ADVANCING EARTH SYSTEM MODEL CALIBRATION: A DIFFUSION-BASED METHOD

Yanfang Liu
Oak Ridge National Laboratory
1 Bethel Valley Rd, Oak Ridge, TN, USA
liuy@ornl.gov

Dan Lu
Oak Ridge National Laboratory
1 Bethel Valley Rd, Oak Ridge, TN, USA
lud1@ornl.gov

Zezhong Zhang
Oak Ridge National Laboratory
1 Bethel Valley Rd, Oak Ridge, TN, USA
zhangz2@ornl.gov

Feng Bao
Florida State University
Tallahassee, FL, USA
bao@math.fsu.edu

Guannan Zhang
Oak Ridge National Laboratory
1 Bethel Valley Rd, Oak Ridge, TN, USA
zhangg@ornl.gov

ABSTRACT

Understanding of climate impact on ecosystems globally requires site-specific model calibration. Here we introduce a novel diffusion-based uncertainty quantification (DBUQ) method for efficient model calibration. DBUQ is a score-based diffusion model that leverages Monte Carlo simulation to estimate the score function and evaluates a simple neural network to quickly generate samples for approximating parameter posterior distributions. DBUQ is stable, efficient, and can effectively calibrate the model given diverse observations, thereby enabling rapid and site-specific model calibration on a global scale. This capability significantly advances Earth system modeling and our understanding of climate impacts on Earth systems. We demonstrate DBUQ's capability in E3SM land model calibration at the Missouri Ozark AmeriFlux forest site. Both synthetic and real-data applications indicate that DBUQ produces accurate parameter posterior distributions similar to those generated by Markov Chain Monte Carlo sampling but with 30X less computing time. This efficiency marks a significant stride in model calibration, paving the way for more effective and timely climate impact analyses.

1 INTRODUCTION AND MOTIVATION

Land surface models, such as the Energy Exascale Earth System Model (E3SM) [1; 2], Land Model (ELM), serve as an essential tool in enhancing our understanding of how ecosystems respond to climate change and this understanding is crucial for developing strategies to mitigate and adapt to the ongoing and future effects of the change. ELM simulates key processes such as water dynamics, energy exchanges, and biogeochemical cycles occurring on terrestrial surfaces. It involves a large number of parameters [3], many of which are not measured and default values are usually used either from surveys of broadly defined plant functional types or by benchmarking the model simulations against global data sets. However, due to differences in climate, soil, and vegetation types between geographic regions, assigning uniform values to these site-specific parameters resulted in inaccurate model simulations at individual sites. Additionally, the deterministic parameter values at a single site do not consider the parameter uncertainty where different parameter sets can produce similar model simulations. Therefore, an efficient uncertainty quantification (UQ) method is required to enable the site-by-site parameter estimation, thus improving model's predictability and advancing our understanding of climate impacts on ecosystems globally.

Markov Chain Monte Carlo (MCMC) sampling is a widely adopted method for estimating parameter uncertainty [4; 5; 6]. It involves generating a series of samples from the posterior probability density function (PDF) to quantify uncertainty. Ideally, with sufficient iterations, these samples converge to the true PDF. However, MCMC is notoriously computationally intensive, often necessitating hundreds of thousands of model evaluations, which are not fully parallelizable. Rapidly quantifying uncertainty in land surface modeling is crucial, as it underpins informed decision-making, adaptive management, and effective risk management and resource allocation in the face of pressing climate change. To mitigate these computational demands, machine learning (ML)-based UQ methods have been introduced. Some researchers employ ML to create fast surrogate models, accelerating model evaluations during MCMC sampling [7; 8; 9; 10]. Others leverage generative modeling techniques like normalizing flows for direct UQ problem-solving [11; 12; 13; 14; 15; 16]. Surrogate modeling demands a globally accurate surrogate across the entire parameter space and requires a new MCMC simulation whenever site-specific likelihood functions vary. Whereas, normalizing flows hinge on an invertible neural network structure, which requires costly computation of inverse mappings and Jacobian determinants. These limitations hinder their effective applications for ELM parameter estimation, underscoring the need for more efficient approaches.

In this study, we introduce a novel diffusion-based uncertainty quantification method (DBUQ) for efficient parameter estimation. Diffusion models are a class of generative ML models used to generate samples from a given data distribution [17; 18; 15]. The process begins by sampling from a prescribed noise distribution—typically an i.i.d. Gaussian distribution—and then iteratively transforming that sample via a learned denoiser, until it approximates a sample from the target distribution (here the posterior PDF of ELM parameters). The differences in the denoising process result in various diffusion models, with the score-based diffusion model standing out for its solid theoretical foundation and its capability to produce high-quality samples. This method uses a neural network (NN) to learn the score function and then repeatedly solves a reverse stochastic differential equation (SDE) using the learned score function to draw target samples. However, this method is computationally intensive due to the iterative reverse process required to generate each sample and the necessity of precise score estimation at every iteration. Additionally, diffusion models are typically trained in an unsupervised manner due to the lack of labeled data. The unsupervised learning of the score function requires storing a large number of stochastic paths of the forward SDE, which significantly increases the computational cost furthermore.

To address these challenges, we develop a training-free score estimation that leverages the Monte Carlo estimator for direct approximation of the score function. With the estimated score function, we then generate labeled data by solving an ordinary differential equation (ODE), instead of the expensive reverse SDE. Following this, we employ the generated labeled data to train a simple NN to learn the sample generator in a supervised manner and lastly evaluate the trained NN on the observation data to quickly generate target posterior samples. Our DBUQ method is stable, efficient, and can learn parameter posterior distributions given diverse observations, thereby enabling rapid, site-specific parameter estimation on a global scale. We validate the method's efficacy by applying it to the calibration of eight parameters in the ELM, using five years of latent heat flux measurements from the Missouri Ozark AmeriFlux forest site. The performance of DBUQ is evaluated by comparing its estimated parameter distributions against those obtained from MCMC sampling.

2 DBUQ FOR ELM CALIBRATION

We estimate ELM parameters using annual average latent heat fluxes (LH) data collected at the Missouri Ozark AmeriFlux site [19] from 2006 to 2010 (i.e., five LH variables). ELM involves more than 60 parameters and eight parameters are responsible for more than 80% of the variation in the LH [3]. These parameters control rooting distribution with depth (*rootb_par* [0.5,4.5]), the specific leaf area at the top of the canopy (*slatop* [0.01, 0.06]), the fraction of leaf nitrogen in RuBisCO (*flnr* [0.1, 0.4]), the fine root carbon:nitrogen ratio (*frootcn* [25, 65]), the fine root to leaf allocation ratio (*froot_leaf* [0.3, 1.8]), the base rate of maintenance respiration (*br_mr* [1.5e-6, 4.5e-6]), the critical day length to initiate autumn senescence (*crit_dayl* [35000, 55000]), and the phenology for carbon uptake (*crit_onset_gdd* [500, 1300]). The prior ranges of the parameters are listed above after the parameter names.

Given the observed LH data y, we aim to estimate posterior distribution $p(X|Y = y)$ of the eight parameters X, based on limited pairs of samples ($\mathcal{D}_s = \{(x_j, y_j)\}$) generated from the ELM simulations.

To efficiently solve this problem, our DBUQ method first formulates a parameterized generative model $X = F(Y, Z; \theta)$; next it trains a NN using a standard mean squared error (MSE) loss to estimate the model F; then for a given observation y, we can quickly generate numerous samples from the target distribution $p(X|Y = y)$ by evaluating F at standard-Gaussian-distributed samples Z. To construct the training data for the NN, we first generate 1000 pairs of samples $\mathcal{D}_s$. Then, we use a mini-batch-based Monte Carlo approach to estimate the score function based on $\mathcal{D}_s$. Next, we solve an ODE to generate the training data based on the score function estimate. Details of our DBUQ method are presented in Appendix A. For this eight-parameter estimation problem, we generate 2000 samples to approximate their posterior distribution.

We evaluate DBUQ's performance by comparing its estimated posterior distribution with MCMC results. For a fair comparison, we use the same number of prior samples $\mathcal{D}_s$ in the MCMC simulation. We first build a surrogate model based on $\mathcal{D}_s$ and then perform the MCMC sampling on the surrogate using the EMCEE algorithm [20]. Twenty chains are launched and each evolves 50000 samples. After discarding 40000 samples as a burn-in period, we select every 100^{th} of the remaining 10000 samples from each chain as the final set (i.e., 2000 samples) to approximate the parameter posterior distribution. After building the surrogate model, the MCMC sampling takes about 5 hours. In contrast, the entire implementation of our DBUQ method takes less than 10 minutes, where estimating the score function and solving the ODE take about 5 minutes and the training of NN to estimate the generative model F takes another 4 minutes. The time used in evaluation of F to generate the 2000 parameter posterior samples is negligible, less than a second. The speedup of DBUQ is 30X. More importantly, DBUQ solves an amortized Bayesian inference, i.e., after F is learned, for any observation y, we can quickly evaluate F to approximate $p(X|Y = y)$ without re-calculating the score function and solving the ODE. In contrast, MCMC simulation needs to be rerun for different observations due to the change of its likelihood function.

3 RESULTS

We apply the DBUQ method to both synthetic and real observations. In the synthetic case, we pick one ELM-simulated LH sample from $\mathcal{D}_s$ as a synthetic observation and use the corresponding parameter sample as the synthetic "truth" to evaluate the method's accuracy. In the real-data case, the parameters are calibrated using the real observations from the forest site. Figure 1 illustrates the parameter estimation results for DBUQ and MCMC in the synthetic case. Notably, both methods yield comparable posterior PDFs for marginal and joint distributions, aligning well with our domain knowledge. Specifically, the parameter *rootb_par* exhibits sensitivity to LH, as evidenced by its posterior PDF, which is effectively constrained by the LH observations, resulting in a distinctly narrow shape. Furthermore, the parameters *slatop* and *flnr* are inherently positively correlated, a relationship that is accurately reflected in their estimated joint distributions.

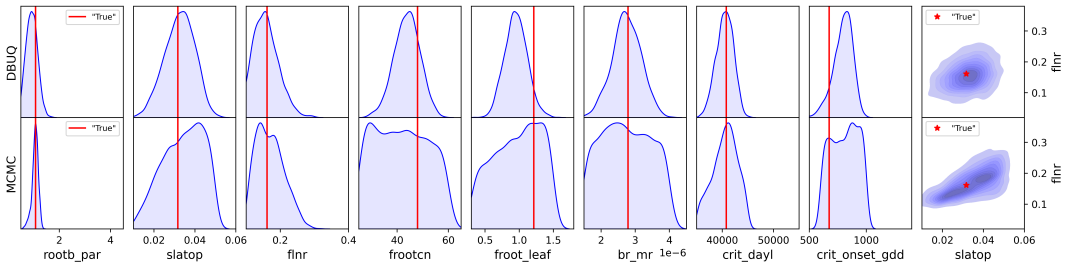

Figure 1: Parameter posterior distributions estimated by DBUQ and MCMC in the synthetic case, where the red color highlights the synthetic "true" value.

For nonlinear problems without known analytical solutions for parameter posteriors, the precise shape of their PDFs is unknown. To enhance the precision of our parameter estimation analysis, we further analyze the posterior PDFs of the simulated LH. The rationale is that: if the posterior uncertainty of the parameter is effectively captured, then the generated LH posterior samples should encompass the observed value closely. As depicted in Figure 2(a), the LH samples derived from both DBUQ and MCMC methods exhibit a tight distribution around the synthetic "true" observations, with their quantiles showing remarkable similarity. This pattern underscores the efficacy of DBUQ in quantifying

parameter uncertainty and solving inverse problems with precision. Notably, DBUQ not only mirrors the results obtained through MCMC sampling but achieves this with 30 times less computational time (10 minutes for DBUQ approximation vs. 5 hours for MCMC sampling). Moreover, once DBUQ learned the generative model F, it can instantaneously generate corresponding parameter posterior distributions for any given observations, all without the need for retraining the NN. To showcase this capability and to investigate the generalization properties of our DBUQ method further, we evaluate the generative model using a distinct set of synthetic observations. The findings, detailed in Appendix B, reaffirm DBUQ's accuracy for parameter estimations.

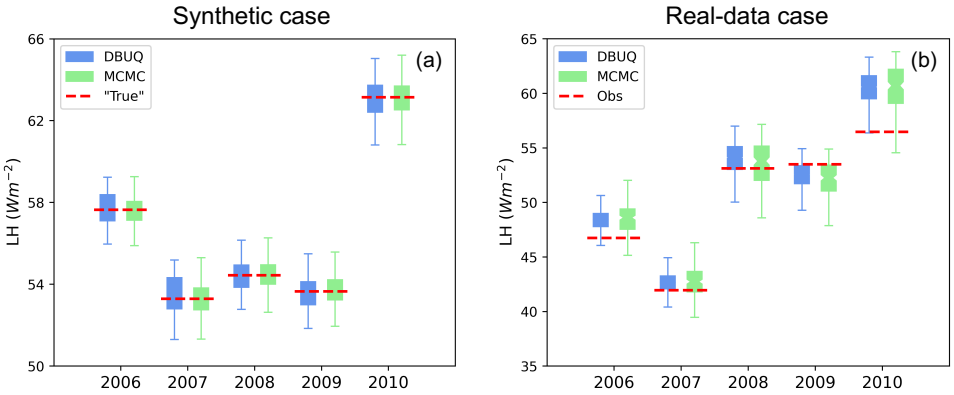

Figure 2: Boxplots summarize LH predictions from the parameter posterior samples of DBUQ and MCMC. "True" in (a) presents synthetic observation; Obs in (b) is real observation.

In the real-data case, the DBUQ method reuses the trained generative model F to compute parameter posterior distributions based on the actual observation data. The parameter estimation results are shown in Figure 3 and the generated LH posterior samples are summarized in Figure 2(b) along with the MCMC approximations. These figures clearly demonstrate that DBUQ's performance aligns closely with that of MCMC, both in terms of parameter UQ and LH predictions. The parameter posteriors generated by both methods exhibit analogous PDF shapes in both marginal and joint distributions. Moreover, it's noteworthy that the LH prediction samples produced by DBUQ and MCMC not only closely mirror each other but also effectively encapsulate the observed data (Figure 2(b)), reflecting the reasonable UQ and accurate predictions.

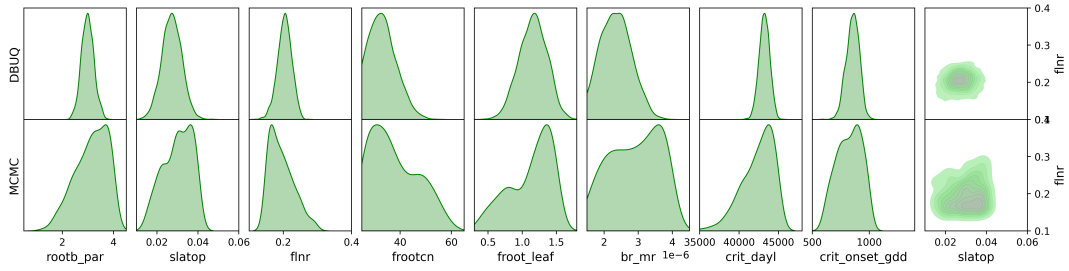

Figure 3: Parameter posterior distributions estimated by DBUQ and MCMC in the real-data case.

4 Conclusion and Future Work

We develop a diffusion-based uncertainty quantification (DBUQ) method and evaluate its performance for ELM model calibration at one AmeriFlux site. Our evaluations in both synthetic and real-data applications underscore DBUQ's robust capability to precisely quantify parameter uncertainty. The resultant posterior distributions can be reasonably explained with our domain knowledge, aligning closely with both synthetic benchmarks and predictive insights. A comparative analysis reveals that the posterior estimates generated by DBUQ closely match those derived from MCMC approximations, yet DBUQ stands out by drastically reducing the computational burden, necessitating 30 times less computational time. Moreover, once the generative model is trained, our DBUQ method quickly

generates parameter posterior samples for any given observations. This rapid inverse modeling and UQ capability significantly enhances site-specific model calibration on a global scale, marking a transformative step towards more efficient and timely assessments of climate impacts on our Earth systems. Looking ahead, we aim to extend the application of DBUQ across additional AmeriFlux sites, thereby augmenting the global predictability of the ELM model and enriching our understanding of the Earth's climate dynamics and its impacts. The integration of AI techniques like DBUQ in climate science holds profound potential for enhancing our response to climate change challenges.

REFERENCES

[1] L. R. Leung, D. C. Bader, M. A. Taylor, and R. B. McCoy, "An introduction to the e3sm special collection: Goals, science drivers, development, and analysis," *Journal of Advances in Modeling Earth Systems*, vol. 12, no. 11, p. e2019MS001821, 2020.

[2] J.-C. Golaz, P. M. Caldwell, L. P. Van Roekel, M. R. Petersen, Q. Tang, J. D. Wolfe, G. Abeshu, V. Anantharaj, X. S. Asay-Davis, D. C. Bader, S. A. Baldwin, G. Bisht, P. A. Bogenschutz, M. Branstetter, M. A. Brunke, S. R. Brus, S. M. Burrows, P. J. Cameron-Smith, A. S. Donahue, M. Deakin, R. C. Easter, K. J. Evans, Y. Feng, M. Flanner, J. G. Foucar, J. G. Fyke, B. M. Griffin, C. Hannay, B. E. Harrop, M. J. Hoffman, E. C. Hunke, R. L. Jacob, D. W. Jacobsen, N. Jeffery, P. W. Jones, N. D. Keen, S. A. Klein, V. E. Larson, L. R. Leung, H.-Y. Li, W. Lin, W. H. Lipscomb, P.-L. Ma, S. Mahajan, M. E. Maltrud, A. Mametjanov, J. L. McClean, R. B. McCoy, R. B. Neale, S. F. Price, Y. Qian, P. J. Rasch, J. E. J. Reeves Eyre, W. J. Riley, T. D. Ringler, A. F. Roberts, E. L. Roesler, A. G. Salinger, Z. Shaheen, X. Shi, B. Singh, J. Tang, M. A. Taylor, P. E. Thornton, A. K. Turner, M. Veneziani, H. Wan, H. Wang, S. Wang, D. N. Williams, P. J. Wolfram, P. H. Worley, S. Xie, Y. Yang, J.-H. Yoon, M. D. Zelinka, C. S. Zender, X. Zeng, C. Zhang, K. Zhang, Y. Zhang, X. Zheng, T. Zhou, and Q. Zhu, "The doe e3sm coupled model version 1: Overview and evaluation at standard resolution," *Journal of Advances in Modeling Earth Systems*, vol. 11, no. 7, pp. 2089–2129, 2019.

[3] D. Ricciuto, K. Sargsyan, and P. Thornton, "The impact of parametric uncertainties on biogeochemistry in the e3sm land model," *Journal of Advances in Modeling Earth Systems*, vol. 10, no. 2, pp. 297–319, 2018.

[4] T. Ziehn, M. Scholze, and W. Knorr, "On the capability of monte carlo and adjoint inversion techniques to derive posterior parameter uncertainties in terrestrial ecosystem models," *Global Biogeochemical Cycles*, vol. 26, no. 3, 2012.

[5] J. A. Vrugt, "Markov chain monte carlo simulation using the dream software package: Theory, concepts, and matlab implementation," *Environmental Modelling Software*, vol. 75, pp. 273–316, 2016.

[6] O. Hararuk, J. Xia, and Y. Luo, "Evaluation and improvement of a global land model against soil carbon data using a bayesian markov chain monte carlo method," *Journal of Geophysical Research: Biogeosciences*, vol. 119, no. 3, pp. 403–417, 2014.

[7] D. Lu and D. Ricciuto, "Efficient surrogate modeling methods for large-scale earth system models based on machine-learning techniques," *Geoscientific Model Development*, vol. 12, no. 5, pp. 1791–1807, 2019.

[8] S. Razavi, B. A. Tolson, and D. H. Burn, "Review of surrogate modeling in water resources," *Water Resources Research*, vol. 48, no. 7, 2012.

[9] T. Weber, A. Corotan, B. Hutchinson, B. Kravitz, and R. Link, "Technical note: Deep learning for creating surrogate models of precipitation in earth system models," *Atmospheric Chemistry and Physics*, vol. 20, no. 4, pp. 2303–2317, 2020.

[10] M. J. Asher, B. F. W. Croke, A. J. Jakeman, and L. J. M. Peeters, "A review of surrogate models and their application to groundwater modeling," *Water Resources Research*, vol. 51, no. 8, pp. 5957–5973, 2015.

[11] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," *Journal of Machine Learning Research*, vol. 22, no. 57, pp. 1–64, 2021.

[12] A. Khorashadizadeh, K. Kothari, L. Salsi, A. A. Harandi, M. de Hoop, and I. Dokmanić, "Conditional injective flows for bayesian imaging," *IEEE Transactions on Computational Imaging*, vol. 9, pp. 224–237, 2023.

[13] M. Yang, P. Wang, D. del Castillo-Negrete, Y. Cao, and G. Zhang, "A pseudo-reversible normalizing flow for stochastic dynamical systems with various initial distributions," *arXiv preprint arXiv:2306.05580*, 2023.

[14] Y. Liu, M. Yang, Z. Zhang, F. Bao, Y. Cao, and G. Zhang, "Diffusion-model-assisted supervised learning of generative models for density estimation," *Journal of Machine Learning for Modeling and Computing*, vol. 5, no. 1, pp. 25–38, 2023.

[15] F. Bao, Z. Zhang, and G. Zhang, "An ensemble score filter for tracking high-dimensional nonlinear dynamical systems," *arXiv-eprint, arXiv:2309.00983*, 2023.

[16] F. Bao, Z. Zhang, and G. Zhang, "A score-based nonlinear filter for data assimilation," *arXiv-eprint, arXiv:2306.09282*, 2023.

[17] L. Baldassari, A. Siahkoohi, J. Garnier, K. Solna, and M. V. de Hoop, "Conditional score-based diffusion models for bayesian inference in infinite dimensions," *arXiv preprint arXiv:2305.19147*, 2023.

[18] F. Bao, Z. Zhang, and G. Zhang, "A unified filter method for jointly estimating state and parameters of stochastic dynamical systems via the ensemble score filter," *arXiv-eprint, arXiv:2312.10503*, 2023.

[19] D. Baldocchi, E. Falge, L. Gu, R. Olson, D. Hollinger, S. Running, P. Anthoni, C. Bernhofer, K. Davis, R. Evans, J. Fuentes, A. Goldstein, G. Katul, B. Law, X. Lee, Y. Malhi, T. Meyers, W. Munger, W. Oechel, K. T. P. U, K. Pilegaard, H. P. Schmid, R. Valentini, S. Verma, T. Vesala, K. Wilson, and S. Wofsy, "Fluxnet: A new tool to study the temporal and spatial variability of ecosystem-scale carbon dioxide, water vapor, and energy flux densities," *Bulletin of the American Meteorological Society*, vol. 82, no. 11, pp. 2415 – 2434, 2001.

[20] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman, "emcee: The mcmc hammer," *Publications of the Astronomical Society of the Pacific*, vol. 125, p. 306–312, Mar 2013.

[21] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021.

[22] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Comput.*, vol. 23, p. 1661–1674, jul 2011.

[23] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, Curran Associates, Inc., 2020.

APPENDIX

A DIFFUSION-BASED UNCERTAINTY QUANTIFICATION (DBUQ) METHOD

In this work, we develop a score-based diffusion generative model for amortized Bayesian inference. Given an observation y, we want to sample from the posterior distribution $p(X|Y=y) \propto p(y|X)p(X)$ of parameter X to estimate its uncertainty, based on a finite number of i.i.d. samples $\mathcal{D}_s = \{(x_j, y_j)\}_{j=1}^J \subset \mathbb{R}^d \times \mathbb{R}^q$.

According to Bayes' rule,

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

where $p(X)$ is the parameter prior distribution and $p(Y|X)$ is the Gaussian likelihood function defined by

$$p(Y|X) \propto \exp\left(-\frac{1}{2}(Y-g(X))^\top \Sigma^{-1}(Y-g(X))\right), \tag{2}$$

with $g(X)$ being the physical model, e.g., the E3SM land model here. The dataset $\mathcal{D}_s$ includes $\{x_j\}_{j=1}^J$ generated from parameter prior distribution and $\{y_j = g(x_j)\}_{j=1}^J$ obtained by simulating the physical model $g(x)$.

In our DBUQ method, we build a parameterized generative model F and then learn the F and evaluate the F to generate the target posterior samples of X, i.e.,

$$X = F(Y, Z; \theta) \text{ with } Y \in \mathbb{R}^q,\ Z \in \mathbb{R}^d, \tag{3}$$

where F is the generative model that maps the observation variable Y and a reference variable Z to the parameter variable X, and θ represents parameters of the generative model. To learn and evaluate F, we first use a mini-batch-based Monte Carlo approach to estimate the score function based on the dataset $\mathcal{D}_s$. Then, we solve the reverse-time ODE in the diffusion model based on the estimated score function to generate labeled dataset $\{(x_j, y_j, z_j)\}_{j=1}^J$. Next, we train a feedforward neural network on these labeled pairs to learn the generative model F in Eq. (3). Lastly, given an observation y, we evaluate the trained F to generate numerous samples from the target distribution $p(X|Y=y)$ on Z drawn from the standard Gaussian. In the following sections, we introduce each step in details.

A.1 THE SCORE-BASED DIFFUSION MODEL

In this subsection, we provide a concise overview of score-based diffusion models. The score-based diffusion model includes two processes: the forward process and the backward process, both of which are defined in a standard-temporal domain $t \in [0,1]$. The forward process is given by a forward stochastic differential equation (SDE):

$$dZ_t = b(t)Z_t dt + \sigma(t)dW_t, \text{with } Z_0 = X|Y \text{ and } Z_1 = Z, \tag{4}$$

and the backward process is given by an associated reverse-time SDE:

$$dZ_t = \left[b(t)Z_t - \sigma^2(t)S(Z_t,t)\right]dt + \sigma(t)dB_t, \text{with } Z_0 = X|Y \text{ and } Z_1 = Z, \tag{5}$$

where W_t is a standard d-dimensional Brownian motion, B_t is the backward Brownian motion, $b(t)$ is the drift coefficient, $\sigma(t)$ is the diffusion coefficient, and $S(Z_t,t)$ is the score function.

The forward SDE in Eq. (4) is defined to gradually adding noise to the given initial distribution

$$Q(Z_0) = p(X|Y) \text{ with } Z_0 = X|Y,$$

to a tractable reference distribution $Q(Z_1) = \mathcal{N}(0, \mathbf{I}_d)$. It is shown in [21; 22; 23] that by properly choosing the drift and diffusion coefficients in the linear SDE of Eq. (4), the target distribution $Q(Z_0)$ can be transferred to the standard Gaussian distribution $\mathcal{N}(0, \mathbf{I}_d)$. In this work, $b(t)$ and $\sigma(t)$ in Eq. (4) are defined by

$$b(t) = \frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t} \quad \text{and} \quad \sigma^2(t) = \frac{\mathrm{d}\beta_t^2}{\mathrm{d}t} - 2\frac{\mathrm{d}\log\alpha_t}{\mathrm{d}t}\beta_t^2, \tag{6}$$

where the two processes α_t and β_t are defined by

$$\alpha_t = 1 - t, \quad \beta_t^2 = t \quad \text{for} \quad t \in [0,1]. \tag{7}$$

The definitions in Eq. (6) and Eq. (7) can ensure that the conditional density function $Q(Z_t|Z_0)$ for any fixed Z_0 is the following Gaussian distribution:

$$Q(Z_t|Z_0) = \mathcal{N}(\alpha_t Z_0, \beta_t^2 \mathbf{I}_d), \tag{8}$$

which leads to $Q(Z_1|Z_0) = Q(Z_1) = \mathcal{N}(0, \mathbf{I}_d)$. Thus, the reverse-time SDE in Eq. (5) can transform the terminal distribution $Q(Z_1) = \mathcal{N}(0, \mathbf{I}_d)$ to the initial distribution $Q(Z_0)$.

The score function in Eq. (5) is defined by

$$S(Z_t, t) := \nabla_z \log Q(Z_t), \tag{9}$$

which is uniquely determined by the initial distribution $Q(Z_0)$ and the coefficients $b(t)$, $\sigma(t)$ of the forward SDE in Eq. (4). Substituting $Q(Z_t) = \int Q(Z_t, Z_0)dZ_0 = \int Q(Z_t|Z_0)Q(Z_0)dZ_0$ into Eq. (9) and using the conditional density function $Q(Z_t|Z_0)$ in Eq. (8), we can rewrite the score function as

$$\begin{aligned}
S(Z_t, t) &= \nabla_z \log \left(\int_{\mathbb{R}^d} Q(Z_t|Z_0)Q(Z_0)dZ_0 \right) \\
&= \frac{\int_{\mathbb{R}^d} -\frac{Z_t - \alpha_t Z_0}{\beta_t^2} Q(Z_t|Z_0)Q(Z_0)dZ_0}{\int_{\mathbb{R}^d} Q(Z_t|Z_0')Q(Z_0')dZ_0'} \\
&= \frac{\int_{\mathbb{R}^d} -\frac{Z_t - \alpha_t [X|Y]}{\beta_t^2} Q\left(Z_t|[X|Y]\right) p(X|Y)d[X|Y]}{\int_{\mathbb{R}^d} Q\left(Z_t|[X'|Y]\right) p(X'|Y)d[X'|Y]} \\
&= \frac{\int_{\mathbb{R}^d} -\frac{Z_t - \alpha_t [X|Y]}{\beta_t^2} Q\left(Z_t|[X|Y]\right) \frac{p(Y|X)p(X)}{p(Y)}d[X|Y]}{\int_{\mathbb{R}^d} Q\left(Z_t|[X'|Y]\right) \frac{p(Y|X')p(X')}{p(Y)}d[X'|Y]} \\
&= \frac{\int_{\mathbb{R}^d} -\frac{Z_t - \alpha_t [X|Y]}{\beta_t^2} Q\left(Z_t|[X|Y]\right) p(Y|X)p(X)d[X|Y]}{\int_{\mathbb{R}^d} Q\left(Z_t|[X'|Y]\right) p(Y|X')p(X')d[X'|Y]} \\
&= \int_{\mathbb{R}^d} -\frac{Z_t - \alpha_t Z_0}{\beta_t^2} w_t(Z_t, Z_0)dZ_0,
\end{aligned} \tag{10}$$

where $p(X)$ is prior distribution in Eq. (1), $p(Y|X)$ is the likelihood function in Eq. (2), and the weight function $w_t(Z_t, Z_0)$ is defined by

$$w_t(Z_t, Z_0) = w_t(Z_t, [X|Y]) := \frac{Q\left(Z_t|[X|Y]\right) p(Y|X)p(X)}{\int_{\mathbb{R}^d} Q\left(Z_t|[X'|Y]\right) p(Y|X')p(X')d[X'|Y]}, \tag{11}$$

satisfying that $\int_{\mathbb{R}^d} w_t(Z_t, Z_0)dZ_0 = 1$. In our study, the integrals/expectations in the score function $S(Z_t, t)$ of Eq. (10) can be approximated by Monte Carlo estimators with the samples in dataset $\mathcal{D}_s = \{(x_j, y_j)\}_{j=1}^J$. The definition of the reverse-time SDE in Eq. (5) indicates that the samples in $\mathcal{D}_s$ are also generated from the target distribution $Q(Z_0)$, by initiated at samples from $Q_1(Z_1) = \mathcal{N}(0, \mathbf{I}_d)$. Thus, the score function of Eq. (10) can be estimated by

$$S(Z_t, t) \approx \bar{S}(Z_t, t) := \sum_{n=1}^{N} -\frac{Z_t - \alpha_t [x_{j_n}|y]}{\beta_t^2} \bar{w}_t(Z_t, [x_{j_n}|y]), \tag{12}$$

using a mini-batch of the dataset $\mathcal{D}_s$ with batch size $N \leq J$, denoted by $\{(x_{j_n}, y_{j_n})\}_{n=1}^N$, where the weight $w_t(Z_t, [x_{j_n}|y])$ is calculated by

$$w_t(Z_t, [x_{j_n}|y]) \approx \bar{w}_t(Z_t, [x_{j_n}|y]) := \frac{Q(Z_t|x_{j_n})p(y|x_{j_n})p(x_{j_n})}{\sum_{n'=1}^N Q(Z_t|x_{j_{n'}})p(y|x_{j_{n'}})p(x_{j_{n'}})}, \tag{13}$$

and $Q(Z_t|x_{j_n})$ is the Gaussian distribution given in Eq. (8). This means $w_t(Z_t, Z_0)$ can be estimated by the normalized probability density values $\{Q(Z_t|x_{j_n})p(y|x_{j_n})p(x_{j_n})\}_{n=1}^N$. In practice, the size of the mini-batch $\{(x_{j_n}, y_{j_n})\}_{n=1}^N$ can be flexibly adjusted to balance the efficiency and accuracy.

Under the assumption that the likelihood function is a Gaussian distribution and prior distribution is a uniform distribution, we have

$$\bar{w}_t(Z_t, [x_{j_n}|y]) = \frac{\exp\left\{\frac{-(Z_t-\alpha_t x_{j_n})^2}{2\beta_t^2}\right\}\exp\left\{-\frac{1}{2}(y-y_{j_n})^\top\Sigma^{-1}(y-y_{j_n})\right\}}{\sum_{n'=1}^N \exp\left\{\frac{-(Z_t-\alpha_t x_{j_{n'}})^2}{2\beta_t^2}\right\}\exp\left\{-\frac{1}{2}(y-y_{j_{n'}})^\top\Sigma^{-1}(y-y_{j_{n'}})\right\}}, \tag{14}$$

where $y_{j_n} = g(x_{j_n})$ is the output of the physical model.

A.2 Supervised learning of the generative model

In this subsection, we describe how to use the score approximation scheme given in Section A.1 to generate labeled data and use such data to train the generative model F. Due to the stochastic nature of the reverse-time SDE in Eq. (5), the relationship between the initial state $Z_0 = X|Y$ and the terminal state Z_1 is not deterministic or smooth. Thus, labeled data (x, y, z) can not be directly generated by Eq. (5). It has been shown that the ordinary differential equation (ODE) corresponding to Eq. (5), defined by

$$dZ_t = \left[b(t)Z_t - \frac{1}{2}\sigma^2(t)S(Z_t, t)\right]dt \;\text{ with }\; Z_0 = X|Y \text{ and } Z_1 = Z, \tag{15}$$

shares the same marginal probability density functions as the reverse-time SDE in Eq. (5). In addition, this ODE has unique solution and thus provides much smoother function relationship between the initial state Z_0 and terminal state Z_1. Thus, we adopt the ODE in Eq. (15) to generate labeled data (x, y, z).

Given the data set $\mathcal{D}_s = \{(x_j, y_j)\}_{j=1}^J \subset \mathbb{R}^d \times \mathbb{R}^q$, we now establish the conditional generative model $F(Y, Z; \theta)$ for Bayesian inference for a set of training samples

$$\{(x_m|y_m, z_m) \,:\, (x_m, y_m) \sim p(x, y) \text{ for } m = 1, \ldots, M\}, \tag{16}$$

where x_m's follow the prior distribution $p(x)$, y_m's are drawn from the likelihood function $p(y)$ and Z_m's are obtained by running the forward model.

Specifically, we first separately draw M random samples of variable Z, denoted by $\{z_1, \ldots, z_M\}$, from the standard Gaussian distribution, and M random samples of variable y, denoted by $\{y_1, \ldots, y_M\}$ from the marginal likelihood $p(Y)$. For $m = 1, \ldots, M$, we solve the ODE in Eq. (15) from $t = 1$ to $t = 0$ and collect the state $Z_0 = x_m|y_m$, where the score function is computed using Eq. (12), Eq. (13), and the dataset $\mathcal{D}_s = \{(x_j, y_j)\}_{j=1}^J$. The labeled training dataset is denoted by

$$\mathcal{D}_{\text{train}} := \{(x_m, y_m, z_m) : Z_0 = x_m|y_m, \;\text{ for }\; m = 1, \ldots, M\}. \tag{17}$$

Note that the x_m's in $\mathcal{D}_{\text{train}}$ may not belong to $\mathcal{D}_s$ and the quantity M could be arbitrarily large. After obtaining $\mathcal{D}_{\text{train}}$, the generative model $F(Y, Z; \theta)$ in Eq. (3) is trained using supervised learning with the MSE loss. Lastly, given an observation y, we evaluate the trained F to generate numerous samples from the target distribution $p(X|Y = y)$ to estimate parameter X and quantify its posterior uncertainty.

B Results of the second synthetic case

This section provides the results of another synthetic case, where we pick a different sample from the ELM simulation as the synthetic "truth" than the one presented in the main text. The following Figure 4 summarizes the parameter estimation results from the DBUQ and the MCMC and Figure 5 shows the corresponding LH predictions. This synthetic case once again demonstrates DBUQ's competence in accurate inverse modeling and parameter uncertainty quantification.

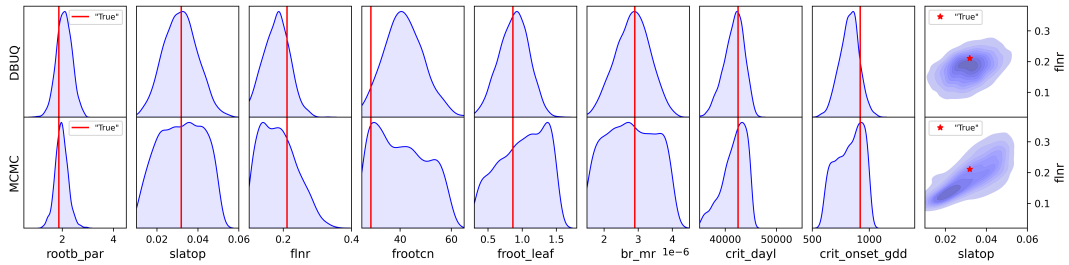

Figure 4: Parameter posterior distributions estimated by the DBUQ and MCMC in the second synthetic case.

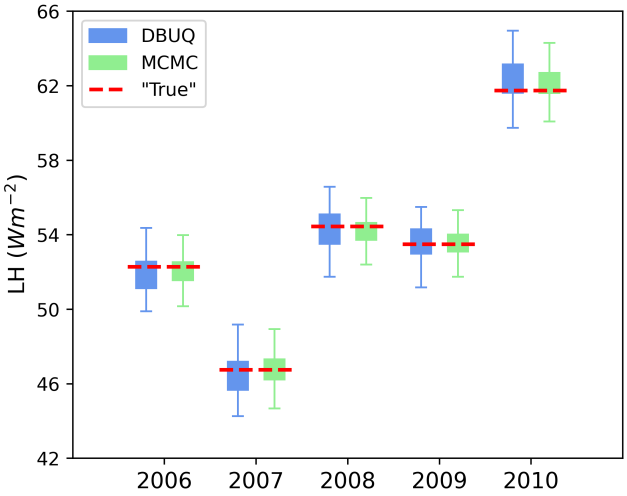

Figure 5: Prediction results of LH in the second synthetic case.