Accepted in the Annals of Applied Statistics

PARAMETER CALIBRATION IN WAKE EFFECT SIMULATION MODEL WITH STOCHASTIC GRADIENT DESCENT AND STRATIFIED SAMPLING

BY BINGJIE LIU[1], XUBO YUE[2], EUNSHIN BYON[2], RAED AL KONTAR[2]

[1]*Zhejiang Lab*
Zhejiang, China 311121
bingjiel@zhejianglab.edu

[2]*Department of Industrial and Operations Engineering*
University of Michigan, Ann Arbor, MI 48109
maxyxb@umich.edu ebyon@umich.edu alkontar@umich.edu*

As the market share of wind energy has been rapidly growing, wake effect analysis is gaining substantial attention in the wind industry. Wake effects represent a wind shade cast by upstream turbines to the downwind direction, resulting in power deficits in downstream turbines. To quantify the aggregated influence of wake effects on the power generation of a wind farm, various simulation models have been developed, including Jensen's wake model. These models include parameters that need to be calibrated from field data. Existing calibration methods are based on surrogate models that impute the data under the assumption that physical and/or computer trials are computationally expensive, typically at the design stage. This, however, is not the case where large volumes of data can be collected during the operational stage. Motivated by the wind energy application, we develop a new calibration approach for big data settings without the need for statistical emulators. Specifically, we cast the problem into a stochastic optimization framework and employ stochastic gradient descent to iteratively refine calibration parameters using randomly selected subsets of data. We then propose a stratified sampling scheme that enables choosing more samples from noisy and influential sampling regions and thus, reducing the variance of the estimated gradient for improved convergence. Through both theoretical and numerical studies on wind farm data, we highlight the benefits of our variance-conscious calibration approach.

1. Introduction. Advances in numerical algorithms and computing power bring computer models to the forefront of design, control, interpretation, and analysis of many complex physical and/or engineered systems. A computer model is a set of functions that simulate the behavior of a real-world system with inputs and outputs. Such models include simulations, computational models, and dynamical equations, amongst others. Given the input variables, say, operational conditions of a system, a computer model generates outputs such as system performance metrics. Besides input values, additional parameters often need to be specified *a priori* in computer models. These parameters are referred to as calibration parameters. Different from input variables that are directly observable or controllable, calibration parameters are often unobservable and their appropriate values should be estimated (or tuned) based on either physical laws or empirical data such that the computer model output can be aligned with observational data. For complex systems, physical laws to accurately identify the parameters are most often unavailable. This necessitates the estimation of the unknown parameters from data. Such an estimation procedure is referred to as parameter calibration in literature (Kennedy and O'Hagan, 2001).

*Corresponding author. Supported by the U.S. National Science Foundation, Division of Information and Intelligent Systems (IIS), under Grant IIS-1741166.

Keywords and phrases: Jensen's wake model, stochastic optimization, variance reduction, wind energy.

Fig 1: Illustration of wake effects (excerpted from Zwakman (2014))

Calibration has been an active area of research over the last couple of decades due to its relevance in many modern applications. One such application is wind power systems which are among the fastest-growing renewable energy sources (International Energy Agency, 2015). In wind power systems, it is often crucial to quantify wake effects, as they substantially affect the power generation performance in multi-turbine wind farms (Barthelmie and Pryor (2013)). Wake represents the wind shade cast by upwind turbines to the downwind direction where turbines at the upwind direction disturb and slow down the wind power for others (Fig. 1). This leads to energy loss in downstream turbines. As the scale of both wind turbines and wind farms grows, estimating wake effects has gained substantial attention in the wind industry (Churchfield, 2013). Tracing back to the 1980s, a wide variety of models were proposed to estimate wake effects. The most notable model that stood the test of time is the Jensen model (Jensen, 1983) which estimates wind speed deficits at downstream turbines by applying momentum equations. The Jensen model became the basis of a large variety of applications and extensions in literature (Ainslie, 1988; Larsen, 1988; Frandsen, 1992).

Engineering wake models often introduce additional parameters that need to be calibrated. For instance, to run Jensen's model, besides the input, we need to specify a wake decay coefficient that affects the wind speed deficits in downstream turbines. Figure 2 shows the wind

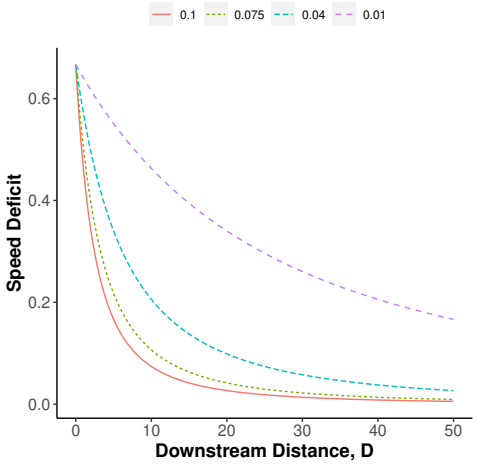


Fig 2: Influence of wake decay parameter on wind speed deficit in Jensen's wake model in a two-turbine setting. Here, the wind speed deficit implies the difference of wind speeds at the two turbines, relative to the speed at the upstream turbine. The downstream distance on the x-axis refers to the distance between upstream and downstream turbines.

deficit at a downstream turbine in a simple two-turbine setting. As the wake decay coefficient decreases, the wind speed deficit increases. In other words, when the wake is substantial, a smaller value should be employed in the Jensen's model. Its recommended values are 0.075 and 0.04 for land-based and offshore wind farms, respectively, in wind industry (Katic, Højstrup and Jensen, 1986; DTU Wind Energy, 2015; Staid, 2015; Barthelmie et al., 2010). Recent studies in You et al. (2017, 2018); Göçmen et al. (2016), however, find that these values do not represent the wake effects accurately in their studied wind farms. This problem sets forth the need for statistical calibration and represents the key motivation behind this paper.

Although significant advances have been made in the calibration field, prior studies are predominantly based on one key assumption: *computer and/or physical experiments are sufficiently time-demanding, so a limited number of physical trials and computer experiments can be carried out*. As a result, literature has dealt with the calibration problem through statistical emulators (often referred to as surrogates) where interpolation methods, most often Gaussian processes (GPs), are deployed to impute outputs from unobserved inputs under different values of calibration parameters. The foundations of modern calibration, laid by the Bayesian calibration approach, were introduced in the seminal papers by Kennedy and O'Hagan (2001); Higdon et al. (2004). Let $\boldsymbol{y}(\boldsymbol{x})$ and $\boldsymbol{y}^c(\boldsymbol{x}, \boldsymbol{\theta})$, respectively, denote outputs from the physical system and computer model at input $\boldsymbol{x}$ and calibration parameters $\boldsymbol{\theta}$. The Bayesian calibration approach uses a linkage model to formulate the relationship between physical and simulation responses as $\boldsymbol{y}(\boldsymbol{x}) = \rho \boldsymbol{y}^c(\boldsymbol{x}, \boldsymbol{\theta}) + \gamma(\boldsymbol{x}) + e$, where ρ is a scaling coefficient, $\gamma(\boldsymbol{x})$ represents the discrepancy between the physical system and computer model at input $\boldsymbol{x}$, and e denotes random noise. Kennedy and O'Hagan (2001) develop surrogate models for $\boldsymbol{y}^c(\boldsymbol{x}, \boldsymbol{\theta})$ and $\gamma(\boldsymbol{x})$ with Gaussian processes and estimate the model parameters, including $\boldsymbol{\theta}$, in a Bayesian framework.

This Bayesian framework set forth many extensions and applications in the literature. For example, several studies employ Bayesian calibration to accommodate multivariate output (Paulo, García-Donato and Palomo, 2012), to adjust computer models for reducing the discrepancy between engineering models and actual systems (Joseph and Yan, 2015) and to link the low accuracy experiment with the high accuracy experiment (Qian and Wu, 2008). Higdon et al. (2004) consider a fast computer model that enables one to generate a large number of simulation data with different values of the parameters. However, they consider scant field observations. Similarly, Gramacy et al. (2015) accommodate large simulation data, but with limited field data, That being said, recent work by Tuo and Wu (2015) points out that the Bayesian approach can lead to unreasonable estimates when the computer model exhibits non-negligible bias, and they propose a new calibration method that minimizes the discrepancy between the two responses in a frequentist framework.

In contrast to the above literature, we focus on the calibration problem where both physical data and computer response data are not scarce; we consider calibration in Big Data settings. As the volume and velocity of data collection grow fast, many applications are now capable of collecting large volumes of operational data. Further, with accelerating advancements in computing technology, a large number of computer experiments have become increasingly common. Take, for example, the growing number of medium- or low-fidelity simulators where a sufficiently large number of data from computer trials can be easily attained. This is also the case in our motivating case study. The collection of 10-minute average measurements in a wind farm's Supervisory Control And Data Acquisition (SCADA) system generates 52,560 data points each year. With multiple years of data, one can easily obtain hundreds of thousands of data points. The run time of momentum equation-based wake simulation models ranges from seconds to minutes, depending on a wind farm size and other terrain complexities. Because these simulators run fast, there is no scarcity concern of observations from computer experiments.

In such cases, instead of relying on, and learning from, emulators, we can directly take advantage of a sufficient amount of data generated from both physical operations and computational experiments to perform parameter calibration. Such an approach circumvents the need to depend on a possibly erroneous model (i.e. the surrogate) and reduces its corresponding error propagation. In addition to that, using flexible and non-parametric surrogates such as GPs is often infeasible in such large data settings. For instance, with N data points, exact GPs have a complexity of $O(N^3)$ (Rasmussen (2003)). While state-of-the-art literature on approximate GPs (mainly under a variational inference framework) reduce the complexity (Damianou, Titsias and Lawrence (2016); Álvarez and Lawrence (2011); Snelson and Ghahramani (2006)), emulators are often limited to small data regimes which in turn has enabled their use in traditional calibration settings that assume computationally expensive computer and/or physical experiments.

In this study, we develop a calibration approach to guide large-scale computer experiments in a computationally efficient and robust way without the need for statistical emulators. We summarize our main contributions. First, we cast the calibration problem into the stochastic optimization, in particular, stochastic gradient descent (SGD), to iteratively fine-tune the calibration parameters. Stochastic optimization offers unique computational advantages where inference is done via mini-batch optimization. This allows scaling to large data size regimes in calibration problems. Next, motivated by our wind farm application, we incorporate a stratified sampling scheme that enables choosing more samples from noisy and influential sampling regions into the SGD framework and thus, reducing the variance of the estimated gradient for improved convergence. For effective implementation of stratified sampling, we dynamically partition the input domain, guided by the classification and regression tree (CART), based on the most up-to-date information obtained over iterations. To the best of our knowledge, this is the first study that introduces the dynamic stratified sampling in the stochastic optimization framework.

Unlike aforementioned studies that pre-generate computer model outputs over some pre-selected parameter values and estimate calibration parameters using the already obtained dataset, our approach generates computer model outputs *on the fly* with the most updated information obtained throughout an iterative procedure. This procedure enables us to collect more informative data as we learn. Numerical studies in a wide range of settings and wind energy case study demonstrate the unique benefits of the proposed approach. The results show that our approach achieves superior performance relative to Bayesian calibration techniques in a fraction of the time required for the latter.

Our dynamic stratified sampling approach can be easily integrated with most stochastic optimization methods. We choose two representative SGD approaches–the mini-batch SGD (one of the most popular SGD algorithms) and adaptive SGD (recent state-of-the-art) and compare them with our proposed variance-reduced approaches via stratified sampling. Our empirical analysis suggests that stratified sampling helps reduce computational burden greatly. In particular, our approach excels when the output variance is heterogeneous, as it allows us to concentrate on high noise regions to better stabilize the variance and hence achieve fast convergence.

The results of the case study with offshore and land-based wind farm data suggest that the calibrated wake decay coefficient is around 0.10 and 0.12, respectively. These values are quite different from the recommended values of 0.04 and 0.075, respectively, for offshore and land-based wind farms in the literature. We believe that the well-calibrated Jensen's model can help find optimal operational controls (e.g., torque and blade pitch controls) to maximize the wind farm's power generation (Kheirabadi and Nagamune, 2019) and also help design a new wind farm layout.

The organization of the paper is as follows: Section 2 develops SGD-based procedures to calibrate computer parameters. Section 3 presents implementation results with numerical

examples. Section 4 demonstrates the advantage of the proposed approach through the wind farm case study. Section 5 provides concluding remarks.

2. Methodology.

2.1. *Problem Formulation.* Let $\boldsymbol{x} \subseteq \mathbb{R}^m$ denote an input vector of dimension m in a system. Let $\boldsymbol{y}(\boldsymbol{x})$ denote the physical system response at input $\boldsymbol{x}$. The response can be a scalar if the system consists of one unit, or it can be a vector for a multi-unit system such as a multi-turbine wind farm. Let $\boldsymbol{y}^c(\boldsymbol{x}^c, \boldsymbol{\theta})$ denote the response vector from the computer model with $\boldsymbol{x}^c$ being an input of the computer model (e.g., wind condition in Jensen's wake model) and $\boldsymbol{\theta} \in \Theta$ ($\Theta \subseteq \mathbb{R}^p$) being parameters to be calibrated (e.g., wake decay parameter in Jensen's wake model). We consider a deterministic computer model that generates a fixed output, given $(\boldsymbol{x}^c, \boldsymbol{\theta})$. The input $\boldsymbol{x}^c$ could be a subset of the physical system input $\boldsymbol{x}$. For notation simplicity, we use the same notation $\boldsymbol{x}$ as inputs to both physical system and computer model.

The calibration parameter $\boldsymbol{\theta}$ is unknown. In Tuo and Wu (2015, 2016), the calibration problem is viewed as *"finding the combination of the model parameters, under which the computer outputs match the physical responses."* They formulate the calibration problem to find θ that minimizes the L_2 distance between the physical response surface and the computer output. Similarly, our goal is to identify the parameter value $\boldsymbol{\theta}$ that can minimize the discrepancy between the real system response $\boldsymbol{y}(\boldsymbol{x})$ and computer model response $\boldsymbol{y}^c(\boldsymbol{x}, \boldsymbol{\theta})$. Thus, we formulate the parameter calibration problem as

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \Theta} L(\boldsymbol{\theta}), \tag{2.1}$$

where $L(\boldsymbol{\theta})$ can be any form of loss function. Several loss functions can be considered. Among them, L_2 norm has been commonly used in the literature (Tuo and Wu, 2016). In this study, the following L_2 norm is employed due to its mathematical tractability.

$$L(\boldsymbol{\theta}) = \int ||\boldsymbol{y}(\boldsymbol{x}) - \boldsymbol{y}^c(\boldsymbol{x}, \boldsymbol{\theta})||_2^2 dP(\boldsymbol{x}, \boldsymbol{y}). \tag{2.2}$$

Because the joint probability density $P(\boldsymbol{x}, \boldsymbol{y})$ is unknown, the problem in (2.2) cannot be directly solved. Therefore, with real observed data points, we minimize the empirical loss. Suppose that a dataset $\{(\boldsymbol{x}_j, \boldsymbol{y}_j),\, j = 1, \cdots, N\}$ of size N is collected from a physical system, where $\boldsymbol{x}_j$ represents the j^{th} input, $\boldsymbol{y}_j$ is the output at $\boldsymbol{x}_j$. The empirical loss becomes

$$F(\boldsymbol{\theta}) = \frac{1}{N} \sum_{j=1}^{N} \left[\boldsymbol{y}(\boldsymbol{x}_j) - \boldsymbol{y}^c(\boldsymbol{x}_j, \boldsymbol{\theta})\right]' \left[\boldsymbol{y}(\boldsymbol{x}_j) - \boldsymbol{y}^c(\boldsymbol{x}_j, \boldsymbol{\theta})\right]. \tag{2.3}$$

Finding the optimal $\boldsymbol{\theta}$ that minimizes $F(\boldsymbol{\theta})$ is similar to the nonlinear least squares estimation problem because $\boldsymbol{y}^c$ is possibly nonlinear. However, unlike typical nonlinear least square problems with a pre-specified parametric form for $\boldsymbol{y}^c$, we treat the simulation model as black box and thus, there is no analytically closed form expression for the computer model output $\boldsymbol{y}^c(\cdot, \cdot)$.

Among several possible techniques for optimization, let us first consider the gradient descent which iteratively updates $\boldsymbol{\theta}$ using the approximated gradient information. Specifically, let $\boldsymbol{\theta}^{(i)}$ denote the iterate obtained at the i^{th} iteration and let $F_j(\boldsymbol{\theta}^{(i)})$ denote the empirical loss value with the j-th sample in N data points, evaluated at $\boldsymbol{\theta}^{(i)}$, i.e.,

$$F_j(\boldsymbol{\theta}^{(i)}) = \left[\boldsymbol{y}(\boldsymbol{x}_j) - \boldsymbol{y}^c(\boldsymbol{x}_j, \boldsymbol{\theta}^{(i)})\right]' \left[\boldsymbol{y}(\boldsymbol{x}_j) - \boldsymbol{y}^c(\boldsymbol{x}_j, \boldsymbol{\theta}^{(i)})\right]. \tag{2.4}$$

Then, we can update $\boldsymbol{\theta}^{(i)}$ by

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} + \alpha^{(i)}\boldsymbol{d}^{(i)}, \tag{2.5}$$

with

$$\boldsymbol{d}^{(i)} = -\frac{1}{N}\sum_{j=1}^{N}\nabla F_j(\boldsymbol{\theta}^{(i)}), \tag{2.6}$$

for $\alpha^{(i)} > 0$ where $\nabla F_j(\boldsymbol{\theta}^{(i)})$ denotes the gradient of $F_j(\boldsymbol{\theta}^{(i)})$, $\alpha^{(i)}$ is the step size and $\boldsymbol{d}^{(i)}$ implies the parameter update direction at the i^{th} iteration.

When $\boldsymbol{\theta}^{(i)}$ is updated based on all N data points and N is large, it can be computationally inefficient. To reduce the computational burden, SGD avoids the full gradient evaluation by randomly choosing a subset of data from the original dataset. Let us consider the mini-batch SGD. Let n ($< N$) denote the sample size per iteration. The SGD empirical loss function becomes

$$F_{SGD}(\boldsymbol{\theta}^{(i)}) = \frac{1}{n}\sum_{j=1}^{n}F_j(\boldsymbol{\theta}^{(i)}), \tag{2.7}$$

where $F_j(\boldsymbol{\theta}^{(i)})$ denotes the loss function value at the randomly sampled $\boldsymbol{x}_j$. The SGD finds the update direction as

$$\boldsymbol{d}^{(i)}_{SGD} = -\frac{1}{n}\sum_{j=1}^{n}\nabla F_j(\boldsymbol{\theta}^{(i)}) \equiv -\nabla F_{SGD}(\boldsymbol{\theta}^{(i)}). \tag{2.8}$$

In implementing SGD, $\nabla F_j(\boldsymbol{\theta}^{(i)})$ needs to be estimated, because $F_j(\boldsymbol{\theta}^{(i)})$ does not take a closed-form expression. We can obtain the estimate by taking the finite-differencing or alike numerical differentiation methods. For example, for 1-dimensional parameter, $\nabla F_j(\theta^{(i)})$ can be obtained by

$$\nabla F_j(\theta^{(i)}) = \frac{F_j(\theta^{(i)} + h) - F_j(\theta^{(i)} - h)}{2h}, \tag{2.9}$$

where h is the small interval used for numerical differentiation. The multivariate version of numerical difference method and some variants are available in Yuan, Ng and Tsui (2013).

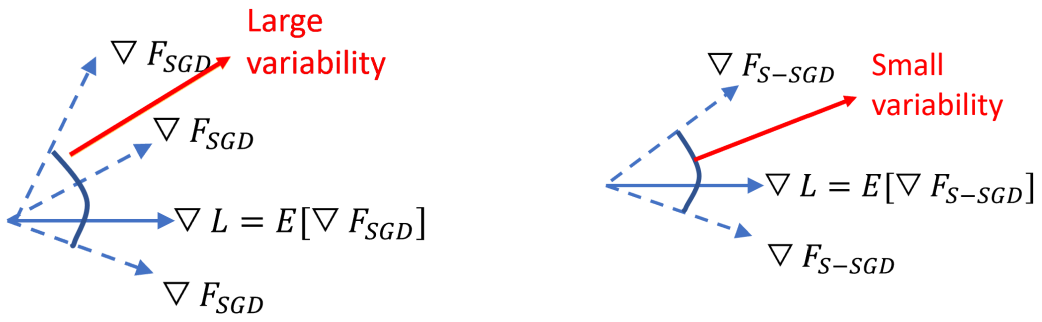
Let $\nabla L(\boldsymbol{\theta}^{(i)})$ denote the true gradient of the loss function at $\boldsymbol{\theta}^{(i)}$. We assume $\nabla F_{SGD}(\boldsymbol{\theta}^{(i)})$ is an unbiased estimator of the true gradient. While being unbiased, it is random and thus, it does not always coincide with the true gradient. The search direction is called a descent direction if the sample gradient is aligned with the true gradient, mathematically,

$$\nabla F_{SGD}(\boldsymbol{\theta}^{(i)})^T\nabla L(\boldsymbol{\theta}^{(i)}) > 0. \tag{2.10}$$

The descent search direction is obtained on the average (Bollapragada, Byrd and Nocedal, 2018), because we have

$$\mathbb{E}[\nabla F_{SGD}(\boldsymbol{\theta}^{(i)})^T\nabla L(\boldsymbol{\theta}^{(i)})] = ||\nabla L(\boldsymbol{\theta}^{(i)})||^2 > 0, \tag{2.11}$$

However, the descent direction in (2.10) does not necessarily hold, because the sample is chosen randomly. In fact, a randomly selected subset of data would lead to the noisy estimation of gradient. In particular, when the data size n is small, the variance of the sample gradient $\nabla F_{SGD}(\boldsymbol{\theta}^{(i)})$ could be large (see Figure 3a). Such large uncertainty causes the algorithm's stalling or slow convergence. When the sample gradient is largely different from its true gradient $\nabla L(\boldsymbol{\theta}^{(i)})$, the algorithm would struggle to improve. To circumvent such issue, we incorporate the variance reduction technique, specifically, stratified sampling, into the stochastic optimization framework. With stratified sampling, the sample gradient becomes less uncertain, aiming that it is closer to the true gradient and descent direction is achieved more frequently (see Figure 3b).



(a) Large variance in SGD

(b) Reduced variance in S-SGD

Fig 3: Variance comparison of sample gradient between SGD and S-SGD.

2.2. *Stratified stochastic gradient descent (S-SGD).* Several variance reduction techniques are available, including importance sampling and stratified sampling. In this study, we employ stratified sampling. In the future, we plan to explore importance sampling and other variance reduction techniques. The underlying idea of stratified sampling is that, when we estimate the mean, smaller variance can be achieved when the within-stratum variability becomes generally smaller than the between-strata variability.

To implement the stratified sampling, we divide the whole input space into a finite number of K disjoint subsets, Ω_k, $k = 1, \cdots, K$, also referred to as strata (we will discuss how to partition the input space at the end of Section 2.2.1.) Let p_k denote the probability of the k^{th} stratum, i.e., $p_k = Pr(\boldsymbol{x} \in \Omega_k)$. In practice, p_k can be computed from the proportion of input samples that belong to the k^{th} stratum among the total N data points. Unlike the original SGD which randomly samples inputs, our idea is to draw more samples from the strata where the variance is large.

2.2.1. *Stratification.* Let n_k denote the number of samples randomly selected from the k^{th} stratum such that $n = \sum_k n_k$ and F_{kj} denote the empirical loss with the j^{th} sample $\boldsymbol{x}_{kj}$, drawn from the k^{th} stratum, i.e.,

$$F_{kj}(\boldsymbol{\theta}^{(i)}) = \left[\boldsymbol{y}(\boldsymbol{x}_{kj}) - \boldsymbol{y}^c(\boldsymbol{x}_{kj}, \boldsymbol{\theta}^{(i)})\right]' \left[\boldsymbol{y}(\boldsymbol{x}_{kj}) - \boldsymbol{y}^c(\boldsymbol{x}_{kj}, \boldsymbol{\theta}^{(i)})\right]. \tag{2.12}$$

The empirical loss with n sampled data points at $\boldsymbol{\theta}^{(i)}$ is

$$F_{S-SGD}(\boldsymbol{\theta}^{(i)}) = \sum_k p_k F_k(\boldsymbol{\theta}^{(i)}), \tag{2.13}$$

where $F_k(\boldsymbol{\theta}^{(i)})$ is the average empirical loss at the k^{th} stratum with n_k random samples as

$$F_k(\boldsymbol{\theta}^{(i)}) = \frac{1}{n_k} \sum_{j=1}^{n_k} F_{kj}(\boldsymbol{\theta}^{(i)}). \tag{2.14}$$

Then, the sample gradient at the i^{th} iteration becomes

$$\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)}) = \sum_k p_k \nabla F_k(\boldsymbol{\theta}^{(i)}) \tag{2.15}$$

with

$$\nabla F_k(\boldsymbol{\theta}^{(i)}) = \frac{1}{n_k} \sum_{j=1}^{n_k} \nabla F_{kj}(\boldsymbol{\theta}^{(i)}), \tag{2.16}$$

where $\nabla F_{kj}(\boldsymbol{\theta}^{(i)})$ denotes the sample gradient of $F_{kj}(\boldsymbol{\theta}^{(i)})$. For any positive integer n_k, $\nabla F_k(\boldsymbol{\theta}^{(i)})$ is assumed to be an unbiased estimator of the true gradient $\nabla L_k(\boldsymbol{\theta}^{(i)})$ at the k^{th} stratum, as $\boldsymbol{x}_{kj}$ is randomly sampled within the stratum. Therefore, $\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})$ provides an unbiased estimation of the true gradient. Then our goal is to find the computational budget n_k that can minimize the variance of the gradient direction, given the total sample size n, as

$$\begin{aligned} &\arg\min_{n_k} \; Var\left(\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)})\right) \\ &s.t. \qquad \sum_k n_k = n \end{aligned} \tag{2.17}$$

Plugging $\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})$ in (2.15)-(2.16) into the objective function in (2.17), we obtain

$$\begin{aligned} Var\left(\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)})\right) &= \sum_k p_k^2 Var\left(\nabla F_k(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)})\right) \\ &= \sum_k \frac{p_k^2 \sigma_k^2}{n w_k}, \end{aligned} \tag{2.18}$$

where $\sigma_k^2 = Var\left(\nabla F_{kj}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)})\right)$ represents the variance at the k^{th} stratum and $w_k = n_k/n$ implies the weight of the k^{th} stratum.

Then the optimization problem in (2.17) can be rewritten as

$$\begin{aligned} &\arg\min_{w_k} \; \sum_k \frac{p_k^2 \sigma_k^2}{n w_k} \\ &s.t. \qquad \sum_k w_k = 1 \end{aligned} \tag{2.19}$$

By using the Lagrangian multiplier, we can obtain the optimal w_k as

$$w_k = \frac{p_k \sigma_k}{\sum_k p_k \sigma_k}, \tag{2.20}$$

which leads to

$$n_k = n \cdot \frac{p_k \sigma_k}{\sum_k p_k \sigma_k}. \tag{2.21}$$

This sample size is known to be the optimal allocation in the stratified random sampling when the costs of drawing a sample are equal (Lohr, 2010).

When n_k is not an integer, we round it to the nearest integer. However, when n_k becomes zero after rounding, the sampling gradient at the k^{th} stratum $\nabla F_k(\boldsymbol{\theta}^{(i)})$ in (2.16) is not defined. As a remedy, we assign a small sample size n_0 to all strata. With the remaining computational resource $n - Kn_0$, we apply the allocation rule in (2.21). Then the final allocation for the k^{th} stratum becomes

$$n_k = n_0 + (n - Kn_0) \cdot \frac{p_k \sigma_k}{\sum_k p_k \sigma_k}. \tag{2.22}$$

The allocation rule in (2.22) has important implications. First, while the original mass p_k for each stratum is taken into consideration, our approach allocates larger computational budgets to more noisy strata with larger σ_k to control the variance of the sample stochastic direction. On the other hand, in less noisy strata, a small number of samples is sufficient for clearly obtaining gradient information. This aspect makes S-SGD much more beneficial than SGD when the noise is heterogeneous. We will demonstrate it in Section 3.

In (2.22), σ_k^2 $\left(= Var\left(\nabla F_{kj}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)})\right)\right)$ is unknown, so we approximate it using data. First, we estimate the true gradient $\nabla L(\boldsymbol{\theta}^{(i)})$ with the sample gradient $\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})$ in (2.15) and use the sample variance as follows.

(2.23)
$$\hat{\sigma}_k^2 = \frac{1}{n_k - 1} \sum_{j=1}^{n_k} \left(\left(\nabla F_{kj}(\boldsymbol{\theta}^{(i)})\right)^T \nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)}) - \left(\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})\right)^T \nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)}) \right)^2.$$

To obtain $\hat{\sigma}_k^2$, n_k should be greater than 1. As such, we set the smallest sample size n_0 in (2.22) to be greater than 1, i.e., $n_0 \geq 2$.

Finally, we discuss how to partition the input space. In (2.18), small variance can be achieved when the within-stratum variability (that is, σ_k) is small. It is challenging to find optimal partitioning variables and their corresponding splitting points that minimize the within-stratum variability. However, we note that reducing the within-stratum variability is aligned with the spirit of CART (Breiman et al., 1984). CART is a tree-based machine learning method that seeks to decrease the impurity in each leaf node by splitting the input domain in a recursive manner (Breiman et al., 1984; Ning et al., 2017). As CART takes a stepwise procedure, it does not provide optimal splitting. But it has been proven effective in many applications, and it can be easily incorporated into the proposed optimization framework. Moreover, by splitting the input domain with CART at each iteration, our stratification becomes dynamic, meaning that the strata obtained in one iteration could be different from the strata in other iterations. Thus, as we proceed iterations, we can get different strata using the most updated parameter value $\boldsymbol{\theta}^{(i)}$. In implementing CART in the proposed framework, we use $\nabla F_{kj}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)})$, with $\nabla L(\boldsymbol{\theta}^{(i)})$ approximated by $\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})$, as the response variable, and the final leaf nodes are regarded as strata.

2.2.2. *Parameter updates.* Once n_k is obtained from stratified sampling, we randomly sample $\boldsymbol{x}_{kj}$, $j = 1, 2, \cdots, n_k$, from the k^{th} stratum for $k = 1, 2, \cdots, K^{(i)}$, where $K^{(i)}$ is the number of strata obtained from the CART splitting at iteration i. Then we get the updating direction $\boldsymbol{d}_{S-SGD}^{(i)} = -\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})$ with $\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})$ being defined in (2.15)-(2.16), and $\boldsymbol{\theta}^{(i)}$ is updated to $\boldsymbol{\theta}^{(i+1)}$ by $\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} + \alpha^{(i)} \boldsymbol{d}_{S-SGD}^{(i)}$. We iterate the procedure until some stopping criterion is satisfied. As a stopping rule, we stop the iteration when $(\boldsymbol{\theta}^{(i+1)} - \boldsymbol{\theta}^{(i)})/\boldsymbol{\theta}^{(i)}$ becomes sufficiently small. Algorithm 1 summarizes the procedure of S-SGD where lines #4-#7 and #16 explain how to sample a subset of data using the stratified sampling and lines #13-#16 summarize how to partition the input domain via CART.

In updating $\boldsymbol{\theta}^{(i)}$, we need to define the step size $\alpha^{(i)}$. We use the backtracking line search, summarized in Algorithm 2, which is used in the most advanced SGD versions (Paquette and Scheinberg, 2020; Bollapragada, Byrd and Nocedal, 2018). The basic idea of the backtracking line search is as follows. We set the step size at $\alpha^{(i)} = \zeta^{(i)} \alpha_0$ with $\zeta^{(i)} \geq 1$ (Line #4). $\zeta^{(i)}$ depends on the variance of the gradient. If such $\alpha^{(i)}$ does not result in a sufficient decrease in the objective function, we decrease $\alpha^{(i)}$ iteratively. Specifically, let $L^{(i)} = 1/\alpha^{(i)}$ and we set $L^{(i)} = \eta L^{(i)}$ with $\eta > 1$ (Line #8). This inner iteration for adjusting the step size (Lines #7-#10) proceeds until we achieve a sufficient decrease. In our implementation, we use $\eta = 1.5$ (Bollapragada, Byrd and Nocedal, 2018).

Next, we will discuss how to get $\zeta^{(i)}$. Bollapragada, Byrd and Nocedal (2018) propose a variance-based rule to set $\zeta^{(i)}$ as $\zeta^{(i)} = max(1, 2/a^{(i)})$ in the SGD setting, where $a^{(i)} = \frac{Var(\nabla F_{SGD}(\boldsymbol{\theta}^{(i)}))}{||\nabla F_{SGD}(\boldsymbol{\theta}^{(i)})||^2} + 1$. We further modify the procedure to accommodate our variance reduction property as follows.

(2.24)
$$a^{(i)} = \frac{Var(\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)}))}{||\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})||^2} + 1.$$

Algorithm 1 S-SGD

Input: sample size n, smallest small size $n_0(\geq 2)$, starting point of parameter $\boldsymbol{\theta}_0$, initial number of strata $K^{(0)}$.
Initialization: Set the initial parameter value $\boldsymbol{\theta}^{(1)} = \boldsymbol{\theta}_0$. Evenly divide the input domain into $K^{(0)}$ strata and collect probability mass $p_1, \cdots, p_{K^{(0)}}$. Assign $w_k^{(0)} = p_k$ for $k = 1, \cdots, K^{(0)}$. Set the iteration index $i = 1$.
while Convergence condition is not met **do**
  Update $w_k^{(i)} = w_k^{(i-1)}$ for $k = 1, \cdots, K^{(i-1)}$.
  Update $n_k = w_k^{(i)}(n - Kn_0) + n_0$ for $k = 1, \cdots, K^{(i-1)}$.
  Update $K^{(i)} = K^{(i-1)}$.
  Draw $\boldsymbol{x}_{kj}$, $j = 1, \cdots, n_k$, from each stratum for $k = 1, \cdots, K^{(i)}$.
  Compute $\nabla F_{kj}(\boldsymbol{\theta}^{(i)})$ in (2.9) for each sample $j \in \{1, \cdots, n_k\}$ and $\nabla F_k(\boldsymbol{\theta}^{(i)})$ in (2.16) for each stratum, $k \in \{1, \cdots, K^{(i)}\}$.
  Obtain $\nabla F(\boldsymbol{\theta}^{(i)})$ in (2.15).
  Compute the updating direction $d^{(i)} = -\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})$.
  Set $\alpha^{(i)}$ using Algorithm 2.
  Update $\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} + \alpha^{(i)} d^{(i)}$.
  Build a regression tree $T^{(i)}$ using $\left(\boldsymbol{x}_{kj}, \nabla F_{kj}(\boldsymbol{\theta}^{(i)})^T \nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})\right)$, $k = 1, \cdots, K^{(i)}$, $j = 1, \cdots, n_k$, and set $K^{(i)}$ with the number of leaf nodes in the resulting regression tree.
  Obtain $\hat{\sigma}_k^2$ using (2.23) for $k = 1, \cdots, K^{(i)}$.
  Obtain the probability mass $p_1, \cdots, p_{K^{(i)}}$.
  Compute $w_k^{(i)} = \frac{p_k \hat{\sigma}_k^{(i)}}{\sum_{k'=1}^{K} p_{k'} \hat{\sigma}_{k'}^{(i)}}$ for $k = 1, \cdots, K^{(i)}$.
  Set $i \leftarrow i + 1$.
end while
Output: $\boldsymbol{\theta}^{(i)}$.

Algorithm 2 Backtracking line search to determine the step size α_0 in S-SGD

Input: $\alpha_0 > 0, \eta > 1$
Initialization
Compute $\zeta^{(i)} = max(1, 2/a^{(i)})$ with $a^{(i)}$ in (2.24).
Set $\alpha^{(i)} = \alpha_0 \zeta^{(i)}$.
Set $L^{(i)} = 1/\alpha^{(i)}$
$F' = F_{S-SGD}(\boldsymbol{\theta}^{(i)} - 1/L^{(i)} \nabla F(\boldsymbol{\theta}^{(i)}))$
while $F' > F_{S-SGD}(\boldsymbol{\theta}^{(i)}) - \frac{1}{2L^{(i)}} ||\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})||^2$ **do**
  Set $L^{(i)} = \eta L^{(i)}$.
  Compute $F' = F_{S-SGD}(\boldsymbol{\theta}^{(i)} - \frac{1}{L^{(i)}} \nabla F(\boldsymbol{\theta}^{(i)}))$
end while
Output: $\alpha^{(i)}$.

Note that we have $\zeta^{(i)} \in [1, 2]$, because $a^{(i)} \geq 1$. The numerator can be obtained by

$$Var\left(\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})\right) = \sum_k \frac{p_k^2}{n_k} Var\left(\nabla F_{kj}(\boldsymbol{\theta}^{(i)})\right). \tag{2.25}$$

Here we estimate $Var\left(\nabla F_{kj}(\boldsymbol{\theta}^{(i)})\right)$ in the last equation with the sample variance of $F_{kj}(\boldsymbol{\theta}^{(i)})$'s for $j \in \{1, 2, \cdots, n_k\}\}$ at the k^{th} stratum.

Our variance-conscious approach provides additional benefits in determining the step size using (2.24) in comparison with SGD. Because S-SGD produces a smaller variance, given

the same sample size, the initial step size in S-SGD at each iteration is larger, compared to SGD. Therefore, S-SGD can move faster than SGD. But, at the same time it guards against an unduly large jump when the gradient estimation is uncertain; note that when the variance is large even with the stratified sampling, S-SGD makes a small move with small $\zeta^{(i)}$ to accommodate the uncertainty.

In summary, the proposed S-SGD provides more benefits over SGD, because its parameter updating direction is more stable with reduced variance, better aligning with the true direction. Further, S-SGD with the variance-controlled backtracking allows a larger step size than SGD, while avoiding an unnecessarily big move when the estimated variance is large.

Remark: Because the loss function (and its gradient) is estimated via Monte Carlo integration, other sampling methods such as Latin hypercube sampling (LHS), quasi-Monte Carlo sampling (QMC), and Latin hyperrectangle sampling (LHRS) may be considered. First, similar to the stratified sampling, LHS uses stratification. It divides the input's cumulative density function into multiple strata and randomly draws one sample from each stratum (Owen, 2013). While LHS can reduce the variance of the crude Monte Carlo (CMC) estimator, its primary focus is to recreate the input distribution with fewer iterations than CMC. Thus, it only uses the input distribution, while the output variability is not directly taken into consideration. Similarly, QMC, which uses a low-discrepancy sequence, enjoys faster convergence than CMC by using deterministic inputs (Owen, 2013). However, similar to LHS, it does not utilize the variability of the output function. To address this limitation, LHRS generalizes LHS by considering the output variability when partitioning the input domain (Mease and Bingham, 2006). This technique is closely related to our method in that it allows non-equal cell probabilities. However, LHRS assumes the variance of output within a cell (i.e., stratum) grows with the cell size. This assumption is reasonable when any prior knowledge about the output structure is not available. In our approach, however, we obtain knowledge about the loss function and its gradient throughout an iterative process. Hence, stratified sampling, which uses the actual output variability, provides a better sampling scheme in the proposed framework.

2.2.3. *Uncertainty quantification.* The Bayesian approach provides rich information for uncertainty quantification (UQ). While the SGD-based procedure typically focuses on finding a point estimation, some studies show that it also provides a capability to quantify the uncertainty in the resulting estimator. Building on the result in Polyak (1990) and Ruppert (1988), Polyak and Juditsky (1992) show that the average iterate $\bar{\boldsymbol{\theta}}^{(M)} = 1/M \sum_{i=1}^{M} \boldsymbol{\theta}^{(i)}$ converges in distribution to a multivariate normal random vector under certain conditions. Recently, some studies derive consistent estimators of the asymptotic covariance of $\bar{\boldsymbol{\theta}}^{(M)}$ that can be used in an online fashion without storing all the data required to compute the estimators (Chen et al., 2020; Fang, Xu and Yang, 2018).

In this section, we use the procedure presented in Polyak and Juditsky (1992) to capture the uncertainty of the calibrated parameter. They show that $\bar{\boldsymbol{\theta}}^{(M)}$ converges to the global minimum $\boldsymbol{\theta}^*$ when the loss function is strongly convex with a Lipschitz gradient and it holds

$$\sqrt{M}(\bar{\boldsymbol{\theta}}^{(M)} - \boldsymbol{\theta}^*) \xrightarrow{d} MVN(0, A^{-1}SA^{-1}), \tag{2.26}$$

for $\alpha^{(i)} = \alpha^{(0)} i^{-\delta}$ with $\delta \in (0.5, 1)$ as M gets larger, where $A = \nabla^2 L(\boldsymbol{\theta}^*)$ is the Hessian matrix of $L(\boldsymbol{\theta})$ at $\boldsymbol{\theta}^*$, and $S = E([\nabla F(\boldsymbol{\theta}^*)][\nabla F(\boldsymbol{\theta}^*)]^T)$ is the covariance matrix of $\nabla F(\boldsymbol{\theta}^*)$ (Chen et al., 2020).

The result in (2.26) allows us to quantify the uncertainty in estimating the calibration parameter and makes it possible to construct a confidence interval (CI). Because both A and

S metrics are unknown, we estimate them by their sample versions,

$$\hat{A} = \nabla^2 F_{S-SGD}(\bar{\boldsymbol{\theta}}^{(M)}) = \sum_k p_k \nabla^2 F_k(\bar{\boldsymbol{\theta}}^{(M)}) \tag{2.27}$$

with

$$\nabla^2 F_k(\bar{\boldsymbol{\theta}}^{(M)}) = \frac{1}{n_k} \sum_{j=1}^{n_k} \nabla^2 F_{kj}(\bar{\boldsymbol{\theta}}^{(M)}), \tag{2.28}$$

where $\nabla^2 F_{kj}(\bar{\boldsymbol{\theta}}^{(M)})$ denotes the sample Hessian of $F_{kj}(\bar{\boldsymbol{\theta}}^{(M)})$. We also estimate S in a similar manner.

2.3. *Adaptive stratified SGD.* In Section 2.2 we regulate the variance using the variance reduction technique. Another remedy to reduce variance is to increase the sample size. Bollapragada, Byrd and Nocedal (2018) discuss an adaptive SGD, referred to as A-SGD in this study, to adaptively increase the sample size. Let $n^{(i)}$ denote the sample size at the i^{th} iteration. In this section, we incorporate the adaptive sampling approach into the S-SGD framework. The adaptive version of S-SGD is referred to as the adaptive stratified SGD (AS-SGD) in the subsequent discussion.

While S-SGD aims to provide stable variance, the resulting variance could be still large if the sample size is too small. The key idea of AS-SGD is to increase the sample size, as needed. At each iteration, we start with the initial sample size n_0 to obtain $Var\left(\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta})\right)$ in (2.18). Then we increase the sample size until the resulting relative variance is smaller than a pre-specified threshold κ, as shown below in (2.29)

$$\frac{Var\left(\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta})\right)}{\|\nabla L(\boldsymbol{\theta}^{(i)})\|^4} \leq \kappa^2, \tag{2.29}$$

for $\kappa > 0$, given $\boldsymbol{\theta}^{(i)}$ at the i^{th} iteration. This test is called *inner product test*.

In addition to the inner product test, we include another test, called *orthogonality test*, to ensure a convergence to the optimal point when the true loss function is strongly convex (Bollapragada, Byrd and Nocedal, 2018). The basic idea is to impose a bound on the component of the sample gradient orthogonal to the true gradient as follows. For $v > 0$, given $\boldsymbol{\theta}^{(i)}$ at the i^{th} iteration,

$$\frac{\mathbb{E}\left\| \nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)}) - \frac{\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)})}{\|\nabla L(\boldsymbol{\theta}^{(i)})\|^2} \nabla L(\boldsymbol{\theta}^{(i)}) \right\|^2}{\|\nabla L(\boldsymbol{\theta}^{(i)})\|^4} \leq \nu^2, \tag{2.30}$$

When either of the inequalities in (2.29) and (2.30) is not satisfied, we obtain an additional sample of size n_c. We iterate this process until both conditions are met. In our implementation, we approximate the numerator and denominator with the corresponding sampling variance and $\|\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})\|^4$, respectively. We choose $\kappa = 0.9$ and $\nu = 5.84$, as suggested in Bollapragada, Byrd and Nocedal (2018). Algorithm 3 summarizes the AS-SGD procedure, where Lines #12-#20 explain the iterative procedure that adaptively determines the sample size.

While our adaptive sampling idea hinges upon A-SGD (Bollapragada, Byrd and Nocedal, 2018), it has several differences. First, in Bollapragada, Byrd and Nocedal (2018), once they estimate the variance, they use the estimated value to choose the sample size that satisfies both tests. However, with a small initial sample size, the variance estimate is often large and

Algorithm 3 AS-SGD

Input: smallest small size $n_0(\geq 2)$, initial and incremental sample size n_c, starting point of parameter $\boldsymbol{\theta}_0$, initial number of strata $K^{(0)}$.
Initialization
Set the initial parameter value $\boldsymbol{\theta}^{(1)} = \boldsymbol{\theta}_0$. Evenly divide the input domain into $K^{(0)}$ strata and collect probability mass $p_1, \cdots, p_{K^{(0)}}$. Assign $w_k^{(0)} = p_k$ for $k = 1, \cdots, K^{(0)}$. Set the iteration index $i = 1$.
while Convergence condition is not met **do**
  Set $n = n_c$.
  Update $w_k^{(i)} = w_k^{(i-1)}$ for $k = 1, \cdots, K^{(i-1)}$.
  Update $n_k^* = w_k^{(i)}(n - Kn_0) + n_0$ for $k = 1, \cdots, K^{(i-1)}$.
  Update $K^{(i)} = K^{(i-1)}$.
  Set $n_k = n_k^*$
  Draw x_{jk}, $j = 1, \cdots, n_k$, from each stratum $k = 1, \cdots, K^{(i)}$.
  Compute $F_{kj}(\boldsymbol{\theta}^{(i)})$ for each sample $j \in \{1, \cdots, n_k\}$ and $\nabla F_k(\boldsymbol{\theta}^{(i)})$ for each stratum $k \in \{1, \cdots, K^{(i)}\}$.
  Conduct the inner product test in (2.29).
  while the inner product test in (2.29) or orthogonality test in (2.30) fails **do**
    Compute $n_k' = w_k^{(i)} n_c$ for $k = 1, \cdots, K^{(i)}$.
    Draw x_{jk}, $j = n_k + 1, \cdots, n_k + n_k'$, from each stratum $k = 1, \cdots, K^{(i)}$.
    Compute $F_{kj}(\boldsymbol{\theta}^{(i)})$ for each sample $j = n_k + 1, \cdots, n_k + n_k'$.
    Update $\nabla F_k(\boldsymbol{\theta}^{(i)})$ for each stratum $k = 1, \cdots, K^{(i)}$.
    Set $n_k = n_k + n_k'$.
    Set $n = n + n_c$.
    Conduct the inner product test in (2.29).
  end while
  Obtain $\nabla F(\boldsymbol{\theta}^{(i)})$ in (2.15).
  Compute the updating direction $d^{(i)} = -\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})$.
  Set $\alpha^{(i)}$ using Algorithm 2.
  Update $\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} + \alpha^{(i)} d^{(i)}$.
  Build a regression tree $T^{(i)}$ using $\left(\boldsymbol{x}_{kj}, \nabla F_{kj}(\boldsymbol{\theta}^{(i)})^T \nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})\right)$, $k = 1, \cdots, K^{(i)}$, $j = 1, \cdots, n_k$, and set $K^{(i)}$ with the number of leaf nodes in the resulting regression tree.
  Obtain $\hat{\sigma}_k^2$ using (2.23) for $k = 1, \cdots, K^{(i)}$.
  Obtain the probability mass $p_1, \cdots, p_{K^{(i)}}$.
  Compute $w_k^{(i)} = \frac{p_k \hat{\sigma_k}^{(i)}}{\sum_{k'=1}^{K} p_{k'} \hat{\sigma_k'}^{(i)}}$ for $k = 1, \cdots, K^{(i)}$.
  Set $i \leftarrow i + 1$.
end while
Output: $\boldsymbol{\theta}^{(i)}$.

uncertain, and applying the tests with uncertain estimates often increases the sampling size rapidly. On the contrary, we incrementally add samples with refined variance estimates, until the tests are satisfied. Next, in our adaptive scheme, we use the same initial sample size at each iteration, whereas Bollapragada, Byrd and Nocedal (2018) start with the sample size obtained from the previous iteration and thus, $n^{(i)}$ is monotonically non-decreasing. When $n^{(i-1)}$ is unnecessarily large with the aforementioned inaccurate estimation of variance, A-SGD continues to use the large sample size in later iterations. As a result, our implementation results in Section 3 show that the required number of A-SGD grows very fast in many instances. On the other hand, our AS-SGD approach shows more stable growth, saving a substantial computational burden over A-SGD.

Finally, we show the convergence properties of AS-SGD. Proofs are available in the supplementary material. Thus far, for notation simplicity, we use $\mathbb{E}[\cdot]$ and $Var[\cdot]$ to denote the expectation and variance at the i^{th} iteration. In fact, they imply the conditional expectation and variance, given $\boldsymbol{\theta}^{(i)}$. Now we slightly modify the notations in the following theoretical results. Let s_i denote the randomly selected batch data sampled at iteration i and $s_1 : s_i$ the random samples drawn up to iteration i. Then $\mathbb{E}_{s_1:s_i}[\cdot]$ and $Var_{s_1:s_i}[\cdot]$, respectively, denote the conditional expectation and variance, given $s_1 : s_i$ (or equivalently, given $\boldsymbol{\theta}^{(i)}$), whereas $\mathbb{E}[\cdot]$ and $Var[\cdot]$ are the unconditional mean and expectation, respectively.

First, Lemma 1 shows that $\mathbb{E}_{s_1:s_i}\left[\left\|\nabla F_{_{S-SGD}}(\boldsymbol{\theta}^{(i)})\right\|^2\right]$ is bounded under the following assumptions.

ASSUMPTION 1. *We assume the stochastic gradient is unbiased, i.e.,* $\mathbb{E}[\nabla F_k(\boldsymbol{\theta}^{(i)})] = \nabla L_k(\boldsymbol{\theta}^{(i)})$ *for* $k = 1, \cdots, K^{(i)}$.

ASSUMPTION 2. *The loss function* $L(\boldsymbol{\theta})$ *is* L*-smooth.*

LEMMA 1. *Given Assumptions 1 and 2, under the AS-SGD procedure, we have*

$$\mathbb{E}_{s_1:s_i}\left[\left\|\nabla F_{_{S-SGD}}(\boldsymbol{\theta}^{(i)})\right\|^2\right] \leq \frac{Var_{s_1:s_i}\left[\nabla F_{_{S-SGD}}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)})\right]}{\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2} + (\nu^2+1)\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2$$
$$\leq (1+\kappa^2+\nu^2)\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2,$$

for any $i \geq 1$.

Next, Theorem 1 shows the bound on difference between the loss function at any iterate and optimal parameter value and presents the convergence result, when $L(\boldsymbol{\theta})$ is H-strongly convex and L-smooth over $\mathbb{R}^m$. Theorem 1 also holds for the convex function with $H = 0$. Let $Var_{s_1:s_i}[\cdot]$ denote the conditional variance, given that we are at $\boldsymbol{\theta}^{(i)}$.

THEOREM 1. *Suppose* $L(\boldsymbol{\theta})$ *is* H*-strongly convex and* L*-smooth. If* $\alpha^{(i)} \geq \alpha_0$ *for some* $\alpha_0 > 0$ *and* $\alpha^{(i)} \leq \alpha^{max}$ *for some* $\alpha^{max} < \frac{1}{(\nu^2+\kappa^2+1)L}$, $\forall i$, *where* L *denotes the Lipschitz constant, the AS-SGD procedure satisfies*

$$\min_{i\in[M]}\left(2\alpha^{(i)}(1-\alpha^{(i)}(\nu^2+1)L)\right)\mathbb{E}\left[\left(L(\boldsymbol{\theta}^{(i)}) - L(\boldsymbol{\theta}^*)\right)\right]$$
$$\leq \frac{1}{M}\left(\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(1)} - \boldsymbol{\theta}^*\right\|^2\right] - \mathbb{E}\left[\left\|\boldsymbol{\theta}^{(M+1)} - \boldsymbol{\theta}^*\right\|^2\right]\right)$$
$$+\frac{1}{M}\sum_{i=1}^{M}\left((\alpha^{(i)})^2\mathbb{E}\left[\frac{Var_{s_1:s_i}\left[\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T\nabla L(\boldsymbol{\theta}^{(i)})\right]}{\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2}\right]\right) \tag{2.31}$$

where $[M] = \{1, 2, \cdots, M\}$. *Furthermore,*

$$\lim_{M\to\infty}\min_{i\in[M]} L(\boldsymbol{\theta}^{(i)}) - L(\boldsymbol{\theta}^*) = 0, \tag{2.32}$$

and because $L(\boldsymbol{\theta}^{(i)})$ *is non-increasing over iterations, it holds*

$$\lim_{M\to\infty} L(\boldsymbol{\theta}^{(M)}) = L(\boldsymbol{\theta}^*). \tag{2.33}$$

In the non-convex case, it is hard to bound $L(\boldsymbol{\theta}^{(i)}) - L(\boldsymbol{\theta}^*)$. However, Theorem 2 shows that AS-SGD achieves the convergence to a stationary point. This point can either be a saddle point or a local optimal point when the true loss function is non-convex.

THEOREM 2. *Suppose $L(\boldsymbol{\theta})$ is L-smooth and bounded. If $\alpha^{(i)} \geq \alpha_0$ for some $\alpha_0 > 0$ and $\alpha^{(i)} \leq \alpha^{max}$ for some $\alpha^{max} < \frac{1}{(\nu^2+\kappa^2+1)L}$, $\forall i$, the AS-SGD procedure satisfies*

$$\lim_{M\to\infty} \min_{i\in[M]} E\left(\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|\right) = 0. \tag{2.34}$$

The results in Theorems 1 and 2 are built on Lemma 1 where we use the inner product and orthogonality tests. The reduced variance via stratified sampling helps satisfy these tests more often without increasing sample sizes, thus saving computational efforts. In particular, in Theorem 1, the proposed stratified sampling reduces the variance term $Var_{s_1:s_i}\left[\nabla F_{_{S-SGD}}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)})\right]$ in (2.31), so we can get a tighter upper bound. The tighter bound would help the expected loss close to the optimality, accelerating the optimization procedure. Similarly, to obtain the result in Theorem 2, we use the bound in Lemma 1 where a tighter upper bound for $\mathbb{E}_{s_1:s_i}\left[\left\|\nabla F_{_{S-SGD}}(\boldsymbol{\theta}^{(i)})\right\|^2\right]$ can be achieved via the reduced variance term $Var_{s_1:s_i}\left[\nabla F_{_{S-SGD}}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)})\right]$.

It should be noted that the theoretical results in this section are derived under the AS-SGD procedure. With S-SGD, if $\mathbb{E}\left[\left\|\nabla F_{_{S-SGD}}(\boldsymbol{\theta}^{(i)})\right\|^2\right]$ is bounded, similar results would be derived. Our implementation results in Sections 3 and 4 demonstrate that S-SGD provides comparable results with AS-SGD in most cases. This is due to the reduced variance by using the stratified sampling, when an appropriate sample size is used in S-SGD.

3. Numerical Examples. We present our implementation results using various numerical examples to compare the performances of our proposed methods, S-SGD and AS-SGD, with alternative methods, including SGD, A-SGD and Bayesian calibration method.

3.1. *Problem Setting.* We consider five problems with different features.

1. Example 1 (Ex 1)
   - Physical process: $y(x) = m(x) + e$ with $m(x) = \exp(x/10)\sin x$ for $x \sim U(0, 2\pi)$ and $e \sim N(0, 0.1)$.
   - Simulation model: $y^c(x, \theta) = m(x) - |\theta + 1|(\sin\theta x + \cos\theta x)$, where the true θ is -1. The simulation model is a perfect computer model with the true parameter.
2. Example 2 (Ex 2)
   - Physical process: same as in Ex 1.
   - Simulation model: $y^c(x, \theta) = m(x) - \sqrt{\theta^2 - \theta + 1}(\sin\theta x + \cos\theta x)$. The simulation model is an imperfect computer model where the true θ is not available.
3. Example 3 (Ex 3)
   - Physical process: $y(x) = m(x) + e$ with $m(x) = -(x-2)^2 + 4$ for $x \sim U(0, 4)$ and $e \sim N(0, |x-2|)$.
   - Simulation model: $y^c(x, \theta) = -(x-\theta)^2 + 4$. The simulation model is a perfect computer model with the true parameter $\theta = 2$.
4. Example 4 (Ex 4)
   - Physical process: $y(\boldsymbol{x}) = m(\boldsymbol{x}) + e$ with

$$m(\boldsymbol{x}) = \left(1 - \exp\left(-\frac{1}{2x_2}\right)\right) \frac{0.2 \cdot 1000x_1^3 + 1900x_1^2 + 2092x_1 + 60}{0.1 \cdot 100x_1^3 + 500x_1^2 + 4x_1 + 20}$$

for $x_1, x_2 \sim U(0,4)$ and $e \sim N(0, 0.5)$.

- Simulation model:

$$y^c(\boldsymbol{x}, \theta) = \left(1 - \exp\left(-\frac{1}{2x_2}\right)\right) \frac{2\theta \cdot 1000x_1^3 + 1900x_1^2 + 2092x_1 + 60}{\theta \cdot 100x_1^3 + 500x_1^2 + 4x_1 + 20},$$

which is a perfect computer model with the true parameter $\theta = 0.1$.

5. Example 5 (Ex 5)
   - Physical process: $y(x) = m(x) + e$ with $m(x) = (x_1 - 2)^2 + (x_2 - 2)^2$ for $x_1, x_2 \sim U(0,4)$ and $e \sim N(0, |x_2 - 2|)$.
   - Simulation model: $y^c(x, \theta) = (x_1 - \theta)^2 + (x_2 - \theta)^2$. The simulation model is a perfect computer model with the true parameter $\theta = 2$.

Ex 1 and Ex 2 are taken from Tuo and Wu (2015); Ex 1 represents an example with a perfect computer model, whereas Ex 2 shows the case with an imperfect computer model. In Ex 3, the noise is heterogeneous; the variance of y increases, as x is away from 2. In these examples, we do not use the bold notations for input and output, because they are 1-dimensional. Ex 4, taken from Gramacy et al. (2015), includes the 2-dimensional input vector, x_1 and x_2. Lastly, Ex 5 represents a 2-dimensional case with heterogeneous noise variance.

For each numerical example, we conduct 1000 independent experiments. Because SGD-based approaches cannot guarantee the global optimality when the loss function is non-convex, which is unknown in the black-box optimization, we use multiple starting points for θ. In our implementation, we use 5 starting points. In each experiment, we randomly draw 1000 data points from the aforementioned physical and simulation models to form a training set. After obtaining the final θ from each of the five starting points, we choose the best θ that provides the smallest RMSE in the training set, computed as follows.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N_T} (y(x_i) - y^c(x_i, \theta))^2}{N_T}}, \tag{3.1}$$

where N_T is the size in the training set. When RMSE is used to evaluate the calibration performance in the testing set, N_T becomes the testing set size.

In each iteration in SGD and S-SGD, 10% of the training data (that is, $n = 100$) are sampled to update the parameter at each iteration. We terminate the iteration when the relative difference between the current parameter value and the previous value becomes smaller than 10^{-3} in all of the four methods. In AS-SGD, the initial and incremental sample sizes at each iteration are set at 100. In implementing the stratified sampling in S-SGD and AS-SGD, we employ the CART method to decide the strata at each iteration, including the number of strata, partitioning variables and points. In particular, to determine the tree size, we prune a full tree using the cross-validation technique (Therneau and Atkinson, 2019). In implementing CART within our framework, we use R-package `rpart` (Therneau et al., 2019) and use the complexity parameter (cp) to decide the tree size.

3.2. *Implementation results.* We compare the results from SGD, A-SGD, S-SGD, and AS-SGD in Tables 1-3. First, we evaluate the estimation accuracy. Table 1 reports the average of calibrated parameter values from 1000 experiments. The results indicate that all methods calibrate the parameters well, ending in values close to the true values. In Ex 2 where the simulation model is imperfect, we cannot evaluate the calibration accuracy, but all methods produce similar values. Later we will compare our results with those from the Bayesian calibration in Section 3.4.

Next, we evaluate the estimation efficiency of the four SGD-based approaches. Table 2 reports the average number of iterations in each method. A-SGD uses the least number of

	Ex 1	Ex 2	Ex 3	Ex 4	Ex5
SGD	-1.00	-0.18	2.01	0.09	1.95
ASGD	-1.03	-0.18	2.01	0.09	1.95
S-SGD	-0.98	-0.13	2.01	0.09	1.95
AS-SGD	-0.98	-0.13	2.01	0.09	1.95
True value	-1.00	N/A	2.00	0.10	2.00

TABLE 1
Average of calibrated parameter values

iterations in several cases, but S-SGD and AS-SGD also require a reasonably small number of iterations in all cases. It should be also noted that S-SGD converges much faster than SGD in Ex 3, Ex 4, and Ex 5, demonstrating the benefits of the stratified sampling.

	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5
SGD	5.06	5.83	39.87	151.64	19.47
ASGD	**3.66**	5.14	**8.15**	**4.66**	14.23
S-SGD	3.83	**4.44**	10.53	6.53	8.60
AS-SGD	3.82	4.52	9.64	6.71	**8.56**

TABLE 2
Average number of iterations in 1000 experiments (Number in bold font is the smallest average number of iterations among the studied methods.)

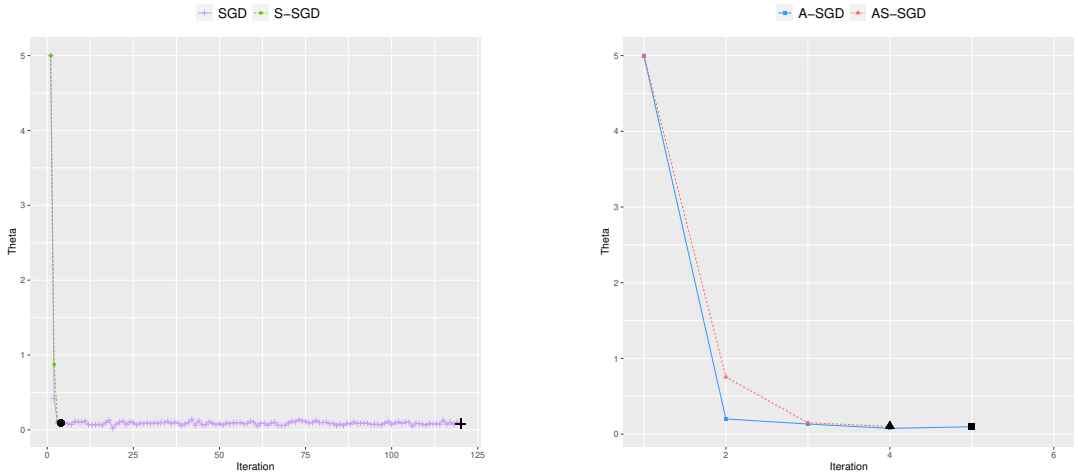
While A-SGD converges fast in Ex 1, Ex 3, and Ex 4, it does so at the expense of large computational budgets. Table 3 summarizes the average total number of data samples until convergence with five different starting points in 1000 experiments. In all examples, A-SGD uses a much larger number of data samples, compared to S-SGD and AS-SGD. We cap the sample size at each iteration at the size of the training set (that is, 1000) in A-SGD and AS-SGD. We observe that the sample size of A-SGD grows rapidly and it eventually samples the almost entire dataset in later iterations in most cases. Regarding the comparison with SGD, the number of data samples used in S-SGD is smaller than SGD in all examples. Notably, the advantage of stratification becomes more substantial in Ex 3, Ex 4, and Ex 5. Between S-SGD and AS-SGD, AS-SGD performs slightly better than S-SGD in Ex 3, whereas S-SGD uses fewer samples in other cases. These results in Table 3 indicate that the benefits of computational efficiency from stratification are clear when the noise exhibits heterogeneous variance (Ex 3 and Ex 5) and the input is multi-dimensional (Ex 4 and Ex 5).

	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5
SGD	2531.60	2913.20	19935.50	75819.70	9733.40
ASGD	11265.20	17034.88	10841.29	6969.26	43567.71
S-SGD	**1914.50**	**2218.20**	5263.20	**3264.00**	**4299.90**
AS-SGD	2218.20	2556.80	**5180.10**	4565.40	6683.60

TABLE 3
Average of total number of data samples in 1000 experiments (Number in bold font shows the smallest number of samples among the studied methods.)

To further illustrate the performance differences among the methods, we investigate how $\theta^{(i)}$ changes. Figure 4 depicts one of the trajectories of $\theta^{(i)}$ over iterations from each method. Because all of the four methods converge relatively fast in Ex 1 $\sim$ Ex 3 and Ex 5, we compare the trends in Ex 4 only. We show the trajectories when the initial parameter is set to be

5, which is far from the true value. The different patterns between SGD and S-SGD in Figure 4(a) demonstrate that the stratification accelerates the convergence. Both SGD and S-SGD converge in a similar manner at early iterations. However, the fluctuation of the parameter in SGD is essentially random on the plateau over many iterations. This occurs because the variance of SGD is large and thus, with a small-size random sample, the result changes iteration by iteration. On the contrary, S-SGD controls the variance by reallocating budgets over the input space, so it does not show such fluctuating patterns. In the comparison between A-SGD and AS-SGD in Figure 4(b), their convergence patterns are similar. Recall that A-SGD uses large budgets at each iteration, which helps stabilizing the variance as reported in Table 3. AS-SGD, with a smaller sample size than A-SGD, shows fast convergence.



(a) Ex 4 with SGD and S-SGD

(b) Ex 4 with A-SGD and AS-SGD

Fig 4: Trajectory of parameter value over iterations. Black solid markers represent the final values at the last iteration in each method.

Lastly, we study how the budget is reallocated in the proposed stratified approach. In the CART-based stratification, the number of strata and the partitioning points dynamically change over iterations. So, it is difficult to show how S-SGD changes the weights, as iterations proceed. Thus, for the illustration purpose, we evenly divide the input domain into 4 fixed strata and show the budget in each stratum in Figure 5 for Ex 3. Here, we divide the input domain into the four equally distanced strata, (0,1], (1,2], (2,3], and (3,4). The initial weight across the four strata is the same, that is, 25%. Recall that in Ex 3, the output variance gets larger, as the input is away from 2. We can see that S-SGD assigns more weights on the first and last strata where the variance is large. We observe similar patterns in AS-SGD, so we omit the results of AS-SGD.

3.3. *Sensitivity Analysis.* The AS-SGD needs to set κ and ν in the inner product test in (2.29) and the orthogonality test in (2.30), respectively. We set their values as suggested in Bollapragada, Byrd and Nocedal (2018). For κ, the main idea is to control the sample gradient close to the true gradient and thus, to have the descent direction frequently that satisfies $\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)}) > 0$. Bollapragada, Byrd and Nocedal (2018) set the lower bound of its one-sided prediction interval to be positive. With $\kappa = 0.9$ the probability of attaining the positive lower bound is sufficiently large. For the choice of ν, it controls the tangent of the

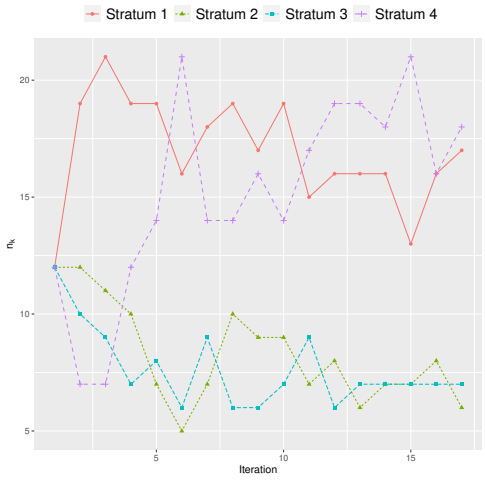


Fig 5: Computational budget reallocation in S-SGD with equally distanced strata

angle between $\nabla F_{SGD}(\boldsymbol{\theta}^{(i)})$ and $\nabla L(\boldsymbol{\theta}^{(i)})$. It needs to keep the angle away from $90°$, but at the same time the angle should be large enough to avoid increasing sample size in most iterations. For this, $\nu = tan80° = 5.84$ is suggested. For more details, we refer to Bollapragada, Byrd and Nocedal (2018).

In this section, to investigate the influence of these parameters on the computational performance, we conduct sensitivity analysis in a range of settings, summarized in Tables 4-6. The results suggest that the performance of AS-SGD is not sensitive to different parameter settings in terms of the number of iterations, number of samples, and calibrated parameter values.

	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5
$\kappa = 0.6, \nu = 5.84$	3.92	4.68	9.20	6.50	8.62
$\kappa = 0.7, \nu = 5.84$	3.94	4.67	9.43	6.70	8.59
$\kappa = 0.8, \nu = 5.84$	3.90	4.62	9.67	6.64	8.60
$\kappa = 0.9, \nu = 5.84$	3.82	4.52	9.64	6.71	8.56
$\kappa = 0.95, \nu = 5.84$	3.87	4.54	9.76	6.56	8.62
$\kappa = 0.9, \nu = 5$	3.84	4.55	9.68	6.62	8.60
$\kappa = 0.9, \nu = 10$	3.85	4.56	9.71	6.66	8.57

TABLE 4

Average number of iterations in 1000 experiments with different κ and ν in AS-SGD.

	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5
$\kappa = 0.6, \nu = 5.84$	2514.80	2880.70	5189.10	4964.00	7786.60
$\kappa = 0.7, \nu = 5.84$	2428.00	2762.60	5195.10	4926.70	7294.50
$\kappa = 0.8, \nu = 5.84$	2304.30	2672.30	5245.60	4741.20	6938.50
$\kappa = 0.9, \nu = 5.84$	2218.20	2556.80	5180.10	4565.40	6683.60
$\kappa = 0.95, \nu = 5.84$	2220.70	2546.50	5231.10	4426.20	6635.00
$\kappa = 0.9, \nu = 5$	2235.30	2578.40	5200.90	4518.30	6644.8
$\kappa = 0.9, \nu = 10$	2239.10	2582.80	5199.60	4555.40	6632.10

TABLE 5

Average of total number of data samples in 1000 experiments with different κ and ν in AS-SGD.

	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5
$\kappa = 0.6, \nu = 5.84$	-1.01	-0.15	2.01	0.09	1.95
$\kappa = 0.7, \nu = 5.84$	-1.00	-0.14	2.01	0.09	1.95
$\kappa = 0.8, \nu = 5.84$	-1.00	-0.13	2.01	0.09	1.95
$\kappa = 0.9, \nu = 5.84$	-0.98	-0.13	2.01	0.09	1.95
$\kappa = 0.95, \nu = 5.84$	-0.99	-0.15	2.01	0.09	1.95
$\kappa = 0.9, \nu = 5$	-1.00	-0.13	2.01	0.09	1.95
$\kappa = 0.9, \nu = 10$	-1.00	-0.13	2.01	0.09	1.95
True value	-1.00	N/A	2.00	0.10	2.00

TABLE 6

Average of calibrated parameter values with different κ and ν in AS-SGD

Moreover, we conduct numerical experiments with different incremental sample sizes n_c's in AS-SGD and summarize the results in Tables 7-9. Let n_{c_0} denote the initial batch size at each iteration. While a smaller n_c uses fewer number of samples in Table 8, the difference is not significant. Overall, the results indicate that different incremental sample sizes result in comparable performance.

	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5
$n_c = 0.5n_{c_0}$	3.87	4.55	9.78	6.74	8.57
$n_c = n_{c_0}$	3.82	4.52	9.64	6.71	8.56
$n_c = 1.5n_{c_0}$	3.93	4.54	9.62	6.72	8.58

TABLE 7

Average number of iterations in 1000 experiments with different incremental sample size in AS-SGD.

	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5
$n_c = 0.5n_{c_0}$	2096.00	2440.60	5094.65	4571.20	6536.05
$n_c = n_{c_0}$	2218.20	2556.80	5180.10	4565.40	6683.60
$n_c = 1.5n_{c_0}$	2397.80	2692.60	5315.50	4647.20	6740.25

TABLE 8

Average of total number of data samples in 1000 experiments with different incremental sample size in AS-SGD.

	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5
$n_c = 0.5n_{c_0}$	-1.00	-0.13	2.01	0.09	1.95
$n_c = n_{c_0}$	-0.98	-0.13	2.01	0.09	1.95
$n_c = 1.5n_{c_0}$	-0.99	-0.14	2.01	0.09	1.95
True value	-1.00	N/A	2.00	0.10	2.00

TABLE 9

Average of calibrated parameter values with different incremental sample size in AS-SGD

3.4. *Comparison with Bayesian calibration.* We implement the Bayesian calibration model proposed by Kennedy and O'Hagan (2001), that is, the full model with discrepancy term. In our implementation, we use the R-package `calibrator` (Hankin, 2019). We adopt a normal prior distribution for θ and estimate θ through a three-stage optimization algorithm, following the procedure in `calibrator`. Specifically, in the first two stages, we estimate

some auxiliary hyperparameters in the Bayesian calibration model and at the last stage, we estimate θ conditioned on those hyperparameters.

In Ex 1-3 and Ex 5, we use 1,000 data points in the training set, which are the same sample sizes as in our approach. In Ex 4, 2,000 training data points are used for the Bayesian approach, which is twice as many as that in the proposed approach. We also observe that the Bayesian approach is sensitive to the choice of prior. We consider two cases, one with the prior mean close to the true value (Bayesian-1) and the other with a randomly chosen prior mean (Bayesian-2). The variance of prior was set to be 5 in all cases. The Bayesian approach is computationally intensive. With 1000 data points, each experiment requires around 5 days to run.

The second and third rows of Table 10 summarize the results from 25 experiments of the Bayesian approach, whereas the fourth and fifth rows are from the 1000 experiments of the proposed approaches. The Bayesian method is sensitive to the prior specification. Even with the correctly specified prior in Bayesian-1, the calibrated parameters deviate from the true values in most cases. When the prior mean is chosen randomly in Bayesian-2, the estimated parameters are quite different from the true values. On the contrary, our procedures generate results closer to the true parameter values in most cases.

	Ex 1	Ex 2	Ex 3	Ex 4	Ex 5
Bayesian-1	-1.42	-0.26	1.93	0.14	2.65
Bayesian-2	2.95	1.08	2.87	0.23	3.17
S-SGD	-0.98	-0.13	2.01	0.09	1.95
AS-SGD	-0.98	-0.13	2.01	0.09	1.95
True value	-1.00	N/A	2.00	0.10	2.00

TABLE 10

Comparison of calibrated parameter values between Bayesian and proposed approaches.

3.5. *Uncertainty Quantification Results.* Table 11 summarizes the CIs for the parameters from S-SGD using (2.26)-(2.28). In these examples, the loss function is convex in Ex 3 only. However, we implement the procedure in all examples except Ex 2 (note that Ex 2 represents the incomplete computer model and there is no true value). Recall that we use 5 starting points and choose the best estimate that yields the smallest RMSE. The resulting estimate and its trajectory are used to construct the CI. The second row reports the average parameter value $\pm$ 1.96 times the average standard deviation from 1000 experiments for each example. The third row shows the percentage of CIs that include the true value. We note that the confidence interval covers the true parameter with a high probability so the coverage rates are close to 100%. This is because the parameter iterate converges to a point that is very close to the true parameter in most cases.

	Ex 1	Ex 3	Ex 4	Ex 5
CI	-1.00 ± 0.09	2.01 ± 0.08	0.09 ± 0.07	2.01 ± 0.18
Coverage rate	100.0%	99.5%	99.8%	100.0%
True value	-1.00	2.00	0.10	2.00

TABLE 11

Results for 95% confidence interval using S-SGD from 1000 experiments

We believe similar results would hold for AS-SGD with adaptive sample sizes. However, it is not straightforward to derive the covariance matrix S in (2.26), because the sample size

varies throughout iterations in AS-SGD. Constructing an asymptotically valid CI in AS-SGD is a subject of our future study.

4. Case Study. In this section, we use the multiple wake model in a multi-turbine wind farm setting (Katic, Højstrup and Jensen, 1986) and estimate the wake decay coefficient using data collected from two operational wind farms, WF 1 and WF 2. Table 12 summarizes the information about the wind farms used in this study. Due to data confidentiality, detailed information cannot be provided. Figure 6 shows the layout of WF1, where we make modifications to the original layouts, including omitting several turbines, due to confidentiality.

Wind farm	Location	Data size	Number of turbines	Wind farm layout
WF 1	offshore	1206	36	regular
WF 2	land-based	1659	40	irregular

TABLE 12
Information about two wind farms in the case study

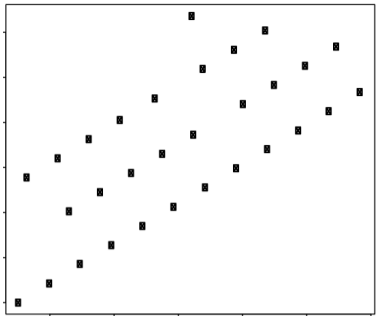


Fig 6: Layout of WF 1. Solid circles represent turbine locations.

In WF 1 and WF 2, each data records the power output from turbines and environmental measurements (wind speed, turbulence intensity) collected at the meteorological tower (or met mast). In this case study, we use data collected when the met mast is not under wake because the input of the Jensen's model should be the free-flow wind speed. In our datasets, 1206 and 1659 data points, respectively, satisfy this condition in WF 1 and WF 2. Admittedly, these data sizes are not large-scale. But we use these datasets to demonstrate the benefits of the proposed approach. For the thrust coefficient in Jensen's model, we use the default setting which is 8/9.

As input variables, we use the 10-minute average wind speed and turbulence intensity. Here, turbulence intensity is defined as the ratio of the standard deviation of wind speed to the average wind speed during the 10-minute interval (Molland and Turnock (2011)), quantifying the stability of wind. The turbulence intensity ranges from 0.1 to 1.5 in WF 1 and from 0.3 to 2.5 in WF 2. The wind speed ranges from 3 to 15 m/s in WF 1 and from 4 to 13 m/s in WF 2.

For the output variables, we use the generated power from turbines in the wind farms. It should be noted that the Jensen's model generates the wind speed deficit at each downstream turbine as the model output, but it does not compute the generated power. As such, we estimate the power curve that connects the incoming wind speed to the power output. All turbines in each wind farm considered in this study have the same specification, so we assume that every turbine in the same wind farm exhibits the same power curve. To build the power curve,

we use data from upstream turbines in the front row (which are not under wake) and employ a B-spline function (Lee et al., 2013). Once the Jensen's model generates the incoming wind speed at each downstream turbine, we plug the resulting speed into the power curve and get the power output as the simulation model output. In other words, our simulation model is the combination of Jensen's wake model and the power curve.

We randomly choose 70% of data points as a training set to calibrate the parameter and use the remaining set to evaluate the estimation accuracy. Similar to the settings in the numerical examples, in each iteration in SGD and S-SGD, 10% of the training data are used to update the parameter at each iteration. In A-SGD and AS-SGD, the initial and incremental sample size at each iteration is set at 10% of training data. The stopping criteria are when the relative difference between the current parameter value and the previous value becomes smaller than 10^{-3}. We repeat the experiment independently 100 times for each method.

Table 13 summarizes the case study results for WF 1, including the average number of iterations in the second column, the total number of samples in the third column, and resulting wake decay parameter values in the fourth column. The last column shows the normalized RMSE that compares the power outputs from the simulation model with the calibrated parameter to actual power outputs in the testing set. In WF 1, S-SGD reduces the average number of iterations and computational budget over SGD by about 10%. The advantage of stratification becomes much more obvious with AS-SGD. AS-SGD reduces the number of iterations and computational budget substantially, compared to SGD and A-SGD. In particular, AS-SGD reduces the number of samples by 27% and 87% over SGD and A-SGD, respectively. Lastly, in terms of the estimation accuracy, four SGD-based methods show similar accuracy in terms of the calibrated parameter values and RMSE in both wind farms, as shown in the last two columns.

	Avg. Iter.	Total # of samples	θ	RMSE
SGD	21.98	9343.20	0.10	0.07
ASGD	19.94	52617.31	0.10	0.07
S-SGD	19.47	8275.60	0.10	0.07
AS-SGD	**11.21**	**6780.90**	0.10	0.07

TABLE 13

Performance comparison in WF 1 (Number in the bold font shows the best performance among the studied methods.

Figure 7 further shows the example of parameter trajectory over iterations in each method for WF 1. The resulting values are similar in the four methods, but AS-SGD stops the iteration much earlier than other methods. Similar to the observation made in the numerical examples, the pattern at earlier iterations are similar, that is, all methods quickly move the parameter. In SGD, however, the parameter varies more, requiring more iterations until convergence.

Next, we compare the sample size changes in the two adaptive methods, A-SGD and AS-SGD, in Figure 8 for WF1. A-SGD increases the sample size rapidly throughout the iteration and uses all the data points in the training dataset in the end. On the contrary, AS-SGD keeps the sample size at the same level for a while and only employs a larger set in the last iteration.

Similarly, the proposed variance-reduced approaches show advantages for WF 2, as summarized in Table 14. Although A-SGD takes the least number of iterations, it uses the largest number of samples, more than twice those of S-SGD and AS-SGD. Conversely, AS-SGD converges with the least number of samples, and S-SGD exhibits similar performance. As for the estimation accuracy, the four SGD-based methods show comparable accuracy in terms of the calibrated parameter values and RMSE in both wind farms, as shown in the last two columns of Table 14.

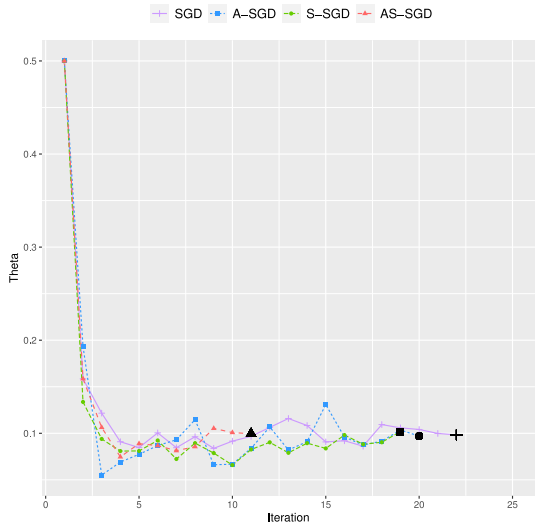


Fig 7: Parameter trajectory. Black solid markers represent the final values in each method.

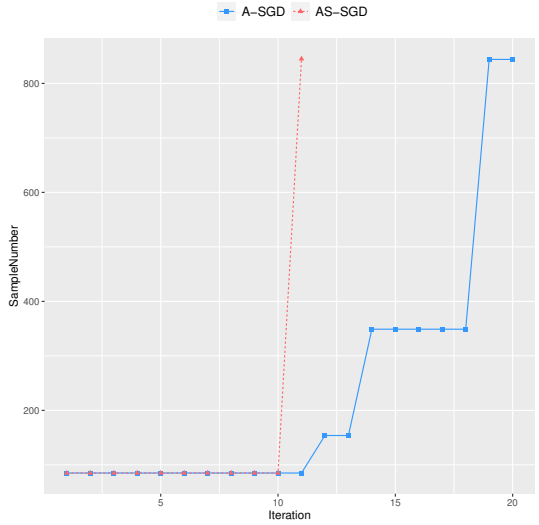


Fig 8: Examples of sample size trajectory in A-SGD and AS-SGD

	Avg. Iter.	Total # of samples	θ	RMSE
SGD	12.51	7317.18	0.12	0.10
ASGD	**4.28**	12413.16	0.12	0.10
S-SGD	9.60	5910.17	0.12	0.10
AS-SGD	9.07	**5870.46**	0.12	0.10

TABLE 14

Performance comparison in WF 2 (Number in the bold font shows the best performance among the studied methods.

5. Conclusion. We study the parameter calibration problem when data sizes from both physical systems and computer experiments are large. We formulate the problem in the

stochastic optimization framework. Despite the opportunities that the abundance of data brings, making reliable estimations can be still time-consuming and computationally expensive when the data size is large. The proposed SGD approach, enhanced with the variance reduction mechanisms, relieves computational burden substantially. Our approach also leads to more reliable estimation results, compared to the traditional Bayesian approach. We validate the estimation performance of our approach with a wide range of numerical settings and wind farm case study. Our analysis shows that the proposed approach is beneficial when the noise variance is heterogeneous and the influence of input variables is different.

In the future, we plan to study other variance reduction techniques, such as importance sampling (Choe, Byon and Chen, 2015; Choe, Pan and Byon, 2016; Li, Ko and Byon, 2021), in more depth and investigate their advantages and disadvantages in a wide variety of problem circumstances. We will also study other stochastic optimization approaches including trust-region based optimization (Shashaani, Hashemi and Pasupathy, 2018). Wake effects also influence the reliability of downstream turbines. We will investigate how the wake influences structural and mechanical loads on the turbine structure (Choe, Pan and Byon, 2016).

Acknowledgements. The authors thank the editor, associate editor and reviewers for suggestions that helped us greatly improve this manuscript. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

BIBLIOGRAPHY

INTERNATIONAL ENERGY AGENCY (2015). Medium-Term Renewable Energy Market Report 2015 - Market Analysis and Forecasts to 2020 Technical Report.

AINSLIE, J. F. (1988). Calculating the flowfield in the wake of wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics* **27** 213–224.

ÁLVAREZ, M. A. and LAWRENCE, N. D. (2011). Computationally efficient convolved multiple output Gaussian processes. *The Journal of Machine Learning Research* **12** 1459–1500.

BARTHELMIE, R. J. and PRYOR, S. (2013). An overview of data for wake model evaluation in the Virtual Wakes Laboratory. *Applied energy* **104** 834–844.

BARTHELMIE, R. J., PRYOR, S. C., FRANDSEN, S. T., HANSEN, K. S., SCHEPERS, J. G., RADOS, K., SCHLEZ, W., NEUBERT, A., JENSEN, L. E. and NECKELMANN, S. (2010). Quantifying the impact of wind turbine wakes on power Output at offshore wind farms. *Journal of Atmospheric and Oceanic Technology* **27** 1302-1317.

BOLLAPRAGADA, R., BYRD, R. and NOCEDAL, J. (2018). Adaptive sampling strategies for stochastic optimization. *SIAM Journal on Optimization* **28** 3312–3343.

BREIMAN, L., FRIEDMAN, J., STONE, C. J. and OLSHEN, R. A. (1984). *Classification and regression trees*. CRC press.

CHEN, X., LEE, J. D., TONG, X. T. and ZHANG, Y. (2020). Statistical inference for model parameters in stochastic gradient descent. *The Annals of Statistics* **48** 251 – 273.

CHOE, Y., BYON, E. and CHEN, N. (2015). Importance sampling for reliability evaluation with stochastic simulation models. *Technometrics* **57** 351–361.

CHOE, Y., PAN, Q. and BYON, E. (2016). Computationally efficient uncertainty minimization in wind turbine extreme load assessments. *Journal of Solar Energy Engineering* **138** 041012.

CHURCHFIELD, M. (2013). Review of Wind Turbine Wake Models and Future Directions (Presentation) Technical Report, National Renewable Energy Laboratory (NREL), Golden, CO.

DAMIANOU, A. C., TITSIAS, M. K. and LAWRENCE, N. D. (2016). Variational inference for latent variables and uncertain inputs in Gaussian processes. *The Journal of Machine Learning Research* **17** 1425–1486.

FANG, Y., XU, J. and YANG, L. (2018). Online bootstrap confidence intervals for the stochastic gradient descent estimator. *Journal of Machine Learning Research* **19** 1-21.

FRANDSEN, S. (1992). On the wind speed reduction in the center of large clusters of wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics* **39** 251 - 265.

GRAMACY, R. B., BINGHAM, D., HOLLOWAY, J. P., GROSSKOPF, M. J., KURANZ, C. C., RUTTER, E., TRANTHAM, M. and DRAKE, R. P. (2015). Calibrating a large computer experiment simulating raiative shock hydrodynamics. *The Annals of Applied Statistics* **9** 1141–1168.

GÖÇMEN, T., VAN DER LAAN, P., RÉTHORÉ, P.-E., DIAZ, A. P., LARSEN, G. C. and OTT, S. (2016). Wind turbine wake models developed at the technical university of Denmark: A review. *Renewable and Sustainable Energy Reviews* **60** 752 - 769.

HANKIN, R. (2019). calibrator: Bayesian calibration of complex computer codes. *R package, version 1.2-8*.

HIGDON, D., KENNEDY, M., CAVENDISH, J. C., CAFEO, J. A. and RYNE, R. D. (2004). Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing* **26** 448–466.

JENSEN, N. O. (1983). *A note on wind generator interaction*.

JOSEPH, V. R. and YAN, H. (2015). Engineering-driven statistical adjustment and calibration. *Technometrics* **57** 257-267.

KATIC, I., HØJSTRUP, J. and JENSEN, N. (1986). A simple model for cluster efficiency. In *European Wind Energy Association Conference and Exhibition* 407–410.

KENNEDY, M. C. and O'HAGAN, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63** 425–464.

KHEIRABADI, A. C. and NAGAMUNE, R. (2019). A quantitative review of wind farm control with the objective of wind farm power maximization. *Journal of Wind Engineering and Industrial Aerodynamics* **192** 45 - 73.

LARSEN, G. C. (1988). A simple wake calculation procedure Technical Report, Risø National Laboratory, Denmark.

LEE, G., BYON, E., NTAIMO, L. and DING, Y. (2013). Bayesian spline method for assessing extreme loads on wind turbines. *Annals of Applied Statistics* **7** 2034–2061.

LI, S., KO, Y. M. and BYON, E. (2021). Nonparametric importance sampling for wind turbine reliability analysis with stochastic computer models. *to appear in Annals of Applied Statistics*.

LOHR, S. L. (2010). *Sampling: Design and Analysis*, 2 ed. Brooks/Cole, Cengage Larning.

MEASE, D. and BINGHAM, D. (2006). Latin hyperrectangle sampling for computer experiments. *Technometrics* **48** 467-477.

MOLLAND, A. F. and TURNOCK, S. R. (2011). *Marine rudders and control surfaces: principles, data, design and applications*. Elsevier.

NING, S., BYON, E., WU, T. and LI, J. (2017). A sparse partitioned-regression model for nonlinear system–environment interactions. *IISE Transactions* **49** 814-826.

OWEN, A. B. (2013). *Monte Carlo theory, methods and examples*. Book in progress Online version available at https://statweb.stanford.edu/~owen/mc/.

PAQUETTE, C. and SCHEINBERG, K. (2020). A stochastic line search method with expected complexity analysis. *SIAM Journal on Optimization* **30** 349–376.

PAULO, R., GARCÍA-DONATO, G. and PALOMO, J. (2012). Calibration of computer models with multivariate output. *Computational Statistics & Data Analysis* **56** 3959 - 3974.

POLYAK, B. T. (1990). New stochastic approximation type procedures. *Automation and Remote Control* **7** 937-1008.

POLYAK, B. T. and JUDITSKY, A. B. (1992). Acceleration of Stochastic Approximation by Averaging. *SIAM Journal on Control and Optimization* **30** 838-855.

QIAN, P. Z. G. and WU, C. F. J. (2008). Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. *Technometrics* **50** 192-204.

RASMUSSEN, C. E. (2003). Gaussian processes in machine learning. In *Summer School on Machine Learning* 63–71. Springer.

DTU WIND ENERGY (2015). Wind resources for energy production of wind turbines. http://www.wasp.dk/waspdetailswakeeffectmodel. Accessed: 2015-09-18.

RUPPERT, D. (1988). Efficient estimations from a slowly convergent Robbins-Monro process Technical Report, *Cornell University Operations Research and Industrial Engineering*.

SHASHAANI, S., HASHEMI, F. S. and PASUPATHY, R. (2018). ASTRO-DF: A class of adaptive sampling trust-region algorithms for derivative-free stochastic optimization. *SIAM Journal on Optimization* **28** 3145–3176.

SNELSON, E. and GHAHRAMANI, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems* 1257–1264.

STAID, A. (2015). Statistical modeling to support power system planning, PhD thesis, Johns Hopkis University, Washington D.C.

THERNEAU, T. M. and ATKINSON, E. J. (2019). An introduction to recursive partitioning using the RPART routines. *R package*.

THERNEAU, T., ATKINSON, B., RIPLEY, B. and RIPLEY, M. B. (2019). rpart: recursive partitioning and regression trees. *R package, version 4.1-15*.

TUO, R. and WU, C. F. J. (2015). Efficient calibration for imperfect computer models. *The Annals of Statistics* **43** 2331–2352.

TUO, R. and WU, J. C. (2016). A theoretical framework for calibration in computer models: parameterization, estimation and convergence properties. *SIAM/ASA Journal on Uncertainty Quantification* **4** 767–795.

YOU, M., BYON, E., JIN, J. J. and LEE, G. (2017). When wind travels through turbines: A new statistical approach for characterizing heterogeneous wake effects in multi-turbine wind farms. *IISE Transactions* **49** 84-95.

YOU, M., LIU, B., BYON, E., HUANG, S. and JIN, J. (2018). Direction-dependent power curve modeling for multiple interacting wind turbines. *IEEE Transactions on Power Systems* **33** 1725-1733.

YUAN, J., NG, S. H. and TSUI, K. L. (2013). Calibration of stochastic computer models using stochastic approximation methods. *IEEE Transactions on Automation Science and Engineering* **10** 171-186.

ZWAKMAN, J. (2014). Wind turbines and the environment. https://www.wijkplatformsvelsen.nl/ijmuiden-noord/2014/11/23/windturbines-en-het-milieu/. Accessed: 2020-09-11.

APPENDIX A: PROOF OF LEMMA 1

Since the orthogonality test is satisfied, at iteration i we have

$$\begin{aligned}
&\mathbb{E}_{s_1:s_i}\left[\left\|\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)}) - \frac{\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)})}{\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2}\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2\right] \\
&\qquad = \mathbb{E}_{s_1:s_i}\left[\left\|\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})\right\|^2\right] - \frac{\mathbb{E}_{s_1:s_i}\left[(\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)}))^2\right]}{\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2} \\
&\qquad \leq \nu^2 \left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2,
\end{aligned}$$

Therefore, it holds

$$\mathbb{E}_{s_1:s_i}\left[\left\|\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})\right\|^2\right] \leq \frac{\mathbb{E}_{s_1:s_i}\left[(\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)}))^2\right]}{\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2} + \nu^2 \left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2.$$

By the definition of conditional variance, we have

$$\mathbb{E}_{s_1:s_i}\left[(\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)}))^2\right] = Var_{s_1:s_i}\left[\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)})\right] + \left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^4.$$

Therefore, we obtain

$$\begin{aligned}
\mathbb{E}_{s_1:s_i}\left[\left\|\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})\right\|^2\right] &\leq \frac{Var_{s_1:s_i}\left[\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)})\right] + \left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^4}{\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2} + \nu^2 \left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2 \\
&= \frac{Var_{s_1:s_i}\left[\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T \nabla L(\boldsymbol{\theta}^{(i)})\right]}{\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2} + (\nu^2+1)\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2 \qquad \text{(A.1)} \\
&\leq (1+\kappa^2+\nu^2)\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2,
\end{aligned}$$

where the last inequality holds due to the inner product test.

APPENDIX B: PROOF OF THEOREM 1

We first show that the expected loss is decreasing, because

$$\begin{aligned}\mathbb{E}_{s_1:s_i}\left[L(\boldsymbol{\theta}^{(i+1)})\right] &\leq L(\boldsymbol{\theta}^{(i)}) - \alpha^{(i)}\mathbb{E}_{s_1:s_i}\left[\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T\nabla L(\boldsymbol{\theta}^{(i)})\right] + \frac{L(\alpha^{(i)})^2}{2}\mathbb{E}_{s_1:s_i}\left[\left\|\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})\right\|^2\right] \\ &\leq L(\boldsymbol{\theta}^{(i)}) - \alpha^{(i)}\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2 + \frac{L(\alpha^{(i)})^2}{2}\left((\kappa^2+\nu^2+1)\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2\right). \\ &\leq L(\boldsymbol{\theta}^{(i)}) - \frac{\alpha^{(i)}}{2}\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2, \end{aligned} \tag{B.1}$$

where the second last inequality holds from Lemma 1 and the last equation is due to $\alpha^{(i)} < \frac{1}{(\kappa^2+\nu^2+1)L}$.

Next, we have

$$\begin{aligned}&\mathbb{E}_{s_1:s_i}\left[\left\|\boldsymbol{\theta}^{(i+1)} - \boldsymbol{\theta}^*\right\|^2\right] \\ &= \mathbb{E}_{s_1:s_i}\left[\left\|\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}^* - \alpha^{(i)}\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})\right\|^2\right] \\ &= \left\|\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}^*\right\|^2 - 2\alpha^{(i)}\mathbb{E}_{s_1:s_i}\left[\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T(\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}^*)\right] + (\alpha^{(i)})^2\mathbb{E}_{s_1:s_i}\left[\left\|\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})\right\|^2\right] \\ &= \left\|\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}^*\right\|^2 - 2\alpha^{(i)}\nabla L(\boldsymbol{\theta}^{(i)})^T(\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}^*) + (\alpha^{(i)})^2\mathbb{E}_{s_1:s_i}\left[\left\|\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})\right\|^2\right]\end{aligned}$$

By the law of total expectation, we obtain

$$\begin{aligned}&\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(i+1)} - \boldsymbol{\theta}^*\right\|^2\right] \\ &= \mathbb{E}\left[\left\|\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}^*\right\|^2\right] - 2\alpha^{(i)}\mathbb{E}\left[\nabla L(\boldsymbol{\theta}^{(i)})^T(\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}^*)\right] + (\alpha^{(i)})^2\mathbb{E}\left[\mathbb{E}_{s_1:s_i}\left[\left\|\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})\right\|^2\right]\right] \\ &\leq \mathbb{E}\left[\left\|\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}^*\right\|^2\right] - 2\alpha^{(i)}\mathbb{E}\left[\nabla L(\boldsymbol{\theta}^{(i)})^T(\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}^*)\right] + (\alpha^{(i)})^2\mathbb{E}\left[\frac{Var_{s_1:s_i}\left[\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T\nabla L(\boldsymbol{\theta}^{(i)})\right]}{\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2}\right] \\ &\qquad + (\alpha^{(i)})^2(\nu^2+1)\mathbb{E}\left[\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2\right],\end{aligned}$$

where the last inequality is from (A.1). Also, by convexity, we have

$$-2\alpha^{(i)}\mathbb{E}\left[\nabla L(\boldsymbol{\theta}^{(i)})^T(\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}^*)\right] \leq -2\alpha^{(i)}\mathbb{E}\left[(L(\boldsymbol{\theta}^{(i)}) - L(\boldsymbol{\theta}^*))\right].$$

Therefore, it holds

$$\begin{aligned}&\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(i+1)} - \boldsymbol{\theta}^*\right\|^2\right] \\ &\qquad \leq \mathbb{E}\left[\left\|\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}^*\right\|^2\right] - 2\alpha^{(i)}\mathbb{E}\left[(L(\boldsymbol{\theta}^{(i)}) - L(\boldsymbol{\theta}^*))\right]\end{aligned}$$

$$
\begin{aligned}
&+(\alpha^{(i)})^2\mathbb{E}\left[\frac{Var_{s_1:s_i}\left[\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T\nabla L(\boldsymbol{\theta}^{(i)})\right]}{\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2}\right]+(\alpha^{(i)})^2(\nu^2+1)\mathbb{E}\left[\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2\right]\\
\leq\ &\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(i)}-\boldsymbol{\theta}^*\right\|^2\right]-2\alpha^{(i)}\mathbb{E}\left[(L(\boldsymbol{\theta}^{(i)})-L(\boldsymbol{\theta}^*))\right]\\
&+(\alpha^{(i)})^2\mathbb{E}\left[\frac{Var_{s_1:s_i}\left[\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T\nabla L(\boldsymbol{\theta}^{(i)})\right]}{\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2}\right]+2(\alpha^{(i)})^2L(\nu^2+1)\mathbb{E}\left[(L(\boldsymbol{\theta}^{(i)})-L(\boldsymbol{\theta}^*))\right]\\
=\ &\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(i)}-\boldsymbol{\theta}^*\right\|^2\right]-2\alpha^{(i)}\mathbb{E}\left[(L(\boldsymbol{\theta}^{(i)})-L(\boldsymbol{\theta}^*))\right](1-\alpha^{(i)}(\nu^2+1)L)\\
&+(\alpha^{(i)})^2\mathbb{E}\left[\frac{Var_{s_1:s_i}\left[\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T\nabla L(\boldsymbol{\theta}^{(i)})\right]}{\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2}\right],
\end{aligned}
$$

where the second last equation uses the fact that $\mathbb{E}\left[\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2\right]\leq 2L\mathbb{E}\left[(L(\boldsymbol{\theta}^{(i)})-L(\boldsymbol{\theta}^*))\right]$ for the L-smooth function. By rearranging it, we have

$$
\begin{aligned}
&\left(2\alpha^{(i)}(1-\alpha^{(i)}(\nu^2+1)L)\right)\mathbb{E}\left[\left(L(\boldsymbol{\theta}^{(i)})-L(\boldsymbol{\theta}^*)\right)\right]\\
&\leq\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(i)}-\boldsymbol{\theta}^*\right\|^2\right]-\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(i+1)}-\boldsymbol{\theta}^*\right\|^2\right]+(\alpha^{(i)})^2\mathbb{E}\left[\frac{Var_{s_1:s_i}\left[\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T\nabla L(\boldsymbol{\theta}^{(i)})\right]}{\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2}\right].
\end{aligned}
$$

Summing and averaging the above inequality over $i=1,2,\cdots,M$ yields

$$
\begin{aligned}
&\frac{1}{M}\sum_{i=1}^{M}\left(2\alpha^{(i)}(1-\alpha^{(i)}(\nu^2+1)L)\right)\mathbb{E}\left[\left(L(\boldsymbol{\theta}^{(i)})-L(\boldsymbol{\theta}^*)\right)\right]\\
&\leq\frac{1}{M}\sum_{i=1}^{M}\left(\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(i)}-\boldsymbol{\theta}^*\right\|^2\right]-\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(i+1)}-\boldsymbol{\theta}^*\right\|^2\right]+(\alpha^{(i)})^2\mathbb{E}\left[\frac{Var_{s_1:s_i}\left[\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T\nabla L(\boldsymbol{\theta}^{(i)})\right]}{\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2}\right]\right),
\end{aligned}
$$

Thus, we have

$$
\begin{aligned}
&\min_{i\in[M]}\left(2\alpha^{(i)}(1-\alpha^{(i)}(\nu^2+1)L)\right)\mathbb{E}\left[\left(L(\boldsymbol{\theta}^{(i)})-L(\boldsymbol{\theta}^*)\right)\right]\\
&\leq\tfrac{1}{M}\left(\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(1)}-\boldsymbol{\theta}^*\right\|^2\right]-\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(M+1)}-\boldsymbol{\theta}^*\right\|^2\right]\right)\\
&\quad+\tfrac{1}{M}\textstyle\sum_{i=1}^{M}\left((\alpha^{(i)})^2\mathbb{E}\left[\frac{Var_{s_1:s_i}\left[\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T\nabla L(\boldsymbol{\theta}^{(i)})\right]}{\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2}\right]\right)
\end{aligned}
$$

where $[M]=\{1,2,\cdots,M\}$.

Finally, we show that as $M\to\infty$, $\lim_{M\to\infty}\min_{i\in[M]}\mathbb{E}\left(L(\boldsymbol{\theta}^{(i)})-L(\boldsymbol{\theta}^*)\right)=0$. By the inner product test, we have

$$
(\alpha^{(i)})^2\frac{Var_{s_1:s_i}\left[\nabla F_{S-SGD}(\boldsymbol{\theta}^{(i)})^T\nabla L(\boldsymbol{\theta}^{(i)})\right]}{\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2}\leq(\alpha^{(i)})^2\kappa^2\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2\leq 2(\alpha^{(i)})^2L\kappa^2(L(\boldsymbol{\theta}^{(i)})-L(\boldsymbol{\theta}^*)).
$$

Therefore, we get

$$\frac{1}{M}\sum_{i=1}^{M}\left(2\alpha^{(i)}(1-\alpha^{(i)}(\nu^2+1)L)\right)\mathbb{E}\left[\left(L(\boldsymbol{\theta}^{(i)})-L(\boldsymbol{\theta}^*)\right)\right]$$

$$\leq \frac{1}{M}\left(\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(1)}-\boldsymbol{\theta}^*\right\|^2\right]-\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(M+1)}-\boldsymbol{\theta}^*\right\|^2\right]\right)+\frac{1}{M}\sum_{i=1}^{M}\left(2(\alpha^{(i)})^2L\kappa^2\mathbb{E}\left[L(\boldsymbol{\theta}^{(i)})-L(\boldsymbol{\theta}^*)\right]\right),$$

which leads to

$$\frac{1}{M}\sum_{i=1}^{M}\left(2\alpha^{(i)}(1-\alpha^{(i)}(\nu^2+\kappa^2+1)L)\right)\mathbb{E}\left[\left(L(\boldsymbol{\theta}^{(i)})-L(\boldsymbol{\theta}^*)\right)\right]$$

$$\leq \frac{1}{M}\left(\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(1)}-\boldsymbol{\theta}^*\right\|^2\right]-\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(M+1)}-\boldsymbol{\theta}^*\right\|^2\right]\right).$$

It holds

$$\min_{i\in[M]}\left(2\alpha^{(i)}(1-\alpha^{(i)}(\nu^2+\kappa^2+1)L)\right)\mathbb{E}\left[L(\boldsymbol{\theta}^{(i)})-L(\boldsymbol{\theta}^*)\right]$$

$$\leq \frac{1}{M}\left(\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(1)}-\boldsymbol{\theta}^*\right\|^2\right]-\mathbb{E}\left[\left\|\boldsymbol{\theta}^{(M+1)}-\boldsymbol{\theta}^*\right\|^2\right]\right).$$

This result implies that

$$\lim_{M\to\infty}\min_{i\in[M]}\left(2\alpha^{(i)}(1-\alpha^{(i)}(\nu^2+\kappa^2+1)L)\right)\mathbb{E}\left(L(\boldsymbol{\theta}^{(i)})-L(\boldsymbol{\theta}^*)\right)=0.$$

The term $2\alpha^{(i)}(1-\alpha^{(i)}(\nu^2+\kappa^2+1)L)$ does not converge to zero since we assume $\alpha^{(i)}\geq\alpha_0$ for some $\alpha_0>0$ and $\alpha^{(i)}\leq\alpha^{max}$ for some $\alpha^{max}<\frac{1}{(\nu^2+\kappa^2+1)L}$, $\forall i$.

Hence we have

$$\lim_{M\to\infty}\min_{i\in[M]}\mathbb{E}\left[L(\boldsymbol{\theta}^{(i)})-L(\boldsymbol{\theta}^*)\right]=0. \tag{B.2}$$

Besides, $\mathbb{E}\left[L(\boldsymbol{\theta}^{(i)})\right]$ is decreasing, as shown in (B.1). Therefore, we have

$$\lim_{M\to\infty}\mathbb{E}\left[L(\boldsymbol{\theta}^{(M)})\right]-L(\boldsymbol{\theta}^*)=0.$$

APPENDIX C: PROOF OF THEOREM 2

Using the results in (B.1) and taking the total expectation, it holds

$$\frac{\alpha^{(i)}}{2}\mathbb{E}\left(\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2\right)\leq\mathbb{E}\left(L(\boldsymbol{\theta}^{(i)})-L(\boldsymbol{\theta}^{(i+1)})\right).$$

Therefore, we obtain

$$\frac{1}{M}\sum_{i=1}^{M}\frac{\alpha^{(i)}}{2}\mathbb{E}\left(\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2\right)\leq\frac{1}{M}\mathbb{E}\left(L(\boldsymbol{\theta}^{(1)})-L(\boldsymbol{\theta}^{(M+1)})\right).$$

This implies

$$\min_{i\in[M]} \frac{\alpha^{(i)}}{2}\mathbb{E}\left(\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2\right) \leq \frac{1}{M}\mathbb{E}\Big(L(\boldsymbol{\theta}^{(1)}) - L(\boldsymbol{\theta}^{(M+1)})\Big).$$

Therefore,

$$\lim_{M\to\infty}\min_{i\in[M]} \frac{\alpha^{(i)}}{2}\mathbb{E}\left(\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2\right) \to 0.$$

Because $\alpha^{(i)}$ is always lower bounded by a non-zero value, it holds

$$\lim_{M\to\infty}\min_{i\in[M]} \mathbb{E}\left(\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|^2\right) \to 0,$$

resulting in

$$\lim_{M\to\infty}\min_{i\in[M]} \mathbb{E}\left(\left\|\nabla L(\boldsymbol{\theta}^{(i)})\right\|\right) \to 0.$$