable, such as outdoor dry-bulb temperature and relative humidity; and
2. $t = t_1, t_2, \ldots, t_q$ denotes the calibration parameters of the simulation model. These t are the model parameters that need to be estimated (e.g. infiltration rate or boiler efficiency).

To specify a GP model, a mean function and a covariance function need to be defined. The GP model for the simulator $\eta(x, t)$ is defined with a mean function that returns a zero vector and a covariance function Σ_η that depends on the observable inputs x and the calibration parameters t (Eq. 2).

$$\Sigma_{\eta,ij} = \frac{1}{\lambda_\eta} exp\left\{ -\sum_{k=1}^{p} \beta_k^\eta |x_{ik} - x_{jk}|^\alpha - \sum_{k'=1}^{q} \beta_{p+k'}^\eta |t_{ik'} - t_{jk'}|^\alpha \right\} \quad (2)$$

where,

λ_η is the precision hyperparameter of this GP model,

$\beta_1^\eta, \ldots \beta_{p+q}^\eta$ are the correlation hyperparameters of this GP model,

p is the number of input factors x,

q is the number of calibration parameters t.

Likewise, the GP model for the discrepancy term $\delta(x)$ is defined with a mean function that returns a zero vector, and a covariance function Σ_δ that depends on the observable inputs x (Eq. 3).

$$\Sigma_{\delta,ij} = \frac{1}{\lambda_\delta} exp\left\{ -\sum_{k=1}^{p} \beta_k^\delta |x_{ik} - x_{jk}|^\alpha \right\} \quad (3)$$

where,

λ_δ is the precision hyperparameter of this GP model,

$\beta_1^\delta, \ldots, \beta_p^\delta$ are the correlation hyperparameters of this GP model,

p is the number of input factors x.

To establish a statistical relationship between the observations y and the model predictions η, the observations $y_1, \ldots, y_n \in \mathbb{R}$ and predictions $\eta_1, \ldots, \eta_m \in \mathbb{R}$ are combined in a single $n + m$ vector $z = [y_1, \ldots, y_n, \eta_1, \ldots, \eta_m]$ [27]. The resulting likelihood function is then given by Eq. (4) below.

$$\mathcal{L}(z \mid t, \beta^\eta, \lambda_\eta, \beta^\delta, \lambda_\delta, \lambda_\epsilon) \propto |\Sigma_z|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2}(z-\mu)^T \Sigma_z^{-1} (z-\mu) \right\} \quad (4)$$

where,

$$\Sigma_z = \Sigma_\eta + \begin{bmatrix} \Sigma_\delta + \Sigma_y & 0 \\ 0 & 0 \end{bmatrix}, \ \Sigma_y = I_n/\lambda_\epsilon \quad (5)$$

Σ_η is a $(n + m) \times (n + m)$ matrix computed based on Eq. (2),

Σ_δ is a $n \times n$ matrix computed based on Eq. (3),

Σ_y is the $n \times n$ covariance matrix used to account for observation errors and is given by I_n/λ_ϵ.

Therefore, the resulting joint posterior probability density depends on the unknown calibration parameters $t_1^f, \ldots, t_q^f$, the GP correlation hyperparameters ($\beta_1^\eta, \ldots, \beta_{p+q}^\eta$, $\beta_1^\delta, \ldots, \beta_p^\delta$), and the GP precision hyperparameters (λ_η, λ_δ, λ_ϵ). Correspondingly, prior probability distributions also need to be defined for all calibration parameters and GP hyperparameters. The priors used are described in greater detail in the Sections 3.1 and 3.4.

2.3. No-U-Turn Sampler (NUTS) HMC algorithm

The non-linearity of the simulator (or emulator) makes it hard to analytically sample from the high-dimensional posterior distribution. Therefore, a common approach is to use MCMC methods to explore the posterior distribution. We use the No-U-Turn Sampler (NUTS), an extension of the Hamiltonian Monte Carlo (HMC) for the MCMC sampling.

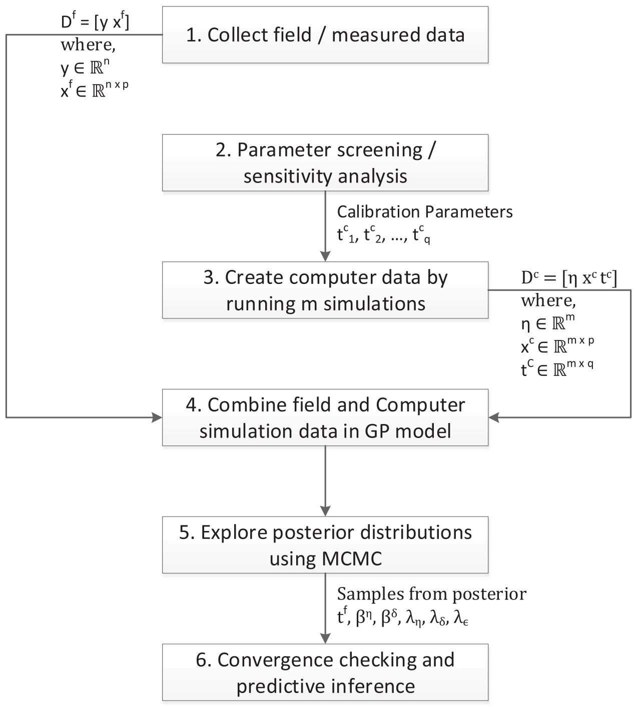
HMC is a specific algorithm that avoids the random walk behavior that afflicts many MCMC algorithms by using first-order gradient information to guide the MCMC sampling process, thus allowing it to achieve faster convergence to high-dimensional posterior distributions [29,30]. Therefore, HMC is suitable for the KOH framework since the posterior distribution is typically high-dimensional consisting of the calibration parameters t and the GP hyperparameters $\beta_1^\eta, \ldots, \beta_{p+q}^\eta$, $\beta_1^\delta, \ldots, \beta_p^\delta$, λ_η, λ_δ, λ_ϵ. However, HMC requires the tuning of two sampling parameters (a scaling factor ϵ and the number of leapfrog steps L), and poor choices of either can result in an ineffective implementation of HMC. To overcome the challenges of tuning, Hoffman and Gelman [31] developed the NUTS algorithm that optimizes HMC adaptively and provides automated tuning of ϵ and L, thus making it possible to run the NUTS algorithm without any user intervention. For a more in-depth understanding of the concepts of HMC and why it performs well on complex problems, the interested reader may refer to the intuitive introductions by Betancourt [32].

3. Evaluation of calibration settings

Using a case study, we evaluate the sensitivity of the posterior distributions of the calibration parameters to the number of model parameters used for the calibration, and investigate the influence of different choices of prior distributions on the calibration results, including their effect on the posterior distributions of the calibration parameters, model bias, as well as agreement between calibrated predictions and the measured data. The Bayesian calibration is carried out using the code described in Section 4.

Fig. 1 shows an overview of the calibration framework applied to the case study and can be summarized as:

1. Collect data for the observable output $y(x^f)$ and observable inputs x^f. Using the measured data, create a field dataset $D^f = [y, x_1^f, \ldots, x_p^f]$ (illustrated in Table 4).
2. Identify calibration parameters $t_1, \ldots, t_q$ using parameter screening with Morris method [33].



D^f = [y x^f] where, y ∈ ℝ^n x^f ∈ ℝ^{n x p}
1. Collect field / measured data
2. Parameter screening / sensitivity analysis
Calibration Parameters t^c_1, t^c_2, ..., t^c_q
3. Create computer data by running m simulations
D^c = [η x^c t^c] where, η ∈ ℝ^m x^c ∈ ℝ^{m x p} t^c ∈ ℝ^{m x q}
4. Combine field and Computer simulation data in GP model
5. Explore posterior distributions using MCMC
Samples from posterior t^f, β^η, β^δ, λ_η, λ_δ, λ_ε
6. Convergence checking and predictive inference

Fig. 1. Bayesian calibration procedure, based on Heo et al. [13] and adapted from Chong et al. [10].

3. Run m simulations at the same observable input factors $x^c = x^f$. Using the simulation predictions $\eta(x^c,\ t^c)$, create a computer simulation dataset $D^c = [\eta, x^c_1, \ldots, x^c_p, t^c_1, \ldots, t^c_q]$ (illustrated in Table 5).
4. Combine D^f and D^c in a GP model using the approach described in Higdon et al. [27].
5. Explore the joint posterior distribution using MCMC.
6. Check for convergence and assess accuracy of the calibrated model (predictive inference).

3.1. Case study description

The case study building is the site entrance to the National Renewable Energy Laboratory (NREL) building located in Colorado, U.S.A (Fig. 2). The building is an existing 850 ft^2 one story building that was built in 1994, and it is occupied 24 hours a day by one to four people. Space conditioning is provided by a split system direct expansion (DX) cooler with a natural gas furnace. 18 months (July 2012 to December 2013) of measured monthly electricity energy consumption data was used for this study. Two-thirds of this data (12 months) was used as a training dataset for calibrating the model, while the remaining 6 months were used as a hold-out test dataset. Since the aim of this paper is the demonstration of the Bayesian calibration process, we focus only on electricity consumption and not natural gas because electricity is used throughout the whole year, while natural gas consumption only occurs from October to April.

Actual Meteorological Year (AMY) hourly weather data is used for the calibration period being considered. Data (field measurements, AMY weather file, OpenStudio BEM, and information on the actual building, parameters, etc.) for this case study can be found on NREL's GitHub Repository[1]. For the purpose of this paper, the OpenStudio model was converted to an EnergyPlus model. The observable inputs x used in this case study are the outdoor dry-bulb temperature, the outdoor relative humidity, and the direct solar radiation rate per unit area.

It is important to be transparent and explicit about every choice and assumption used for the calibration [34]. For this case study, the following prior distributions were defined for the GP hyperparameters:

- $\rho^\eta_1, \ldots, \rho^\eta_{p+q} \sim Beta(a = 1, b = 0.3)$: These correlation hyperparameters are reparameterization of $\beta^\eta_1, \ldots, \beta^\eta_{p+q}$ using $\rho^\eta_i = \exp(-\beta^\eta_i/4), i = 1, \ldots, (p+q)$. Considering that $\beta^\eta_i > 0$, the reparameterization places ρ^η_i in the range [0,1]. Therefore, the *Beta* distribution makes a suitable prior, since it is a family of continuous distributions defined on the interval [0,1]. Choosing $a = 1$ and $0 < b < 1$ places most of the prior mass close to 1, reflecting our expectation that only a subset of the inputs will influence the simulator output. For this case study, we use $b = 0.3$, which means $P(\rho^\eta_i < 0.9) \approx \frac{1}{2}$.
- $\rho^\delta_1, \ldots, \rho^\delta_p \sim Beta(a = 1, b = 0.3)$: For similar reasons and because we expect a small subset of the inputs to have an effect on the discrepancy term $\delta(x)$, we also specify independent $Beta(1,\ 0.3)$ distributions for the correlation hyperparameters $\delta(x)$.
- $\lambda_\eta \sim Gamma(a = 10, b = 10)$: a represents the shape, while b is the rate parameter. Since the outputs were standardized to have unit variance ($\sigma^2_\eta = 1$), we expect the precision parameter ($\lambda_\eta = \frac{1}{\sigma^2_\eta}$) to also have a value close to one. By specifying $a = b = 10$, $\mathbb{E}(\lambda_\eta) = \frac{a}{b} = 1$.
- $\lambda_\delta \sim Gamma(a = 10, b = 0.3)$: $\mathbb{E}(\lambda_\delta) \approx 33$, which gives a variance of approximately 3%. This is because we expect a small model bias $\delta(x)$ (based on initial analysis of the BEM) and also to help mitigate possible confounding issues (i.e., the discrepancy term subsumes the variation that should be used to tune the calibration parameter t^f).
- $\lambda_\epsilon \sim Gamma(a = 10, b = 0.03)$: $\mathbb{E}(\lambda_\epsilon) \approx 333$, which gives a variance of 0.3% because we expect observation errors to be even smaller than the discrepancy term $\delta(x)$.

The choice of relatively strong priors for the precision hyperparameters is driven by the need to distinguish between the bias term and the observation errors and the belief that the magnitude of the variances should be in the order $\sigma^2_\epsilon < \sigma^2_\delta < \sigma^2_\eta$. A detailed discussion about the impact of (un)suitable priors for the discrepancy term on the overall calibration results is given by Brynjarsdottir and O'Hagan [35], which showed that a very specific prior is required to correctly identify the model bias of complex models. On the other hand, setting very specific priors might limit the range of values that can be explored during the MCMC sampling. However, the priors for the hyperparameters are not hard constraints, i.e., they are not bounded distributions. Therefore, the resulting posterior can still be significantly different from the priors given strong evidence from the measured data. This was shown in Menberg et al. [36] with respect to the precision hyperparameters.

3.2. Parameter screening

The first step to calibrating a building energy model is to conduct a sensitivity analysis and uncertainty quantification to identify the most influential uncertain model parameters and their ranges. This is a crucial step for BEM calibration to mitigate possible non-identifiability that could result when the calibration parameters are not uniquely identifiable (i.e., different parameter sets result in similar model performance measure), and thus impeding the reliability of the calibration. Selecting the correct number of model parameters for the calibration is important because if the amount of measured data is insufficient to identify the calibration parameters, the resulting posterior distribution may be subjected to an

[1] https://github.com/NREL/OpenStudio-analysis-spreadsheet/tree/develop/Calibration_example

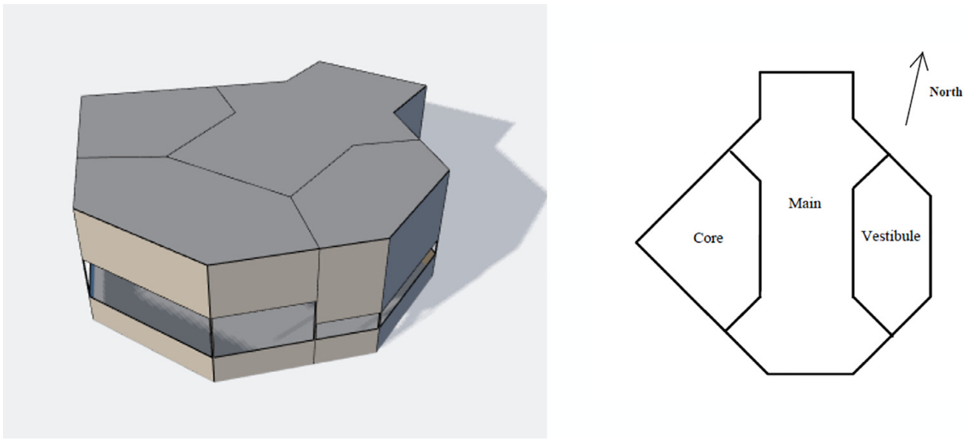
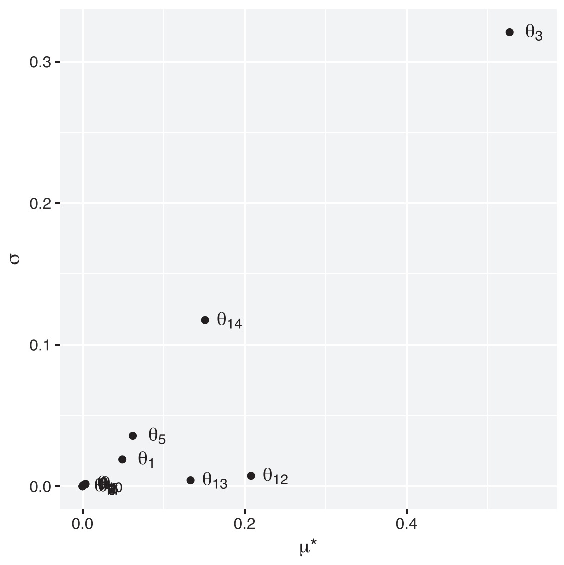


Fig. 2. 3D (left) and plan (right) views the of the building energy model.



Symbol	Parameter
θ_1	Cooling Setpoint
θ_2	Heating Setpoint
θ_3	AHU fan efficiency
θ_4	AHU fan motor efficiency
θ_5	DX cooling COP
θ_6	Gas burner efficiency
θ_7	Baseboard nominal capacity
θ_8	Exterior wall R-value
θ_9	Exterior wall density
θ_{10}	Roof R-value
θ_{11}	Roof density
θ_{12}	Electric equipment power density
θ_{13}	Lighting power density
θ_{14}	Space infiltration

Fig. 3. Results from Morris method.

unacceptable degree of uncertainty. Additionally, using fewer parameters would also reduce computation cost given that Bayesian calibration is computationally prohibitive in a high-dimensional parameter space.

The uncertain parameters in the model and their ranges are listed in Table A.6. To realistically reflect cases where no prior knowledge is available, broad ranges were assigned to the parameters (e.g. AHU fan efficiency and power densities). The Morris method [33] is used to select the influential uncertain model parameters because it has been shown to be a good compromise between accuracy and computational effort in the context of building energy models [37,38]. Fig. 3 shows the result of the sensitivity analysis, using the modified mean μ^* proposed by Campolongo et al. [39] and standard deviation σ to help screen-out non-sensitive parameters. To provide a wider exposition of potential identifiability issues, the uncertain model parameters θ were ranked using the modified mean μ^* (Table 2) and Bayesian calibration using Stan (listing 3) was carried out with varying number of calibration parameters.

To create the calibration input data file D^c, i.e. the training dataset for the Gaussian process emulator, $m = 30$ simulations were run at different parameter values. Parameter values were determined using Latin hypercube sampling (LHS). This means that we try to cover as much space as possible in the multi-dimensional space of the calibration parameters with $m = 30$ computer simulation runs. A rule of thumb for training GP models is to have 10 LHS samples per parameter [40], although some studies showed that fewer well-chosen samples also led to reasonable results [10,13,23].

Table 2
Ranked list of model parameters by their sensitivity.

Symbol	Parameter	μ^*
θ_3	AHU fan efficiency	0.53
θ_{12}	Electric equipment power density	0.21
θ_{14}	Space infiltration	0.15
θ_{13}	Lighting power density	0.13
θ_5	DX cooling COP	0.06
θ_1	Cooling setpoint	0.05
θ_{10}	Roof R-value	0.00
θ_8	Exterior wall R-value	0.00
θ_2	Heating setpoint	0.00
θ_9	Exterior wall density	0.00
θ_{11}	Roof density	0.00
θ_7	Baseboard nominal capacity	0.00
θ_4	AHU fan motor efficiency^2	0.00
θ_6	Gas burner efficiency	0.00

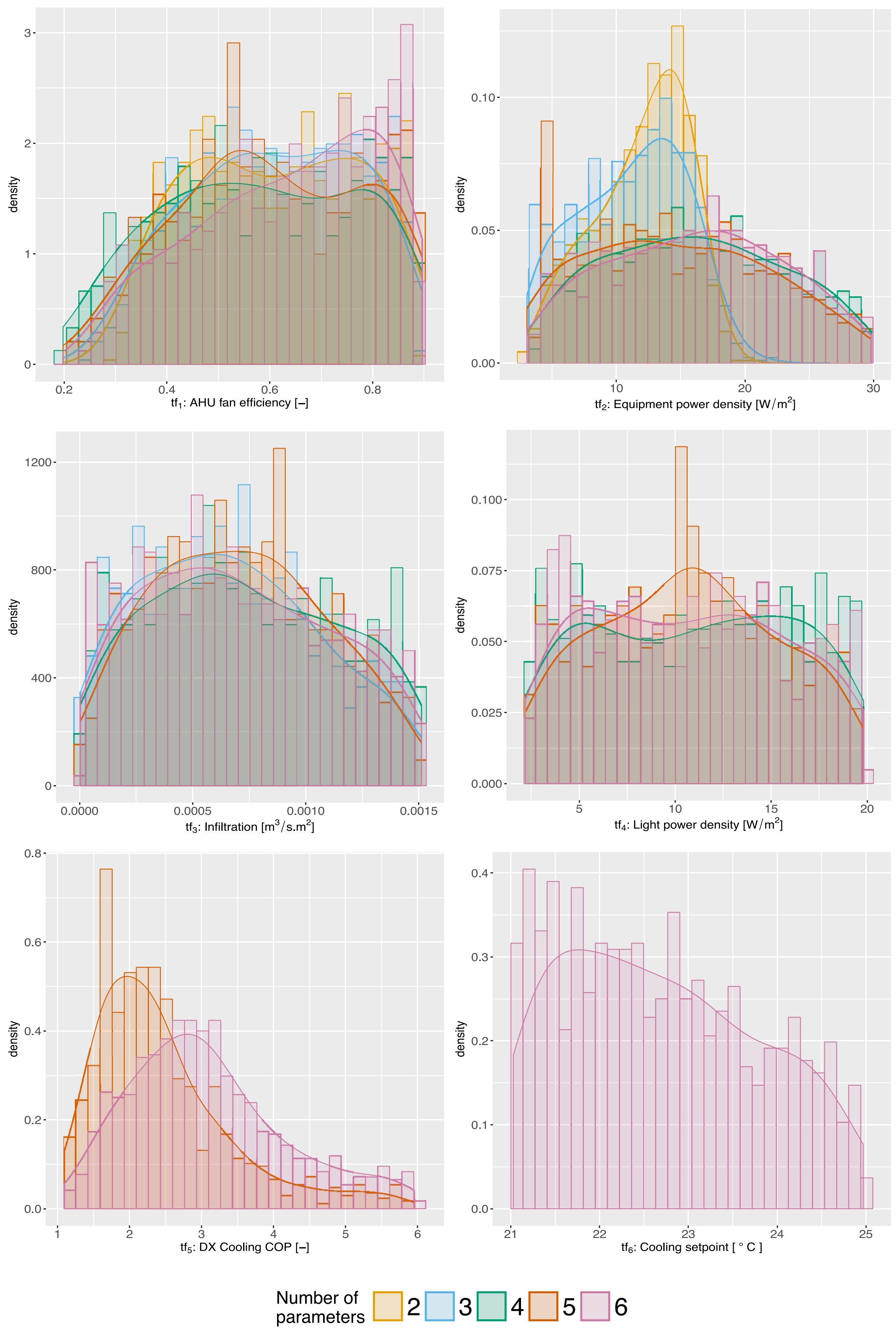


Fig. 4. Effect of number of calibration parameters on posterior distribution.

3.3. *Posterior parameter identifiability*

Fig. 4 shows the resulting posterior distributions when 2, 3, 4, 5 or 6 calibration parameters were used for the Bayesian calibration respectively. The parameters were selected based on their ranking, i.e., if three parameters were used for the calibration, they are the three parameters with the highest μ^* values (Table 2).

In general, Fig. 4 shows that over-parameterization occurs in our case study when four or more parameters are used for the calibration, which is indicated by an increase in posterior uncertainty, when more parameters are included in the calibration. In particular, the posterior distributions for equipment power density t_2^f (top right plot of Fig. 4) shows that with two or three calibration parameters, the mean posterior is approximately 12 W/m^2 with an approximate range of between 5 W/m^2 and 18 W/m^2. However, with four or more calibration parameters, the posterior distribution appears uniformly distributed between 5 W/m^2 and 30 W/m^2, indicating a significant reduction in identifiability due to over-parameterization. In other words, as more model parameters are used for the calibration there is a loss of posterior precision because more parameter combinations are now likely given the measured data, and the data is insufficient to infer these calibration parameters more precisely.

It can also be observed that although an increase in posterior uncertainty was observed for equipment power density t_2^f, there was no noticeable increase for AHU fan efficiency t_1^f, which has been shown to be the most sensitive parameter. One possible explanation is because the posterior uncertainty with two calibration parameters already covers the entire range of the prior set for t_1^f ([0.1, 0.9]). This indicates that the measured data used is non-informative about the calibration parameters because the data does not contain any information with respect to AHU efficiency for this particular case study. As a result, it seems that the posterior uncertainty does not increase when the number of calibration parameters was increased.

It is important to note that having too little calibration parameters might also result in over-fitting and an unreasonably tight posterior distribution that no longer covers the true parameter value. While the issue of correct and reliable posterior inference of calibration parameters has been examined before in the context of BEM [15,23], the case study here presents an example on the potential magnitude of the effects caused by over-parameterization and/or over-fitting. It is also important to keep in mind that the posteriors of the hyper-parameters of the Gaussian process are inherently linked to the model parameters through the joint multivariate distribution so that those parameters are prone to exhibit similar effects with an increasing number of model calibration parameters. Thus, while over-parameterization and over-fitting are ubiquitous issues in model calibration, they should be carefully considered in the context of Bayesian calibration.

3.4. Choice of priors for model parameters

Given measured data, a likelihood function, and properly defined prior distributions, the KOH framework will produce a well-calibrated model, which allows inference about true calibration parameters, and insights into the magnitude of different error terms. However, the quality of this calibration depends entirely on the quality of the inputs. Consequently, an important step in Bayesian calibration is choosing priors for the calibration parameters.

In an ideal case, the modeler has access to detailed building or case study specific information from building reports, site visits or databases, and the choice of a very precise prior distribution is uncontroversial. At other times, the modeler could construct informative priors by interviewing experts or using information from specifications and past case studies [41]. More commonly, *ad hoc* prior distributions are chosen based on default parameter values and ranges in the hope that they represents reasonable priors that lead to sensible results. It is worth mentioning that when modeling a real building, defects (such as a typical performance gap issue of substituted materials or poor detailing leading to air circulation within the fabric) may also lead to divergence between model and data that are so large that they are beyond the reasonable formulation of the priors within the KOH framework.

Priors as currently implemented in previous studies have never been evaluated on how they may affect the quality of the calibration beyond the fit of model results to measured data. Using the case study, we investigate the impact of three different levels of prior distributions (non-informative, weakly informative, specific informative) on the posterior results of the calibration. The previous section showed that over-parameterization results from using four or more calibration parameters, giving rise to poor posterior identifiability. Consequently, only the top 3 most influential parameters (Table 2) will be considered here, and they are the AHU fan efficiency t_1^f; electric equipment power density t_2^f; and space infiltration t_3^f. The three levels of priors are:

- Flat (non-informative) priors: uniform priors. $t_1^f, t_2^f, t_3^f \sim \mathcal{U}(0, 1)$.
- Specific informative priors: normal priors with a standard deviation of 0.05. $t_1^f \sim \mathcal{N}(0.5, 0.05)$, $t_2^f \sim \mathcal{N}(0.7, 0.05)$, and $t_3^f \sim \mathcal{N}(0.14, 0.05)$.
- Weakly informative priors: normal priors with a standard deviation of 0.2. $t_1^f \sim \mathcal{N}(0.5, 0.2)$; $t_2^f \sim \mathcal{N}(0.7, 0.2)$; and $t_3^f \sim \mathcal{N}(0.14, 0.2)$.

It is important to note that these prior probability distributions are defined with respect to the normalized range [0, 1] of the calibration parameters t^f. The upper and lower bounds defined for these parameters represent their practical constraints (e.g. fan efficiency t_1^f would not have a value less than 0 or greater than 1). In addition, the normal distribution is used for weakly and specific informative priors because it is unbounded and therefore provides support on all real numbers within the normalized range [0, 1]. The mean parameter of the normal priors are then assigned based on the expected value of the calibration parameters. Since the mean of the normal distribution is its expected value, we assign a mean to the calibration parameters that we believe to be its true value based on available information (i.e., their initial values as shown in Table A.6).

Fig. 5 shows the resulting posterior distributions for each calibration parameter. The corresponding prior for each posterior is indicated using dashed lines of the same color. From Fig. 5, it can be observed that with flat priors and the chosen bounds, the resulting posterior is also relatively uninformative. Consequently, the posterior (solid yellow density plot) is indicative of a posterior that is primarily driven by the measured data. In contrast, using very specific informative priors leads to posteriors that are highly constrained, suggesting that the posterior is driven primarily by the prior, dominating any influence from the data. However, weakly informative priors show a compromise between using flat and highly informative priors (Fig. 5). Additionally, the shift in the posterior from the prior, also suggests that the resulting posterior is influenced by the priors but not so strongly so as to rule out values that might make sense according to the data.

While these observations depend on both the likelihood and the prior, they also highlight the potential impact on the resulting posterior. In particular, it can be observed that t_2^f converges towards different posterior mode values with different levels of priors. Given that the posterior with flat and weakly informative priors have very similar modes, it seems likely that the specific informative prior used might be too strong that sensible parameter values could have been excluded.

Fig. 6 shows the resulting box plots for the calibrated predictions $\eta(x, t) + \delta(x)$ (left plot), the model predictions $\eta(x, t)$ (middle plot), and the distribution of the normalized posterior model discrepancy $\frac{\delta(x)}{y(x)}$ (right plot) when different priors are used for the calibration parameters. It can be observed (for both training and test dataset) that the model tends to overestimate building energy consumption when specific informative priors are used (middle plot of Fig. 6). Nonetheless, this overestimation is well adjusted by a discrepancy term $\delta(x)$ that is more "negative" (right

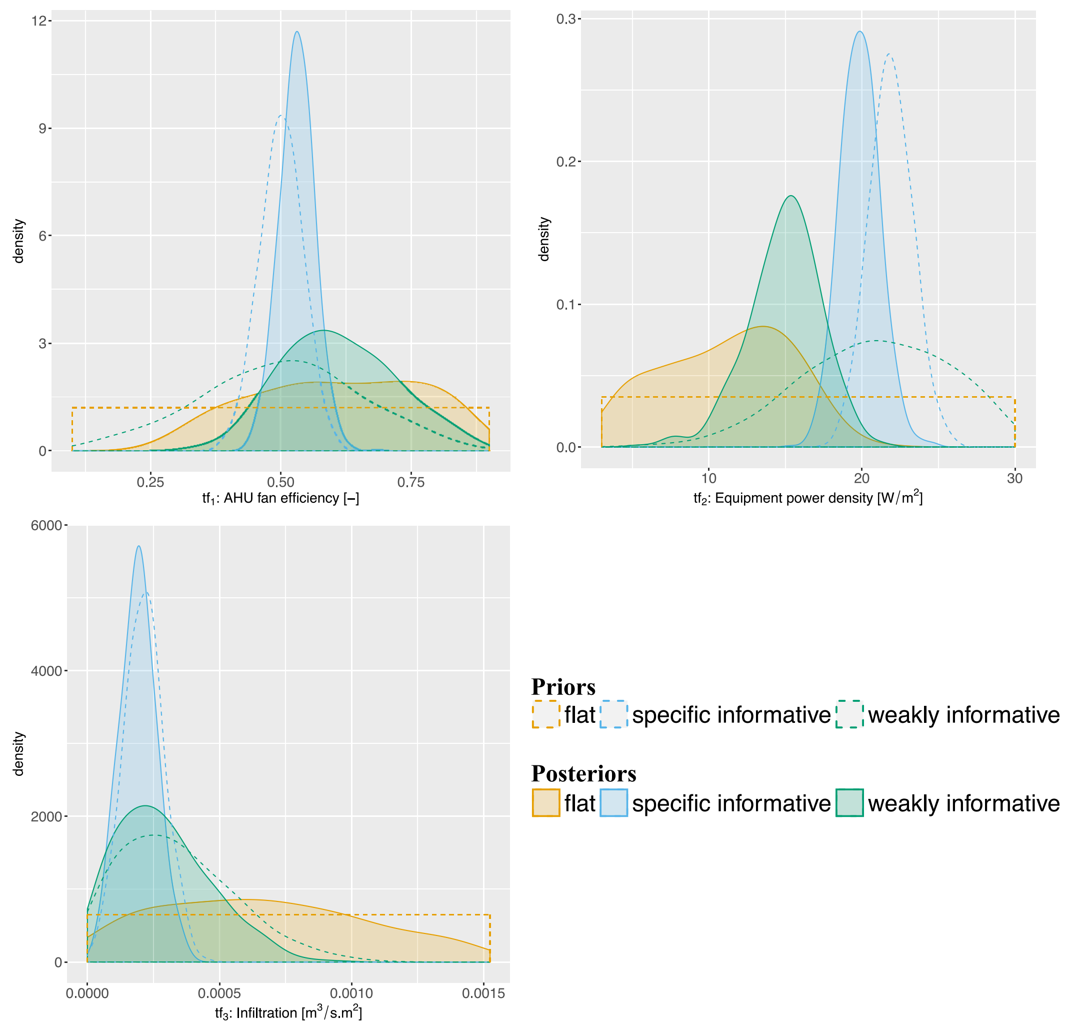


Fig. 5. Comparison of posterior distributions with 3 different levels of priors (flat: $\mathcal{U}(0, 1)$; specific informative: $\mathcal{N}(0, 0.05)$; and weakly informative; $\mathcal{N}(0, 0.2)$. Corresponding priors are indicated using a dashed density plot of the same color.

Table 3
CVRMSE and NMBE (to 2 decimal places) for training and test dataset using different choice of prior distributions for the calibration parameters t^f.

	Training dataset		Test dataset	
Prior distribution	CVRMSE [%]	NMBE [%]	CVRMSE [%]	NMBE [%]
Flat	1.51	0.00	6.38	-4.04
Specific informative	2.16	0.23	6.62	-4.81
Weakly informative	1.52	0.04	6.59	-4.35

Table 4
Field data $D^f \in \mathbb{R}^{n\times(1+p)}$ as they appear in a data frame.

y	$x1$	$x2$	$\cdots$	xp
y_1	$x1_1$	$x2_1$	$\cdots$	xp_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
y_n	$x1_n$	$x2_n$	$\cdots$	xp_n

Table 5
Simulation data $D^c \in \mathbb{R}^{m\times(1+p+q)}$ as they appear in a data frame .

η	$x1$	$x2$	$\cdots$	xp	$t1$	$t2$	$\cdots$	tq
η_1	$x1_1$	$x2_1$	$\cdots$	xp_1	$t1_1$	$t2_1$	$\cdots$	tq_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
η_m	$x1_m$	$x2_m$	$\cdots$	xp_m	$t1_m$	$t2_m$	$\cdots$	tq_m

plot of Fig. 6), producing calibrated predictions that are very similar to predictions when less informative (flat and weakly informative) priors are used for the calibration parameters (left plot of Fig. 6). This observation is further reinforced by a consistently low CVRMSE and NMBE across different levels of priors (Table 3). In particular, with respect to the training dataset, all three models would be considered very well calibrated according to current standards such as ASHRAE Guideline 14 [5] and the IPMVP [4] given the small deviation between the predicted outcome and the measured data. As might be expected, predictions for the test dataset perform worse than predictions for the training dataset (higher CVRMSE and NMBE values) since the model was calibrated against the training data. Nonetheless, CVRMSE and NMBE for the test dataset still meet the requirements set by ASHRAE Guideline 14 [5] and the IPMVP [4]. This is despite the resulting posterior distributions with specific informative priors being significantly different (Fig. 5).

These results suggest that the discrepancy term $\delta(x)$ as modeled using a GP model is able to absorb the discrepancy that is caused by using stricter priors for the calibration parameters. Consequently, even though the error between the calibrated predictions $\eta(x, t) + \delta(x)$ and the observations $y(x)$ are small, it might be misleading to interpret the posteriors of the calibration parameters as though they are representative of the true parameter values. Additionally, the analysis shows that different parameter sets can provide CVRMSEs and NMBEs that are equally satisfactory based on

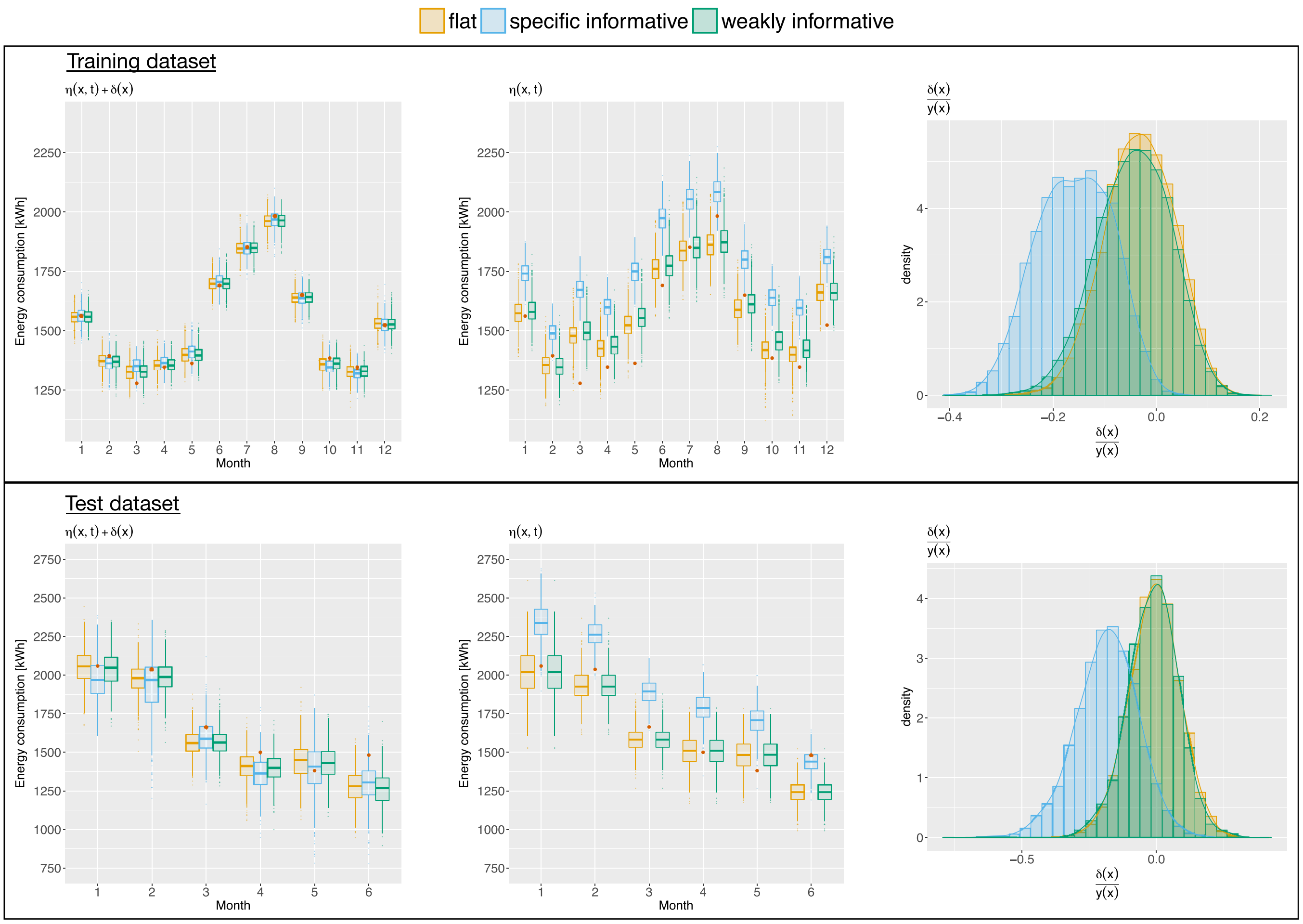


Fig. 6. Comparison of posterior predictions $\eta(x,t)+\delta(x)$, predictions without discrepancy term $\eta(x,\, t)$, and model discrepancy normalized by measured data $\frac{\delta(x)}{y(x)}$, with 3 different levels of priors (flat: $\mathcal{U}(0,1)$; specific informative: $\mathcal{N}(0,0.05)$; and weakly informative; $\mathcal{N}(0,0.2)$ for training and test dataset.

well-established standards [4,5], which is indicative of a good fit between the calibrated predictions and the observations but does not mean the posteriors are good estimates of the true parameter values and its associated uncertainty.

Based on these observations, it is therefore recommended that weakly informative rather than highly constrained specific informative priors be used for the calibration parameters, unless there is strong supporting evidence or belief that the parameter should be close to a particular value. The idea is that the loss in precision by using a less informative prior is less serious than the gain in robustness by including parts of the parameter space that might be relevant [42]. As was shown in Section 3.3, where over-parameterization led to lower posterior precision, lower precision might indeed reflect reality in some cases. It is therefore important that a good balance be found between model prediction accuracy and an acceptable level of posterior uncertainty.

It can also be observed from Fig. 6 and Table 3 that the error between the predicted outcome and the measured data decreases with less prior information. In other words, a lower CVRMSE and NMBE is observed when the priors have larger variance. This is expected because given a prior with larger variance, the posterior is largely influenced by the data, leading to a better fit between measurements and predictions. However, this defeats the purpose of using a Bayesian approach if some prior information about the system is known and presumably result in a model that might be “over-calibrated”.

Therefore, to summarize, the priors assigned are case dependent and should match the genuine state of prior knowledge of the system being modeled. In cases where there is no prior knowledge, a non-informative or flat prior should be used. Similarly, where there is prior knowledge, the use of weakly informative rather than specific informative prior is recommended.

3.5. Model convergence

Trace plots of multiple chains and the Gelman-Rubin statistic $\hat{R}$ are used to check for adequate convergence. Fig. 7 shows an example of the trace plots for the calibration parameters ($t_1^f, \ldots, t_3^f$), GP correlation hyperparamters ($\beta_1^\eta, \ldots, \beta_6^\eta, \beta_1^\delta, \ldots, \beta_3^\delta$), and GP precision hyperparameters (λ_η, λ_δ, λ_ϵ), when three calibration parameters with weakly informative priors were used for the calibration. The first 250 iterations (50%) were discarded as warm-up (burn-in) to reduce the influence of starting values. A visual inspection suggests that after 250 iterations (warm-up period), all components of the posterior distribution ($t_1^f, \ldots, t_3^f, \beta_1^\eta, \ldots, \beta_6^\eta, \beta_1^\delta, \ldots, \beta_3^\delta, \lambda_\eta, \lambda_\delta, \lambda_\epsilon$) are well mixed and have converged to a common stationary distribution. As mentioned in Section 4.4, we are looking for trace plots that look like a “fat, hairy caterpillar” that does not bend. $\hat{R}$ is also within 1 ± 0.1 for all calibration parameters and hyperparameters of the GP model.

4. Code for Bayesian calibration

This section walks through the code for the application of the Bayesian calibration framework, showing how to set up, run and visualize the results. We use the probabilistic programming lan-

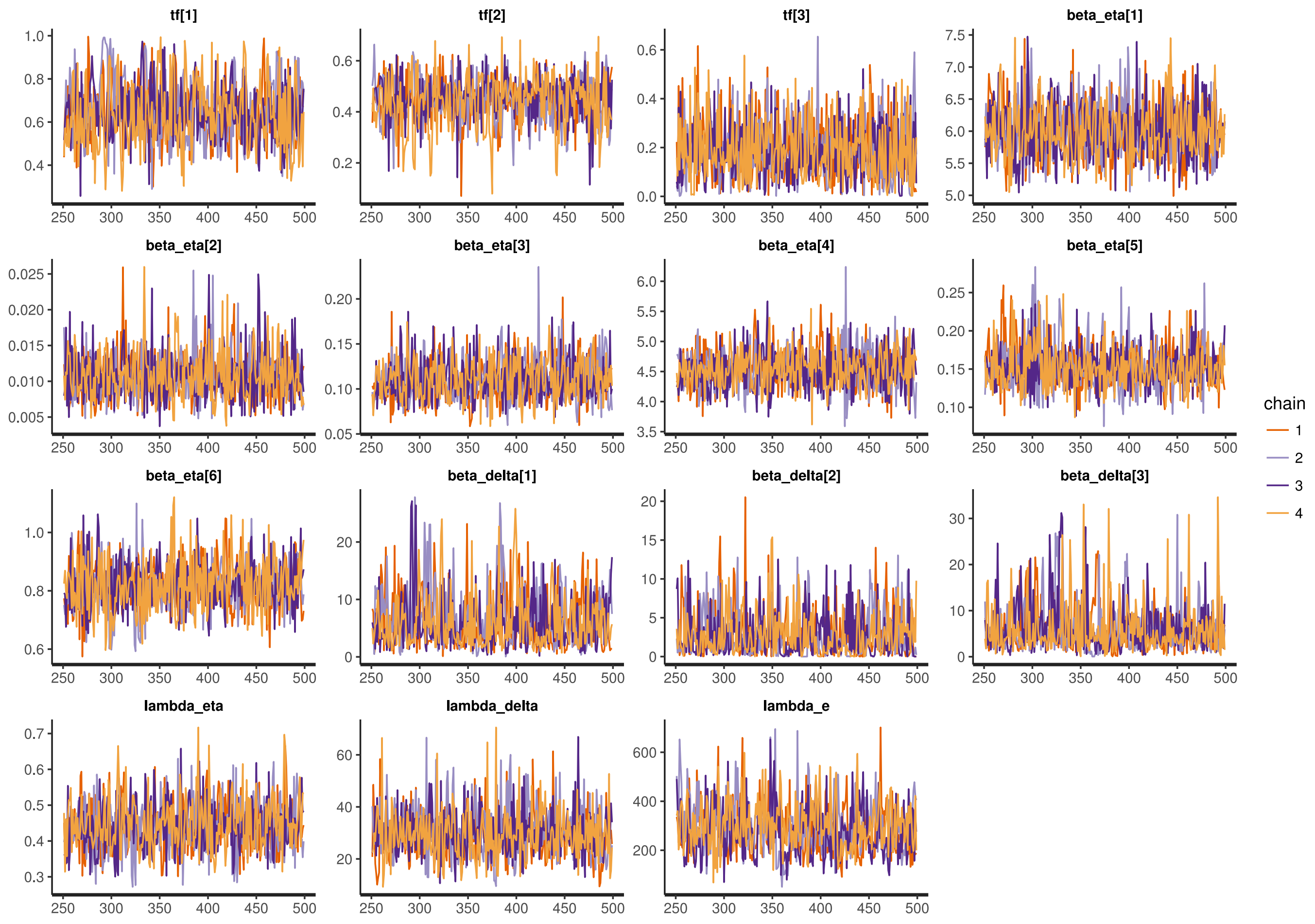


Fig. 7. Trace plots of the calibration parameters t^f, GP correlation hyperparameters (β^η and β^δ), and GP precision hyperparameters (λ_η, λ_δ and λ_ϵ). Different colors indicate different MCMC chains.

guage Stan because programs written in Stan can interface with many different environments, including R, Python, Matlab, Julia, Stata, Mathematical, and command-line terminal. While the calibration is demonstrated through a R interface [43] in this paper, the provided Stan codes are easily portable across any of the above-mentioned environments. The code provided includes[2]:

- R code for setting up the data, running the Stan model, and extracting the results of the calibration (Appendix B).
- Stan code for KOH framework (Appendix C).
- Stan code for KOH framework with predictive inference at new input settings (Appendix D).

4.1. Setting up the data

Suppose the two datasets D^f and D^c are saved as `DATAFIELD.csv` and `DATACOMP.csv` respectively, where

- D^f illustrated in Table 4 denotes the field dataset and comprises a series of n observed output $y_1, y_2, \ldots, y_n \in \mathbb{R}$ paired with corresponding observed inputs $x_1, x_2, \ldots, x_n \in \mathbb{R}^p$; and
- D^c illustrated in Table 5 denotes the computer simulation dataset and comprises a series of m simulated outputs $\eta_1, \eta_2, \ldots, \eta_m \in \mathbb{R}$ paired with corresponding inputs $x_1, x_2, \ldots, x_nm \in \mathbb{R}^p$ and calibration parameters $t_1, t_2, \ldots, t_nm \in \mathbb{R}^q$.

[2] Source code available on GitHub (https://github.com/adChong/bc-stan)

First, the data is read into R (listing 1, lines 4–5) and various parts of D^f (Table 4) and D^c (Table 5) extracted (listing 1, lines 8–11 and 14–18). We set the values of `x_pred` in R to the observed inputs `xf` to check the posterior predictions `y_pred` against the observed output y (listing 3, lines 20–21). Next, the measured output y and the simulation output η are standardized using the mean and standard deviation of η so that the simulation output has mean 0 and unit variance (listing 1, lines 24–27). This is followed by the transformation of the observable inputs x (listing 1, lines 30–37) and calibration parameters t (listing 1, lines 40–44) so that they lie in the interval [0,1].

4.2. Model calibration using observations

The next step is to write the Stan code in a text file with `.stan` extension. Listing 2 in Appendix C shows the Stan code for fitting the GP model and the MCMC sampling. Comments in Stan are indicated with a double slash "//". This Stan model consists of five blocks, the `data` block, the `transformed data` block, the `parameters` block, the `transformed parameters` block, and the `model` block.

The data block (listing 2, lines 1–12) declares and reads in the data for the model through memory. After the data is read into Stan, the `transformed data` block (listing 2, lines 14–25) is used to define transformed data.

Next, the random variables are defined in the `parameters` (listing 2, lines 27–40) and `transformed parameters` (listing 2, lines 42–48) block. This includes the calibration parameters $(t_1^f, \ldots, t_q^f)$, GP correlation hyperparameters

$(\beta_1^\eta, \ldots, \beta_{p+q}^\eta,\ \beta_1^\delta, \ldots, \beta_p^\delta)$, and the GP precision hyperparameters $(\lambda_\eta,\ \lambda_\delta,\ \lambda_\epsilon)$.

To model the correlation hyperparameters, we reparameterize them in the `transformed parameters` block using $\rho_i = \exp(-\beta_i/4)$ [44,45] (listing 2, lines 47–48).

Finally, the `model` block (listing 2, lines 51–111) is used to specify the computation of the covariance function, the prior distributions and the likelihood function. This block begins with the declaration of variables (listing 2, lines 53–59). This is followed by combining `xf`, `tf`, `xc` and `tc` in a single matrix `xt` (listing 2, lines 62–65), for the subsequent computation of Σ_η. The computation of the covariance matrices Σ_η, Σ_δ, and Σ_z is carried out in the `model` block and is best understood by comparing the code with Eqs. 2,3, and 5.

Computation of Σ_η is executed according to Eq. 2 and corresponds to line 68 and lines 71–78 of listing 2. This is followed by the computation of Σ_δ according to Eq. 3 (listing 2, line 81 and lines 84–91). The covariance matrix Σ_z can then be calculated following Eq. 5, which translates to lines 94–95 and 98–100 (listing 2).

Prior distributions for the parameters and the transformed parameters are specified on lines 103–107 of the `model` block (listing 2). Note that in `stan`, if no priors are specified, a non-informative or flat prior will be assigned. Therefore, within the context of listing 2, since no priors were assigned to the calibration parameters `tf`, flat non-informative priors will be assigned. The transformed parameters `rho_eta` and `rho_delta` are assigned `beta` priors (listing 2, lines 103–104). The statement `rho_eta[1:(p+q)] ~ beta(1.0, 0.3)` means that each random variable in the range `1:(p+q)` inclusive would be assigned a `beta(1.0, 0.3)` prior. The precision hyperparameters (`lambda_eta`, `lambda_delta`, `lambda_e`) were assigned `gamma` priors (listing 2, lines 105–107). Reasons for using `beta` priors for the transformed correlation hyperparameters (`rho_eta`, `rho_delta`) and `gamma` priors for the precision hyperparameters are elaborated in Section 3.1.

Lastly, line 110 of listing 2 shows the multivariate normal likelihood function (eq. 4). Cholesky decomposition (listing 2, line 109) is used for a more efficient implementation of the MCMC sampling.

4.3. Predictive inference at new input settings

It is often useful to be able to compute the corresponding outputs as a function of the inputs x and t. This allows us to look for trends in the quality of the model predictions as well as provide predictive inference at new input settings. The Stan code shown in listing 3 (Appendix D) is a fully Bayesian approach that implements predictive inference directly within Stan. However, one downside is that the Stan model might have a longer runtime compared to the Stan model without predictive inference (listing 2). On a MacBook Pro-2.3 GHz Intel Core i7 16 GB Memory, the code with predictive inference took about 65 minutes while the code without predictive inference took about 30 minutes to run[3]. This is because of the notable higher dimensions of the posterior distribution, since it now includes the predictions y_{pred}. An alternative would be to use a smaller but representative subset of the datasets D^f (Table 4) and D^c (Table 5) for the calibration [10]. Another possibility would be to compute the predictions outside of Stan in a partially Bayesian approach[4]. Both these methods are not within the scope of this paper and hence would not be discussed here.

[3] The runtime reported is for 1 MCMC chain with 500 iterations, using data for the case study building and weakly informative priors, where $D^f \in \mathbb{R}^{12\times4}$ and $D^c \in \mathbb{R}^{360\times7}$

[4] Code for predictions inference using R available on https://github.com/adChong/bc-stan

Predictive inference can be carried out in Stan using the joint distribution of the observed or measured output y, the simulation output η, and the unobserved predictions y_{pred} (listing 3, line 69). In other words, the output vector `z` is the concatenation of `y`, `eta`, and `y_pred`. Therefore, the data block now includes an input matrix of the new design points `x_pred` at which to make the predictions (listing 3, line 14). Consequently, the unobserved predictions `y_pred` for the corresponding input matrix `x_pred` are modeled as random variables in the *parameters* block (listing 3, line 46) and will be sampled when the model is executed.

Additionally, the variable `X` is declared in the *transformed data* block to combine the observed inputs `xf` and the `x_pred` into a single matrix (listing 3, line 28). The covariance function is then applied to this combined input matrix for the computation of Σ_δ in the `model` block (listing 3, lines 93–94 and lines 97–104). The variable `xt` in the *model* block is also used to combine `xf`, `tf`, `xc`, `tc` and `x_pred` in a single matrix (listing 3, lines 72–77), which is subsequently used for the computation of Σ_η (listing 3, line 80 and lines 83–90).

To fit the Stan model, the code shown in listing 3 is saved as `bcWithPred.stan`. The Stan model is then called from within R with the function `stan` of the `rstan` package (listing 1, lines 56–59). For the code shown here, 4 chains with 500 iterations each is specified. Executing lines 52 and 53 (listing 1) sets Stan to run the Markov chains in parallel using multiple cores. For the best efficiency, it is recommended to use as many processors as the hardware and RAM allows (up to the number of chains).

4.4. Evaluate convergence

The first step after running the `stan` function is to check if convergence has been achieved. To achieve convergence, each individual chain must achieve stationarity and all chains must show inter-chain mixing within the target parameter range. In the present paper, 4 chains of 500 iterations were used [10]. This means that 250 iterations were used for warm-up, and the subsequent 250 samples for analyzing the posterior distributions. Following Chong et al. [10], two practical ways to diagnose lack of convergence are included in the code in listing 1 (Appendix B).

The first is to look at the trace plots of multiple chains using the function `traceplot` (listing 1, lines 62–63). The argument `pars` provides the option to select the parameters of interest. As illustrated in lines 62–63 of listing 1, we chose to plot the trace plots for the parameters t^f, β^η, β^δ, λ_η, λ_δ, and λ_ϵ. As all hyperparameters are inherently linked to each other in the joint posterior distribution, it is important to assess convergence and the rationality of posterior values for all parameters. Convergence problems with one or more parameters can hint at structural issues of badly defined problems, which might also affect posterior inference of presumably well-converged parameters. Visual inspection of the trace plots allows us to check if the chains are well-mixed and whether (and how fast) different chains are converging to the same target distribution. Adequate convergence is indicated by trace plots that looks like a “fat, hairy caterpillar” that does not bend [46,47].

The second convergence diagnostics is the Gelman-Rubin statistic $\hat{R}$, which is the ratio of between-chain variance to within-chain variance [30]. Therefore, if different chains have converged to the same stationary posterior distribution, $\hat{R}$ should be approximately 1 ± 0.1. Each parameter has an associated $\hat{R}$ value, and can be obtained using the `print` function (listing 1, lines 66–67). The `print` function also provides the quantiles of the posterior distribution for each parameter. Similarly, the `pars` argument can be used to select parameters of interest.

4.5. Summarizing the result

The result of the Bayesian calibration is a joint posterior probability distributions of the predictions (y_{pred}), calibration parameters ($t_1^f, \ldots, t_q^f$), and the GP hyperparameters ($\beta_1^\eta, \ldots, \beta_{p+q}^\eta$, $\beta_1^\delta, \ldots, \beta_p^\delta$, λ_η, λ_δ, λ_ϵ). Of immediate interest is the posterior distribution of the calibration parameters t^f. The histograms for these parameters can be easily plotted with the `stan_hist` function (listing 1, line 70).

Given the additional focus on prediction, it is useful to determine if predictions from the calibrated model match the measured data reasonably well. Reality checks such as this are important in calibration frameworks that build a lot of structure into the fitted model, particularly when small amounts of field data are used for the calibration. Samples from the posterior distribution of the predictions (`y_pred`) are first extracted from the fitted model and then converted back to its original scale (listing 1, lines 74–75). For a visual comparison, we overlay box-plots of the predictions and the measured data on the same graph, against various input factors x (listing 1, lines 79–94).

5. Conclusions

This paper provides practical guidelines to the Bayesian calibration of building energy models (BEMs) using the probabilistic programming language Stan. A step-by-step guidance through the Bayesian calibration code is provided to ease the reader into its application to building energy models. Using a building case study, this paper also investigates the impact of (1) the number of calibration parameters and the (2) choice of prior distributions for these calibration parameters on the calibration results.

Preliminary findings from this case study show that over-parameterization leading to less precise posterior distributions can occur when the number of parameters ($\geq$ 4 for this case study) used for the BEM calibration is larger than can be uniquely identified with the limited measured data. However, this number is specific to the current case study and more case studies are necessary before a more generalized guideline on the recommended number of parameters can be provided. With respect to the choice of different prior distributions, the case study indicates that using a weakly informative prior generally represents a sensible choice, as it allows the Bayesian algorithm to take into account the prior information, as well as the data provided through the likelihood. More importantly, the case study also reveals that it can be misleading to interpret any posterior distribution as being representative of the true parameter values simply because the deviation of the calibrated predictions from the measurements is small.

Based on the observations from this case study, the following suggestions are recommended for the Bayesian calibration of BEMs:

- Carefully identify and select sensitive parameters for the calibration. Repeated calibration runs with different numbers of model parameters can help to identify identifiability issues that may occur due to over-parameterization.
- Use weakly informative priors rather than highly constrained specific informative priors so that the prior is able to exclude unreasonable parameter values, but not so strong as to rule out values that make sense according to the measured data. The loss in precision when using weaker priors is less serious than the $a - priori$ elimination of parameter values based on subjective beliefs.
- When using specific informative prior distributions, be transparent and clearly state the reason for using each prior.
- A low CVRMSE and NMBE indicates good agreement between the calibrated predictions and the measured outcomes, but does not provide sufficient information that the posterior estimates of the calibration parameters are good estimates of their true values and the associated uncertainties.
- Use trace-plots and Gelman-Rubin statistic $\hat{R}$ for convergence diagnostics and check that the posterior distributions are well mixed and have converged to a common stationary distribution.

Although this paper provides guidelines and the code for the Bayesian calibration of BEM, there are still many issues that need to be resolved. Therefore, future work should include assessing the impact of the choice of priors for the GP hyperparameters on posterior distributions and prediction results, as well as evaluating the impact of the observable inputs (x). To improve the practical application of Bayesian calibration of BEM further, more case studies are also required to further corroborate the findings observed in this specific case study. In particular, it would be useful to have case studies that included validation with true parameter values.

Appendix A. Details of uncertain parameters

Table A1
List of uncertain parameters and their range.

Model parameter	Symbol	Initial value	Min	Max
HVAC:				
Space cooling setpoint [°C]	θ_1	23.9	21	25
Space heating setpoint [°C]	θ_2	18.3	15	20.5
Fan efficiency* [-]	θ_3	0.5	0.1	0.9
Fan motor efficiency* [-]	θ_4	0.9	0.86	1.0
DX cooling COP [-]	θ_5	3	1	6
Gas burner efficiency [-]	θ_6	0.95	0.8	1.0
Baseboard nominal capacity [W]	θ_7	1500	1200	1800
Envelope thermal properties:				
Exterior wall R-value multiplier [-]	θ_8	1.0	0.9	1.1
Exterior wall density multiplier [-]	θ_9	1.0	0.98	1.02
Roof R-value multiplier [-]	θ_{10}	1.0	0.9	1.1
Roof density multiplier [-]	θ_{11}	1.0	0.98	1.02
Internal loads:				
Electric equipment power density [W/m^2]	θ_{12}	21.9	3	30
Lighting power density [W/m^2]	θ_{13}	14.6	2	20
Space infiltration [$m^3/s - m^2$]	θ_{14}	0.001524	0	0.001524

*Fan efficiency is the total efficiency of the fan (the ratio of the power delivered to the fluid to the electrical input power), while fan motor efficiency is the shaft power divided by the electrical power consumed

Appendix B. Code for setting up the data

```
library(rstan)

# read in field and computer simulation data
DATACOMP <- read.csv("DATACOMP.csv", header = TRUE)
DATAFIELD <- read.csv ("DATAFIELD.csv", header = TRUE)

# get dimensions of dataset
p <- ncol(DATAFIELD) - 1 # number of input factors
q <- ncol(DATACOMP) - p - 1 # number of calibration parameters
n <- nrow(DATAFIELD) # sample size of observed field data
m <- nrow(DATACOMP) # sample size of computer simulation data

# extract data from DATAFIELD (Table 3) and DATACOMP (Table 4)
y <- DATAFIELD[,1] # observed output
xf <- DATAFIELD[,2:(1+p)] # observed input
eta <- DATACOMP[,1] # simulation output
xc <- DATACOMP[,2:(1+p)] # simulation input
tc <- DATACOMP[,(2+p):(1+p+q)] # calibration parameters

x_pred <- xf # design points for predictions
n_pred <- nrow(x_pred) # design points for predictions

# standardization of output y and eta
eta_mu <- mean(eta, na.rm = TRUE) # mean value
eta_sd <- sd(eta, na.rm = TRUE) # standard deviation
y <- (y - eta_mu) / eta_sd
eta <- (eta - eta_mu) / eta_sd

# Put design points xf and xc on [0,1]
x <- rbind(as.matrix(xf), as.matrix(xc))
for (i in (1:ncol(x))){
  x_min <- min(x[,i], na.rm = TRUE)
  x_max <- max(x[,i], na.rm = TRUE)
  xf[,i] <- (xf[,i] - x_min) / (x_max - x_min)
  xc[,i] <- (xc[,i] - x_min) / (x_max - x_min)
  x_pred[,i] <- (x_pred[,i] - x_min) / (x_max - x_min)
}

# Put calibration parameters t on domain [0,1]
for (j in (1:ncol(tc))){
  tc_min <- min(tc[,j], na.rm = TRUE)
  tc_max <- max(tc[,j], na.rm = TRUE)
  tc[,j] <- (tc[,j] - tc_min) / (tc_max - tc_min)
}

# create data as list for input to Stan
```

Listing 1. R code for setting up the data and running the model in Stan.

```r
stan_data <- list(n=n, m=m, n_pred=n_pred, p=p, y=y, q=q, eta=eta,
                  xf=as.matrix(xf), xc=as.matrix(xc),
                  x_pred=as.matrix(x_pred), tc=as.matrix(tc))

# set stan to execute multiple Markov chains in parallel
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())

# run model in stan
fit <- stan(file = "bcWithPred.stan",
            data = stan_data,
            iter = 500,
            chains = 4)

# plot traceplots, excluding warm-up
stan_trace(fit, pars = c("tf", "beta_eta", "beta_delta",
                         "lambda_eta", "lambda_delta", "lambda_e"))

# summarize results
print(fit, pars = c("tf", "beta_eta", "beta_delta",
                    "lambda_eta", "lambda_delta", "lambda_e"))

# posterior probability distribution of tf
stan_hist(fit, pars = c("tf"))

# extract predictions, excluding warm-up and
# convert back to original scale
samples <- rstan::extract(fit)
y_pred <- samples$y_pred * eta_sd + eta_mu
n_samples <- nrow(y_pred)

# for loop to visualize predictions at different input x
for (i in (1:p)) {
  field_data <- data.frame(yf=DATAFIELD[, 1],
                           xf=signif(DATAFIELD[, (i+1)], 3))
  pred_data <- matrix(data=t(y_pred),
                      nrow=length(y_pred), ncol = 1)
  plot_data <- data.frame(apply(field_data, 2, rep,
                                n_samples), pred = pred_data)
  # save plot as png file
  png(paste("plot", i, ".png", sep = ""))
  plt <- ggplot(data = plot_data, aes(y=pred, x=xf, group=xf)) +
    geom_boxplot(outlier.size=0.2) +
    geom_point(data = field_data, aes(x=xf, y=yf),
               color="#D55E00", size=0.8)
  print(plt)
  dev.off()
}
```

Listing 1. Continued

Appendix C. Stan code for KOH Bayesian calibration

```
data {
  int<lower=0> n; // number of field data
  int<lower=0> m; // number of computer simulation
  int<lower=0> p; // number of observable inputs x
  int<lower=0> q; // number of calibration parameters t
  vector[n] y; // field observations
  vector[m] eta; // output of computer simulations
  matrix[n, p] xf; // observable inputs corresponding to y
  // (xc, tc): design points corresponding to eta
  matrix[m, p] xc;
  matrix[m, q] tc;
}

transformed data {
  int<lower=1> N;
  vector[n+m] z; // z = [y, eta]
  vector[n+m] mu; // mean vector

  N = n + m;
  // set mean vector to zero
  for (i in 1:N) {
    mu[i] = 0;
  }
  z = append_row(y, eta);
}

parameters {
  // tf: calibration parameters
  // rho_eta: reparameterization of beta_eta
  // rho_delta: reparameterization of beta_delta
  // lambda_eta: precision parameter for eta
  // lambda_delta: precision parameter for bias term
  // lambda_e: precision parameter of observation error
  row_vector<lower=0,upper=1>[q] tf;
  row_vector<lower=0,upper=1>[p+q] rho_eta;
  row_vector<lower=0,upper=1>[p] rho_delta;
  real<lower=0> lambda_eta;
  real<lower=0> lambda_delta;
  real<lower=0> lambda_e;
}

transformed parameters {
  // beta_delta: correlation parameter for bias term
  // beta_e: correlation parameter of observation error
  row_vector[p+q] beta_eta;
  row_vector[p] beta_delta;
  beta_eta = -4.0 * log(rho_eta);
  beta_delta = -4.0 * log(rho_delta);
```

Listing 2. Stan code for Bayesian calibration.

```
}

model {
  // declare variables
  matrix[N, (p+q)] xt;
  matrix[N, N] sigma_eta; // simulator covariance
  matrix[n, n] sigma_delta; // bias term covariance
  matrix[N, N] sigma_z; // covariance matrix
  matrix[N, N] L; // cholesky decomposition of covariance matrix
  row_vector[p] temp_delta;
  row_vector[p+q] temp_eta;

  // xt = [[xt,tf],[xc,tc]]
  xt[1:n, 1:p] = xf;
  xt[(n+1):N, 1:p] = xc;
  xt[1:n, (p+1):(p+q)] = rep_matrix(tf, n);
  xt[(n+1):N, (p+1):(p+q)] = tc;

  // diagonal elements of sigma_eta
  sigma_eta = diag_matrix(rep_vector((1 / lambda_eta), N));

  // off-diagonal elements of sigma_eta
  for (i in 1:(N-1)) {
    for (j in (i+1):N) {
      temp_eta = xt[i] - xt[j];
      sigma_eta[i, j] = beta_eta .* temp_eta * temp_eta';
      sigma_eta[i, j] = exp(-sigma_eta[i, j]) / lambda_eta;
      sigma_eta[j, i] = sigma_eta[i, j];
    }
  }

  // diagonal elements of sigma_delta
  sigma_delta = diag_matrix(rep_vector((1 / lambda_delta), n));

  // off-diagonal elements of sigma_delta
  for (i in 1:(n-1)) {
    for (j in (i+1):n) {
      temp_delta = xf[i] - xf[j];
      sigma_delta[i, j] = beta_delta .* temp_delta * temp_delta';
      sigma_delta[i, j] = exp(-sigma_delta[i, j]) / lambda_delta;
      sigma_delta[j, i] = sigma_delta[i, j];
    }
  }
```

Listing 2. Continued

```

  // computation of covariance matrix sigma_z
  sigma_z = sigma_eta;
  sigma_z[1:n, 1:n] = sigma_eta[1:n, 1:n] + sigma_delta;

  // add observation errors
  for (i in 1:n) {
    sigma_z[i, i] = sigma_z[i, i] + (1.0 / lambda_e);
  }

  // Specify priors here
  rho_eta[1:(p+q)] ~ beta(1.0, 0.3);
  rho_delta[1:p] ~ beta(1.0, 0.3);
  lambda_eta ~ gamma(10, 10); // gamma (shape, rate)
  lambda_delta ~ gamma(10, 0.3);
  lambda_e ~ gamma(10, 0.03);

  L = cholesky_decompose(sigma_z); // cholesky decomposition
  z ~ multi_normal_cholesky(mu, L);
}
```

Listing 2. Continued

Appendix D. Stan code with predictive inference

```
data {
  int<lower=1> n; // number of field data
  int<lower=1> m; // number of computer simulation
  int<lower=1> n_pred; // number of predictions
  int<lower=1> p; // number of observable inputs x
  int<lower=1> q; // number of calibration parameters t
  vector[n] y; // field observations
  vector[m] eta; // output of computer simulations
  matrix[n, p] xf; // observable inputs corresponding to y
  // (xc, tc): design points corresponding to eta
  matrix[m, p] xc;
  matrix[m, q] tc;
  // x_pred: new design points for predictions
  matrix[n_pred, p] x_pred;
}

transformed data {
  int<lower = 1> N;
  vector[n+m] y_eta;
  vector[n+m+n_pred] mu; // mean vector
  matrix[n+n_pred, p] X; // X=[xf, x_pred]

  N = n + m + n_pred;
  // set mean vector to zero
  for (i in 1:N) {
    mu[i] = 0;
  }
  X = append_row(xf, x_pred);
  y_eta = append_row(y, eta); // y_eta = [y, eta]
}

parameters {
  // tf: calibration parameters
  // rho_eta: reparameterization of beta_eta
  // rho_delta: reparameterization of beta_delta
  // lambda_eta: precision parameter for eta
  // lambda_delta: precision parameter for bias term
  // lambda_e: precision parameter of observation error
  // y_pred: predictions
  row_vector<lower=0, upper=1>[q] tf;
  row_vector<lower=0, upper=1>[p+q] rho_eta;
  row_vector<lower=0, upper=1>[p] rho_delta;
  real<lower=0> lambda_eta;
  real<lower=0> lambda_delta;
  real<lower=0> lambda_e;
  vector[n_pred] y_pred;
}
```

Listing 3. Stan code for Bayesian calibration with predictive inference.

```

transformed parameters {
  // beta_delta: correlation parameter for bias term
  // beta_e: correlation parameter of observation error
  row_vector[p+q] beta_eta;
  row_vector[p] beta_delta;
  beta_eta = -4.0 * log(rho_eta);
  beta_delta = -4.0 * log(rho_delta);
}

model {
  // declare variables
  matrix[N, p+q] xt;
  matrix[N, N] sigma_eta; // simulator covarinace
  matrix[n+n_pred, n+n_pred] sigma_delta; // bias term covariance
  matrix[N, N] sigma_z; // covariance matrix
  matrix[N, N] L; // cholesky decomposition of covariance matrix
  vector[N] z; // z = [y, eta, y_pred]
  row_vector[p] temp_delta;
  row_vector[p+q] temp_eta;

  z = append_row(y_eta, y_pred); // z = [y, eta, y_pred]

  // xt = [[xf,tf],[xc,tc],[x_pred,tf]]
  xt[1:n, 1:p] = xf;
  xt[1:n, (p+1):(p+q)] = rep_matrix(tf, n);
  xt[(n+1):(n+m), 1:p] = xc;
  xt[(n+1):(n+m), (p+1):(p+q)] = tc;
  xt[(n+m+1):N, 1:p] = x_pred;
  xt[(n+m+1):N, (p+1):(p+q)] = rep_matrix(tf, n_pred);

  // diagonal elements of sigma_eta
  sigma_eta = diag_matrix(rep_vector((1 / lambda_eta), N));

  // off-diagonal elements of sigma_eta
  for (i in 1:(N-1)) {
    for (j in (i+1):N) {
      temp_eta = xt[i] - xt[j];
      sigma_eta[i, j] = beta_eta .* temp_eta * temp_eta';
      sigma_eta[i, j] = exp(-sigma_eta[i, j]) / lambda_eta;
      sigma_eta[j, i] = sigma_eta[i, j];
    }
  }
```

Listing 3. Continued

```

  // diagonal elements of sigma_delta
  sigma_delta = diag_matrix(rep_vector((1 / lambda_delta),
    n+n_pred));

  // off-diagonal elements of sigma_delta
  for (i in 1:(n+n_pred-1)) {
    for (j in (i+1):(n+n_pred)) {
      temp_delta = X[i] - X[j];
      sigma_delta[i, j] = beta_delta .* temp_delta * temp_delta';
      sigma_delta[i, j] = exp(-sigma_delta[i, j]) / lambda_delta;
      sigma_delta[j, i] = sigma_delta[i, j];
    }
  }

  // computation of covariance matrix sigma_z
  sigma_z = sigma_eta;
  sigma_z[1:n, 1:n] = sigma_eta[1:n, 1:n] +
    sigma_delta[1:n, 1:n];
  sigma_z[1:n, (n+m+1):N] = sigma_eta[1:n, (n+m+1):N] +
    sigma_delta[1:n, (n+1):(n+n_pred)];
  sigma_z[(n+m+1):N, 1:n] = sigma_eta[(n+m+1):N, 1:n] +
    sigma_delta[(n+1):(n+n_pred),1:n];
  sigma_z[(n+m+1):N, (n+m+1):N] = sigma_eta[(n+m+1):N, (n+m+1):N] +
    sigma_delta[(n+1):(n+n_pred), (n+1):(n+n_pred)];

  // add observation errors
  for (i in 1:n) {
    sigma_z[i, i] = sigma_z[i, i] + (1.0 / lambda_e);
  }

  // Specify priors here
  rho_eta[1:(p+q)] ~ beta(1.0, 0.3);
  rho_delta[1:p] ~ beta(1.0, 0.3);
  lambda_eta ~ gamma(10, 10); // gamma (shape, rate)
  lambda_delta ~ gamma(10, 0.3);
  lambda_e ~ gamma(10, 0.03);

  L = cholesky_decompose(sigma_z); // cholesky decomposition
  z ~ multi_normal_cholesky(mu, L);
}
```

Listing 3. Continued

References

[1] M.C. Kennedy, A. O'Hagan, Bayesian calibration of computer models, J. Royal Stat. Soc. Series B (Stat. Methodol.) 63 (3) (2001) 425–464.
[2] G. Augenbroe, Trends in building simulation, Build. Environ. 37 (8) (2002) 891–902.
[3] T.A. Reddy, Literature review on calibration of building energy simulation programs: uses, problems, procedures, uncertainty, and tools, ASHRAE Trans. 112 (1) (2006).
[4] EVO, International performance measurement and verification protocol: concepts and options for determining energy and water savings volume 1, Efficiency Valuation Organization (2012).
[5] ASHRAE, Guideline 14–2002, measurement of energy and demand savings, American Society of Heating, Ventilating, and Air Conditioning Engineers, Atlanta, Georgia (2002).
[6] US DOE FEMP, M&V guidelines: measurement and verification for performance-based contracts, version 4.0, Energ. Effic. Renew. Energ. (2015).
[7] S. De Wit, G. Augenbroe, Analysis of uncertainty in building design evaluations and its implications, Energy Build. 34 (9) (2002) 951–958.
[8] A. Booth, R. Choudhary, D. Spiegelhalter, Handling uncertainty in housing stock models, Build. Environ. 48 (2012) 35–47.
[9] A. Booth, R. Choudhary, D. Spiegelhalter, A hierarchical bayesian framework for calibrating micro-level models with macro-level data, J. Build. Perform. Simul. 6 (4) (2013) 293–318.
[10] A. Chong, K.P. Lam, M. Pozzi, J. Yang, Bayesian calibration of building energy models with large datasets, Energy Build. 154 (2017) 343–355.
[11] A. Chong, K.P. Lam, A comparison of mcmc algorithms for the bayesian calibration of building energy models, in: Proceedings of the 15th IBPSA Building Simulation Conference, 2017.
[12] ISO, ISO 13790:2008 energy performance of buildings - calculation of energy use for space heating and cooling, International Organization for standardization (2008).
[13] Y. Heo, R. Choudhary, G. Augenbroe, Calibration of building energy models for retrofit analysis under uncertainty, Energy Build. 47 (2012) 550–560.
[14] Y. Heo, G. Augenbroe, R. Choudhary, Quantitative risk management for energy retrofit projects, J. Build. Perform. Simul. 6 (4) (2013) 257–268.
[15] Y. Heo, D.J. Graziano, L. Guzowski, R.T. Muehleisen, Evaluation of calibration efficacy under different levels of uncertainty, J. Build. Perform. Simul. 8 (3) (2015) 135–144.
[16] Y.-J. Kim, C.-S. Park, Stepwise deterministic and stochastic calibration of an energy simulation model for an existing building, Energy Build. 133 (2016) 455–468.
[17] M.H. Kristensen, R. Choudhary, R.H. Pedersen, S. Petersen, Bayesian calibration of residential building clusters using a single geometric building representation, in: Building Simulation 2017, 2017, pp. 1650–1659.
[18] M.H. Kristensen, R. Choudhary, S. Petersen, Bayesian calibration of building energy models: comparison of predictive accuracy using metered utility data of different temporal resolution, Energy Procedia. 122 (2017) 277–282.
[19] Q. Li, G. Augenbroe, J. Brown, Assessment of linear emulators in lightweight bayesian calibration of dynamic building energy models for parameter estimation and performance prediction, Energy Build. 124 (2016) 194–202.
[20] H. Lim, Z.J. Zhai, Comprehensive evaluation of the influence of meta-models on bayesian calibration, Energy Build. (2017).
[21] BS, EN 15603:2008 energy performance of buildings. overall energy use and definition of energy ratings (2008).
[22] M. Manfren, N. Aste, R. Moshksar, Calibration and uncertainty analysis for computer models–a meta-model based approach for integrated building energy simulation, Appl. Energy 103 (2013) 627–641.
[23] K. Menberg, Y. Heo, R. Choudhary, Efficiency and reliability of bayesian calibration of energy supply system models, in: Proceedings of the 15th IBPSA Building Simulation Conference, 2017.
[24] J. Sokol, C.C. Davila, C.F. Reinhart, Validation of a bayesian-based method for defining residential archetypes in urban building energy models, Energy Build. 134 (2017) 11–24.
[25] W. Tian, S. Yang, Z. Li, S. Wei, W. Pan, Y. Liu, Identifying informative energy data in bayesian calibration of building energy models, Energy Build. 119 (2016) 363–376.
[26] Y. Heo, G. Augenbroe, D. Graziano, R.T. Muehleisen, L. Guzowski, Scalable methodology for large scale building energy improvement: relevance of calibration in model-based retrofit analysis, Build Environ. 87 (2015) 342–350.
[27] D. Higdon, M. Kennedy, J.C. Cavendish, J.A. Cafeo, R.D. Ryne, Combining field data and computer simulations for calibration and prediction, SIAM J. Scient. Comput. 26 (2) (2004) 448–466.
[28] C.E. Rasmussen, C.K. Williams, Gaussian processes for machine learning, Vol. 1, MIT press Cambridge, 2006.
[29] R.M. Neal, Probabilistic inference using markov chain monte carlo methods (1993).
[30] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Rubin, Bayesian data analysis, Vol. 2, Taylor & Francis, 2014.
[31] M.D. Hoffman, A. Gelman, The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo, J. Mach. Learn. Res. 15 (1) (2014) 1593–1623.
[32] M. Betancourt, A conceptual introduction to Hamiltonian Monte Carlo, Report, 2016.
[33] M.D. Morris, Factorial sampling plans for preliminary computational experiments, Technometrics 33 (2) (1991) 161–174.
[34] A. Gelman, C. Hennig, Beyond subjective and objective in statistics, J. Royal Stat. Soc. Series A (Stat. Soc.) 180 (4) (2017) 967–1033.
[35] J. Brynjarsdttir, A. O'Hagan, Learning about physical parameters: the importance of model discrepancy, Inverse Probl. 30 (11) (2014) 114007.
[36] K. Menberg, Y. Heo, R. Choudhary, Influence of error terms in bayesian calibration of energy system models, J. Build. Perform. Simul. (2018) 1–15, doi:10.1080/19401493.2018.1475506.
[37] W. Tian, A review of sensitivity analysis methods in building energy analysis, Renew. Sustain. Energ. Rev. 20 (2013) 411–419.
[38] K. Menberg, Y. Heo, R. Choudhary, Sensitivity analysis methods for building energy models: comparing computational costs and extractable information, Energy Build. 133 (2016) 433–445.
[39] F. Campolongo, J. Cariboni, A. Saltelli, An effective screening design for sensitivity analysis of large models, Environment. Model. Softw. 22 (10) (2007) 1509–1518.
[40] J.L. Loeppky, J. Sacks, W.J. Welch, Choosing the sample size of a computer experiment: a practical guide, Technomet. 51 (4) (2009) 366–376.
[41] I.A. Macdonald, Quantifying the effects of uncertainty in building simulation, University of Strathclyde, 2002.
[42] Stan Development Team, Prior choice recommendations, 2017, (https://github.com/stan-dev/stan/wiki/Prior-Choice-Recommendationsa).
[43] Stan Development Team, Rstan: the r interface to stan, version 2.16.2, 2017, (http://mc-stan.orgb).
[44] D. Higdon, C. Nakhleh, J. Gattiker, B. Williams, A bayesian calibration approach to the thermal problem, Comput. Methods Appl. Mech. Eng. 197 (29) (2008) 2431–2441.
[45] S. Guillas, J. Rougier, A. Maute, A. Richmond, C. Linkletter, Bayesian calibration of the thermosphere-ionosphere electrodynamics general circulation model (tie-gcm), Geosci. Model Dev. 2 (2) (2009) 137.
[46] D. Lunn, C. Jackson, N. Best, A. Thomas, D. Spiegelhalter, The BUGS book: A practical introduction to bayesian analysis, CRC press, 2012.
[47] T. Sorensen, S. Vasishth, Bayesian linear mixed models using stan: a tutorial for psychologists, linguists, and cognitive scientists, arXiv preprint arXiv:1506.06201 (2015).