INVERTIBLE NEURAL NETWORKS FOR E3SM LAND MODEL CALIBRATION AND SIMULATION

Dan Lu, Daniel Ricciuto, Jiaxin Zhang
Oak Ridge National Laboratory
1 Bethel Valley Rd, Oak Ridge, TN, USA
{lud1,ricciutodm,zhangj}@ornl.gov

ABSTRACT

We apply an invertible neural network (INN) for E3SM land model (ELM) calibration and simulation with eight parameters at the Missouri Ozark AmeriFlux forest site. INN provides bijective (two-way) mappings between inputs and outputs, thus it can solve probabilistic inverse problems and forward approximations simultaneously. We demonstrate INN's inverse and forward capability in both synthetic and real-data applications. Results indicate that INN produces accurate parameter posterior distributions similar to Markov Chain Monte Carlo sampling and it generates model outputs close to the ELM simulations. Additionally, both the inverse and forward evaluations in INN are computationally efficient which allows for rapid integration of observations for parameter estimation and fast model predictions.

1 INTRODUCTION AND MOTIVATION

The Energy Exascale Earth System Model (E3SM) [1; 2], Land Model (ELM) simulates terrestrial water, energy, and biogeochemical processes in terrestrial surfaces and is an important tool for improving our understanding of ecosystem responses to climate change. ELM 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 sites, assigning uniform values to these site specific parameters may result in inaccurate model simulations at a single site showing large model discrepancy from site observations. Thus, this requires model calibration against site observations and performing parameter estimation at every site to improve model's predictability globally. Additionally, the parameter estimation should be computationally efficient, so we can quickly integrate multiple sources of observation streams and streamline this process to every sites.

However, calibrating ELM is challenging because of strong nonlinearity, poorly constrained model parameters, and significant computational costs [4]. The strong model nonlinearity and unconstrained parameters require quantifying uncertainties in parameter estimates to interpret inversion results correctly. Unfortunately, estimating uncertainty in nonlinear inverse problems is computationally expensive. For example, a Markov chain Monte Carlo (MCMC) sampling typically requires hundreds of thousands, or even millions of model evaluations to get convergence and these model simulations cannot be done fully in parallel [5; 6; 7]. To reduce the computational costs, surrogate modeling has been integrated into MCMC sampling [8; 9; 10; 11]. A surrogate model, which can be constructed using polynomials or neural networks, approximates the actual simulation model based on paired samples of model inputs and outputs and then replaces the simulation model in the MCMC calculation to reduce the single model evaluation cost thus the total sampling cost. This two-step model calibration approach, i.e., first building a surrogate of the forward model and then performing the inverse modeling on the surrogate for parameter estimation, has obtained increasing popularity in the earth science community. The limitation is that we need to build a globally accurate surrogate in the entire parameter space and whenever we change the likelihood function, either due to the change of observations or the likelihood formulation, we need to rerun the MCMC. This prevents the two-step approach from rapid integration of site observations where data are continuously collected and from the ELM calibration at every single sites where many similar inversions must be performed.

In this study, we present an invertible neural network (INN) which solves probabilistic inverse problems and forward approximations simultaneously [12; 13]. INN provides bijective (two-way) mappings $\boldsymbol{x} \leftrightarrow \boldsymbol{d}$ between model inputs $\boldsymbol{x}$ (i.e., model parameters) and outputs $\boldsymbol{d}$ (e.g., system states or flux variables). The network is invertible by construction, thus providing a unique opportunity where we can train it on the well-understood forward process $\boldsymbol{x} \rightarrow \boldsymbol{d}$ and get the inverse $\boldsymbol{d} \rightarrow \boldsymbol{x}$ for free by backward evaluation. Additionally, both forward and inverse mapping in INN are efficiently computable, and both mappings have a tractable Jacobian, which allows explicit computation of posterior probabilities. We demonstrate an application of INN by calibrating eight parameters in the ELM against five years of measurements of latent heat fluxes at the Missouri Ozark AmeriFlux forest site. We evaluate INN's calibration results by comparing its estimated parameter distributions with the MCMC sampling and assess its forward approximations in comparison with the ELM simulations.

The main contributions of this work are:

1. We present an invertible neural network to efficiently address both the inverse and forward modeling problems simultaneously.
2. We apply INN for ELM inverse model calibration and forward simulation on both synthetic and real observation data.
3. In inverse calibration, INN produces parameter posterior distributions similar to the MCMC sampling; in forward evaluation, INN generates model outputs close to the ELM simulations.
4. After training, INN solves both inverse and forward problems in seconds, and thus can be used for accurate parameter estimation and prediction in scenarios where rapid evaluations are required.

2 Invertible Neural Networks

INN is a class of networks that provide bijective mappings between inputs and outputs (the INN structure is presented in Appendix A). When evaluating INN in the forward direction, we obtain a solution of the forward modeling, and when INN is evaluated in the reverse direction, it produces the solution of the inverse problem. To consider the nonunique solutions of parameters $\boldsymbol{x}$, additional latent variables $\boldsymbol{z}$ are introduced to augment the original model outputs $\boldsymbol{d}$, such that the pair comprising one $\boldsymbol{x}$ and the augmented vector $[\boldsymbol{d}, \boldsymbol{z}]$ is unique and the relationship between $\boldsymbol{x}$ and $[\boldsymbol{d}, \boldsymbol{z}]$ is one-to-one. Then, we can use INN to learn the bijective mapping $\boldsymbol{x} \leftrightarrow [\boldsymbol{d}, \boldsymbol{z}]$ and solve both the forward and inverse problems simultaneously. According to [12], we train INN to approximate the forward process and meanwhile ensure the latent variables $\boldsymbol{z}$ predicted by the network are distributed according to a chosen distribution, e.g., a Gaussian distribution (theoretically any distributions can be used as proved in [12]). Then the solution of the inverse problem, i.e., the parameter posterior distributions $p(\boldsymbol{x}|\boldsymbol{d}_{obs})$, can be obtained thereafter by evaluating the network backwards given observations $\boldsymbol{d}_{obs}$ with $\boldsymbol{z}$ randomly sampled from its chosen distribution. In implementation, we first draw many samples of $[\boldsymbol{d}_{obs}, \boldsymbol{z}]$ where $\boldsymbol{z}$ is generated from the chosen Gaussian distribution and $\boldsymbol{d}_{obs}$ is held constant. Next, we evaluate the trained INN backwards for these samples, which will transform the distribution $p(\boldsymbol{z})$ to the parameter posterior distribution $p(\boldsymbol{x}|\boldsymbol{d}_{obs})$. This backward evaluation takes about 3 seconds in this study.

The loss function includes three components: the model output loss $\mathcal{L}_{\boldsymbol{d}}$ which penalizes deviations between model outputs $\boldsymbol{d}$ and network predictions $f_{\boldsymbol{d}}(\boldsymbol{x})$, the loss for latent variables $\mathcal{L}_{\boldsymbol{z}}$ which penalizes the mismatch between the joint distribution of network outputs $q(\boldsymbol{d}, \boldsymbol{z})$ and the product of their marginal distributions $p(\boldsymbol{d})p(\boldsymbol{z})$, and the loss on the inputs $\mathcal{L}_{\boldsymbol{x}}$ which matches the distribution of backward predictions $q(\boldsymbol{x})$ against the input prior distribution $p(\boldsymbol{x})$. We use mean squared loss for $\mathcal{L}_{\boldsymbol{d}}$, and Maximum Mean Discrepancy (MMD) [14] for $\mathcal{L}_{\boldsymbol{z}}$ and $\mathcal{L}_{\boldsymbol{x}}$ to measure the difference between two distributions. In summary, the total loss $\mathcal{L}$ of INN is represented as

$$\mathcal{L} = \alpha\mathbb{E}[(\boldsymbol{d} - f_{\boldsymbol{d}}(\boldsymbol{x}))^2] + \beta\mathrm{MMD}[q(\boldsymbol{d}, \boldsymbol{z}), p(\boldsymbol{d})p(\boldsymbol{z})] + \gamma\mathrm{MMD}[q(\boldsymbol{x}), p(\boldsymbol{x})], \tag{1}$$

where $q(\boldsymbol{d}, \boldsymbol{z}) = p(\boldsymbol{x})/|J(\boldsymbol{x})$ and $J(\boldsymbol{x})$ is the Jacobian of the forward transform embodied in the network; $q(\boldsymbol{x}) = p(\boldsymbol{d})p(\boldsymbol{z})/|J(\boldsymbol{d}, \boldsymbol{z})|$ and $J(\boldsymbol{d}, \boldsymbol{z})$ is the Jacobian of the backward transform. In training, the hyperparameters α, β, and γ are selected manually to give a balance between the three losses. In this study we observe that the performance of INN is not very sensitive to the three hyperparameter values. After training, INN produces parameter posterior distributions $p(\boldsymbol{x}|\boldsymbol{d}_{obs})$ for a given observation $\boldsymbol{d}_{obs}$, as well as an approximation function f that maps model parameters to its corresponding outputs, which can serve as a fast-to-evaluate surrogate of the forward model.

3 E3SM LAND MODEL CALIBRATION PROBLEM

The E3SM land model (ELM) simulations are performed at the Missouri Ozark AmeriFlux site [15] that has been collecting eddy covariance data since 2004 [16]. We use the annual average latent heat fluxes (LH) collected in the site from 2006 to 2010 (i.e., five model output quantities) for ELM parameter estimation. ELM involves more than 60 parameters. According to the sensitivity analysis of [3], seven model parameters were responsible for more than 80% of the variation in the LH. These seven parameters are a factor controlling 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]), and the critical day length to initiate autumn senescence (crit_dayl [35000, 55000]). Along with one additional parameter crit_onset_gdd [500, 1300], a total of eight parameters are estimated. The parameter crit_onset_gdd, which represents the number of accumulated growing degree days needed to initiate spring leaf-out, was previously encoded as a constant in ELM. However, given a recent analysis highlighting the importance of phenology for carbon uptake [17], we think it is important to include this parameter for calibration. The ranges of the eight parameters are listed above after the parameter names.

In this study, we use INN to learn the relationship between the eight parameters and the five model outputs of LH. We have 1000 pairs of ELM simulation samples. We use the 800 samples for INN training and the remaining 200 samples as validation set for hyperparameters turning (details in Appendix B). The training process takes about 10 mins. After training, we can use INN to perform forward simulation given a parameter value, and estimate parameter posterior distributions given the observations. Both forward and inverse evaluations take seconds. To evaluate the INN's inverse modeling performance, we compare its estimated parameter probability density functions (PDFs) with MCMC results. For the MCMC simulation, we take the two-step approach, first building a surrogate model based on the 1000 prior samples and then perform the MCMC on the surrogate. The widely applied EMCEE algorithm [18] is employed to draw MCMC samples. It uses 16 chains, each of which contains 50000 samples including a burn-in period of 40000 samples. The burn-in samples are discarded and every 100^{th} of the remaining 10000 samples in each chain are included in the final set (i.e., 1600 samples) for the parameter posteriors approximation. In this two-step approach, after building the surrogate model, the MCMC sampling takes about 5 hours.

4 RESULTS

We apply INN to both synthetic and real-data observations. In the synthetic case, we pick one ELM-simulated LH sample from the validation set as a synthetic observation and use the corresponding parameter sample as the synthetic "truth" to evaluate the INN's inverse modeling accuracy. In the real-data case, the parameters are calibrated using the real observations from the forest site. We first evaluate INN's inverse capability in the synthetic case and then discuss the results in the real-data case; lastly we assess INN's performance in forward model approximations. We use the normalized root mean squared error (NRMSE) and coefficient of determination (R^2) for performance evaluation; both metrics are defined in Appendix C.

Figure 1 presents the parameter estimation results of both INN and MCMC in the synthetic case. Overall, INN and MCMC produce comparable results in terms of both point estimation accuracy quantified by the maximum a posteriori (MAP) and the uncertainty quantification (UQ) assessed using the posterior PDFs. The NRMSE between the MAP estimates and the synthetic "truth" over the eight parameters is 0.002 for INN and 0.019 for MCMC, where INN shows a better point estimation. Figure 1 also indicates that the MAP estimates of INN are closer to the "truth" than the MCMC results. Additionally, the posterior PDFs estimated by INN and MCMC have similar shapes both for marginal distributions and joint distributions. These estimated posteriors are consistent with our domain knowledge. For example, parameter rootb_par is sensitive to LH, thus the values of rootb_par should be well constrained by the LH observations. And we see that the prior uncertainty of rootb_par is greatly reduced by the model calibration to a small posterior uncertainty with a narrow PDF shape. Also, we know that parameters slatop and flnr have a strong positive correlation and we observe that their estimated joint distributions reflect this property.

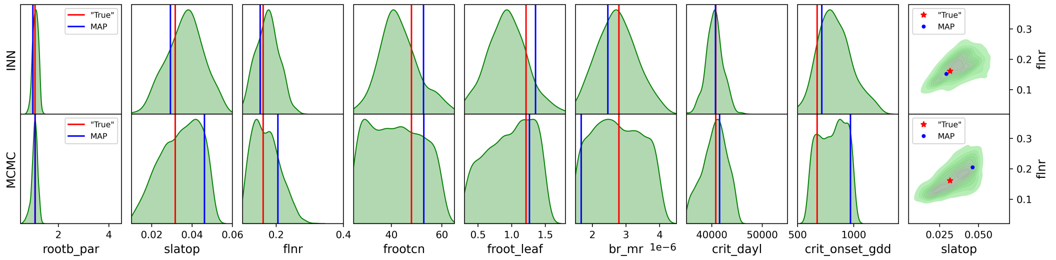

Figure 1: Parameter posterior distributions estimated by INN and MCMC in the synthetic case, where the blue color highlights maximum a posteriori (MAP) estimate and the red color highlights the synthetic “true” value.

For nonlinear problems without analytical solutions of the parameter posteriors, we do not know the exact shape of their PDFs. To further assess the accuracy of the parameter UQ, we investigate the posteriors of the model outputs. If the parameter posterior uncertainty is reasonably quantified, the generated LH posterior samples should enclose the observed value. Figure 2(a) indicates that the LH samples from both INN and MCMC are closely distributed around the synthetic “true” observations; their MAP predictions almost overlap with the "true" observation and their quantiles are also similar to each other. This suggests that INN reasonably quantifies the parameter uncertainty and solves the inverse problems accurately. It produces similar results with the MCMC sampling but using much smaller computational costs (10 mins used for INN training Vs. 5 hours for MCMC sampling after surrogate construction). Additionally, after INN is trained, for any given observations, INN can produce corresponding parameter posterior distributions in seconds without retraining the network. For further demonstration and to explore generalization properties of the trained INN, we evaluate the network backwards for another set of synthetic observations and present the results in Appendix D. This second synthetic case again demonstrates INN’s competence in accurate inverse modeling and parameter UQ.

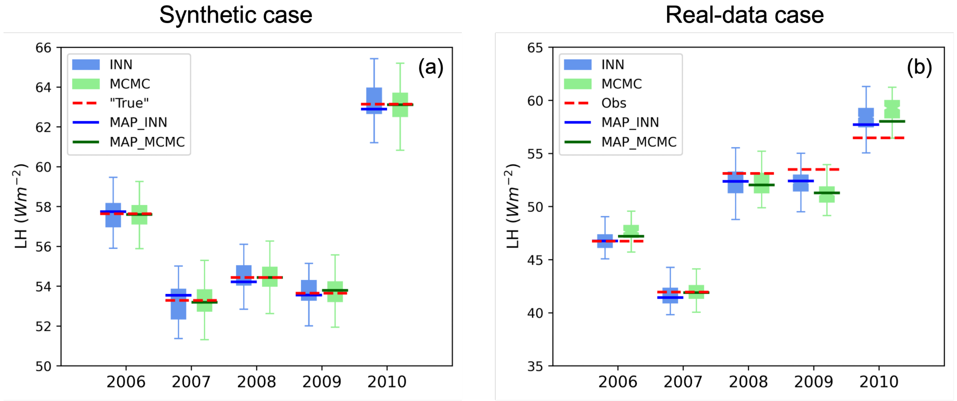

Figure 2: Boxplots summarize LH predictions from the parameter posterior samples of INN and MCMC; MAP_INN represents the prediction from the MAP estimate of INN and MAP_MCMC represents the prediction from the MAP estimate of MCMC. “True” in (a) present synthetic observation; Obs in (b) is real observation.

In the real-data case, we reuse the trained INN and evaluate it backwards for the real observations. 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. The figures indicate that INN produces similar results as MCMC both in the parameter UQ and in the LH predictions. Their generated parameter posteriors, including marginal and joint distributions, show similar PDF shapes and their MAP estimates are close to each other. Additionally, the LH prediction samples from both methods accurately enclose the observed data (Figure 2(b)). Their MAP estimates have a nice fit with the observed LH, where the NRMSE values of INN and MCMC estimates are 0.017 and 0.026, respectively, and the R^2 values of INN and MCMC predictions are 0.97 and 0.92, respectively.

We draw similar conclusions in both the synthetic and real-data cases: INN can solve the inverse problems effectively and efficiently; it provides accurate posteriors of both parameters and predictions

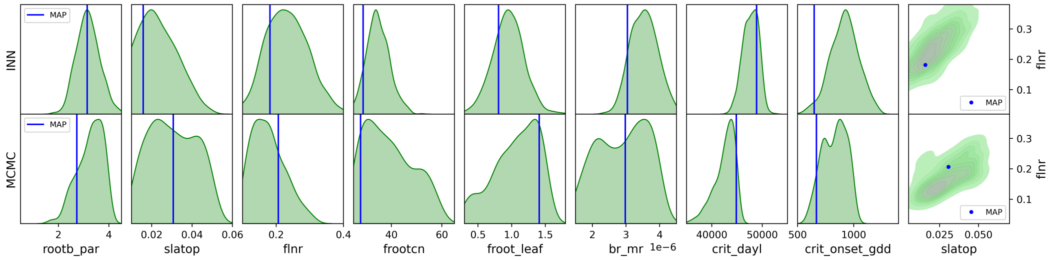

Figure 3: Parameter posterior distributions estimated by INN and MCMC in the real-data case.

which show similar PDF shapes and MAP estimates with the MCMC. However, the computational time of INN is much less than that of MCMC. In both cases, MCMC takes 5 hours for sampling after surrogate construction. In contrast, the evaluation time of INN is only 3 seconds after training. The training of INN takes 10 mins and the same trained network is used in both cases.

Since INN is trained bidirectionally, it also provides approximations of forward model simulations. When evaluating the trained INN forwards with a set of parameter values, the network provides an approximation to the forward process and yields LH simulations. In this case, INN serves as a surrogate model of the ELM for model outputs simulation. To assess the forward capability of INN, we perform the forward evaluation for the 1000 prior samples. If INN produces an accurate approximation of ELM, it should generate LH simulations similar to the ELM outputs. Figure 4 indicates that INN provides a good approximation/surrogate to the ELM. The INN predictions show a strong consistency with ELM simulations, where the R^2 of all the five quantities consistently have values of 1.0. As the ELM simulation is computationally expensive (one ELM run takes 2-3 hours in this study), INN provides an accurate and fast-to-evaluate surrogate which will greatly reduce the computational costs in modeling tasks where many forward evaluations are required.

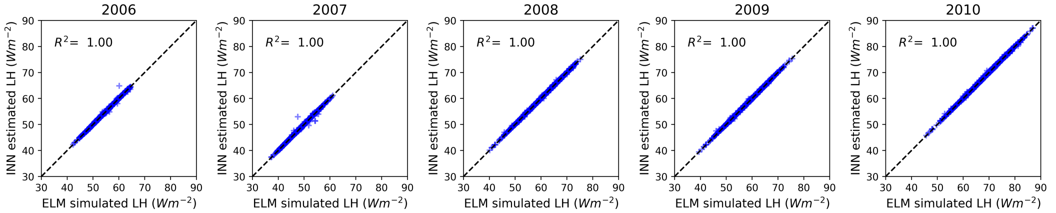

Figure 4: Comparison between INN forward evaluated LH and ELM simulated LH for the 1000 prior samples.

5 Discussion and future work

We apply the invertible neural networks for ELM model calibration and simulation at one AmeriFlux site. In both synthetic and real-data applications, we demonstrate that INN can accurately solve the inverse problems by generating parameter posterior distributions that can be interpreted by our domain knowledge and justified by the synthetic truth and predictions. The INN-estimated posteriors are similar to the MCMC approximations, but INN greatly reduces the computational time and infrastructure for computing parameter posteriors and predictions. Additionally, INN produces accurate forward model approximations which can serve as an effective fast-to-evaluate surrogate of the ELM at the site. In the future, we will apply INN to other AmeriFlux sites for parameter estimation and improve the ELM predictability globally.

INN can be widely applied to a variety of parameter estimation problems. But for grid-based parameters with hundreds of dimensions, it may require a large number of training data and a long training time. Additionally, for some calibration problems where the observations are outside the training samples, INN may have relatively low accuracy in approximating the parameter distributions because of the extrapolation error in network training.

ACKNOWLEDGMENTS

This research is supported as part of the Terrestrial Ecosystem Science-Science Focus Area (TES-SFA) project and part of the Artificial Intelligence Initiative of the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the US DOE under contract DE-AC05-00OR22725. It is also sponsored by the Data-Driven Decision Control for Complex Systems (DnC2S) project funded by the US DOE, Office of Advanced Scientific Computing Research.

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] D. Lu, D. Ricciuto, M. Stoyanov, and L. Gu, "Calibration of the e3sm land model using surrogate-based global optimization," *Journal of Advances in Modeling Earth Systems*, vol. 10, no. 6, pp. 1337–1356, 2018.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] L. Ardizzone, J. Kruse, C. Rother, and U. Köthe, "Analyzing inverse problems with invertible neural networks," in *International Conference on Learning Representations*, 2019.

[13] J. Kruse, L. Ardizzone, C. Rother, and U. Köthe, "Benchmarking invertible architectures on inverse problems," 2021.

[14] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. 25, pp. 723–773, 2012.

[15] 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.

[16] L. Gu, S. G. Pallardy, B. Yang, K. P. Hosman, J. Mao, D. Ricciuto, X. Shi, and Y. Sun, "Testing a land model in ecosystem functional space via a comparison of observed and modeled ecosystem flux responses to precipitation regimes and associated stresses in a Central U.S. forest," *Journal of Geophysical Research (Biogeosciences)*, vol. 121, pp. 1884–1902, July 2016.

[17] J. Xia, S. Niu, P. Ciais, I. A. Janssens, J. Chen, C. Ammann, A. Arain, P. D. Blanken, A. Cescatti, D. Bonal, N. Buchmann, P. S. Curtis, S. Chen, J. Dong, L. B. Flanagan, C. Frankenberg, T. Georgiadis, C. M. Gough, D. Hui, G. Kiely, J. Li, M. Lund, V. Magliulo, B. Marcolla, L. Merbold, L. Montagnani, E. J. Moors, J. E. Olesen, S. Piao, A. Raschi, O. Roupsard, A. E. Suyker, M. Urbaniak, F. P. Vaccari, A. Varlagin, T. Vesala, M. Wilkinson, E. Weng, G. Wohlfahrt, L. Yan, and Y. Luo, "Joint control of terrestrial gross primary productivity by plant phenology and physiology," *Proceedings of the National Academy of Sciences*, vol. 112, no. 9, pp. 2788–2793, 2015.

[18] 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.

[19] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," 2017.

APPENDIX

A INVERTIBLE NEURAL NETWORK STRUCTURE

A typical design of an invertible neural network (INN) contains a serial sequence of reversible blocks [19], each of which consists of two coupled layers. Each block's input vector $\boldsymbol{u}$ (e.g., $\boldsymbol{u}$ can be the model parameter vector) is split into two halves $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, which are transformed by an affine function with coefficients exp(s_i) and t_i to produce the output ($\boldsymbol{v}_1$, $\boldsymbol{v}_2$):

$$\boldsymbol{v}_1 = \boldsymbol{u}_1 \odot \exp(s_2(\boldsymbol{u}_2)) + t_2(\boldsymbol{u}_2), \boldsymbol{v}_2 = \boldsymbol{u}_2 \odot \exp(s_1(\boldsymbol{v}_1)) + t_1(\boldsymbol{v}_1). \tag{2}$$

Given the output $\boldsymbol{v} = [\boldsymbol{v}_1, \boldsymbol{v}_2]$, these expressions are trivially invertible:

$$\boldsymbol{u}_2 = (\boldsymbol{v}_2 - t_1(\boldsymbol{v}_1)) \odot \exp(-s_1(\boldsymbol{v}_1)), \boldsymbol{u}_1 = (\boldsymbol{v}_1 - t_2(\boldsymbol{u}_2)) \odot \exp(-s_2(\boldsymbol{u}_2)). \tag{3}$$

Note that functions s_i and t_i do not need to be invertible. In our implementation, they are realized by a succession of several fully connected layers with leaky ReLU activations. To improve interaction between parameters of the input vector, we add a permutation layer after each reversible block.

B NETWORK CONFIGURATION

The INN used in this work is designed using four reversible blocks, each of which contains fully connected subnetworks. Each affine function (i.e., s_i and t_i in Appendix A) is implemented using a neural network with 3 fully connected layers each of which contains 128 hidden units with leaky ReLU activation functions. The ADAM optimizer is used with a learning rate of 0.001. We use a relatively small batch size of 10 considering the small number of training data. The network has converged after 1000 epochs when both the training loss and validation loss become stationary.

C PERFORMANCE EVALUATION METRICS

We consider two evaluation metrics, normalized root mean squared error (NRMSE), and coefficient of determination (R^2).

$$\text{NRMSE} = \sqrt{\frac{\sum_{i=1}^{N}\left(Y_i^{obs} - Y_i^{sim}\right)^2}{\sum_{i=1}^{N}\left(Y_i^{obs}\right)^2}}. \tag{4}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(Y_i^{obs} - Y_i^{sim}\right)^2}{\sum_{i=1}^{N}\left(Y_i^{obs} - \overline{Y}^{obs}\right)^2} \tag{5}$$

where Y_i^{obs} is the ith observation, Y_i^{sim} is the ith simulated value, $\overline{Y}^{obs}$ is the mean of observation, and N is the total number of observations. NRMSE measures the estimation errors in a squared sense normalized by the observations; its value varies from the optimal 0 to a large positive number. The lower the NRMSE value is, the better the model simulation performance. R^2 is a normalized statistic that determines the relative magnitude of the residual variance compared to the observation variance. It indicates how well the plot of observed versus simulated data fits the 1:1 consistency line. R^2 value ranges from negative infinity to 1.0. A value of 1.0 corresponds to a perfect match of model simulations to observations, and a negative R^2 indicates that the model performs worse than the observed mean.

D 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 5 summarizes the parameter estimations results from the INN and the MCMC and Figure 6 shows the corresponding LH predictions. This synthetic case once again demonstrates INNs' competence in accurate inverse modeling and parameter UQ.

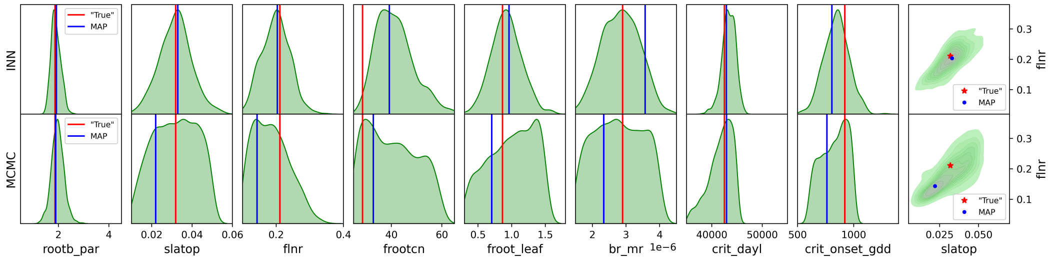

Figure 5: Parameter posterior distributions estimated by the INN and MCMC in the second synthetic case.

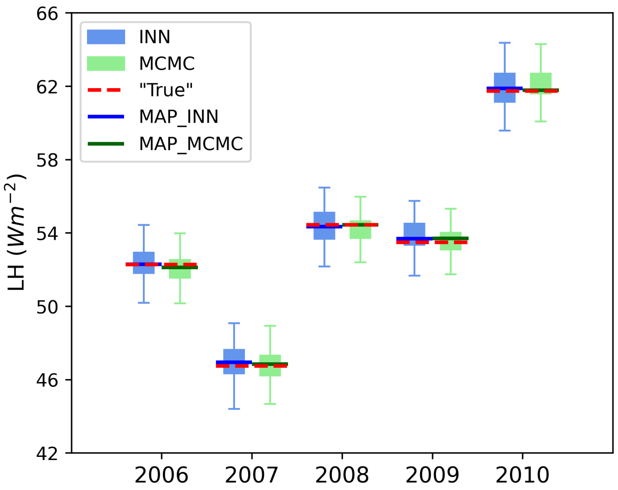

Figure 6: Prediction results of LH in the second synthetic case.