A framework for variational inference and data assimilation of soil biogeochemical models using state space approximations and normalizing flows

H. W. Xie[1,†], D. Sujono[2,†], T. Ryder[3], E. B. Sudderth[2], S. D. Allison[1,4]

[1]Center for Complex Biological Systems, University of California, Irvine, Irvine, CA, United States of America
[2]Department of Computer Science, University of California, Irvine, Irvine, CA, United States of America
[3]School of Mathematics Statistics and Physics, Newcastle University, Newcastle, United Kingdom
[4]Department of Ecology and Evolutionary Biology, University of California, Irvine, Irvine, CA, United States of America
[†]Authors contributed equally to this work.

Key Points:

- Soil biogeochemical models (SBMs) represent the dynamics of soil systems and predict their responses to global warming.
- SBMs are fit to data sets including soil respiration and organic matter measurements to assess model accuracy and estimate parameter values.
- We demonstrate a Bayesian inference workflow in this study that can be used for efficient SBM data fitting and parameter estimation.

Corresponding author: Hua Xie, `xiehw@uci.edu`

Abstract

Soil biogeochemical models (SBMs) simulate element transfer processes between organic soil pools. These models can be used to specify falsifiable quantitative assertions about soil system dynamics and their responses to global surface temperature warming.

To determine whether SBMs are useful for representing and forecasting data-generating processes in soils, it is important to conduct data assimilation and fitting of SBMs conditioned on soil pool and flux measurements to validate model predictive accuracy. SBM data assimilation has previously been carried out in approaches ranging from visual qualitative tuning of model output against data to more statistically rigorous Bayesian inferences that estimate posterior parameter distributions with Markov chain Monte Carlo (MCMC) methods. MCMC inference is better able to account for data and parameter uncertainty, but the computational inefficiency of MCMC methods limits their ability to scale assimilations to larger data sets.

With formulation of efficient and statistically rigorous SBM inference frameworks remaining an open problem, we demonstrate the novel application of a variational inference framework that uses a method called normalizing flows to approximate SBMs that have been discretized into state space models. We fit the approximated SBMs to synthetic data sourced from known data-generating processes to identify discrepancies between the inference results and true parameter values and ensure functionality of our method. Our approach trades estimation accuracy for algorithmic efficiency gains that make SBM data assimilation more tractable and achievable under computational time and resource limitations.

Plain Language Summary

Soil biogeochemical models (SBMs) simulate soil systems in a quantifiable and falsifiable manner. Climate researchers rely on SBMs to predict how soil systems could be globally affected by climate change. However, SBMs differ widely in their predictions of changes in soil measurements including rates of soil carbon dioxide emissions. It is unclear which SBMs offer more realistic climate projections, and the establishment of statistical techniques to rigorously compare the predictive performance of SBMs is still a work in progress. We make a contribution to SBM comparison efforts by developing a statistical framework to assess SBM accuracy that leverages deep learning for computational efficiency gains. Results of our case study demonstrate that we can fit two SBMs to soil observation data and estimate ranges of SBM parameter values compatible with those observations. Our modular framework is flexible and stimulates future work to improve on our procedure with modifications of our existing methods.

1 Introduction

Soil biogeochemical models (SBMs) are differential equation systems that represent dynamics of organic matter transfer between soil pools, including the soil organic (SOC), dissolved organic (DOC), and microbial biomass carbon (MBC) pools. The state variables of SBMs typically are densities or masses of elements in those pools (Manzoni & Porporato, 2009), and heterotrophic soil CO_2 emissions can be estimated from those state values and microbial parameters (Allison et al., 2010). As soil microbe communities influencing organic mass transfer dynamics evolve and shift under the selection pressures of terrestrial warming, SBMs have become an important tool for soil scientists and biogeochemists to quantify changes in soil system activity and predict future heterotrophic soil respiration levels (Sulman et al., 2018; Saifuddin et al., 2021).

SBMs offer falsifiability of their dynamics through their depiction of biological soil processes as interpretable mathematical equations governed by model parameters θ. How-

ever, the formulation of statistically sound frameworks to assess the dynamical validity and predictive accuracy of SBMs remains an open problem in soil biogeochemistry (Luo et al., 2016; Xie et al., 2020; Bradford et al., 2021; Georgiou et al., 2021; Raczka et al., 2021). One approach for assessing SBM utility involves comparing models by their ability to assimilate soil observation data with suitable θ values, assuming that models that can more accurately describe the past will also be better at predicting the future (Wieder et al., 2014; Bradford et al., 2021). Past SBM fit evaluations have ranged from visual juxtapositions of manually calibrated model outputs against empirical observations (Sulman et al., 2014; Wieder et al., 2015) to quantitative frequentist comparisons involving correlation coefficients and root-mean-square errors (Todd-Brown et al., 2013, 2014; Wieder et al., 2014). In an effort to account for uncertainty in θ values and data observations and encode expert domain beliefs, other comparisons have involved the use of Bayesian Markov chain Monte Carlo (MCMC) inference methods and goodness-of-fit metrics with some success (Hararuk et al., 2014; Hararuk & Luo, 2014; J. Li et al., 2019; Xie et al., 2020; Saifuddin et al., 2021; Wang et al., 2022).

MCMC transition sampling methods, such as the Gibbs (Geman & Geman, 1987) and No-U-Turn (NUTS) (Hoffman & Gelman, 2014) samplers deployed in widely-used probabilistic programming platforms like JAGS (Plummer, 2003), Stan (Carpenter et al., 2017), and PyMC (Salvatier et al., 2016), are powerful algorithms for inference, but their relative computational cost presently limits their ability to scale for use on model comparisons involving more complex SBM systems conditioned on larger data sets spanning decades (Kucukelbir et al., 2017). Stochastic gradient optimization variational inference (VI) is an alternative approach to Bayesian inference and model-fitting that trades asymptotic exactness and the ability to estimate non-parametric posterior distributions for increased computational efficiency and simplicity (Blei et al., 2017). It does so by reframing Bayesian inference from a transition sampling problem to an optimization objective of maximizing a metric called the evidence lower bound (ELBO), which corresponds to minimizing the discrepancy between an approximate parametric posterior and true posterior distribution (Salimans et al., 2015).

VI on differential equation models benefits from the use of stochastic differential equation (SDE) over ordinary differential equation (ODE) systems. SDE noise provides a means of adjusting and correcting for proposals of system initial conditions and underlying dynamics that are inconsistent with the true data-generating process sourcing the data observations (Whitaker, 2016; Särkkä & Solin, 2019; Wiqvist et al., 2021). Additionally, noise-driven fluctuation and variation in state trajectories can account for outlier data measurements during inference to reduce optimization pressures that can drive rigid deterministic models into unstable θ regimes. SDE noise thereby improves inference flexibility, stability, and efficiency through the acommodation and mitigation of discrepancies between model outputs and data generation or observation. Furthermore, SDEs offer a more realistic and accurate representation of the stochasticity that is inherent to biological processes across all scales (Golightly & Wilkinson, 2011; Abs et al., 2020; Browning et al., 2020). The ability to effectively fit SDEs is an advantage of VI over many established MCMC methods; off-the-shelf MCMC implementations are frequently not designed to tolerate the noisy likelihood estimates of SDEs (Golightly & Wilkinson, 2010; Fuchs, 2013; Chen et al., 2014).

With the goal of applying VI to SBMs in mind, we formulated SDE versions of the linear deterministic "conventional" (CON) SBM system (Allison et al., 2010; J. Li et al., 2014) to establish an SCON family of models and leverage the versatility of stochastic optimization. As is the case for CON, SCON models have three state variables representing SOC, DOC, and MBC densities in a soil system. We parameterized two SCON variants, "constant diffusion" SCON (SCON-C) and "state-scaling diffusion" SCON (SCON-SS). Diffusion coefficients are model parameters that govern the noise dynamics of an

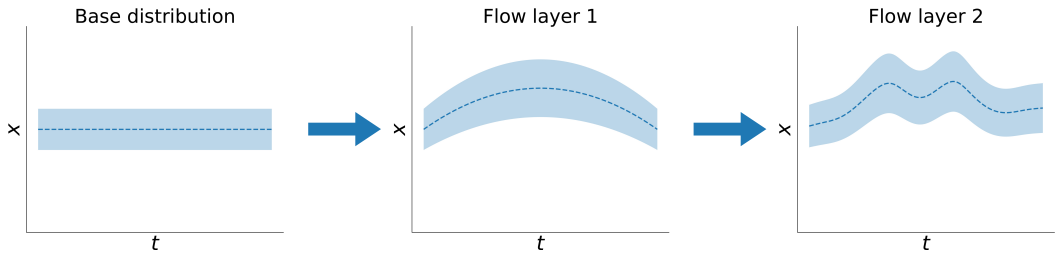

Figure 1. In our study, we use normalizing flows to approximate SCON soil biogeochemical model solution trajectories x over time t. The flow operates in a generative direction, mapping a simpler base distribution to a more complex one representing SCON output.

SDE system. In SCON-C, diffusion, or noise, was set to be independent of time and states, while in SCON-SS, noise was made to depend on and scale with state values.

We used a class of methods called *normalizing flows* to approximate SCON models in our inference approach. In simple terms, flows can be thought of as one or more layers of random variable mappings that transform an initial base probability distribution to a new distribution (Papamakarios et al., 2021). When we deploy flows to transform a simpler probability density into a more complex one (Figure 1), as we do in our study, it is classified as a *generative* normalizing flow (Kobyzev et al., 2020). The flow approximation refashions SCON from an SDE that depicts state variable dynamics $\frac{\mathrm{d}x}{\mathrm{d}t}$ in continuous time to a probabilistic state space model that specifies distributions of state measurements y_t noisily observed from underlying states x_t in discrete time (Särkkä & Solin, 2019).

The replacement of differential equation solver integration with state space models to approximate dynamical systems offers substantial computational efficiency gains in inference (Ryder et al., 2018; Särkkä & Solin, 2019). At each inference iteration or epoch, rather than sequentially computing state trajectories x one time step at a time with solvers including Euler, Runge-Kutta, and Adams' schemes, as was demonstrated in studies like Xie et al. (2020), we can instead simultaneously sample multiple x in one vectorized draw from a flow-transformed state space distribution object. This increased efficiency allows us to more capably assimilate SBMs with time series data sets spanning longer periods under computing resource limitations, especially when highly parallelizable graphical processing units (GPUs) can be leveraged.

Drawing from the methodologies of previous work that test various inference approaches (Golightly & Wilkinson, 2006; Whitaker et al., 2017; Ryder et al., 2018, 2021), our study demonstrates functional stochastic VI of flow-approximated SBMs conditioned on soil observations data y that includes various soil pool and respiration measurements. To support the notion that our VI approach is operational, we show that it can fit model output to y sourced from a known data-generating process and estimate model θ posteriors in line with the true θ values used by that process.

Hence, to begin our study workflow, we generated synthetic y consisting of SOC, DOC, and MBC state and heterotrophic CO_2 respiration rate observations corresponding to SCON-C and SCON-SS data-generating processes. The processes used "true" θ values randomly sampled from constrained data-generating distributions that were chosen to produce faster and more dramatic SOC decay dynamics reminiscent of organic matter decomposition at soil surface, which contrasts with the slower and deeper soil decomposition depicted in J. Li et al. (2014) and Xie et al. (2020). Faster decay provided our inference approach with substantive dynamical information in shorter time series to operate and optimize on. We then conditioned our state space model VI on those syn-

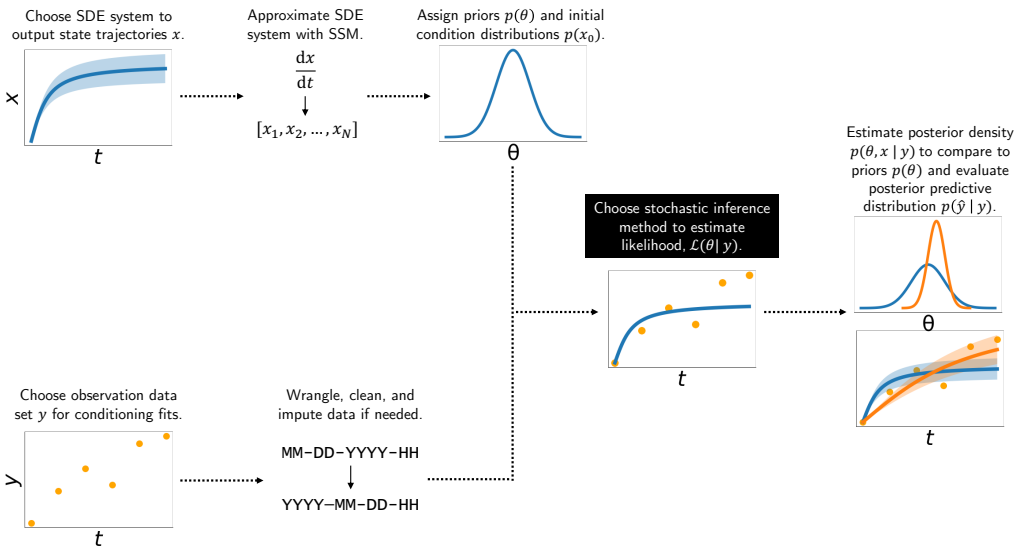

Figure 2. A workflow diagram summarizing the steps involved in our study's stochastic variational Bayesian framework. Our workflow efficiently conducts inference and data assimilation on stochastic differential equation (SDE) soil biogeochemical models (SBMs) with their approximation into state space models (SSMs). Our modular workflow is designed to serve as a basis for building future soil biogeochemical model inferences, as the "black box" inference method used can be modified or substituted. Our "black box" inference method of choice was stochastic gradient descent mean-field variational inference. Within the nodes of the diagram, blue lines and shading correspond to prior means and distributions, while orange lines and shading correspond to posterior means and distributions. Orange dots represent observations upon which the inference is conditioned.

thetic y for estimation of approximate posterior densities $q(\theta)$ that were compared with prior densities $p(\theta)$. Priors were made to be equivalent to our data-generating distributions.

Ultimately, we found that our VI approach allowed us to reasonably fit y. When possible, our fits were checked against solutions from a Kalman smoother algorithm, and we observed that the flow fits were mostly consistent with the Kalman solutions. Crucially, we were also able to recover some of the true θ values used by our data-generating processes against model identifiability limitations that could not be resolved by the extent of information contained in our synthetic data. *Model identifiability* can be summarized as the ability to update prior beliefs about θ and align model to true θ based on available data. Our identifiability issues related to the presence of ambiguous SCON equation terms involving the multiplication of more than two parameters. Our work offers insights and suggestions for improving the identification of θ, which is of interest for experimentalists and biogeochemists who are interested in building effective data sets for SBM inference.

2 Materials and Methods

2.1 Inference workflow overview

The general steps constituting our study's SBM data assimilation workflow are outlined in Figure 2. We established SCON-C and SCON-SS to serve as known data-generating processes whose true θ values can be compared with the inferred posteriors to test our flow VI method. Discrepancies between the true θ and posterior means inform on the effectiveness of our selected inference algorithm. Differences between the priors and posterior densities further indicate algorithm efficacy and additionally point to the informativeness of the data y for identifying and constraining posteriors.

True SCON-C and SCON-SS θ were sampled from data-generating distributions truncated between lower and upper support bounds to ensure that data-generating processes would remain in parameter regimes with faster state decay corresponding to soil surface decomposition occurring on the order of thousands of hours, rather than tens or hundreds of thousands. This allows us to generate shorter data sets y that enable reduced computational loads and faster turnaround times for testing our inference algorithm while retaining dynamical richness that can inform the algorithm to estimate more certain posteriors. We used logit-normal distributions to handle truncation in our data-generating, prior, and posterior distributions, which we will describe in section 2.3. Our inference priors matched our data-generating distributions.

Synthetic data y were observed and processed from our data-generating SDE solution trajectories. We parameterized our SCON models based in time units of hours, so observations were collected every 5 hours by default. State space approximation of SDE output, which we will describe in section 2.4, requires regular time series discretization (Kalman, 1960), so in an empirical setting, all existing, imputed, or missing observations must coincide with discrete time steps of the state space model in our approach and cannot transpire in between. Different SDE approximation methods would be needed for irregular time discretization.

We selected mean-field stochastic VI as our black box inference method for its mathematical simplicity and efficiency. Mean-field inference makes the simplifying assumption that model parameters are independently distributed. This aligns with our synthetic data-generating processes, in which our true θ values are sampled from independent logit-normal distributions. VI frames Bayesian inference as an optimization goal of finding the set of mean-field posterior distributions that best describes y. The optimization process takes place over a number of training iterations in which θ values are sampled at each iteration and the likelihood of the resultant model output conditioned on y and θ is evaluated in fulfillment of the objective of the VI algorithm to locate θ corresponding to higher model likelihood. We present an overview of our VI implementation and key algorithm steps in section 2.5.

We used normalizing flows to approximate SCON-C and SCON-SS from continuous-time SDEs to time-discretized state space models. These state space approximations then served as our bases for VI optimization. A brief treatment on state space models is given in section 2.4. Flow state space approximation increased the computational efficiency of sampling SCON solution trajectories (also referred to as *latent variables*, *states*, or *paths* in machine learning literature) such that multiple trajectories would be simultaneously collected from a flow distribution object rather than sequentially simulated from a differential equation solver at each training iteration. The flow is assembled through deep neural network layers that transform simpler random input into more complex approximated SCON output. The constituent pieces of the machine learning architecture underlying our flow are detailed in section 2.6.

Per equations (3) and (4), SCON-C is a completely linear SDE. Consequently, SCON-C flow-approximated x and its fit of y can be visually benchmarked against output from

an instance of the Kalman smoother algorithm summarized in section 2.7. Given a known data-generating process and observation error, a Kalman smoother exactly solves the true mean latent path x of the SDE data-generating process sourcing y. We successfully compared SCON-C flow x to the true x solution computed by the smoother, which we describe in section 3.1. The smoother algorithm cannot resolve the non-linear diffusion depicted in equation (5), so SCON-SS flow output could not be validated in the same manner.

2.2 SCON SDE parameterization and data generation

SDE system equations are frequently written with the state value derivatives $\mathrm{d}x$ on the left-hand side, and consist of a drift coefficient vector, frequently notated as α, and a diffusion coefficient matrix, notated as β, on the right-hand side. For biological SDE models, a square-root diffusion structure is frequently used such that these systems follow the form

$$\mathrm{d}x_t = \alpha(x_t, t, \theta)\mathrm{d}t + \sqrt{\beta(x_t, t, \theta)}\mathrm{d}W_t \tag{1}$$

where $\mathrm{d}W_t$ denotes a continuous stochastic Wiener process. Evolution of SDE trajectories x across a simulation duration T in time increments $\mathrm{d}t$ can be interpreted as a series of small steps whose values are independently drawn from a normal distribution with mean $\alpha(x_t, t)\mathrm{d}t$ and variance $\beta(x_t, t)\mathrm{d}t$ (Särkkä & Solin, 2019).

Like the CON model introduced in Allison et al. (2010), SCON has three state dimensions made up of soil organic C (SOC), dissolved organic C (DOC), and microbial biomass C (MBC) densities. We notate total state dimensions with $\mathcal{D}$, so $\mathcal{D} = 3$ for all systems in the SCON family. SOC, DOC, and MBC are respectively notated in the system equations as S, D, and M. Thus, x_t, the solutions of the continuous SCON system at time t, expand to the vector,

$$x_t = \begin{bmatrix} S_t \\ D_t \\ M_t \end{bmatrix} \tag{2}$$

and observations of the system y_t are similarly three-dimensional.

We established two SCON versions for inference and data generation use, SCON-C and SCON-SS. SCON-C and SCON-SS share the same underlying α drift vector, equivalent to the deterministic CON dynamics and following the form:

$$\begin{bmatrix} \mathrm{d}S \\ \mathrm{d}D \\ \mathrm{d}M \end{bmatrix} = \begin{bmatrix} \mathcal{I}_S + a_{DS} \cdot k_D \cdot D + a_M \cdot a_{MSC} \cdot k_M \cdot M - k_S \cdot S \\ \mathcal{I}_D + a_{SD} \cdot k_S \cdot S + a_M \cdot (1 - a_{MSC}) \cdot k_M \cdot M - (u_M + k_D) \cdot D \\ u_M \cdot D - k_M \cdot M \end{bmatrix} \mathrm{d}t + \beta^{0.5} \begin{bmatrix} \mathrm{d}W_S \\ \mathrm{d}W_D \\ \mathrm{d}W_M \end{bmatrix} \tag{3}$$

where β now refers to the diffusion matrix component of the SDE and the W_S, W_D, and W_M elements of the Wiener process vector represent random draws from the distribution $\mathcal{N}(0, \sqrt{\mathrm{d}t})$.

For simplification purposes, the β diffusion matrices of both systems were made to be diagonal only and free of covariance diffusion terms. SCON-C diffusion dynamics are given by

$$\beta = \begin{bmatrix} c_S & 0 & 0 \\ 0 & c_D & 0 \\ 0 & 0 & c_M \end{bmatrix} \tag{4}$$

while SCON-SS diffusion dynamics are

$$\beta = \begin{bmatrix} s_S \cdot S & 0 & 0 \\ 0 & s_D \cdot D & 0 \\ 0 & 0 & s_M \cdot M \end{bmatrix} \tag{5}$$

Thus, SCON-C diffusion noise is *additive*, meaning it is independent of the values of states S, D, and M, and also *stationary*, meaning that is not a function of t. Meanwhile, SCON-SS noise is *multiplicative*, meaning it is dependent on the states. As such, SCON-C is linear in drift and diffusion, while SCON-SS is linear in drift but non-linear in diffusion.

$\mathcal{I}_S$ and $\mathcal{I}_D$ respectively represent the exogenous input of C mass in units of $\mathrm{mg\,C\,g^{-1}\,soil\,h^{-1}}$ into the SOC and DOC soil pools from litter decay and can be modeled as constants or functions. We used sinusoidal litter input functions with annual periods that assumed litterfall peaking through late summer and early fall in a pattern resembling those observed in tropical forest ecosystems (Giweta, 2020). The functions are given by

$$\mathcal{I}_{S,t} = 0.001 + 0.0005 \cdot \sin\left(\frac{2\pi}{365 \cdot 24} t\right) \tag{6}$$

$$\mathcal{I}_{D,t} = 0.0001 + 0.00005 \cdot \sin\left(\frac{2\pi}{365 \cdot 24} t\right) \tag{7}$$

As was previously instituted for CON (Allison et al., 2010; J. Li et al., 2014), the SCON linear first-order decay parameters $k_{i\in\{S,D,M\}}$ remain dependent on temperature. Temperature sensitivity of the $k_{i\in\{S,D,M\}}$ linear first-order decay parameters is enforced by a function derived from the original Arrhenius equation (Arrhenius, 1889),

$$k_{i,t} = k_{i,\,\mathrm{ref}} \exp\left[-\frac{Ea_{k_i}}{R}\left(\frac{1}{\mathrm{temp}_t} - \frac{1}{\mathrm{temp}_{\mathrm{ref}}}\right)\right] \tag{8}$$

where R is the ideal gas constant $8.314\ \mathrm{J\,K^{-1}\,mol^{-1}}$ and $\mathrm{temp}_{\mathrm{ref}}$ specifies a "reference" equilibrium temperature which we set at 283 K.

Through changing values of $k_{i\in\{S,D,M\}}$, SCON systems respond to day-night and seasonal temperature cycles through the composite sinusoid forcing function,

$$\mathrm{temp}_t = \mathrm{temp}_{\mathrm{ref}} + \frac{5t}{80 \cdot 365 \cdot 24} + 10 \cdot \sin\left(\frac{2\pi}{24} t\right) + 10 \cdot \sin\left(\frac{2\pi}{365 \cdot 24} t\right) \tag{9}$$

The function assumes a gradual linear increase in mean soil surface temperature by 5 °C over 80 years from the start of the simulation, in line with the upper bound of mean surface temperature increases predicted in emissions scenarios by 2100 (O'Neill et al., 2017).

SDE systems rarely admit tractable analytic solutions. To sample state trajectories accurately approximating SCON-C and SCON-SS dynamics and construct our synthetic time series data y, we used the long-established and reliable Euler-Maruyama SDE solver (Maruyama, 1955) to numerically integrate solution paths x corresponding to θ randomly sampled from logit-normal distributions. Our solver step size was set to $\mathrm{d}t = 0.1$ hour. We note that we recover the exact SCON-C and SCON-SS processes in continuous time as $\mathrm{d}t$ is decreased to 0.

If inference involved conditioning with CO_2 observations in y in addition to state measurements, model CO_2 respiration rate would be computed from the time-corresponding x state values with the equation

$$r_{CO_2,t} = (1 - a_{SD}) \cdot k_{S,t} \cdot S_t + (1 - a_{DS}) \cdot k_{D,t} \cdot D_t + (1 - a_M) \cdot k_{M,t} \cdot M_t \tag{10}$$

where $r_{CO_2,t}$ is in units of $\mathrm{\mu g\,g^{-1}\,soil\,h^{-1}}$. We then sliced x and CO_2 time series at some regular interval, i.e. every 1 hour or 5 hours, and normally sampled about the sliced values with an observation error vector σ_{obs} in the manner of

$$y_t \sim \mathcal{N}(x_t, \eta_{\mathrm{obs}}) \tag{11}$$

to arrive at y. We lower bounded y such that $y \in \mathbb{R}_{\geq 0}$ to preclude nonsense negative state measurements. We used constant η_{obs} that was 10% of the overall state mean such that

$$\eta_{\text{obs}} = 0.1 \odot \begin{bmatrix} \bar{S} & 0 & 0 \\ 0 & \bar{D} & 0 \\ 0 & 0 & \bar{M} \end{bmatrix} \tag{12}$$

where $\odot$ indicates elementwise multiplication. This corresponds to an empirical scenario where measurement instruments and processes introduce a stable level of observation noise.

We generated and conditioned inferences on synthetic y of up to 5000 hours in total timespan T. Data-generating θ distribution hyperparameters were chosen to produce stable and informative state dynamics in a shorter span of time and minimize the memory footprint of the data set under available computing resources. We used elevated $k_{i,\,\text{ref}}$ means compared to previous literature values (Allison et al., 2010; J. Li et al., 2014; Xie et al., 2020). Sampled θ values and T scale are thereby reminiscent of an organic decay process occurring at the soil surface, rather than a slower subterranean decomposition. θ data-generating distribution hyperparameters, equivalent to the prior distribution $p(\theta)$ hyperparameters, along with the biogeochemical interpretations associated with each θ, are detailed in Table S1.

2.3 The generalized univariate logit-normal distribution

We used a univariate logit-normal distribution family for our data-generating, informed prior $p(\theta)$, and mean-field variational posterior $q(\theta|y)$ probability density functions. To avoid being restricted to the standard $[0, 1]$ distribution support that the logit-normal density is typically associated with in statistics, we defined a generalized form of the family whose supports could be enclosed between an arbitrary positive $[a, b]$, where $a, b \in \mathbb{R}_{\geq 0}$ and $b > a$. Generalized logit-normal distributions provide similar utility to truncated normal distributions used previously in SBM inference projects for constraining θ values to finite supports (Xie et al., 2020), but are more stable for backpropagation, as the inverse cumulative distribution function of the truncated normal distribution has inherent stability issues close to support boundaries.

We notate logit-normal distribution parameters in order of desired "target" mean μ, standard deviation σ, support lower bound a, and upper bound b akin to

$$\theta \sim \mathscr{LN}(\mu, \sigma, a, b) \tag{13}$$

Via passage through a sigmoid function, logit-normal distributions are transformed from normal distributions $\mathscr{N}(\check{\mu}, \check{\sigma})$, where $\check{\mu}$ and $\check{\sigma}$ are respectively the "parent" mean and standard deviation distribution parameters:

$$\check{\theta} \sim \mathscr{N}(\check{\mu}, \check{\sigma}) \tag{14}$$

$$\theta^{\text{mid}} = \frac{1}{1 + \exp(-\check{\theta})} \tag{15}$$

$$\theta = (b - a) \cdot \theta^{\text{mid}} + a \tag{16}$$

The logit-normal distribution has no closed form probability density function and its probability moments are not analytically resolvable, so no formula can be deduced that allows us to make random variable transformations between logit normal and normal distributions. Hence, to arrive at a particular logit-normal density with target μ and σ in each VI optimization iteration to sample from, we must first numerically solve for the parent $\check{\mu}$ and $\check{\sigma}$ of a normal distribution that corresponds to the desired logit-normal properties following the transformations from equations (14) to (16). We can do this with root-finding algorithms like the bisection method that search for an appropriate $\check{\mu}$ in the truncated support interval between a and b and $\check{\sigma}$ within a provided range of tolerated standard deviation values (Daunizeau, 2017).

2.4 State space model approximation of the SDE

Instead of optimizing SCON θ via an iterative SDE solver, we optimized time-discretized state space models approximating the SCON-C and SCON-SS SDEs. State space models describe the distribution of a discrete sequence of observations y sourced from discrete latent states x. They can take the general form

$$x_t \sim p(x_t|x_{t-1}, \theta) \tag{17}$$

$$y_t \sim p(y_t|x_t, \theta) \tag{18}$$

Equation (17) indicates that the transition from x_{t-1} to x_t occurs at a probability density of $p(x_t|x_{t-1}, \theta)$ and that subsequent states of a state space model depend on previous ones, thus indicating that x constitutes a Markov chain. Equation (18) specifies that y_t is observed from x_t at a density of $p(y_t|x_t, \theta)$. An initial state x_0 must be nominated to compute x and it can be set as a constant, or informed as a density, $p(x_0)$, which we do in our case.

The state space model θ are the same model parameters as in the SDE counterpart. When accounting for the SDE α drift and β diffusion dynamics, x_t, the latent states of the state space model at time t can be written as

$$x_t = x_{t-1} + \alpha(x_{t-1}, \theta)\Delta t + \epsilon_t \sqrt{\beta(x_{t-1}, \theta)\Delta t} \tag{19}$$

with the same α and β as in (1). ϵ_t is a random noise vector of length $\mathcal{D}$ independently sampled via $\epsilon_t \sim \mathcal{N}(0, I_\mathcal{D})$. $I_\mathcal{D}$ is an identity matrix with number of diagonal elements equal to $\mathcal{D}$. Δt is the state space model time step, not to be confused with SDE solver time step $\mathrm{d}t$. We used $\Delta t = 1.0$ hour for our state space model approximations in contrast to the aforementioned $\mathrm{d}t = 0.1$ for Euler-Maruyama solving of our data generating processes.

There is overlap between SDEs and state space models. Both depict the evolution of state values in a series of steps where future values depend on past ones. Both require the specification of initial conditions x_0 to compute state trajectories.

However, SDEs and state space models treat time differently. A key distinction that makes state space model approximation helpful for inference efficiency is that Δt can be made relatively large versus SDE solver $\mathrm{d}t$. This is helpful for common biological inference and data assimilation situations where y is sparsely observed due to the expense and difficulty of collecting measurements.

Differential equation systems are instead typically numerically integrated and like state space models, are solved in discrete steps, as only smooth analytic solutions can only be obtained from the simplest systems. But, the differential equation integration procedures still assume that states are evolving in continuous time. The integrating solvers almost always require relatively small integration time steps $\mathrm{d}t$ that are as close to 0 as possible; the solvers tend to fail at higher $\mathrm{d}t$.

The divergent handling of time in state space models and SDEs renders them more apt for different objectives. State space models are better suited for estimating observations over long T, whereas SDEs are required for precise analyses of accurately simulated system dynamics. With their differing but related purposes, we can ultimately use state space models to represent discrete observations from continuous SDEs.

2.5 VI optimization

Under a Bayesian statistics framework, the goal of statistical inference broadly consists of estimation of the θ posterior density function for some model, $p(\theta|y)$, conditioned

on some data set y via Bayes' rule,

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} \tag{20}$$

$p(y|\theta)$, also notated as $\ell(\theta|y)$, is the likelihood function, which indicates model goodness-of-fit across various values of individual parameters comprising θ. $p(\theta)$ is the prior probability representing beliefs about θ uncertainty. $p(y)$ is the probability density of the observed data.

The prior density $p(\theta)$ can be specified in an informed fashion, as we did in our workflow with distributions that matched our known data-generating distributions, or with less certainty in the absence of empirical information or domain experience. In most cases, $p(y|\theta)$ is not obtainable in closed analytic form and has to be numerically estimated with methods including Monte Carlo sampling and Laplace approximation (Reid, 2015). Additionally, $p(y)$, sometimes known as the *marginal evidence*, tends to be unresolvable (Gelman et al., 2013; McElreath, 2020). Thus, we take advantage of the proportionality relationship based on (20),

$$p(\theta|y) \propto p(y|\theta)p(\theta) = p(y, \theta) \tag{21}$$

to estimate $p(\theta|y)$ and practically conduct inference.

For Bayesian inference on state space models, we additionally need to account for the transition and observation densities generalized in equations (17) and (18), which influence the θ posterior. Estimation of the posterior of θ in state space model inference must occur along with estimation of the posterior of x, whether in a joint or marginal fashion, in a case such as ours where the transition and observation distributions are not known. We opted for joint estimation. The joint posterior density of θ and x is notated as $p(\theta, x|y)$. We arrive at an expression for $p(\theta, x|y)$ by substituting (17) and (18) into (21):

$$\begin{aligned} p(\theta, x|y) &\propto p(y, \theta, x) && (22)\\ &= p(y|\theta, x)p(\theta, x) && (23)\\ &= p(y|\theta, x)p(\theta)p(x|\theta) && (24)\\ &= p(\theta) \prod_{i \in \mathfrak{N}} p(y_i|x_i, \theta) \prod_{t=1}^{T} p(x_t|x_{t-1}, \theta) && (25) \end{aligned}$$

T denotes the final discretized integer time index of x. Since we set state space model $\Delta t = 1.0$ hour, our final time index matches total synthetic experiment hours and T can signify both. We also use T to represent the set of x state space model discretization indices not including the initial state at $t = 0$. We can then adopt a $\mathfrak{T} \subseteq T$ to indicate the set of y observation time indices not including an initial observation at $t = 0$, which is required in our VI procedure. The total number of x discretizations is $N = T + 1$ when including the $t = 0$ index. $\mathfrak{N} = \{0\} \cup \mathfrak{T}$ notates the full set of y indices.

In stochastic VI on state space models, we optimize a parametric density $q(\theta, x)$ to match the true joint posterior $p(\theta, x|y)$ as closely as possible. Per the probability chain rule, $q(\theta, x)$ expands to,

$$q(\theta, x) = q(x|\theta)q(\theta) \tag{26}$$

The density functions we select for our marginalized $q(\theta)$ and $q(x|\theta)$ are known as our *variational families*. As mentioned in section 2.3, we chose a mean-field logit-normal variational family for $q(\theta)$ that assumed independent univariate distributions per θ such that

$$q(\theta) = q(\theta_1, \theta_2, \ldots, \theta_{\mathcal{P}}) = \prod_{j=1}^{\mathcal{P}} q_j(\theta_j) \tag{27}$$

where $\mathcal{P}$ is the total number of individual SBM θ and $\mathcal{P} = 14$ for SCON-C and SCON-SS (Table S1). We chose a class of normalizing flow called a *neural moving average flow* in Ryder et al. (2021) for our $q(x|\theta)$ variational family, which we will describe subsequently in section 2.6.

We can index individual representatives of our joint variational family by the properties and hyperparameters of the distribution symbolized in aggregate by $\phi_{(\theta,x)}$ such that an instance of $q(\theta, x)$ is notated as $q(\theta, x; \phi_{(\theta,x)})$. $q(\theta, x; \phi_{(\theta,x)})$ can be decomposed into $q(\theta; \phi_\theta)q(x|\theta; \phi_x)$ since $\phi_{(\theta,x)} = (\phi_\theta, \phi_x)$. ϕ are termed *variational parameters*, as they represent the distribution settings that can be varied and tuned to adjust the approximation. For neural network models like flows, variational parameters would include the hidden neural network parameters and weights. A particular distribution can be referred to by its variational parameter index in shorthand.

Our VI framework seeks a set of variational parameters ϕ that minimizes discrepancies between $q(\theta, x; \phi_{(\theta,x)})$ and $p(\theta, x|y)$, the approximate and true posteriors. One measure of distance between distributions customarily employed in statistics and machine learning literature is called the *Kullback-Leibler (KL) divergence*, notated as $D_{\mathrm{KL}}[q(\theta, x; \phi_{(\theta,x)})||p(\theta, x|y)]$ (Kullback & Leibler, 1951; Perez-Cruz, 2008; Joyce, 2011). Targeting the KL divergence for minimization, our optimization objective can then be mathematically stated as,

$$q(\theta, x; \phi^*_{(\theta,x)}) = \mathrm{argmin}_{q(\theta,x;\phi_{(\theta,x)})}(D_{\mathrm{KL}}[q(\theta, x; \phi_{(\theta,x)})||p(\theta, x|y)]) \tag{28}$$

where $\phi^*_{(\theta,x)}$ indexes the set of variational parameters that corresponds to the idealized global KL divergence minimum. After several omitted steps that can be referenced in greater detail from Blei et al. (2017), we proceed from (28) to

$$D_{\mathrm{KL}}[q(\theta, x; \phi_{(\theta,x)})||p(\theta, x|y)] = \mathbb{E}_{q(\theta,x;\phi_{(\theta,x)})}[\log q(\theta, x; \phi_{(\theta,x)})] - \mathbb{E}_{q(\theta,x;\phi_{(\theta,x)})}[\log p(y|\theta, x)] \tag{29}$$

$$= \mathbb{E}_{q(\theta,x;\phi_{(\theta,x)})}[\log q(\theta, x; \phi_{(\theta,x)})] - \mathbb{E}_{q(\theta,x;\phi_{(\theta,x)})}[\log p(y, \theta, x)] + \log p(y) \tag{30}$$

where the expectations $\mathbb{E}$ subscripted with $q(\theta, x; \phi_{(\theta,x)})$ are taken with respect to the density $q(\theta, x; \phi_{(\theta,x)})$.

Reviewing the notion that $p(y)$ and in turn, the log marginal evidence, are typically unavailable (Christensen et al., 2010), we then rely on a reduced and rearranged expression that constitutes the ELBO function, notated as $\mathcal{L}$,

$$\mathcal{L}[\phi_{(\theta,x)}] = \mathbb{E}_{q(\theta,x;\phi_{(\theta,x)})}[\log p(y, \theta, x)] - \mathbb{E}_{q(\theta,x;\phi_{(\theta,x)})}[\log q(\theta, x; \phi_{(\theta,x)})] \tag{31}$$

$$= \mathbb{E}_{q(\theta,x;\phi_{(\theta,x)})}[\log p(y, \theta, x) - \log q(\theta, x; \phi_{(\theta,x)})] \tag{32}$$

$$= \mathbb{E}_{q(\theta,x;\phi_{(\theta,x)})} \langle \log p(y, \theta, x) - \log[q(x|\theta; \phi_x)q(\theta; \phi_\theta)]\rangle \tag{33}$$

$$= \mathbb{E}_{q(\theta,x;\phi_{(\theta,x)})} \langle \log p(y, \theta, x) - \log q(x|\theta; \phi_x) - \log q(\theta; \phi_\theta)\rangle \tag{34}$$

Substituting in (25) for $p(y, \theta, x)$ results in

$$\mathcal{L}[\phi_{(\theta,x)}] = \mathbb{E}_{q(\theta,x;\phi_{(\theta,x)})} \langle \log \left[p(\theta) \prod_{i \in \mathfrak{N}} p(y_i|x_i, \theta) \prod_{t=1}^{T} p(x_t|x_{t-1}, \theta) \right] - \log q(x|\theta; \phi_x) - \log q(\theta; \phi_\theta)\rangle \tag{35}$$

which, recalling that the set of total y indices $\mathfrak{N} = \{0\} \cup \mathfrak{T}$, expands into

$$\mathcal{L}[\phi_{(\theta,x)}] = \mathbb{E}_{q(\theta,x;\phi_{(\theta,x)})} \langle \log p(\theta) + \log p(y_0|x_0, \theta) + \sum_{i \in \mathfrak{T}} \log p(y_i|x_i, \theta) + \sum_{t=1}^{T} \log p(x_t|x_{t-1}, \theta) - \log q(x|\theta; \phi_x) - \log q(\theta; \phi_\theta)\rangle \tag{36}$$

We will decompose the marginal variational log-density of x, $\log q(x|\theta;\phi_x)$, in more granular detail when we describe the architecture of the neural moving average flow in section 2.6.

The ELBO function is called as such because it is the lower bound of the log marginal evidence:

$$\log p(y) = \mathcal{L}[\phi_{(\theta,x)}] + D_{\mathrm{KL}}[q(\theta,x;\phi_{(\theta,x)})||p(\theta,x|y)] \tag{37}$$

$$\geq \mathcal{L}[\phi_{(\theta,x)}] \tag{38}$$

Maximizing $\mathcal{L}[\phi_{(\theta,x)}]$, or minimizing the negative ELBO $-\mathcal{L}$, as we need to do in machine learning libraries like PyTorch that implement gradient descent rather than ascent, is commensurate to minimizing $D_{\mathrm{KL}}[q(\theta,x;\phi_{(\theta,x)})||p(\theta,x|y)]$. Hence, $\mathcal{L}[\phi_{(\theta,x)}]$ is our optimization objective function.

For pithier description of the ELBO gradient, $\nabla\mathcal{L}$, used to update $\phi_{(\theta,x)}$ via automatic differentiation, we set $\log p(y,\theta,x) - \log q(\theta,x;\phi_{(\theta,x)})$ in (32) equal to $\mathscr{R}(\theta,x,y,\phi)$, where $\mathscr{R}$ is a log-density ratio function. This reduces the ELBO equation to

$$\mathcal{L}[\phi_{(\theta,x)}] = \mathbb{E}_{q(\theta,x;\phi_{(\theta,x)})}\left[\mathscr{R}(\theta,x,y,\phi_{(\theta,x)})\right] \tag{39}$$

and the ELBO gradient is

$$\nabla\mathcal{L}[\phi_{(\theta,x)}] = \nabla_\phi \left\langle \mathbb{E}_{q(\theta,x;\phi_{(\theta,x)})}\left[\mathscr{R}(\theta,x,y,\phi_{(\theta,x)})\right] \right\rangle \tag{40}$$

$$= \nabla_\phi \left[\int_\theta \int_x q(\theta,x;\phi_{(\theta,x)}) \mathscr{R}(\theta,x,y,\phi_{(\theta,x)}) \mathrm{d}x\mathrm{d}\theta \right] \tag{41}$$

$$= \int_\theta \int_x \nabla_\phi \left[q(\theta,x;\phi_{(\theta,x)}) \mathscr{R}(\theta,x,y,\phi_{(\theta,x)}) \right] \mathrm{d}x\mathrm{d}\theta \tag{42}$$

which decomposes to

$$\nabla\mathcal{L}[\phi_{(\theta,x)})] = \int_\theta \int_x q(\theta,x;\phi_{(\theta,x)}) \nabla_\phi \left[\mathscr{R}(\theta,x,y,\phi_{(\theta,x)})\right] \mathrm{d}x\mathrm{d}\theta + \int_\theta \int_x \mathscr{R}(\theta,x,y,\phi_{(\theta,x)}) \nabla_\phi \left[q(\theta,x;\phi_{(\theta,x)})\right] \mathrm{d}x\mathrm{d}\theta \tag{43}$$

Note that the gradients ∇_ϕ are taken with respect to the variational parameters. This presents a complication, as examining the second term of (43), we are left with the situation that $\nabla_\phi \left[q(\theta,x;\phi_{(\theta,x)})\right]$ is by and large unavailable, as q can be sampled from, but is usually not analytically differentiable. Our joint variational family q is no exception to that pattern; our marginal mean-field $q(\theta;\phi_\theta)$ has the straightforward analytic form given in (27), but use of the neural moving average flow as the variational family for $q(x|\theta;\phi_x)$ precludes the overall joint density $q(\theta,x;\phi_{(\theta,x)})$ from having an orderly closed form.

To ultimately compute the gradient of an expectation as in (40) in numerical fashion, we thereby turn to the *reparameterization trick* set forth in Salimans and Knowles (2013) and Kingma and Welling (2014). The reparameterization trick involves setting (θ,x) as an output of an invertible, deterministic, and differentiable function $g(z,\phi_{(\theta,x)})$, where z is a random vector sampled from some fixed density $q(z)$. This enables us to tractably take a gradient of a simpler fixed distribution whose probability density function is easier to differentiate and not dependent on the variational parameters ϕ (Ruiz et al., 2016).

In our case, z elements are sampled from standard normal distributions and undergo invertible transformations to proceed to x. θ is still directly sampled from its mean-field logit-normal family described in section 2.5 as part of the operations of g. Hence, $\mathcal{L}$ can be estimated with each VI training iteration with Monte Carlo sampling of z and θ entries starting with the steps

$$z^{(s)} \sim \mathcal{N}(0, I_N) \tag{44}$$

$$\theta^{(s)}, x^{(s)} = g(z^{(s)}, \phi_{(\theta,x)}) \tag{45}$$

where I_N is an identity matrix with number of diagonal entries matching the total x discretization indices N. The superscript (s) indexes an individual draw from a distribution. We can then re-frame (40) from an analytically intractable gradient of an expectation to a numerically assessable expectation of a gradient with

$$\nabla\mathcal{L}[\phi_{(\theta,x)}] = \nabla_\phi \left\langle \mathbb{E}_{q(z)} \left[\mathscr{R}(\theta, x, y, \phi_{(\theta,x)}) \right] \right\rangle \tag{46}$$

$$= \mathbb{E}_{q(z)} \left[\nabla_\phi \left\langle \mathscr{R}(\theta, x, y, \phi_{(\theta,x)}) \right\rangle \right] \tag{47}$$

$$\approx \frac{1}{\mathcal{S}} \sum_{s=1}^{\mathcal{S}} \nabla_\phi \left\langle \mathscr{R}(\theta^{(s)}, x^{(s)}, y, \phi_{(\theta,x)}) \right\rangle \tag{48}$$

$$\widehat{\nabla\mathcal{L}[\phi_{(\theta,x)}]} = \frac{1}{\mathcal{S}} \sum_{s=1}^{\mathcal{S}} \nabla_\phi \left\langle \mathscr{R} \left[g(z^{(s)}, \phi_{(\theta,x)}), y, \phi_{(\theta,x)}) \right] \right\rangle \tag{49}$$

$\mathcal{S}$ denotes the total number of independent θ and z samples drawn per training iteration. $\widehat{\nabla\mathcal{L}[\phi_{(\theta,x)}]}$ specifies the Monte Carlo estimate of $\nabla\mathcal{L}[\phi_{(\theta,x)}]$.

2.6 Masked neural moving average flow architecture

Delineating a normalizing flow more formally than in section 1, a general flow is a chain of *bijections*, or invertible transformation functions mapping an object in a set one-to-one to an object in another set. Flows can be decomposed into

$$x = g(z) = g_M \circ g_{M-1} \circ \cdots \circ g_m \circ \cdots \circ g_1(z) \tag{50}$$

where $\circ$ notates function composition operations and M marks the total number of bijections. In the generative direction, our flow takes us from a random object z to a random object x corresponding to a set of approximated SCON state trajectories.

A generative flow is constructed such that computation of $\log q(x|\theta;\phi_x)$ in (36) is available to facilitate the optimization of $q(x|\theta;\phi_x)$. The log-density of x is available through the change of variables formula:

$$\log q(x) = \sum_{t=1}^{T} \varphi(z_t) - \log|\det J| \tag{51}$$

$$\log q(x) = \sum_{t=1}^{T} \varphi(z_t) - \log \prod_{m=1}^{M} |\det J_m| \tag{52}$$

$$\log q(x) = \sum_{t=1}^{T} \varphi(z_t) - \sum_{m=1}^{M} \log|\det J_m| \tag{53}$$

where J is the Jacobian matrix of the overall transformation and J_m is the Jacobian of bijection g_m with respect to the intermediate transformed variable $g_{m-1} \circ g_{m-2} \circ \cdots \circ g_1(z)$. We use $\varphi(z_t)$ to indicate the log-density of each element of z, z_t. We establish that z here is equivalent to the z introduced in section 2.5, so each $\varphi(z_t)$ is then a unit standard normal log-density in our framework. We notate the density function of z with $q(z)$. Since $q(z)$ is the starting distribution before transformations are layered, it is also termed the *base distribution*.

The particular flow we implemented as the marginal variational family for $q(x|\theta)$ was patterned after the original neural moving average flow introduced in Ryder et al. (2021). Neural moving average flows include *affine* bijections (Dinh et al., 2015; Kingma et al., 2016; Dinh et al., 2017; Papamakarios et al., 2017) among the functions constituting g in which an x^{out} is transformed from an x^{in} in the general form of

$$x^{\text{out}} = \mu + \sigma \odot x^{\text{in}} \tag{54}$$

where $\odot$ represents elementwise multiplication to denote that μ, σ, and x^{in} can be matrices and vectors in addition to scalars, though our explicit situation involves scalars. μ and σ are respectively known as shift and scale values of the bijection and it is required that $\sigma \in \mathbb{R}_+$. Cumulative μ and σ values of a flow are usually implemented as trained outputs of a neural network and are super- and subscripted to identify the transformation layer and input elements they correspond to. They are notated as such by convention and not to be confused with the similarly notated target logit-normal mean and standard deviation parameters in section 2.3.

These linear affine transformations are basic in structure and consequently are individually not so *expressive*, or able to flexibly transition a base distribution into substantially different distributions of varying complexity. However, when layered repeatedly and stacked, their cumulative expressivity increases and with sufficient layers, composite affine functions can come to embody any distribution that is log-concave and bookended by declining density tails (Lee et al., 2021), which represents a large swath of probability distributions.

Neural moving average flows are specifically distinguished from other flows containing affine layers through their execution of affine bijections with 1-dimensional convolutional neural networks (CNNs). To apply 1-dimensional CNNs rather than 2-dimensional CNNs, we note that for systems with $\mathcal{D} > 1$, like SCON family instances, we must begin with z in a 1-dimensional "melted" form that is $\mathcal{D} \cdot T$ elements in length before reshaping the final transformed x to a $\mathcal{D} \times T$ matrix matching the SDE solution structure demonstrated in (2) following the conclusion of g. Thus, in equations (44) and (53), we replace T with $\mathcal{D} \cdot T$ in our implementation.

Through *masking*, in which inputs to the convolution patch are zeroed out through multiplication by weights, the flow is imbued with an *autoregression* property in which the σ_i and μ_i values producing an ith element of an output vector x does not depend and convolve on any element $z_{j \geq i}$ in the base input vector. This autoregression is critical for the intent of arranging the computation of $\sum_{m=1}^{M} \log |\det J_m|$ in (53) to be manageable. The autoregression ensures that J is a diagonal matrix whose non-zero elements are the σ scale parameters underlying the overall transformation, which simplifies calculation of $\det J$ and $\log q(x)$ to

$$\log q(x) = \sum_{t=1}^{T} \varphi(z_t) - \sum_{m=1}^{M} \sum_{t=1}^{T} \log \sigma_t^m \tag{55}$$

where σ_t^m is the shift parameter of the bijection producing the tth term of the mth affine layer output of length T.

Figure 3b portrays a schematic of the autoregressive convolutions and affine bijections used in our specific neural moving average flow implementation. The operations occur within *residual blocks*, component pieces of deep learning networks consisting of organizations of layers oriented toward the mitigation of training and approximation error that can otherwise snowball with greater network depth. Residual blocks do this with the use of *skip connections*, which preserve and carry over output from previous layers to serve as input to subsequent layers and in doing so prevent noisy degradation of information cascading through the network (He et al., 2016).

In each residual block, we perform two masked 1-dimensional convolutions, Convolution A and Convolution B, that each have a kernel length of 3 elements and a stride length of 1. To enshrine autoregressiveness of the flow, Convolution A applies a kernel masked as $[1, 0, 0]$ that outputs a shift and scale value pair. The Convolution A operation and associated affine bijection can be generally expressed as

$$(\mu_i, \sigma_i) = f_i^{\text{A}}(x_{i-1}^{\text{in}}) \tag{56}$$

$$x_i^{\text{out}} = \mu_i + \sigma_i \cdot x_i^{\text{in}} \tag{57}$$

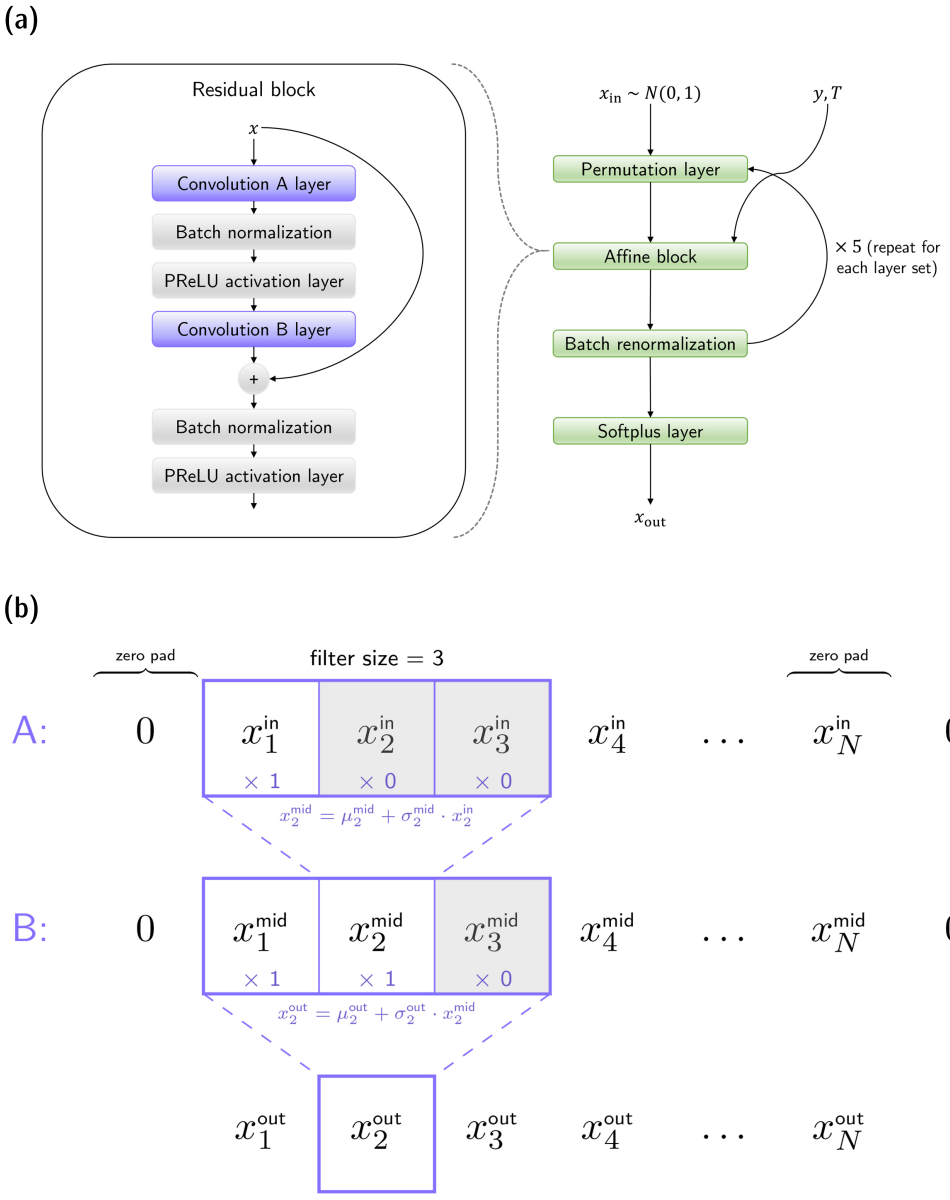

Figure 3. Architecture blueprint of the neural moving average flow used as the marginal variational family for $q(x|\theta)$. **(a)** outlines the sequence of layers and operations. The affine block is a residual block in which the autoregressive convolution operations that distinguish neural moving average flows occur. **(b)** illustrates the two bijections, Convolution A and Convolution B, that link three hypothetical layers x^{in}, x^{mid}, and x^{out} together in each instance of an affine layer in our particular flow. Convolution A applies a $[1, 0, 0]$ mask, while Convolution B applies a $[1, 1, 0]$ mask. The example affine μ and σ parameters are indexed by superscripts and subscripts respectively identifying the layer and element they are associated with.

where μ_i, σ_i, x_i^{in}, and x_i^{out} are scalar elements of vectors and f_i^{A} is the Convolution A operation. The subsequent Convolution B involves a single stride kernel masked as $[1, 1, 0]$ and it can be expressed together with its associated bijection as

$$(\mu_i, \sigma_i) = f_i^{\text{B}}(x_{i-1}^{\text{in}}, x_i^{\text{in}}) \tag{58}$$

$$x_i^{\text{out}} = \mu_i + \sigma_i \cdot x_i^{\text{in}} \tag{59}$$

Combined, the two convolutions in sequence have a total receptive field length of 2.

To be able to produce the μ and σ parameters associated with the affine transformation of vector endpoint elements under autoregressive alignment, both convolutions require *zero padding*, in which zero elements are added to either end of the vector. As can be gleaned from Figure 3b, without zero padding, the kernels producing $(\mu_1^{\text{mid}}, \sigma_1^{\text{mid}})$ and $(\mu_1^{\text{out}}, \sigma_1^{\text{out}})$ would lack 1 element to convolve on, and the kernels sourcing $(\mu_N^{\text{mid}}, \sigma_N^{\text{mid}})$ and $(\mu_N^{\text{out}}, \sigma_N^{\text{out}})$ would by overhang their vectors by 1. A zero pad of length 1 was thereby sufficient for our purposes.

Simplified from our actual implementation and not pictured in Figure 3b is our expansion of input into many *channels*, which are duplicates of the input vector that are stacked on top of each other in a matrix. At each convolution stage, the same kernel is applied in parallel across all the channels. Enlarging channel depth broadens the space of neural network weight values constituting f_i^{A} and f_i^{B} that can be explored per training iteration. We set the number of channels at 96 for both convolutions and did not experiment further with channel depth. Also not pictured in Figure 3b, but implied in Figure 3a, is the injection of auxiliary features extracted from y and observation indices $\mathfrak{N}$ in the form of vectors stacked on top of the input channels to inform training of the neural network weights associated with the shift and scale values. Further elaboration on the incorporation of auxiliary information is available in the supplement of Ryder et al. (2021).

In the overall flow procedure, the convolutions and affine bijections in the affine residual block are linked with other transformations that we organize into repeatable sets of layers. The order of transformations for each layer set is outlined in Figure 3a. Preceding the affine blocks are *order-reversing permutations*, in which element order of a vector input is flipped such that a vector $[x_1^{\text{in}}, x_2^{\text{in}}, ..., x_N^{\text{in}}]$ becomes $[x_1^{\text{out}}, x_2^{\text{out}}, ..., x_N^{\text{out}}] = [x_N^{\text{in}}, x_{N-1}^{\text{in}}, ..., x_1^{\text{in}}]$. Order-reversing permutations are a method of extending the expressivity and stability of a flow by enabling more complex dependency structures while preserving flow autoregression (Papamakarios et al., 2021). We found that adding order reversals allowed us to modestly boost our ELBO learning rates. The permutations can be seamlessly interspersed between other transformations since their absolute Jacobian determinant is valued at 1, so they do not affect the computation of $\log q(x)$.

Differing from the neural moving average flow of Ryder et al. (2021), our flow follows affine blocks with *batch renormalization* transformations. Batch renormalization is a simple extension of *batch normalization*, which is a means of normalizing and regularizing our variational samples such that our optimization is less influenced by random fluctuations in neural network weights and sample characteristics from one training iteration to the next (Ioffe & Szegedy, 2015). Similar in intent but not operation to permutations, batch normalization and renormalization are applied to bolster algorithm stability and flexibility with increasing layer depth. They empirically allow VI algorithms to tolerate higher learning rates (Bjorck et al., 2018), poor initialization of variational parameters ϕ (Zhu et al., 2020), and erratic base distribution $z^{(s)}$ draws.

Batch normalization and renormalization overlap in the following steps that compute a *batch mean* $\mu_{\mathcal{S}}$ and *batch standard deviation* $\sigma_{\mathcal{S}}$ from input x^{in} samples, not to

be confused with the affine bijection and logit-normal μ and σ:

$$\mu_{\mathcal{S}} = \frac{1}{\mathcal{S}} \sum_{s=1}^{\mathcal{S}} x_s^{\text{in}} \tag{60}$$

$$\sigma_{\mathcal{S}} = \sqrt{\varepsilon + \frac{1}{\mathcal{S}} \sum_{s=1}^{\mathcal{S}} (x_s^{\text{in}} - \mu_{\mathcal{S}})^2} \tag{61}$$

where ε is a small constant added for stability. $\mu_{\mathcal{S}}$ and $\sigma_{\mathcal{S}}$ are involved in computation of the optimization objective—again, $\mathcal{L}[\phi_{(\theta,x)}]$ for our purposes—during the model training phase. They also update a lagging *running average* $\mu_{\mathcal{R}}$ and *running mean* $\sigma_{\mathcal{R}}$ that are less sensitive to change. $\mu_{\mathcal{R}}$ and $\sigma_{\mathcal{R}}$ are used after training of the model—the joint variational family $q(\theta, x; \phi_{(\theta,x)})$ in this setting—has been halted to estimate the objective metric at the testing stage.

In the testing phase, batch renormalization and normalization are equivalent in transforming input to output:

$$x^{\text{mid}} = \frac{x^{\text{in}} - \mu_{\mathcal{R}}}{\sigma_{\mathcal{R}}} \tag{62}$$

$$x^{\text{out}} = \gamma \cdot x^{\text{mid}} + \Upsilon \tag{63}$$

The collection of γ_t^m and Υ_t^m parameters in each flow layer set are learned neural network outputs. Batch renormalization diverges from batch normalization during training with the steps

$$r = \frac{\sigma_{\mathcal{S}}}{\sigma_{\mathcal{R}}} \tag{64}$$

$$d = \frac{\mu_{\mathcal{S}} - \mu_{\mathcal{R}}}{\sigma_{\mathcal{R}}} \tag{65}$$

$$x^{\text{mid}} = \frac{x^{\text{in}} - \mu_{\mathcal{S}}}{\sigma_{\mathcal{S}}} \cdot r + d \tag{66}$$

$$x^{\text{out}} = \gamma \cdot x^{\text{mid}} + \Upsilon \tag{67}$$

where r and d are variable correction factors. r and d are intended to limit the divergence between batch and running sample characteristics. r is clipped between the interval $[1/r_{\max}, r_{\max}]$, where $r_{\max}$ is gradually increased to 3 over the course of inference, and d is clipped between the interval $[-d_{\max}, d_{\max}]$, where $d_{\max}$ is gradually increased to 5. These intervals were established based on guidelines from previous empirical work (Ioffe, 2017). Batch normalization is a special case of batch renormalization where $r = 1$ and $d = 0$.

Batch renormalization's changes more tightly correlate the batch and running sample characteristics and have been documented to minimize discrepancy between train and test objectives (Ioffe, 2017). We observed this with our ELBO results, where consistent gaps remained between the train and test $\mathcal{L}[\phi_{(\theta,x)}]$ until we swapped batch normalization for renormalization. Batch renormalization also improves training on low batch sizes (Ioffe, 2017; Summers & Dinneen, 2020), and in our position where variational path samples were limited by GPU video memory constraints, renormalization was helpful for decreasing the total number of training iterations we needed for algorithm convergence.

With batch (re)normalization layers, $\log q(x)$ accrues log determinant Jacobian summation terms corresponding to those transformations and develops from (55) to become, in the training phase,

$$\log q(x) = \sum_{t=1}^{T} \varphi(z_t) - \sum_{m=1}^{M} \sum_{t=1}^{T} \left[\log \sigma_t^m - \log r_t^m - \log \gamma_t^m + \log \sigma_{\mathcal{S},t}^m \right] \tag{68}$$

or in the testing phase,

$$\log q(x) = \sum_{t=1}^{T} \varphi(z_t) - \sum_{m=1}^{M} \sum_{t=1}^{T} \left[\log \sigma_t^m - \log \gamma_t^m + \log \sigma_{\mathcal{R},t}^m\right] \tag{69}$$

where we now take M to mark the total number of layer sets rather than layers as we did before in (55). This assumes that each layer set always includes 1 single affine block and 1 batch renormalization layer. Substituting (68) or (69) into (36) for $\log q(x|\theta;\phi_x)$ leads respectively to our fully decomposed train or test $\mathcal{L}[\phi_{(\theta,x)}]$ calculation unless an optional single softplus transformation is used to ensure constraint of flow output to $\mathbb{R}_{\geq 0}$. In that case, the resulting train $\log q(x)$ is

$$\log q(x) = \sum_{t=1}^{T} \varphi(z_t) - \sum_{m=1}^{M} \sum_{t=1}^{T} \left[\log \sigma_t^m - \log r_t^m - \log \gamma_t^m + \log \sigma_{\mathcal{S},t}^m\right] - \sum_{t=1}^{T} \log(-\mathrm{e}^{-x_t} + 1) \tag{70}$$

where x is our terminally transformed random variable following softplus constraint. Setting

$$\lambda_t = \varphi(z_t) - \log(-\mathrm{e}^{-x_t} + 1) - \sum_{m=1}^{M} \left(\log \sigma_t^m - \log r_t^m - \log \gamma_t^m + \log \sigma_{\mathcal{S},t}^m\right) \tag{71}$$

$$\log q(x) = \sum_{t=1}^{T} \lambda_t \tag{72}$$

our fully decomposed train $\mathcal{L}[\phi_{(\theta,x)}]$ calculation that we use in each iteration of VI optimization (Algorithm 1) then consolidates from (36) into

$$\mathcal{L}[\phi_{(\theta,x)}] = \mathbb{E}_{q(\theta,x;\phi_{(\theta,x)})}\langle \log p(\theta) + \log p(y_0|x_0,\theta) - \log q(\theta;\phi_\theta) + \sum_{i\in\mathfrak{T}} \log p(y_i|x_i,\theta) + \sum_{t=1}^{T} [\log p(x_t|x_{t-1},\theta) - \lambda_t]\rangle \tag{73}$$

with softplus flow termination. The test ELBO equation is equivalent except for use of a different λ_t assignment that lacks the $\log r_t^m$ term and swaps $\sigma_{\mathcal{S},t}^m$ for $\sigma_{\mathcal{R},t}^m$.

We note that it is not required for the total permutation layers, affine blocks, and batch renormalization layers constituting a neural moving average flow architecture to match in count; we can choose to omit certain layers in a layer set. To slightly reduce the neural network size, we would frequently use 1 less batch renormalization layer than total affine blocks or permutation layers, omitting batch renormalization in the first layer set since we empirically observed little qualitative difference in visual fit quality between running with 3, 4, or 5 batch renormalizations. If the numbers of affine blocks and batch renormalization layers do not match, then the log Jacobian determinant summations in (68) to (71) need to be adjusted accordingly.

It is apparent that each layer set of our neural moving average flow corresponds to a matrix of hidden parameters, including affine and batch renormalization parameters, of dimensions $[T, h]$, where h is the count of hidden parameters per layer set. Thus, when conditioning on long, dense T data that is complex in such a manner that would require many layer sets for flow representation, we note that a different choice of marginal variational family for $q(x|\theta)$ aside from the neural moving average flow may be appropriate for minimizing computational expense.

Algorithm 1: Synopsis of the operations occuring in each iteration of our soil biogeochemical state space model VI framework

Data: Time series matrix y of soil pool state and other observations, including CO_2 respiration measurements

Result: $q(\theta, x; \phi_{(\theta,x)})$ corresponding to the $\mathcal{L}[\phi_{(\theta,x)}]$ value at the stoppage of stochastic gradient optimization

Define $q(\theta; \phi_\theta)$ and $q(x|\theta; \phi_x)$;
Initialize (ϕ_θ, ϕ_x);
$n \leftarrow$ total desired training iterations;
for $i \leftarrow 1$ **to** n **do**
  for $s \leftarrow 1$ **to** $\mathcal{S}$ **do**
    Draw $\theta^{(s)} \sim q(\theta; \phi_\theta)$;
    Draw $x^{(s)} \sim q(x|\theta; \phi_x)$ transformed from $z^{(s)}$;
  end
  Compute $\mathcal{L}[\phi_{(\theta,x)}]$ (or $-\mathcal{L}[\phi_{(\theta,x)}]$ for gradient descent) as per (73);
  Compute the gradient $\nabla\widehat{\mathcal{L}[\phi_{(\theta,x)}]}$ from (49) with automatic differentiation;
  Update variational parameters $\phi_{(\theta,x)}$ based on the gradient;
end

2.7 Kalman smoother validation

When a state space model is linear in drift and its diffusion is stationary and additive, as is the state space model approximation of SCON-C, the posterior density $p(x|y)$ can be determined analytically and precisely in closed form with the Kalman smoother algorithm, provided the algorithm is fed the true θ and observation noise (Kalman, 1960; Rauch et al., 1965). Flow VI in contrast can only numerically estimate $p(x|y)$ through a variational approximation, but has the critical advantage of being capable of functioning without exact knowledge of θ given uninformed prior distributions and is able to estimate the joint density $p(x, \theta|y)$ via variational approximations. Thus, comparing a Kalman-derived true $p(x|y)$ to a post-optimization $q(x|\theta; \phi_x)$ can be a revealing means of benchmarking flow approximation performance and accuracy before applying an architecture with confidence to approximation, optimization, and θ inference of models like SCON-SS that cannot be resolved by the smoother.

The Kalman smoother procedure is a two part process consisting of a *forward pass* followed by a *backward pass*. The forward pass computes a "filtering" posterior $p(x_t|y_{0:t})$, which notates the posterior of x_t given observations up to the time indexed by t, going forward in time from $t = \{0, \ldots, T\}$. The backward pass computes a "smoothing" posterior $p(x_t|y)$, which notates the posterior of x_t given all observations, going backward in time from $t = \{T, \ldots, 0\}$. Reconciling the "filtering" and "smoothing" posteriors produces the true $p(x|y)$. A comprehensive explication of Kalman smoothing is available in Särkkä (2013).

2.8 Flow neural network training tuning choices

We settled on using 5 layer sets of permutation, affine, and batch renormalization layers for our neural moving average flow. This offered qualitatively superior fits over flow architectures with lower layer set counts. For inferences of duration $T = 5000$ with $\Delta t = 1.0$ with 5 layers, maximum training batch size $\mathcal{S}$ at 16 GB of VRAM was 31, so we set $\mathcal{S} = 31$. For $T = 1000$, we used $\mathcal{S} = 150$, though use of smaller $\mathcal{S}$ also appeared functional. For $T = 5000$ inferences we used 120000 non-warmup ELBO train-

ing iterations. For $T = 1000$ inferences we used 60000 non-warmup ELBO training iterations.

With respect to gradient optimizers including AdaMax (Kingma & Ba, 2015), which was the particular optimizer we selected to carry out gradient descent, the *learning rate* is a hyperparameter that scales the objective gradient and in doing so regulates the extent to which neural network weights can updated with each training iteration. The learning rate can be adjusted over the course of training based on a schedule. It is frequently decayed over the course of training to promote convergence of our objective function toward a maximum (for gradient ascent) or minimum (for gradient descent) ((You et al., 2019)). We chose a step decline schedule for learning rate decay. For our $T = 5000$ inferences, we started with a pre-decay ELBO learning rate of 1×10^{-2} and decayed it by a factor of 0.6 every 10000 iterations. For our $T = 1000$ inferences, we started with a pre-decay learning rate of $4{\times}10^{-3}$ and decayed it by a factor of 0.6 every 5000 iterations.

We employed *training warmup*, in which we began optimization with a phase of low learning rate at $1{\times}10^{-6}$ before increasing the rate to its initial pre-decay levels. As has been demonstrated previously (Goyal et al., 2017), we found warmup allowed us to use higher pre-decay learning rates, experience more stable ELBO loss trajectories, and converge to lower average ELBO values over training (Figure S1). We found 5000 warmup iterations to be sufficient for those purposes.

2.9 Software and hardware

With respect to the computational software and hardware powering the inference operations, our DGP and inference code was developed for a Python 3.9.7 environment distributed by Anaconda (*Anaconda Software Distribution*, 2020) and used the Numpy 1.20.3 (Harris et al., 2020) and PyTorch 1.10.2 (Paszke et al., 2019) software libraries. PyTorch 1.10.2 was compiled with the Nvidia CUDA 10.2 toolkit. The inferences were run on one Nvidia Tesla V100 GPU at a time updated to CUDA version 11.4.0 with a maximum of 16 GB of video random access memory and two Intel Xeon Gold 6148 CPU cores clocked at 2.40 GHz on the University of California, Irvine HPC3 cluster. Our flow VI framework code modules, data-generating notebooks, and synthetic data are available via the address `https://doi.org/10.5281/zenodo.6969782`.

The deterministic CON $p(\theta|y)$ posteriors compared with flow VI $q(\theta;\phi_\theta)$ in Figure 5 were estimated using Stan's NUTS algorithm, which is an extension of the Hamiltonian Monte Carlo inference algorithm (Hoffman & Gelman, 2014). Application of Hamiltonian Monte Carlo for data assimilation and inference of SBMs is further described in Xie et al. (2020) and intuition behind the algorithm can be found in Betancourt (2017). The Stan inference was conducted on a 2017 Intel MacBook Pro in an R 4.0.4 environment using Stan 2.29.1 (Carpenter et al., 2017) through the CmdStanR interface (Gabry & Češnovar, 2021). The NUTS simulation ran with 2 chains of 1000 warmup iterations and 5000 sampling iterations each. In our experience, 1000 warmup iterations were sufficient for locating the bulk of the posterior density.

3 Results

We generated synthetic y of various lengths, dimensions (i.e. whether CO_2 observations were included in addition to state information), and regular observation densities (i.e. whether we observed measurements from our SCON family data generating processes every 1 or 5 hours). We explored the validity of our state space model VI approach for data assimilation and posterior identification of model θ with inferences conditioned on those y. Below, our results suggest the neural moving average flow framework was functional for approximating the SCON family of SDE systems as state space models,

fitting y, constraining posteriors, and recovering some true θ values. We also demonstrate subsequently that stochastic gradient optimization in our case was more stable, efficient, and capable at θ identification than an MCMC procedure involving deterministic ODE models adapted from Xie et al. (2020) conditioned on the same y.

3.1 Flow-approximated SCON-C converges to fit synthetic data

Following optimization, an SCON-C state space model approximated by our neural moving average flow implementation reasonably assimilated a $T = 5000$ hour y produced by an SCON-C data-generating process that included CO_2 observations (Figure 4a). The relatively flat $-\mathcal{L}[\phi_{(\theta,x)}]$ trajectory steadily hovering between -1550 and -1600 in the latter half of variational training iterations indicates that our flow VI algorithm converged to a local ELBO minimum (Figure S2).

The mean of the marginal posterior density of latent states $q(x|\theta;\phi_x)$ was estimated from 250 x samples drawn from the joint variational density after ELBO training. The mean latent SOC, DOC, and MBC paths and state-derived CO_2 measurements corresponding to the SCON-C flow sit centrally between the y data points and observation noise across the entire time series (Figure 4a). The latent means are able to adhere to many of the sharp peaks and valleys in the dynamics of the data and the flow CO_2 mean was able to reproduce the rapid oscillatory behavior of the observed CO_2 time series.

Upon closer qualitative inspection and comparison to the true latent distribution computed by a Kalman smoother (Figure 4b), we note the presence of visual discrepancies between the Kalman and flow means and 95% $q(x)$ diffusion distribution intervals. Firstly, the extent of SOC diffusion noise is substantially underestimated by the flow, which is line with documentation in literature that a mean-field VI approach tends to underestimate posterior uncertainty compared to more complex full-rank approaches (Kucukelbir et al., 2017). For the other two states, DOC and MBC, the extent of diffusion noise is more consistent to that which is observed in the Kalman output, but the flow DOC and MBC densities and means appear noisier and more uneven than the Kalman means.

Still, the flow encouragingly is generally congruous with the true Kalman solution in dynamics. The flow means fall entirely within the bounds of the 95% Kalman diffusion interval from $t = 0$ to 500 as can be seen in Figure 4b and we observed for this particular optimization that they almost always remain within those Kalman diffusion bounds through the rest of the time series. Also, we see that the CO_2 mean and distribution calculated from the 250 SCON-C state space model x draws closely matches their Kalman counterparts. The ability of the flow to align with the Kalman smoother in latent state densities improves our confidence in the ability of the neural moving average flow to approximate systems that are non-linear in diffusion, like SCON-SS.

3.2 SCON-C flow VI marginal θ posteriors indicate appropriate optimization

Beyond fitting data, we needed to ascertain that proper posterior optimization was occurring for confidence in inference algorithm function. In our setting, we would expect our posterior densities to at least always be as informed and certain about θ values as our prior densities, not less. With a mean-field logit-normal variational family for $q(\theta;\phi_\theta)$, evidence of suitable optimization would come in the form of marginal posterior densities being narrower than priors to indicate greater certainty after the introduction of information from y along with posterior means separating from prior means and approaching the true θ used by the data-generating process.

Figure 5 indicates that valid posterior optimization indeed occurred in our SCON-C state space model inference to support the notion that our flow VI framework was func-

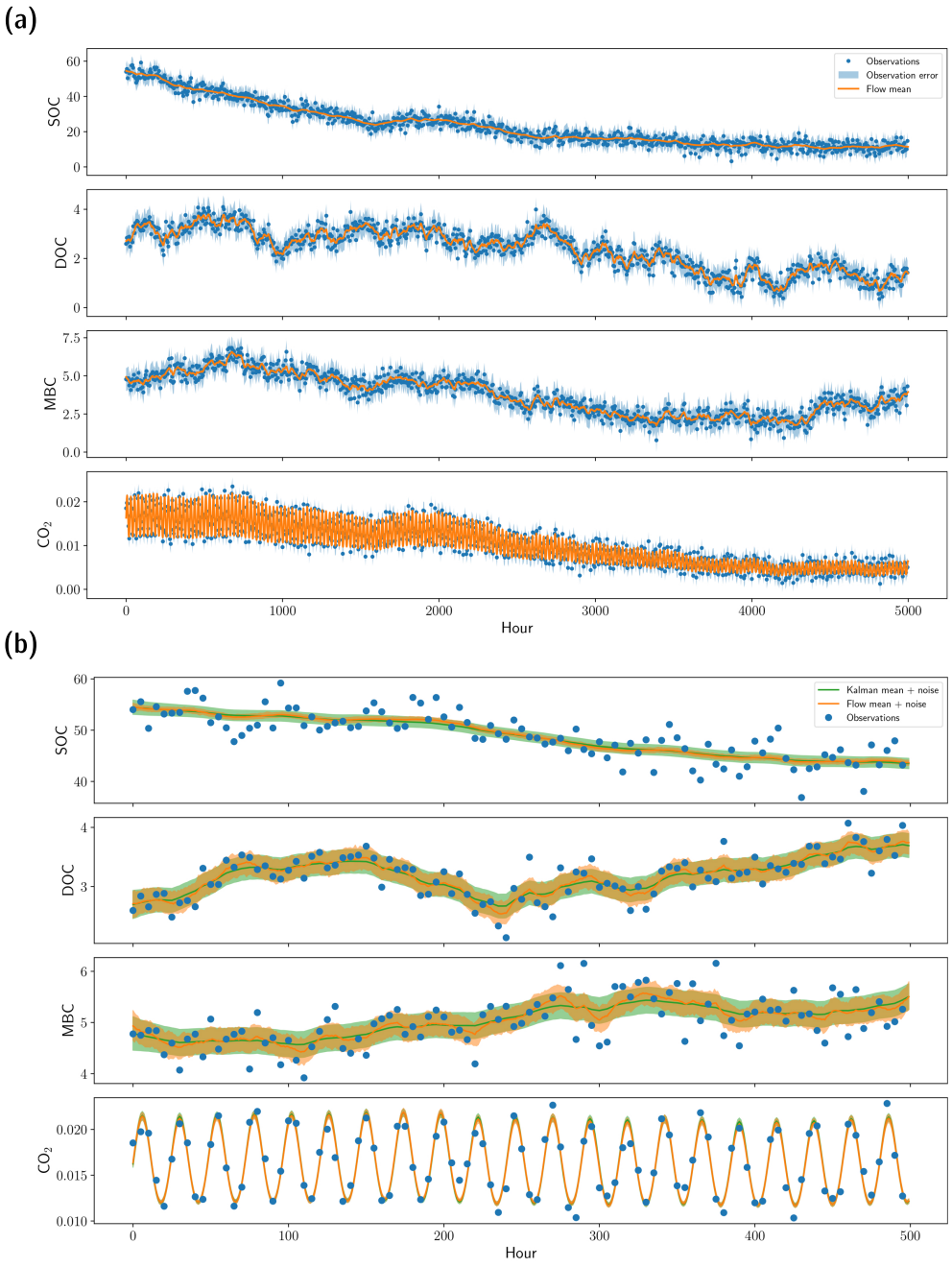

Figure 4. Marginal posterior $q(x|\theta;\phi_x)$ soil pool state means (orange lines) of the SCON-C state space model approximated by the neural moving average flow following VI optimization. The means are estimated from 250 x samples drawn from the optimized joint density $q(\theta, x; \phi_{(\theta,x)})$. The states are in units of mg C g^{-1} soil. In **(a)**, the trajectories of flow-approximated state means are compared to the synthetic observations an SCON-C $T = 5000$ hour y backgrounded by the 95% interval of the observation noise (blue dots over blue shading). In **(b)**, we zoom into a subset of the above plot from $t = 0$ to 500 hour and additionally compare the state means and 95% interval of the diffusion distribution of the optimized model to the true posterior means and 95% diffusion noise computed by a Kalman smoother with knowledge of the true θ values.

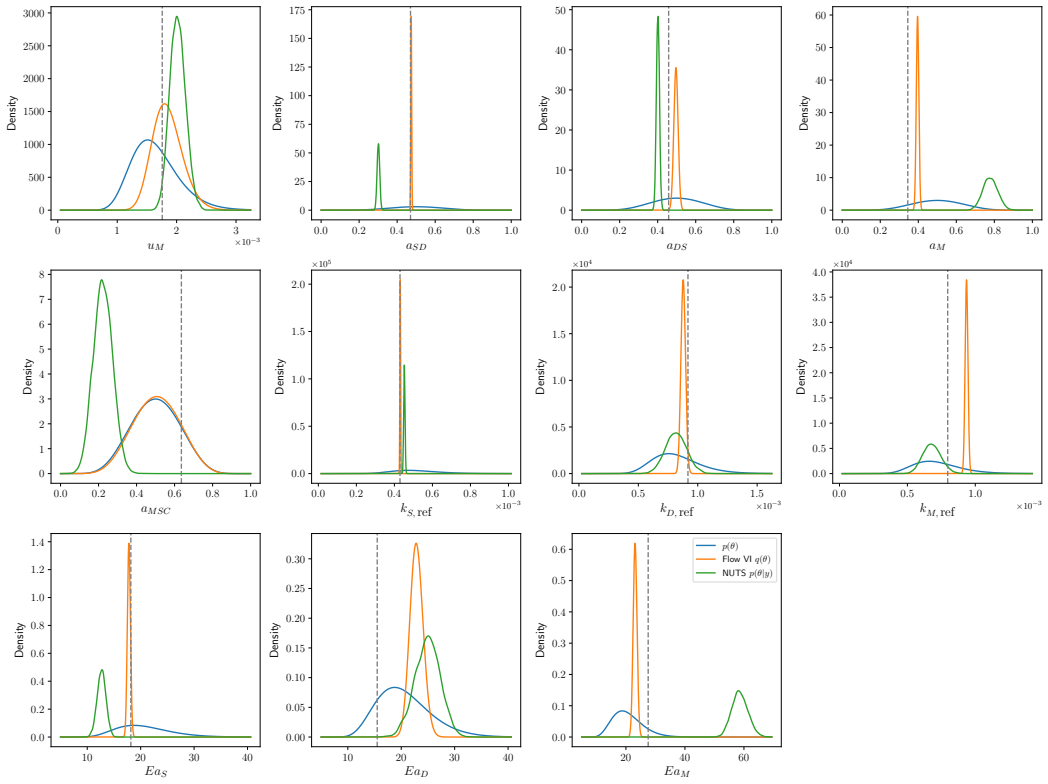

Figure 5. SCON-C state space model marginal $q(\theta;\phi_\theta)$ posterior densities following flow VI optimization (orange) compared to mean-field prior $p(\theta)$ densities (blue) and non-parametric CON ODE marginal $p(\theta|y)$ posterior densities estimated with Stan's NUTS algorithm (green). Flow VI and NUTS were conditioned on the same $T = 5000$ hour y generated by an SCON-C SDE. The true θ values sampled during data generation are marked by vertical dashed gray lines. Being a deterministic ODE system, CON does not have β diffusion θ, so subplots portraying the marginal $q(\theta;\phi_\theta)$ densities for the SCON-C state space model c_S, c_D, and c_M θ were not included in this figure due to a lack of comparison.

tional. Almost all the marginal posterior densities narrowed compared to the priors with the information learned from y by the algorithm. Moreover, many of the marginal $q(\theta;\phi_\theta)$ means drifted closer to the true θ, including the means of u_M, a_{SD}, and $k_{S,\,\mathrm{ref}}$.

We contrasted the flow VI parametric $q(\theta;\phi_\theta)$ posterior densities to the non-parametric $p(\theta|y)$ posterior densities estimated with an SBM inference framework conditioned on the same $T = 5000$ SCON-C y that was previously applied in Xie et al. (2020). This prior framework involves Stan's NUTS algorithm and can only infer θ of deterministic models, so the CON system that the SCON family was parameterized from served as the basis for inference in this approach. With the flexibility and stability afforded by the ability of stochastic optimization to adjust for poor initial condition proposals, noisy state path fluctuations, and outlier observations, the flow VI framework expectedly outperformed the deterministic NUTS workflow. The flow VI marginal $q(\theta;\phi_\theta)$ densities were all-around better informed and identified, exemplified by the subplots corresponding to the u_M, a_{SD}, a_M, $k_{S,\,\mathrm{ref}}$, $k_{D,\,\mathrm{ref}}$, Ea_S, and Ea_M θ (Figure 5). Moreover, some NUTS posterior densities, including those corresponding to the a_{MSC}, a_M, and Ea_S θ, consolidated near their lower or upper support bounds, which points to the deterministic model inference method compensating for its lack of versatility with more extreme θ proposals.

Scrutiny of the posterior for the transfer fraction parameter a_{MSC} brings the issue of θ identifiability limitations to our attention. We see that the SCON-C flow VI marginal a_{MSC} posterior density barely budged from the a_{MSC} $p(\theta)$ density post-optimization (Figure 5). For good posterior identifiability, the a_{MSC} posterior should both narrow substantially to signal reduced uncertainty and shift its density peak toward the true a_{MSC} value.

3.3 Flow VI can effectively assimilate both full and reduced SCON-SS state space approximations

After visually demonstrating the ability of the flow VI framework to optimize $q(\theta, x;\phi_{(\theta,x)})$ through the fitting of the approximated SCON-C state space model to synthetic SCON-C y and the informing and identification of some marginal $q(\theta;\phi_\theta)$ densities, we proceeded to test if the flow VI approach could similarly function with a moderately more complex model in SCON-SS that is non-linear in diffusion.

Reviewing the fact that the SCON-SS state space model diffusion is not stationary or additive, it was no longer possible for us to validate SCON-SS $q(x|\theta;\phi_x)$ estimated from post-optimization x samples against a true $p(x|y)$ determined by a Kalman smoother. Nonetheless, we observed that flow VI was able to optimize $q(\theta, x;\phi_{(\theta,x)})$ adequately enough to fit the approximated SCON-SS state space model $q(x|\theta;\phi_x)$ means to $T = 5000$ y generated by an SCON-SS SDE. As was the case for the SCON-C flow, the mean latent SOC, DOC, and MBC trajectories of the trained SCON-SS flow traced a central route through the observed state values and diffusion noise (Figure S3). The trajectories were able to follow the peaks and valleys of the state dynamics recorded in y, and the flow CO_2 mean derived from the sampled states tightly replicated the y CO_2 oscillations.

SCON-SS $q(\theta;\phi_\theta)$ posterior densities were consistent with proper optimization from information learned in y. Juxtaposed with priors $p(\theta)$, marginal $q(\theta;\phi_\theta)$ densities mostly narrowed and did not move drastically away from their corresponding true θ to inhibit identifiability (Figure S4). There were clear exceptions for the state-scaling diffusion θ posteriors due to reasonable flow neural network approximation error that prompts an overestimate of diffusion noise and, again, for the a_{MSC} posterior. The modest shift of some θ posterior means away from the true θ is counterbalanced by movement of other related θ, like in the circumstance of the Ea_D posterior mean being counterbalanced by Ea_M to satisfy equation (10).

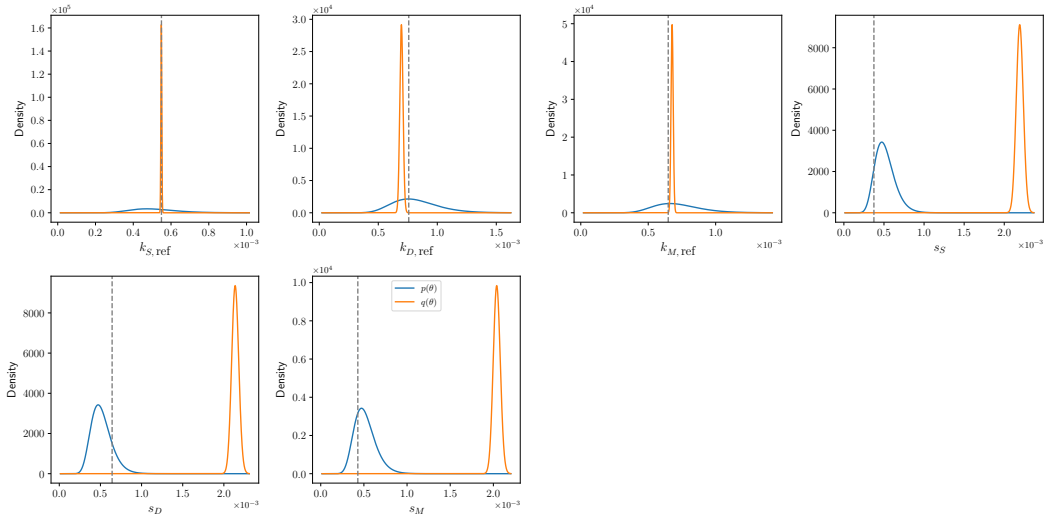

Figure 6. Optimized marginal posterior $q(\theta;\phi_\theta)$ densities (orange) of a reduced SCON-SS model with all but the $k_{i\in S,D,M,\,\mathrm{ref}}$ decay and state-scaling diffusion θ fixed compared to mean-field priors $p(\theta)$ (blue).

For one more test to corroborate proper functioning of our flow VI framework, we established a reduced SCON-SS model with all θ fixed in value except for the $k_{i\in S,D,M,\,\mathrm{ref}}$ linear decay and state-scaling β diffusion parameters. Mirroring above procedures, a synthetic $T = 5000$ y was produced by a reduced SCON-SS data-generating process to condition an SCON-SS state space model optimization. For an appropriately behaving inference algorithm, we would expect that removing degrees of freedom should bolster θ identifiability.

We verified that identifiability was indeed clarified and improved in the remaining drift θ (Figure 6). The marginal $k_{S,\,\mathrm{ref}}$ $q(\theta;\phi_\theta)$ posterior density was tightly constrained right about the true $k_{S,\,\mathrm{ref}}$ value. The $k_{D,\,\mathrm{ref}}$ and $k_{M,\,\mathrm{ref}}$ posterior means did not align exactly with their true θ, but unambiguously offset each other in a manner that plainly fulfilled (3) and (10).

We were unable to fix the state-scaling diffusion θ without breaking the flow VI framework, as it became apparent that the algorithm needed to maintain the ability to overestimate diffusion noise to work. This makes intuitive sense as the flow neural network approximation process will always come with some amount of noisy approximation error that adds to the base diffusion of the unapproximated system. The algorithm can no longer work if the flow diffusion noise needs to be fixed at about the same level as it is in the unapproximated system, as it leaves no room for the approximation error to overflow into. So, with the diffusion θ left unfixed during VI training, the algorithm once more overestimated their $q(\theta;\phi_\theta)$ means, but this is not a cause for concern since the discrepancy can be explained by neural network approximation error.

3.4 Increasing information in y alters SCON posterior certainty and identifiability

The preceding results all involved y that had CO_2 respiration observations included. Inference conditioned on just state observations is also possible and in our experience was able to fit the data well, but it was much less effective for constraining posteriors and identifying θ (Figure S5). Without CO_2 information, marginal $q(\theta;\phi_\theta)$ posterior den-

sities tended to be wider and less informed and density means were frequently farther away from true θ, as exemplified by panels corresponding to the a_{SD} and Ea_S SCON-C θ in Figure S5.

We separately observed that increasing the amount of information in y by lengthening duration T of the time series greatly benefitted posterior identifiability (Figure S6). Alternatively, θ identifiability was boosted without elongating T by bolstering observation density. Individually, the two actions trade off between improvements. In comparison to densifying observations across $T = 1000$ such that the set of observation indices $\mathfrak{N}$ matches the set of state space model discretization indices N, extending T to 5000 more tightly constrained $q(\theta; \phi_\theta)$ posterior densities for all θ and concentrated a_{SD}, $k_{S,\,\mathrm{ref}}$, $k_{D,\,\mathrm{ref}}$, Ea_S, and Ea_M posterior means closer to the true θ.

However, increasing observation density for $T = 1000$ data had the benefit of further constraining posterior densities without also enlarging the divergence in identification of the true SCON-C β diffusion θ, c_S, c_D, and c_M by the means of their corresponding $q(\theta; \phi_\theta)$ densities. The enlarged divergence and uncertainty of the diffusion θ in the $T = 5000$ hour inference compared to the $T = 1000$ inferences is not unexpected. Cumulative approximation error of state space x trajectories compounds for the flow with greater T in a manner typical to approximation methods. Larger accrued approximation error then corresponds to estimation of greater diffusion noise during inference.

4 Discussion

We developed a stochastic SBM data assimilation and inference framework that is a versatile, stable, and computationally efficient alternative to MCMC approaches assimilating deterministic ODE systems, especially when GPU hardware is available. The framework involves approximation of SBMs as state space models whose state trajectories can be sampled at reduced computational and temporal cost in comparison to SDEs.

In our demonstration, we carried out state space model approximation with a class of normalizing flows called neural moving average flows that successively transition random variables from simpler to more complex distributions with the stacking of neural network layers. We applied this framework to fit approximated representatives of the SCON family of SBMs to synthetic data. Conditioning with synthetic rather than empirical data allowed us to visualize discrepancies between estimated posterior densities, data-generating densities, and true θ values used by the data-generating process for an assessment of framework performance.

Flow-approximated SCON-C state trajectories were able to effectively track state and CO_2 observations after variational optimization and graphically align with the true latent state distributions determined by a Kalman smoother. Following Kalman validation of our SCON-C inference, we then successfully assimilated synthetic observations and estimated posteriors with SCON-SS, which is non-linear in diffusion and modestly more complex than SCON-C.

4.1 More data promotes model θ identifiability and constraining of posteriors

In terms of implications for experimental work focused on producing data sets suitable for SBM inference and data assimilation, we firstly recommend that CO_2 respiration measurements be collected and included in y. CO_2 information is highly beneficial for informing the posteriors of SBMs like SCON for which CO_2 efflux equations have been established (Figure S5). Additionally, the collection of supplemental measurements, such as radiolabeled C densities linked to SBM pool transfer fraction θ, should further constrain and identify θ posteriors. Our results indicate that just the CO_2 and state obser-

vations were not enough to effectively identify the marginal posterior of the SCON MBC-to-SOC transfer θ, a_{MSC} (Figure 5, S4, S5, S6).

With respect to a_{MSC} posterior identifiability or lack thereof, inspection of the SCON system drift in (3) and CO_2 efflux rate equation in (10) suggests that the consistent lack of identifiability is not the consequence of a general algorithm issue but instead stems from a dearth of information in y to further constrain the marginal a_{MSC} $q(\theta;\phi_\theta)$ density. The a_{MSC} parameter appears in two terms of (3) that are each the product of four elements, the $a_M \cdot a_{MSC} \cdot k_M \cdot M$ term in the dS equation denoting C mass transfer from the MBC to SOC soil pool and the $a_M \cdot (1-a_{MSC}) \cdot k_M \cdot M$ term in the dD equation denoting C transfer from the MBC to DOC soil pool. The posterior densities of the a_M and k_M θ in those terms appear in (10) and are accordingly better constrained and identified with CO_2 measurements in y. This is not the case for a_{MSC}, which is not present in (10). Informing of a_{MSC} can thereby only occur through the state measurements in y, and as only one element in the drift product terms, a_{MSC} can take many values between its $[0, 1]$ support bounds without greatly affecting the products.

Furthermore, our results suggests that raising both study time or data collection frequency would improve posterior estimation accuracy and identifiability in our framework (Figure S6). But, under budget and personnel limitations, empiricists creating inference data sets should prioritize one or the other depending on the specific SBMs targeted for data assimilation and their research objectives. For model comparison of naive stochastic SBM parameterizations where the conceived diffusion θ are less biologically meaningful, accumulating approximation error is less of a concern and prioritizing the maximization of T would be reasonable. In scenarios involving SBMs parameterized with more biogeochemically sophisticated β matrices where accurate estimation of system diffusion θ takes precedence and falsification of specific dynamics is the goal, it would be more important to minimize approximation error with denser observations.

4.2 Future work and research directions

Having demonstrated functional flow VI on more compact synthetic data sets, we highlight some engineering expansions and modifications to our existing framework that would facilitate efficient SBM inferences conditioned on empirical data sourced from long-term ecological (LTER) experiments like those documented in Melillo et al. (2017) and Wood et al. (2019). Efficiently scaling to these data sets is a key priority to assimilate them into SBMs on the scale of hours rather than weeks, as experienced in Xie et al. (2020), for statistically rigorous head-to-head model comparison and selection.

The T of data sets sourced from LTER experiments can be on the order of 100,000 to 200,000 hours, much larger than the peak $T = 5000$ hour timespan we explored in our study. With the ability of our framework to leverage GPU hardware, our $T = 5000$ inferences typically ran between one to two days to ensure convergence, but even more limiting than time were GPU memory thresholds preventing adequate variational sample sizes with T much longer than 5000. A way forward for conditioning inferences on y with longer T is to avoid simulating state space model x for the entire T at each training iteration, and this can be done in stochastic gradient optimization with the leveraging of the *mini-batching* technique. Under a mini-batching scheme, y is partitioned into smaller sub-sequences y_i during training, where $i \in 1 : \mathcal{B}$ and $\mathcal{B}$ is the total number of partitions. In each training iteration, a y_i can then be randomly selected for likelihood evaluation such that the SBM only needs to simulate an x_i subsequence for calculation of the optimization objective. Mini-batching is targeted for future incorporation in our framework, having been demonstrated in recent flow-related machine learning literature including Papamakarios et al. (2021) and Ryder et al. (2021).

LTER data sets tend to have constituent observation vectors whose elements greatly vary in information density and measurement intervals due in part to the varying phys-

ical practicality associated with the sampling and measurement of different observations. Hence, it would also be helpful to engineer our flow to handle irregular and "ragged" observations. Moreover, alterations can be made to the flow network architecture to enable more efficient conditioning on SBM θ values and allow for feature extraction from additional auxiliary information, such as the time elapsed between observations.

Beyond flow engineering and architecture, other relevant research priorities include the study of less naive SCON treatments for inference use. SCON family representatives that explicitly and mechanistically model system diffusion as a function of the underlying system reaction stoichiometry can be formulated (Golightly & Wilkinson, 2011; Fuchs, 2013) and the stability and predictive accuracy associated with different diffusion covariance structures can be compared. Moreover, stochastic parameterizations of SBMs that simulate mass transfer with Michaelis-Menten dynamics and would be non-linear in drift should be investigated so that their predictive accuracy can be compared to those of linear drift models under our VI framework. This would go toward an existing priority in biogeochemistry to examine whether explicit representation of enzyme catalysis in SBMs improves model performance (J. Li et al., 2014; Sulman et al., 2014; Wieder et al., 2015; J. Li et al., 2019; Xie et al., 2020).

Application of our VI approach to head-to-head model comparison and selection begets a need for incorporation of goodness-of-fit quantification into our framework. MCMC has access to metrics like the widely application information criterion (Vehtari et al., 2017), leave-one-out cross-validation, and leave-future-out cross-validation (Bürkner et al., 2020) for Bayesian predictive accuracy quantification, but with their established formulations, these metrics cannot be computed under a VI procedure without prohibitive computational expense (Dao et al., 2022). The development of Bayesian goodness-of-fit metrics for VI is still an open area (Yao et al., 2018; Giordano et al., 2018), but there has been recent work adapting cross-validation for VI that is promising for integration with a state space model inference pipeline (Magnusson et al., 2019; Dao et al., 2022).

4.3 Conclusion

Going forward, we recommend that inference approaches involving state space model approximation of stochastic SBMs be used in future biogeochemical data assimilation, fitting, and model comparison research in pursuit of superior computational stability, flexibility, and efficiency. SDE systems are far more robust than ODE systems at accommodating prior density, initial condition, and model structure proposals that are inconsistent with the true data generating process (Whitaker, 2016; Wiqvist et al., 2021). Then, state space approximation greatly reduces the burden of sampling SDE model state trajectories for likelihood evaluation. Rather than integrating an SDE solver $\mathcal{S}$ times at great computational cost with each algorithm training iteration, we can efficiently sample $\mathcal{S}$ paths from the variational approximation in one pass. Additionally, the discrete nature of state space models integrates well with likelihood estimation conditioned on sparsely observed data sets from long term ecological research where fine-grained knowledge of continuous state dynamics of a model are not necessary or useful for the inference algorithm. State space model discretization can be handled much more coarsely, which facilitates more efficient scaling to larger T.

Many of the steps of our data assimilation framework are common to those of other Bayesian inference approaches and hence, a wealth of options exist for modification of this approximated SBM inference framework depending on computational resources and desired posterior estimation accuracy. Different non-variational black box inference methods that are compatible with state space approximation of SDEs can be substituted, such as sequential Monte Carlo algorithms (Golightly & Kypraios, 2018), stochastic gradient Hamiltonian Monte Carlo (Chen et al., 2014), stochastic gradient langevin dynamics (Brosse

et al., 2018), and stochastic gradient Markov chain Monte Carlo (Aicher et al., 2019; Nemeth & Fearnhead, 2021).

Resesarchers using variational Bayesian methods for their black box can opt for $q(\theta)$ variational approximations that are more complex than mean-field representation. These include full-rank multivariate logit-normal families in which θ are not assumed to be independent and covariance is established. The full-rank modeling of covariance mitigates underestimation of $q(\theta)$ uncertainty, which is prevalent in mean-field inference, to correspond to wider marginal $q(\theta)$ densities (Kucukelbir et al., 2017; Sujono et al., 2022).

Additionally, when memory availability inhibits the establishment of larger neural networks to train affine shift and scale values for long T or when another method is known to be faster and more convenient for approximating a particular SBM class, different variational families can be used in place of neural moving average flows for representation and optimization of $q(x|\theta)$. These include multivariate normal distributions with specialized covariance structures (Archer et al., 2015), automatic differention VI (Kucukelbir et al., 2017) with Gauss-Markov distributions (Sujono et al., 2022), neural stochastic differential equations (Tzen & Raginsky, 2019; Jia & Benson, 2019; X. Li et al., 2020), and recurrent neural networks (Krishnan et al., 2017; Ryder et al., 2018), among others. Thus, our framework is flexible and can be repurposed as needed for assimilation of different SBMs or Earth system models that vary in complexity and simulation requirements.

5 Open Research

A repository containing synthetic time series data of soil pool state and CO_2 observations, Python notebooks containing the code for SCON-C and SCON-SS data-generating processes, and Python modules scaffolding the neural moving average flow VI algorithm are available at `https://doi.org/10.5281/zenodo.6969782`. Stan code for the deterministic CON inference whose results were compared to in Figure 5 is available at `https://doi.org/10.5281/zenodo.6969769`.

Acknowledgments

We would like to thank Anton Obukhov (ETH Zurich) for his PyTorch truncated normal distribution class that was used in exploratory work that this study was built on and Andrew Golightly (Durham University) for his advice on ODE-to-SDE conversion and reparameterization.

This research was financially supported by the U.S. National Science Foundation (grant no. DEB-1900885 and IIS-1816365), the U.S. Department of Energy (grant no. DE-SC0014374 and DE-SC0020382), and a UC Irvine ICS Exploration Research Award.

The authors affirm that they have no financial conflict of interest.

References

Abs, E., Leman, H., & Ferrière, R. (2020). A multi-scale eco-evolutionary model of cooperation reveals how microbial adaptation influences soil decomposition. *Communications Biology*, *3*(1). Retrieved from https://doi.org/10.1038/s42003-020-01198-4 doi: 10.1038/s42003-020-01198-4

Aicher, C., Ma, Y.-A., Foti, N. J., & Fox, E. B. (2019). Stochastic Gradient MCMC for State Space Models. *SIAM Journal on Mathematics of Data Science*, *1*(3). Retrieved from https://doi.org/10.1137/18M1214780 doi: 10.1137/18M1214780

Allison, S. D., Wallenstein, M. D., & Bradford, M. A. (2010). Soil-carbon response to warming dependent on microbial physiology. *Nature Geoscience*, *3*(5). Retrieved from https://doi.org/10.1038/ngeo846 doi: 10.1038/ngeo846

Anaconda Software Distribution. (2020). Anaconda Inc. Retrieved 2022-06-17, from https://docs.anaconda.com/

Archer, E., Park, I. M., Buesing, L., Cunningham, J., & Paninski, L. (2015). *Black box variational inference for state space models.* arXiv. Retrieved from https://arxiv.org/abs/1511.07367 doi: 10.48550/ARXIV.1511.07367

Arrhenius, S. (1889). Über die Dissociationswärme und den Einfluss der Temperatur auf den Dissociationsgrad der Elektrolyte. *Zeitschrift für Physikalische Chemie*, *4U*(1). Retrieved from https://doi.org/10.1515/zpch-1889-0408 doi: 10.1515/zpch-1889-0408

Betancourt, M. (2017). *A Conceptual Introduction to Hamiltonian Monte Carlo.* arXiv. Retrieved from http://arxiv.org/abs/1701.02434

Bjorck, N., Gomes, C. P., Selman, B., & Weinberger, K. Q. (2018). Understanding Batch Normalization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 31). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2018/file/36072923bfc3cf47745d704feb489480-Paper.pdf

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, *112*(518). doi: 10.1080/01621459.2017.1285773

Bradford, M. A., Wood, S. A., Addicott, E. T., Fenichel, E. P., Fields, N., González-Rivero, J., ... Wieder, W. R. (2021). Quantifying microbial control of soil organic matter dynamics at macrosystem scales. *Biogeochemistry*, *156*(1). Retrieved from https://doi.org/10.1007/s10533-021-00789-5 doi: 10.1007/s10533-021-00789-5

Brosse, N., Durmus, A., & Moulines, E. (2018). The promises and pitfalls of Stochastic Gradient Langevin Dynamics. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 31). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2018/file/335cd1b90bfa4ee70b39d08a4ae0cf2d-Paper.pdf

Browning, A. P., Warne, D. J., Burrage, K., Baker, R. E., & Simpson, M. J. (2020). Identifiability analysis for stochastic differential equation models in systems biology. *Journal of The Royal Society Interface*, *17*(173). Retrieved from https://royalsocietypublishing.org/doi/abs/10.1098/rsif.2020.0652 doi: 10.1098/rsif.2020.0652

Bürkner, P.-C., Gabry, J., & Vehtari, A. (2020). Approximate leave-future-out cross-validation for Bayesian time series models. *Journal of Statistical Computation and Simulation*, *90*(14). Retrieved from https://doi.org/10.1080/00949655.2020.1783262 doi: 10.1080/00949655.2020.1783262

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., ... Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, *76*(1). doi: 10.18637/jss.v076.i01

Chen, T., Fox, E., & Guestrin, C. (2014). Stochastic Gradient Hamiltonian Monte Carlo. In E. P. Xing & T. Jebara (Eds.), *Proceedings of the 31st International Conference on Machine Learning* (Vol. 32). Beijing, China: PMLR. Retrieved from https://proceedings.mlr.press/v32/cheni14.html

Christensen, R., Johnson, W., Branscum, A., & Hanson, T. E. (2010). *Bayesian Ideas and Data Analysis: An Introduction for Scientists and Statisticians* (1st ed.). Taylor & Francis.

Dao, V. H., Gunawan, D., Tran, M.-N., Kohn, R., Hawkins, G. E., & Brown, S. D. (2022). Efficient selection between hierarchical cognitive models: Cross-validation with variational Bayes. *Psychological Methods*. doi: 10.1037/met0000458

Daunizeau, J. (2017). *Semi-analytical approximations to statistical moments of sigmoid and softmax mappings of normal variables.* arXiv. Retrieved from https://arxiv.org/abs/1703.00091 doi: 10.48550/ARXIV.1703.00091

Dinh, L., Krueger, D., & Bengio, Y. (2015). NICE: Non-linear Independent Components Estimation. In Y. Bengio & Y. LeCun (Eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings.* Retrieved from http://arxiv.org/abs/1410.8516

Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2017). Density estimation using Real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.* OpenReview.net. Retrieved from https://openreview.net/forum?id=HkpbnH9lx

Fuchs, C. (2013). *Inference for Diffusion Processes: With Applications in Life Sciences.* Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-642-25969-2 doi: 10.1007/978-3-642-25969-2

Gabry, J., & Češnovar, R. (2021). *CmdStanR.* Retrieved 2022-05-18, from https://mc-stan.org/cmdstanr/

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., & Rubin, D. (2013). *Bayesian Data Analysis* (3rd ed.). New York: Taylor & Francis. Retrieved from https://doi.org/10.1201/b16018

Geman, S., & Geman, D. (1987). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. In M. A. Fischler & O. Firschein (Eds.), *Readings in Computer Vision.* San Francisco (CA): Morgan Kaufmann. Retrieved from https://www.sciencedirect.com/science/article/pii/B978008051581650057X doi: 10.1016/B978-0-08-051581-6.50057-X

Georgiou, K., Malhotra, A., Wieder, W. R., Ennis, J. H., Hartman, M. D., Sulman, B. N., ... Jackson, R. B. (2021). Divergent controls of soil organic carbon between observations and process-based models. *Biogeochemistry*. Retrieved from https://doi.org/10.1007/s10533-021-00819-2 doi: 10.1007/s10533-021-00819-2

Giordano, R., Broderick, T., & Jordan, M. I. (2018). Covariances, Robustness, and Variational Bayes. *Journal of Machine Learning Research*, *19*(51). Retrieved from http://jmlr.org/papers/v19/17-670.html

Giweta, M. (2020). Role of litter production and its decomposition, and factors affecting the processes in a tropical forest ecosystem: a review. *Journal of Ecology and Environment*, *44*(1). Retrieved from https://doi.org/10.1186/s41610-020-0151-2 doi: 10.1186/s41610-020-0151-2

Golightly, A., & Kypraios, T. (2018). Efficient SMC^2 schemes for stochastic kinetic models. *Statistics and Computing*, *28*(6). Retrieved from https://doi.org/10.1007/s11222-017-9789-8 doi: 10.1007/s11222-017-9789-8

Golightly, A., & Wilkinson, D. (2010). Markov chain Monte Carlo algorithms for SDE parameter estimation. In N. D. Lawrence, M. Girolami, M. Rattray, & G. Sanguinetti (Eds.), . Cambridge, MA, USA: MIT Press.

Golightly, A., & Wilkinson, D. J. (2006). Bayesian sequential inference for nonlinear multivariate diffusions. *Statistics and Computing*, *16*(4). Retrieved from https://doi.org/10.1007/s11222-006-9392-x doi: 10.1007/s11222-006-9392-x

Golightly, A., & Wilkinson, D. J. (2011). Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. *Interface Focus*, *1*(6). Retrieved from https://pubmed.ncbi.nlm.nih.gov/23226583 doi: 10.1098/rsfs.2011.0047

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., ... He, K. (2017). *Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour.* arXiv. Retrieved from https://arxiv.org/abs/1706.02677 doi: 10.48550/ARXIV.1706.02677

Hararuk, O., & Luo, Y. (2014). Improvement of global litter turnover rate predictions using a Bayesian MCMC approach. *Ecosphere*, *5*(12). Retrieved from https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1890/ES14-00092.1 doi: 10.1890/ES14-00092.1

Hararuk, O., Xia, J., & Luo, Y. (2014). 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*, *119*(3). Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2013JG002535 doi: 10.1002/2013JG002535

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825). Retrieved from https://doi.org/10.1038/s41586-020-2649-2 doi: 10.1038/s41586-020-2649-2

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* doi: 10.1109/CVPR.2016.90

Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, *15*(47). Retrieved from http://jmlr.org/papers/v15/hoffman14a.html

Ioffe, S. (2017). Batch Renormalization: Towards Reducing Minibatch Dependence in Batch-Normalized Models. In *Proceedings of the 31st International Conference on Neural Information Processing Systems.* Red Hook, NY, USA: Curran Associates Inc.

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37.* JMLR.org.

Jia, J., & Benson, A. R. (2019). Neural Jump Stochastic Differential Equations. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems.* Red Hook, NY, USA: Curran Associates Inc.

Joyce, J. M. (2011). Kullback-Leibler Divergence. In M. Lovric (Ed.), *International Encyclopedia of Statistical Science.* Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-642-04898-2_327 doi: 10.1007/978-3-642-04898-2_327

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, *82*(1). Retrieved from https://doi.org/10.1115/1.3662552 doi: 10.1115/1.3662552

Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. In Y. Bengio & Y. LeCun (Eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.* Retrieved from http://arxiv.org/abs/1412.6980

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., & Welling, M. (2016). Improved Variational Inference with Inverse Autoregressive Flow. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc.

Kingma, D. P., & Welling, M. (2014). Auto-Encoding Variational Bayes. In Y. Bengio & Y. LeCun (Eds.), *2nd international conference on learning representations, ICLR 2014, banff, ab, canada, april 14-16, 2014, conference track proceedings*. Retrieved from http://arxiv.org/abs/1312.6114

Kobyzev, I., Prince, S., & Brubaker, M. (2020). Normalizing Flows: An Introduction and Review of Current Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Retrieved from http://dx.doi.org/10.1109/TPAMI.2020.2992934 doi: 10.1109/tpami.2020.2992934

Krishnan, R. G., Shalit, U., & Sontag, D. (2017). Structured Inference Networks for Nonlinear State Space Models. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI Press.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., & Blei, D. M. (2017). Automatic Differentiation Variational Inference. *Journal of Machine Learning Research*, *18*(1), 430-474.

Kullback, S., & Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, *22*(1). Retrieved from https://doi.org/10.1214/aoms/1177729694 doi: 10.1214/aoms/1177729694

Lee, H., Pabbaraju, C., Sevekari, A., & Risteski, A. (2021). *Universal Approximation for Log-concave Distributions using Well-conditioned Normalizing Flows*. arXiv. Retrieved from https://arxiv.org/abs/2107.02951 doi: 10.48550/ARXIV.2107.02951

Li, J., Wang, G., Allison, S. D., Mayes, M. A., & Luo, Y. (2014). Soil carbon sensitivity to temperature and carbon use efficiency compared across microbial-ecosystem models of varying complexity. *Biogeochemistry*, *119*(1-3). doi: 10.1007/s10533-013-9948-8

Li, J., Wang, G., Mayes, M. A., Allison, S. D., Frey, S. D., Shi, Z., ... Melillo, J. M. (2019). Reduced carbon use efficiency and increased microbial turnover with soil warming. *Global Change Biology*, *25*(3). Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/gcb.14517 doi: 10.1111/gcb.14517

Li, X., Wong, T.-K. L., Chen, R. T. Q., & Duvenaud, D. (2020). Scalable Gradients for Stochastic Differential Equations. In S. Chiappa & R. Calandra (Eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics* (Vol. 108). PMLR. Retrieved from https://proceedings.mlr.press/v108/li20i.html

Luo, Y., Ahlström, A., Allison, S. D., Batjes, N. H., Brovkin, V., Carvalhais, N., ... Zhou, T. (2016). Toward more realistic projections of soil carbon dynamics by Earth system models. *Global Biogeochemical Cycles*, *30*(1). Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2015GB005239 doi: 10.1002/2015GB005239

Magnusson, M., Andersen, M., Jonasson, J., & Vehtari, A. (2019). Bayesian leave-one-out cross-validation for large data. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th International Conference on Machine Learning* (Vol. 97). PMLR. Retrieved from https://proceedings.mlr.press/v97/magnusson19a.html

Manzoni, S., & Porporato, A. (2009). Soil carbon and nitrogen mineralization: Theory and models across scales. *Soil Biology and Biochemistry*, *41*(7). Retrieved from http://dx.doi.org/10.1016/j.soilbio.2009.02.031 doi: 10.1016/j.soilbio.2009.02.031

Maruyama, G. (1955). Continuous Markov processes and stochastic equations. *Rendiconti del Circolo Matematico di Palermo*, *4*(1). Retrieved from https://doi

.org/10.1007/BF02846028 doi: 10.1007/BF02846028

McElreath, R. (2020). *Statistical Rethinking: A Bayesian Course with Examples in R and Stan, 2nd Edition* (2nd ed.). CRC Press. Retrieved from http://xcelab.net/rm/statistical-rethinking/

Melillo, J. M., Frey, S. D., DeAngelis, K. M., Werner, W. J., Bernard, M. J., Bowles, F. P., ... Grandy, A. S. (2017). Long-term pattern and magnitude of soil carbon feedback to the climate system in a warming world. *Science*, *358*(6359). doi: 10.1126/science.aan2874

Nemeth, C., & Fearnhead, P. (2021). Stochastic Gradient Markov Chain Monte Carlo. *Journal of the American Statistical Association*, *116*(533). Retrieved from https://doi.org/10.1080/01621459.2020.1847120 doi: 10.1080/01621459.2020.1847120

O'Neill, B. C., Kriegler, E., Ebi, K. L., Kemp-Benedict, E., Riahi, K., Rothman, D. S., ... Solecki, W. (2017). The roads ahead: Narratives for shared socioeconomic pathways describing world futures in the 21st century. *Global Environmental Change*, *42*. Retrieved from https://www.sciencedirect.com/science/article/pii/S0959378015000060 doi: 10.1016/j.gloenvcha.2015.01.004

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing Flows for Probabilistic Modeling and Inference. *Journal of Machine Learning Research*, *22*(57). Retrieved from http://jmlr.org/papers/v22/19-1028.html

Papamakarios, G., Pavlakou, T., & Murray, I. (2017). Masked Autoregressive Flow for Density Estimation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems.* Red Hook, NY, USA: Curran Associates Inc.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32.* Curran Associates, Inc. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Perez-Cruz, F. (2008). Kullback-Leibler divergence estimation of continuous distributions. In *2008 IEEE International Symposium on Information Theory.* doi: 10.1109/ISIT.2008.4595271

Plummer, M. (2003). JAGS: A Program for Analysis of Bayesian Graphical Models using Gibbs Sampling. In K. Hornik, F. Leisch, & A. Zeileis (Eds.), *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)* (Vol. 3). Vienna, Austria.

Raczka, B., Hoar, T. J., Duarte, H. F., Fox, A. M., Anderson, J. L., Bowling, D. R., & Lin, J. C. (2021). Improving CLM5.0 Biomass and Carbon Exchange Across the Western United States Using a Data Assimilation System. *Journal of Advances in Modeling Earth Systems*, *13*(7). Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020MS002421 (e2020MS002421 2020MS002421) doi: 10.1029/2020MS002421

Rauch, H. E., Tung, F., & Striebel, C. T. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, *3*(8).

Reid, N. M. (2015). Approximate likelihoods. In L. Guo & Z.-M. Ma (Eds.), *Proceedings of the 8th International Congress on Industrial and Applied Mathematics.* Beijing, China: Higher Education Press.

Ruiz, F. R., Titsias RC AUEB, M., & Blei, D. (2016). The Generalized Reparameterization Gradient. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 29). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/

paper/2016/file/f718499c1c8cef6730f9fd03c8125cab-Paper.pdf

Ryder, T., Golightly, A., McGough, A. S., & Prangle, D. (2018). Black-Box Variational Inference for Stochastic Differential Equations. In J. Dy & A. Krause (Eds.), *Proceedings of the 35th International Conference on Machine Learning* (Vol. 80). PMLR. Retrieved from https://proceedings.mlr.press/v80/ryder18a.html

Ryder, T., Prangle, D., Golightly, A., & Matthews, I. (2021). The neural moving average model for scalable variational inference of state space models. In C. de Campos & M. H. Maathuis (Eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence* (Vol. 161, pp. 12–22). PMLR. Retrieved from https://proceedings.mlr.press/v161/ryder21a.html

Saifuddin, M., Abramoff, R. Z., Davidson, E. A., Dietze, M. C., & Finzi, A. C. (2021). Identifying Data Needed to Reduce Parameter Uncertainty in a Coupled Microbial Soil C and N Decomposition Model. *Journal of Geophysical Research: Biogeosciences*, *126*(12). Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2021JG006593 doi: 10.1029/2021JG006593

Salimans, T., Kingma, D., & Welling, M. (2015). Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd International Conference on Machine Learning* (Vol. 37). Lille, France: PMLR. Retrieved from https://proceedings.mlr.press/v37/salimans15.html

Salimans, T., & Knowles, D. A. (2013). Fixed-Form Variational Posterior Approximation through Stochastic Linear Regression. *Bayesian Analysis*, *8*(4). Retrieved from https://doi.org/10.1214/13-BA858 doi: 10.1214/13-BA858

Salvatier, J., Wiecki, T. V., & Fonnesbeck, C. (2016). Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, *2*. Retrieved from https://doi.org/10.7717/peerj-cs.55 doi: 10.7717/peerj-cs.55

Särkkä, S. (2013). *Bayesian Filtering and Smoothing*. Cambridge: Cambridge University Press. Retrieved from https://www.cambridge.org/core/books/bayesian-filtering-and-smoothing/C372FB31C5D9A100F8476C1B23721A67 doi: 10.1017/CBO9781139344203

Särkkä, S., & Solin, A. (2019). *Applied Stochastic Differential Equations*. Cambridge University Press. doi: 10.1017/9781108186735

Sujono, D., Xie, H. W., Allison, S., & Sudderth, E. B. (2022). Variational Inference for Soil Biogeochemical Models. In *ICML 2022 2nd AI for Science Workshop*. Retrieved from https://openreview.net/forum?id=3_HrwqrPd4U

Sulman, B. N., Moore, J. A., Abramoff, R., Averill, C., Kivlin, S., Georgiou, K., ... Classen, A. T. (2018). Multiple models and experiments underscore large uncertainty in soil carbon dynamics. *Biogeochemistry*, *141*(2). doi: 10.1007/s10533-018-0509-z

Sulman, B. N., Phillips, R. P., Oishi, A. C., Shevliakova, E., & Pacala, S. W. (2014). Microbe-driven turnover offsets mineral-mediated storage of soil carbon under elevated CO_2. *Nature Climate Change*, *4*(12). doi: 10.1038/nclimate2436

Summers, C., & Dinneen, M. J. (2020). Four Things Everyone Should Know to Improve Batch Normalization. In *International Conference on Learning Representations*. Retrieved from https://openreview.net/forum?id=HJx8HANFDH

Todd-Brown, K. E. O., Randerson, J. T., Hopkins, F., Arora, V., Hajima, T., Jones, C., ... Allison, S. D. (2014). Changes in soil organic carbon storage predicted by earth system models during the 21st century. *Biogeosciences*, *11*(8), 2341-2356. Retrieved from https://bg.copernicus.org/articles/11/2341/2014/ doi: 10.5194/bg-11-2341-2014

Todd-Brown, K. E. O., Randerson, J. T., Post, W. M., Hoffman, F. M., Tarnocai, C., Schuur, E. A. G., & Allison, S. D. (2013). Causes of variation in soil carbon simulations from CMIP5 Earth system models and comparison with obser-

vations. *Biogeosciences*, *10*(3). Retrieved from https://bg.copernicus.org/articles/10/1717/2013/ doi: 10.5194/bg-10-1717-2013

Tzen, B., & Raginsky, M. (2019). *Neural Stochastic Differential Equations: Deep Latent Gaussian Models in the Diffusion Limit.* arXiv. Retrieved from https://arxiv.org/abs/1905.09883 doi: 10.48550/ARXIV.1905.09883

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, *27*(5). doi: 10.1007/s11222-016-9696-4

Wang, S., Luo, Y., & Niu, S. (2022). Reparameterization Required After Model Structure Changes From Carbon Only to Carbon-Nitrogen Coupling. *Journal of Advances in Modeling Earth Systems*, *14*(4). Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2021MS002798 doi: 10.1029/2021MS002798

Whitaker, G. A. (2016). *Bayesian inference for stochastic differential mixed-effects models* (Unpublished doctoral dissertation). Newcastle University.

Whitaker, G. A., Golightly, A., Boys, R. J., & Sherlock, C. (2017). Bayesian Inference for Diffusion-Driven Mixed-Effects Models. *Bayesian Analysis*, *12*(2). Retrieved from https://doi.org/10.1214/16-BA1009 doi: 10.1214/16-BA1009

Wieder, W. R., Boehnert, J., & Bonan, G. B. (2014). Evaluating soil biogeochemistry parameterizations in Earth system models with observations. *Global Biogeochemical Cycles*, *28*(3). Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2013GB004665 doi: 10.1002/2013GB004665

Wieder, W. R., Grandy, A. S., Kallenbach, C. M., Taylor, P. G., & Bonan, G. B. (2015). Representing life in the Earth system with soil microbial functional traits in the MIMICS model. *Geoscientific Model Development*, *8*(6). doi: 10.5194/gmd-8-1789-2015

Wiqvist, S., Golightly, A., McLean, A. T., & Picchini, U. (2021). Efficient inference for stochastic differential equation mixed-effects models using correlated particle pseudo-marginal algorithms. *Computational Statistics & Data Analysis*, *157*. Retrieved from https://www.sciencedirect.com/science/article/pii/S0167947320302425 doi: 10.1016/j.csda.2020.107151

Wood, T. E., González, G., Silver, W. L., Reed, S. C., & Cavaleri, M. A. (2019). On the shoulders of giants: Continuing the legacy of large-scale ecosystem manipulation experiments in Puerto Rico. *Forests*, *10*(3). doi: 10.3390/f10030210

Xie, H. W., Romero-Olivares, A. L., Guindani, M., & Allison, S. D. (2020). A Bayesian approach to evaluation of soil biogeochemical models. *Biogeosciences*, *17*(15). Retrieved from https://bg.copernicus.org/articles/17/4043/2020/ doi: 10.5194/bg-17-4043-2020

Yao, Y., Vehtari, A., Simpson, D., & Gelman, A. (2018). Yes, but Did It Work?: Evaluating Variational Inference. In J. Dy & A. Krause (Eds.), *Proceedings of the 35th International Conference on Machine Learning* (Vol. 80). PMLR. Retrieved from https://proceedings.mlr.press/v80/yao18a.html

You, K., Long, M., Wang, J., & Jordan, M. I. (2019). *How Does Learning Rate Decay Help Modern Neural Networks?* arXiv. Retrieved from https://arxiv.org/abs/1908.01878 doi: 10.48550/ARXIV.1908.01878

Zhu, Q., Bi, W., Liu, X., Ma, X., Li, X., & Wu, D. (2020). A Batch Normalized Inference Network Keeps the KL Vanishing Away. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.acl-main.235 doi: 10.18653/v1/2020.acl-main.235