Wake Effect Parameter Calibration with Large-Scale Field Operational Data using Stochastic Optimization

Pranav Jain[a,1], Sara Shashaani[a,2,*], Eunshin Byon[b,3]

[a] *North Carolina State University*
915 Partners Way, Raleigh, NC, 27695, USA
[b] *University of Michigan*
33 Hanes Hall, Ann Arbor, MI 48109, USA

Abstract

This study aims to show the application of stochastic optimization for efficient and robust parameter calibration of engineering wake models. Standard values of the wake effect parameters are generally used to predict power using engineering wake models, but some recent studies have shown that these values do not result in accurate prediction. The proposed approach estimates the wake effect parameters using operational data available from actual wind farms to minimize the prediction error of the wake model by using trust-region optimization. To further improve computational efficiency, we implement stratified adaptive sampling. We employ decision trees to stratify the data and propose two ways of adapting the sampling budget to the constructed strata: budget allocation with dynamic weights and fixed weights. We extend our analysis to determine the functional relationship between the turbulence intensity and wake decay coefficient. Our experiments suggest that wake parameters or a functional relationship between turbulence intensity and wake decay coefficient may need adjustments (from assumed standard values) for a particular wind farm using operational data from that wind farm to characterize the wake effect better.

1. Introduction

In wind power systems, a wind turbine acts as an obstacle to the free stream wind speed resulting in the development of a wake effect that is characterized by reduced wind speed and increased turbulence in the downstream direction. A turbine present in the wake of another upstream turbine generates less power and is under more structural and mechanical load. In a wind farm, this wake phenomenon is amplified as a turbine might be affected by wakes due to multiple turbines. The wake effect is of major importance for various engineering applications like predicting the annual energy production and wind farm layout optimization [1, 2, 3, 4]. Thus it becomes necessary to numerically estimate these wake interactions to enhance the wind farm performance.

Various computational fluid dynamics (CFD) techniques, such as large-eddy simulations (LES), can be used to model the wake phenomenon accurately, but they are computationally overwhelming [5, 6]. The LES simulations on a single turbine performed by Sedaghatizadeh et al. [7] needed 200 hours to complete using a multi-core system. In Churchfield et al. [8], it took one million CPU hours to perform one LES simulation of 10 minutes of real-time operation for a wind farm consisting of 48 turbines. The LES simulations run by Breton et al. [9] were able to improve the computational time of the above wind farm but still needed ten thousand CPU hours. These CFD models provide useful insights into wake mechanisms over a broad spectrum of atmospheric conditions. However, enormous computational resource requirement limits the usability of these methods for simulating or understanding large-scale wind farm operations for wind farm optimization.

*Corresponding author
Email addresses: pjain23@ncsu.edu (Pranav Jain), sshashaa2@ncsu.edu (Sara Shashaani), ebyon@umich.edu (Eunshin Byon)
[1] Ph.D Candidate in Edward P. Fitts Department of Industrial and Systems Engineering
[2] Assistant Professor in Edward P. Fitts Department of Industrial and Systems Engineering
[3] Associate Professor in Department of Industrial and Operations Engineering

Another school of thought is to devise purely data-driven models. Göçmen and Giebel [10] applied a purely data-driven approach to estimate the power using Long short term memory recurrent neural network using one hour historical data to predict the wind speed at downstream turbines. In [11] and [12], new statistical models based on the Gaussian Markov random field are presented to characterize heterogeneous wind deficits over free-flow wind conditions. However, pure data-driven approaches cannot capture a general characteristic of wind flow inside a general wind farm. Instead their results are very specific to the data at hand (or specific to the studied wind farm) and may not be generalized to estimate or understand the wind field in other wind farms. While these models can achieve high accuracy, pure data-driven approaches may not adhere to physics and their results could be less interpretable than those of physics-based models.

This study focuses on the engineering wake models that offer computational simplicity by expressing the wake phenomena using parametric analytical formulations. One can classify the engineering models as kinematic models and field models. Field models are implicit models that estimate the flow field at each point in the downstream direction [13]. The Ainslie model [14] is one of the classic field models which determines the flow field by numerically estimating Reynold's Averaged Navier Stokes (RANS) equations. The field models are more sophisticated as compared to the kinematic models. Still, the computation time needed to solve these models renders them unable to be used for large-scale implementation, similar to the CFD-based wake models [15].

On the other hand, kinematic wake models are derived by solving the conservation of mass and momentum equations to get explicit formulations. The Jensen model, suggested by Jensen [16] and later modified by Katic et al. [17], is one of the earliest kinematic models. This model assumes that the wake propagates linearly in the downstream direction at a rate driven by a constant called the wake decay coefficient (WDC). Its wide use in the industry is due to its ease of implementation and reasonable accuracy. There are also newly developed extensions of the Jensen model to improve its accuracy. The model proposed by Frandsen et al. [18] assumes instantaneous wake expansion in the downstream direction and expresses this initial expansion rate in terms of the thrust coefficient. Bastankhah and Porté-Agel [19] proposed a model where a Gaussian wake shape is employed to model the wake profile. Gebraad et al. [20] developed a multizone model which assumes three different WDCs based on the region of the wake. While these models improved the accuracy of the power generated at each turbine, they also increased the number of parameters needed to characterize the wake. Larsen [21] proposed a model whose calibration is relatively complicated as they used the RANS equations in conjunction with mixing layer theory to model wake. Considering that the original Jensen model formulates the wake based on the wind speed deficit and does not consider the influence of turbulence on the wake, recent studies extend the model by expressing the WDC as a function of the hub height, surface roughness, and atmospheric stability [22, 23, 24]. Recently, Howland et al. [25] present a flow control model that looks at the collective effect of simultaneously yawed and waked turbines in a wind farm.

The research on kinematic wake models that has been mentioned thus far focuses on the analytical development of wake modeling. The accuracy of these models is significantly influenced by the choice of wake parameters. Since they are influenced by a variety of wind farm characteristics, such as geography, terrain effects, and wind farm layout, these parameters could be unique to each wind farm. This paper creates a novel way of identifying the appropriate values of wake parameters in the kinematic wake models using operational data from a wind farm. The primary advantage of the proposed methodology over the strategy that is solely data-driven is that it still upholds the "physics" embedded in the kinematic wake model, even while it makes use of operational data to enhance performance. More physics entails more interpretability and more trust in practical application. Schreiber et al. [26] used the Bastankhah and Porté-Agel's [19] wake model as their baseline model and identified the model parameters directly from operational data by expressing each parameter as the sum of a baseline constant value and correction term. Then they estimated the correction term via maximum likelihood estimation. This type of approach to identify parameter values using operational (or physical trial) data is called parameter calibration in the statistical literature. Parameter calibration involves tuning the parameters such a way that the prediction error of the analytical model is reduced and closely matches the physical data. In the literature, Bayesian calibration approaches have been extensively used for parameter calibration in several applications [27]. Typically, with a limited number of computer simulations and/or physical trials, Bayesian approaches use Gaussian Processes (GP) to quantify uncertainties. However, as the data size grows, the computational time of the Bayesian approach increases rapidly, and thus, the Bayesian calibration gets inefficient, if not infeasible, for a big data setting [28].

The geographic location and the terrain effects play an important role in the development of the wake, implying that the WDC can be unique to each wind farm. Kinematic wake models with accurate parameters can be efficiently used to design cooperative controls, such as yaw control, among turbines in a utility-scale wind farm [25]. This study thus aims to demonstrate a methodology that can be used to calibrate the wind farm-specific parameters in the engineering wake model using a large size of field data collected from an operational wind farm. Due to the lack of a methodology that can handle large-scale data, most studies in wake effects analysis are limited to using small-scale datasets [29, 30]. However, the resulting calibration would be inaccurate when the small-scale datasets do not statistically represent the physical process. Therefore, our goal is to use field data that covers a wide spectrum of operational conditions.

These big data settings have encouraged using stochastic optimization-based algorithms for parameter calibration [28, 31]. Among several optimization methods, we use a derivative-free Trust-Region (TR) based algorithm due to the versatility and stable performance of its class in the presence of stochastic noise. The TR based stochastic optimization sequentially builds local response surfaces guided by the optimization procedure. This procedure allows us to obtain the estimated outputs from the engineering wake model with more refined wake parameters throughout the iterative procedure. As such, more informative local surrogate models are built using the most updated information based on the parameter search trajectory. Specifically, our approach builds surrogates for the loss function that measures the difference between the engineering model output and physical observation. Doing so effectively guides computer experiments to identify the best subsets of data for achieving computational efficiency and estimation robustness. While being effective, the original TR algorithm faces noisy estimation when a subset of data is used at each iteration, which can slow down the calibration process. To address the limitation, we integrate adaptive and stratified sampling strategies into the TR based algorithm to reduce the computational cost of getting sufficiently accurate estimates.

To evaluate the proposed stochastic optimization-based calibration approach, we first implement the algorithm with data from an offshore and a land-based wind farm using two wake models: the original Jensen's model, and the extended model that formulates the wake decay parameter to be linearly dependent on the turbulence intensity (TI). We compare our approach's prediction accuracy with the suggested values in the literature to validate our calibration procedure. Our approach shows superior computational efficiency and robustness over an alternative optimization approach. We then extend our approach to calibrate parameters of an analytical wake model [25] with a Gaussian profile to showcase that the proposed approach can be easily applied to other advanced engineering wake models.

Our contribution in this paper is two-fold: (i) we present a new data-driven stochastic optimization-based approach using operational data to calibrate parameters in wake models; (ii) we further improve the algorithm in terms of both accuracy and efficiency by reconciling the statistical variance reduction techniques into the optimization framework. For further clarification, our objective is not to determine the wake decay parameter in the Jensen wake model for all wind farms. As mentioned earlier, each wind farm has its unique characteristics and with the help of operational data from each wind farm, this research aims to provide a new method for determining the proper values of wake parameters in kinematic wake models that are specific to that wind farm.

In the remainder of the paper, Section 2 discusses the engineering wake model in more detail. Section 3 presents the stochastic optimization framework to solve the wake calibration problem. Section 4 describes the proposed algorithms that search for the WDC efficiently and robustly. Implementation and results are presented in Section 5. Section 6 concludes the paper.

2. Engineering Wake Model

The Jensen model, an analytical engineering model, is one of the very first wake models. It is based on the law of conservation of mass with some simple assumptions like constant wind speed within the wake cross section and linear propagation of wake in the downstream direction. It was initially proposed to model the wake of a single turbine and later modified by Katic et al. [17] to incorporate a multi-turbine setting. Below we review the Jensen wake model and its extensions.

The wake profile in Jensen model is assumed to have a top-hat shape, shown in Figure 1. The diameter

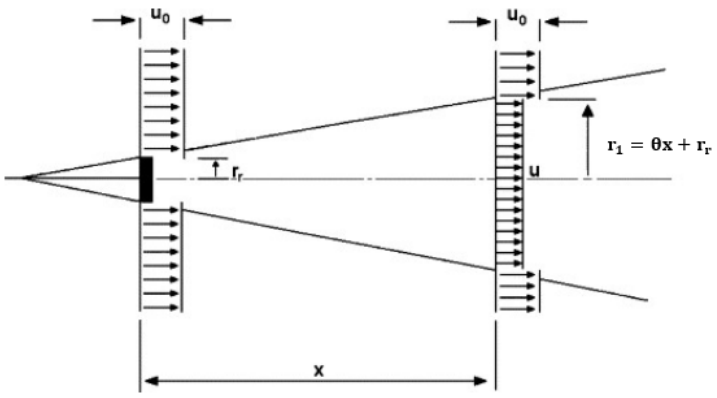


Figure 1: Top-hat structure of the Jensen wake model, where r_r is the rotor radius and u_0 is the free-stream wind speed (excerpted from [16]).

D_w of the wake at a downstream distance x can be estimated as

$$D_w = D + 2\theta x, \tag{1}$$

where D is the diameter of the turbine and θ represents the WDC. The wind speed u within the wake is less than the free-stream wind speed and can be evaluated as

$$u = u_0\Big[1 - \frac{1 - \sqrt{1 - C_t}}{D_w/D}\Big], \tag{2}$$

with u_0 being the free stream wind speed and C_t the thrust coefficient of the turbine, a design parameter specified by the turbine manufacturer.

The simplicity of this model makes it extremely viable to use but it's accuracy largely depends on how well we can estimate the WDC. Its widely accepted values in both industry and literature are 0.075 for land-based wind farms, and 0.04 for offshore wind farms [17, 32, 33]. However, several studies suggest that these recommended values do not accurately represent the wind speed deficits (or power deficits) in downstream turbines [11, 12, 30]. It may thus be beneficial to calibrate a wind farm-specific WDC using the operational data from the wind farm.

The original Jensen model and its extension to multi-turbine wind farms do not take into account the local wind conditions of wind farms, because they use constant values for the WDC. Later, based on boundary layer theory, Frandsen [34] suggested that the WDC can be expressed in terms of the turbine hub height and the surface roughness of the wind farm. It was also suggested that the wake decay can be influenced by turbulence. As the hub height of a turbine increases, the effect of surface roughness on the wake becomes less prominent. In particular, the hub height for offshore turbines is typically taller than those of land-based turbines (e.g., 100 meters), and it is projected to grow to about 150 meters [35]. Peña et al. [24] claim that the intensity of wake decay is predominantly based on the atmospheric stability at higher hub heights. A stable atmosphere resits the motion of air in the vertical direction [36]. The atmospheric stability is known to directly affect the efficiency of wind farm. In general a wind farm is expected to be less efficient under stable atmospheric conditions [37].

In literature, wake decay and atmospheric stability are often linked via TI. Barthelmie et al. [22] have shown that TI impacts normalized power output of an offshore wind farm. TI is the ratio of the standard deviation of the wind speed to the average wind speed during a certain time interval, say, 10 minutes. Higher TI implies more mixing which causes faster dissipation of the wake in the downstream direction and thus, less prominent the effect of wake decay. Generally, unstable atmospheric conditions have higher TI and stable conditions have lower TI. However, under neutral conditions a wide range of TI values can be observed [22, 38].

In attempts to relate the TI to the WDC θ, several studies suggested a linear equation for the WDC in

terms of TI [23, 38] under certain specific conditions of atmospheric stability as

$$\theta = p + q \times \text{TI}, \tag{3}$$

where p and q are positive constants. Alblas et al. [39] show that it holds $\theta \approx 0.5 \times \text{TI}$ under neutral conditions, whereas under stable and unstable conditions θ is of the order of TI or 1.5 to 1.9 times TI respectively for offshore wind farms. Peña et al. [38] estimate the WDC as $\theta \approx 0.4 \times \text{TI}$ for a flat homogeneous terrain with hub heights ranging from 40 m–60 m for on shore wind farms under stable wind conditions. Further, Niayifar and Porté-Agel [40] used LES data to express the WDC as $\theta = 0.003678 + 0.3837 \times (\text{TI})_{\text{mod}}$ for $0.065 < (\text{TI})_{\text{mod}} < 0.15$, where $(\text{TI})_{\text{mod}}$, the modelled TI, is a combination of measured TI and wake-added TI [13]. Modelling the WDC as a function of TI may improve the accuracy of the Jensen model, but existing studies suggest different relationship between the WDC and TI. With such inconsistency and uncertainties, it becomes necessary to calibrate the coefficient (and the relationship between θ and TI) more accurately. This calls for a new approach to solve the calibration problem using data-driven stochastic optimization.

We further consider another state-of-the-art flow control model based on a blade element theory discussed in [25]. This flow control model assumes a Gaussian shape of the stream-wise wake based on the lifting line model, and the same Gaussian shape is also used to model the lateral wake [41]. We refer to this model as the Gaussian wake model hereafter. This Gaussian wake model has two parameters: the wake spreading coefficient k_w and the proportionality constant σ_0. The spreading coefficient k_w is used to estimate the downstream effective wake diameter, and the downstream width of the Gaussian profile is determined by multiplying the Gaussian proportionality constant σ_0 by the effective wake diameter. These parameters depend on the site-specific wind conditions and the wind farm layout and influence the prediction accuracy of the wake model [25]. We present a data-driven stochastic optimization approach using operational wind farm data to calibrate these parameters.

3. Stochastic Optimization for Wake Calibration (SOWC)

We formulate the calibration problem using data-driven stochastic optimization in Section 3.1 and present a new approach to efficiently solve the problem in Section 3.2.

3.1. Problem Formulation

Consider an engineering wake model (e.g., the Jensen wake model) which generates a response vector $y^c(\theta; x)$ for a given weather condition x (e.g., wind speed) and a set of parameters θ. For instance, θ is one-dimensional in (2) or two-dimensional, i.e., $[p, q]^\intercal$ in (3). The accuracy of the wake model is then evaluated by comparing its generated response $y^c(\theta; x) \in \mathbb{R}^b$ with the observed response at the same input x. Consider a dataset containing the field observations $\mathcal{D} = \{\langle x_j, y_j \rangle\}_{j=1}^n$, $x_j \in \mathbb{R}^a$ being the operational input condition and $y_j \in \mathbb{R}^b$ being the output response vector at multiple turbines in a wind farm. In particular, y_j is a vector containing the observed wind power at each of the b downstream turbines, whereas $y^c(\theta; x_j)$ results from the output of the wake model that simulates the power at each of those b downstream turbines with the same input vector x_j. In this study, we consider steady state conditions. Hence, $\langle x_j, y_j \rangle$ represents the average measurements during a short time interval, e.g., 10 minutes.

We seek a calibration parameter that minimizes the difference between the wake model output and observed data. We refer to this difference as loss, and denote it by $\ell(y^c(\theta; x), y) : \mathbb{R}^b \to \mathbb{R}$. Since both $y^c(\theta; x)$ and y are vector-valued, we consider the loss to be the L_2 norm of their difference, i.e., the sum of their squared residuals. Then we can formulate the calibration problem as

$$\begin{aligned} \min_{\theta} \quad & f(\theta) := \mathbb{E}_{X,Y}[\ell(y^c(\theta; X), Y)] \\ \text{subject to} \quad & \theta_{min} \le \theta \le \theta_{max}, \end{aligned} \tag{4}$$

where θ_{min} and θ_{max} are the lower and upper bounds for θ. Here, we assume that the function $f(\theta)$ is bounded below and its gradient Lipschitz continuous, that is, there exists $L < \infty$ such that $\|\nabla_\theta f(\theta_1) - \nabla_\theta f(\theta_2)\| \le L\|\theta_1 - \theta_2\|$. The input X and output Y follow an unknown joint distribution function. Further, the wake

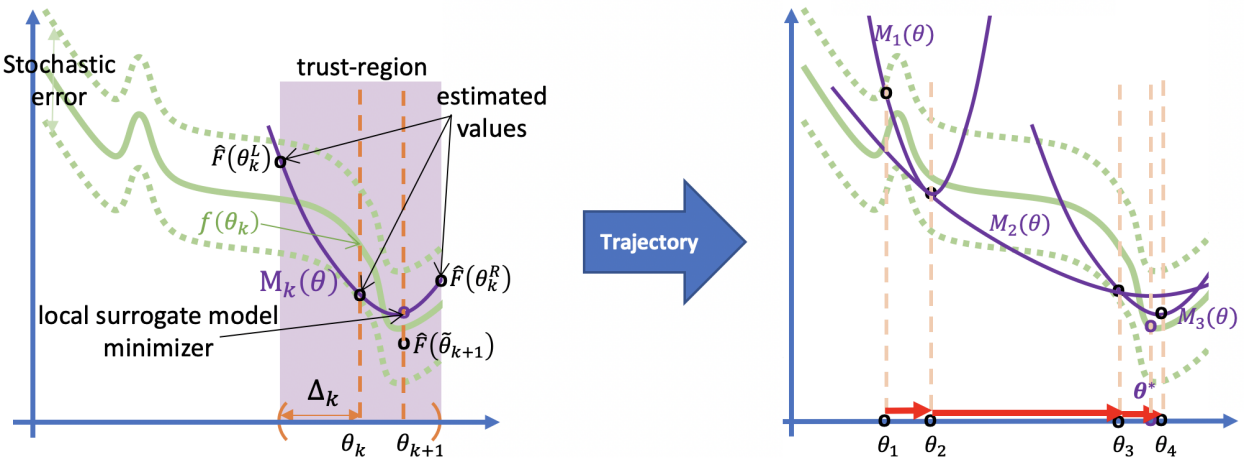


Figure 2: Overview of TR operation: At each iteration, the local model in purple curve approximates the true loss value in green curve around each θ_k. In a derivative-free setting, the local model is fitted on close-by solutions, e.g., θ_k^L, θ_k^R. TR optimization uses the minimizer of the local model $M_k(\theta)$ inside a neighborhood of size Δ_k to make progress in the search space. The figure on the right illustrates the trajectory of local models in purple and parameter adjusted in red.

model output $y^c(\theta; x)$ does not take a closed-form expression. Therefore, we solve this problem based on the principle of empirical risk minimization with the objective function of

$$F(\theta) := \frac{1}{|\mathcal{D}|} \sum_{\langle x_j, y_j \rangle \in \mathcal{D}} \ell(y^c(\theta; x_j), y_j)$$

with $|\mathcal{D}|$ being the total sample size of the operational dataset $\mathcal{D}$.

When the data size is large, using the entirety of the available dataset for risk minimization may not be efficient, because it needs to run the wake model for every data point. It may also result in overfitting which will give inaccurate parameter values. To address it, we employ a stochastic optimization approach that uses random subsets of the data set. Suppose n ($n \ll |\mathcal{D}|$) data points are randomly chosen from the entire dataset $\mathcal{D}$. Then, we estimate the expected value in (4) with a sample average approximation, i.e.,

$$\hat{F}(\theta) := \frac{1}{|\mathcal{S}(\theta)|} \sum_{\langle x_j, y_j \rangle \in \mathcal{S}(\theta)} \ell(y^c(\theta; x_j), y_j) \tag{5}$$

with $|\mathcal{S}(\theta)|$ being the sample size of the subset sampled from the data, $\mathcal{S}(\theta) \subseteq \mathcal{D}$. It is assumed, in the presence of a large $\mathcal{D}$, that $\hat{F}(\theta)$ converges to $f(\theta)$ almost surely, as the size of $\mathcal{S}(\theta)$ grows large. Despite a common strategy for sampling, i.e., a fixed subset of data to use throughout the optimization, we show subsets as functions of the parameter value gain more efficiency; we will discuss this further in the next section.

3.2. Adaptive Trust-Region based Optimization

To solve the above optimization problem, we use a TR based method [42]. The TR method iteratively approximates the true objective function $f(\theta)$ by creating an easy-to-handle local model in a small neighborhood using randomly chosen subsets of data. Optimizing this local model within that neighborhood provides a new candidate solution to the true objective function if accepted, as an improved solution updates the parameter value towards the optimality.

Specifically, suppose that θ_k is the current recommended parameter value and $\mathcal{S}(\theta_k)$ is the subset of data drawn from the entire dataset $\mathcal{D}$ at the k^{th} iteration. Let $M_k(\theta)$ denote the local surrogate model at the k^{th} iteration that approximates the true loss function $f(\theta)$ around θ_k in the region of size Δ_k. This region, denoted by $\mathcal{B}_k$, is often a closed ball around the incumbent solution θ_k, i.e., $\mathcal{B}_k = \{\theta : \|\theta - \theta_k\|_2 \leq \Delta_k\}$. Then we find the candidate of the next incumbent θ_{k+1} that minimizes $M_k(\cdot)$ in $\mathcal{B}_k$. Thus, the TR radius, Δ_k, limits the size of the next step.

Note that we cannot compute the gradient of the objective function to get the next incumbent, because the functional form of $\ell(\cdot)$ (or $y^c(\cdot)$) is unknown. That is, direct observations of gradients, $\nabla_\theta \ell(y^c(\theta; x_j), y_j)$, are unavailable. This is why we employ the derivative-free stochastic optimization by building a local model $M_k(\theta)$ (typically a quadratic function) using estimated objective functions at multiple θs within $\mathcal{B}_k$. Let $\Theta_k = \{\theta_k, \theta_k^{(j)} \in \mathcal{B}_k,\ j = 1, 2, \cdots, m\}$ be a set including θ_k and several other solutions around it. With interpolation, we fit a $M_k(\cdot)$ whose gradient and higher order derivatives will capture the behavior of the objective function around θ_k. This set needs to be poised such that the matrix

$$P(\Phi, \Theta_k) = \begin{bmatrix} \phi^1(\theta_k^{(1)}) & \phi^2(\theta_k^{(1)}) & \cdots & \phi^m(\theta_k^{(1)}) \\ \phi^1(\theta_k^{(2)}) & \phi^2(\theta_k^{(2)}) & \cdots & \phi^m(\theta_k^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi^1(\theta_k^{(m)}) & \phi^2(\theta_k^{(m)}) & \cdots & \phi^m(\theta_k^{(m)}) \end{bmatrix},$$

is nonsingular with a polynomial basis $\Phi(z) = (\phi^1(z), \phi^2(z), \cdots, \phi^m(z))$ on $\mathbb{R}^d$. The local model $M_k(\theta) = \sum_{j=1}^m \hat{\beta}_j \phi^j(\theta)$ is constructed as a Taylor-like approximation where $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \hat{\beta}_2, \cdots, \hat{\beta}_m)$ is obtained by solving the system of linear equations

$$P(\Phi, \Theta_k)\hat{\boldsymbol{\beta}} = (\hat{F}(\theta_k^{(1)}), \hat{F}(\theta_k^{(2)}), \cdots, \hat{F}(\theta_k^{(m)}))^\intercal. \tag{6}$$

With a quadratic basis the model is $M_k(\theta) = \hat{\beta}_1 + (\theta - \theta_k)^\intercal(\hat{\beta}_2, \cdots, \hat{\beta}_{d+1}) + \frac{1}{2}(\theta - \theta_k)^\intercal H_k(\theta - \theta_k)$, where H_k is an approximated Hessian using $\hat{\beta}_{d+2}, \cdots, \hat{\beta}_m$. The number of points needed to build the model m depends on the dimension of the problem d and the order of polynomial interpolation. For a quadratic $M_k(\cdot)$ with a complete Hessian $m = d(d+3)/2$ [43], but when $d > 1$, to reduce m we can build a quadratic $M_k(\cdot)$ with a diagonal Hessian, which will only need $m = 2d$ points [44]. For a one-dimensional problem, generating a quadratic model needs $m = 2$ points, i.e., $\Theta_k = \{\theta_k, \theta_k^L, \theta_k^R\}$ as shown in Figure 2, where $\theta_k^L = \theta_k - \Delta_k$ and $\theta_k^R = \theta_k + \Delta_k$. The performance of the TR optimization depends on the quality of $M_k(\theta)$, which depends on the existence of $\hat{\boldsymbol{\beta}}$ and how poised the set Θ_k is. A good quality model will ensure that $\|\nabla_\theta M_k(\theta) - \nabla f(\theta)\| \leq \mathcal{O}(\Delta_k^2)$ for all $\theta \in \mathcal{B}_k$. Specifically, for the derivative-free TR, additional quality steps to ensure a lock-step between $\nabla_\theta M_k(\theta_k)$ and Δ_k is needed: if Δ_k is too large as compared to the gradient of the model at θ_k, we reduce the TR radius [45]. This critical step can be relaxed for those iterations that are not near optimality for faster performance.

The left panel of Figure 2 illustrates a single iteration where the next candidate solution $\tilde{\theta}_{k+1}$ is selected by minimizing the local model $M_k(\theta)$ within Δ_k distance from the incumbent solution θ_k. The candidate point is then accepted if the loss estimate also sufficiently reduces, in comparison to the reduction in the model value, by moving to the candidates solution from θ_k. This test is done through computing what is called the success ratio, $\hat{\rho}_k = \frac{\text{reduction in the loss estimates}}{\text{reduction in the model}}$ and checking whether it is at least as large as the success threshold η_1, which is a user-specified parameter of the TR algorithms. If successful, as a vote of confidence, the next iteration starts with $\Delta_{k+1} > \Delta_k$. Otherwise, we let $\theta_{k+1} = \theta_k$ and reduce the TR radius ($\Delta_{k+1} < \Delta_k$) to look more closely in that region. The convergence of this algorithm to a (local) optimum is guaranteed when $\Delta_k \to 0$ as $k \to \infty$ almost surely. The TR radius converges to zero because as the algorithm approaches the optimal solution, it keeps reducing the radius in pursuit of a better solution which it cannot find.

Since the local surrogate model $M_k(\theta)$ is built on the estimates of the loss function using a subset of the data, large stochastic errors can substantially impact the algorithm's performance. One approach to address this sensitivity to noise is to reduce the stochastic error by increasing the sample size only when distinguishing better solutions requires higher precision. The fundamental idea in this adaptive sampling strategy is that in the stochastic setting we use smaller subsets of data when the current incumbent is far away from the optimal solution and use larger subsets as we near optimality. This adaptive sampling allows us to balance the computational effort over iterations. Shashaani et al. [43] devised an almost surely convergent stochastic optimization algorithm that leverages adaptive sampling within a TR framework by using higher orders of the TR radius as a measure of optimality gap. Their algorithm is referred to as Adaptive Sampling TR Optimization for Derivative-Free Stochastic Oracles (ASTRODF).

Specifically, the sample size $\mathcal{S}(\theta_k)$ adapts the standard error of the estimator as small as a threshold that

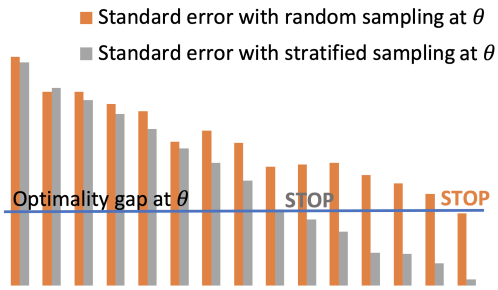


Figure 3: Adaptive sampling stops adding samples once the standard error drops below an approximate optimality gap. The stopping occurs faster with the stratified sampling (orange) than without (gray) because it further reduces the standard error with the same number of samples.

equals $\kappa\Delta_k^2/\sqrt{\lambda_k}$, that is,

$$|\mathcal{S}(\theta_k)| = \min\left\{n \geq \lambda_k : \sqrt{\frac{\hat{\sigma}_F^2(\theta_k)}{n}} \leq \frac{\kappa}{\sqrt{\lambda_k}}\Delta_k^2\right\}, \tag{7}$$

where $\hat{\sigma}_F^2(\theta_k)$ is the variance estimate of the loss value $\ell(y^c(\theta_k; x_j), y_j)$ at each individual sample in (5), assuming each data point is independently and identically sampled from the dataset $\mathcal{D}$. Hence the left-hand side (LHS) in (7) is the standard error of the loss estimate $\hat{F}(\theta_k)$. In the right-hand side (RHS), κ is a positive constant, and $\lambda_k = \mathcal{O}(\log k)$ is a slowly increasing deterministic sequence that serves two purposes: (i) it deflates the optimality threshold and enforces larger samples in the later iterations; (ii) it ensures that sample size grows large even if by chance the standard error at later iterations is small. With the condition in (7), the adaptive sampling strategy ensures that the sample size is initially small but increases as the TR radius becomes small indicating that the incumbent solution is hard to beat. Note that the sample size at each iteration becomes stochastic and is calculated using the optimality error and the TR radius Δ_k during that iteration. Figure 3 illustrates the idea of adaptive sampling, where the horizontal blue line represents the threshold of standard error, i.e., RHS of the inequality in (7), at a fixed solution. The height of the vertical bars represents the standard error of the loss function estimate at that solution, and the left to right direction shows this error decreases as the sample size increases. We stop adding more samples as soon as the estimated vertical bars drop below the threshold, indicating that the estimation error, is commensurate with the optimality gap.

In the new improvement to this algorithm that we describe in Section 4, we integrate variance reduction to ASTRODF to shorten the bars (see the gray bars in Figure 3) and stop sampling earlier without losing any convergence guarantees. In this new framework, we not only set the size of the samples adaptively, but we also decide which points to be added to the samples.

4. Stratified Adaptive Sampling and Trust-Region Optimization

For efficiency improvement and robustness, we integrate stratified sampling within ASTRODF. We first describe the stratified sampling in Section 4.1 and propose the new algorithm in Section 4.2

4.1. Stratified Sampling

Stratified sampling is a variance reduction technique that divides the input domain $\mathcal{D}$ into multiple strata due to the erratic behavior and impact of the data from each stratum on the objective function estimate. Instead of assuming that all the data follows a single distribution, we allow each stratum to have a unique data distribution. This design directly uses the heterogeneity in the input data to our benefit. The central idea is that we allocate portions of the budget (overall samples) to each stratum based on its contribution to the variance of the objective function estimator. By allocating more computational budget to the more important strata, we reduce the estimation variance that leads to the acceleration of our calibration procedure.

We divide the available data $\mathcal{D}$ into I disjoint strata such that $\mathcal{D} = \cup_i^I \mathcal{D}_i$ and $\mathcal{D}_i \cap_{i \neq i'} \mathcal{D}_{i'} = \emptyset$; we will discuss how to divide $\mathcal{D}$ to form I strata in Section 4.2. Let $p_i = |\mathcal{D}_i|/|\mathcal{D}|$ be the ratio of total points in stratum i and $\hat{\sigma}_{F,i}(\theta)$ be the estimated output variability at θ in stratum i for $i = 1, 2, \cdots, I$. If the estimated objective in stratum i is $\hat{F}_i(\theta)$, i.e., the sample average of the loss value at stratum i, with $\mathcal{S}_i(\theta) \subseteq \mathcal{D}_i$ being the sample set drawn from stratum i and $n_i(\theta) = |\mathcal{S}_i(\theta)|$, then the overall estimated mean is

$$\hat{F}(\theta) = \sum_{i=1}^{I} p_i \hat{F}_i(\theta) \tag{8}$$

and the estimated sample variance is

$$\widehat{\mathrm{Var}}(\hat{F}(\theta)) = \sum_{i=1}^{I} p_i^2 \widehat{\mathrm{Var}}(\hat{F}_i(\theta)) = \sum_{i=1}^{I} \frac{p_i^2 \hat{\sigma}_{F,i}^2(\theta)}{n_i(\theta)}, \tag{9}$$

with $\hat{\sigma}_{F,i}^2(\theta) = (n_i(\theta)-1)^{-1} \|\ell(y^c(\theta; \boldsymbol{x}_i), \boldsymbol{y}_i) - \hat{F}_i(\theta)\mathbf{1}_{n_i(\theta)}\|_2^2$, where $\ell(y^c(\theta; \boldsymbol{x}_i), \boldsymbol{y}_i)$ is a vector of loss functions for stratum i, $\{\langle x_j, y_j \rangle \in \mathcal{S}_i(\theta)\}$, and $\mathbf{1}_{n_i(\theta)}$ is an $n_i(\theta)$-dimensional vector of ones. The sample variance in (9) is proven [46] to be smaller than the sample variance without stratification and the difference between the two is maximized when $n_i(\theta)$, the sample size for a given θ from $\mathcal{D}_i$, is proportional to p_i and $\hat{\sigma}_{F,i}(\theta)$, i.e.,

$$w_i(\theta) = \frac{n_i(\theta)}{\sum_{j=1}^{I} n_j(\theta)} = \frac{p_i \hat{\sigma}_{F,i}(\theta)}{\sum_{j=1}^{I} p_j \hat{\sigma}_{F,j}(\theta)}. \tag{10}$$

There are several questions in designing an integrated framework of TR optimization with adaptive and stratified sampling: (i) How to stratify the input space and how much does that cost?; (ii) For each stratum, how to adaptively allocate the budget and when to stop adding samples?; and (iii) How to reuse observations and data utilized in previous iterations? Next, we describe the new algorithm that addresses all these questions.

4.2. Stratified-ASTRODF with Dynamic Weights (SOWC-1)

Before successful stratified sampling, we need successful stratification (or partitioning) strategies. An effective way for stratification is to divide the data such that within-stratum variance is small and between-stratum variance is large [47]. The ideal stratification should minimize the total variance in (9). However, the variance of the estimated loss $\hat{F}(\theta)$ depends on the incumbent θ, implying that we need to partition the data at every visited solution. But doing so incurs additional computational burden. In a previous study, Liu et al. [28] demonstrate a dynamic stratification integrated with stochastic gradient methods. However, partitioning with a small dataset drawn in each iteration may increase the stochasticity during the search procedure and ultimately reduce the efficiency.

As a remedy, we assume that the physical response variance σ_Y is closely connected to the loss variance $\sigma_F(\theta)$ irrespective of θ and asynchronously partition the input data before the optimization. Although this use of proxy variance does not precisely track the change in the output variance at different WDCs in the search space, it has the advantage of being computed only once and reused at every iteration. To divide the input space, we borrow the classification and regression tree (CART) idea and form the I disjoint strata with I being a user-specified parameter of the algorithm. CART groups the data with similar characteristics together by minimizing the overall sum of squared errors [48]. It helps achieve the minimum total variance of the response outputs. Figure 4 shows the application of CART for stratification for a single experiment. In this work we have used TI to divide the data into multiple strata as TI is closely related to the variance.

The next challenge is how to initiate the budget allocation for each stratum to estimate their variance before invoking the adaptive sampling. Given the trajectory of the search and all the data used until right before iteration k, this initial sample size for stratum i, denoted by $\lambda_{k,i}$, would remain unchanged through sampling in iteration k. First, we use the most recent budget allocation $w_i(\theta_k)$ for stratum i following (10). Then the initial sample size choice for stratum i will be

$$\lambda_{k,i} = \lceil n^0 + w_i(\theta_k) \times \left(\max\{n_{\mathrm{all}}^0, \lambda_k\} - n^0 I\right) - 0.5 \rceil, \tag{11}$$

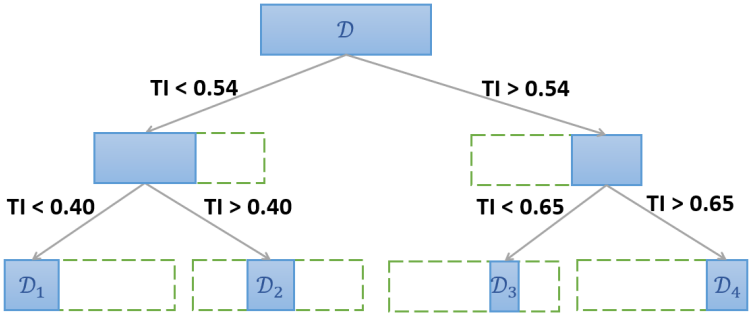


Figure 4: Stratification with CART partitions the data based on TI.

where n^0 and n^0_{all} are user-specified parameters representing the minimum sample size for each stratum and the minimum overall sample size. We allocate at least n^0 samples at each stratum to avoid too large estimation error at the beginning. For illustration, suppose we have $I = 2$ strata with the most recent weights $w_1(\theta_k) = 0.3$ and $w_2(\theta_k) = 0.7$, assuming the lower bound at an early iteration is $\lambda_k = 10$, and the minimum size parameters are $n^0 = 2, n^0_{\text{all}} = 40$. This gives $\lambda_{k,1} = 13$ and $\lambda_{k,2} = 27$ as the initial sample size for each of the two strata. If we were at a much later iteration with, say, $\lambda_k = 50$, these numbers would change to $\lambda_{k,1} = 16$ and $\lambda_{k,2} = 34$.

Using $\lambda_{k,i}$'s, we adaptively determine the initial sample size of each stratum at the kth iterate θ_k. Recall that the sample size is decided based on the inequality in (7) [43], where the LHS is the estimation error and the RHS is the threshold (indicating the optimality gap). With stratified sampling, the variance estimate is given by $\widehat{\text{Var}}(\hat{F}(\theta))$ in (9). For the RHS, we employ the same threshold in (7) with some modifications. Specifically, we replace the squared TR radius with the estimated loss itself. While the local optimality in TR is measured by the square of the model gradient norm that is kept in lock-step with the TR radius, we also know that $\hat{F}(\theta_k) = \mathcal{O}_p(\Delta_k^2)$ because the models constructed need to have sufficient quality, ensured by correct placement of the solutions in $\mathcal{B}_k$. Hence, we decide the sample size of each stratum as

$$(N_i(\theta_k),\ \forall i = 1, 2, \cdots, I) = \min \left\{ n_i \geq \lambda_{k,i},\ \forall i = 1, 2, \cdots, I : \sqrt{\widehat{\text{Var}}(\hat{F}(\theta_k))} \leq \frac{1}{\sqrt{\lambda_k}} \hat{F}(\theta_k) \right\}. \tag{12}$$

In (12), there are two major modifications to the adaptive sampling rule in (7). First, the LHS now uses the function estimates via stratified sampling instead of the TR radius directly. This modification is important, because it avoids algorithm's sloppiness at the beginning due to the inappropriate choice of initial TR size, which is a user-specified parameter. In other words, by this replacement we can reduce the sensitivity of the algorithm to its parameters; note that we remove the constant parameter κ here too. Additionally, this change in the sampling rule ensures that in iteration k, all visited solutions $\theta \in \Theta_k \cup \{\tilde{\theta}_{k+1}\}$ determine samples needed for their estimate's precision using their own $\hat{F}(\theta)$, instead of using a fixed RHS via Δ_k^2 for all θ. Our modified adaptive sampling rule resembles progressively controlling the loss' coefficient of variation, which is defined as $\text{CV}(Z) = \frac{\sigma(Z)}{\mathbb{E}[Z]}$ for a random variable Z. Lastly, we collectively consider all strata when adding new samples, rather than doing an adaptive sampling with each stratum separately. This integration of stratified sampling and adaptive sampling helps obtain maximum efficiency.

As listed in Algorithm 2, at every new WDC value for which we need to evaluate the loss during iteration k, we first use the information obtained for θ_k, the recommended solution from iteration $k-1$. With (11) we determine the initial sample size for each stratum through $w_i(\theta_k)$. Once new samples drawn for θ_k, we use those most updated $w_i(\theta_k)$ values for the following solutions to be visited, e.g., the poised set for model construction and the candidate incumbent for the next iteration. Why is the choice of initial sample size for each stratum crucial? If our initial sample size is too small, our weights can be inadequate, and eventually, we will spend a lot of time getting to a good point in the search trajectory. On the other hand, large initial sample sizes can guarantee a good estimation of weights at the expense of utilizing a lot of our computational budget. After initialization of samples in each stratum, if the stopping time condition (12) is not satisfied, we add a single point, update the standard error from (9) and recheck the stopping time condition. We use

what we call the selective randomized method, inspired by [49, 50], to determine the stratum to which this point should be added. Based on this strategy, samples are added one by one to randomly selected strata according to the probability mass function (pmf) described above until the stopping criteria is met. At stopping, ideally the ratio of the number of samples from stratum i to the total number of samples should be equal to the weight of that stratum, i.e., $w_i(\theta_k) = q_i(\theta_k)$ where $q_i(\theta_k) := N_i(\theta_k)/\sum_{j=1}^{I} N_j(\theta_k)$. However, since in the adaptive setting, the new samples are added randomly, it is possible that for some strata, $q_i(\theta_k)$ is greater or smaller than the corresponding weight. To use the optimal allocation information in the adaptive sampling, we estimate the probability of adding the point to stratum i as

$$\pi_i^{\mathrm{d}} = \Pr\{\text{selecting stratum } i\} = \frac{w_i(\theta_k)\mathbb{I}\{w_i(\theta_k) > q_i(\theta_k)\}}{\sum_{j=1}^{I} w_j(\theta_k)\mathbb{I}\{w_j(\theta_k) > q_j(\theta_k)\}}. \tag{13}$$

Using $(\pi_1^{\mathrm{d}}, \pi_2^{\mathrm{d}}, \cdots, \pi_I^{\mathrm{d}})$ obtained from dynamic weights in (13) as the pmf for stratum selection guarantees that those strata whose allocation is lower than optimal receive more likelihood for selection. The adaptive sampling in the derivative-free setting has to be conducted not just for the incumbent solutions, but also for those solutions that will be used for the local model construction and those suggested as the candidate incumbents by minimizing the local model. Recall that for large sample sizes, the standard error with random sampling will be more than the standard error with stratified sampling. Thus with the new stratified adaptive sampling strategy, we stop at a smaller sample size; see the grey bars in Figure 3, as compared to the orange bars. We refer to the proposed stratified variant of ASTRODF with dynamic weights as SOWC-1.

4.3. Stratified ASTRODF with Fixed Weights (SOWC-2)

In SOWC-1, the role of the dynamic weights computed by (10) as a budget allocation rate is in choice of initial sample size per stratum (11) and adaptive addition of samples through (13). We now consider a variant of the proposed approach that does not allow budget allocation ratio for each stratum to vary across iterations. We adopt this fixed budget allocation strategy as an alternative because the dynamic scheme causes a complexity, as every new sample changes $\hat{\sigma}_{F,i}(\theta)$ for all θ visited at iteration k, and hence changes the weights themselves. In short, this dynamic change in the weights can incur additional variability in the algorithm performance. For the sake of more stability, we consider a static weight for each stratum to lower the variability throughout the search, as suggested in our earlier work [51], i.e.,

$$v_i = \frac{p_i \hat{\sigma}_{Y,i}}{\sum_{j=1}^{I} p_j \hat{\sigma}_{Y,j}}. \tag{14}$$

Similar to the logic we use for the stratification, these fixed weights are based on the assumption that the distribution of our estimated loss function will closely mimic the distribution of the observed outputs. One advantage of this static weight scheme is the ease of implementation, since the weights are same for each iterate, however, it can result in skewed estimates as it does not take into account the sampled distribution. In SOWC-2, we also use the static weights in selecting a stratum for adaptive addition of new samples, akin to (13) but with v_i instead of $w_i(\theta_k)$; we call these fixed probabilities $(\pi_1^{\mathrm{f}}, \pi_2^{\mathrm{f}}, \cdots, \pi_I^{\mathrm{f}})$.

Table 1 shows the difference between the computational budget allocation weights $w_i(\theta_k)$ and v_i in SOWC-1 and SOWC-2, respectively, in one of our experiments. In SOWC-1, more data points from the high TI condition (the last stratum) are used in the calibration, whereas SOWC-2 assigns similar efforts across multiple strata. We note that despite using fixed weights in SOWC-2, the TI interval in each stratum is different, rendering the stratified sampling philosophy: sample less from less important weather conditions and more from important conditions for the calibration purpose.

Algorithm 1 summarizes the procedure of SOWC-1 and SOWC-2, where we combine stratified sampling with the updated adaptive sampling criteria (12) that is detailed in Algorithm 2, to use fixed/dynamic weights and randomized allocation.

5. Implementation Results

We use the multi-turbine extension of the Jensen wake model which takes into account multiple wakes and partial shadows [17] to calibrate the WDC in (1)-(2). We use the L_2 loss, i.e., $\ell(y^c(\theta; x), y) = \|y^c(\theta; x) - y\|_2^2$

Table 1: Stratification details for the two suggested methods in one experiment[4].

Bins	TI range	Probability mass (p_i)	SOWC-1		SOWC-2	
			$\hat{\sigma}_{F,i}(\theta)$	$w_i(\theta)$	$\hat{\sigma}_{Y,i}$	v_i
1	0.08–0.40	0.32	0.01	0.27	0.02	0.24
2	0.40–0.54	0.27	0.01	0.23	0.04	0.29
3	0.54–0.65	0.17	0.01	0.14	0.05	0.21
4	0.65–1.48	0.24	0.03	0.36	0.04	0.26

[1] For SOWC-1 $\hat{\sigma}_{F,i}(\theta)$ and $w_i(\theta)$ values are reported at θ from the last iteration.

Algorithm 1 S-ASTRODF for Wake Parameter Calibration

Input: Available dataset $\mathcal{D}$, initial solution θ_0 and TR radius Δ_0, number of strata I, maximum budget T, minimum sample size for each stratum $n^0 \geq 2$, total minimum sample size $n^0_{\text{all}} \geq n^0 I$, and success threshold $\eta_1 > 0$.
initialization: Set calls $= 0$ and iteration $k = 0$. Determine the strata by CART and compute their properties $\{(\mathcal{D}_i, p_i, \hat{\sigma}_{Y,i}, v_i),\ \forall i = 1, 2, \cdots, I\}$ including fixed weights v_i following (14).
while calls $< T$ **do**
    Generate Θ_k, a poised set within $\mathcal{B}_k$.
    Estimate $\hat{F}(\theta)$ using Algorithm 2 for all $\theta \in \Theta_k$.
    Set calls $=$ calls $+ \sum_{\theta\in\Theta_k} \sum_{i=1}^{I} N_i(\theta)$.
    Generate a surrogate model $M_k(\cdot)$ by interpolation using $(\Theta_k, \hat{F}(\Theta_k))$ following (6).
    If the model gradient $\nabla M_k(\theta_k)$ is small relative to Δ_k, shrink the TR and go to step 4.
    Minimize $M_k(\cdot)$ within $\mathcal{B}_k$ to obtain a candidate solution $\tilde{\theta}_{k+1}$.
    Estimate $\hat{F}(\tilde{\theta}_{k+1})$ using Algorithm 2.
    Set calls $=$ calls $+ \sum_{i=1}^{I} N_i(\tilde{\theta}_{k+1})$.
    Compute the success ratio $\hat{\rho}_k = \frac{\hat{F}(\theta_k) - \hat{F}(\tilde{\theta}_{k+1})}{M_k(\theta_k) - M_k(\tilde{\theta}_{k+1})}$.
    if $\hat{\rho}_k > \eta_1$ **then**
        Set $\theta_{k+1} = \tilde{\theta}_{k+1}$ and $\Delta_{k+1} > \Delta_k$.
    else
        Set $\theta_{k+1} = \theta_k$ and $\Delta_{k+1} < \Delta_k$.
    end if
    Set dynamic $w_i(\theta_k)$ at the new incumbent solution following (10).
    Set $k = k + 1$ and go to step 4.
end while
output: Final calibrated wake parameter θ_k and its estimated loss $\hat{F}(\theta_k)$.

in all of our experiments. However, other types of loss functions, e.g., L_1 loss, can easily and flexibly be considered. We first calibrate the WDC as a constant value, and then consider the TI dependent calibration suggested in [22]. To provide reproducible implementation, our code, as well as the dataset used in Section 5.3, can be found in `https://github.com/sshashaa/s-astro-df.git`.

5.1. Data Description

We have data collected from two wind farms: WFa an offshore wind farm, and WFb an onshore wind farm. Table 2 summarizes some details of the two wind farms. Figure 6 shows the modified layout of WFa with some turbines omitted due to confidentiality. The dataset consists of 10-minute average power generated by each wind turbine along with ambient wind conditions collected in the met mast, such as the 10-minute average free-flow wind speed, 10-minute average wind direction, TI, etc. Each wind farm has one meteorological tower (met mast). The turbine spacing between each row in WFa is about 11 times rotor diameter.

The power generated by wind turbines is normalized by dividing each term by the maximum available value. We have used data when the met mast is not under wake, where wind direction ranges from 165° to 315° in WFa and 165° to 195° in WF2. Moreover, TI ranges from 0.1 to 1.5 and wind speed ranges from 3

Algorithm 2 Stratified Adaptive Sampling with Fixed/Dynamic Weights

input: TR radius Δ_k, deflation factor λ_k, details of the strata including dynamic weights for the incumbent solution $\{(\mathcal{D}_i, p_i, \hat{\sigma}_{Y,i}, v_i, w_i(\theta_k)),\ \forall i = 1, 2, \cdots, I\}$, and solution of interest θ.
for $i = 1, 2, \cdots, I$ **do**
    Compute $\lambda_{k,i}$ following (11) that uses $w_i(\theta_k)$ for SOWC-1 or v_i for SOWC-2.
    Draw $N_i(\theta) = \lambda_{k,i}$ random samples from $\mathcal{D}_i$ and form $\mathcal{S}_i(\theta)$ to compute $\hat{\sigma}_{F,i}(\theta)$.
    Update $w_i(\theta)$ for SOWC-1.
end for
Calculate the sample mean $\hat{F}(\theta)$ in (8) and sample variance $\widehat{\mathrm{Var}}(\hat{F}(\theta))$ in (9).
while $\widehat{\mathrm{Var}}(\hat{F}(\theta)) > \frac{\kappa^2}{\lambda_k}\hat{F}(\theta)^2$ **do**
    Randomly select stratum j following the pmf $(\pi_i^{\mathrm{d}})_{i=1}^I$ for SOWC-1 or $(\pi_i^{\mathrm{f}})_{i=1}^I$ for SOWC-2.
    Draw a random data point from $\mathcal{D}_j$ and add to the sample set $\mathcal{S}_j$; increase $N_j(\theta)$ by 1.
    Update $\hat{\sigma}_{F,j}(\theta)$, $\widehat{\mathrm{Var}}(\hat{F}(\theta))$, $\hat{F}(\theta)$, and if SOWC-1, $w_i(\theta)$ for all $i = 1, 2, \cdots, I$.
end while
output: Estimated loss $\hat{F}(\theta)$ and the adaptive sample sizes $N_i(\theta_k)$ for all $i = 1, 2, \cdots, I$.

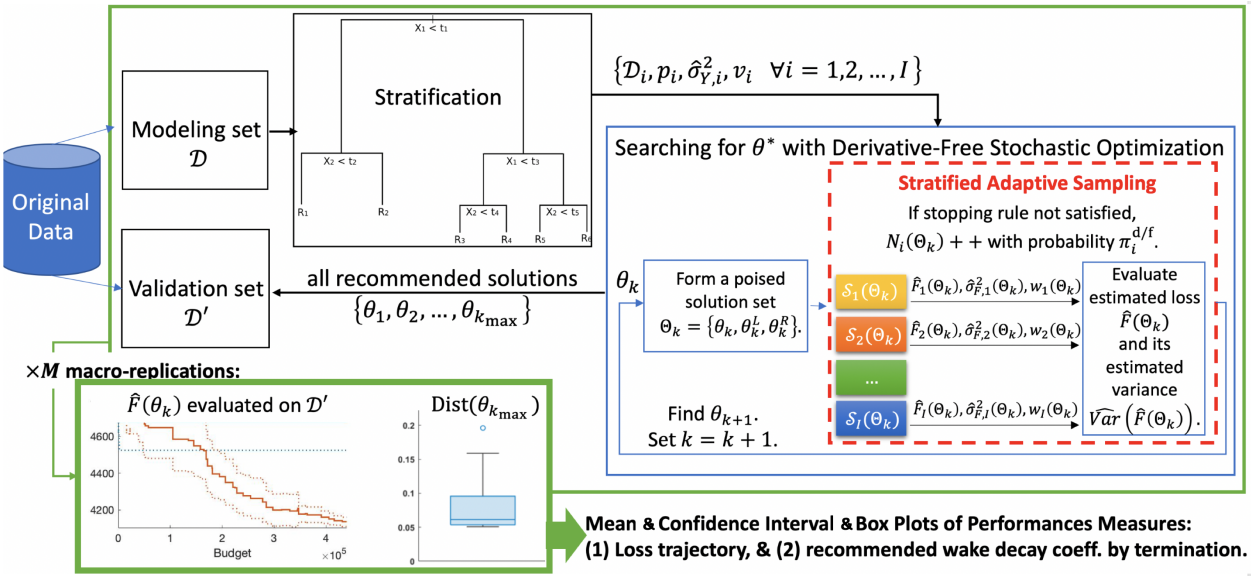


Figure 5: Overall procedure consists of several macro-replications of dividing the data into modeling set, used for training and optimization, and validation set, used for testing and evaluation. The outputs shown in the bottom right box illustrate the loss trajectory and final solution with a fixed computational budget across the macro-replications.

to 15 m/s in our dataset for WFa. For WFb, TI ranges from 0.3 to 2.5 and wind speed ranges from 4 to 13 m/s. Due to the data confidentiality required by the data provider, we omit more details about the two wind farms.

Table 2: Information about the two wind farms.

Wind farm	Type	Data size	Number of turbines	Layout
WFa	offshore	9742	around 30	regular
WFb	land-based	1659	around 40	irregular

The Jensen wake model estimates the incoming wind speed at each turbine. But our dataset does not contain the incoming wind speeds at downstream turbines. To compare the output from the Jensen wake model with actual measurements, we estimate the power curve using data in one of the upstream turbines closely located to the met mast. For the power curve construction, we use B-splines [52]. Once Jensen wake

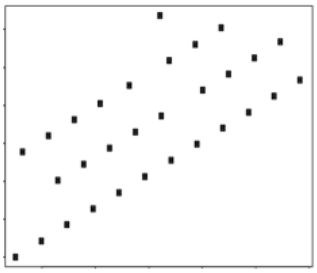


Figure 6: Layout of WFa.

model outputs incoming wind speeds, we use the resulting power curve to estimate the power outputs. For the thrust coefficient in Jensen model, we use the manufacturer's provided values.

An outlook to the full procedure is depicted in Figure 5. To evaluate the performance of the proposed approach, we divide the data into two sets: the modeling set ($\mathcal{D}$) and the validation set ($\mathcal{D}'$). The modeling set comprises of 70% of the data and is used exclusively for the parameter calibration. We use the remaining 30% of data for evaluating the calibration performance. We conduct the experiment 20 times (that is, 20 macro-replications) with different modeling and validation sets. For each experiment, we divide the modeling set into four strata ($I = 4$) using CART. The initial TR radius (Δ_0) is 0.08, and the success threshold (η_1) is 0.10. The minimum sample size for each stratum (n^0) is set at 2 and the total minimum sample size (n^0_{all}) is 40. The deflation factor λ_k is an increasing function of the iteration number k and is set as $\lambda_k = 80(\log k)^{1.5}$. It increases exponentially with iteration number, ensuring that towards the later half of the search we have good estimates for MSE. We have restricted our search space to $\theta \in (0, 1)$.

5.2. Numerical Results

This section reports the implementation results of the proposed methods. We compare our approach with other stochastic optimization methods along with the comparison of the optimally calibrated parameter values against the recommended values. The results are evaluated using MSE which is defined as the mean of squared errors for all the turbines divided by the number of observations. We first summarize point calibration results where the WDC is modeled as a constant value for WFa and WFb, followed by the functional calibration results with the WDC as a linear function of TI for WFa.

5.2.1. Point Calibration

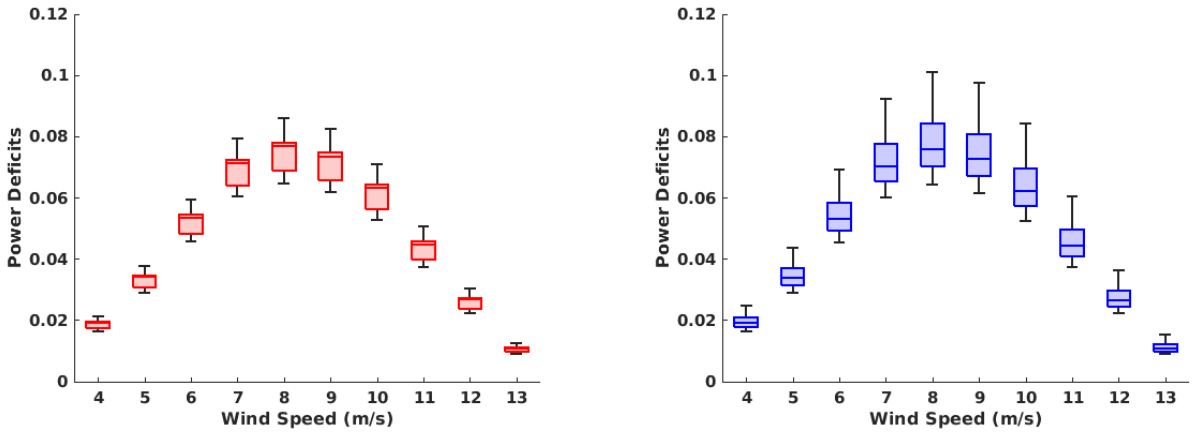
Table 3 first compares the results for WFa from the proposed approach with those when the recommended value of $\theta = 0.040$ for an offshore wind farm is employed. It summarizes the estimated WDC and the normalized MSE evaluated over 20 validation sets. Our proposed methods, SOWC-1 and SOWC-2, suggest the WDC to be around 0.053 which is slightly different from the recommended value. The third column suggests that the proposed approach can reduce the overall MSE by 0.87%, compared to the recommended value. Further analysis shows that our approach achieves 2.38% reduction of MSE for turbines placed in the second row, while the MSEs are comparable for turbines in the third and fourth rows. Overall, slightly higher MSEs at the reference value of 0.04 indicate that using this default value may not be optimal, highlighting the need for wind farm-specific WDC calibration.

Table 3: Comparison of MSE and estimated WDC for different methods for WFa. The second column summarizes the estimated WDC, the third column summarizes the overall MSE, and columns four to size summarize the MSE for the turbines in the second, third, and fourth rows, respectively (the value in the parenthesis is the standard error from 20 macro-replications.)

Algorithm	WDC	MSE ($\times 10^{-3}$)			
Performance	(θ)	Overall	2nd row	3rd row	4th row
Reference [17]	0.040	4.611 (0.023)	4.332 (0.022)	4.595 (0.026)	4.900 (0.028)
SOWC-1	0.053 (0.010)	4.574 (0.025)	4.194 (0.023)	4.601 (0.028)	5.045 (0.029)
SOWC-2	0.052 (0.019)	4.578 (0.025)	4.198 (0.023)	4.593 (0.028)	5.035 (0.029)

Figures 7a and 7b shows the box plots of power deficits in the turbines across different wind speeds when we optimally calibrate the WDC and blindly adopt the reference value for WFa. We observe clear differences

in the power deficit distributions in both cases. With the optimally calibrated value of $\theta = 0.053$, Figure 7a shows the maximum power deficit at the most downstream turbines is approximately 43% compared to the upstream turbines. On the contrary, when we employ the recommended value of $\theta = 0.040$, the maximum power deficit is around 54% in Figure 7b. Further, compared to Figure 7b, Figure 7a with $\theta = 0.053$ demonstrates that the range of power deficits among turbines is smaller, compared to that with $\theta = 0.040$. This result demonstrates the advantage of calibrating the WDC value, specific to each wind farm, using its operational data.



(a) Estimated power deficit for $\theta = 0.053$

(b) Estimated power deficit for $\theta = 0.040$

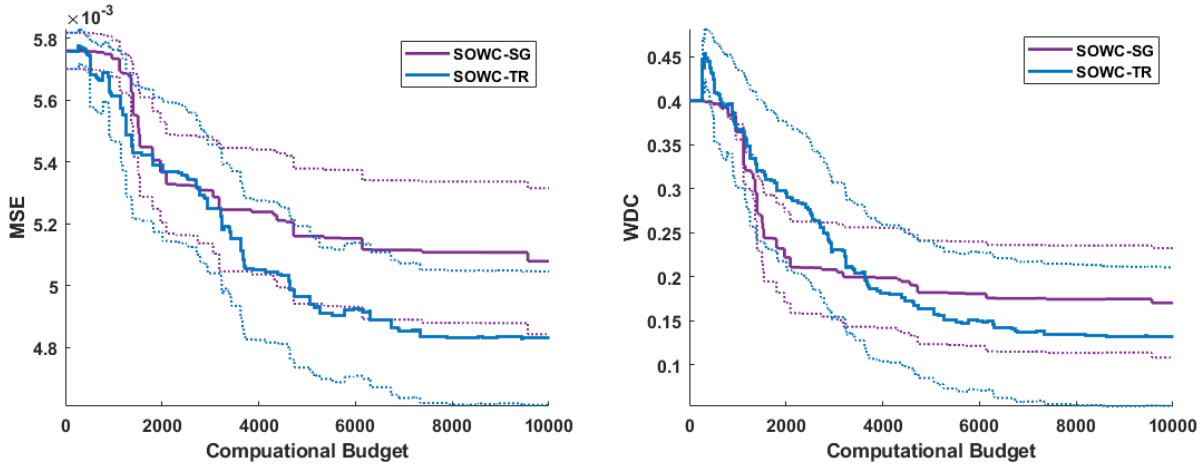
Figure 7: Comparison of the estimated power deficits for the optimally calibrated WDC ($\theta = 0.053$) and the standard value ($\theta = 0.040$) for WFa. Power deficit is computed as the difference of the estimated power output at each turbine and the maximum estimated power output amongst all the turbines.

Next, we compare our approach with alternative stochastic optimization methods. Liu et al. [28] adopted the SG for wake decay calibration and showed its superior performance over the Bayesian approach. As the TR-based optimization and SG are two representative stochastic optimization methods, we compare the performance of TR with SG. To compare the performance of various algorithms we use certain performance measures like mean-confidence interval curves. Specifically, after a macro-replication is run we get a set of intermediate solutions along with their corresponding budget. The MSE is then evaluated at these intermediate solutions using the remaining 30% set, the validation set, to get these performance measures.

Figure 8a plots the mean-confidence interval curves for the normalized MSE for WFa against the utilized budget for the original TR-based algorithm (SOWC-TR) and SG (SOWC-SG). SOWC-TR exhibits a sudden drop in the MSE value initially, implying fast convergence towards optimal solution. SOWC-SG, on the other hand, requires more computational budget to converge towards the optimum which is a well-known problem of SG. The slow convergence of SG can be attributed to the noisy steps that the algorithm takes during each iteration. This is visible in Figure 8b, which plots the mean-confidence intervals of the trajectory of the WDC during the search for WFa. In SG the intermediate solution is updated at each iteration and these frequent updates are computationally expensive. TR-based methods, on the other hand, update the intermediate solution only when there is sufficient reduction in MSE. The performance of SG also depends on the accuracy of the gradient estimates. For problems like WDC calibration in engineering wake models, the gradient cannot be expressed explicitly. Thus more computational budget is needed to get good estimates of the gradient. Overall, the derivative free TR-based optimization approach, which eliminates the need for good gradient estimates, exhibit many advantages over SG.

Figures 9a and 9b show the box plots for the normalized MSE and the estimated WDC, respectively for WFa. The MSE and estimated WDC obtained from 20 macrorepliations of SOWC-TR have a smaller spread (interquartile range), suggesting smaller variance amongst the experiments. This is of particular importance in stochastic optimization because it signifies the robustness of the algorithm. Comparing the mean values of MSE of the two algorithms we can conclude that SOWC-TR gives better results than SOWC-SG. Figures 8 and 9 illustrate that TR-based optimization can be efficiently used for WDC calibration.

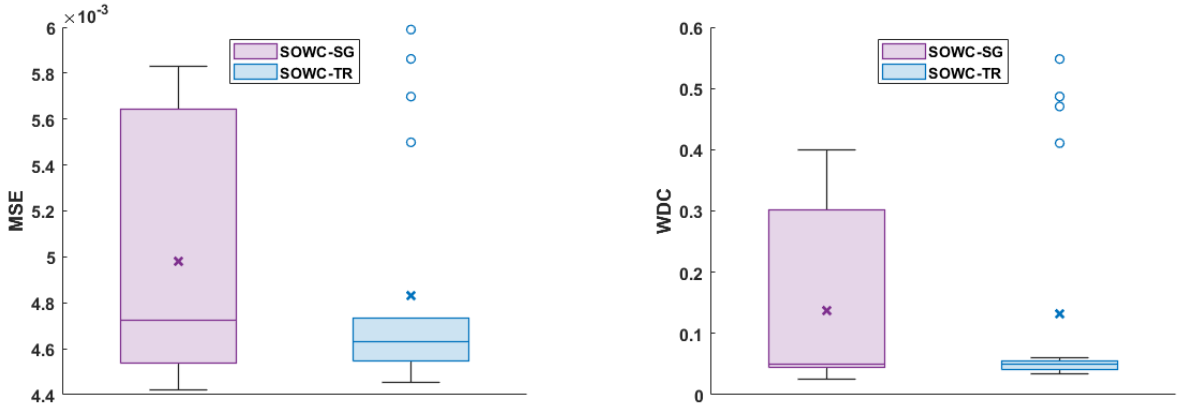
Given that SOWC-TR provides superior performance than SOWC-SG, we further compare the perfor-



(a) MSE trajectory.

(b) WDC trajectory.

Figure 8: Comparison of stochastic gradient (SOWC-SG) and TR optimization (SOWC-TR) over 20 macro-replications with mean and 95% CI convergence curves for WFa. The MSE and WDC are plotted against the number of times Jensen wake model is called during optimization.



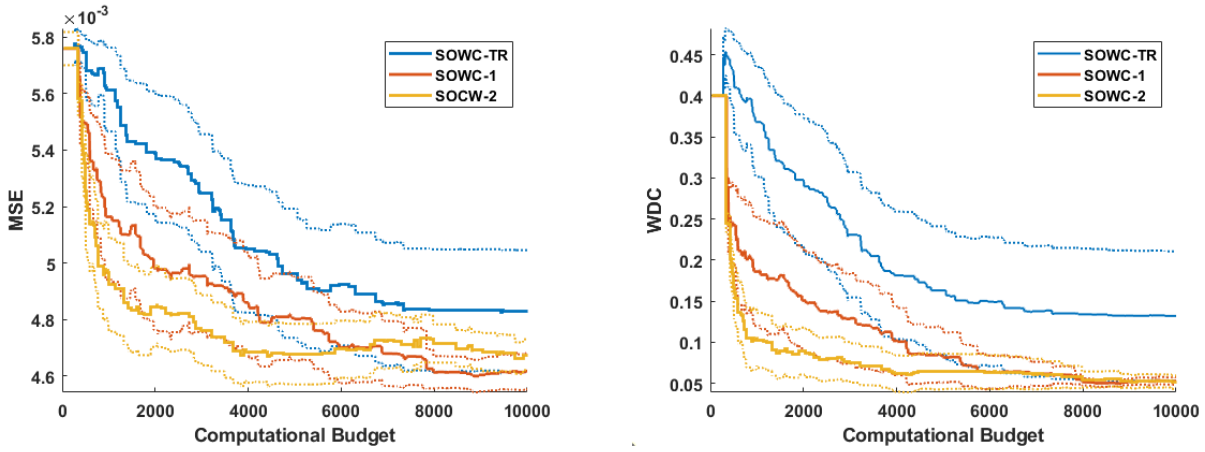
(a) Box plots for MSE.

(b) Box plots for calibrated WDC.

Figure 9: Comparison of the distribution of the outcome of SOWC-SG and SOWC-TR over 20 macro-replications gives mean MSE ($\times 10^{-3}$) for SOWC-SG = 5.000 and for SOWC-TR = 4.800 for WFa.

mance of SOWC-TR with its enhanced version with adaptive stratified sampling: SOWC-1 with dynamic weights and SOWC-2 with fixed weights. Even though SOWC-TR performs better than SOWC-SG, from Figures 9 and 11, we can see that the output of SOWC-TR still has some outliers, indicating high variance. We aim to reduce this variance by employing stratified sampling, thus warranting more accurate estimates. This can be seen in the convergence curves for normalized MSE (Figure 10a). The confidence intervals in Figure 10a are much narrower for SOWC-1 and SOWC-2, as compared to SOWC-TR. Further, during initial stages of optimization the reduction of MSE for the stratified sampling methods is much more than that of standard TR-based algorithm. This shows that by using stratified sampling we get better estimates with higher certainty. Looking at the trajectory of WDC in Figure 10b, we can see that the estimated values of WDC for SOWC-1 and SOWC-2 are more uniform across multiple macro-replications. The box plots for MSE (Figure 11a) and WDC (Figure 11b) show that SOWC-1 has no outliers and SOWC-2 has one outlier. This is extremely important because it shows that the algorithms are less sensitive to input uncertainty.

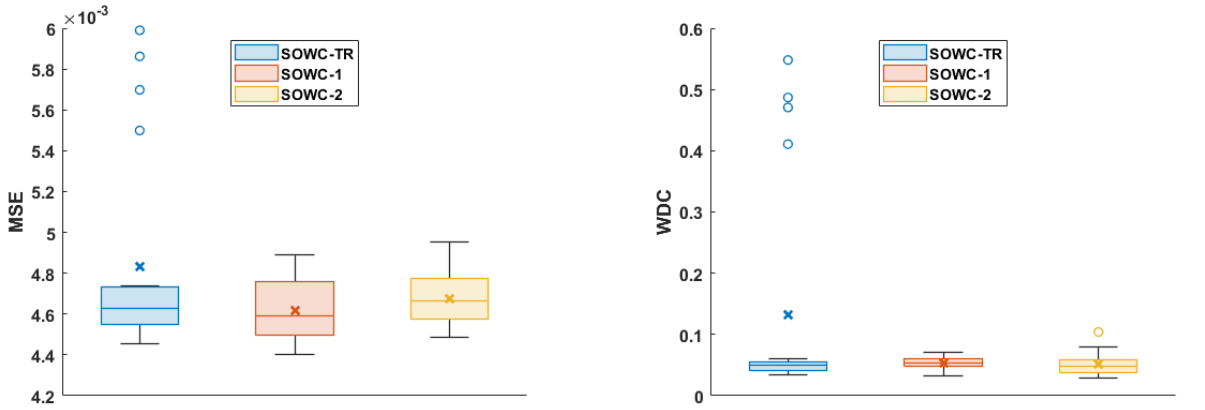
There is a slight difference in the performance of SOWC-1 and SOWC-2. SOWC-1 achieves slightly lower MSE (Figures 10a and 11a) as compared to SOWC-2. This is expected because in SOWC-1 the weights for



(a) MSE trajectory.

(b) WDC trajectory.

Figure 10: Comparison of the original TR optimization (SOWC-TR) and the proposed approaches: adaptive stratified sampling with dynamic weights (SOWC-1) and with fixed weights (SOWC-2) over 20 macro-replications with mean and 95% CI convergence curves for WFa.



(a) Optimal MSE distribution.

(b) Optimal WDC distribution.

Figure 11: Comparison of the optimal solution over 20 macro-replications for WFa shows while over-performing SOWC-TR and barely different from SOWC-2, SOWC-1 achieves a smaller variance.

stratified sampling are estimated using the variance of the loss function, whereas in SOWC-2 the weights are fixed and do not depend on the intermediate values of the WDC during the calibration process. SOWC-1, thus, captures the behavior of loss more accurately as compared to SOWC-2. This can be observed in Figure 10 by the amount of wriggling in the convergence curves for SOWC-2 and their smoothness for SOWC-1. We also observe SOWC-1 tends to yield more similar results in 20 different macro-replications, providing more robust solutions. However, we would like to mention that even though SOWC-1 marginally outperforms SOWC-2, SOWC-2's ease of implementation makes it a very handy tool.

In summary, both SOWC-1 and SOWC-2 outperform their alternatives, including SOWC-SG and SOWC-TR, by achieving faster convergence with higher certainty. Stratified sampling further improves the performance of TR-based optimization, making it more robust and efficient.

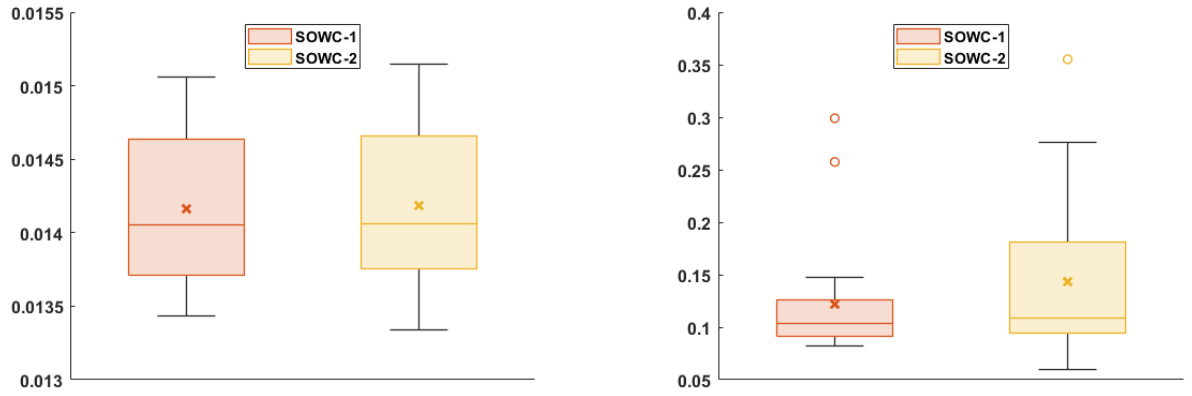
Next, we investigate the performance of our proposed calibration approach using SOWC-1 and SOWC-2 for the land-based wind farm, WFb. Table 4 compares the results of SOWC-1 and SOWC-2 with those of the recommended value of $\theta = 0.075$ for land-based wind farms. Even though the recommended values of WDC for SOWC-1 and SOWC-2 are different, their MSEs are rather comparable. The MSE with the recommended $\theta = 0.075$ is slightly higher than the MSEs derived from the calibrated values. Despite the

limited availability of data for WFb, the proposed calibration approaches exhibit improved performance.

Table 4: Comparison of MSE and estimated WDC for different methods for WFb. The second column summarizes the estimated WDC and the third column summarizes the overall MSE (the value in the parenthesis is the standard error from 20 macro-replications.)

Algorithm	WDC (θ)	MSE ($\times 10^{-3}$)
Reference [17]	0.075	14.276 (0.109)
SOWC-1	0.122 (0.056)	14.123 (0.119)
SOWC-2	0.143 (0.076)	14.140 (0.119)

Figures 12a and 12b show the distributions of MSE and WDC across 20 macro-replication for SOWC-1 and SOWC-2, respectively. While MSEs from SOWC-1 and SOWC-2 are comparable, the interquartile range for SOWC-1 is smaller, suggesting more certainty, compared to SOWC-2.



(a) Box plots for MSE.

(b) Box plots for calibrated WDC.

Figure 12: Results from 20 macro-replications for WFb: While MSEs from SOWC-1 and SOWC-2 are comparable, SOWC-1 produces more consistent WDC calibration results with less variation.

5.2.2. Functional Calibration

Some studies suggest that the value of the WDC depends on the local atmospheric conditions and thus, a constant value of WDC does not take into account these variations [39, 24]. Taking local atmospheric conditions into consideration, in this section, we model the WDC as a linear function of TI in (3) for WFa. The dependence of WDC on TI varies according to the atmospheric stability conditions. There are different ways to determine the atmospheric stability conditions. Obukhov Length [53] is typically used to categorize atmospheric stability. However, with the data available, we do not have enough information to determine the Obukhov Length.

Another way to determine atmospheric stability is via the wind profile power law, which states that if wind speed u_r at a reference height z_r is known, then wind speed u at some height z can be determined as

$$\left(\frac{u}{u_r}\right) = \left(\frac{z}{z_r}\right)^{\alpha}, \tag{15}$$

where α is the wake shear coefficient [54, 55]. The value of the wake shear coefficient can be used to determine near-neutral atmospheric conditions. The value of α varies with the terrain, and for lake or ocean surface, it is assumed to be 0.1 [56] but for wind power offshore operations, other values of α have been used [57]. Since we cannot determine the exact value of the wake shear coefficient at the studied wind farm location due to the limited information available to us, we use wind speeds measured at two different heights, one below and the other at the hub height, to determine α in (15) in each data record. Then, with the goal of

finding the parameters p and q in (3) under near-neutral conditions, we choose a subset of the dataset that exhibits the near-neutral conditions using the resulting wake shear coefficient values. Specifically, we divide the data into two sets for different ranges of α; the first set comprises of lower α values between 0.05 and 0.1, and the second set has α between 0.1 and 0.15. The size of the two sets is 2,034 and 1,174, respectively.

In literature, the WDC is often assumed to be directly proportional to TI ($\theta = c \times \text{TI}$) [23, 38, 39]. This is done because it is easier to estimate a single parameter. Calibrating two parameters increases the dimensionality of the problem making it more complex to solve. However, with the robustness and the efficiency that the our methods offer, the two-dimensional problem can be solved with relative ease.

We perform stochastic optimization on both sets to determine the optimal values of the intercept (p) and the slope (q). The setup is almost the same as the point calibration setup, the only difference being that now we have a two-dimensional problem. Alblas et al. [39] suggested a relation between the WDC and TI, $\theta \approx 0.5 \times \text{TI}$, for offshore wind farms under neutral conditions. We use this as a reference to compare the results of our optimization which are summarized in Tables 5 and 6. The second column in the table reports the estimated intercept, and the third column summarizes the estimated slope. The fourth column contains the values of MSE obtained from 20 validation sets for the entire wind farm. The last three columns contain the values of MSE for the turbines in the second, third, and fourth rows respectively.

Table 5 shows the results for the first data set having lower values of α. The optimal functional relations obtained from SOWC-1 and SOWC-2 are $\theta = 0.068 + 0.033 \times \text{TI}$ and $\theta = 0.079 + 0.026 \times \text{TI}$, respectively. The MSE values from SOWC-1 and SOWC-2 are almost equal to the MSE value if we use the relation suggested in [39]. The results for the second set are summarized in Table 6. For the second set, the optimal functional relations suggested by SOWC-1 and SOWC-2 are $\theta = 0.059 + 0.114 \times \text{TI}$ and $\theta = 0.059 + 0.098 \times \text{TI}$, respectively. The MSE reductions in SOWC-1 and SOWC-2 are more clear for larger α. In particular, similar to what we observed for point calibration, we obtain the substantial MSE reduction for turbines in the second row; for larger α, the MSE reduction is almost 19.01%, whereas it is approximately 1.72% for smaller α.

Table 5: Comparison of MSE and estimated WDC for $0.05 < \alpha < 0.1$ for WFa (the value in the parenthesis is the standard error from 20 macro-replications).

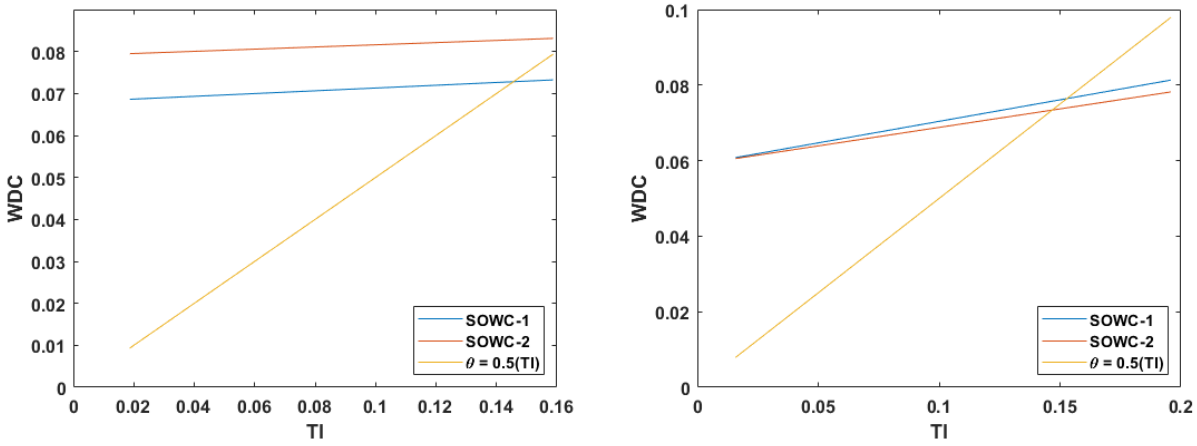
Algorithm	intercept	slope	MSE ($\times 10^{-3}$)			
Performance	(p)	(q)	Overall	2nd row	3rd row	4th row
Reference [39]	0.000	0.500	12.365 (0.182)	11.817 (0.174)	12.653 (0.183)	12.739 (0.211)
SOWC-1	0.068 (0.065)	0.033 (0.069)	12.379 (0.190)	11.569 (0.172)	12.747 (0.198)	12.999 (0.222)
SOWC-2	0.079 (0.105)	0.026 (0.035)	12.430 (0.190)	11.588 (0.172)	12.807 (0.198)	13.082 (0.222)

Table 6: Comparison of MSE and estimated WDC for $0.1 < \alpha < 0.15$ for WFa (the value in the parenthesis is the standard error from 20 macro-replications).

Algorithm	intercept	slope	MSE ($\times 10^{-3}$)			
Performance	(p)	(q)	Overall	2nd row	3rd row	4th row
Reference [39]	0.000	0.500	12.406 (0.225)	11.270 (0.224)	12.578 (0.253)	13.671 (0.229)
SOWC-1	0.059 (0.035)	0.114 (0.056)	9.957 (0.241)	9.039 (0.235)	10.221 (0.254)	10.836 (0.256)
SOWC-2	0.059 (0.023)	0.098 (0.051)	9.952 (0.241)	9.039 (0.235)	10.212 (0.254)	10.829 (0.256)

Figure 13 plots the three functional relations for two different α ranges. The WDC values suggested by SOWC-1 and SOWC-2 increase steadily with TI. Even for lower values of TI, the suggested functional relations give a respectable value. On the other hand, the WDC value estimated using the relation suggested in [39] is very low for lower TI values and then gets close to the values suggested by SOWC-1 and SOWC-2 for higher TI values.

The lower MSE achieved using stochastic optimization indicates that using the relation of $\theta \approx 0.5 \times \text{TI}$ is not optimal for this studied wind farm. Further the slope and intercept values obtained for the two sets are slightly different. This implies that the values of slope and intercept may need to be calibrated separately depending on α. An interesting thing to note is that the optimization algorithms always give a non-zero value for intercept. This suggests that direct proportionality between the WDC and TI cannot always be assumed.



(a) First subset ($0.05 < \alpha < 0.1$).

(b) Second subset ($0.1 < \alpha < 0.15$).

Figure 13: Comparison of the optimal functional relations between WDC and TI and the relation suggested by Alblas et al [39] for the two subsets for WFa.

5.3. Extension to Other Wake Models

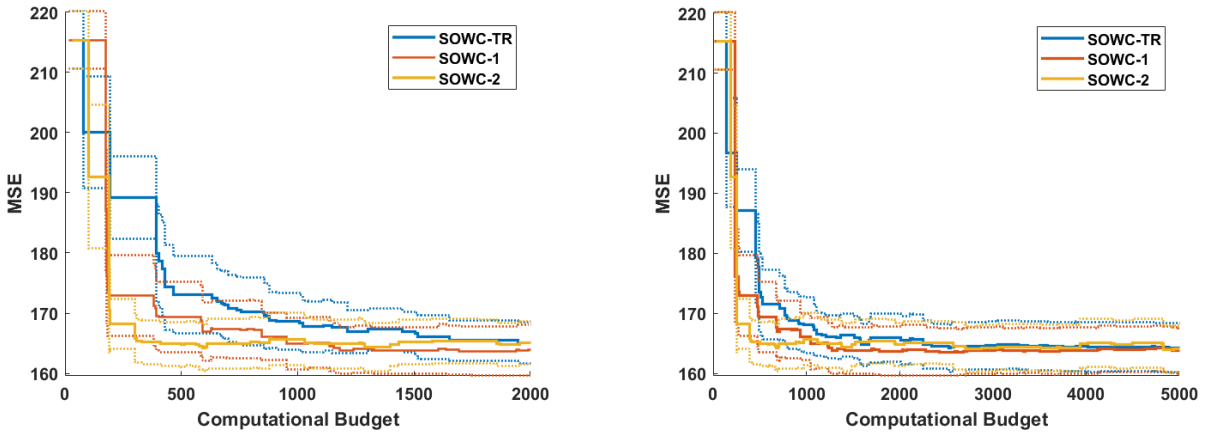
Jensen's model was among the first engineering wake models. Recent literature contains several improvements to Jensen's model. We implement the proposed calibration methodology to a more sophisticated wake model that integrates the lifting line and power yaw models assuming a Gaussian wake [25]. This integrated model also uses a secondary steering model that takes into account non-zero lateral wind speeds downstream of yawed turbines.

We apply our methodology to calibrate the wake parameters (k_w, σ_0) of this Gaussian model for a simple two-turbine setting. We assume that the parameters for both turbines are the same. The layout of the two turbines is available in Figure 8.6 of [58] (see Pair 1 therein). These turbines are a part of the larger wind farm, but there are no turbines within ten times the rotor diameter distance of this pair. To calibrate the model parameters, we use the operational data from this pair of turbines, including wind speeds and power being recorded at each turbine and the wind direction measured at the met mast [58]. The dataset does not include the yaw angle information, and we assume zero yaw angle. More information about the wind farm can be found in Section 8.6 of the "Data Science for Wind Energy" book [58], and the dataset can be accessed online at [59].

The bearing angle between the two turbines is 307.1°. Thus we have filtered the data to include wind directions between $[122.8°, 131.4°]$ and $[302.8°, 311.4°]$, where the wake phenomenon is expected to be most prominent. The size of this filtered data is 1,804. Though the power curve should be estimated using the free stream wind speed at the met mast, due to the unavailability of these wind speeds in the dataset available to us, we have used the wind speed at the upstream turbine to determine the power curve using B-splines [52], where the wind direction determines the upstream turbine. First, we stratify the data into two subsets based on the wind direction ranges. Then wind speed is used to further stratify the data in each of these intervals of wind direction using CART.

Figure 14 depicts the trajectories of MSE of the proposed stratified sampling-based methods, SOWC-1 and SOWC-2, as well as those of the original TR method without stratification, SOWC-TR. All of the three methods converge well with sufficient computational resources, e.g., when the computational budget allows up to 5000 evaluations, as shown in Figure 14(b). However, the stratified sampling-based methods exhibit faster convergence given limited computational resources, e.g., when the allowed budget is only up to 2,000 evaluation, as shown in Figure 14(a). Here, each evaluation implies each call of the Gaussian wake model with one data record. This faster convergence is essential in calibrating parameters in real use cases because actual wind farm sizes are significantly greater and data quantities are typically larger. This result is consistent with the finding that the non-stratified SOWC-TR requires a much larger computational budget for calibration when calibrating the Jensen's model parameter, as illustrated in Figure 10.

The outcomes of 20 macro-replications are summarized in Table 7. There is a small difference between



(a) Budget = 2000.

(b) Budget = 5000

Figure 14: Comparison of MSEs between SOWC-TR and the proposed SOWC-1 and SOWC-2 for calibrating the Gaussian wake model [25] with different computational budgets.

the calibrated parameter values from SOWC-1 and SOWC-2. This is because the MSEs are similar around the final converged values of $k_w = 1.80$ and $sigma_0 = 0.32$ (notice that the MSE of each technique is within $\pm$ 1 standard error of another approach).

Table 7: Averages of calibrated wake parameters and MSE for the Gaussian wake model in the studied wind farm from 20 macro-replications(the value in the parenthesis is the standard error.)

Budget	Algorithm	k_w	σ_0	MSE
5,000	SOWC-1	0.172 (0.009)	0.314 (0.007)	163.767 (1.931)
	SOWC-2	0.186 (0.008)	0.333 (0.007)	164.148 (1.971)
2,000	SOWC-1	0.178 (0.008)	0.318 (0.005)	163.906 (2.165)
	SOWC-2	0.180 (0.011)	0.328 (0.011)	165.121 (1.811)

6. Concluding Remarks

Accurate estimation of the WDC is of significant importance for the performance of the Jensen wake model. This work shows the applicability of data-driven stochastic optimization for wake calibration. We use novel stochastic optimization methods to get reliable estimates for the WDC. The robustness of the TR-based method is further improved by applying variance reduction techniques like stratified sampling. Decision trees are used to split similar data into multiple bins efficiently. Adaptive sampling is used in conjunction with stratified sampling to determine the optimal sample size from each stratum during optimization. With effective adaptive and stratified sampling, we pick the right sets of data points to estimate the loss function, leading to coherent computational budget usage. We present two algorithms: SOWC-1 uses dynamic weights that are calculated using the local loss function, which changes according to the search trajectory. SOWC-2 uses fixed weights using the variance of the original data, assuming that the variance of original data is positively correlated to the variance of the loss function. These proposed methods outperform the widely used SG methods.

We demonstrate a unique method that can be used to determine the wind farm-specific optimal parameter values of the Jensen model. We extend the proposed methodology to solve two-dimensional functional calibration problems. Using large datasets, one can efficiently use the new point and functional calibration approach. To illustrate that our approach can be easily applied to other advanced wake models, we implement it with the Gaussian wake model. Overall we show that the proposed strategy can result in efficient and robust calibration in engineering wake models and can be extended to other wake models [18, 19, 21, 25].

The proposed approach enables an understanding to power deficit patterns in existing wind farms, which can help design new wind farms optimally. Moreover, with use of wind profile power law for functional calibration, we show calculation of the wake shear coefficient and determine the points recorded when the atmospheric stability is near-neutral. In the future research, more information about the data will be explored to determine atmospheric stability. Our other future research direction is to implement dynamic stratification wherein the partitioning changes according to the search trajectory. Future research directions also involve calibrating wake parameters for individual turbines using collaborative learning approaches [60] and investigating how the wake effects influence the turbine reliability in a wind farm [25, 61].

Acknowledgement

This work was supported in part by the U.S. National Science Foundation under Grants IIS-1741166, CMMI-2226347, and CMMI-2226348.

References

[1] Enrico GA Antonini, David A Romero, and Cristina H Amon. Optimal design of wind farms in complex terrains using computational fluid dynamics and adjoint methods. Applied Energy, 261:114426, 2020.

[2] Cristina L Archer, Ahmadreza Vasel-Be-Hagh, Chi Yan, Sicheng Wu, Yang Pan, Joseph F Brodie, and A Eoghan Maguire. Review and evaluation of wake loss models for wind energy applications. Applied Energy, 226:1187–1207, 2018.

[3] Mingwei Ge, Ying Wu, Yongqian Liu, and Qi Li. A two-dimensional model based on the expansion of physical wake boundary for wind-turbine wakes. Applied energy, 233:975–984, 2019.

[4] Javier Serrano González, Ángel Luis Trigo García, Manuel Burgos Payán, Jesús Riquelme Santos, and Ángel Gaspar González Rodríguez. Optimal wind-turbine micro-siting of offshore wind farms: A grid-like layout approach. Applied energy, 200:28–38, 2017.

[5] Alayna Farrell, Jennifer King, Caroline Draxl, Rafael Mudafort, Nicholas Hamilton, Christopher J Bay, Paul Fleming, and Eric Simley. Design and analysis of a wake model for spatially heterogeneous flow. Wind Energy Science, 6(3):737–758, 2021.

[6] Lu Zhan, Stefano Letizia, and Giacomo Valerio Iungo. Optimal tuning of engineering wake models through lidar measurements. Wind Energy Science, 5(4):1601–1622, 2020.

[7] Nima Sedaghatizadeh, Maziar Arjomandi, Richard Kelso, Benjamin Cazzolato, and Mergen H Ghayesh. Modelling of wind turbine wake using large eddy simulation. Renewable Energy, 115:1166–1176, 2018.

[8] Matthew Churchfield, Sang Lee, Patrick Moriarty, Luis Martinez, Stefano Leonardi, Ganesh Vijayakumar, and James Brasseur. A large-eddy simulation of wind-plant aerodynamics. In 50th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition, page 537, 2012.

[9] S-P Breton, J Sumner, Jens Nørkær Sørensen, Kurt Schaldemose Hansen, Sasan Sarmast, and Stefan Ivanell. A survey of modelling methods for high-fidelity wind farm simulations using large eddy simulation. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 375(2091):20160097, 2017.

[10] Tuhfe Göçmen and Gregor Giebel. Data-driven wake modelling for reduced uncertainties in short-term possible power estimation. In Journal of Physics: Conference Series, volume 1037, page 072002. IOP Publishing, 2018.

[11] Mingdi You, Eunshin Byon, Jionghua Jin, and Giwhyun Lee. When wind travels through turbines: A new statistical approach for characterizing heterogeneous wake effects in multi-turbine wind farms. IISE Transactions, 49(1):84–95, 2017.

[12] Mingdi You, Bingjie Liu, Eunshin Byon, Shuai Huang, and Jionghua Jin. Direction-dependent power curve modeling for multiple interacting wind turbines. IEEE Transactions on power systems, 33(2):1725–1733, 2017.

[13] Antonio Crespo, J Hernandez, and Sten Frandsen. Survey of modelling methods for wind turbine wakes and wind farms. Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology, 2(1):1–24, 1999.

[14] John F Ainslie. Calculating the flowfield in the wake of wind turbines. Journal of wind engineering and Industrial Aerodynamics, 27(1-3):213–224, 1988.

[15] RJ Barthelmie and SC Pryor. Meteorology and wind resource assessment for wind farm development. In Wind Energy Systems, pages 3–e28. Elsevier, 2011.

[16] Niels Otto Jensen. A note on wind generator interaction. 1983.

[17] I Katic, Jørgen Højstrup, and Niels Otto Jensen. A simple model for cluster efficiency. In European wind energy association conference and exhibition, volume 1, pages 407–410. A. Raguzzi Rome, Italy, 1986.

[18] Sten Frandsen, Rebecca Barthelmie, Sara Pryor, Ole Rathmann, Søren Larsen, Jørgen Højstrup, and Morten Thøgersen. Analytical modelling of wind speed deficit in large offshore wind farms. Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology, 9(1-2):39–53, 2006.

[19] Majid Bastankhah and Fernando Porté-Agel. A new analytical model for wind-turbine wakes. Renewable energy, 70:116–123, 2014.

[20] Pieter MO Gebraad, FW Teeuwisse, Jan-Willem van Wingerden, Paul A Fleming, Shalom D Ruben, Jason R Marden, and Lucy Y Pao. A data-driven model for wind plant power optimization by yaw control. In 2014 American Control Conference, pages 3128–3134. IEEE, 2014.

[21] Gunner Chr Larsen. A simple wake calculation procedure. Risø National Laboratory, 1988.

[22] RJ Barthelmie, Matthew J Churchfield, Patrick J Moriarty, Julie K Lundquist, GS Oxley, S Hahn, and SC Pryor. The role of atmospheric stability/turbulence on wakes at the egmond aan zee offshore wind farm. In Journal of Physics: Conference Series, volume 625, page 012002. IOP Publishing, 2015.

[23] Thomas Duc, Olivier Coupiac, Nicolas Girard, Gregor Giebel, and Tuhfe Göçmen. Local turbulence parameterization improves the jensen wake model and its implementation for power optimization of an operating wind farm. Wind Energy Science, 4(2):287–302, 2019.

[24] Alfredo Peña and Ole Rathmann. Atmospheric stability-dependent infinite wind-farm models and the wake-decay coefficient. Wind Energy, 17(8):1269–1285, 2014.

[25] Michael F Howland, Jesús Bas Quesada, Juan José Pena Martínez, Felipe Palou Larrañaga, Neeraj Yadav, Jasvipul S Chawla, Varun Sivaram, and John O Dabiri. Collective wind farm operation based on a predictive model increases utility-scale energy production. Nature Energy, 7(9):818–827, 2022.

[26] Johannes Schreiber, Carlo L Bottasso, Bastian Salbert, and Filippo Campagnolo. Improving wind farm flow models by learning from operational data. Wind Energy Science, 5(2):647–673, 2020.

[27] Marc C Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63(3):425–464, 2001.

[28] Bingjie Liu, Xubo Yue, Eunshin Byon, and Raed Al Kontar. Parameter calibration in wake effect simulation model with stochastic gradient descent and stratified sampling. The Annals of Applied Statistics, 16(3):1795–1821, 2022.

[29] T. Duc, O. Coupiac, N. Girard, G. Giebel, and T. Göçmen. Local turbulence parameterization improves the Jensen wake model and its implementation for power optimization of an operating wind farm. Wind Energy Science, 4(2):287–302, 2019.

[30] Tuhfe Göçmen, Paul Van der Laan, Pierre-Elouan Réthoré, Alfredo Peña Diaz, Gunner Chr Larsen, and Søren Ott. Wind turbine wake models developed at the technical university of denmark: A review. Renewable and Sustainable Energy Reviews, 60:752–769, 2016.

[31] Cheoljoon Jeong, Ziang Xu, Albert S Berahas, Eunshin Byon, and Kristen Cetin. Multiblock parameter calibration in computer models. INFORMS Journal on Data Science, 2023.

[32] Rebecca Jane Barthelmie, S C Pryor, Sten Tronæs Frandsen, Kurt Schaldemose Hansen, JG Schepers, K Rados, W Schlez, A Neubert, LE Jensen, and S Neckelmann. Quantifying the impact of wind turbine wakes on power output at offshore wind farms. Journal of Atmospheric and Oceanic Technology, 27(8):1302–1317, 2010.

[33] Andrea Staid. Statistical modeling to support power system planning. Ph. D. Thesis, 2015.

[34] Sten Frandsen. On the wind speed reduction in the center of large clusters of wind turbines. Journal of Wind Engineering and Industrial Aerodynamics, 39(1-3):251–265, 1992.

[35] Office of Energy Efficiency & Renewable Energy. Wind Turbines: the Bigger, the Better. Available: `https://www.energy.gov/eere/articles/wind-turbines-bigger-better`, 2021.

[36] James F Manwell, Jon G McGowan, and Anthony L Rogers. Wind energy explained: theory, design and application. John Wiley & Sons, 2010.

[37] LE Jensen. Array efficiency at horns rev and the effect of atmospheric stability. Dong Energy Presentation, 2007.

[38] Alfredo Peña, Pierre-Elouan Réthoré, and M Paul van der Laan. On the application of the jensen wake model using a turbulence-dependent wake decay coefficient: the sexbierum case. Wind Energy, 19(4):763–776, 2016.

[39] Laurens Alblas, Wim Bierbooms, and Dick Veldkamp. Power output of offshore wind farms in relation to atmospheric stability. In Journal of Physics: Conference Series, volume 555, page 012004. IOP Publishing, 2014.

[40] Amin Niayifar and Fernando Porté-Agel. Analytical modeling of wind farms: A new approach for power prediction. Energies, 9(9):741, 2016.

[41] Carl R Shapiro, Dennice F Gayme, and Charles Meneveau. Modelling yawed wind turbine wakes: a lifting line approach. Journal of Fluid Mechanics, 841:R1, 2018.

[42] Andrew R Conn, Nicholas IM Gould, and Philippe L Toint. Trust region methods. SIAM, 2000.

[43] Sara Shashaani, Fatemeh S Hashemi, and Raghu Pasupathy. ASTRO-DF: A class of adaptive sampling trust-region algorithms for derivative-free stochastic optimization. SIAM Journal on Optimization, 28(4):3145–3176, 2018.

[44] Yunsoo Ha, Sara Shashaani, and Quac Tran-Dinh. Improved complexity of turst-region optimization for zeroth-order stochastic oracles with adaptive sampling. In S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, editors, Proceedings of the 2021 Winter Simulation Conference, Piscataway, New Jersey, 2021. Institute of Electrical and Electronics Engineers, Inc.

[45] Andrew R Conn, Katya Scheinberg, and Luis N Vicente. Introduction to derivative-free optimization. SIAM, 2009.

[46] Sheldon Ross. Chapter 9 - variance reduction techniques. In Sheldon Ross, editor, Simulation (Fifth Edition), pages 153 – 231. Academic Press, fifth edition edition, 2013.

[47] Christiane Lemieux. Monte carlo and quasi-monte carlo sampling. Springer Science & Business Media, 2009.

[48] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. Classification and regression trees. CRC press, 1984.

[49] Charles Tong. Refinement strategies for stratified sampling methods. Reliability Engineering & System Safety, 91(10-11):1257–1265, 2006.

[50] De-gan Zhang, Chen-hao Ni, Jie Zhang, Ting Zhang, Peng Yang, Jia-xu Wang, and Hao-ran Yan. A novel edge computing architecture based on adaptive stratified sampling. Computer Communications, 183:121–135, 2022.

[51] Pranav Jain, Sara Shashaani, and Eunshin Byon. Wake effect calibration in wind power systems with adaptive sampling based optimization. In Proceedings of the IISE Annual Conference., pages 43–48. Institute of Industrial and Systems Engineers (IISE), 2021.

[52] Giwhyun Lee, Eunshin Byon, Lewis Ntaimo, and Yu Ding. Bayesian spline method for assessing extreme loads on wind turbines. The Annals of Applied Statistics, pages 2034–2061, 2013.

[53] AM Obukhov. Turbulence in an atmosphere with a non-uniform temperature. Boundary-layer meteorology, 2(1):7–29, 1971.

[54] Eunshin Byon, Eduardo Pérez, Yu Ding, and Lewis Ntaimo. Simulation of wind farm operations and maintenance using discrete event system specification. Simulation, 87(12):1093–1117, 2011.

[55] Shuoran Li, Young Myoung Ko, and Eunshin Byon. Nonparametric importance sampling for wind turbine reliability analysis with stochastic computer models. The Annals of Applied Statistics, 15(4):1850 – 1871, 2021.

[56] ML Ray, AL Rogers, and JG McGowan. Analysis of wind shear models and trends in different terrains. University of Massachusetts, Department of Mechanical and Industrial Engineering, Renewable Energy Research Laboratory, 2006.

[57] Alfredo Pena Diaz, Torben Mikkelsen, Sven-Erik Gryning, Charlotte Bay Hasager, Andrea N Hahmann, Merete Badger, Ioanna Karagali, and Michael Courtney. Offshore vertical wind shear: Final report on norsewind's work task 3.1. 2012.

[58] Yu Ding. Data science for wind energy. CRC Press, 2019.

[59] Yu Ding. Wake effect dataset. `https://doi.org/10.5281/zenodo.5516558`, September 2021.

[60] Ying Lin, Kaibo Liu, Eunshin Byon, Xiaoning Qian, Shan Liu, and Shuai Huang. A collaborative learning framework for estimating many individualized regression models in a heterogeneous population. IEEE Transactions on Reliability, 67(1):328–341, 2017.

[61] Youngjun Choe, Qiyun Pan, and Eunshin Byon. Computationally efficient uncertainty minimization in wind turbine extreme load assessments. Journal of Solar Energy Engineering, 138(4), 2016.