Using Bayesian deep learning approaches for uncertainty-aware building energy surrogate models

Paul Westermann & Ralph Evins

March 2021

Energy and AI 3 (2021) 100039

Contents lists available at ScienceDirect

Energy and AI

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

Using Bayesian deep learning approaches for uncertainty-aware building energy surrogate models

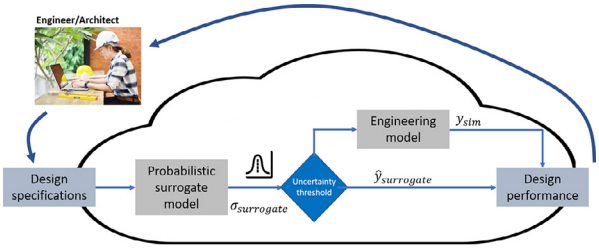
Paul Westermann*, Ralph Evins

Energy and Cities Group Department of Civil Engineering, University of Victoria, Canada

HIGHLIGHTS

- Developing uncertainty-aware engineering surrogate models.
- Comparing deep Bayesian neural networks and Gaussian process models.
- Uncertainty estimates can identify and mitigate errors in surrogate models.
- A concept to hybridize engineering models and data-driven models.

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 6 October 2020
Received in revised form 17 November 2020
Accepted 7 December 2020

Keywords:
Surrogate modelling
Metamodel
Building performance simulation
Uncertainty
Bayesian deep learning
Gaussian Process
Bayesian neural network

ABSTRACT

Fast machine learning-based surrogate models are trained to emulate slow, high-fidelity engineering simulation models to accelerate engineering design tasks. This introduces uncertainty as the surrogate is only an approximation of the original model.

Bayesian methods can quantify that uncertainty, and deep learning models exist that follow the Bayesian paradigm. These models, namely Bayesian neural networks and Gaussian process models, enable us to give predictions together with an estimate of the model's uncertainty. As a result we can derive uncertainty-aware surrogate models that can automatically identify unseen design samples that may cause large emulation errors. For these samples the high-fidelity model can be queried instead. This paper outlines how the Bayesian paradigm allows us to hybridize fast but approximate and slow but accurate models.

In this paper, we train two types of Bayesian models, dropout neural networks and stochastic variational Gaussian Process models, to emulate a complex high dimensional building energy performance simulation problem. The surrogate model processes 35 building design parameters (inputs) to estimate 12 annual building energy performance metrics (outputs). We benchmark both approaches, prove their accuracy to be competitive, and show that errors can be reduced by up to 30% when the 10% of samples with the highest uncertainty are transferred to the high-fidelity model.

1. Introduction

A wealth of concepts exist to explore the design of new and existing buildings to improve the building sector's large climate footprint [1]. Scaling them is challenging, as usually each building is designed individually, responding to the cultural context, climatic conditions, surrounding buildings and design preferences. This impedes the distribution of centrally-derived design paradigms to the level of individual building projects.

Architects and engineers play a vital role in bridging the gap between high-level ideas and individual building projects. Often they use building performance simulation (BPS) tools to assess the energy and environmental performance of various design options and balance them against design preferences. The computational expense and associated

Abbreviations: BDL, Bayesian deep learning; BNN, Bayesian neural network; SVGP, stochastic-variational Gaussian Process; DoE, design-of-experiment; ReLU, rectified linear unit.

* Corresponding author.
E-mail addresses: pwestermann@uvic.ca (P. Westermann), revins@uvic.ca (R. Evins).

https://doi.org/10.1016/j.egyai.2020.100039

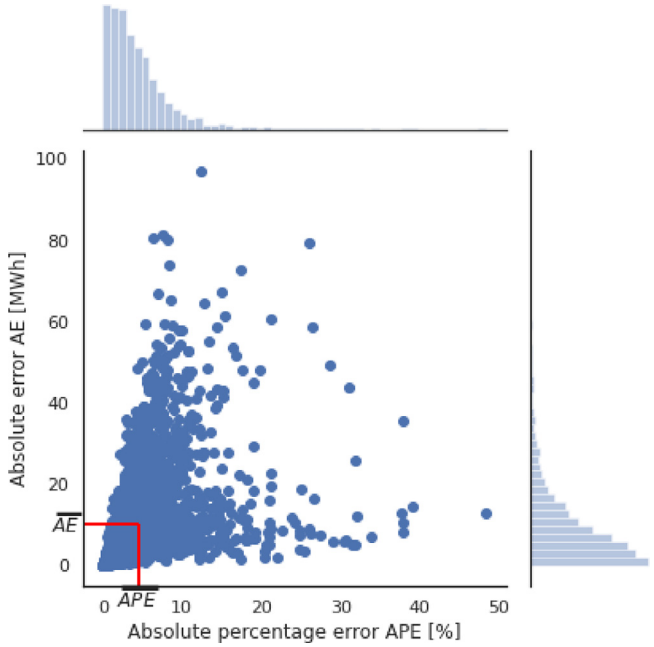


Fig. 1. Distribution of errors of a surrogate model. The plot shows the error of a surrogate model which emulates the simulation of the heating demand of an office building (see case study in Section 4). While the average absolute error $\overline{AE}$ and absolute percentage error $\overline{APE}$ are low (indicated by the red lines), large errors can occur. This study aims to identify the large errors using estimates of surrogate model uncertainty.

waiting time, however, prohibits exhaustive design space exploration and optimization. This has led researchers to train machine learning models on simulation input and output data to emulate building simulation models [2].

The computational speed of these so-called 'surrogate models' has been the basis for a range of innovations in the field of building simulation, for example, interactive early-stage design tools (e.g. ELSA [3], Building Pathfinder [4], Net-Zero Navigator[5]), faster optimization algorithms [6], and detailed design sensitivity and uncertainty analysis [7][8]. A recent survey of building designers confirmed that those who received realtime feedback from a surrogate model arrived at higher performing building designs [9].

The growing use of surrogate models turns attention to the robustness of their accuracy. The accuracy of a surrogate model is measured by the error of the surrogate model to estimate the physics-based simulation results, which is considered the ground truth.[1] Studies have shown satisfactory average accuracy on test data [11] which can be influenced by the type and the complexity of inputs [12] and the selection of outputs [5].

Nonetheless, average errors computed on test data can be deceiving (see Fig. 1). Test data usually consists of design samples distributed uniformly in the design space and may not reflect the portion of the space the building designer is interested in. Large errors on specific building designs may occur (i.e. heteroscedasticity of the errors), affecting important design choices and potentially lowering the energy performance of the final building design.

Bayesian methods offer a framework to quantify the uncertainty stemming from the inadequacy of an approximate model (epistemic uncertainty) and recent developments in Bayesian deep learning (BDL) managed to integrate Bayesian concepts into large machine learning models [13,14]. As a result BDL-based surrogate models can express for which inputs their estimates are uncertain. In our case, a Bayesian surrogate model produces a building performance estimate as a probability distribution, where the entropy or variance of that distribution allow us to quantify the uncertainty. The architect or building designer is therefore provided with a level of confidence in the performance results and thus can define uncertainty thresholds above which the high-fidelity model, here the BPS tool, is queried to guarantee high confidence results (see Fig. 2).

In this study, we explore two different Bayesian models, Bayesian neural networks [15] and stochastic variational Gaussian process models [16], to quantify epistemic uncertainty in surrogate models (see Section 2). Both models were chosen as they scale well to large surrogate modelling problems with many inputs and outputs which requires to train the models on large datasets. We benchmark the overall accuracy against non-Bayesian surrogate models, validate the quality of the uncertainty estimate, and quantify how a *hybridization* of fast but approximate and slow but accurate models reduces the error of a surrogate model while computational costs increase only slightly (see Section 5 ff.).

2. Background

2.1. Motivation for surrogate modelling

The core motivation to emulate a physics-based high-fidelity model is computational efficiency; simulation outputs can be estimated many orders of magnitude faster, effectively in real-time. This allows a holistic design space analysis which would be infeasible with a slow simulation model. Various applications of surrogate modelling are found in the building domain as well as other domains [18,19]:

- General design space exploration: The relationship between design parameters and performance is interactively explored to improve the user's understanding of the design problem [9,20]. This can happen on the single building level or on the urban level [21]. Often a parallel-coordinates plot is used to visualize the multi-dimensional problem space [5].
- Design optimization: The surrogate model is trained and queried to accelerate iterative optimization algorithms [22–24]. Adaptively training the surrogate model on new simulation samples collected at each optimization iteration can further increase optimization performance [6].
- Sensitivity analysis: The surrogate model is used to run the extensive sampling (thousands of simulation runs) required for global sensitivity analysis methods [7].
- Design uncertainty analysis: Several types of uncertainties exist during the building design process - caused by undetermined design parameters, uncertain contextual parameters (e.g. surrounding buildings, carbon factors, etc.), and vague design constraints [25]. This uncertainty is often quantified using Monte Carlo sampling methods, where samples from uncertain parameter distributions are drawn and simulated to quantify how that parameter uncertainty propagates to building performance uncertainty. With a surrogate model, these uncertainties can rapidly be calculated and updated throughout the design process [8].
- Simulation model calibration: An accurate calibration of a simulation model is required to assess retrofit design choices for an existing building. The calibration, i.e. the process of determining uncertain building parameters, often relies either on iterative optimiziation algorithms [26], or on Bayesian calibration of these uncertain parameters [27]. In both cases simulations are iteratively run to closely match simulation outputs with measured sensor data by adjusting the unknown parameters. One can use surrogate models to reduce the computational limitations of these approaches. Note that

[1] Please note, that the surrogate model accuracy does not reflect how well the underlying simulation model matches a real-world building. The reader is referred to [10] and many other studies, that address the gap between simulation model and the real building.

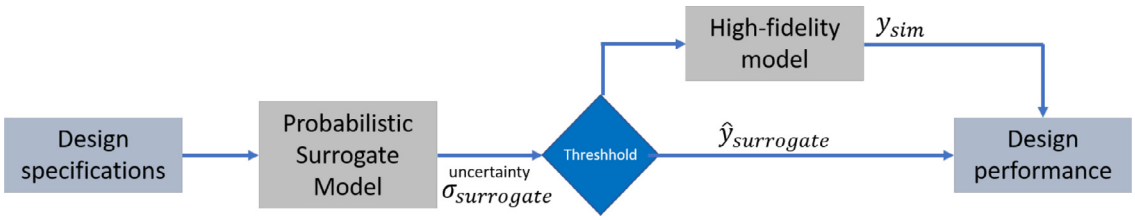


Fig. 2. Uncertainty estimates to link high-fidelity model and a surrogate model. The surrogate model provides both a performance estimate $\hat{y}_{surrogate}$ and an uncertainty estimate $\hat{\sigma}_{surrogate}$. If the uncertainty is large, a high-fidelity model (e.g. a building energy simulation) is queried to produce accurate estimates y_{sim} of an engineering design (e.g. a building). Please compare to [17] who introduced a similar concept.

simulation model calibration can be done both for a specific building [28] or multiple buildings [29]. The latter commonly requires an archetype model whose parameters are repeatedly calibrated using measurements of the considered buildings [30].

2.2. Surrogate model derivation

In surrogate modelling, we fit a machine learning model to a simulation dataset $D = \{x_n, y_n\}_{n=1}^{N} = (X, Y)$ consisting of N samples, where the inputs x_n correspond to the simulation parameters and y_n to real-valued outputs of the simulation run recorded for sample n [19].[2] In the case of building energy surrogate models, the simulation parameters are the building design parameters (e.g. insulation value of the walls) and the outputs are the simulated building performance metrics like the aggregated annual energy consumption or greenhouse gas emissions [2]. Studies also exist with time series outputs, like hourly energy demand [21].

For deriving the surrogate model the modeller first needs to carefully specify the design problem, which includes choosing the free design parameters and the performance objectives as well as all other important contextual parameters (surrounding buildings, etc.). Then simulations are run to create the simulation dataset D. The idea is to gain maximum information about the design space (the collection of all possible parameter combinations) per simulation run. Tailored sampling schemes exist, called design-of-experiment methods [31], e.g. Latin-Hypercube-sampling that uniformly distributes samples in the multidimensional input space. The number of samples must be specified (e.g. 10-1000 samples per parameter dimension [2]) and is adjusted if model accuracy on test samples is too low.

2.3. Accuracy in surrogate modelling

The accuracy of a surrogate model is quantified by how well its building performance estimates match true, physics-based simulation outputs. We assume the simulation model as our ground-truth model, and disregard the mismatch between the simulation model and the real-world building when calculating the surrogate's accuracy throughout the paper.

Metrics like the coefficient of determination (R^2), the mean absolute percentage error ($MAPE$), or the root-mean-squared-error ($RMSE$) can be used to quantify accuracy [32]. Based on [5,11], accuracies of $R^2 > 0.99$ are feasible when estimating annually aggregated performance metrics, e.g. heating demand, but they can be significantly lower when more complex performance metrics are estimated.

As mentioned above, surrogate model accuracy is commonly reported as one metric, implying homoscedastic errors. This may not always hold, i.e. the errors may depend on the choice of inputs (heteroscedasticity). By using Bayesian deep learning [13], we aim to train surrogates that are aware of where in the design space, i.e. for which building designs $x \in X$, the model is uncertain and may produce large errors.

2.4. Uncertainty in surrogate models

A mathematical function f of the simulation is not explicitly available. We use the surrogate model to find an estimate $\hat{f}$ to approximate that function. The most important cause of uncertainty in surrogate modelling is how plausible the determined $\hat{f}$ is (model uncertainty or *epistemic* uncertainty) [13]. For the most part, this uncertainty is caused by the training set $D = (X, Y)$ which contains only a finite set of points within the space of possible simulation parameter combinations X (the design space) and associated building performance Y. Theoretically, epistemic uncertainty can be reduced to zero given more and more data [13].

We consider the problem of surrogate modelling as free of *aleatoric uncertainty*, which represents noise or other unknowns impacting the observations.[3] Therefore, we only deal with epistemic uncertainty. We propose that quantifying this uncertainty can be a powerful aid in surrogate modelling as it acknowledges that we have to train our model with a limited number of simulation samples that represent a fraction of the design space, which makes the surrogate model uncertain. Bayesian modelling now allows us to reason under that uncertainty, while still benefiting from the advantages of surrogate modelling, i.e. the computational efficiency for large scale design space exploration.

2.4.1. Other sources of uncertainty in building performance simulation

The scope of this study is specifically set on estimating the uncertainty caused by training a surrogate model to emulate a simulation model (see Fig. 3). It does not consider or compute any other sources of uncertainty prevailing in building performance modelling, which may include uncertainty in design parameter and model specification, uncertainty in the properties of the final construction and uncertainty stemming from assumptions of internal (e.g. occupant behaviour) and external (e.g. climate) conditions [25]. Where uncertainty in surrogate modelling is purely caused by the modelling process (epistemic), uncertainty in specifying a simulation model is aleatoric. For more insights on the uncertainties tackling the mismatch between the simulation model and the constructed building, the reader is referred to [34] instead.

3. Bayesian modelling for surrogate models

Bayesian probability theory offers us grounded tools to quantify model uncertainty [35].

To understand the core idea of Bayesian modelling, we consider a parametric model $y = f(x, \Theta)$, where x is the input, f is a space of possible models (see Fig. 4) and Θ is the set of model parameters

[2] Also categorical outputs can be considered but practical examples are lacking in building simulation literature.

[3] In the case of sensor data, this can correspond to sensor noise. Here, we consider simulation runs to be deterministic, i.e. the impact of numerical noise to be small. In the case of numerical building simulation, here EnergyPlus [33], this corresponds to the numerical noise of solving the thermodynamic-based differential equations.

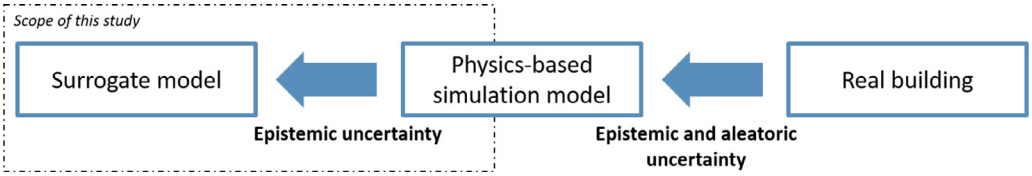


Fig. 3. Uncertainty in surrogate modelling, and uncertainty in building performance simulation.

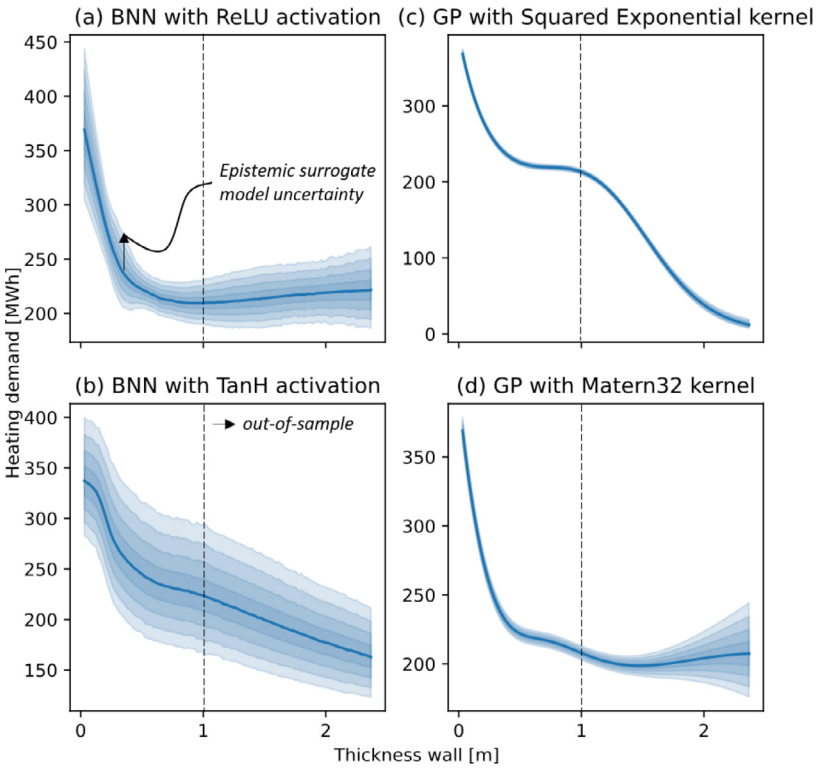


Fig. 4. Heating demand estimated with a Bayesian neural network, and the associated epistemic uncertainty. In particular, the uncertainty of the surrogate model is large when the building has a wall thickness wider than $1m$, which is wider than all samples contained in the training data (out-of-sample).

(for example, the weights in a neural network). Instead of finding a single Θ, in Bayesian modelling we search for a collection of Θ, which likely has produced the output Y given X. In our case we search for a collection of surrogate models with different weights.

The Bayesian theorem, as shown in Eq. (1), is applied to find a collection which likely has produce Y given X. Based on our prior knowledge on the distribution of the model weights $p(\Theta)$ and combined with the likelihood function $p(Y|X,\Theta) = \prod_{n=1}^{N} p(y_n|x_n,\Theta)$, which quantifies the probability that a specific model parameter set generated the observations (X,Y), the posterior of the model parameters can be computed.

$$p(\Theta|Y,X) = \frac{p(Y|X,\Theta)p(\Theta)}{p(Y|X)} \tag{1}$$

where $p(Y|X)$ is called the marginal likelihood. It represents the probability of the observed data given the model f with all possible model parameters. It is a scalar that normalizes the posterior. Given the posterior, we can now infer about future data in form of a predictive distribution:

$$p(y_*|x_*,X,Y) = \int p(y_*|x_*,\Theta)p(\Theta|X,Y)d\Theta \tag{2}$$

The mean and variance or entropy can be derived, where the latter two provide information on the uncertainty in the estimated values. In the building surrogate modelling setting, we predict an expected building performance, e.g. annual heating demand, and an associated uncertainty given building design parameters, e.g. the thickness of the wall (see Fig. 4).

3.1. Variational inference

The true posterior of the weights $p(\Theta|Y,X)$ however, is commonly intractable. This is particularly the case in the big data regime when more complex models are required [16]. In the small data regime (below a few thousand samples) posterior inference with a standard Gaussian Process Bayesian model is feasible and was successfully applied for building surrogate models [28,36]. However, with increasing complexity, for example more inputs and outputs (e.g. [12]), standard GPs have major shortcomings:

- The model complexity is limited as it only consists of one layer, i.e. the outputs of the GP are not used as inputs to another GP. This prohibits modeling hierarchical structures and abstract information [14].
- Computational cost increase with the cubically ($\mathcal{O}(n^3)$) with the number of samples n. This prohibits increasing the size of the surrogate model training set to improve the model accuracy (for example, to train a complex, tailored kernel with many hyperparameters [35]).

Instead, recent advances in variational inference (VI) allow us to approximate the true posterior of Θ in big data problems [37]. We pick an approximate variational distribution over the (latent) model parameters $q_\nu(\Theta)$ with its own variational parameters ν. Now we search for ν that minimizes the divergence to the true posterior which is quantified by the so-called *Kullback-Leibler (KL) divergence*. Thereby the marginalization, i.e. the integration required to calculate the true posterior, is turned into an optimization problem which is often easier to solve. The

approximative distribution of q can be used to form predictions about unseen samples.

3.1.1. Variational inference for training scalable surrogate models

Scalable variational inference methods have been developed both to do approximative inference with Bayesian neural networks (BNN) [13] and with Gaussian process models [38]. We picked one approach of each type (BNN, GP) that can be used "off-the-shelve", that is scalable to 10'000 and more training samples, and that has shown high performance in previous publications [16,17]. They are introduced in the following sections.

The interested reader is referred to [39] for an introduction to Bayesian deep learning approaches. Pearce et al. [40] provides a comparison of various BNN types; different Gaussian process modeltypes which rely on variational inference are explained in [38].

3.2. Deep Bayesian neural networks

The concept of a Bayesian neural network (BNN) is an extension of standard network architectures (e.g. feed-forward neural network, convolutional neural network, or recurrent neural network) to follow the Bayesian modelling paradigm [41]. In a BNN we sample the neural network weights from a prior distribution rather than having a single fixed value as in normal neural networks, for example, from a Gaussian $\Theta \sim N(0, I)$ [39]. Instead of optimising the network weights directly, we average over all possible weights, called marginalisation. Given the stochastic output of the BNN $f^{\Theta}(x)$, we receive a model likelihood $p(y|f^{\Theta}(x))$. Based on the dataset D, Bayesian inference is used to compute the posterior over the weights $p(\Theta|X, Y)$. This posterior captures the set of all plausible model parameters. This distribution allows predictions on unseen data.

As mentioned above the exact posterior is intractable, and different approximations exist [15,40]. In these approximate inference techniques, the posterior $p(\Theta|X, Y)$ is fitted with a simple distribution $q(\Theta)$. Here we consider the Dropout variational inference approach as it has shown great performance when benchmarked against other methods [15,17].

3.2.1. Dropout variational inference

Dropout variational inference is a variational inference approach, i.e. it allows to find a $q_v^*(\Theta)$ that minimises the Kullback-Leibler divergence to the true model posterior, that neither requires to change the architecture of common network architectures nor to change the optimisation algorithm for training the network [39]. The inference of the posterior is done by training a model which uses stochastic dropout on every neuron layer [42] (see Fig. 5). This stochastic dropout is also used to remove neurons when performing predictions. By repeating the predictions (stochastic forward passes), we create a distribution of outputs, which was shown to minimize the KL divergence [39].

This KL divergence objective is formally given in the following, where we approximate $p(\Theta|X, Y)$ with $q(\Theta)$ [13,39]:

$$Ł(\Theta, p) = -\frac{1}{N}\sum_{i=1}^{N} \log\, p(y_i|f^{\widehat{\Theta_i}}(x_i)) + \frac{1-p}{2N}||\theta||2 \tag{3}$$

with N data points, dropout probability p, weight samples $\widehat{\Theta_i} \sim q_v^*(\Theta)$, and θ the set of the sample distribution's parameters to be optimised (weight matrices in the dropout case). Note that for each data point in the training set dropout is applied, which provides us with N samples of Θ_i.

When performing dropout variational inference the T stochastic forward passes provide us with the epistemic uncertainty given by the variance $Var(y)$:

$$Var(y) \approx \frac{1}{T}\sum_{t=1}^{T} f^{\widehat{\Theta_t}}(x)^T f^{\widehat{\Theta_t}}(x_t) - E(y)^T E(y) \tag{4}$$

with predictions in this epistemic model done by approximating the predictive mean: $E(y) \approx \frac{1}{T}\sum_{t=1}^{T} f^{\widehat{\Theta_t}}(x)$. Note that in this formulation we assumed no noise inherent in the data and therefore, $Var(y)$ is zero when we have no parameter uncertainty.

3.3. Gaussian processes in the big data regime

Gaussian Processes models are attractive for non-parametric Bayesian modelling [35]. They use a Gaussian Process prior for a stochastic, latent function f to describe the relationship between X and Y (see Fig. 5). The function values $f(x)$ are assumed to be sampled from that Gaussian with zero mean and covariance matrix K, i.e. $f \sim \mathcal{N}(0, K)$. The choice of covariance function impacts various aspects of the GP model and also determines which model parameters Θ to be tuned. These model parameters are optimized when training the GP model.

However, given the above-mentioned limitations of standard Gaussian Process models (see Section 3.1), sparse GP approximations have been developed to handle large datasets by lowering the computational complexity to $\mathcal{O}(nm^2)$ [38,43].[4] They rely on the use of inducing variables (or pseudo-inputs), i.e. a reduced set of latent variables with size $m << n$ to represent the actual data set D with n samples. The m inducing points are GP realisations $u = f(z)$ at the inducing locations Z which are in the same space as the observed inputs X (but not necessarily part of X). When training the SVGP, the locations of the inducing points Z and the covariance parameters Θ are optimally chosen to minimize the KL divergence. Important is that the locations Z are parameters to shape the variational approximate distribution $q(f)$, rather than being part of the model parameters Θ, i.e. the covariance function with parameters Θ are calculated *for* the inducing locations Z.

In comparison to sparse GPs [43], stochastic variational GPs [16] allow mini-batch training which further reduces computational complexity to $\mathcal{O}(n_{batch}m^2)$. Since [16] and others, multi-layered deep Gaussian Process models have been developed, too, but are not considered in this study as our case study data set is still of limited size and complexity [14,44]. However, our SVGP model may be regarded as a one-layered deep GP [45].

4. Case study: surrogate models for the design of net-zero energy buildings

4.1. Objective

We use a case study on a popular topic in the building domain, the design of buildings with net-zero energy demand, to train and assess the two Bayesian model types introduced above. It shall serve as an example showcasing the use of both model types for building surrogate modelling, but should not be considered as an exhaustive comparison of the two. For that purpose the reader is referred to other studies instead, e.g. [17,44].

4.2. Case study building

We emulate the simulation outcomes of one archetype building contained in the Net-Zero navigator project [5]. As part of the Net-Zero navigator project, building simulation surrogate models are hosted on a web-platform which allows users to receive building energy consumption of archetype buildings given a large set of building design parameters in real time. So far the platform relied on common deterministic neural network surrogates, whose building performance estimation accuracy was validated on separate building designs not contained in the

[4] This blog post provides a summary on the history on sparse Gaussian Process models: https://www.prowler.io/blog/sparse-gps-approximate-the-posterior-not-the-model.

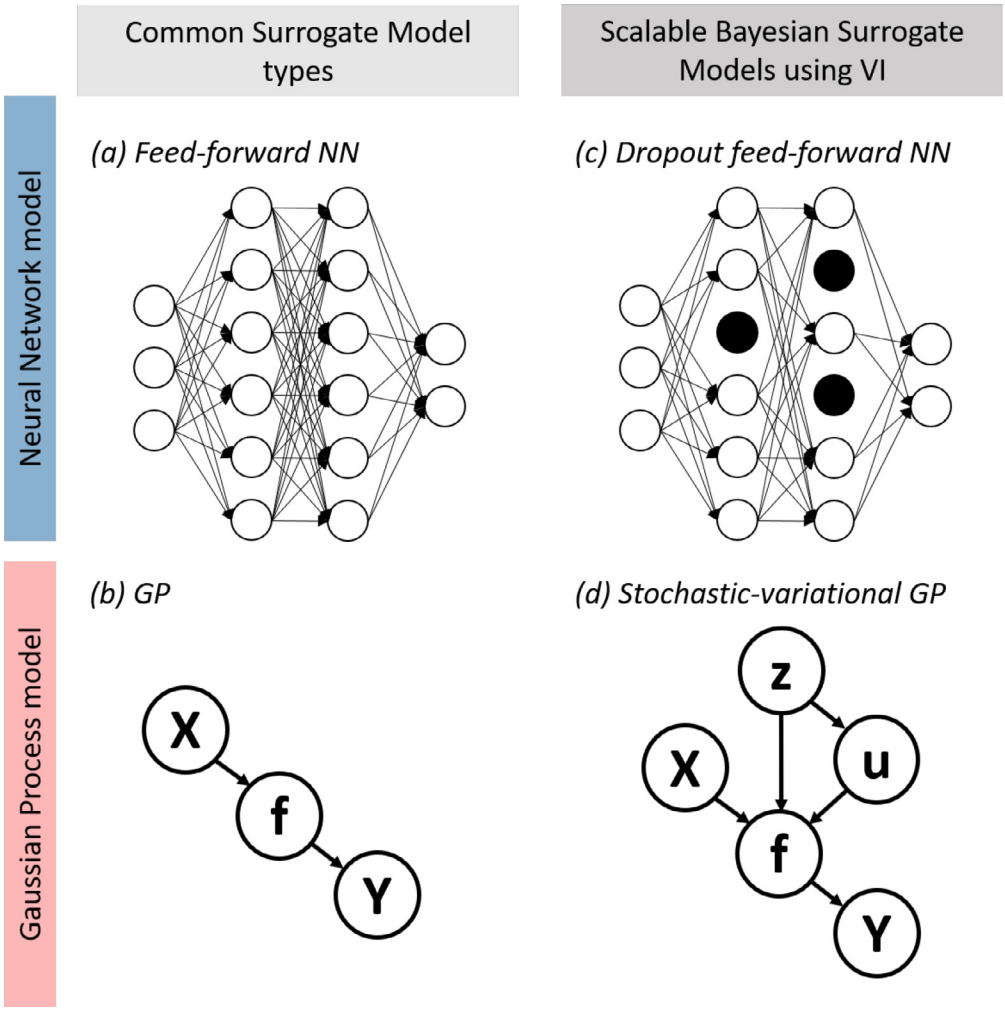


Fig. 5. Considered variational-inference approaches to turn existing surrogate modelling architectures into scalable Bayesian models [15,16].

training data. All the simulation runs for training and testing were collected using the well-known building performance assessment program EnergyPlus [46]. Currently, deterministic surrogate models are used.

In this case study, we build a surrogate model of a medium office archetype building, where 35 design parameters are free to choose and the building energy performance is quantified by 12 separate performance metrics (see Fig. 6). The office architecture is based on work from the US DOE Canmet-Energy which derived commercial prototype building models. The development of the parameter set, the choice of performance metrics, and software to generate the (parametric) simulation data set, however, was developed individually for that project, where the parameter ranges are directly based on requirements in the Canadian building sector [47]. The mechanical systems are parametrized to capture a wide variety of configurations allowing direct manipulation of the air-side system (incl. heat recovery ventilation, various pump efficiencies) and plant equipment performance of various systems (heat pump, electric resistance heater, biogas furnace, natural gas furnace, air conditioning system). This allows us to explore a large HVAC system design space on a high-level (incl. multi-system setups). All details on the building may be found in [5].

4.2.1. Data set and transformations

We sample the design space using 10'000 simulation runs, where the individual parameter combinations in the dataset are picked using the space-filling Latin-Hypercube-sampling (LHS) [31]. Similarly, we run additional 3000 simulations and use it as a separate test set. The number of simulations runs required to fit an accurate surrogate model was previously studied in [5], where it was found that 10'000 runs are suitable for the considered building. Each building simulation run took approximately 2 min and 10 s using 1 CPU and 4 GB RAM, but varied depending on the parameter choices.

Prior to training, we standardized the uniformly distributed inputs with different ranges to be normally distributed with zero mean. Furthermore, we transformed the 12 output variables to also be close to a normal distribution. Therefore, adaptive Box-Cox transformations was applied [48]. It adaptively finds transformation parameters to transform various kinds of distributions (here of 12 different outputs) to normal distributions. This, in particular, increased the accuracy of the multi-output neural network compared to other transformations.

4.3. Model architectures

In this section we provide details on the dropout Bayesian neural network and the stochastic variational Gaussian Process model we trained to emulate the simulation model of the case study building.

4.3.1. BNN model architecture and implementation

We implemented a dropout neural network using the Keras Tensorflow API [49,50] based on the work from Gal and Gahramani [15]. Our network is a feed-forward neural network with 2 hidden layers of 512 neurons which are activated with a leaky rectified linear (ReLU) function. Training was done within 1200 epochs using a

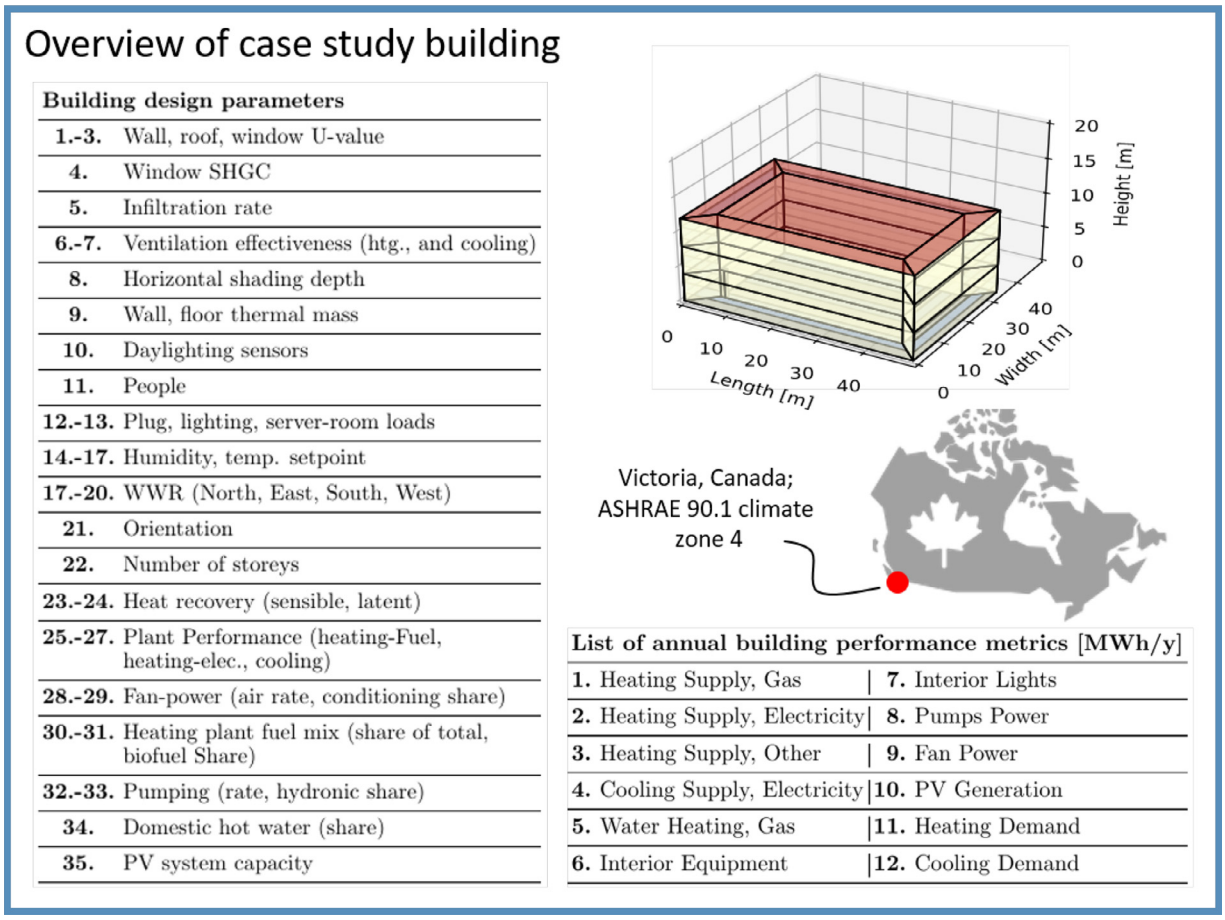


Fig. 6. Overview of the case study building. The building design parameters correspond to the surrogate model inputs and the annual performance metrics to the surrogate model outputs.

batch size of 128 samples. A dropout rate of 5% was set. All mentioned parameters ($n_{layers} \in [1, 2, 3]$, $n_{neurons} = [256, 512, 1024]$, dropout rate $\in [5\%, 10\%, 20\%]$) were analysed in a 5-fold cross-validation. The model with the highest accuracy on the test set was picked. Furthermore, we analysed the impact of the dropout rate on the uncertainty quality (see Section 4.4), but no significant change in the performance was observed, which agrees with the observation from [15], that the uncertainty estimates of models, that use different dropout rates, converge with the training progress.

4.3.2. GP model architecture and implementation

We built a stochastic variational Gaussian Process model based on [16] using the GPy implementation [51]. The final model has a Matern32 covariance function with a fixed noise term ($\approx 0.001\%$ of the mean absolute value of the respective output) and it uses a Gaussian likelihood function. We applied one separate lengthscale per output for the covariance function. Our sparse Gaussian process model used 400 inducing points, which we initialized randomly drawing from a uniform distribution. Training was performed on mini-batches of 100 samples using the Adadelta optimizer.

The covariance function was picked after running a 5-fold cross validation (both squared-exponential, and Matern32 kernels were considered). Although the observed dataset is deterministic, we considered a fixed noise level in the model ($\approx 0.001\%$ of the mean absolute value of the outputs) as it produced much more accurate models. This implies that variance of the one layered Gaussian process model in [16] is too small and a deep Gaussian process may be a better choice for our problem.

4.4. Evaluation criteria

We evaluate the models with regard to multiple objectives: (i) the model accuracy, (ii) uncertainty accuracy, (iii) the effectiveness of uncertainty-estimate-based issue-raising.

4.4.1. R^2 score, $MAPE$ and APE_{90} score to quantify prediction accuracy

Our error metrics cover two often used metrics in the field, i.e. the R^2 [11] and the Mean Absolute Percentage Error (MAPE) [52].

$$R^2(Y, \hat{Y}) = 1\text{-}\frac{\sum_{i=1}^{n}(y_i\text{-}\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i\text{-}\bar{Y})^2} \tag{5}$$

$$\text{MAPE}(Y, \hat{Y}) = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i\text{-}\hat{y}_i|}{y_i} \tag{6}$$

where $\hat{Y}$ corresponds to the matrix of predicted values, Y is the matrix of simulated building performance values. When the error term, $Y\text{-}\hat{Y}$ approaches zero, R^2 approaches one, and MAPE goes to zero.

The given two error metrics provide insight into the overall performance of the models. However, they may disguise large errors which occur for few samples. Therefore, we added the APE_{90} error. It represents the 90th percentile of the absolute errors sorted by ascending magnitude, and therefore, allows to estimate maximum model errors while accounting for possible occurrences of outliers.

4.4.2. Accuracy of the uncertainty estimate

In a well-calibrated Bayesian model the uncertainty estimates capture the true data distribution, for example, a 95% posterior confidence

interval also contains the true simulation outcome in 95% of the times [53]. Quantifying the level of calibration is a well-known concept in classification [54] but has also been used for regression problems recently [53,55].

Formally, we say that the uncertainty estimates of the surrogate model are well-calibrated if

$$\frac{\sum_{n=1}^{N}\{y_t \leq F_t^{-1}(p)\}}{N} \rightarrow p \text{ for all p} \in [0,1] \tag{7}$$

where F_t is the cumulated density function targeting y_t and $F_t^{-1} = inf\{y : p \leq F_t(y_t)\}$ is the quantile function. Here we consider each prediction as a standard, symmetric Gaussian distribution $\mathcal{N}(\mu(X), \sigma(X))$.[5] The confidence intervals can be computed using the inverse cumulated density function. To assess the calibration quality, we count the fraction of observations in the test data falling in the prediction confidence intervals derived from the quantile function (see Fig. 8, left).

We show the level of calibration of the Bayesian models in Fig. 8 (left), where perfectly calibrated uncertainty estimates would be aligned with the diagonal. To quantitatively compare different calibration curves, one can also compute the absolute difference between the confidence curve and the diagonal, called the calibration error or the area under the curve (AUC) [55]. The problem of assessing the calibration quality based on the calibration plot is that it can suggest perfect quality with homoscedastic uncertainty estimates, i.e. constant uncertainty estimates for any input. Therefore, we also quantify the *sharpness* of the uncertainty estimates by calculating the overall variance in the uncertainty [53] (see Section 5).

4.4.3. Discard-ranking to quantify the effectiveness of uncertainty estimates for surrogate model application

While having accurate uncertainty estimates is the one thing, in building surrogate modelling we are mostly concerned with warning model users, when the model is uncertain and recommend to rather run a simulation instead (see Fig. 2). Therefore, we derive a ranking of the samples in the test set based on the magnitude of their uncertainty. This provides two conclusions. First, if it strongly overlaps with the actual surrogate model error the uncertainty estimates are an effective heteroscedastic warning mechanism. Second, we can use the ranking to calculate how much the average error can be reduced when referring a certain percentage of most uncertain samples (here 10% or 20%) to the high-fidelity simulation program than processing it with a surrogate model.

Both aspects are addressed when plotting the mean error computed on discrete percentiles of the test data, where the test data is sorted by the magnitude of the uncertainty. We can compare that curve to the mean error computed using test data sorted by the magnitude of the computed error (oracle ranking). A large distance between the two curves can tell us that the surrogate's uncertainty estimates are not helpful to predict when it is inaccurate. Furthermore, by looking at the slope of the curve, we can see by how much the mean error can be reduced if we discard all samples with uncertainties above a certain threshold.

5. Results

In this section, we show the results of the case study where we derived uncertainty-aware surrogate models to replaced building energy simulation models.

In the case study, we trained two different Bayesian machine learning models to provide epistemic uncertainty estimates, i.e. a deep Bayesian dropout neural network (here abbreviated by BNN) and a stochastic variational Gaussian Process model (SVGP) approach. We scrutinize the performance of both approaches by comparing their predictive accuracy, by comparing the quality of the uncertainty estimates, and by quantifying how effectively the uncertainty estimates allow us to identify possible surrogate prediction errors.

5.1. Model accuracy and uncertainty quality

5.1.1. Accuracy

We benchmark the accuracy of the two model types, dropout neural networks and SVGP models. The performance was quantified using three performance metrics as introduced above (see Section 4.4). Each model was trained five times to generate robust results. The results are shown in Fig. 7 and Table 1 in the Appendix; details on the model layout and training process can be found in Sections 4.3.1 and 4.3.2).

Both considered models reach an accuracy of $R^2 > 0.97$ on all the outputs, when predicting building performance of buildings contained in the test data. The BNN is more accurate with $R^2 \geqslant 0.99$ (also see Table 1). Mean percentage errors of $MAPE < 13.2\%$ for the GP model and $MAPE < 9.82\%$ for BNN were found. The largest errors occur when estimating the energy demand provided by different heating sources (i.e. the different fuel types), and the air-side system energy demand. Small surrogate model errors are found for the other building performance targets like the photovoltaic (PV) generation, or energy demand for interior lights and equipment.

To prove robustness of surrogate model estimates, we specifically look at the largest errors it produces. Therefore, we complement our analysis of the mean absolute percentage error with an analysis of the distribution of the absolute percentage errors observed for each sample in the test data. We extract the 90-th percentile of the distribution as a proxy of the largest error found while ignoring outliers. We abbreviate this metric with APE_{90}. APE_{90} errors are found reaching up to 22.3% (30.5%) for the BNN model (GP model), highlighting the demand for increasing the robustness.

5.1.2. Uncertainty calibration

When uncertainty estimates are perfectly calibrated, the derived confidence interval, e.g. the 90% confidence interval, contains the true outcome in the right number of cases, i.e. 90% of the times for the given example. This is illustrated in Fig. 8, where we counted for how many times the true simulation outcome was contained in the estimated confidence interval. With a perfectly calibrated Bayesian model the estimated confidence and fraction of the test samples within that interval should perfectly align (dashed line). The region below the dashed line indicates an overly confident model (i.e. confidence bands are too narrow), the region above the dashed line means that the model is too careful having too large confidence bands.

We find that the BNN model is well-calibrated, while the GP model is overly confident (Fig. 8, left). The low quality of uncertainty estimates of the GP model can also be seen on the right, where we display the distribution of all uncertainty estimates collected for predictions of the test data samples. The average magnitude of uncertainty in the GP model indicates its too high confidence, and the small width of the distribution indicates that the uncertainty estimates tend to be homoscedastic, i.e. a similar uncertainty is predicted independently of the model inputs. This width of the distribution is also called the *sharpness* of uncertainty estimates (see Section 4.4). In case of the BNN, the sharpness is better and uncertainty estimates depict a significant level of variance.

We can conclude that the uncertainty estimates of the BNN are well-calibrated and provide heteroscedastic uncertainty estimates.

5.1.3. Using uncertainty estimates to increase robustness

In this section we study how effective the epistemic uncertainty estimates are to predict inaccuracies of the surrogate model.

The concept is as follows. We sort the uncertainty estimates on the test data by scale, where we assume that surrogate model estimates are more inaccurate when it is uncertain. The samples with high uncertainty will be evaluated by the high fidelity simulation program instead of the surrogate model (see Fig. 2). As a consequence, the surrogate model

[5] This is not necessarily true and possibly a recalibration step is required [53].

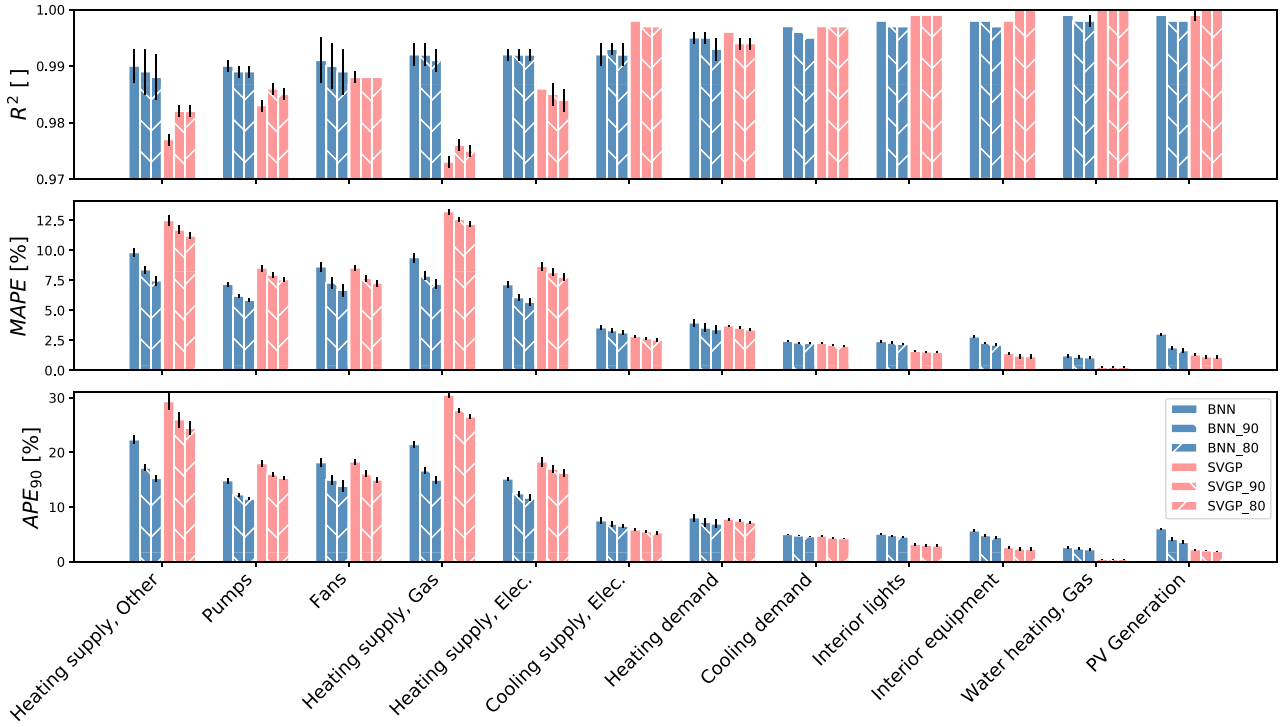


Fig. 7. Summary of results on the use of deep, uncertainty-aware surrogate models. The plot shows the accuracy, quantified using three different error metrics, of both Bayesian learning approaches for all twelve outputs considered in the case study. The figures also include performance metrics when we use the uncertainty estimates to identify error-prone samples in the test data (textured bars, for details see Section 5.1.3).

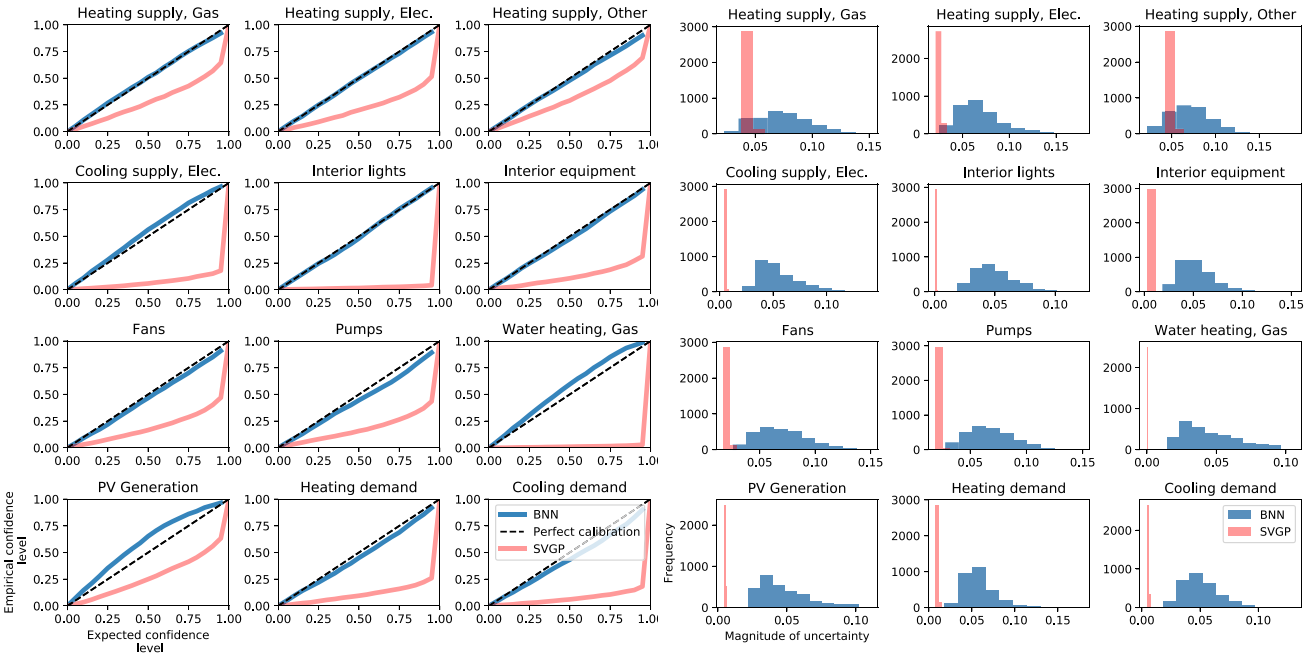


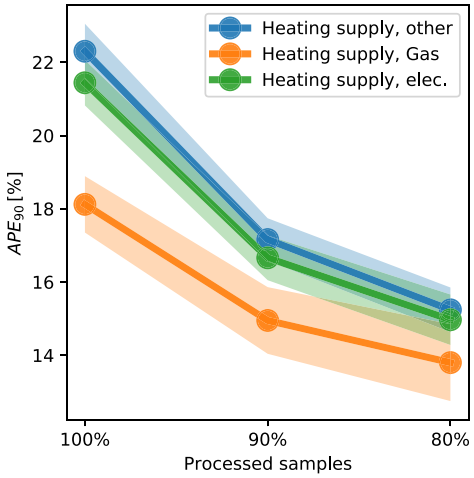
Fig. 8. Visualization of the quality of uncertainty estimates of the BNN and the SVGP. The quality is quantified by how well-calibrated and sharp the uncertainty estimates are. In both regards, the BNN outperforms the SVGP in this study.

user, here a building designer, is provided with estimates produced by the surrogate model only when it has high confidence, and with actual simulation results when the surrogate model has low confidence. The number of samples processed by the computationally expensive simulation model should be traded-off against an increase in runtime. Here, we handle this trade-off by defining an uncertainty threshold above which the simulation program is queried.

We define this threshold as the 90th- or 80th-percentile of all uncertainties observed on our test data set. The rationale behind that choice is that only 10% (or 20%) of all samples are transferred to the slow

Table 1
Results of the accuracy of the Bayesian models.

	R^2		$MAPE$		$APE90$	
	BNN	SVGP	BNN	SVGP	BNN	SVGP
Pumps [MWh/y]	**0.990** ± 0.001	0.983 ± 0.001	**7.180** ± 0.180	8.530 ± 0.260	**14.830** ± 0.510	17.950 ± 0.610
Heating supply, Other [MWh/y]	**0.990** ± 0.003	0.977 ± 0.001	**9.820** ± 0.350	12.490 ± 0.430	**22.300** ± 0.750	29.300 ± 1.480
Fans [MWh/y]	**0.991** ± 0.004	0.988 ± 0.001	8.630 ± 0.380	**8.530** ± 0.250	**18.120** ± 0.770	18.280 ± 0.540
Heating supply, Elec. [MWh/y]	**0.992** ± 0.001	0.986 ± 0.000	**7.150** ± 0.290	8.670 ± 0.360	**15.130** ± 0.290	18.260 ± 0.900
Heating supply, Gas [MWh/y]	**0.992** ± 0.002	0.973 ± 0.001	**9.400** ± 0.380	13.230 ± 0.220	**21.440** ± 0.620	30.480 ± 0.520
Cooling supply, Elec. [MWh/y]	0.992 ± 0.002	**0.998** ± 0.000	3.550 ± 0.200	**2.820** ± 0.100	7.490 ± 0.560	**5.820** ± 0.200
Heating demand [MWh/y]	0.995 ± 0.001	**0.996** ± 0.000	3.960 ± 0.330	**3.710** ± 0.080	8.040 ± 0.710	**7.800** ± 0.250
Cooling demand [MWh/y]	**0.997** ± 0.000	**0.997** ± 0.000	2.440 ± 0.050	**2.270** ± 0.060	4.980 ± 0.090	**4.700** ± 0.110
Interior lights [MWh/y]	0.998 ± 0.000	**0.999** ± 0.000	2.410 ± 0.100	**1.590** ± 0.080	5.050 ± 0.160	**3.150** ± 0.270
Interior equipment [MWh/y]	**0.998** ± 0.000	**0.998** ± 0.000	2.790 ± 0.100	**1.410** ± 0.120	5.650 ± 0.200	**2.600** ± 0.250
Water heating, Gas [MWh/y]	0.999 ± 0.000	**1.000** ± 0.000	1.220 ± 0.130	**0.250** ± 0.070	2.590 ± 0.260	**0.430** ± 0.090
PV Generation [MWh/y]	**0.999** ± 0.000	**0.999** ± 0.001	3.030 ± 0.090	**1.290** ± 0.090	6.040 ± 0.100	**2.200** ± 0.150



	Share of samples processed	$\Delta MAPE$ 90%	$\Delta MAPE$ 80%	ΔAPE_{90} 90%	ΔAPE_{90} 80%
BNN model outputs [MWh/y]	Heating supply, Gas	-16.2	-23.5	-22.3	-30.2
	Heating supply, Elec.	-15.1	-20.7	-17.6	-22.9
	Heating supply, Other	-14.7	-23.8	-23.0	-31.7
	Cooling supply, Elec.	-6.5	-11.3	-7.6	-12.7
	Interior lights	-5.0	-9.5	-5.1	-9.7
	Interior equipment	-17.9	-23.7	-15.4	-21.2
	Fans	-15.4	-22.5	-17.5	-23.8
	Pumps	-13.6	-18.5	-17.2	-22.6
	Water heating, Gas	-9.0	-13.9	-9.3	-14.7
	PV Generation	-37.3	-45.2	-31.8	-41.6
	Heating demand	-10.4	-13.9	-9.7	-13.7
	Cooling demand	-5.3	-7.8	-5.4	-7.4

Fig. 9. Recorded surrogate model error reduction after transferring uncertain samples to the high-fidelity simulation model. The data shows the error if either 100%, 90% or 80% of the building design samples are processed by the surrogate model and the rest processed by the high-fidelity model. In that way, the average error of samples processed by surrogate models can be decreased (here quantified by the 90-percentile absolute percentage error).

simulation program. Finding a suitable threshold is more difficult and should also be based on the preferences of the building designer.

In Fig. 9, the decrease in the error of the surrogate model predictions is illustrated for the three target variables covering the heat supply of different fuel sources. These targets produced the largest errors (see Section 5.1.1) and thus, we focus on increasing the surrogate robustness particularly for them. Discarding the 10% samples with the highest uncertainty on the test data, we can decrease the APE_{90} error in estimating the annual heating supply with a gas furnace from 21.44% to 16.66%.[6] This is equivalent to a reduction of ≈ 22%.

The $MAPE$ error on the other surrogate model outputs was reduced by 4% to 18%, and the APE_{90} by 5% to 25% (see Fig. 9). In particular, the significant reduction of the APE_{90} error proofs the increase in the robustness of the surrogate model predictions.

6. Discussion

Surrogate models have shown to help architects and building designers to rapidly assess the energy performance of their designs [9]. However, by being only approximative, concerns about the robustness of the surrogate model accuracy arise. A Bayesian approach for surrogate modelling, allows to not only provide a performance estimate but also inform about the confidence of the approximating surrogate model and potentially, to identify parts of the design space where the surrogate model may provide inaccurate results.

This first analysis of the use of Bayesian surrogate models revealed essential properties on the robustness of surrogate models, and how Bayesian modelling can be an aid for effective reasoning on the energy performance of buildings under the epistemic uncertainty of surrogates. The goal was to augment surrogates such that we can maintain the benefits of surrogate models while minimizing the risk associated with the uncertainty of surrogate models.

6.1. Lacking robustness of surrogate models

Surrogate model accuracy is often reported with error metrics like the R^2 or $MAPE$ scores. They are important but can be deceiving. A high coefficient of explained variance (R^2) or a low mean absolute percentage error $MAPE$, may disguise that the surrogate may produce quite large errors in certain fractions of the design space. For example, we found that the 90-percentile absolute percentage error can be as high as 22.3% although an $R^2 = 0.99$ suggests very high performance (see Table 1). This motivates, that indeed measures to identify surrogate inaccuracies could lessen the risk associated with surrogate modelling.

[6] To calculate these errors, we exclude the 10% or 20% most important samples from Eqs. 5 and 6. For example, the 16.66% error was computed on the 90% remaining samples in the test set.

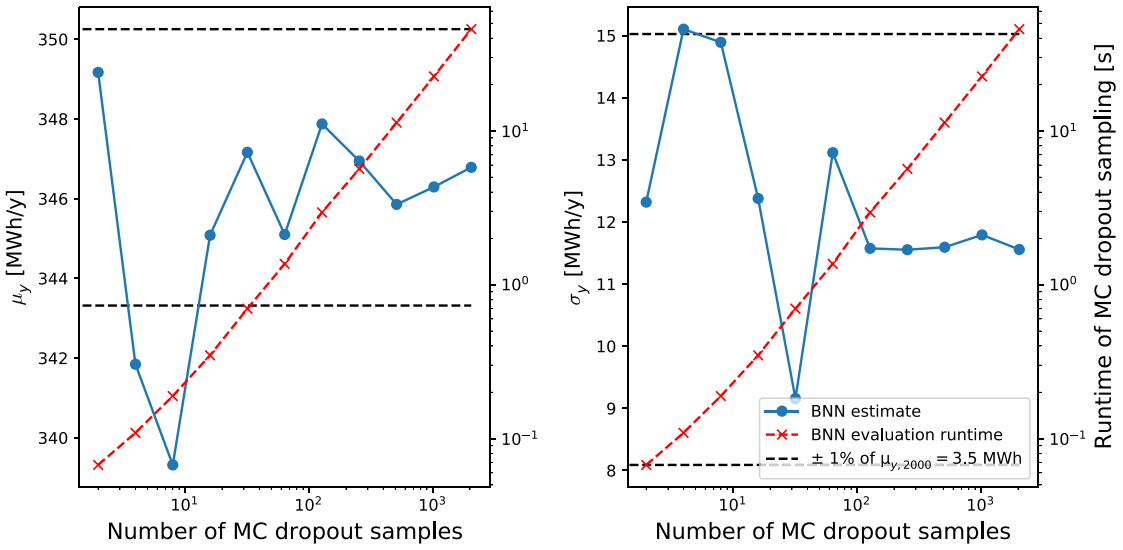


Fig. 10. Convergence of BNN estimates with an increasing number of Monte Carlo dropout samples. The plot shows BNN heating demand estimates and uncertainty estimates with increasing number of MC samples (see case study in Section 4). Both approximately converge after conducting 30 random dropout runs, which takes around 0.8 s (without parallelization).

6.2. Bayesian learning to express surrogate confidence

Results on the quality of uncertainty estimates of the dropout neural network validated that it can be used to effectively express confidence on its predictions, e.g. one can formulate that the heating demand for a building with a wall of $1m$ thickness is between $220MWh/year$ and $230MWh/year$ with a 90% confidence (see Fig. 4).

On the other hand, while being almost as accurate as the neural network model, we found that the stochastic variational Gaussian Process model produces miscalibrated uncertainty estimates. Please note, that this finding cannot be generalized as methods exist to calibrate uncalibrated estimates [53], and in other studies deep Gaussian process models were found to produce a larger variance in the uncertainty estimates [44]. Nonetheless, the results on the SVGP models highlight that assessing the quality of Bayesian uncertainty estimates is important.

6.3. Practical issues of Bayesian surrogate models

We leveraged the uncertainty estimates of the BNN to raise warnings when the surrogate model is highly uncertain. By defining a threshold, here the 90-percentile or 80-percentile of the uncertainty estimates on the test data, we could reduce the APE_{90} error by up to 40%. This is a significant first step towards the hybridization of fast, low-fidelity, and slow, high-fidelity models.

Still, practical issues have to be solved. For example, the question arises on how to implement the routing between the surrogate model and high-fidelity model runs. Simulations could be carried out in the background while the user would be working with the uncertain surrogate model estimates as a start. In our case the results would be updated after 2 minutes and 10 seconds, which corresponds to the approximate runtime of one simulation.

Another issue is that the computational cost of evaluating a Bayesian model increases compared to a deterministic surrogate model. When using dropout BNNs, we perform Monte Carlo (MC) dropout, i.e. we repeatedly evaluate the BNN whereas in each run the set of "dropped" neurons changes and therewith, the outputs of the network change. Mean μ and standard deviation σ of the estimates converge with increasing numbers of MC evaluations, which is shown in Fig. 10. We performed between 10 and 2000 MC evaluations and reported the mean and the standard deviation of the resulting estimates. We consider both mean and standard deviation to have converged, when they remain within a band of ± 1% of the mean we observed after 2000 MC dropout runs.

In the plot we visualized the convergence of the heating demand estimates for a single building design. The plot implies that it takes approximately 0.8 s, which corresponds to 30 MC dropout runs, for both the mean and uncertainty estimates to converge. Without parallelization, this would mean that MC dropout sampling of a BNN is 30 times slower than the evaluation of a common feed-forward neural network, and it would prevent interactive building design processes. However, the independent MC dropout runs can easily be parallelized to multiple cores. Please note that the convergence rate depends on the specific building design parameters (surrogate model inputs) or the considered building performance output (surrogate model outputs). A first heuristic check for various inputs and outputs indicated that estimates always converged within 100 or less MC dropout runs.

These and other questions have to be studied in more detail before integrating Bayesian surrogate models into software products for building designers.

6.4. Accuracy of the Bayesian model compared to a deterministic surrogate model

We can compare the results of this study to a non-Bayesian feed-forward neural network trained on the same dataset (see Table 2 in the Appendix). Details on the non-bayesian network used can be found in [5]. It has a very similar layout to the dropout BNN (2 hidden layers with 512 neurons, leaky rectified linear unit activation function) and was trained using the same cost function and optimizer (1200 training epochs with Adam optimizer).

The R^2, $MAPE$ and APE_90 scores of the deterministic model computed on the test data are better for most outputs when no uncertainty based sample filtering is applied (see Table 2). However, when using uncertainty thresholds the Bayesian model produces lower $MAPE$ and APE_90 errors proposing that the BNN is a useful means to increase the robustness of surrogate models.[7]

[7] Here, we used a uniformly distributed set of building design samples as our test data. However, this may not be representative of actual design processes. In future, a comparison of both neural network types (Bayesian surrogate model,

Table 2
Comparison of Bayesian dropout neural network (BNN) and non-bayesian deterministic neural network (ANN). The performance of the dropout neural network (BNN) is provided with and without the application of uncertainty-based thresholding (90%/80%).

(i) R^2-score				
	ANN	BNN	$BNN_{90\%}$	$BNN_{80\%}$
Pumps [MWh/y]	**0.992** ± 0.000	0.990 ± 0.001	0.989 ± 0.001	0.989 ± 0.001
Heating supply, Other [MWh/y]	**0.995** ± 0.001	0.990 ± 0.003	0.989 ± 0.004	0.988 ± 0.004
Fans [MWh/y]	**0.994** ± 0.002	0.991 ± 0.004	0.990 ± 0.004	0.989 ± 0.004
Heating supply, Elec. [MWh/y]	**0.994** ± 0.000	0.992 ± 0.001	0.992 ± 0.001	0.992 ± 0.001
Heating supply, Gas [MWh/y]	**0.995** ± 0.001	0.992 ± 0.002	0.992 ± 0.002	0.991 ± 0.002
Cooling supply, Elec. [MWh/y]	**0.994** ± 0.001	0.992 ± 0.002	0.993 ± 0.001	0.992 ± 0.002
Heating demand [MWh/y]	**0.996** ± 0.000	0.995 ± 0.001	0.995 ± 0.001	0.993 ± 0.002
Cooling demand [MWh/y]	**0.997** ± 0.000	0.997 ± 0.000	0.996 ± 0.000	0.995 ± 0.000
Interior lights [MWh/y]	**0.999** ± 0.000	0.998 ± 0.000	0.997 ± 0.000	0.997 ± 0.000
Interior equipment [MWh/y]	**0.999** ± 0.000	0.998 ± 0.000	0.998 ± 0.000	0.997 ± 0.000
Water heating, Gas [MWh/y]	**1.000** ± 0.000	0.999 ± 0.000	0.998 ± 0.000	0.998 ± 0.001
PV Generation [MWh/y]	**1.000** ± 0.000	0.999 ± 0.000	0.998 ± 0.000	0.998 ± 0.000
(ii) MAPE				
	ANN	BNN	$BNN_{90\%}$	$BNN_{80\%}$
Pumps [MWh/y]	6.480 ± 0.170	7.180 ± 0.180	6.200 ± 0.130	**5.850** ± 0.130
Heating supply, Other [MWh/y]	8.550 ± 0.630	9.820 ± 0.350	8.380 ± 0.310	**7.480** ± 0.410
Fans [MWh/y]	7.610 ± 1.000	8.630 ± 0.380	7.300 ± 0.470	**6.690** ± 0.540
Heating supply, Elec. [MWh/y]	6.530 ± 0.370	7.150 ± 0.290	6.070 ± 0.270	**5.670** ± 0.320
Heating supply, Gas [MWh/y]	8.040 ± 0.220	9.400 ± 0.380	7.880 ± 0.370	**7.190** ± 0.400
Cooling supply, Elec. [MWh/y]	3.280 ± 0.260	3.550 ± 0.200	3.320 ± 0.200	**3.150** ± 0.170
Heating demand [MWh/y]	3.710 ± 0.290	3.960 ± 0.330	3.550 ± 0.370	**3.410** ± 0.370
Cooling demand [MWh/y]	**2.240** ± 0.160	2.440 ± 0.050	2.310 ± 0.050	2.250 ± 0.060
Interior lights [MWh/y]	**1.830** ± 0.170	2.410 ± 0.100	2.290 ± 0.090	2.180 ± 0.070
Interior equipment [MWh/y]	2.810 ± 0.390	2.790 ± 0.100	2.290 ± 0.080	**2.130** ± 0.090
Water heating, Gas [MWh/y]	**0.660** ± 0.060	1.220 ± 0.130	1.110 ± 0.130	1.050 ± 0.120
PV Generation [MWh/y]	**1.650** ± 0.120	3.030 ± 0.090	1.900 ± 0.150	1.660 ± 0.180
(iii) APE_{90}				
	ANN	BNN	$BNN_{90\%}$	$BNN_{80\%}$
Pumps [MWh/y]	12.450 ± 0.530	14.830 ± 0.510	12.280 ± 0.310	**11.480** ± 0.230
Heating supply, Other [MWh/y]	20.400 ± 1.480	22.300 ± 0.750	17.160 ± 0.580	**15.240** ± 0.610
Fans [MWh/y]	15.810 ± 1.540	18.120 ± 0.770	14.950 ± 0.910	**13.800** ± 1.050
Heating supply, Elec. [MWh/y]	13.790 ± 0.810	15.130 ± 0.290	12.470 ± 0.490	**11.670** ± 0.640
Heating supply, Gas [MWh/y]	18.320 ± 0.640	21.440 ± 0.620	16.660 ± 0.610	**14.970** ± 0.690
Cooling supply, Elec. [MWh/y]	6.780 ± 0.560	7.490 ± 0.560	6.920 ± 0.460	**6.540** ± 0.320
Heating demand [MWh/y]	7.670 ± 0.550	8.040 ± 0.710	7.260 ± 0.740	**6.940** ± 0.770
Cooling demand [MWh/y]	4.620 ± 0.300	4.980 ± 0.090	4.710 ± 0.090	**4.610** ± 0.090
Interior lights [MWh/y]	**3.840** ± 0.330	5.050 ± 0.160	4.790 ± 0.170	4.560 ± 0.170
Interior equipment [MWh/y]	5.320 ± 0.960	5.650 ± 0.200	4.780 ± 0.200	**4.450** ± 0.240
Water heating, Gas [MWh/y]	**1.340** ± 0.100	2.590 ± 0.260	2.350 ± 0.270	2.210 ± 0.250
PV Generation [MWh/y]	**2.460** ± 0.320	6.040 ± 0.100	4.120 ± 0.300	3.530 ± 0.350

7. Conclusion and outlook

In this study we proposed to augment and hybridize physics-based simulation software with Bayesian (deep) learning surrogate models. By quantifying the surrogate model (epistemic) uncertainty, the Bayesian paradigm acknowledges that surrogate models are approximations of original simulation models, and it offers a tool to effectively reason under that incurred uncertainty while exploiting the much faster runtime of surrogate models to produce engineering performance estimates.

In a case study we showcased the application of Bayesian surrogate models for the design of net-zero energy buildings. We found that dropout neural network models provided well-calibrated uncertainty estimates, which can be used to identify building design choices for which the surrogate model produces large errors. The latter enables us to refer those designs to the high-fidelity energy simulation tool to assure accurate estimates for the architect or building designer. That referral process significantly lowered the errors in comparison to a common deterministic surrogate model.

Although all findings are bound to the case study of a building simulation surrogate, results motivate to apply Bayesian learning to other fields where surrogate models are commonly used [19].

In future, we foresee that Bayesian models will allow us to *hybridize* data-driven surrogate models and high-fidelity simulation models [18]. This particularly requires studies on how hybrid models can work in practice in a surrogate model-based design process.

Apart from that, future research could make use of Bayesian surrogate models for *generalizing* surrogate models to cover more building design problems [12,56]. The Bayesian paradigm could help identifying when the surrogate model is used for design problems it was not trained for. Finally, Bayesian learning forms a foundation for adaptively sampling simulation runs, for which the surrogate model is particularly uncertain. This progress, called active learning, will be explored in an upcoming study [57].

Code and Data availability

The entire source code of this work, the EnergyPlus description file (*.idf*) of the building template, and instructions on how to download the data used in this study are available in a GitLab repository.[8]

non-bayesian surrogate model) that takes architectural design preferences into account when choosing the test data should be considered.

[8] https://gitlab.com/energyincities/building_surrogate_modelling

Declaration of Competing Interest

The authors wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgement

This research was supported by grant funding from CANARIE via the BESOS project (CANARIE RS-327).

References

[1] John Dulac CD, Abergel T. Tracking buildings. Tech. Rep.. Internation Energy Agency; 2019. URL: https://www.iea.org/reports/tracking-buildings
[2] Westermann P, Evins R. Surrogate modelling for sustainable building design – a review. Energy Build 2019;198:170–86. doi:10.1016/j.enbuild.2019.05.057.
[3] Jusselme T. Data-driven method for low-carbon building design at early stages. EPF Lausanne; 2020. Ph.D. thesis.
[4] Open Technologies. The building pathfinder. URL http://www.buildingpathfinder.com/.
[5] Paul Westermann, David Rulff, Kevin Cant, Gaelle Faure, Ralph Evins. Net-zero navigator: a platform for interactive net-zero building design using surrogate modellingURL http://www.enerarxiv.org/page/thesis.html?id=1975.
[6] Waibel C, Wortmann T, Evins R, Carmeliet J. Building energy optimization: an extensive benchmark of global search algorithms. Energy Build 2019;187:218–40.
[7] Rivalin L, Stabat P, Marchio D, Caciolo M, Hopquin F. A comparison of methods for uncertainty and sensitivity analysis applied to the energy performance of new commercial buildings. Energy Build 2018;166:489–504.
[8] Hester J, Gregory J, Kirchain R. Sequential early-design guidance for residential single-family buildings using a probabilistic metamodel of energy consumption. Energy and Buildings 2017;134:202–11. doi:10.1016/j.enbuild.2016.10.047. URL <Go to ISI>://WOS:000390624800018
[9] Brown NC. Design performance and designer preference in an interactive, data–driven conceptual building design scenario. Des Stud 2020.
[10] De Wilde P. The gap between predicted and measured energy performance of buildings: a framework for investigation. Autom Constr 2014;41:40–9.
[11] Ostergard T, Jensen RL, Maagaard SE. A comparison of six metamodeling techniques applied to building performance simulations. Applied Energy 2018;211:89–103. doi:10.1016/j.apenergy.2017.10.102. URL <Go to ISI>://WOS:000425075600008
[12] Westermann P, Evins R. Using a deep temporal convolutional network as a building energy surrogate model that spans multiple climate zones. Appl Energy 2020;264:114715.
[13] Kendall A, Gal Y. What uncertainties do we need in Bayesian deep learning for computer vision?. In: Advances in neural information processing systems; 2017. p. 5574–84.
[14] Damianou A, Lawrence N. Deep Gaussian processes. In: Artificial intelligence and statistics; 2013. p. 207–15.
[15] Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: International conference on machine learning; 2016. p. 1050–9.
[16] Hensman J, Fusi N, Lawrence ND. Gaussian processes for big data. In: Uncertainty in artificial intelligence. Citeseer; 2013. p. 282.
[17] Filos A, Farquhar S, Gomez AN, Rudner TG, Kenton Z, Smith L, et al. A systematic comparison of Bayesian deep learning robustness in diabetic retinopathy tasks. arXiv preprint arXiv:1912.10481 2019.
[18] Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J, Carvalhais N, et al. Deep learning and process understanding for data-driven earth system science. Nature 2019;566(7743):195–204.
[19] Wang GG, Shan S. Review of metamodeling techniques in support of engineering design optimization. J MechDes 2007;129(4):370–80.
[20] Ritter F, Geyer P, Borrmann A. Simulation-based decision-making in early design stages. In: 32nd CIB W78 conference, Eindhoven, The Netherlands; 2015. p. 27–9.
[21] Vazquez-Canteli J, Demir AD, Brown J, Nagy Z. Deep neural networks as surrogate models for urban energy simulations. Journal of Physics: Conference Series 2019;1343:012002. IOP Publishing
[22] Prada A, Gasparella A, Baggio P. On the performance of meta-models in building design optimization. Appl Energy 2018;225:814–26.
[23] Eisenhower B, O'Neill Z, Narayanan S, Fonoberov VA, Mezic I. A methodology for meta-model based optimization in building energy models. Energy and Buildings 2012;47:292–301. doi:10.1016/j.enbuild.2011.12.001. URL <Go to ISI>://WOS:000301989800034
[24] Bre F, Roman N, Fachinotti VD. An efficient metamodel-based method to carry out multi-objective building performance optimizations. Energy Build 2020;206:109576.
[25] Hopfe CJ, Hensen JL. Uncertainty analysis in building performance simulation for design support. Energy Build 2011;43(10):2798–805.
[26] Coakley D, Raftery P, Keane M. A review of methods to match building energy simulation models to measured data. RenewSustainEnergy Rev 2014;37:123–41.
[27] Manfren M, Aste N, Moshksar R. Calibration and uncertainty analysis for computer models - a meta-model based approach for integrated building energy simulation. Applied Energy 2013;103:627–41. doi:10.1016/j.apenergy.2012.10.031. URL <Go to ISI>://WOS:000314669500059
[28] Heo Y, Choudhary R, Augenbroe G. Calibration of building energy models for retrofit analysis under uncertainty. Energy Build 2012;47:550–60.
[29] Sokol J, Davila CC, Reinhart CF. Validation of a Bayesian-based method for defining residential archetypes in urban building energy models. Energy Build 2017;134:11–24.
[30] Kristensen MH, Hedegaard RE, Petersen S. Hierarchical calibration of archetypes for urban building energy modeling. Energy Build 2018;175:219–34.
[31] Garud SS, Karimi IA, Kraft M. Design of computer experiments: a review. Comput Chem Eng 2017;106:71–95.
[32] Roman ND, Bre F, Fachinotti VD, Lamberts R. Application and characterization of metamodels based on artificial neural networks for building performance simulation: a systematic review. Energy Build 2020:109972.
[33] Crawley DB, Lawrie LK, Winkelmann FC, Buhl WF, Huang YJ, Pedersen CO, et al. Energyplus: creating a new-generation building energy simulation program. Energy Build 2001;33(4):319–31.
[34] Tian W, Heo Y, De Wilde P, Li Z, Yan D, Park CS, et al. A review of uncertainty analysis in building energy assessment. Renew Sustain Energy Rev 2018;93:285–301.
[35] Rasmussen CE. Gaussian processes in machine learning. In: Advanced lectures on machine learning. Springer; 2004. p. 63–71.
[36] Østergård T, Jensen RL, Maagaard SE. Building simulations supporting decision making in early design–a review. Renewable and Sustainable Energy Reviews 2016;61:187–201. URL https://www.sciencedirect.com/science/article/pii/S136403211600280X
[37] Blei DM, Kucukelbir A, McAuliffe JD. Variational inference: a review for statisticians. J AmStat Assoc 2017;112(518):859–77.
[38] Bauer M, van der Wilk M, Rasmussen CE. Understanding probabilistic sparse Gaussian process approximations. In: Advances in neural information processing systems; 2016. p. 1533–41.
[39] Gal Y. Uncertainty in deep learning. University of Cambridge 2016;1(3).
[40] Pearce T, Zaki M, Brintrup A, Anastassacos N, Neely A. Uncertainty in neural networks: Bayesian ensembling arXiv preprint arXiv:1810.05546
[41] Neal RM. Bayesian learning for neural networks, 118. Springer Science & Business Media; 1995.
[42] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. JMachLearnRes 2014;15(1):1929–58.
[43] Titsias M. Variational learning of inducing variables in sparse gaussian processes. In: Artificial intelligence and statistics; 2009. p. 567–74.
[44] Salimbeni H, Deisenroth M. Doubly stochastic variational inference for deep Gaussian processes. In: Advances in neural information processing systems; 2017. p. 4588–99.
[45] Svendsen DH, Morales-Álvarez P, Ruescas AB, Molina R, Camps-Valls G. Deep Gaussian processes for biogeophysical parameter retrieval and model inversion. ISPRS J Photogramm Remote Sens 2020;166:68–81.
[46] Crawley DB, Pedersen CO, Lawrie LK, Winkelmann FC. Energyplus: energy simulation program. ASHRAE J 2000;42(4):49.
[47] National Energy Code of Canada for Buildings 2017. National Research Council Canada (NRCan); 2017. URL https://nrc.canada.ca/en/certifications-evaluations-standards/codes-canada/codes-canada-publications/national-energy-code-canada-buildings-2017.
[48] Box GE, Cox DR. An analysis of transformations. J R Stat Soc 1964;26(2):211–43.
[49] Chollet F., et al. Keras. 2015.
[50] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. Tensorflow: a system for large-scale machine learning.. In: OSDI, 16; 2016. p. 265–83.
[51] GPy. GPy: A gaussian process framework in python. URL http://github.com/SheffieldML/GPy, since; 2012.
[52] Edwards RE, New J, Parker LE, Cui B, Dong J. Constructing large scale surrogate models from big data and artificial intelligence. Applied Energy 2017;202:685–99. doi:10.1016/j.apenergy.2017.05.155. URL <Go to ISI>://WOS:000407188500055
[53] Kuleshov V, Fenner N, Ermon S. Accurate uncertainties for deep learning using calibrated regression. In: International conference on machine learning; 2018. p. 2796–804.
[54] Platt J, et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. AdvLarge Margin Classifiers 1999;10(3):61–74.
[55] Scalia G, Grambow CA, Pernici B, Li Y-P, Green WH. Evaluating scalable uncertainty estimation methods for deep learning based molecular property prediction. J Chem Inf Model 2020.
[56] Geyer P, Singaravel S. Component-based building performance prediction using systems engineering and machine learning. Appl Energy 2017;228:1439–53.
[57] Westermann P, Evins R. Adaptive sampling for building simulation surrogate model derivation using the Lola-Voronoi algorithm. In: International Building Performance Association (IBPSA), editor. Proceedings of the international building performance simulation association, 16; 2019. p. 1559–63. doi:10.26868/25222708.2019.211232.