Water Resources Research

METHOD

10.1029/2022WR033808

An Improved Copula-Based Framework for Efficient Global Sensitivity Analysis

Hongli Liu[1,2], Martyn P. Clark[2], Shervan Gharari[3], Razi Sheikholeslami[4], Jim Freer[2], Wouter J. M. Knoben[2], Christopher B. Marsh[3], and Simon Michael Papalexiou[5]

[1]Department of Civil and Environmental Engineering, University of Alberta, Edmonton, AB, Canada, [2]Centre for Hydrology, University of Saskatchewan, Canmore, AB, Canada, [3]Centre for Hydrology, University of Saskatchewan, Saskatoon, SK, Canada, [4]Department of Civil Engineering, Sharif University of Technology, Tehran, Iran, [5]Department of Civil Engineering, University of Calgary, Calgary, AB, Canada

Key Points:

- We improve the VarIance-based Sensitivity analysis using COpUlaS (VISCOUS) global sensitivity analysis framework in its handling of marginal densities of the Gaussian mixture copula model
- We evaluate VISCOUS and demonstrate how its performance is affected by function dimension, input-output size, and non-identifiability
- We provide a didactic example and an open-source Python code called pyVISCOUS to make VISCOUS easier to understand and apply

Correspondence to:

H. Liu,
hongli.liu@ualberta.ca

Citation:

Liu, H., Clark, M. P., Gharari, S., Sheikholeslami, R., Freer, J., Knoben, W. J. M., et al. (2024). An improved copula-based framework for efficient global sensitivity analysis. *Water Resources Research*, *60*, e2022WR033808. https://doi.org/10.1029/2022WR033808

Received 3 OCT 2022
Accepted 22 DEC 2023

Abstract Global sensitivity analysis (GSA) enhances our understanding of computational models and simplifies model parameter estimation. VarIance-based Sensitivity analysis using COpUlaS (VISCOUS) is a variance-based GSA framework. The advantage of VISCOUS is that it can use existing model input-output data (e.g., water model parameters-responses) to estimate the first- and total-order Sobol' sensitivity indices. This study improves VISCOUS by refining its handling of marginal densities of the Gaussian mixture copula model (GMCM). We then evaluate VISCOUS using three types of generic functions relevant to water system models. We observe that its performance depends on function dimension, input-output data size, and non-identifiability. Function dimension refers to the number of uncertain input factors analyzed in GSA, and non-identifiability refers to the inability to estimate GMCM parameters. VISCOUS proves powerful in estimating first-order sensitivity with a small amount of input-output data (e.g., 200 in this study), regardless of function dimension. It always ranks input factors correctly in both first- and total-order terms. For estimating total-order sensitivity, it is recommended to use VISCOUS when the function dimension is not very high (e.g., less than 20) due to the challenge of producing sufficient input-output data for accurate GMCM inferences (e.g., more than 10,000 data). In cases where all input factors are equally important (a rarity in practice), VISCOUS faces non-identifiability issues that impact its performance. We provide a didactic example and an open-source Python code, pyVISCOUS, for broader user adoption.

Plain Language Summary Global sensitivity analysis is a method used to better understand and estimate parameters in computational models. VarIance-based Sensitivity analysis using COpUlaS (VISCOUS) is a framework for this purpose. It estimates the sensitivity of model outcomes to different uncertain model input factors by using the existing input and output data (e.g., water model parameters and responses). This study improved VISCOUS and tested it with various functions. We found that its performance depends on the number of input factors, the amount of input and output data available, and our ability to determine VISCOUS's parameters. VISCOUS is good at estimating the importance of individual input factors, even with limited data (e.g., 200) and numerous input factors. It always correctly ranks input factor importance, whether individually or collectively. When estimating the importance of input factors together, VISCOUS is recommended when the number of input factors is not very high (e.g., <20), as it is challenging to generate enough input and output data for estimating VISCOUS's parameters. When all input factors hold equal importance (though rare in practice), VISCOUS's performance is impacted due to the difficulty of estimating VISCOUS's parameters. To help people use VISCOUS, we provide an example and an open-source Python code, pyVISCOUS.

1. Introduction

Sensitivity analysis investigates how the uncertainty of model output can be attributed to the different uncertain input factors and their interactions (Pianosi et al., 2016). The model output refers to the variable obtained after the model is executed. The model input factors, also known as input variables, refer to any aspects of the model that can be changed before model execution, such as model parameters, initial states, forcing data, model parameterization, and model temporal/spatial resolution in case of dynamic models (Pianosi et al., 2016). The most common input factor in sensitivity analysis is model parameters (Norton, 2015).

Sensitivity analysis is useful in many ways, such as ranking input factors, fixing negligible factors, determining the region of the input space that has a substantial control on model output, and prioritizing data acquisition processes to focus on the model inputs that have the largest effect on the desired outcome (Nossent et al., 2011; Razavi & Gupta, 2015; Saltelli et al., 2008; van Griensven et al., 2006). Sensitivity analysis can also lend insight into the dominant processes that govern spatiotemporal variability of a system by exploring the full spectrum of its behavior and the strengths and weaknesses of the water system model (Demaria et al., 2007; Markstrom et al., 2016; Razavi et al., 2021).

Sensitivity analysis methods can be generally classified into local and global methods. Local sensitivity analysis methods evaluate the effects of the input variations around a specific point in the input space, and global sensitivity analysis (GSA) evaluates the effects of the input variations across the entire input space (Pianosi et al., 2016). In GSA, a well-established and widely used method is the variance-based approach, for example, the method of Sobol' which decomposes the total variance into contributions from different input factors (Homma & Saltelli, 1996; Sobol', 2001).

Variance-based methods are attractive because they are model independent, they measure interaction effects among input factors, and they handle groups of input factors (Saltelli et al., 2008). The major challenge associated with application of variance-based methods is their computational cost, because they require model evaluations for a considerable number of input samples. Running a model for a large number of input samples may be difficult to achieve if the model is computationally expensive. Therefore, much recent research aims to find efficient numerical algorithms to compute variance-based sensitivity indices (Hu & Mahadevan, 2019; Sheikholeslami et al., 2019).

To overcome the aforementioned computational bottleneck, Sheikholeslami et al. (2021) developed a computationally frugal GSA framework called VISCOUS (VarIance-based Sensitivity analysis using COpulaS). VISCOUS first uses a Gaussian Mixture Copula Model (GMCM) to approximate the joint probability distribution between the input (e.g., the perturbations in the model parameters) and output data (e.g., the model responses given parameter perturbations); and then approximates the first- and total-order Sobol' sensitivity indices based on the fitted GMCM. VISCOUS belongs to the class of given-data approach, also known as the data-driven approach. This approach allows GSA to be applied to existing input-output data, regardless of whether the underlying relationships or mechanisms are known. It is beneficial for computationally intensive models when the input-output data exist (Sheikholeslami & Razavi, 2020).

In comparison to other variance-based GSAs, VISCOUS provides an advantage by eliminating the need for input-output data to follow specific sampling strategies, as required in traditional Monte Carlo methods for Sobol' sensitivity indices (e.g., Homma and Saltelli, 1996, Saltelli, 2002). This is because input-output data in VISCOUS are not used to directly calculate Sobol' sensitivity indices but are used for training the GMCM. Therefore, input-output data can be from previous model runs for other modeling purposes, such as calibration and uncertainty analysis. Moreover, VISCOUS does not impose assumptions on the structure of input-output data, as required in many emulator-based GSA, especially those employing ANOVA (Analysis of Variance). For example, assumptions about negligible higher-order interactions, as required in Borgonovo et al. (2012), Plischke et al. (2013), and Stanfill et al. (2015), are not enforced by VISCOUS. This characteristic enhances VISCOUS's applicability in diverse models.

The motivation of this research is to improve the VISCOUS of Sheikholeslami et al. (2021). In Sheikholeslami et al. (2021), the GMCM marginal densities are defined as the standard normal distribution along all variable dimensions (i.e., zero mean and unit variance). However, these marginal densities are inefficient as they remain fixed during the GMCM inference process, neglecting the impact of updated GMCM parameters. This may result in biased GMCM parameter estimates and inaccurate sensitivity indices, especially when insufficient iterations are allowed in GMCM inference. The objective of this paper is to improve the VISCOUS methodology by refining the GMCM marginal densities. This methodological advance will lead to a more efficient GMCM inference and improved sensitivity index estimates.

The structure of this paper is organized as follows. Section 2 explains the methodology of the improved VISCOUS framework. Section 3 evaluates VISCOUS using three types of Sobol' functions and demonstrate how its performance is affected by function dimension, input-output data size, and GMCM non-identifiability.

Section 4 introduces the Python code of VISCOUS called pyVISCOUS. The paper concludes with discussion of future work and potential utility of VISCOUS for different modeling applications.

2. Methodology

This section describes the methodology of the improved VISCOUS framework. The essence of VISCOUS is to develop and use the Gaussian mixture copula model (GMCM) to calculate the Sobol' sensitivity indices. To explain VISCOUS, we first review the Sobol' variance-based GSA method and then explain the GMCM method in detail, including the improved handling of the GMCM marginal densities. With both, we provide the derivations of the first- and total-order Sobol' sensitivity indices. Finally, we explain the implementation steps of the VISCOUS framework using Monte Carlo-based approximations.

In the following, random variables are denoted by capital letters, and their values are denoted by lowercase letters. For example, $F_X(x)$ is the cumulative distribution function of the random variable X evaluated at x. Bold face letters denote vectors or matrices, such as $\mathbf{X} = [X_1,\ldots,X_d]$, where d is the number of variables.

2.1. Overview of the Sobol' Global Sensitivity Analysis

Assume a water system model is expressed as:

$$Y = H(X_1, X_2, \ldots, X_d) \tag{1}$$

where a total of d input factors are evaluated in sensitivity analysis. Assume the variance of model outputs is a good proxy of output uncertainty and input factors are random and independent. The variance of model response (Y) can be decomposed into partial variances: first-order variance (V_i), second-order variance (V_{ij}), …, until d-order variance ($V_{i..d}$) (Saltelli, 2002; Saltelli et al., 2008).

$$V(Y) = \sum_{i=1}^{d} V_i + \sum_{i=1}^{d}\sum_{j=i+1}^{d} V_{ij} + \ldots + V_{i..d} \tag{2}$$

where $V(Y)$ is the variance of the model response Y.

The first-order sensitivity index (S_i) is calculated as:

$$S_i = \frac{V_i}{V(Y)} = \frac{V(E(Y|X_i))}{V(Y)} \tag{3}$$

where $V(E(Y|X_i))$ is the variance of mean Y over X_i alone. It represents the contribution of the single input X_i to the variance of response Y. The first-order sensitivity index is also called the main effect sensitivity index.

The total-order sensitivity index (S_{Ti}) is calculated as:

$$S_{Ti} = 1 - S_{\sim i} = 1 - \frac{V(E(Y|\mathbf{X}_{\sim i}))}{V(Y)} \tag{4}$$

where $V(E(Y|\boldsymbol{X}_{\sim i}))$ is the variance of mean Y over all $\mathbf{X}$ except X_i. It represents the total contribution of non-X_i, denoted as $\mathbf{X}_{\sim i}$, to the variance of response Y. The total-order sensitivity index is also called the total effect sensitivity index. It includes not only the first-order effects of an input variable but also its higher-order interactions with other input variables.

Sobol' sensitivity indices range from zero to one. The closer an index value is to one, the better the associated input variable explains the model output. Moreover, from Equations 3 and 4, we see that the calculation of conditional expectations, $E(Y|X_i)$ and $E(Y|\boldsymbol{X}_{\sim i})$, is the cornerstone of the variance-based sensitivity analysis. In the following sections, we will explain the GMCM method and the use of it to calculate $E(Y|X_i)$ and $E(Y|\boldsymbol{X}_{\sim i})$.

2.2. Gaussian Mixture Copula Model (GMCM)

Assume a random vector $[\mathbf{X},Y] = [X_1,\ldots,X_d,Y]$, and each element has a continuous cumulative distribution function (CDF). If X is a continuous random variable with CDF F_X, then F_X is uniformly distributed between zero and one

based on the probability integral transform theorem. Therefore, $\left[F_{X_1}(x_1),\ldots,F_{X_d}(x_d),F_Y(y)\right]=\left[u_{x_1},\ldots,u_{x_d},u_y\right]$, where each $u\in[0,1]$ follows the uniform distribution.

The joint distribution $F_{\mathbf{X},Y}$ can be expressed as a function of the marginal distributions based on Sklar's theorem (Sklar, 1959; Tewari et al., 2011):

$$F_{\mathbf{X},Y}(\mathbf{x},y)=C(\mathbf{u}_x,u_y) \tag{5}$$

where $\boldsymbol{x}=[x_1,\ldots,x_d]$, $\mathbf{u}_x=[u_1,\ldots,u_d]$. $F_{\mathbf{X},Y}$ is the joint CDF of $(\boldsymbol{X},Y)$. C is the copula function defined as the joint CDF of $(\mathbf{u}_x,u_y)$. The copula function specifies the distribution of $(\mathbf{X},Y)$ by specifying their marginal distributions and linking the marginal distributions through the copula function.

The joint PDF of $(\mathbf{X},Y)$, $f_{\mathbf{X},Y}$, is obtained by computing the derivative of Equation 5:

$$f_{\mathbf{X},Y}(\mathbf{x},y)=\frac{\partial^{d+1}C(\mathbf{u}_x,u_y)}{\partial u_{x_1}\cdot\ldots\cdot\partial u_{x_d}\cdot\partial u_y}\cdot\prod_{i=1}^{d}\frac{\partial u_{x_i}}{\partial x_i}\cdot\frac{\partial u_y}{\partial y}=c(\mathbf{u}_x,u_y)\cdot\prod_{i=1}^{d}f_{X_i}(x_i)\cdot f_Y(y) \tag{6}$$

where $c(\mathbf{u}_x,u_y)$ is the copula density. In GMCM, a Gaussian Mixture Model (GMM) is used to approximate the copula function as there is no simple analytical formula for the copula function (Tewari et al., 2011).

2.2.1. Gaussian Mixture Model (GMM)

A GMM is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions (Singh, 2019; Xu & Jordan, 1996). The GMM CDF is denoted by:

$$F^{GMM}_{\mathbf{Z}_x,Z_y}(\mathbf{z}_x,z_y)=\sum_{k=1}^{K}\lambda_k\cdot\Phi\left(\mathbf{z}_x,z_y|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right),$$

$$\text{where } z_{x_i}=\Phi^{-1}_{z_{x_i}}\left(u_{x_i}\right),z_y=\Phi^{-1}_{z_y}\left(u_{\boldsymbol{y}}\right),i=1,\ldots,d \tag{7}$$

where K is the total number of Gaussian components or clusters. λ_k is the weight of the kth Gaussian component. $\lambda_k>0$ and $\sum_{k=1}^{K}\lambda_k=1$. Φ is the CDF of a multivariate Gaussian distribution with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$. The GMM parameter vector combines the weights, mean vectors and covariance matrices of all the Gaussian components, notated as $\boldsymbol{\Theta}$ in the rest of the paper.

In the improved VISCOUS, $\Phi_{z_{x_i}}$ and Φ_{z_y} are the GMM marginal CDF for z_{x_i} and z_y, respectively. $\Phi^{-1}_{z_{x_i}}$ and $\Phi^{-1}_{z_y}$ are the corresponding inverse distribution function. There is no closed form expression for the inverse function, so a linear interpolation is used to obtain the inverse values based on the GMM parameters $\boldsymbol{\Theta}$ (Tewari et al., 2011). z_{x_i} and z_y are the obtained inverse values of u_{x_i} and u_y, respectively. More details about the GMM are in Appendix A.

2.2.2. Gaussian Mixture Copula Model (GMCM)

The GMCM function is derived from the GMM. When the Gaussian mixture copula function is approximated by a GMM:

$$C(\mathbf{u}_x,u_y)\approx F^{GMM}_{\mathbf{Z}_x,Z_y}(\mathbf{z}_x,z_y) \tag{8}$$

the copula density, $c(\mathbf{u}_x,u_y)$, is approximated by:

$$c(\mathbf{u}_x,u_y)\approx f^{GMM}_{\mathbf{Z}_x,Z_y}(\mathbf{z}_x,z_y)\cdot\prod_{i=1}^{d}\frac{\partial z_{x_i}}{\partial u_{x_i}}\cdot\frac{\partial z_y}{\partial u_y}=\frac{f^{GMM}_{\mathbf{Z}_x,Z_y}(\mathbf{z}_x,z_y)}{\prod_{i=1}^{d}\phi_{z_{x_i}}\left(z_{x_i}\right)\cdot\phi_{z_y}(z_y)}$$

$$\text{where } f^{GMM}_{\mathbf{Z}_x,Z_y}(\mathbf{z}_x,z_y)=\sum_{k=1}^{K}\lambda_k\cdot\phi\left(\mathbf{z}_x,z_y|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right) \tag{9}$$

where $f^{GMM}_{\mathbf{Z}_x,Z_y}(\mathbf{z}_x,z_y)$ is the GMM PDF. ϕ is the PDF of a multivariate Gaussian distribution with mean $\boldsymbol{\mu}_k$ and covariance Σ_k. $\phi_{z_{x_i}}$ and ϕ_{z_y} are the GMM marginal PDF for z_{x_i} and z_y, respectively.

From Equation 9, we see that the dependence structure of the GMCM is obtained from the GMM. Moreover, the GMCM function shares the same parameter set as the GMM function. The GMCM parameters $\boldsymbol{\Theta}$ are estimated by a modified Expectation-Maximization (EM) algorithm of Tewari et al. (2011) (detailed in Appendix B).

Here it is worth noting the algorithmic advancement relative to the VISCOUS of Sheikholeslami et al. (2021). In Sheikholeslami et al. (2021), the marginal densities $\Phi_{z_{x_i}}$ and Φ_{z_y} are defined as the standard normal distribution along all variable dimensions (i.e., zero mean and unit variance). As such, the marginal densities are independent of the formula of GMCM. This has a subsequent impact on the GMCM inference as the marginal densities ignore the updated GMCM parameters and remain fixed in the inference process. This may lead to optimizing GMCM parameters taking longer than necessary, and introducing biases in these parameter estimates and inaccurate sensitivity index estimates. The presented methodology in this paper overcomes this shortcoming by defining the GMCM marginal densities based on the formula of GMCM and adopting iteratively updated marginal densities based on the GMCM parameters in the inference process. This methodological advance helps to obtain a GMCM function that fits the input-output data more efficiently and provides better sensitivity index estimates.

2.3. GMCM-Based Sobol' Sensitivity Index Estimation

As explained in Session 2.1, the variance-based sensitivity index estimation relies on the conditional expectations, $E(Y|X_i)$ and $E(Y|\boldsymbol{X}_{\sim i})$. The following explains the general use of GMCM to compute the conditional PDF, $f_{Y|\mathbf{X}}$. With $f_{Y|\mathbf{X}}$, it then explains the computation of $E(Y|X_i)$ and the first-order sensitivity index. The computation of $E(Y|\boldsymbol{X}_{\sim i})$ and the total-order sensitivity index follows a similar logic.

2.3.1. Model Conditional PDF of Y

To compute the conditional PDF of $Y, f_{Y|\mathbf{X}}$, we need the joint PDF of $(\mathbf{X},Y)$ and the marginal PDF of $\mathbf{X}$. In GMCM, the joint PDF of $(\mathbf{X},Y), f_{\mathbf{X},Y}$, is estimated based on Equations 6 and 9:

$$f_{\mathbf{X},Y}(\mathbf{x},y) \approx \frac{f^{GMM}_{\mathbf{Z}_x,Z_y}(\mathbf{z}_x,z_y)}{\prod_{i=1}^{d}\phi\left(z_{x_i}\right)\cdot\phi(z_y)}\cdot\prod_{i=1}^{d}f_{X_i}(x_i)\cdot f_Y(y) \tag{10}$$

The marginal PDF of $\mathbf{X}, f_{\mathbf{X}}$, is estimated as:

$$f_{\mathbf{X}}(\mathbf{x}) \approx \frac{f^{GMM}_{\mathbf{Z}_x}(\mathbf{z}_x)}{\prod_{i=1}^{d}\phi\left(z_{x_i}\right)}\cdot\prod_{i=1}^{d}f_{X_i}(x_i) \tag{11}$$

where $f^{GMM}_{\mathbf{Z}_x}(\mathbf{z}_x)$ is the GMM marginal PDF of $\mathbf{Z}_x$ obtained with Equation A5 of Appendix A.

The conditional PDF of $Y, f_{Y|\mathbf{X}}$, is obtained by dividing $f_{\mathbf{X},Y}$ by $f_{\mathbf{X}}$:

$$f_{Y|\mathbf{X}}(y|\mathbf{x}) = \frac{f_{\mathbf{X},Y}(\mathbf{x},y)}{f_{\mathbf{X}}(\mathbf{x})} \approx \frac{f^{GMM}_{\mathbf{Z}_x,Z_y}(\mathbf{z}_x,z_y)}{f^{GMM}_{\mathbf{Z}_x}(\mathbf{z}_x)}\cdot\frac{1}{\phi_{z_y}(z_y)}\cdot f_Y(y) \tag{12}$$

2.3.2. First-Order Sensitivity Index

When the input variable X_i is fixed to a value x_i, the resulting conditional expectation of Y is:

$$E(Y|X_i=x_i) = \int_{\Omega Y} y\cdot f_{Y|\mathbf{X}}(y|x_i)\mathrm{d}y \tag{13}$$

where ΩY is a region of Y over which integration is conducted. Equation 13 is approximated using Equation 12:

$$E(Y|X_i = x_i) \approx \int_{\Omega Y} y \cdot \frac{f^{GMM}_{\mathbf{Z}_x, Z_y}(\mathbf{z}_x, z_y)}{f^{GMM}_{\mathbf{Z}_x}(\mathbf{z}_x)} \cdot \frac{1}{\phi_{z_y}(z_y)} \cdot f_Y(y)\mathrm{d}y \tag{14}$$

Since $\frac{\mathrm{d}u_y}{\mathrm{d}y} = f_Y(y)$, the above equation becomes:

$$E(Y|X_i = x_i) \approx \int_0^1 y \cdot \frac{f^{GMM}_{Z_x, Z_y}(z_x, z_y)}{f^{GMM}_{Z_x}(z_x)} \cdot \frac{1}{\phi_{z_y}(z_y)} \mathrm{d}u_y \tag{15}$$

To drop the dependence upon the specific value x_i, the variance of $E(Y|X_i)$ is estimated by integrating $E(Y|X_i = x_i)$ over the probability density function of X_i, expressed as:

$$V(E(Y|X_i)) = \int_{\Omega x_i} E^2(Y|X_i = x_i)\mathrm{d}x_i - \left[\int_{\Omega x_i} E(Y|X_i = x_i)\mathrm{d}x_i\right]^2 \tag{16}$$

$E(Y|X_i)$ and $V(E(Y|X_i))$ can be estimated using Monte Carlo approximations, which is the content of the next section. With $V(E(Y|X_i))$, the first-order sensitivity index is computed based on Equation 3. Similar approach can be used to calculate $E(Y|\boldsymbol{X}_{\sim i})$, $V(E(Y|\boldsymbol{X}_{\sim i}))$, and the total-order sensitivity index by replacing the X_i with $\mathbf{X}_{\sim i}$ and is detailed in Appendix C. The above also shows that two loops are needed in the computation of $V(E(Y|X_i))$. The inner loop is to compute $E(Y|X_i)$ by integrating over u_y. The outer loop is to compute the variance of $E(Y|X_i)$ by integrating over x_i.

2.4. Steps for Performing VISCOUS

This section explains the implementation steps of the VISCOUS framework using Monte Carlo-based approximations. Six steps are involved (Figure 1). Same as in Section 2.3, we take the first-order sensitivity index of X_i as an example. The procedure is the same for the total-order sensitivity index except replacing X_i with $\boldsymbol{X}_{\sim i}$ and is detailed in Appendix C. Additionally, Appendix D demonstrates the implementation steps using the two-parameter Rosenbrock function. This didactic example aims to help users to better understand the details within the VISCOUS method and apply it for their own applications.

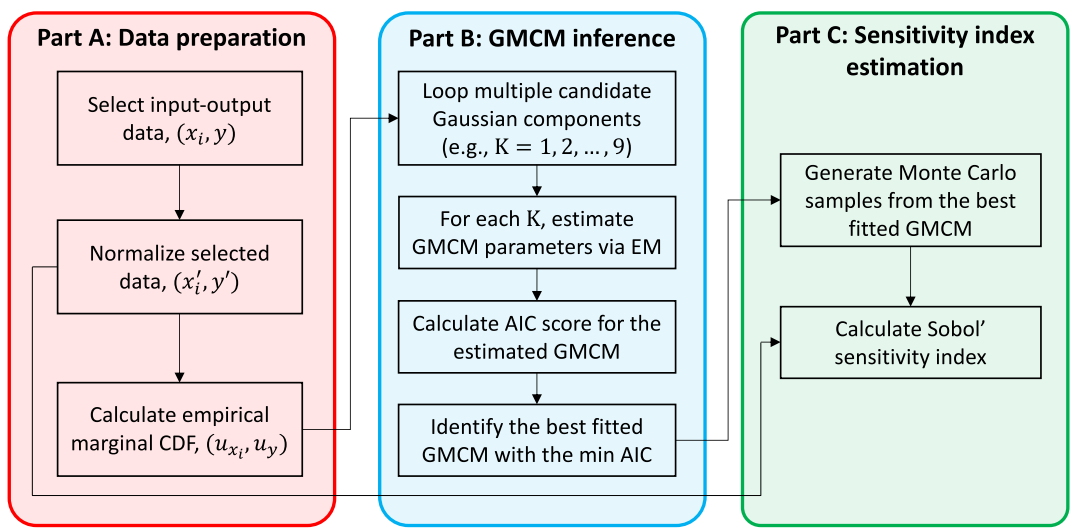

Figure 1. Flowchart of performing the VISCOUS framework. CDF denotes the cumulative distribution function. GMCM denotes the Gaussian mixture copula model. EM denotes the Expectation-Maximization algorithm. AIC denotes the Akaike information criterion.

2.4.1. Part A. Data Preparation

Step 1. Select the evaluated input and output data based on the goal of sensitivity analysis. For example, when calculating the first-order sensitivity index of variable X_i, the selected input-output data are (x_i,y).

Step 2. Normalize the selected input-output data using the min-max normalization method. Normalization transforms data into a common scale without changing the relationships among data. This improves the performance and training stability of the GMCM. The produced normalized data $\left(x_i', y'\right)$ will be used in the calculation of Sobol' sensitivity indices.

Step 3. Calculate the rank-based empirical CDF for each variable of the normalized data, getting the marginal CDF data (u_x,u_y). Rank transformation is a common procedure to get marginal CDFs when the data distribution is unknown or complex (Saltelli and Sobol', 1995). The marginal CDFs are used to derive the inverse CDF values (z_x,z_y) in the following GMCM inference.

2.4.2. Part B. GMCM Inference

Finding the best fitted GMCM involves solving two problems. The first problem is to determine the optimal number of Gaussian components (K). The second problem is to determine the optimal GMCM parameters ($\boldsymbol{\Theta}$). Therefore, the following two steps are conducted interactively.

Step 4. To find the optimal value of K, we use a statistic known as Akaike information criterion (AIC). AIC estimates the quality of a model by balancing its goodness of fit (log-likelihood) and complexity (penalty to the number of model parameters) (Akaike, 1974). Readers can explore alternative model selection criteria based on their data characteristics and analysis goals. For instance, Bayesian information criterion (BIC) is another popular model selection criterion (Vrieze, 2012) and has been added as an alternative in pyVISCOUS.

Step 4 compares the AICs of multiple GMCMs with different Gaussian component (K) values (e.g., $K = 1, 2,\ldots,9$ in this study). For each candidate K value, use a modified EM algorithm to estimate its corresponding GMCM parameters (Step 5), and then compute the AIC score for the estimated GMCM. The GMCM that achieves the lowest AIC value is identified as the best fitted GMCM, and its corresponding K value is the optimal K value.

Step 5. Given a Gaussian component value K, estimate the GMCM parameters using a modified EM algorithm. The EM algorithm is explained in Appendix B. In the EM, the marginal densities of GMCM change with every GMCM parameter update. The corresponding inverse distribution values $(\mathbf{z}_x,z_y)$ vary based on the form of the GMCM. A Python library called Copulas is used to perform the modified EM.

2.4.3. Part C. Sensitivity Index Estimation

Step 6. Once the best fitted GMCM is determined, generate the Monte Carlo samples $\left(z_{x_i}^{MC}, z_y^{MC}\right)$ from the GMCM, and calculate the variance-based first-order sensitivity index based on the samples. Step 6 is detailed in the following.

Based on the inferred GMCM, two rounds of sampling are performed to generate Monte Carlo samples. The first round of sampling generates N_1 samples, namely $\mathbf{z}_1^{MC}$ in Equation 17. $\mathbf{z}_1^{MC}$ provides samples for integration over x_i to obtain $V(E(Y|X_i))$ in the outer loop.

$$\mathbf{z}_1^{MC} = \begin{pmatrix} z_{1,x_i}^{MC} & z_{1,y}^{MC} \\ z_{2,x_i}^{MC} & z_{2,y}^{MC} \\ \vdots & \vdots \\ z_{N_1,x_i}^{MC} & z_{N_1,y}^{MC} \end{pmatrix}, \mathbf{z}_2^{MC} = \begin{pmatrix} z_{r_1,x_i}^{MC} & z_{1,y}'^{MC} \\ z_{r_1,x_i}^{MC} & z_{2,y}'^{MC} \\ \vdots & \vdots \\ z_{r_1,x_i}^{MC} & z_{N_2,y}'^{MC} \end{pmatrix}, r_1 = [1,\ldots,N_1]. \tag{17}$$

The second round of sampling generates N_2 samples, for example, $\mathbf{z}_2^{MC}$ in Equation 17. $\mathbf{z}_2^{MC}$ provides samples for integration over u_y to get $E(Y|X_i = x_i)$ in the inner loop. The second round of sampling needs repeating N_1 times by looping through each sample of $\mathbf{z}_1^{MC}$. Per iteration, N_2 Monte Carlo samples are generated from the inferred GMCM, and then all the values of z_{x_i} are replaced by a sample of $\mathbf{z}_1^{MC}$. See $\mathbf{z}_2^{MC}$ in Equation 17 as an example, the entire first column of $\mathbf{z}_2^{MC}$ is replaced by the r_1^{th} sample of $\mathbf{z}_1^{MC}$, z_{r_1,x_i}^{MC}. N_1 and N_2 can be but do not have to be the same ($N_1 = N_2 = 2{,}000$ in our study).

With the two rounds of Monte Carlo samples, we can approximate $E(Y|X_i)$ and $V(E(Y|X_i))$ in Equations 15 and 16. The conditional expectation $E(Y|X_i)$ in Equation 15 is approximated by:

$$E\left(Y|X_i = x_{r_1,i}^{MC}\right) \approx \frac{1}{N_2}\sum_{r_2=1}^{N_2} F_Y^{-1}\left(u_{r_2,y}^{MC}\right) \cdot \frac{f_{Z_{x_i},Z_y}^{GMM}\left(z_{r_1,x_i}^{MC}, z_{r_2,y}^{MC}\right)}{f_{Z_{x_i}}^{GMM}\left(z_{r_1,x_i}^{MC}\right)} \cdot \frac{1}{\phi_{z_y}\left(z_{r_2,y}^{MC}\right)} \tag{18}$$

where $x_{r_1,i}^{MC}$ is the r_1^{th} sample of $\mathbf{z}_1^{MC}$, $r_1 = [1,\ldots,N_1]$. $\left(z_{r_1,x_i}^{MC}, z_{r_2,y}^{MC}\right)$ is the r_2^{th} sample of $\mathbf{z}_2^{MC}$, $r_2 = [1,\ldots,N_2]$. $u_{r_2,y}^{MC}$ and $\phi_{z_y}\left(z_{r_2,y}^{MC}\right)$ are the marginal CDF and the marginal PDF of the GMM at $z_{r_2,y}^{MC}$, respectively. $F_Y^{-1}\left(u_{r_2,y}\right)$ is the inverse CDF of $u_{r_2,y}$ in the normalized space of Y, $F_Y^{-1}\left(u_{r_2,y}^{MC}\right) = y'^{MC}_{r_2}$.
The variance of $E(Y|X_i)$ in Equation 16 is approximated by:

$$V(E(Y|X_i)) \approx \frac{1}{N_1}\sum_{r_1=1}^{N_1} E^2\left(Y|X_i = x_{r_1,i}^{MC}\right) - \left[\frac{1}{N_1}\sum_{r_1=1}^{N_1} E\left(Y|X_i = x_{r_1,i}^{MC}\right)\right]^2 \tag{19}$$

With Equations 18 and 19, and Equation 3, the first-order sensitivity index can be computed. The procedure for calculating the total-order sensitivity index is similar and detailed in Appendix C.

3. Evaluation of the VISCOUS Framework

This section evaluates the improved VISCOUS framework using three types of Sobol' functions. We will first introduce the three types of functions, followed by comparative performance evaluation. We will also investigate three factors that affect the performance of VISCOUS: function dimension, input-output data size, and non-identifiability. Function dimension means the number of uncertain input factors analyzed in GSA, and non-identifiability refers to the inability to estimate the GMCM parameters.

3.1. Sobol' Function

According to Kucherenko et al. (2011), any model functions can be classified into three types based on their dependence on variables.

- Type A function: Variables are not equally important in terms of sensitivity.
- Type B function: Variables are equally important, and no interaction exists between variables. Therefore, $S_i = S_{Ti}$, $\sum S_i = 1$, and $S_i = 1/n$.
- Type C function: Variables are equally important, and interaction exists between variables. Therefore, $S_i < S_{Ti}$, and $\sum S_i < 1$.

Type A functions are the most common type of functions in practice. For instance, in most water system models, a large proportion of model output variation is often associated with a small proportion of the input factors (Markstrom et al., 2016). In statistics, this is known as the sparsity of effects principle or the Pareto principle (Box & Meyer, 1986). In the context of sensitivity analysis, this phenomenon reflects over-parameterization in model structure or the need for using a wider range of performance metrics for model evaluation.

Type B and C functions have all equally important variables. Equal importance means that all variables have the same sensitivity at all orders (i.e., first-order, second-order, …, and total-order). Type B and C functions differ in the interactions between variables. While these functions are uncommon, they provide valuable insights into the boundaries and limitations of a theory or methodology, aiding in refinement and improvement. Our study, which examines VISCOUS in type B and C functions, allows us to explore the full spectrum of possibilities, validate VISCOUS's robustness, and provide directions for future study.

The popular Sobol' function is adopted to examine the performance of VISCOUS in all three cases (Hu & Mahadevan, 2019; Kucherenko et al., 2011):

$$f(\mathbf{X}) = \prod_{i=1}^{d} \frac{|4X_i - 2| + a_i}{1 + a_i} \tag{20}$$

Set $d = 10$, then $(X_1,\ldots,X_{10})$ are the 10 input variables uniformly distributed in [0,1]. We can conveniently get all the three types of function by changing a_i (Kucherenko et al., 2011).

Table 1
Configurations of Four Sobol' Functions

Function type	Function name	$\boldsymbol{a}$ value
Type A	A1	$a_i = 25\lvert \sin(0.5i) + \cos(0.75i + 2)\rvert$
Type A	A2	$a_1 = a_2 = 0, a_3 = \cdots = a_d = 6.52$
Type B	B	$a_i = 6.52$
Type C	C	$a_i = 0$

Table 1 lists four functions that belong to the above three types of functions. Functions A1 and A2 are both Type A functions and are used to distinguish whether some (not all) variables are equally important in Type A. In function A1, all X variables are differently important. In function A2, X_1 and X_2 are equally important and X_3,…,X_{10} are equally important, but function A2 is more sensitive to X_1 and X_2 than to X_3,…,X_{10}. In functions B and C, all X variables are equally important, but the interactions between the variables are different, as stated above in the definitions of Type B and C functions.

3.2. Sensitivity Index Results

Figure 2 shows the first-order and total-order sensitivity index results of the four functions of Table 1 using the VISCOUS and Sobol' methods as well as the analytical true sensitivity index values. The Sobol' method is based on Saltelli (2002); the analytical truth is calculated based on Saltelli et al. (2004); the calculation of each sensitivity index is repeated 50 times to quantify sampling uncertainty. Each of the 50 experiments uses a different set of input-output sample data with size 10,000; and the Monte Carlo sample sizes are $N_1 = N_2 = 2{,}000$.

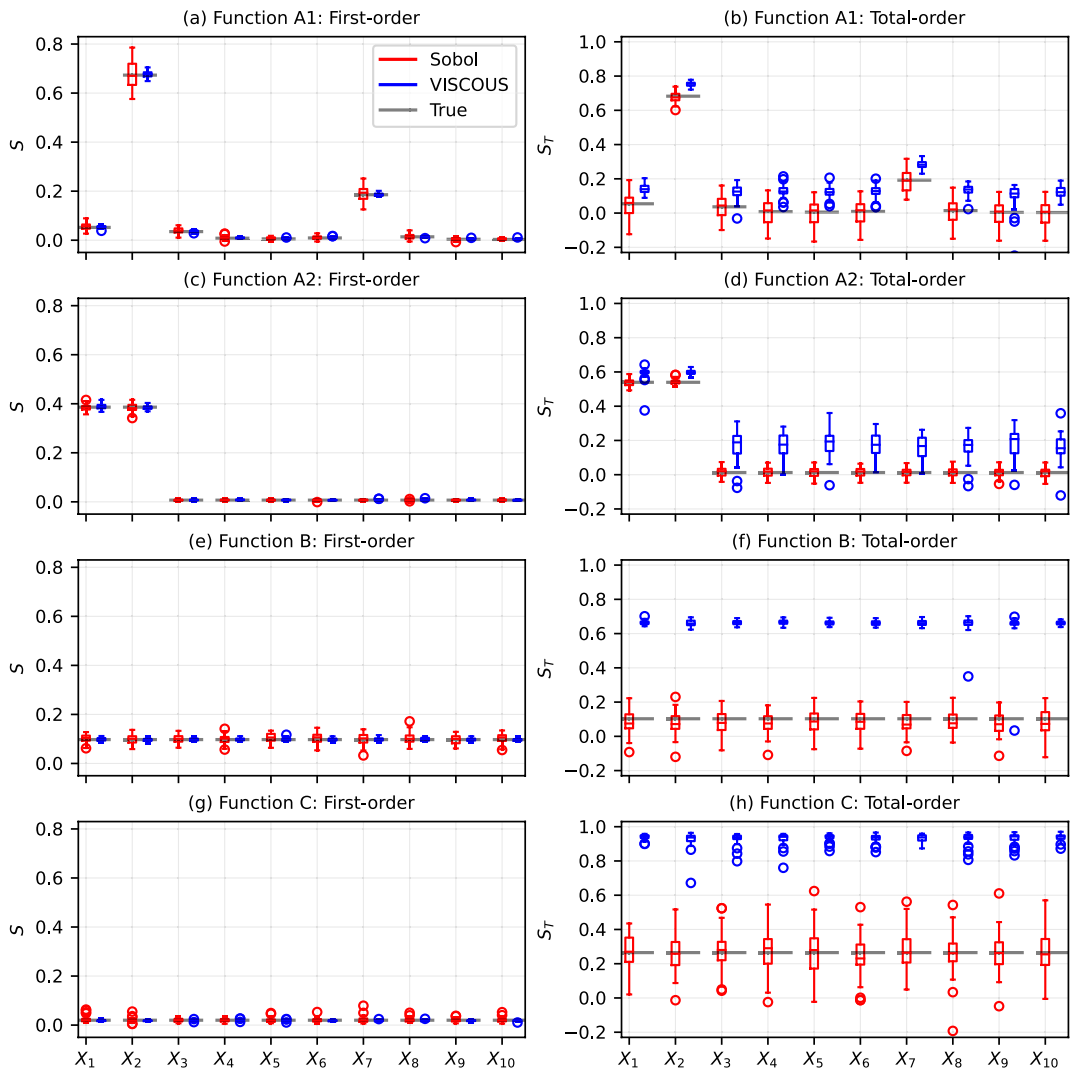

Figure 2. First- and total-order sensitivity index results of the Sobol' method, VISCOUS, and the analytical truth for the four functions of Table 1.

For the first-order sensitivity indices (Figures 2a, 2c, and 2e), VISCOUS generates results matching the truth for all variables in all functions, and its uncertainty of sensitivity estimates is smaller than Sobol's. For the total-order sensitivity indices (Figures 2b, 2d, and 2f), for functions A1 and A2, VISCOUS provides slightly higher sensitivity estimates than the truth; for functions B and C, VISCOUS provides quite different sensitivity estimates from the truth. For all functions, VISCOUS is correct in ordering the sensitivity of each variable. Therefore, if one is interested in first-order sensitivity and input factors ranking, VISCOUS is good at achieving this functionality.

To investigate why VISCOUS behaves differently between Type A functions and Type B and C functions, we examined the results of the Sobol' method. The Sobol' method produces many negative sensitivity indices when the total-order sensitivities approach zero (Figures 2b, 2d, and 2f). Negative sensitivity indices do not make theoretical sense and are instead the result of numerical artifacts in the estimation procedure. Moreover, the Sobol' method produces large uncertainties when the total-order sensitivities are the same across all dimensions (Figures 2f and 2h). These reveal the difficulty of calculating the total-order sensitivities when they are close to zero or the same, in other words, when functions are insensitive or equally sensitive to evaluated variables.

We hypothesize that the performance of VISCOUS in estimating sensitivity indices is affected by three factors: function dimension, input-output data size, and non-identifiability of GMCM inference. The following sections check them one by one.

3.3. Function Dimension

For all types of functions, high dimensionality (the number of function input variables) has no effect on first-order sensitivity estimation, which is a beauty of VISCOUS, but it poses a challenge to total-order sensitivity estimation. This is because the function dimension has different effects on the number of variables involved in GMCM (and GMCM inference) in first- and total-order sensitivity estimations.

Suppose the number of variables involved in GMCM inference is denoted as D. When calculating first-order sensitivity, D is always equal to two regardless of the function dimension, including the evaluated variable itself (X_i) and the evaluated output variable (Y). When calculating total-order sensitivity, D is equal to the function dimension, including all the input variables except the evaluated variable ($\boldsymbol{X}_{\sim i}$) plus the evaluated output variable (Y).

For a GMCM with K components and D variables, the number of GMCM parameters to estimate is equal to $K \times D \times D + K \times D + K$. These include K covariance matrices each of size $D \times D$, K mean vectors of length D, plus a component weight vector of length K. These GMCM parameter values are determined through GMCM inference. When calculating first-order sensitivity, the GMCM has 7K parameters to estimate because $D = 2$. When calculating the total-order sensitivity, the GMCM has $K \times d \times d + K \times d + K$ parameters to estimate because $D = d$ (d is the function dimension). This polynomial growth in the number of GMCM parameters can be a problem for high-dimensional functions because it becomes more challenging to produce a sufficient amount of sample data for making accurate GMCM inferences. For example, assuming a two-component GMM is used in GMCM (i.e., $K = 2$), when the number of X variables varies between 4, 6, 8, 10, 15, 20, 30, and 50, the corresponding number of GMCM parameters becomes 42, 86, 146, 222, 482, 842, 1,862, and 5,012.

To demonstrate the effect of function dimension on VISCOUS performance, we change the number of function variables from 4 to 50 to cover from low-dimensional to high-dimensional cases, and apply VISCOUS to all functions in Table 1. The experiment design is the same as in Section 3.2 except changing the number of function variables. The input-output data size remains 10,000 in all experiments. Figure 3 shows the VISCOUS sensitivity estimate errors. The error is calculated as the mean absolute sensitivity difference between the VISCOUS's result and the analytical truth across all X variables of a function.

For first-order sensitivity index, VISCOUS provides accurate estimates regardless of the function dimension, with a negligible error less than 0.005. For total-order sensitivity index, VISCOUS provides gradually worse estimates as the function dimension increases. Specifically, for Type A functions, when the function dimension is lower than 20, the total-order error increases slowly with the function dimension, and the error is acceptably small, less than 0.2. When the function dimension is higher than 20 (including 20), the total-order error increases rapidly, and the error is large. This difference between total-order errors and first-order errors indicates a potential limitation of the GMCM in capturing complex structures in high-dimensional problems.

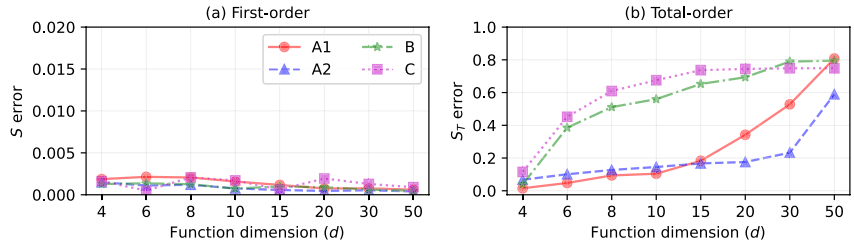

Figure 3. VISCOUS sensitivity estimate errors for different function dimensions. Functions A1, A2, B, and C are defined in Table 1. For each function, the number of X variables varies between 4 and 50; the input-output data size is 10,000; and the error is calculated as the mean absolute sensitivity difference between VISCOUS and the analytical truth across all X variables per function.

Type B and C functions have different total-order error curves from Type A functions. Rapid error increases are observed in Type B and C functions even when the function dimension is low (e.g., d from 4 to 6 in Figure 3b). This implies that, when estimating total-order sensitivity indices, VISCOUS faces difficulties other than high dimensionality, which is particularly influential in Type B and C functions. This will be explained in Section 3.5.

3.4. Input-Output Data Size

To investigate how many input-output data are needed for VISCOUS to provide accurate sensitivity estimates, we changed the input-output data sizes from 200 until 10,000, and applied the VISCOUS framework to functions of Table 1. The results are shown in Figure 4. The first-column of Figure 4 shows the effect of input-output data size on first-order sensitivity estimates. For all functions, the first-order sensitivity estimate error effectively reduces as the input-output data size increases. More importantly, the first-order sensitivity errors are tiny for all functions with even only 200 input-output data (i.e., less than 0.003). This is due to the low number of parameters to be estimated in the first-order sensitivity related GMCM inference as explained in Section 3.3.

The second column of Figure 4 shows the effect of input-output data size on total-order sensitivity estimates. For Type A functions, adding input-output data effectively improves the total-order sensitivity estimates of low- and medium-dimensional functions ($d < 20$). If taking 0.2 as an error threshold, 200 input-output data are needed for VISCOUS to produce accurate total-order sensitivity estimates for 4- and 6-dimensional problems. 400, 750, and 5,000 input-output data are needed for 8-, 10-, and 15-dimensional problems, respectively.

However, adding input-output data does not necessarily improve the total-order sensitivity estimates of high-dimensional functions ($d \geq 20$) given limited input-output data. For example, in function A1 (Figure 4b), the total-order error increases as the data size rises to 1,000 when $d = 30$, and to 10,000 when $d = 50$. This is caused by overfitting. When the GMCM being used is overly complex, the GMCM might fit noise in data rather than capturing the true underlying patterns. As such, the GMCM performs very well on the input-output data but cannot generalize and therefore performs poorly on new data (i.e., GMCM samples in Step 6). This result indicates that estimating total-order sensitivity of high-dimensional functions is difficult because a large amount of input-output data is needed to make good GMCM inferences (e.g., more than 10,000 data). In this case, we recommend applying a screening method (e.g., Elementary Effect Test (Pianosi et al., 2016)) followed by the calculation of the Sobol' total-order sensitivity index on a reduced number of input factors.

Figure 4 also shows that, increasing the input-output data size does not improve the total-order sensitivity estimates for Type B and C functions as effectively as it does for Type A functions. The next section will explain the factor that has a greater effect on the total-order sensitivity estimates of Type B and Type C functions than function dimension and sample size.

3.5. Non-Identifiability of GMCM Inference

We hypothesize that the poor performance of VISCOUS in total-order sensitivity estimates for Type B and Type C functions stems from the non-identifiability of GMCM inference. Non-identifiability is the inability to infer some or all parameters of interest from the available data (Renard et al., 2010; Wagener et al., 2001). There is a considerable body of work on non-identifiability in the control-engineering literature, in the context of dynamical models, spanning over 40 years (Dobre et al., 2012; Guillaume et al., 2019). The following explains the reason behind the non-identifiability of GMCM inference.

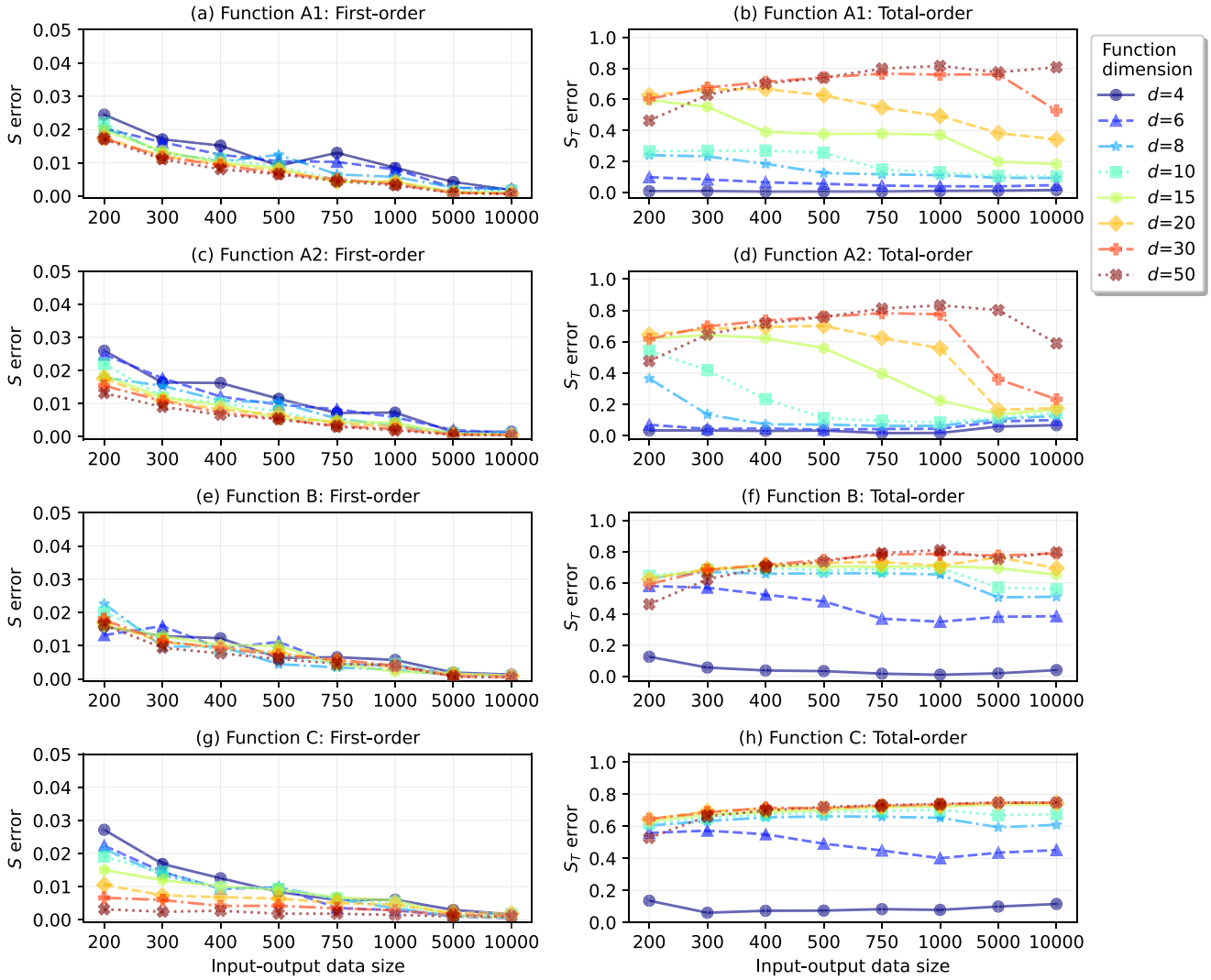

Figure 4. VISCOUS sensitivity estimate errors for different input-output data sizes. Functions A1, A2, B, and C are defined in Table 1. The number of X variables varies between 4 and 50.

3.5.1. Grouped Component Parameters in GMCM Inference

In GMCM, the log-likelihood of all input-output data is expressed by:

$$\log(\mathrm{P}(\mathbf{Z}|\boldsymbol{\Theta})) = \sum_{n=1}^{N} \log\left(\frac{\sum_{k=1}^{K} \lambda_k \cdot \phi\left(\mathbf{z}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\prod_{i=1}^{d} \phi_{z_{x_i}}\left(z_{x_i}\right) \cdot \phi_{z_y}(z_y)}\right) \tag{21}$$

where $\boldsymbol{\Theta} = [\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\Sigma}]$ is the GMCM parameter vector, N is the total number of input-output data used for GMCM inference, and $n = (1,\ldots,N)$. $\boldsymbol{z}_n = (\boldsymbol{z}_{n,x},\boldsymbol{z}_{n,y})$ is the nth inverse distribution values marginally based on the GMCM parameter vector ($\boldsymbol{\Theta}$) and the marginal CDF data ($\mathbf{u}_n$).

Consider a simple example of GMCM with two Gaussian components. The log-likelihood is:

$$\log(\mathrm{P}(\mathbf{Z}|\boldsymbol{\Theta})) = \sum_{n=1}^{N} \log\left(\frac{\lambda_1 \cdot \phi(\mathbf{z}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \lambda_2 \cdot \phi(\mathbf{z}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)}{\prod_{i=1}^{d} \phi_{z_{x_i}}\left(z_{x_i}\right) \cdot \phi_{z_y}(z_y)}\right) \tag{22}$$

Assuming the two Gaussian components are independent, the log-likelihood function can be re-parameterized as:

$$\log(\mathrm{P}(\mathbf{Z}|\boldsymbol{\Theta})) = \sum_{n=1}^{N}\Big\{\log(\phi(\mathbf{z}_n|\boldsymbol{\mu}_{EM},\boldsymbol{\Sigma}_{EM})) - \sum_{n=1}^{d}\log\Big(\phi\Big(z_{n,x_i}|\mu_{EM,z_{x_i}},\sigma^2_{EM,z_{x_i}}\Big)\Big) - \log\Big(\phi\Big(z_{n,y}|\mu_{EM,z_y},\sigma^2_{EM,z_y}\Big)\Big)\Big\}$$

$$\text{where } \boldsymbol{\mu}_{EM} = \lambda_1\boldsymbol{\mu}_1 + \lambda_2\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{EM} = \lambda_1^2\boldsymbol{\Sigma}_1 + \lambda_2^2\boldsymbol{\Sigma}_2 \tag{23}$$

The re-parameterized log-likelihood function depends on the weighted sum of $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$, not on the individual $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$. Therefore, $\boldsymbol{\mu}_{EM}$ and $\boldsymbol{\Sigma}_{EM}$ are identifiable and their inference problem is well posed, but the individual $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$ are not identifiable.

However, VISCOUS needs well-defined inference on the individual component parameters $\boldsymbol{\Theta}_k$. This is because to compute the conditional expectations in variance-based sensitivity indices, both the joint and the marginal distributions of the GMCM are needed (see Equations 15 and C1). The following explains why the non-identifiability has the greatest effect on GMCM inference when the input variables are equally sensitive.

3.5.2. Non-Exchangeable Priors in GMCM Inference

When facing non-identifiability, the strength of the prior information determines if the GMCM inference problem is well-posed (Renard et al., 2010). An inference problem is considered well-posed if it satisfied the following three criteria: a solution must exist, should be unique, and should depend continuously on the given data and assumptions.

The use of non-exchangeable priors can help yield a well-posed GMCM inference problem. Here the non-exchangeable priors mean that the priors for one Gaussian component are distinctly different from the priors for all other Gaussian components:

$$\mu_k \neq \mu_{k'}.\,Or,\,\left[\mu_{k,x},\mu_{k,y}\right] \neq \left[\mu_{k',x},\mu_{k',y}\right]$$

$$\boldsymbol{\Sigma}_k \neq \boldsymbol{\Sigma}_{k'}.\,Or,\,\left[\boldsymbol{\Sigma}_{k,xx}\,\boldsymbol{\Sigma}_{k,xy}\,\boldsymbol{\Sigma}_{k,yx}\,\sigma^2_{k,y}\right] \neq \left[\boldsymbol{\Sigma}_{k',xx}\,\boldsymbol{\Sigma}_{k',xy}\,\boldsymbol{\Sigma}_{k',yx}\,\sigma^2_{k',y}\right] \tag{24}$$

where k and k' represent two different Gaussian components of the GMM (k,$k' \in [1,..,K]$, $k \neq k'$). Otherwise, if $\mu_k = \mu_{k'}$ and $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_{k'}$, then the two priors are exchangeable between the kth and k'th components.

The challenge in generating non-exchangeable priors exists in functions that are equally sensitive to input variables. The equally sensitive variables have the same distribution and same interaction with other variables (including y), the prior information on these variable dimensions is very similar or even the same. If the data used for GMCM inference induce exchangeable priors and cannot discriminate between components, then the data cannot discriminate between the individual component parameters. In this case, it is impossible for any inference algorithm to explicitly discriminate these component parameters.

The higher the function dimension is, the more difficult it is to generate non-exchangeable priors for the equally sensitive input variables. This explains why the total-order sensitivities of VISCOUS deteriorate much faster in Type B and C functions than in Type A functions as the function dimension increases (see Figure 3). VISCOUS currently uses the k-means method to generate priors for GMCM parameters. Appendix E lists approaches to generating non-exchangeable prior information, though applying these approaches is out of scope of this study.

4. pyVISCOUS

pyVISCOUS is the open-source Python implementation of VISCOUS, available at https://github.com/CH-Earth/pyviscous.git (Liu et al., 2023). It is developed to streamline the application of VISCOUS. pyVISCOUS offers straightforward installation options - available both as a Python package via pip or directly from the source. We also provide example notebooks demonstrating the utilization of pyVISCOUS across the Rosenbrock function, four Sobol' functions of Table 1, and a real case study of the Bow at Banff basin, Alberta, Canada. Each example notebook includes well-documented code, guiding users on generating input-output data, setting up and running VISCOUS, and evaluating sensitivity index results.

5. Conclusions

VISCOUS is a variance-based global sensitivity analysis (GSA) framework developed by Sheikholeslami et al. (2021). As a "given-data" method, VISCOUS leverages existing model input and output data (e.g., parameters and responses of water system models) to provide useful approximations of the first- and total- order Sobol' sensitivity indices. The input-output data do not need to follow any specific sampling strategies and thus can be from the previous model runs generated for other modeling purposes, such as calibration and uncertainty analysis. Also, there are no enforced structure assumptions on the input-output data, which enhances VISCOUS's flexibility and applicability to models with complex interactions.

This research has three innovative contributions. First, we improve the VISCOUS methodology by refining the GMCM marginal densities based on the GMCM formula. Second, we conduct comprehensive evaluations of VISCOUS using three types of generic functions and highlight general problems with the application of GSA methods to water system models (e.g., dimensionality challenges associated with computing total-order sensitivity index). Last, we provide a didactic example (Appendix D) and an open-source Python code, pyVISCOUS, to help people understand and apply VISCOUS. pyVISCOUS is model-independent and can be applied with user-provided input-output data.

Our evaluation shows that the performance of VISCOUS is affected by three factors: function dimension, input-output data size, and non-identifiability. VISCOUS is powerful in estimating the first-order sensitivity using a small input-output data set, such as 200 in this study. This holds true across various function dimensions, as VISCOUS is inherently not affected by the function dimension in first-order sensitivity estimation. Moreover, VISCOUS is always correct in ranking input variables in both first- and total-order sensitivity terms regardless of function dimension and input-output data size.

For functions that are differently sensitive to input variables (Type A function, which are common in water system models), VISCOUS can provide good total-order sensitivity estimates for low- and medium-dimensional functions using limited input-output data (e.g., 10,000 or fewer). For instance, in this study, VISCOUS needs only 200 input-output data for 4- and 6-dimensional problems, and 400, 750, and 5,000 input-output data for 8-, 10-, and 15-dimensional problems, respectively. However, like other GSA methods, VISCOUS has difficulties in estimating total-order sensitivities for high-dimensional functions or models. This is because the number of GMCM parameters grows in a polynomial manner with the function dimension, and it is difficult to produce sufficient input-output data to make good GMCM inferences. Therefore, it is advisable to use VISCOUS when the function dimension is not very high (e.g., less than 20). When the function dimension is high, we recommend applying a screening method followed by the calculation of the Sobol' total-order sensitivity index on a reduced number of input factors.

For functions that are equally sensitive to input variables (Type B and C functions, which are rare in water system models), VISCOUS faces a greater challenge than function dimension and data size in total-order sensitivity estimation, that is, the non-identifiability of GMCM inference. The GMCM parameters are grouped in inference, so the individual component parameters are not identifiable. In this context, if a function is equally sensitive to its input variables, the prior information on these variable dimensions is highly exchangeable and cannot be discriminated between the GMCM components. This adds complexity and subjectivity to the GMCM inference. While well-posedness is still achievable, careful consideration and justification of the exchangeable priors are necessary to ensure the validity and robustness of the inference results. VISCOUS currently uses the k-means method to generate priors, and our evaluation confirms that k-means does not perform well for Type B and C functions. Future work is needed to incorporate the method of creating non-exchangeable priors into GMCM inference, so it can handle functions with equally important variables.

We also invite discussion and collaboration with others interested in related issues of sensitivity and uncertainty analysis for computationally expensive models. We seek collaborations to assess pyVISCOUS's effectiveness in large samples of model types and study locations across a variety of hydroclimatic and environmental regimes. This will further help us test, improve, and modify the proposed sensitivity analysis framework.

Appendix A: More Details About the Gaussian Mixture Model (GMM)

This appendix is to show more details about the Gaussian component, GMM conditional and marginal functions. As in Equation 9, the GMM PDF is expressed as:

$$f^{GMM}_{\mathbf{Z}_x,Z_y}(\mathbf{z}_x, z_y) = \sum_{k=1}^{K} \lambda_k \cdot \phi\left(\mathbf{z}_x, z_y | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \tag{A1}$$

where ϕ is the PDF of a multivariate Gaussian distribution with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$.

The Gaussian mean and covariance of a Gaussian component are expressed as:

$$\boldsymbol{\mu}_k = \left[\boldsymbol{\mu}_{k,\mathbf{z}_x}, \mu_{k,z_y}\right] \tag{A2}$$

$$\boldsymbol{\Sigma}_k = \begin{bmatrix} \boldsymbol{\Sigma}_{k,\mathbf{z}_x\mathbf{z}_x} & \boldsymbol{\Sigma}_{k,\mathbf{z}_x z_y} \\ \boldsymbol{\Sigma}_{k,z_y\mathbf{z}_x} & \sigma^2_{k,z_y} \end{bmatrix} \tag{A3}$$

where $\boldsymbol{\Sigma}_{k,\mathbf{z}_x\mathbf{z}_x}$ is the covariance between $\mathbf{z}_x$, $\boldsymbol{\Sigma}_{k,\mathbf{z}_x z_y}$ is the covariance between $\mathbf{z}_x$ and z_y, and $\boldsymbol{\Sigma}_{k,\mathbf{z}_x z_y} = \boldsymbol{\Sigma}_{k,z_y\mathbf{z}_x}$. σ^2_{k,z_y} is the variance of z_y.

The GMM conditional PDF of Z_y given $\mathbf{Z}_x$, $f^{GMM}_{Z_y|\mathbf{Z}_x}$, is derived by:

$$f^{GMM}_{Z_y|\mathbf{Z}_x}(z_y|\mathbf{z}_x) = f_{\mathbf{X},Y}(\mathbf{z}_x, z_y) / f_{\mathbf{X}}(\mathbf{z}_x) \tag{A4}$$

where $f^{GMM}_{\mathbf{Z}_x}$ is the GMM marginal PDF of $\mathbf{Z}_x$ expressed as:

$$f^{GMM}_{\mathbf{Z}_x} = \sum_{k=1}^{K} \lambda_k \cdot \phi\left(\mathbf{z}_x | \boldsymbol{\mu}_{k,\mathbf{z}_x}, \boldsymbol{\Sigma}_{k,\mathbf{z}_x\mathbf{z}_x}\right) \tag{A5}$$

Appendix B: Modified Expectation-Maximization (EM) Algorithm

The modified EM algorithm is to maximize the log-likelihood in GMCM inference. The GMCM log-likelihood is expressed as:

$$\begin{aligned} \log(\mathrm{P}(\mathbf{Z}|\Theta)) &= \sum_{n=1}^{N} \log\left(\frac{\sum_{k=1}^{K} \lambda_k \cdot \phi(\mathbf{z}_n|\mu_k, \boldsymbol{\Sigma}_k)}{\prod_{i=1}^{d} \phi_{z_{x_i}}\left(z_{x_i}\right) \cdot \phi_{z_y}(z_y)}\right) \\ &= \sum_{n=1}^{N} \left\{ \log\left(\sum_{k=1}^{K} \lambda_k \cdot \phi(\mathbf{z}_n|\mu_k, \boldsymbol{\Sigma}_k)\right) - \sum_{i=1}^{d} \log\left(\phi_{z_{x_i}}\left(z_{x_i}\right)\right) - \log\left(\phi_{z_y}(z_y)\right) \right\} \end{aligned} \tag{B1}$$

where N is the total number of samples, $n = (1,\ldots,N)$. The parameter vector $\boldsymbol{\Theta}$ combines the weights, mean vectors and covariance matrices of all the Gaussian components.

Here we use a Python library called Copulas to perform the modified EM algorithm. The algorithm proceeds as follows (Tewari et al., 2011):

1. Initialize the parameter vector $\boldsymbol{\Theta}$ to a set of random values using the k-means method.
2. Calculate the inverse distribution values marginally ($\mathbf{z}_n$) given the parameter vector ($\boldsymbol{\Theta}$) and the marginal CDF data ($\mathbf{u}_n$). In the absence of a closed form expression of the inverse function, a linear interpolation is used to obtain the inverse values empirically.
3. Expectation (E) step: Compute the posterior probability of sample $\mathbf{z}_n$ belonging to each component. It is equal to the ratio of the Gaussian component probability to the sum of all Gaussian component probabilities:

$$\mathrm{P}(L_n = k|\mathbf{z}_n) = \frac{\lambda_k \cdot \phi(\mathbf{z}_n|\mu_k, \boldsymbol{\Sigma}_k)}{\sum_{k=1}^{K} \lambda_k \cdot \phi(\mathbf{z}_n|\mu_k, \boldsymbol{\Sigma}_k)} \tag{B2}$$

where L_n denotes the component label. Moreover, compute the log-likelihood $log(\mathrm{P}(\mathbf{Z}|\mathbf{\Theta}))$ based on Equation B1.

4. Maximization (M) step: Update the parameter vector $\mathbf{\Theta}$ using the just computed posterior probability $\mathrm{P}(L_n = k|\mathbf{z}_n)$ so that the log-likelihood can be maximized:

$$\hat{\lambda}_k = \frac{\sum_{n=1}^{N} \mathrm{P}(L_n = k|\mathbf{z}_n) \cdot \lambda_k}{N} \tag{B3}$$

$$\hat{\mu}_k = \frac{\sum_{n=1}^{N} \mathrm{P}(L_n = k|\mathbf{z}_n) \cdot \mathbf{z}_n}{\sum_{n=1}^{N} \mathrm{P}(L_n = k|\mathbf{z}_n)} \tag{B4}$$

$$\hat{\mathbf{\Sigma}}_k = \frac{\sum_{n=1}^{N} \mathrm{P}(L_n = k|\mathbf{z}_n) \cdot (\mathbf{z}_n - \hat{\mu}_k)^T \cdot (\mathbf{z}_n - \hat{\mu}_k)}{\sum_{n=1}^{N} \mathrm{P}(L_n = k|\mathbf{z}_n)} \tag{B5}$$

5. Iterate steps 2–4 until the log-likelihood converges.

Appendix C: Total-Order Sensitivity Index

Computing the total-order sensitivity index of X_i needs the input-output data $(\mathbf{x}_{\sim i}, y) = (x_1, .., x_{i-1}, x_{i+1}, \ldots, x_d, y)$. The conditional expectation of Y given the specific value $\mathbf{x}_{\sim i}$ is expressed as:

$$E(Y|\mathbf{X}_{\sim i} = \mathbf{x}_{\sim i}) \approx \int_0^1 y \cdot \frac{f^{GMM}_{\mathbf{Z}_{\sim x_i}, Z_y}\left(\mathbf{z}_{\sim x_i}, z_y\right)}{f^{GMM}_{\mathbf{Z}_{\sim x_i}}\left(\mathbf{z}_{\sim x_i}\right)} \cdot \frac{1}{\phi_{z_y}(z_y)} \mathrm{d}u_y \tag{C1}$$

To drop the dependence upon the specific value $\mathbf{x}_{\sim i}$, the variance of $E(Y|X_i)$ is estimated by integrating $E(Y|\mathbf{X}_{\sim i} = \mathbf{x}_{\sim i})$ over the probability density function of $\mathbf{X}_{\sim i}$, expressed as:

$$V(E(Y|\mathbf{X}_{\sim i})) = \int_{\Omega x_i} E^2(Y|\mathbf{X}_{\sim i} = \mathbf{x}_{\sim i}) \mathrm{d}x_i - \left[\int_{\Omega x_i} E(Y|\mathbf{X}_{\sim i} = \mathbf{x}_{\sim i}) \mathrm{d}x_i\right]^2 \tag{C2}$$

The total-order sensitivity index is computed based on $V(E(Y|\mathbf{X}_{\sim i}))$ and Equation 4.

In Monte Carlo-based approximations, the above two equations are estimated as follows. First, use the inferred GMCM to perform two rounds of sampling and generate the Monte Carlo samples (for example, see Equation C3.

$$\mathbf{z}_1^{MC} = \begin{pmatrix} \mathbf{z}^{MC}_{1,\sim x_i} & z^{MC}_{1,y} \\ \mathbf{z}^{MC}_{2,\sim x_i} & z^{MC}_{2,y} \\ \vdots & \vdots \\ \mathbf{z}^{MC}_{N_1,\sim x_i} & z^{MC}_{N_1,y} \end{pmatrix}$$

$$= \begin{pmatrix} z^{MC}_{1,x_1} & z^{MC}_{1,x_{i-1}} & z^{MC}_{1,x_{i+1}} & z^{MC}_{1,x_d} & z^{MC}_{1,y} \\ z^{MC}_{2,x_1} \cdots & z^{MC}_{2,x_{i-1}} & z^{MC}_{2,x_{i+1}} \cdots & z^{MC}_{2,x_d} & z^{MC}_{2,y} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ z^{MC}_{N_1,x_1} & z^{MC}_{N_1,x_{i-1}} & z^{MC}_{N_1,x_{i+1}} & z^{MC}_{N_1,x_d} & z^{MC}_{N_1,y} \end{pmatrix},$$

$$\mathbf{z}_2^{MC} = \begin{pmatrix} \mathbf{z}_{r_1,\sim x_i}^{MC} & z_{1,y}'^{MC} \\ \mathbf{z}_{r_1,\sim x_i}^{MC} & z_{2,y}'^{MC} \\ \vdots & \vdots \\ \mathbf{z}_{r_1,\sim x_i}^{MC} & z_{N_2,y}'^{MC} \end{pmatrix} = \begin{pmatrix} z_{r_1,x_1}^{MC} & z_{r_1,x_{i-1}}^{MC} & z_{r_1,x_{i+1}}^{MC} & z_{r_1,x_d}^{MC} & z_{1,y}'^{MC} \\ z_{r_1,x_1}^{MC} \cdots & z_{r_1,x_{i-1}}^{MC} & z_{r_1,x_{i+1}}^{MC} \cdots & z_{r_1,x_d}^{MC} & z_{2,y}'^{MC} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ z_{r_1,x_1}^{MC} & z_{r_1,x_{i-1}}^{MC} & z_{r_1,x_{i+1}}^{MC} & z_{r_1,x_d}^{MC} & z_{N_2,y}'^{MC} \end{pmatrix}, \tag{C3}$$

$$r_1 = [1, \ldots, N_1]$$

The conditional expectation, $E(Y|\mathbf{X}_{\sim i} = \mathbf{x}_{\sim i})$, is approximated by:

$$E\left(Y|\mathbf{X}_{\sim i} = \mathbf{x}_{r_1,\sim i}^{MC}\right) \approx \frac{1}{N_2} \sum_{r_2=1}^{N_2} F_y^{-1}\left(u_{r_2,y}^{MC}\right) \cdot \frac{f_{\mathbf{Z}_{\sim x_i}, Z_y}^{GMM}\left(\left(\mathbf{z}_{r_1,\sim x_i}^{MC}, z_{r_2,y}^{MC}\right)\right)}{f_{\mathbf{Z}_{\sim x_i}}^{GMM}\left(\mathbf{z}_{r_1,\sim x_i}^{MC}\right)} \cdot \frac{1}{\phi_{z_y}\left(z_{r_2,y}^{MC}\right)} \tag{C4}$$

The conditional variance, $V(E(Y|\mathbf{X}_{\sim i}))$, is approximated by:

$$V(E(Y|\mathbf{X}_{\sim i})) \approx \frac{1}{N_1} \sum_{r_1=1}^{N_1} E^2\left(Y|\mathbf{X}_{\sim i} = \mathbf{x}_{r_1,\sim i}^{MC}\right) - \left[\frac{1}{N_1} \sum_{r_1=1}^{N_1} E\left(Y|\mathbf{X}_{\sim i} = \mathbf{x}_{r_1,\sim i}^{MC}\right)\right]^2 \tag{C5}$$

The total-order sensitivity index can be computed based on Equations C4, C5, and 4.

Appendix D: A Didactic Example of Implementing the VISCOUS Framework

This section uses the two-parameter Rosenbrock function to demonstrate the implementation of the improved VISCOUS framework. This example is intended to help users to understand the details of the VISCOUS methodology, such as the Gaussian components and GMM, and hence help users to utilize the VISCOUS framework for their own applications.

The Rosenbrock function, also referred to as the Valley or Banana function, is a popular test problem for uncertainty analysis, sensitivity analysis, and optimization algorithms (Rosenbrock, 1960). In the two-dimensional form, the Rosenbrock function is defined as:

$$Y = 100\left(X_2 - X_1^2\right)^2 + (1 - X_1)^2, X_1, X_2 \in [-2, 2] \tag{D1}$$

where (X_1,X_2) are the two input variables in range of [−2,2]. The global minimum is at (x_1,x_2) = (1,1), where $y = 0$.

The Rosenbrock function over the domain $[-2,2]^2$ is shown in Figure D1. It involves a long steep valley and a gradually sloping floor. The Rosenbrock function in its two-dimensional form enables us to visualize the function itself and the implementation steps of VISCOUS.

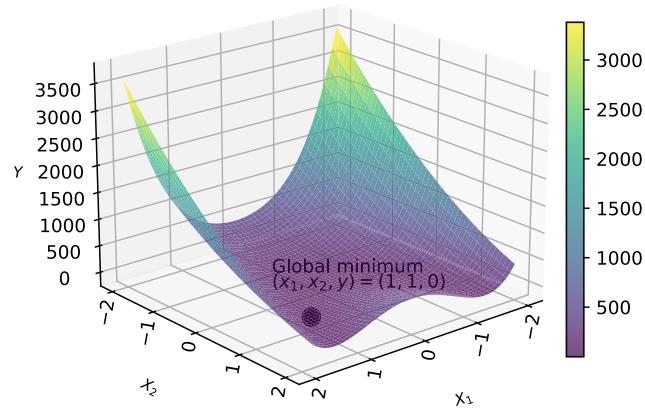

Figure D1. Rosenbrock function in its two-dimensional form.

D1. Part A. Data Preparation

Assume both variables (X_1,X_2) follow a uniform distribution between their lower and upper bounds. When we compute the first-order sensitivity index of X_1 for the Rosenbrock function, (X_1,Y) are included in the VISCOUS framework. We first generate 10,000 sets of (x_1,x_2) by randomly sampling from each variable's uniform distribution, and then calculate the corresponding y based on Equation D1. Following steps 1–3 in Section 2.4, we get three sets of data: input-output data (x_1,y), normalized data $\left(x'_1, y'\right)$, and empirical marginal CDF data (u_{x_1}, u_y). Figure D2 shows the scatter plot of the two-dimensional data among the three data sets.

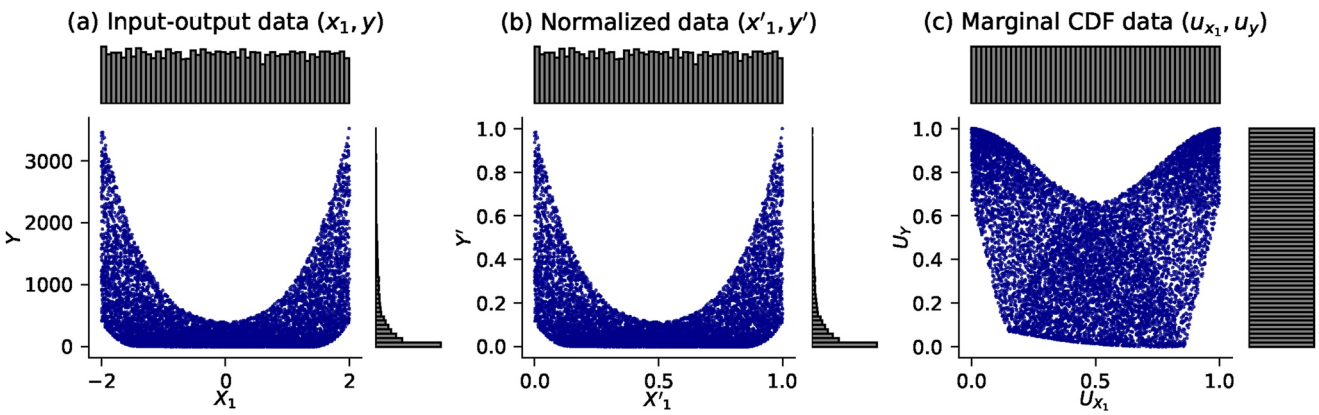

Figure D2. Scatter plot of the two-dimensional input-output data, normalized data, and empirical marginal CDF data. The histograms on the sides represent the marginal distribution.

D2. Part B. GMCM Inference

For ease of visualization, we first used two Gaussian components to estimate the GMCM ($K = 2$). The resulting visualization can help to understand what the Gaussian components are and how they are grouped together to form the GMM.

Based on two Gaussian components, the GMCM density function is expressed as:

$$c\left(u_{x_1}, u_y\right) = \frac{f^{GMM}\left(z_{x_1}, z_y\right)}{\phi_{z_{x_1}}\left(z_{x_1}\right) \cdot \phi_{z_y}(z_y)}$$

$$\text{where } f^{GMM}\left(z_{x_1}, z_y\right) = \lambda_1 \phi\left(z_{x_1}, z_y | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1\right) + \lambda_2 \phi\left(z_{x_1}, z_y | \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2\right)$$

$$\boldsymbol{\mu}_k = \left[\mu_{k,z_{x_1}}, \mu_{k,z_y}\right], \boldsymbol{\Sigma}_k = \begin{bmatrix} \sigma^2_{k,z_{x_1}} & \boldsymbol{\Sigma}_{k,z_{x_1} z_y} \\ \boldsymbol{\Sigma}_{k,z_y z_{x_1}} & \sigma^2_{k,z_y} \end{bmatrix}, k = [1,2] \tag{D2}$$

ϕ is the PDF of a bivariate Gaussian distribution with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$. Figure D3 shows the contour of each Gaussian component and the GMM. The weighted sum of the two bivariate Gaussian distributions (components) makes up the GMM. The two components are well separated and of different weights, and the mixture contour resembles the component contours.

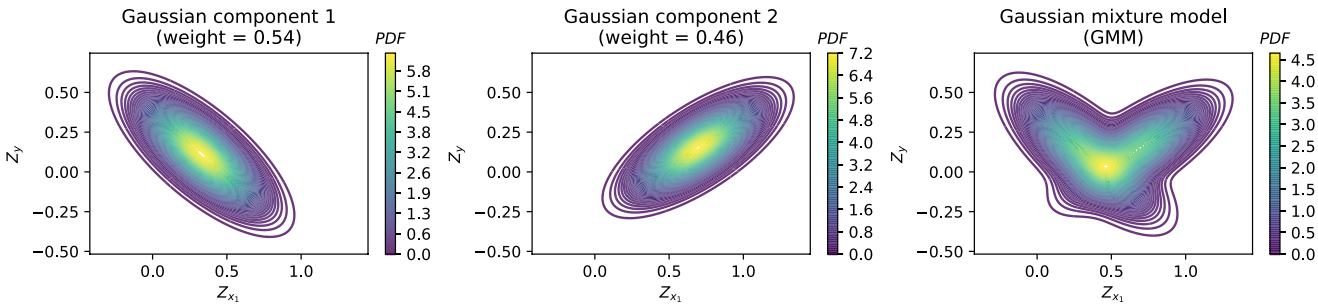

Figure D3. PDFs of two bivariate Gaussian components and the GMM.

The inferred parameter values of the two components are also provided:

$$\lambda_1 = 0.54, \lambda_2 = 0.46$$

$$\boldsymbol{\mu}_1 = \left[\mu_{1,z_{x_1}}, \mu_{1,z_y}\right] = [0.33, 0.11], \boldsymbol{\mu}_2 = \left[\mu_{2,z_{x_1}}, \mu_{2,z_y}\right] = [0.69, 0.15]$$

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} \sigma^2_{1,z_{x_1}} & \text{cov}_{1,z_{x_1} z_y} \\ \text{cov}_{1,z_y z_{x_1}} & \sigma^2_{1,z_y} \end{bmatrix} = \begin{bmatrix} 0.04 & -0.03 \\ -0.03 & 0.03 \end{bmatrix}, \boldsymbol{\Sigma}_2 = \begin{bmatrix} \sigma^2_{2,z_{x_1}} & \text{cov}_{2,z_{x_1} z_y} \\ \text{cov}_{2,z_y z_{x_1}} & \sigma^2_{2,z_y} \end{bmatrix} = \begin{bmatrix} 0.05 & 0.02 \\ 0.02 & 0.02 \end{bmatrix}$$

Figure D4 shows how the two-component GMM aligns with the input-output data. Recall that $\left(u_{x_1}, u_y\right)$ are the marginal CDF for the input-output data (see Figure D2c). We compute the inverse CDF of $\left(u_{x_1}, u_y\right)$ within the GMM, getting $\left(z_{x_1}, z_y\right)$. Figure D4a shows the distribution of $\left(z_{x_1}, z_y\right)$ data in the GMM. Next, we compute the corresponding joint probability density for each data point $\left(z_{x_1}, z_y\right)$ based on the PDF of the GMM (Figure D4b). These probability density values play a crucial role in GMCM inference, specifically serving as key inputs for calculating the log-likelihood in the utilized EM algorithm (see Equation B1). The log-likelihood of this two-components GMCM is 2,697.90. Lastly, to see the appearance of different Gaussian components, we label each $\left(z_{x_1}, z_y\right)$ data point with the Gaussian component to which it exhibits the highest

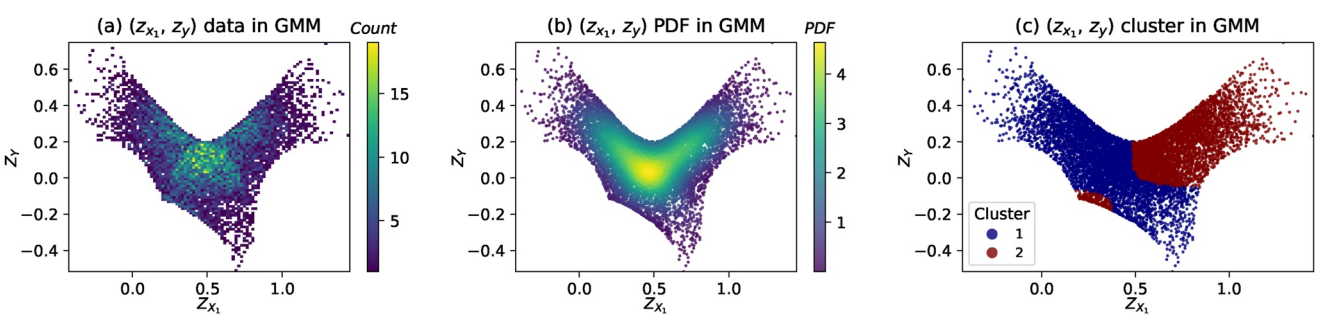

Figure D4. When using two Gaussian components, the histogram (panel a), joint PDF (panel b), and clustering (panel c) results for $\left(z_{x_1}, z_y\right)$.

probability. Figure D4c implicitly reveals each Gaussian component within the GMM, providing insights into their characteristics.

To get a better GMCM, we then repeated the process with different numbers of components up to $K = 9$, and used the AIC criterion and selected the optimal Gaussian component number of eight. Like Figure D4, Figure D5 shows the input-output data in the eight-component GMM. The GMCM log-likelihood increases to 3,814.87. The higher likelihood value represents the better inference result in the EM algorithm. Therefore, the eight-component based GMM better represents $\left(z_{x_1}, z_y\right)$ than the two-component based GMM. This result highlights the effects of the number of Gaussian components on GMM performance.

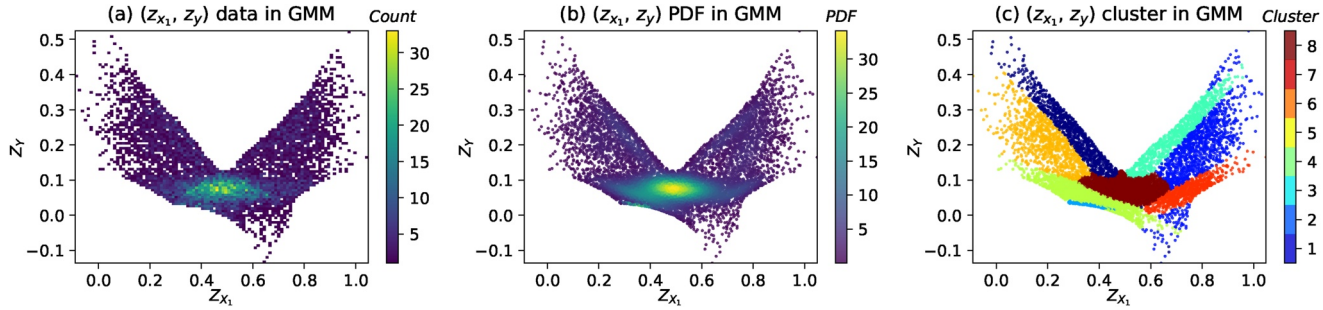

Figure D5. When using eight Gaussian components, the histogram (panel a), joint PDF (panel b), and clustering (panel c) results for $\left(z_{x_1}, z_y\right)$.

D3. Part C. Sensitivity Index Computation

Following the VISCOUS framework, we generated Monte Carlo samples $\left(z_{x_1}^{MC}, z_y^{MC}\right)$ based on the inferred GMCM. Then we calculated the first-order sensitivity based on Equations 3, 18 and 19, and calculated the total-order sensitivity using Equations C2, C3, and 4. To quantify the sampling uncertainty in VISCOUS, we repeated the entire processes 50 times to obtain 50 sets of sensitivity index results. Each experiment uses a different set of input-output sample data with size 10,000; and in sensitivity index estimation, the Monte Carlo sample sizes are $N_1 = N_2 = 2{,}000$.

For comparison, the Sobol' method of Saltelli (2002) was applied to the same 50 sets of sample data, getting 50 sets of Sobol' sensitivity index results. Figure D6 compares the results of VISCOUS and the Sobol' method. For both the first-order and total-order sensitivity indices, VISCOUS produces similar median sensitivity indices as the Sobol' method does.

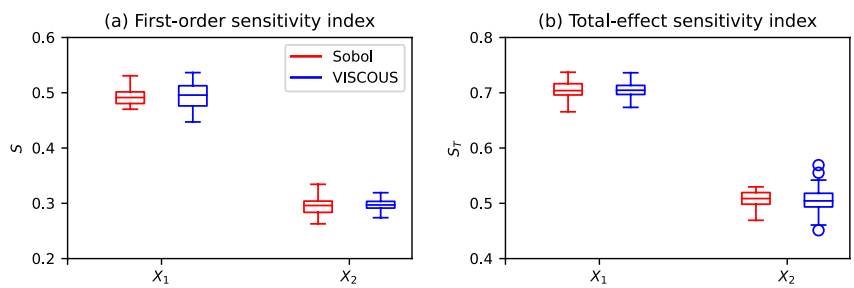

Figure D6. First- and total-order sensitivity index results of the Sobol' method and VISCOUS.

Appendix E: Approaches of Generating Non-Exchangeable Priors

In the literature, there are two main approaches for the GMCM inference to generating non-exchangeable priors. The first solution is to create strong constraints on the prior component means and covariances. Univariate

problems can follow Bartolucci (2005), multivariate problems can follow Di Zio et al. (2007), or use a hierarchical prior (Malsiner-Walli et al., 2017; Teh et al., 2006).

The second approach is ad hoc and includes two steps. It first estimates multiple Gaussian components, and then merges these components according to some criteria. Example criteria include the closeness of the means (Li, 2005), the modality of the obtained mixture density, the degree of overlapping measured by misclassification probabilities, and the entropy of the resulting partition (Malsiner-Walli et al., 2017).

Conflict of Interest

The authors declare no conflicts of interest relevant to this study.

Data Availability Statement

Access to the VISCOUS source code (pyVISCOUS) and all functions utilized in this work is available at https://github.com/CH-Earth/pyviscous.git (Liu et al., 2023).

Acknowledgments
The study is funded by the Global Water Futures programme. The authors wish to acknowledge with respect that they collectively reside on the traditional lands encompassed by Treaties 6, 7, and 8, and the homeland of the Métis Nation. We extend our gratitude to these nations for their enduring care and stewardship of this land and water. Furthermore, we pay homage to the ancestors and custodians of these sacred places for their profound and lasting contributions.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723. https://doi.org/10.1109/TAC.1974.1100705

Bartolucci, F. (2005). Clustering univariate observations via mixtures of unimodal normal mixtures. *Journal of Classification*, *22*(2), 203–219. https://doi.org/10.1007/s00357-005-0014-7

Borgonovo, E., Castaings, W., & Tarantola, S. (2012). Model emulation and moment-independent sensitivity analysis: An application to environmental modelling. *Environmental Modelling & Software*, *34*, 105–115. https://doi.org/10.1016/j.envsoft.2011.06.006

Box, G. E. P., & Meyer, R. D. (1986). An analysis for unreplicated fractional factorials. *Technometrics*, *28*(1), 11–18. https://doi.org/10.1080/00401706.1986.10488093

Demaria, E. M., Nijssen, B., & Wagener, T. (2007). Monte Carlo sensitivity analysis of land surface parameters using the Variable Infiltration Capacity model. *Journal of Geophysical Research*, *112*(D11), 11113. https://doi.org/10.1029/2006JD007534

Di Zio, M., Guarnera, U., & Rocci, R. (2007). A mixture of mixture models for a classification problem: The unity measure error. *Computational Statistics & Data Analysis*, *51*(5), 2573–2585. https://doi.org/10.1016/j.csda.2006.01.001

Dobre, S., Bastogne, T., Profeta, C., Barberi-Heyob, M., & Richard, A. (2012). Limits of variance-based sensitivity analysis for non-identifiability testing in high dimensional dynamic models. *Automatica*, *48*(11), 2740–2749. https://doi.org/10.1016/j.automatica.2012.05.004

Guillaume, J. H. A., Jakeman, J. D., Marsili-Libelli, S., Asher, M., Brunner, P., Croke, B., et al. (2019). Introductory overview of identifiability analysis: A guide to evaluating whether you have the right type of data for your modeling purpose. *Environmental Modelling & Software*, *119*, 418–432. https://doi.org/10.1016/j.envsoft.2019.07.007

Homma, T., & Saltelli, A. (1996). Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering & System Safety*, *52*, 1–17. https://doi.org/10.1016/0951-8320(96)00002-6

Hu, Z., & Mahadevan, S. (2019). Probability models for data-driven global sensitivity analysis. *Reliability Engineering & System Safety*, *187*, 40–57. https://doi.org/10.1016/j.ress.2018.12.003

Kucherenko, S., Feil, B., Shah, N., & Mauntz, W. (2011). The identification of model effective dimensions using global sensitivity analysis. *Reliability Engineering & System Safety*, *96*(4), 440–449. https://doi.org/10.1016/j.ress.2010.11.003

Li, J. (2005). Clustering based on a multilayer mixture model. *Journal of Computational & Graphical Statistics*, *14*(3), 547–568. https://doi.org/10.1198/106186005X59586

Liu, H., Clark, M. P., Gharari, S., Sheikholeslami, R., Freer, J., Knoben, W. J. M., et al. (2023). pyVISCOUS (2.2.1) [Software]. Zenodo. https://doi.org/10.5281/zenodo.10205100

Malsiner-Walli, G., Frühwirth-Schnatter, S., & Grün, B. (2017). Identifying mixtures of mixtures using Bayesian estimation. *Journal of Computational & Graphical Statistics*, *26*(2), 285–295. https://doi.org/10.1080/10618600.2016.1200472

Markstrom, S. L., Hay, L. E., & Clark, M. P. (2016). Towards simplification of hydrologic modeling: Identification of dominant processes. *Hydrology and Earth System Sciences*, *20*(11), 4655–4671. https://doi.org/10.5194/hess-20-4655-2016

Norton, J. (2015). An introduction to sensitivity assessment of simulation models. *Environmental Modelling & Software*, *69*, 166–174. https://doi.org/10.1016/j.envsoft.2015.03.020

Nossent, J., Elsen, P., & Bauwens, W. (2011). Sobol' sensitivity analysis of a complex environmental model. *Environmental Modelling & Software*, *26*(12), 1515–1525. https://doi.org/10.1016/J.ENVSOFT.2011.08.010

Pianosi, F., Beven, K., Freer, J., Hall, J. W., Rougier, J., Stephenson, D. B., & Wagener, T. (2016). Sensitivity analysis of environmental models: A systematic review with practical workflow. *Environmental Modelling & Software*, *79*, 214–232. https://doi.org/10.1016/j.envsoft.2016.02.008

Plischke, E., Borgonovo, E., & Smith, C. L. (2013). Global sensitivity measures from given data. *European Journal of Operational Research*, *226*(3), 536–550. https://doi.org/10.1016/j.ejor.2012.11.047

Razavi, S., & Gupta, H. V. (2015). What do we mean by sensitivity analysis? The need for comprehensive characterization of "global" sensitivity in earth and environmental systems models. *Water Resources Research*, *51*(5), 3070–3092. https://doi.org/10.1002/2014WR016527

Razavi, S., Jakeman, A., Saltelli, A., Prieur, C., Iooss, B., Borgonovo, E., et al. (2021). The future of sensitivity analysis: An essential discipline for systems modeling and policy support. *Environmental Modelling & Software*, *137*, 104954. https://doi.org/10.1016/j.envsoft.2020.104954

Renard, B., Kavetski, D., Kuczera, G., Thyer, M., & Franks, S. W. (2010). Understanding predictive uncertainty in hydrologic modeling: The challenge of identifying input and structural errors. *Water Resources Research*, *46*(5), W05521. https://doi.org/10.1029/2009WR008328

Rosenbrock, H. H. (1960). An automatic method for finding the greatest or least value of a function. *The Computer Journal*, *3*, 175–184. https://doi.org/10.1093/comjnl/3.3.175

Saltelli, A. (2002). Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications*, *145*(2), 280–297. https://doi.org/10.1016/s0010-4655(02)00280-1

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., et al. (2008). Global sensitivity analysis. *The Primer*. https://doi.org/10.1002/9780470725184

Saltelli, A., & Sobol, I. M. (1995). About the use of rank transformation in sensitivity analysis of model output. *Reliability Engineering & System Safety*, *50*(3), 225–239. https://doi.org/10.1016/0951-8320(95)00099-2

Saltelli, A., Tarantola, S., Campolongo, F., & Ratto, M. (2004). *Sensitivity analysis in practice: A guide to assessing scientific models*. Google eBook.

Sheikholeslami, R., Gharari, S., Papalexiou, S. M., & Clark, M. P. (2021). VISCOUS: A variance-based sensitivity analysis using copulas for efficient identification of dominant hydrological processes. *Water Resources Research*, *57*(7). https://doi.org/10.1029/2020wr028435

Sheikholeslami, R., & Razavi, S. (2020). A fresh look at variography: Measuring dependence and possible sensitivities across geophysical systems from any given data. *Geophysical Research Letters*, *47*(20). https://doi.org/10.1029/2020GL089829

Sheikholeslami, R., Razavi, S., Gupta, H. V., Becker, W., & Haghnegahdar, A. (2019). Global sensitivity analysis for high-dimensional problems: How to objectively group factors and measure robustness and convergence while reducing computational cost. *Environmental Modelling & Software*, *111*, 282–299. https://doi.org/10.1016/J.ENVSOFT.2018.09.002

Singh, A. (2019). Gaussian mixture models | clustering algorithm Python. Retrieved from https://www.analyticsvidhya.com/blog/2019/10/gaussian-mixture-models-clustering/

Sklar, A. (1959). Fonctions de répartition à N dimensions et leurs marges. *Publ. L'Institut Stat. L'Universit{é} Paris*, *8*, 229–231.

Sobol, I. M. (2001). Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, *55*(1–3), 271–280. https://doi.org/10.1016/S0378-4754(00)00270-6

Stanfill, B., Mielenz, H., Clifford, D., & Thorburn, P. (2015). Simple approach to emulating complex computer models for global sensitivity analysis. *Environmental Modelling & Software*, *74*, 140–155. https://doi.org/10.1016/j.envsoft.2015.09.011

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, *101*(476), 1566–1581. https://doi.org/10.1198/016214506000000302

Tewari, A., Giering, M. J., & Raghunathan, A. (2011). Parametric characterization of multimodal distributions with non-Gaussian modes. *Proc. - IEEE Int. Conf. Data Mining, ICDM*, 286–292. https://doi.org/10.1109/ICDMW.2011.135

van Griensven, A., Meixner, T., Grunwald, S., Bishop, T., Diluzio, M., & Srinivasan, R. (2006). A global sensitivity analysis tool for the parameters of multi-variable catchment models. *Journal of Hydrology*, *324*(1–4), 10–23. https://doi.org/10.1016/J.JHYDROL.2005.09.008

Vrieze, S. I. (2012). Model selection and psychological theory: A discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). *Psychological Methods*, *17*(2), 228–243. https://doi.org/10.1037/a0027127

Wagener, T., Boyle, D. P., Lees, M. J., Wheater, H. S., Gupta, H. V., & Sorooshian, S. (2001). A framework for development and application of hydrological models. *Hydrology and Earth System Sciences*, *5*(1), 13–26. https://doi.org/10.5194/hess-5-13-2001

Xu, L., & Jordan, M. I. (1996). On convergence properties of the EM algorithm for Gaussian mixtures. *Neural Computation*, *8*(1), 129–151. https://doi.org/10.1162/neco.1996.8.1.129