NONPARAMETRIC IMPORTANCE SAMPLING FOR WIND TURBINE RELIABILITY ANALYSIS WITH STOCHASTIC COMPUTER MODELS

BY SHUORAN LI[1], YOUNG MYOUNG KO[2] AND EUNSHIN BYON[3]

[1]*Department of Statistics, University of Pittsburgh, SHL198@pitt.edu*

[2]*Department of Industrial and Management Engineering, Pohang University of Science and Technology, youngko@postech.ac.kr*

[3]*Department of Industrial and Operations Engineering, University of Michigan, ebyon@umich.edu*

Using aeroelastic stochastic simulations, this study presents an importance sampling method for assessing wind turbine reliability. As the size of modern wind turbines gets larger, structural reliability analysis becomes more important to prevent any catastrophic failures. At the design stage, operational data do not exist or are scarce. Therefore, aeroelastic simulation is often employed for reliability analysis. Importance sampling is one of the powerful variance reduction techniques to mitigate computational burden in stochastic simulations. In the literature, wind turbine reliability assessment with importance sampling has been studied with a single variable, wind speed. However, other atmospheric stability conditions also impose substantial stress on the turbine structure. Moreover, each environmental factor's effect on the turbine's load response depends on other factors. This study investigates how multiple environmental factors collectively affect the turbine reliability. Specifically, we devise a new nonparametric importance sampling method that can quantify the contributions of each environmental factor and its interactions with other factors, while avoiding computational problems and data sparsity issue arising in rare event simulation. Our wind turbine case study and numerical examples demonstrate the advantage of the proposed approach.

1. Introduction. Wind energy is one of the largest sources of renewable electricity in the U.S. (Aziz Ezzat, Jun and Ding (2019)). During the last decade, it contributed to 27% of new capacity additions in the U.S. (Wiser et al. (2020)). In addition to new installations of wind power plants, the growing size of utility-scale turbines contributes to the wind energy penetration in the electricity market. Large blades increase a swept area (i.e., area through which the blades spin) of the rotor, extracting more energy from wind (Ding (2019)). Thus, in an effort to achieve economic competitiveness with conventional generators, wind turbine size has been rapidly growing in the past few decades. The rotor diameter of a commercial wind turbine was about 30 meters (m) in 1990–1995, whereas today's modern land-based turbine's diameter is about 100 m or even longer (Wiser et al. (2016)). For supersized offshore turbines, blade rotor diameters are expected to increase up to 190 m in the near future (Wiser et al. (2016)).

While being more economic, the increasing size, on the other hand, renders the turbine structure experience escalated stresses (Lee et al. (2013)). To avoid catastrophic structural damages or failures, the International Electrotechnical Commission (IEC)'s design standards require turbine manufactures to evaluate the turbine reliability at the design stage. However, at the design stage operational data is not available oftentimes. Even if field measurement campaigns can be conducted, they collect data for a short duration, so data obtained is not sufficient to cover various load conditions (Lee et al. (2013)). Moreover, field campaigns are

usually conducted only for selected prototype turbines, primarily due to enormous campaign costs.

Thanks to the advent of computing power and sophisticated computer models, simulation-based reliability evaluation has gained attention in the wind energy industry (Moriarty (2008)). The U.S. Department of Energy (DOE)'s National Renewable Energy Laboratory (NREL) has been developing and maintaining a set of computer models, including TurbSim (Jonkman (2009)) and FAST (Jonkman and Buhl Jr (2005)), to simulate wind turbine operations. However, to make computer models closely mimic real systems, each simulation run becomes computationally expensive which poses challenges to reliability evaluation, even with advanced computing facilities. Moriarty (2008) used grid computing with 40 desktops for weeks to simulate a wind turbine's structural responses to various environmental conditions. Manuel, Nguyen and Barone (2013) further used a Linux computer cluster with 1024 cores at the U.S. Sandia National Laboratory.

Broadly, there are two major approaches for the reliability estimation with stochastic simulation: Monte Carlo sampling (MCS) approach and emulator-based approach. The emulator-based approach builds a statistical surrogate model for estimating the response surface of the computer model. Gaussian process (GP) gets popularity for constructing surrogate models in the recent computer experiment literature. However, the emulator-based approach mainly focuses on estimating the overall pattern of simulation output over an entire input space. Cannamela, Garnier and Iooss (2008) discussed that emulators tend to smooth a response surface and may not provide a good fit for the tail probability estimation. Thus, this study focuses on the MCS approach.

To relieve the computational burden while improving estimation accuracy (in terms of estimation variance), variance reduction techniques can be employed (Cannamela, Garnier and Iooss (2008)). Importance sampling (IS) is one of the most popular variance reduction techniques to improve simulation efficiency in rare event simulation. Most IS studies consider deterministic computer models that generate a fixed output at a given input (Dubourg, Sudret and Deheeger (2013), Givens and Raftery (1996), Kurtz and Song (2013), Neddermeyer (2009)). Simulations with deterministic computer models are still called stochastic simulations because inputs are often stochastic.

Unlike deterministic computer models, the NREL computer models (Turbsim/FAST) generate random load responses, given a fixed wind condition. We call such computer models stochastic computer models. Further, when a computer model is a black box, it is called a stochastic black box computer model. Choe, Byon and Chen (2015) derived the optimal IS density for stochastic black box computer models, referred to as the stochastic IS (shortly, SIS). However, the optimal density includes an unknown quantity. It is only theoretically optimal and cannot be directly implementable in practice. Several approximations have been discussed in the literature (Cao and Choe (2019), Chen and Choe (2019), Choe, Byon and Chen (2015)), and they have been applied to wind turbine reliability evaluation with a single input variable, wind speed (Chen and Choe (2019), Choe, Pan and Byon (2016), Pan et al. (2020)). A detailed review will be provided in Section 2.

This study is concerned with the wind turbine reliability analysis with multivariate environmental factors. While wind speed is the most important factor, atmospheric stability conditions, such as turbulence intensity and wind shear, also substantially affect structural loads (Gualtieri (2016)). For example, blade tip deflection, one of important load types, tends to get large when the turbulence intensity or wind shear becomes strong (to be discussed in detail in Section 3). Further, each variable's marginal effect on the turbine reliability varies, depending on other environmental variables, in other words, interaction effects exist.

To capture such complicated characteristics, we propose a nonparametric IS method that does not require a prespecified density form a priori. Specifically, to handle the interaction

effects of multivariate environmental factors, we devise a new multivariate kernel method. Even though multiplicative kernels can handle interaction effects, they often suffer data sparsity issue in high-dimensional problems when sufficient data is not collected over the entire input space. This issue is critical in the reliability assessment (more broadly, rare event simulations), because the main objective is to reduce simulation repetitions while minimizing the estimation variance. Moreover, bandwidth selection in the multiplicative kernel becomes more challenging, as the input dimension increases. To address these issues, our approach constructs multiple bivariate kernels, each of which consists of two input environmental factors. Then, to approximate the optimal SIS density we combine the bivariate kernels in an additive manner. In doing so, we assign different weights to each bivariate kernel to reflect the contributions of each variable and its interaction effects with other variables on the response. The proposed approach is referred to as the weighted additive multiplicative kernel SIS method (WAMK-SIS) in this study.

Our wind turbine case study demonstrates that the proposed approach successfully identifies influential environmental variables and their interaction effects so that the simulation process can be intelligently guided toward computational efficiency. As a result, our approach significantly saves computational resources over the crude Monte Carlo (CMC) sampling and outperforms alternative methods. We also present numerical examples to highlight the benefits of the proposed approach in various settings.

The organization of the paper is as follows. Existing approaches are reviewed in Section 2. Section 3 presents the proposed WAMK-SIS approach. We evaluate the performance of WAMK-SIS, in comparison with alternatives, with numerical examples and the wind turbine case study in Sections 4 and 5, respectively. Section 6 concludes the paper.

2. Literature review. This section reviews existing IS approaches used for the reliability evaluation with computer models. Let $\mathbf{X} \in \mathbf{R}^d$ denote a random input vector, following a known density, f, where d denotes the dimension of $\mathbf{X}$. From the input $\mathbf{X}$ a simulator generates an output, $Y \in \mathbf{R}$.

We first review the IS approach, called DIS, when the simulator uses a deterministic computer model where the output Y is a deterministic function of X, say, $Y = g(\mathbf{X})$. With a black box computer model, $g(\cdot)$ is not explicitly known but can be evaluated from simulation. In reliability analysis a quantity of interest is the failure probability, $P(Y > l) = E_X[\mathbb{I}(g(\mathbf{X}) > l)]$, also known as probability of exceedance (POE), where l denotes a structure's resistance level and $\mathbb{I}(\cdot)$ is the indicator function.

The crude Monte Carlo (CMC) method is one of the simplest methods to estimate the failure probability. In CMC we independently draw $\mathbf{X}_i, i = 1, 2, \ldots, N_T$, from its density, f, and then unbiasedly estimate the failure probability using the estimator

$$\hat{P}_{\text{CMC}} = \frac{1}{N_T} \sum_{i=1}^{N_T} \mathbb{I}(g(\mathbf{X}_i) > l),$$

where N_T is the total number of simulation replications. When the failure probability is small, failure events rarely occur, and thus CMC requires a large number of replications until a failure event is observed. It implies that, with computationally expensive computer models, extensive computational resources are needed to obtain an accurate estimate.

To save computational budgets in rare event simulation, IS changes the sampling distribution of $\mathbf{X}$ from $f(\mathbf{x})$ to another distribution, $q(\mathbf{x})$, in an attempt to make failure events occur more frequently with the following estimator:

$$(1) \qquad \hat{P}_{\text{DIS}} = \frac{1}{N_T} \sum_{i=1}^{N_T} \mathbb{I}(g(\mathbf{X}_i) > l) \frac{f(\mathbf{X}_i)}{q(\mathbf{X}_i)},$$

which is an unbiased estimator under the condition that $q(\mathbf{x}) > 0$, whenever $\mathbb{I}(g(\mathbf{x}) > l) f(\mathbf{x}) > 0$. The theoretically optimal density that minimizes the variance of $\hat{P}_{\mathrm{DIS}}$ (Rubinstein and Kroese (2017)) is derived as

$$q_{\mathrm{DIS}}(\mathbf{x}) = \frac{\mathbb{I}(g(\mathbf{x}) > l) f(\mathbf{x})}{P(Y > l)}.$$

In fact, $q_{\mathrm{DIS}}(\mathbf{x})$ renders var$[\hat{P}_{\mathrm{DIS}}]$ zero. However, $q_{\mathrm{DIS}}(\mathbf{x})$ is not practically implementable since $\mathbb{I}(g(\mathbf{x}) > l)$ can be evaluated only by running the simulator with the unknown function $g(\mathbf{x})$ and the denominator is exactly the quantity we want to estimate. Several methods that approximate $q_{\mathrm{DIS}}(\mathbf{x})$ have been studied, including parametric approaches, for example, cross entropy (CE) (de Boer et al. (2005)), metamodel-based (Dubourg, Sudret and Deheeger (2013)) methods and nonparametric approaches (Givens and Raftery (1996), Neddermeyer (2009)).

When the computer model is stochastic, unlike the deterministic computer model, the output is random, even at a fixed input $\mathbf{X} = \mathbf{x}$. This is because a random vector, ϵ, is embedded (or hidden) inside the computer model. Therefore, we can express the output as $Y = g(\mathbf{X}, \epsilon)$ (Choe, Byon and Chen (2015)). Because sampling ϵ is uncontrollable, with a black box computer model one cannot apply the IS approach to ϵ. Instead, one can use IS to bias the density of $\mathbf{X}$ only. Choe, Byon and Chen (2015) proposed an IS method for the stochastic black box computer model and presented two estimators. The first estimator allows multiple replications at each sampled input, whereas the second estimator permits only one replication at each input. Both estimators showed similar estimation performance (Choe, Byon and Chen (2015)), and some advantages of the second estimator were discussed in Ko and Byon (2021). Here, we briefly review the second estimator, defined as

$$(2) \qquad \hat{P}_{\mathrm{SIS}} = \frac{1}{N_T} \sum_{i=1}^{N_T} \mathbb{I}(Y_i > l) \frac{f(\mathbf{X}_i)}{q(\mathbf{X}_i)},$$

where Y_i is an output of $\mathbf{X}_i$, $i = 1, 2, \ldots, N_T$. Its theoretically optimal IS density minimizing the variance of $\hat{P}_{\mathrm{SIS}}$ in (2) is given by

$$(3) \qquad q_{\mathrm{SIS}}(\mathbf{x}) = \frac{1}{C_q} f(\mathbf{x}) \sqrt{s(\mathbf{x})},$$

where $s(\mathbf{x}) = P(Y > l | \mathbf{X} = \mathbf{x})$ denotes the conditional failure probability or conditional POE and C_q is the normalizing constant, that is, $C_q = \int_{\mathcal{X}_f} f(\mathbf{x}) \sqrt{s(\mathbf{x})} \, d\mathbf{x}$ (Choe, Byon and Chen (2015)).

Since the conditional POE is unknown, finding a good estimation of $s(\mathbf{x})$ is essential for the success of SIS implementation. This is also analogue to the DIS method where finding a good approximation of $\mathbb{I}(g(\mathbf{x}) > l)$ in (1) is critical. Metamodel-based regression method was used in Choe, Byon and Chen (2015), Choe, Pan and Byon (2016). Studies in Choe, Lam and Byon (2018), Pan, Ko and Byon (2021) further quantified the estimation uncertainties by constructing the confidence intervals. Recently, Pan et al. (2020) used the SIS approach to efficiently estimate the extreme quantile of the response variable, given the target failure probability level. In these studies a single dimensional input has been considered. With multivariate interacting inputs, constructing a high quality metamodel could be challenging. A poor quality metamodel might miss-specify important input regions, negatively affecting the simulation efficiency.

To address the challenge in constructing a high-quality metamodel, Cao and Choe (2019) employed the CE procedure with Gaussian mixture model (GMM), referred to as CE-SIS. Specifically, they use multiple replications at the same input value, $\mathbf{x}_i$, to obtain a rough

estimate of $s(\mathbf{x}_i)$ and estimate the GMM parameters with the expectation maximization algorithm. While CE-SIS works well for univariate input problems, its performance gets worse and even underperforms CMC as the input dimension increases. Chen and Choe (2019) proposed a two-step procedure where the data generated in the first step is used to estimate the conditional POE, using either parametric or nonparametric regression models. The parametric approach has the same limitation in the aforementioned metamodel-based approach. Their nonparametric approach focused on investigating how the budget allocations between the two stages affect the overall performance, instead of obtaining a good kernel function to approximate the optimal SIS density.

This study proposes a new nonparametric method that does not require a pre-specified form in defining the IS density. In particular, we devise a multivariate kernel approach that can capture the interaction effects among multiple input variables while taking the scalability issue into consideration.

3. Methodology. This section first presents the wind turbine simulator and discusses the proposed WAMK-SIS approach.

3.1. *Wind turbine simulator.* A set of NREL computer models, including TurbSim (Jonkman (2009)) and FAST (Jonkman and Buhl Jr (2005)), have been widely used to simulate a wind turbine's structural load responses (Manuel, Nguyen and Barone (2013), Moriarty (2008)). Given the input environmental condition (e.g., wind speed, turbulence intensity and wind shear) during a short specified time interval (e.g., 10 minutes), TurbSim generates a three-dimensional spatiotemporal wind profile. It uses more than eight million random variables to generate a stochastic wind profile around a rotor plane and passes it to FAST (Jonkman (2009)). Thus, TurbSim is the main source of randomness in the wind turbine simulation. Then, FAST generates load responses during the corresponding time interval. We consider the blade tip deflection (unit: m) as the output variable, Y, in this study. The tip deflection is one of the important load types, because catastrophic failures may occur when a deflected blade hits the tower (Choi et al. (2012), Kong, Bang and Sugiyama (2005)).

In addition to wind speed, V, other environmental factors associated with the atmospheric stability condition, including the turbulence intensity, I, and wind shear, S, also affect the load response (Gualtieri (2016)). Therefore, in this study we include them in our analysis and define the input vector as $\mathbf{X} = (V, I, S)$. Following the international standard, IEC61400-1 (International Electrotechnical Commission (2005)), and industry practice, each replication implies the simulation of wind turbine operation during a 10-minute interval. Then, the three input variables are defined as follows (Jonkman (2009), Lee et al. (2015)):

- Wind speed, V: 10-minute average wind speed.
- Turbulence intensity, I: Coefficient of variation of wind speeds during a 10 minute interval, that is, $I = \frac{\sigma}{V}$, where σ is the standard deviation of wind speeds during a 10-minute interval.
- Wind shear, S: Let $V(h)$ denote the wind speed measured at height h, and let h_{ref} denote the reference height (typically, a turbine's hub height). Then, wind shear is defined as $S = \frac{\ln V(h) - \ln V(h_{\text{ref}})}{\ln h - \ln h_{\text{ref}}}$.

For the turbine model we use the NREL 5.0 MW baseline horizontal-axis wind turbine with three blades (Jonkman et al. (2005)). Its rotor diameter is 126 m and hub height is 90 m. Detailed turbine specification is available in Jonkman et al. (2005).

Figure 1 shows scatter plots between each environmental variable and blade tip deflection, using data generated from the NREL simulators. Here, we use a large number of samples (15,000 samples) for illustration purpose. In our actual case study we use much fewer samples in each experiment. The normalized output is obtained by dividing the tip deflection
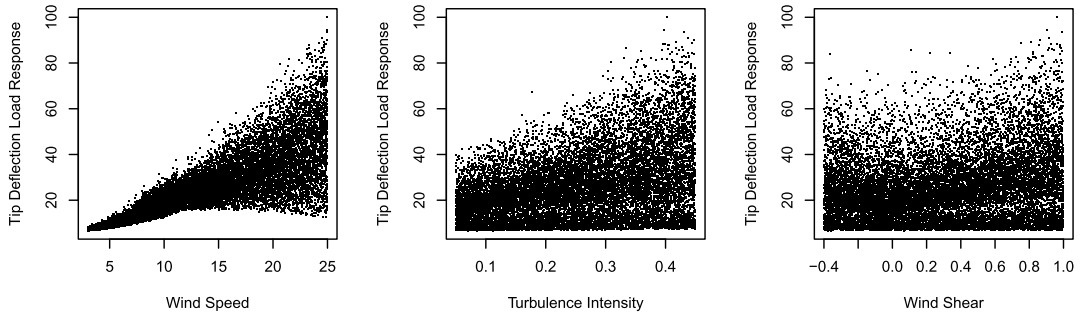
FIG. 1. *Scatter plots of the blade tip deflection vs. each environmental factor.*

by the maximum value we obtain. All three variables affect the tip deflection, among which wind speed appears to be the most significant factor, followed by turbulence intensity. The variability of tip deflection gets larger, as wind speed increases. It implies that, under the high wind speed regime, the influence of the other two variables may be more substantial, compared to the low wind speed regime. We also observe similar patterns with the other two variables, although the patterns are less clear.

Figure 2 further depicts the scatter plots of the tip deflection vs. environmental factors under specific wind speed or turbulence intensity ranges. Notably, the effect of each environmental variable depends on other environmental conditions. On the top panels, under the high wind speed regime (see the rightmost figure), blade tip deflection rapidly changes as the turbulence intensity increases, whereas the change is insignificant under the low wind speed regime (see the leftmost figure). The middle panels show similar patterns; the influence of wind shear becomes more obvious, as wind speed increases. These patterns explain the high variability of the tip deflection under high wind speeds, as observed in Figure 1. Lastly, the bottom panels suggest that the influence of wind shear does not appear to be affected by the turbulence intensity significantly, although tip deflection tends to slowly grow as wind shear increases when the turbulence intensity is low (see the leftmost figure).

In summary, the importance of each variable is different. Further, interaction effects exist among the input variables, and the intensity of interaction effects are different. The interaction effects between wind speed and turbulence intensity and between wind speed and wind shear are substantial, whereas the pattern between turbulence intensity and wind shear is less obvious. These aspects call for a new sampling strategy that accounts for the different importance of each variable and the interactions among multiple environmental variables.

3.2. *Weighted additive multivariate kernel method.* This section devises a nonparametric approximation of the SIS optimal density in (3) which enables us to capture the data characteristics in wind turbine discussed in Section 3.1. In estimating q_{SIS}, the conditional POE, $s(\mathbf{x})$, needs to be estimated. Let $Z_i = \mathbb{I}(Y_i > l)$, that is, $Z_i = \mathbb{I}(g(\mathbf{X}_i, \epsilon) > l)$. We have

$$(4) \qquad\qquad Z_i | \mathbf{X}_i = \mathbf{x} \sim \text{Bernoulli}(s(\mathbf{x})).$$

It implies that $s(\mathbf{x})$ can be viewed as conditional expectation, $E[Z_i | \mathbf{X}_i = \mathbf{x}]$ which allows us to employ the kernel regression (Chen and Choe (2019), Nadaraya (1964)).

For the univariate input, x, the nonparametric estimator of $s(x)$ becomes

$$\hat{s}(x) = \frac{\sum_{i=1}^{n} K(\frac{x - X_i}{h}) Z_i}{\sum_{i=1}^{n} K(\frac{x - X_i}{h})},$$

where (X_i, Z_i), $i = 1, 2, \ldots, n$, are the samples for the kernel regression, $K(\cdot)$ is a univariate kernel function and h is the smoothing bandwidth. If $\mathbf{X}$ is a d-dimensional multivariate input
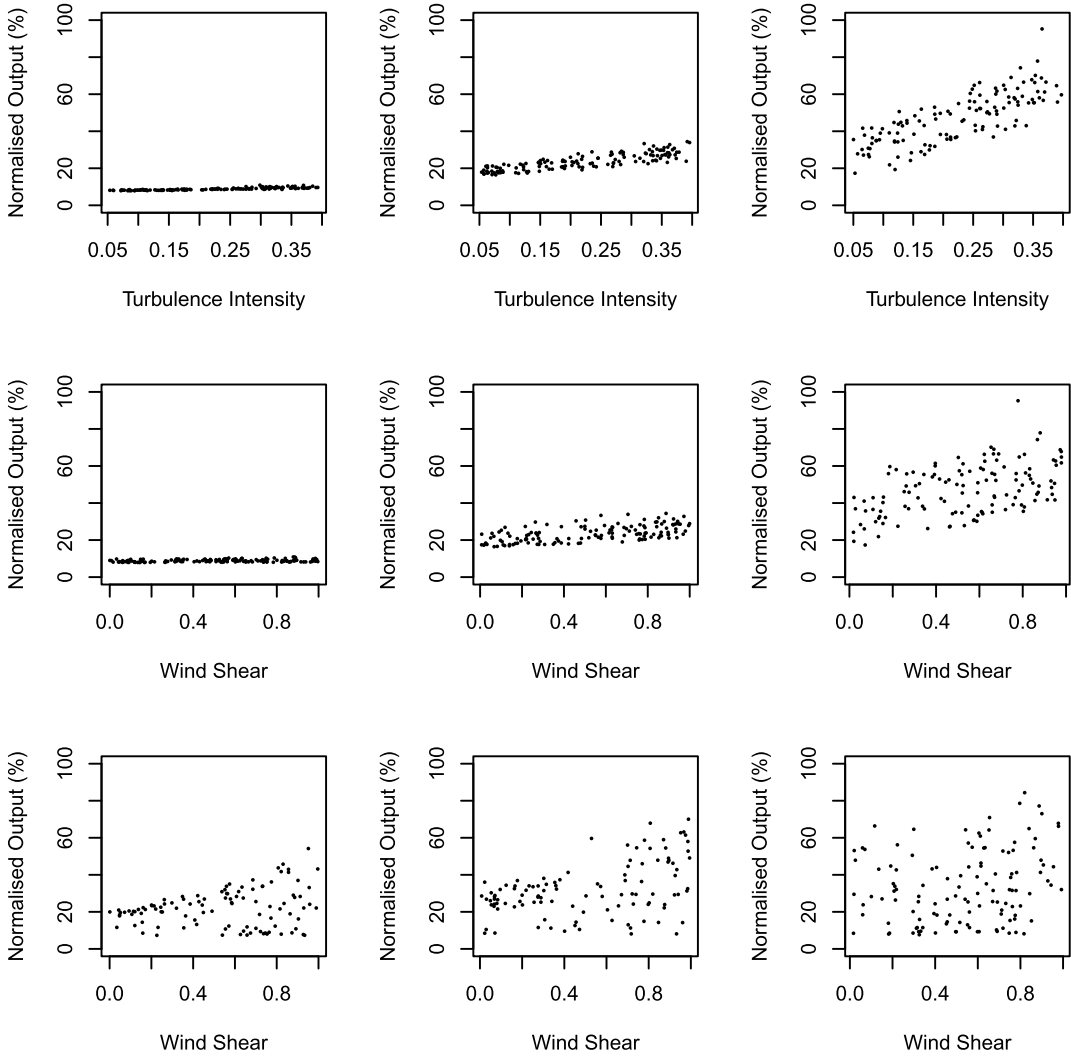
FIG. 2. *Scatter plots of the blade tip deflection vs. environmental factors under specific wind speed or turbulence intensity. Top panels:* $4.1 < V < 4.3, 11.1 < V < 11.3, 23.1 < V < 23.3$. *Middle panels:* $4.1 < V < 4.3$, $11.1 < V < 11.3, 23.1 < V < 23.3$. *Bottom panels:* $0.097 < I < 0.1, 0.247 < I < 0.25, 0.347 < I < 0.35$.

vector, a multivariate kernel function $K_d(\cdot)$ is required. Then, the estimator at a point $\mathbf{x} = (x_1, \ldots, x_d)$ is defined as

$$
(5) \qquad \hat{s}(x) = \frac{\sum_{i=1}^{n} K_d(\frac{x_1 - X_{i1}}{h_1}, \ldots, \frac{x_d - X_{id}}{h_d}) Z_i}{\sum_{i=1}^{n} K_d(\frac{x_1 - X_{i1}}{h_1}, \ldots, \frac{x_d - X_{id}}{h_d})},
$$

where $(X_{i1}, \ldots, X_{id}, Z_i)$, $i = 1, 2, \ldots, n$, is the ith observation and $h_1, \ldots, h_d$ are the bandwidths for input variables.

The key is how to construct the multivariate kernel function, $K_d(\cdot)$. In general, for a d variable kernel regression estimator, the convergence rate will be $O(h^2) + O_P(\sqrt{\frac{1}{nh^d}})$ (Fan and Gijbels (2018), Wasserman (2006)), and under the optimal choice of h, the rate is $O_P(n^{-\frac{2}{4+d}})$. The simplest model for $K_d(\cdot)$ in (5) is to use an additive kernel that is an addition of d univariate kernels. The convergence rate of the additive model is fastest when the true model has no interaction effects. However, it potentially has a large bias because it does not account for interactions among input variables.

To capture interaction effects, one can consider the multiplicative kernel (or full model), a product of d univariate kernels. It will have the least asymptotic bias. However, it has the slowest convergence rate, resulting in a data sparsity problem when we are unable to get a sufficiently large amount of data in d dimensions. Noting that the fundamental goal of IS is to reduce the computational burden, using the pure multiplicative kernel is not aligned well with our main objective. Moreover, bandwidth selection is very important for the kernel regression, especially when the sample size is small, but this procedure is challenging with the full multiplicative kernel for large d. For the well-known leave-one-out cross-validation (CV) method (Li and Racine (2007)), the bandwidth candidate set gets harder to specify properly, and the set becomes very large, as d increases. Another approach is based on the asymptotic mean integrated squared error (AMISE) (to be detailed in Section 3.3). But it requires numerical integration over d-dimension, so it is unstable and computationally inefficient when d is large.

To balance scalability and complexity, the additive multivariate kernel (AMK) is proposed in Lee et al. (2015) where the estimator of the conditional mean function, $\mathbf{m}(x) := E[Y|\mathbf{X} = \mathbf{x}]$, defined as

$$(6) \qquad \hat{\mathbf{m}}(\mathbf{x}) = \frac{1}{k}\big[\tilde{m}(\mathbf{x}^{(1)}) + \cdots + \tilde{m}(\mathbf{x}^{(K)}))\big].$$

Here, $\tilde{m}(\cdot)$ denotes the Nadaraya–Watson (N–W) kernel estimator, and $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(K)}$ are K subdimensional projections of $\mathbf{x}$. For example, for a d-dimensional input vector, $\mathbf{x} = (x_1, \ldots, x_d)$ ($d \geq 3$), we can let $\mathbf{x}^{(1)} = (x_1, x_2)$ which is a projection of $\mathbf{x}$ on the first two dimensions. Other $\mathbf{x}^{(k)}$'s can be similarly defined with a subset of input variables.

Kernel estimators in AMK are equally weighted. However, as discussed in Section 3.1, input variables and their interactions affect the output differently. For the blade deflection, wind speed and turbulence intensity and their interactions appear to be more important than wind shear and its interaction with other two variables. Noting that more attention needs to be paid to more influential variables, in order to identify important input area, we propose a weighted AMK (WAMK) estimator that assigns different weights to each kernel,

$$(7) \qquad \hat{s}(\mathbf{x}) = \sum_k w_k \tilde{s}(\mathbf{x}^{(k)}),$$

where w_k denotes the weight for the kth subkernel, $\tilde{s}(\mathbf{x}^{(k)})$, such that $\sum_k w_k = 1$.

In deciding the subkernels, either bivariate or trivariate kernels can be considered. However, fitting the trivariate subkernel to define the tail probability faces the aforementioned data sparsity issue and computational problem in the bandwidth selection. Apparently, the trivariate kernel requires a larger dataset to provide a reasonably good fit for estimating a tail probability, negating the benefit of IS. In terms of the bandwidth selection, for example, let us consider $d = 4$. With trivariate subkernels there are four subkernels in total; each has three bandwidths. Optimizing 12 bandwidths is not trivial. Because it requires numerical integration over triple variables, finding the bandwidths that minimize the AMISE is practically impossible. Also, applying the CV procedure to 12 bandwidths adds nonnegligible computational overhead. This problem becomes more challenging, as d gets larger.

Most importantly, three (or higher)-factor interactions are weak in many engineering applications. Therefore, we assume $g(\cdot)$ has strong two-factor multiplicative interactions among input variables and weak (or no) higher order interactions and define $\tilde{s}(\mathbf{x}^{(k)})$ using the bivariate kernel. Our implementation results with numerical examples in Section 4 demonstrate that our WAMK estimator with the bivariate kernels not only captures two-factor interactions but also performs well, even when higher-order interactions exist.

Suppose that $\mathbf{x}^{(k)}$ consists of two input variables, x_p and x_q, that is, $\mathbf{x}^{(k)} = (x_p, x_q)$. Let $\tilde{s}(x_p, x_q)$ denote a bivariate N–W estimator, defined as

$$(8) \qquad \tilde{s}(x_p, x_q) = \frac{\sum_{i=1}^{n} K_2(\frac{x_p - X_{pi}}{h_p}, \frac{x_q - X_{qi}}{h_q}) Z_i}{\sum_{i=1}^{n} K_2(\frac{x_p - X_{pi}}{h_p}, \frac{x_q - X_{qi}}{h_q})},$$

where X_{pi} and X_{qi}, respectively, denote the ith samples of x_p and x_q. The subscript "2" in $K_2(\cdot)$ implies the bivariate multiplicative kernel, that is,

$$K_2(x_p, x_q) = K(x_p)K(x_q).$$

We call $\tilde{s}(x_p, x_q)$ a subestimator of $s(\mathbf{x})$. In our implementation, $K_2(x_p, x_q)$ is chosen to be a product of two univariate Gaussian kernels for x_p and x_q.

Then, the WAMK estimator, $\hat{s}(\mathbf{x})$, in (7) can be expressed as

$$(9) \qquad \hat{s}(\mathbf{x}) = \sum_{1 \le p < q \le d} w_{pq} \tilde{s}(x_p, x_q),$$

for $d \ge 2$, where w_{pq} is the weight assigned to the subestimator, $\tilde{s}(x_p, x_q)$.

Next, we need to decide the weight for each subestimator. Recall that Z_i in (4) is a Bernoulli random variable with the success probability, $s(\mathbf{x})$. To measure the goodness of $\tilde{s}(x_p, x_q)$ for $s(\mathbf{x})$, we use the cross-entropy error function which is widely used in binary classifications (Bishop (2006)). The error function for the subestimator, $\tilde{s}(x_p, x_q)$, can be defined as

$$(10) \qquad e(p, q) = -\sum_{i=1}^{n} \{ Z_i \ln(\tilde{s}(X_{pi}, X_{qi})) + (1 - Z_i) \ln(1 - \tilde{s}(X_{pi}, X_{qi})) \}.$$

Note that the error function, $e(p, q)$, gets smaller as the estimation performance of $\tilde{s}(x_p, x_q)$ for Z_i gets stronger. Therefore, the smaller the error function of a subestimator, the greater the weight should be assigned to it. Thus, we propose to define the weight as

$$(11) \qquad w_{pg} = \frac{1/e(p, q)}{\sum_{1 \le p' < q' \le d} 1/e(p', q')}.$$

Admittedly, our choice of weights in (10)–(11) may not provide correct weights (e.g., consistent weights), even when the true $s(\mathbf{x})$ takes the form of $\sum_{p,q} w_{p,q} s(x_p, x_q)$. However, our choice is based on the "likelihood" of the Bernoulli distribution. The cross-entropy error $e(p, q)$ in (10) is the negative of log-likelihood for the bivariate case and quantifies the goodness of fit of $\tilde{s}(x_p, x_q)$. Our setting with multiple variables for $d \ge 3$ is more complicated because Z_i is a function of all input variables (not only x_p and x_q). As such, the cross-entropy error is not exactly the negative log-likelihood. Further, we set $w_{pg} \propto 1/e(p, q)$ in (11) which complicates the derivation of theoretical properties of w_{pg}. Regardless, it carries the fundamental idea of likelihood, measuring the fit of each subkernel $\tilde{s}(\cdot)$.

3.3. *Bandwidth selection.* Bandwidth selection is critical in the multivariate kernel regression. In our wind turbine case we need to choose six bandwidths with three variables, (three subestimators, each with two bandwidths). Optimizing six bandwidths is challenging. The well-known leave-one-out CV approach (Fan and Yim (2004)) is computationally intensive when the candidate set is large. Further, it needs to set a candidate set a priori such that the candidate set includes good bandwidths. If not, $\tilde{s}(x_p, x_q)$ would possibly deviate from true $s(x_p, x_q)$.

In this study we employ the AMISE-based approach. Noting that each subestimator, $\tilde{s}(x_p, x_q)$, is an estimator of $s(x_p, x_q) \equiv P(Y > l | X_p = x_p, X_q = x_q)$, we separately

choose bandwidths for each subestimator by minimizing the corresponding AMISE. Without loss of generality, let us consider the general form of s-dimensional multiplicative kernel $K_s(x_1, \ldots, x_s) = \prod_{i=1}^s K(x_i)$ for the subestimator, $\tilde{s}(\mathbf{x}^{(k)})$, with $\mathbf{x}^{(k)} = (x_1, \ldots, x_s)$; in the wind turbine case study, $K_s(\cdot)$ is a bivariate kernel, $K_2(\cdot)$. The mean integrated squared error (MISE) of $\tilde{s}(\mathbf{x}^{(k)})$ is defined as

$$\text{MISE}(\tilde{s}(\mathbf{x}^{(k)})) = \int \text{MSE}(\tilde{s}(\mathbf{x}^{(k)})) \, d\mathbf{x}^{(k)}$$
$$= \int E\big[(\tilde{s}(\mathbf{x}^{(k)}) - s(\mathbf{x}^{(k)}))^2\big] \, d\mathbf{x}^{(k)}.$$

Under some regularity conditions the asymptotic distribution of $\tilde{s}(\mathbf{x}^{(k)})$ satisfies the following (Bierens (1987), Hansen (2009), Li and Racine (2007)):

$$(12) \quad \sqrt{n|H|}\left(\tilde{s}(\mathbf{x}^{(k)}) - s(\mathbf{x}^{(k)}) - \mu_2(K) \sum_{j=1}^s B_j(\mathbf{x}^{(k)})h_j^2\right) \quad \Rightarrow \quad N\left(0, \frac{\sigma^2(\mathbf{x}^{(k)})R(K_s)}{\tilde{h}(\mathbf{x}^{(k)})}\right).$$

Below, we explain notations in (12). First, $|H| = \prod_{i=1}^s h_i$, and $\mu_2(K) = \int x^2 K(x) \, dx$ which becomes one when K is the Gaussian kernel. Further, we have

$$\sigma^2(\mathbf{x}^{(k)}) = \text{var}(Z|\mathbf{X}^{(k)} = \mathbf{x}^{(k)}) = \tilde{s}(\mathbf{x}^{(k)})(1 - \tilde{s}(\mathbf{x}^{(k)})),$$

because $Z|\mathbf{X}^{(k)} = \mathbf{x}^{(k)}$ obeys the Bernoulli distribution with probability $\tilde{s}(\mathbf{x}^{(k)})$. Moreover,

$$R(K_s) = \int K_s(\mathbf{x}^{(k)})^2 \, d\mathbf{x}^{(k)},$$

$$(13)$$

$$B_j(\mathbf{x}^{(k)}) = \tilde{h}(\mathbf{x}^{(k)})^{-1} \frac{\partial \tilde{s}(\mathbf{x}^{(k)})}{\partial x_j} \frac{\partial \tilde{h}(\mathbf{x}^{(k)})}{\partial x_j} + \frac{1}{2} \frac{\partial^2 \tilde{s}(\mathbf{x}^{(k)})}{\partial^2 x_j^2},$$

where $\tilde{h}(\mathbf{x}^{(k)})$ denotes the limiting pdf of $\tilde{f}(\mathbf{x}^{(k)}) = \frac{1}{n|H|} \sum_{i=1}^n K(H^{-1}(\mathbf{X}_i^{(k)} - \mathbf{x}^{(k)}))$ with $\mathbf{X}_i^{(k)}$ being the kth input sample of $\mathbf{X}^{(k)}$, that is,

$$(14) \qquad \tilde{f}(\mathbf{x}^{(k)}) = \frac{1}{n|H|} \sum_{i=1}^n K(H^{-1}(\mathbf{X}_i^{(k)} - \mathbf{x}^{(k)})) \to \tilde{h}(\mathbf{x}^{(k)}).$$

Here, the first term in $B_j(\mathbf{x}^{(k)})$ in (13) is called design bias, implying the bias from the design/covariates distribution, $\tilde{h}$ should be the true pdf of the covariates and $\tilde{f}$ happens to be a good kernel density estimator.

From (12), as n increases, the MSE of $\tilde{s}(\mathbf{x}^{(k)})$ becomes

$$\text{MSE}(\tilde{s}(\mathbf{x}^{(k)})) \approx \mu_2(K)^2 \left(\sum_{j=1}^s h_j^2 B_j(\mathbf{x}^{(k)})\right)^2 + \frac{\sigma^2(\mathbf{x}^{(k)})R(K_s)}{n|H|\tilde{h}(\mathbf{x}^{(k)})},$$

which leads to the AMISE of $\tilde{s}(\mathbf{x}^{(k)})$, as follows:

$$\text{AMISE}(\tilde{s}(\mathbf{x}^{(k)}))$$

$$(15)$$

$$\approx \mu_2(K)^2 \int \left(\sum_{j=1}^s B_j(\mathbf{x}^{(k)})h_j^2\right)^2 d\mathbf{x}^{(k)} + \frac{R(K_s)}{n \prod_{i=1}^s h_i} \int \frac{\sigma^2(\mathbf{x}^{(k)})}{\tilde{h}(\mathbf{x}^{(k)})} \, d\mathbf{x}^{(k)}.$$

Algorithm 1 Bandwidth selection for the subestimator $\tilde{s}(x_p, x_q)$

1: **(Initialization)** Initialize the bandwidth for each variable by considering a univariate kernel regression with the corresponding variable.
2: **while** Convergence condition is not met **do**
3:    Use the current bandwidths to get $\tilde{s}(\mathbf{x}^{(k)})$.
4:    Update $B_j(\mathbf{x}^{(k)})$ in (13) with $\tilde{s}(\mathbf{x}^{(k)})$ and estimate $\sigma^2(\mathbf{x}^{(k)})$ with $\tilde{s}(\mathbf{x}^{(k)})(1 - \tilde{s}(\mathbf{x}^{(k)}))$.
5:    Optimize AMISE in (16) to obtain new bandwidths.
6:    Set the new bandwidths as current bandwidths
7: **end while**
8: **Output:** h_p, h_q

For the bivariate kernel, AMISE($\hat{s}(\mathbf{x}^{(k)})$) becomes

$$\text{AMISE}(\tilde{s}(\mathbf{x}^{(k)}))$$

$$(16) \qquad \approx \int \left(B_1(\mathbf{x}^{(k)})h_1^2 + B_2(\mathbf{x}^{(k)})h_2^2\right)^2 d\mathbf{x}^{(k)} + \frac{R(K_s)}{nh_1h_2}\int \frac{\sigma^2(\mathbf{x}^{(k)})}{\tilde{h}(\mathbf{x}^{(k)})}\, d\mathbf{x}^{(k)}$$

$$= \int \left(B_1(\mathbf{x}^{(k)})h_1^2 + B_2(\mathbf{x}^{(k)})h_2^2\right)^2 d\mathbf{x}^{(k)} + \frac{R(K_s)}{nh_1h_2}\int \frac{\tilde{s}(\mathbf{x}^{(k)})(1 - \tilde{s}(\mathbf{x}^{(k)}))}{\tilde{h}(\mathbf{x}^{(k)})}\, d\mathbf{x}^{(k)}.$$

Then, the bandwidths can be determined by minimizing AMISE($\hat{s}(\mathbf{x}^{(k)})$). Here, because $\tilde{h}(\mathbf{x}^{(k)})$ is unknown, we replace it with the sampling density, $\tilde{f}(\mathbf{x}^{(k)})$, defined in (14), in our implementation.

If $\tilde{s}(\mathbf{x}^{(k)})$ is a univariate kernel estimator, its bandwidth can be chosen by directly minimizing AMISE, using existing numerical optimization methods. However, for bivariate cases, it takes significant time to get the optimal bandwidths with existing optimization solvers. We note that AMISE($\hat{s}(\mathbf{x}^{(k)})$) in (16) (or (15) for general multivarite kernels) involves $\tilde{s}(\mathbf{x}^{(k)})$ ($B_j(\mathbf{x}^{(k)})$, defined in (13), also requires $\tilde{s}(\mathbf{x}^{(k)})$). If $\tilde{s}(\mathbf{x}^{(k)})$ is known, h_1 and h_2 can be obtained by minimizing AMISE in (16). But $\tilde{s}(\mathbf{x}^{(k)})$ is unknown. In fact, to define $\tilde{s}(\mathbf{x}^{(k)})$, we also need the bandwidths.

To address these difficulties, we choose bandwidths using an iterative algorithm. The idea is to estimate $\tilde{s}(\mathbf{x}^{(k)})$ with the current incumbent bandwidth values and solve (16) to get new h_1 and h_2. Then, we update $\tilde{s}(\mathbf{x}^{(k)})$ with new h_1 and h_2. We iterate this procedure until the change of h_1 and h_2 in consecutive iterations becomes insignificant. To initialize the iteration, we obtain the initial bandwidths by considering the univariate kernel regression with each variable. In our implementation, we observe that a couple of iterations is enough; we repeat the iteration three times. Algorithm 1 describes the bandwidth selection procedure for the bivariate subestimator $\tilde{s}(\mathbf{x}^{(k)})$ with $\mathbf{x}^{(k)} = (x_p, x_q)$.

We illustrate the proposed bandwidth selection result using the numerical example presented in Section 4.1. To show how the errors change as a function of two bandwidths, we set $X_3 = 0$. Thus, we have $Y \sim N(\mu_1(X_1, X_2, 0), 1)$, and the true conditional failure probability becomes a function of X_1 and X_2, where $\mu_1(\cdot)$ is defined in (21). To construct the bivariate kernel, we get 1000 samples of X_1 and X_2 from $U(-5, 5)$ and then generate corresponding response. Then we evaluate each pair of bandwidths (h_1, h_2) using the average of the squared errors at multiple grid points. Specifically, we compute MSE between the kernel estimate and the true conditional failure probability at 10,000 points of $(x_1, x_2) \in \{(-4.9, -4.8, -4.7, \ldots, 5) \times (-4.9, -4.8, -4.7, \ldots, 5)\}$. Figure 3 depicts contour plots of MSE vs. (h_1, h_2) from the proposed AMISE-based procedure. The left panel is the case where the initial bandwidth is obtained from the kernel regression with each variable,
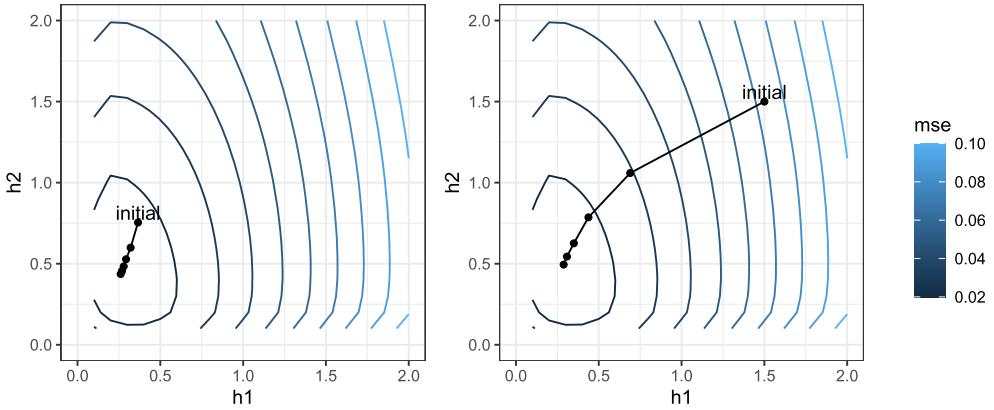
FIG. 3. *Contour plots of MSE vs. bandwidths (h_1, h_2). On the left panel the initial bandwidths are selected with the univariate AMISE. On the right panel the initial bandwidths are chosen arbitrarily at $(1.5, 1.5)$. The line with black solid circles in each figure shows the trajectory of the bandwidths using Algorithm 1.*

while the right panel shows the trajectory from an arbitrarily selected, somewhat bad, initial point. In both cases the bandwidths converge well within a small number of iterations.

While our procedure does not guarantee the optimal bandwidth choice, it appears to generate a sufficiently good one. Unlike CV, our AMISE-based procedure provides a solution directly without defining candidate sets. It converges to reasonably good choices in a small number of iterations in most cases, and thus it is computationally more efficient than CV.

3.4. *Sampling from IS density.* Once the bandwidths are determined, we get the WAMK estimator, $\hat{s}(\mathbf{x})$, in (9). We replace $s(\mathbf{x})$ with $\hat{s}(\mathbf{x})$ to approximate the optimal SIS density $q_{\mathrm{SIS}}(\mathbf{x})$ in (3). The resulting nonparametric SIS density, denoted by $q_{n\mathrm{SIS}}$, is

$$(17) \qquad q_{n\mathrm{SIS}}(\mathbf{x}) = \frac{1}{C'_q} f(\mathbf{x}) \sqrt{\hat{s}(\mathbf{x})},$$

where C'_q is a normalizing constant. This nonparametric SIS density provides unbiased estimation, because $q_{n\mathrm{SIS}}(\mathbf{x}) > 0$ whenever $f(\mathbf{x})s(\mathbf{x}) > 0$.

To sample the input from the nonparametric SIS density, we use the acceptance-rejection algorithm, which can be implemented without knowing the normalizing constant, C'_q, in (17) (Choe, Byon and Chen (2015)). Since $f(\mathbf{x})\sqrt{\hat{s}(\mathbf{x})} \leq f(\mathbf{x})$, we simply use $f(\mathbf{x})$ as the proposal density. The algorithm is described in Algorithm 2. We repeat this procedure until a prespecified number of samples are accepted.

3.5. *Sequential procedure and implementation guideline.* To fully utilize samples, we take a sequential procedure in implementing WAMK-SIS. That is, after we get new samples at each iteration, we update the kernel estimator, $\hat{s}(\mathbf{x})$, using the samples obtained up to the current iteration to better estimate $s(\mathbf{x})$ and use it at the next iteration. Let $\hat{s}^{(t)}(\mathbf{x})$ denote the WAMK estimator used at the tth iteration. From (17) the nonparametric SIS density at the

---

Algorithm 2 Sampling from IS density

1: Sample $\mathbf{x}$ from the input density, f.
2: Sample u from the uniform distribution $U(0, f(\mathbf{x}))$.
3: $u \leq f(\mathbf{x})\sqrt{\hat{s}(\mathbf{x})}$, accept $\mathbf{x}$ as a sample. Otherwise, reject the value of $\mathbf{x}$.
4: **Output: x**

Algorithm 3 WAMK-SIS procedure

1: **Input:** Set the iteration number m, sample size $N^{(t)}$ for $t = 0, \ldots, m$, $q^{(0)}$, where $N^{(0)}$ and $q^{(0)}$ (e.g., uniform distribution) denote the sample size and sampling density of initial samples, respectively.

2: **(Initialization)** Sample input value $X_i^{(0)}$ from density $q^{(0)}$, and run simulation to get output value $Y_i^{(0)}$, $i = 1, \ldots, N^{(0)}$. Set $t = 1$.

3: **while** $t \leq m$ **do**

4:     Update bandwidths for subestimators using Algorithm 1 with the aggregated data sampled up to the previous iteration.

5:     Obtain each subestimator defined in (8) with the aggregated data.

6:     Obtain the weight for each subestimator using (10)–(11).

7:     Obtain the conditional POE estimator in (7), and set it as $\hat{s}^{(t)}(x)$.

8:     Obtain $q_{n\text{SIS}}^{(t)}(\mathbf{x})$ in (18).

9:     Sample input value $X_i^{(t)}$ from density $q_{n\text{SIS}}^{(t)}(\mathbf{x})$ using Algorithm 2, and run simulation to get output value $Y_i^{(t)}$, $i = 1, \ldots, N^{(t)}$.

10:    If $t < m$, set $t = t + 1$; otherwise, estimate the POE using the estimator in (19) or (20).

11: **end while**

12: **Output:** $\hat{P}_{\text{SIS}}$ or $\hat{P}_{\text{SIS}}^{(a)}$.

---

tth iteration becomes

$$(18) \qquad q_{n\text{SIS}}^{(t)}(\mathbf{x}) = \frac{1}{C_q^{(t)}} f(\mathbf{x})\sqrt{\hat{s}^{(t)}(\mathbf{x})}.$$

Then, the failure probability can be estimated by

$$
\begin{aligned}
\hat{P}_{\text{SIS}} &= \frac{1}{m}\sum_{t=1}^{m}\frac{1}{N^{(t)}}\sum_{i=1}^{N^{(t)}}\mathbb{I}(Y_i^{(t)} > l)\frac{f(\mathbf{X}_i^{(t)})}{q_{n\text{SIS}}^{(t)}(\mathbf{X}_i^{(t)})} \\
(19) \qquad &= \frac{1}{m}\sum_{t=1}^{m}\frac{1}{N^{(t)}}\sum_{i=1}^{N^{(t)}}\mathbb{I}(Y_i^{(t)} > l)\frac{C_q^{(t)}}{\sqrt{\hat{s}^{(t)}(\mathbf{X}_i^{(t)})}},
\end{aligned}
$$

where m is the number of iterations, $N^{(t)}$ is the sample size for iteration t and $(X_i^{(t)}, Y_i^{(t)})$, $i = 1, \ldots, N^{(t)}$, denote the samples drawn at iteration t. The total sample size is $N_T = \sum_{t=1}^{m} N^{(t)}$.

With the multivariate input the normalizing constant $C_q^{(t)}$ in (19) is often numerically unstable, in particular, when the input dimension, d, is 3 or higher. In such cases we can instead use the selfnormalized estimator (Owen (2013)), $\hat{P}_{\text{SIS}}^{(a)}$, defined as

$$(20) \qquad \hat{P}_{\text{SIS}}^{(a)} = \frac{1}{m}\sum_{t=1}^{m}\sum_{i=1}^{N^{(t)}}\frac{\mathbb{I}(Y_i^{(t)} > l)f(\mathbf{X}_i^{(t)})/q^{(t)}(\mathbf{X}_i^{(t)})}{\sum_{i=1}^{N^{(t)}}f(\mathbf{X}_i^{(t)})/q^{(t)}(\mathbf{X}_i^{(t)})}.$$

In our implementation we use the selfnormalized estimator in (20). We summarize the WAMK-SIS procedure in Algorithm 3.

4. Numerical examples. We evaluate the performance of WAMK-SIS using three numerical examples with different features. We also compare its performance with alternative approaches. First, we consider CE-SIS where GMM is employed to approximate the optimal SIS density (Cao and Choe (2019)). Additionally, to investigate how much weighted kernels contribute to the efficiency, we compare the performance of WAMK-SIS with its unweighted

version, called AMK-SIS. The AMK-SIS uses equal weights in all subestimators for defining the conditional POE estimator in (7).

4.1. *Example* 1: *Three-dimensional input with weak high-order interaction effect.* We slightly modify the example presented in Cao and Choe (2019), Ackley (2012) and consider the following data-generating structure:

$$\mathbf{X} \sim N(0, I), Y|\mathbf{X} \sim N(\mu_1(\mathbf{X}), 1),$$

where $\mathbf{X} = (X_1.X_2, X_3)$ is a three-dimensional input variable and

$$\mu_1(\mathbf{X}) = 65 - 40 \exp\left(-0.2\sqrt{\frac{X_1^2 + X_2^2}{2}}\right) - 20 \exp(-0.2|X_1|)$$

$$(21) \qquad - 5 \exp\left(-0.2\sqrt{\frac{X_2^2 + X_3^2}{2}}\right) - \sum_{1 \leq i < j \leq 3} \exp(\cos(2\pi X_i X_j))$$

$$- \exp(\cos(2\pi X_1 X_2 X_3)).$$

With $l = 17.90$, we obtain the target failure probability, $P_t = 0.01$, with a large-scale Monte Carlo simulation with 10^7 runs. We evaluate the performance of each method using relative ratio (RR), defined as $\text{RR} = N_T/N_{\text{CMC}}$, where N_{CMC} is the number of CMC simulation replications required to achieve the same standard error (SE) of each method (Choe, Byon and Chen (2015)), defined as

$$N_{\text{CMC}} = \frac{P_t(1 - P_t)}{SE^2}.$$

We first examine the effectiveness of WAMK among different choices of kernel regression. Because (21) includes the three-factor interaction, the simplest additive model has a large bias in estimating $s(\mathbf{x})$. The full multiplicative model with the trivariate kernel has the least asymptotic bias but has the slow convergence rate. The proposed WAMK is an intermediate version of these two extreme cases.

Table 1 compares the results with three different kernels, including SIS with weighted additive model (WAK-SIS), three-factor multiplicative model (MK-SIS) and WAMK-SIS. In WAK-SIS the weights for individual univariate kernels are decided using the cross-entropy error functions, similar to (10)–(11). In all cases, Gaussian kernel is employed. We conduct experiments with different computational budgets, shown in the second column. We set $m = 5$ at each experiment. The sample size in the second column is the total number of simulation runs in the five iterations. When the sample size is small, MK-SIS generates poor results. It even generates a biased POE estimation with 500 sample size, because $\hat{s}(\mathbf{x})$ becomes zero (or close to zero) for some input regions and MK-SIS possibly misses some important areas. Its sample errors and RR are also higher than WAMK-SIS when the sample size is 250 and 1000, implying that the full model is unstable. With larger sample sizes, the performance of MK-SIS becomes better than WAMK-SIS.

On the other hand, WAK-SIS consistently produces worse results than WAMK-SIS, due to its large bias in estimating $s(\mathbf{x})$. Interestingly, despite its biased estimation of $s(\mathbf{x})$, the sample mean from the additive model, shown in the third column of Table 1, is always around the true failure probability, 0.01, so it appears to provide an unbiased POE estimation. It implies that the support of WAK-SIS density covers the support of $p(\mathbf{x})s(\mathbf{x})$. Overall, the results demonstrate that WAMK-SIS balances between the full model and additive model. Its performance is the most stable among the three kernel choices.

Finally, we compare our approach with AMK-SIS and CE-SIS. We use $N^{(0)} = 1000$ and $N^{(t)} = 1000$, $t = 1, \ldots, 5$. Table 2 summarizes the results from 100 experiments of each

TABLE 1
Comparison results between different kernel regressions for Example 1

Kernel type	Sample Size	Mean	SE	RR
MK-SIS	250	0.013445	0.008979	203.69%
WAMK-SIS	250	0.009927	0.003828	37.00%
WAK-SIS	250	0.010853	0.005336	71.89%
MK-SIS	1000	0.010609	0.001794	32.52%
WAMK-SIS	1000	0.010039	0.001657	27.73%
WAK-SIS	1000	0.010035	0.002511	63.69%
MK-SIS	2500	0.010028	0.000857	18.56%
WAMK-SIS	2500	0.009797	0.001074	29.73%
WAK-SIS	2500	0.009772	0.001467	54.36%
MK-SIS	5000	0.010142	0.000617	19.24%
WAMK-SIS	5000	0.010025	0.000769	29.86%
WAK-SIS	5000	0.010075	0.000982	48.67%

method. All of the three IS methods show variance reduction results over CMC. Among them, WAMK-SIS achieves the smallest SE and RR. By allowing different weights for the three subestimators, WAMK-SIS outperforms AMK-SIS. The CE-SIS method determines weights for mixture densities, depending on the importance of component densities, but it does not show strong estimation performance. Interestingly, even AMK-SIS, which uses the equal weight, produces lower SE and RR than CE-SIS.

As a remark, recall that we discuss the limitation of the emulator-based approach for the tail probability estimation in Section 1. Another caveat in the emulator-based approach with multivariate input vectors is the difficulty in integrating the conditional probability. Suppose we obtain an emulator, say, $\tilde{P}(Y > l|\mathbf{X})$, for the conditional failure probability. To estimate $P(Y > l)$, we need to integrate the estimated conditional density over $\mathbf{X}$. However, numerical integration over three or higher dimensional input variables is unstable and takes a significant amount of time. We fit the GP in this example with the same sample size and numerically integrate it to calculate the tail probability. However, the resulting value was not even close to the target tail probability, 0.01. It indicates that the tail probability estimation is sensitive to the emulator quality and the accuracy of multidimensional numerical integration.

4.2. *Example* 2: *Four-dimensional input with weak high-order interaction effect.* We further consider a problem with four-dimensional input and three-way interaction effects,

$$(22) \qquad \mathbf{X} \sim N(0, I), \qquad Y|\mathbf{X} \sim N(\mu_2(\mathbf{X}), 1),$$

TABLE 2
Comparison results from 100 *experiments for Example* 1

Method	Mean	SE	RR
WAMK-SIS	0.010025	0.000769	29.86%
AMK-SIS	0.010032	0.000898	40.74%
CE-SIS	0.009925	0.001251	78.99%

TABLE 3
Comparison results from 100 *experiments for Example* 2

Method	Mean	SE	RR
WAMK-SIS	0.009940	0.000791	31.64%
AMK-SIS	0.010085	0.000929	43.56%
CE-SIS	0.009802	0.002867	415.07%

where $\mathbf{X} = (X_1.X_2, X_3, X_4)$ is a four-dimensional input variable and

(23)
$$\mu_2(\mathbf{X}) = 65 - 40\exp\left(-0.2\sqrt{\frac{X_1^2 + X_2^2}{2}}\right) - 20\exp(-0.2|X_1|)$$
$$- 5\exp\left(-0.2\sqrt{\frac{X_2^2 + X_3^2 + X_4^2}{3}}\right) - \sum_{1 \le i < j \le 3}\exp(\cos(2\pi X_i X_j)).$$

We set $l = 18.99$, corresponding to the failure probability $P_t = 0.01$. We use the same setting as in Section 4.1. In this example, three-way interaction exists among X_2, X_3 and X_4 in addition to the two-factor interactions characterized by the first and last terms in $\mu_2(\mathbf{X})$.

Table 3 shows that WAMK-SIS provides the lowest SE and RR among the three methods, demonstrating that it handles higher-order interaction effects well. CE-SIS performs poorly and even underperforms CMC. This agrees with the results presented in Cao and Choe (2019) where the performance of CE-SIS gets quickly deteriorated, as the input dimension increases.

Figure 4 depicts the trajectories of weights for the subestimators in WAMK-SIS. From the data generating structure in (22)–(23), the interaction effects between X_1 and X_2 are the most significant, and X_1 appears to be the most influential input variable. Accordingly, the bivariate subestimator with X_1 and X_2, $\tilde{s}(x_1, x_2)$, gets the highest weight throughout iterations. The reason the two subestimators, $\tilde{s}(x_1, x_3)$ and $\tilde{s}(x_1, x_4)$, get nonnegligible weights is due to the importance of X_1. The other three subestimators get quite small weights. This result demonstrates that WAMK-SIS successfully captures the importance of each variable and intensity of interaction effects.

4.3. *Example* 3: *Four-dimensional input with strong high-order interaction effect.* Next, let us consider the example that exhibits strong interaction effects among four-dimensional
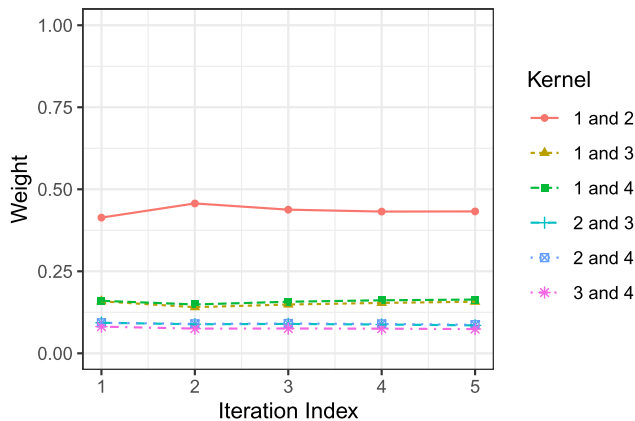

FIG. 4. *Example of weight trajectories of six subestimators in Example* 2.

Method	Mean	SE	RR
WAMK-SIS	0.010113	0.001083	59.29%
AMK-SIS	0.010000	0.001075	58.38%
CE-SIS	0.010089	0.001827	168.67%

input variables with the following complicated structure (Cao and Choe (2019)):

$$\mathbf{X} \sim N(0, I), \qquad Y|\mathbf{X} \sim N(\mu_3(\mathbf{X}), 1),$$

where $\mathbf{X} = (X_1. X_2, X_3, X_4)$ is a four-dimensional input and

$$\mu_3(\mathbf{X}) = 20\left(1 - \exp\left(-0.2\sqrt{\frac{1}{4}\|\mathbf{X}\|^2}\right)\right) + \left(\exp(1) - \exp\left(\frac{1}{4}\sum_{i=1}^{4}\cos(2\pi X_i)\right)\right).$$

We set the threshold at $l = 8.70$, corresponding to the failure probability $P_t = 0.01$. We use the same setup as in Examples 1 and 2.

In Table 4, WAMK-SIS still significantly outperforms CE-SIS in this example. WAMK-SIS and AMK-SIS provide comparable computational gains. This is because the conditional density of the response variable is symmetric, and all input variables equally affect the response. As a result, the weights assigned to all bivariate kernels in WAMK-SIS are almost equal. In this four-dimensional case, CE-SIS presents a worse performance than CMC. Overall, these results demonstrate the promising performance of WAMK-SIS in the presence of high-order (more than two) interaction effects.

5. Wind turbine case study.

5.1. *Problem setting.* As discussed in Section 3.1, we use wind speed, V, turbulence intensity, I and wind shear, S, as input variables, and the 10-minute maximum tip deflection, generated by the NREL simulators, as an output variable, Y. Specifically, we use TurbSim (version 1.50) (Jonkman (2009)) and FAST (version 7.01.00a-bjj) (Jonkman and Buhl Jr (2005)). To evaluate the wind turbine reliability, we are interested in estimating the probability of blade tip deflection exceeding a prespecified threshold l. We set $l = 2.06$ (corresponding to 25.4 % in the normalized value in Figures 1 and 2).

For the wind speed density, denoted as $f_V(x_1)$, we employ a truncated Rayleigh distribution on the interval [3, 25] (m/s) with the scale parameter, $10\sqrt{2/\pi}$, as suggested in Moriarty (2008). Here, 3 m/s is the smallest wind speed at which a turbine starts operation, whereas a turbine stops when wind speed exceeds 25 m/s to protect the turbine structure under strong gust. For turbulence intensity, I, TurbSim user's guide (Jonkman (2009)) provides several turbulence models describing the relationship between V and I. In our study we use Normal Turbulence Model (NTM) Class B, one of the most popular turbulence models. In this model the turbulence intensity is formulated as a function of wind speed. Specifically, we consider the conditional mean and variance of I, given V, as

(24)
$$\mu_I(V) \triangleq E[I|V] = 0.14(0.75V + 5.6)/V,$$
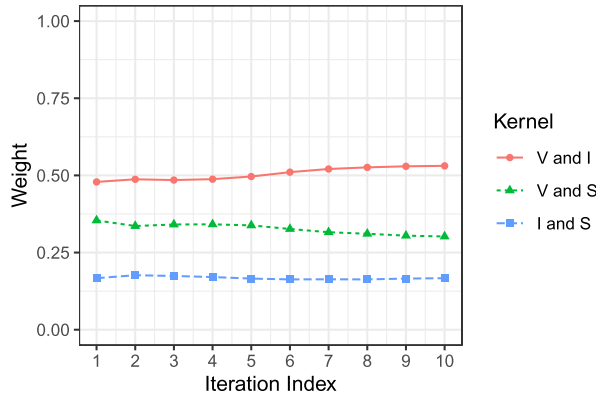$$\sigma_I \triangleq SD[I|V] = 0.05.$$

FIG. 5. *Example of weight trajectories of three subestimators in the wind turbine case study.*

Because I is positive, I conditioning on V is assumed to obey the log-normal distribution. From (24) the conditional distribution, $f_{I|V}(x_2)$, of turbulence intensity, given V, becomes

$$I|V \sim \text{Lognormal}\left(\log\left(\frac{\mu_I^2(V)}{\sqrt{\mu_I^2(V) + \sigma_I^2}}\right), \log\left(1 + \frac{\sigma_I^2}{\mu_I^2(V)}\right)\right).$$

Lastly, the density of wind shear also depends on the wind speed. TurbSim user guide does not provide the wind shear model. Thus, we use observational data from an actual wind turbine, available at Advance Metrology Lab (2020). We fit a linear regression model using the ordinary least squares method and get

$$\mu_S(V) \triangleq E[S|V] = 2.63 \times 10^{-4} V^3 - 1.09 \times 10^{-2} V^2$$
$$+ 1.285 \times 10^{-1} V - 1.32 \times 10^{-1},$$
$$\sigma_S(V) \triangleq SD[S|V] = 7.767 \times 10^{-5} V^3 - 3.43 \times 10^{-3} V^2$$
$$+ 3.4 \times 10^{-2} V + 1.3 \times 10^{-1}.$$

Then, the conditional distribution of wind shear, S, given V, with density $f_{S|V}(x_3)$, is assumed to be the normal distribution as

$$S|V \sim N\left(\mu_S(V), \sigma_S^2(V)\right).$$

Thus, we obtain the joint probability density function of three environmental variables as

$$f_{V,I,S}(x_1, x_2, x_3) = f_V(x_1) f_{I|V}(x_2) f_{S|I,V}(x_3)$$
$$= f_V(x_1) f_{I|V}(x_2) f_{S|V}(x_3).$$

5.2. *Implementation results.* We use $N^{(0)} = 600$ and $N^{(t)} = 200$, $t = 1, \dots, 10$. We first investigate the importance of each bivariate kernel in WAMK-SIS. Recall that among the three variables, wind speed appears to be the most influential factor (see Figure 1). Further, as observed in Figure 2, the interaction effects between wind speed and turbulence intensity (the top panel) are the most strong, followed by the interaction between wind speed and wind shear (the middle panel). The weight trajectories from WAMK-SIS, shown in Figure 5, reflect such different importance of each subestimator. The subestimator with wind speed and turbulence intensity gets the largest weight at around 50%, whereas the smallest weight is assigned to the subestimator with the turbulence intensity and wind shear.

Another interesting point is that, as iterations proceed and more data is collected, WAMK-SIS fine-tunes the weights so that the subestimator with (V, I) gets more weights, whereas the
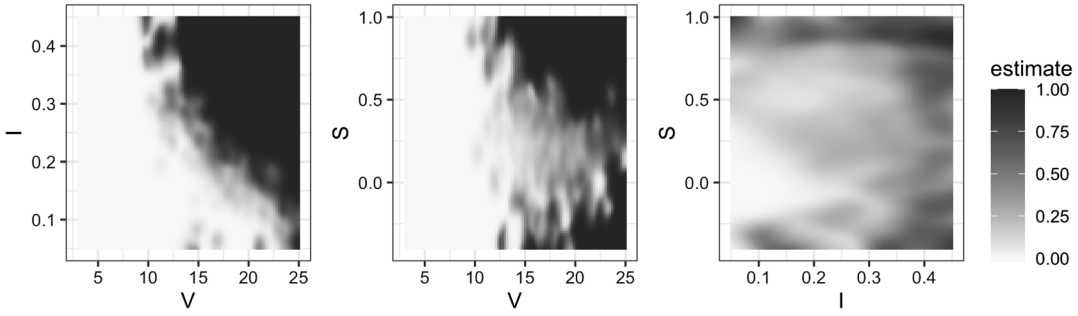
FIG. 6. $\tilde{s}(V, I), \tilde{s}(V, S)$ and $\tilde{s}(I, S)$ in wind turbine case study.

weight for the (V, S) subestimator gets slightly decreasing. This is somewhat different from Figure 4 of Example 2, where the weight trajectories do not exhibit any changing patterns. We believe the difference is because we use a larger sample size, 1000, at each iteration in the numerical example, so WAMK-SIS quickly learns the importance structure, even in early iterations. In the wind turbine case study, due to the computational burden in running the wind turbine simulator, we use a smaller sample size, 200, so WAMK-SIS learns proper weights slower than it does in the numerical example case. However, even in this case study, we can see that the weight allocation does not change substantially over iterations. It indicates that WAMK-SIS captures the importance of each variable and its interaction effects quickly from early iterations.

Figure 6 further depicts the three subestimators, each with two environmental conditions. The darker color is, the higher subestimator value is. It clearly demonstrates how the regression function $s(x_p, x_q)$ looks in each component and helps understand where failure events likely happen. This is a salient feature of the proposed approach with bivariate subkernels which cannot be achieved with the full three-factor multiplicative kernel. Note that the figures echo the results in Figure 5 and agree with our observations in Figure 2. We can see substantial interaction effects in (V, I) and (V, S) pairs. The leftmost figure for $\tilde{s}(V, I)$ suggests that, at mid speed range (e.g., at $V = 15$), the conditional POE becomes larger when the turbulence intensity gets higher, whereas at the low or high speed (e.g., at $V = 5$ or 25), the effect of turbulence intensity is almost negligible, and only wind speed largely affects the conditional POE.

A similar pattern is observed in $\tilde{s}(V, S)$ in Figure 6, when wind shear, S, is positive. Interestingly, $\tilde{s}(V, S)$ tends to increase when wind speed gets large and wind shear becomes smaller below 0. Note that wind shear measures the wind stability in the vertical direction. Typically, wind speed increases at a higher location. The negative value of wind shear means that wind speed at a higher location is smaller than that at a lower location (see its definition in Section 3.1). Such negative wind shear happens at very unstable wind conditions and negatively affects the turbine reliability. Lastly, $\tilde{s}(I, S)$ tends to slightly increase as either I or $|S|$ gets larger. Unlike the other two pairs, the interaction between turbulence intensity and wind shear appears to be insignificant which explains the small weight for $\tilde{s}(I, S)$ in Figure 5.

Table 5 compares the results based on 25 repetitions of each method. All methods use the same computational budgets, that is, $N^{(0)} = 600$ and $N^{(t)} = 200$, $t = 1, \ldots, 10$. WAMK-SIS provides the best results in terms of SE and RR. It turns out that CE-SIS does not provide computational advantage over CMC.

6. Conclusion. This study proposes a nonparametric IS method with multiple interacting input variables. Our method can automatically approximate the unknown conditional

TABLE 5
Comparison results from 25 experiments for the wind turbine case study

Method	Mean	SE	RR
WAMK-SIS	0.049218	0.003413	49.05%
AMK-SIS	0.048078	0.004514	85.81%
CE-SIS	0.050101	0.013824	804.65%

POE, $s(\mathbf{x})$, appearing in the optimal SIS density for stochastic computer models. Motivated from the wind energy application, our approach is designed to estimate the conditional POE with a multidimensional input vector and to capture the interaction effects without causing computational or data sparsity problems. It also accounts for the contribution of each input variable on the output of interests by assigning different weights to subestimators.

The numerical studies in a wide range of settings and the wind turbine case study evaluate the WAMK-SIS performance in comparison with the alternative approaches. The results show that WAMK-SIS considerably outperforms CE-SIS when input dimension gets larger. WAMK-SIS also makes notable improvements over AMK-SIS, when the effects of input variables are different. While our approach primarily aims to capture two-factor multiplicative interactions, it appears to perform well when weak higher-order interactions exist.

In the future we plan to extend the analysis with more input variables. For an offshore wind turbine, additional environmental variables, such as wave height, should be considered. With a higher dimensional input vector, we plan to devise a subestimator selection procedure which could be similar to the variable selection in general regression problems. In the subestimator selection procedure, some negligible subestimators, to which small weights are assigned, can be dropped during the iterative process which may help avoid an overfitting issue and prevent an unnecessarily complicated form for defining the IS density.

The feasibility of using the proposed method would depend on the simulation run time and available computational resources. In our case study each run takes about one minute, and we use a server with 25 cores. If each run needs more time, for example, hours, and/or computational resource is limited, we may reduce the sample size at the expense of deteriorated estimation accuracy. For extremely complex simulations that require several days or even weeks for each run, our approach may not be a feasible option. Reliability analysis with complex simulators that require long run times is another subject of our future study.

REFERENCES

ACKLEY, D. (2012). *A Connectionist Machine for Genetic Hillclimbing* **28**. Springer, Berlin.

ADVANCE METROLOGY LAB (2020). Inland Wind Farm Dataset2 (WT2). Available at https://aml.engr.tamu.edu/book-dswe/dswe-datasets/.

AZIZ EZZAT, A., JUN, M. and DING, Y. (2019). Spatio-temporal short-term wind forecast: A calibrated regime-switching method. *Ann. Appl. Stat.* **13** 1484–1510. MR4019147 https://doi.org/10.1214/19-AOAS1243

BIERENS, H. J. (1987). Kernel estimators of regression functions. In *Advances in Econometrics*: *Fifth World Congress* **1** 99–144. Cambridge Univ. Press, New York.

BISHOP, C. M. (2006). *Pattern Recognition and Machine Learning*. *Information Science and Statistics*. Springer, New York. MR2247587 https://doi.org/10.1007/978-0-387-45528-0

CANNAMELA, C., GARNIER, J. and IOOSS, B. (2008). Controlled stratification for quantile estimation. *Ann. Appl. Stat.* **2** 1554–1580. MR2655671 https://doi.org/10.1214/08-AOAS186

CAO, Q. D. and CHOE, Y. (2019). Cross-entropy based importance sampling for stochastic simulation models. *Reliab. Eng. Syst. Saf.* **191** 106526.

CHEN, Y.-C. and CHOE, Y. (2019). Importance sampling and its optimality for stochastic simulation models. *Electron. J. Stat.* **13** 3386–3423. MR4010983 https://doi.org/10.1214/19-ejs1604

CHOE, Y., BYON, E. and CHEN, N. (2015). Importance sampling for reliability evaluation with stochastic simulation models. *Technometrics* **57** 351–361. MR3384950 https://doi.org/10.1080/00401706.2014.1001523

CHOE, Y., LAM, H. and BYON, E. (2018). Uncertainty quantification of stochastic simulation for black-box computer experiments. *Methodol. Comput. Appl. Probab.* **20** 1155–1172. MR3873620 https://doi.org/10.1007/s11009-017-9599-7

CHOE, Y., PAN, Q. and BYON, E. (2016). Computationally efficient uncertainty minimization in wind turbine extreme load assessments. *J. Sol. Energy Eng.* **138** 041012.

CHOI, K.-S., HUH, Y.-H., KWON, I.-B. and YOON, D.-J. (2012). A tip deflection calculation method for a wind turbine blade using temperature compensated FBG sensors. *Smart Materials and Structures* **21** 025008.

DE BOER, P.-T., KROESE, D. P., MANNOR, S. and RUBINSTEIN, R. Y. (2005). A tutorial on the cross-entropy method. *Ann. Oper. Res.* **134** 19–67. MR2136658 https://doi.org/10.1007/s10479-005-5724-z

DING, Y. (2019). *Data Science for Wind Energy*. CRC Press, Boca Raton, FL.

DUBOURG, V., SUDRET, B. and DEHEEGER, F. (2013). Metamodel-based importance sampling for structural reliability analysis. *Probab. Eng. Mech.* **33** 47–57.

FAN, J. and GIJBELS, I. (2018). *Local Polynomial Modelling and Its Applications*: *Monographs on Statistics and Applied Probability* **66**. Routledge, London. MR1383587

FAN, J. and YIM, T. H. (2004). A cross-validation method for estimating conditional densities. *Biometrika* **91** 819–834. MR2126035 https://doi.org/10.1093/biomet/91.4.819

GIVENS, G. H. and RAFTERY, A. E. (1996). Local adaptive importance sampling for multivariate densities with strong nonlinear relationships. *J. Amer. Statist. Assoc.* **91** 132–141. MR1394067 https://doi.org/10.2307/2291389

GUALTIERI, G. (2016). Atmospheric stability varying wind shear coefficients to improve wind resource extrapolation: A temporal analysis. *Renew. Energy* **87** 376–390.

HANSEN, B. E. (2009). Lecture notes on nonparametrics. Available at https://www.ssc.wisc.edu/~bhansen/718/NonParametrics2.pdf.

INTERNATIONAL ELECTROTECHNICAL COMMISSION (2005). Wind Turbines—Part 1: Design Requirements, IEC/TC88,61400-1 ed.3.

JONKMAN, B. J. (2009). TurbSim user's guide: Version 1.50. Golden, CO: National Renewable Energy Laboratory.

JONKMAN, J. M. and BUHL JR, M. L. (2005). FAST user's guide. Golden, CO: National Renewable Energy Laboratory.

JONKMAN, J., BUTTERFIELD, S., MUSIAL, W. and SCOTT, G. (2005). Definition of a 5-MW reference wind turbine for offshore system development. Golden, CO: National Renewable Energy Laboratory.

KO, Y. M. and BYON, E. (2021). Optimal budget allocation for stochastic simulation with importance sampling: Exploration vs. replication. *IISE Trans.* To appear.

KONG, C., BANG, J. and SUGIYAMA, Y. (2005). Structural investigation of composite wind turbine blade considering various load cases and fatigue life. *Energy* **30** 2101–2114.

KURTZ, N. and SONG, J. (2013). Cross-entropy-based adaptive importance sampling using Gaussian mixture. *Struct. Saf.* **42** 35–44.

LEE, G., BYON, E., NTAIMO, L. and DING, Y. (2013). Bayesian spline method for assessing extreme loads on wind turbines. *Ann. Appl. Stat.* **7** 2034–2061. MR3161712 https://doi.org/10.1214/13-AOAS670

LEE, G., DING, Y., GENTON, M. G. and XIE, L. (2015). Power curve estimation with multivariate environmental factors for inland and offshore wind farms. *J. Amer. Statist. Assoc.* **110** 56–67. MR3338486 https://doi.org/10.1080/01621459.2014.977385

LI, Q. and RACINE, J. S. (2007). *Nonparametric Econometrics*: *Theory and Practice*. Princeton Univ. Press, Princeton, NJ. MR2283034

MANUEL, L., NGUYEN, H. H. and BARONE, M. F. (2013). On the use of a large database of simulated wind turbine loads to aid in assessing design standard provisions. In *Proceedings of the* 51*st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition* **4**.

MORIARTY, P. (2008). Database for validation of design load extrapolation techniques. *Wind Energy*: *An International Journal for Progress and Applications in Wind Power Conversion Technology* **11** 559–576.

NADARAYA, E. A. (1964). On estimating regression. *Theory Probab. Appl.* **9** 141–142.

NEDDERMEYER, J. C. (2009). Computationally efficient nonparametric importance sampling. *J. Amer. Statist. Assoc.* **104** 788–802. MR2541595 https://doi.org/10.1198/jasa.2009.0122

OWEN, A. B. (2013). Monte Carlo Theory, Methods and Examples. Book in progress. Online version available at https://statweb.stanford.edu/~owen/mc/.

PAN, Q., KO, Y. M. and BYON, E. (2021). Uncertainty quantification for extreme quantile estimation with stochastic computer models. *IEEE Trans. Reliab.* **70** 134–145.

PAN, Q., BYON, E., KO, Y. M. and LAM, H. (2020). Adaptive importance sampling for extreme quantile estimation with stochastic black box computer models. *Naval Res. Logist.* **67** 524–547.

RUBINSTEIN, R. Y. and KROESE, D. P. (2017). *Simulation and the Monte Carlo Method. Wiley Series in Probability and Statistics*. Wiley, Hoboken, NJ. MR3617204

WASSERMAN, L. (2006). *All of Nonparametric Statistics. Springer Texts in Statistics*. Springer, New York. MR2172729

WISER, R., HAND, M., SEEL, J. and PAULOS, B. (2016). The Future of Wind Energy, Part 3: Reducing Wind Energy Costs through Increased Turbine Size: Is the Sky the Limit? Available at https://emp.lbl.gov/news/future-wind-energy-part-3-reducing-wind.

WISER, R., BOLINGER, M., HOEN, B., MILLSTEIN, D., RAND, J., BARBOSE, G., DARGHOUTH, N., GORMAN, W., JEONG, S. et al. (2020). Wind Energy Technology Data Update: 2020 Edition. Technical Report, Lawrence Berkeley National Laboratory, Berkeley, CA.