Water Resources Research

RESEARCH ARTICLE
10.1029/2018WR024461

Special Section:
Big Data & Machine Learning in Water Sciences: Recent Progress and Their Use in Advancing Science

Gradient-Based Inverse Estimation for a Rainfall-Runoff Model

Christopher Krapu[1,2], **Mark Borsuk**[1], **and Mukesh Kumar**[2,3]

[1]Department of Civil and Environmental Engineering, Duke University, Durham, NC, USA, [2]Department of Civil, Construction, and Environmental Engineering, University of Alabama, Tuscaloosa, AL, USA, [3]Nicholas School of the Environment, Duke University, Durham, NC, USA

Key Points:
- Gradient-based methods of parameter estimation are used by the deep learning community for high-dimensional parameter estimation
- We demonstrate gradient-based optimization via automatic differentiation using the rainfall-runoff model GR4J implemented in Theano
- We show that two gradient-based methods outperform nongradient methods and are scalable to high-dimensional parameter spaces

Correspondence to:
C. Krapu,
christopher.krapu@duke.edu

Citation:
Krapu, C., Borsuk, M., & Kumar, M. (2019). Gradient-based inverse estimation for a rainfall-runoff model. *Water Resources Research*, *55*. https://doi.org/10.1029/2018WR024461

Received 19 NOV 2018
Accepted 12 JUL 2019
Accepted article online 24 JUL 2019

Abstract Recent advances in deep learning for neural networks with large numbers of parameters have been enabled by automatic differentiation, an algorithmic technique for calculating gradients of measures of model fit with respect to model parameters. Estimation of high-dimensional parameter sets is an important problem within the hydrological sciences. Here, we demonstrate the effectiveness of gradient-based estimation techniques for high-dimensional inverse estimation problems using a conceptual rainfall-runoff model. In particular, we compare the effectiveness of Hamiltonian Monte Carlo and automatic differentiation variational inference against two nongradient-dependent methods, random walk Metropolis and differential evolution Metropolis. We show that the former two techniques exhibit superior performance for inverse estimation of daily rainfall values and are much more computationally efficient on larger data sets in an experiment with synthetic data. We also present a case study evaluating the effectiveness of automatic differentiation variational inference for inverse estimation over 25 years of daily precipitation conditional on streamflow observations at three catchments and show that it is scalable to very high dimensional parameter spaces. The presented results highlight the power of combining hydrological process-based models with optimization techniques from deep learning for high-dimensional estimation problems.

Plain Language Summary We programmed a rainfall-runoff model in a software package designed for optimizing neural networks and found that this enabled application of these tools for estimating unknown parameters of our model. Using simulated data, we compared the effectiveness of two methods employing this technique with two which did not and found that the former were much more effective at estimating large numbers of unknown variables. A case study involving 25 years of data from three catchments was also performed in order to assess the viability of this approach on real-world data.

1. Introduction

Substantial attention has been devoted in recent years to the translation of empirical, data-driven methods arising outside of the Earth and environmental sciences to hydrology (Marçais & de Dreuzy, 2017). These methods require little knowledge about the system being studied and tend to make few assumptions about the underlying dynamics of the data-generating process, but they generally have a large number of uninterpretable parameters. This necessitates assimilation of large observational data sets to adequately constrain the parameter space and provide useful predictive forecasts.

In most modern applications of neural networks, the gradient $\nabla_\theta \boldsymbol{f}(\boldsymbol{x}, \theta)$ of the objective function $\boldsymbol{f}$ with regard to model parameters θ conditional on the observed data $\boldsymbol{x}$ is used to update these model parameters to minimize predictive error. This approach is generally known as *gradient descent*. When the data set is too large to compute the derivative within a reasonable amount of time, small subsets of the data are used at each training step; this approach is labeled *stochastic gradient descent* (Robbins & Monro, 1951). This method, in principle, can be employed whenever the objective function is a differentiable function of the model parameters. Deep neural networks benefit immensely from this approach, as they are expressed as compositions and recurrences of continuous functions for which the gradient is usually well defined, even if it contains a large number of terms. Earlier work in neural networks assumed network forms for which the stochastic gradient descent updates could be defined in a simple, closed-form fashion (Rumelhart et al., 1986). However, as the machine learning research community incorporated more sophisticated

©2019. American Geophysical Union. All Rights Reserved.

activation functions, network structures, and problem formulations, obtaining expressions for the gradient function by hand quickly became infeasible.

A crucial next step in the training of deep models was the usage of automatic differentiation (AD), an algorithmic technique for calculating derivatives exactly, to provide update equations for $\theta \rightarrow \theta^*$ without human intervention. AD makes use of the fact that continuous mathematical functions can be represented as compositions of simpler functions for which the derivative is known. Then, the chain rule is applied iteratively to produce a formula for the gradient that might be too cumbersome to program by hand or write down explicitly. It borrows elements of both symbolic and numerical differentiation, and while an in-depth review and tutorial of AD is beyond the scope of this work, we refer the reader to existing tutorials and reviews on this subject (Baydin et al., 2018). AD is a major component of all modern deep learning frameworks such as TensorFlow (Abadi et al., 2015), Caffe (Jia et al., 2014), Torch (Paszke et al., 2017), and Theano (Bergstra et al., 2010). Within hydrology, applications of neural networks to hydrologic modeling (Dawson & Wilby, 2001; Govindaraju, 2000; Tokar & Johnson, 1999) frequently employ gradient-based training methods to estimate network weights or parameters, and more recent studies have begun to use AD-enabled deep learning frameworks for training neural networks on hydrological data (Zhang et al., 2018). While, in principle, all methods employing AD could be implemented instead with numerical approximations to the derivative, the former is generally not as susceptible to truncation and round-off errors as the latter (Iri et al., 1988).

It is our intent in this study to show that gradient information critical to neural network optimization can also enhance parameter estimation in hydrologic modeling. While there are many families of estimation algorithms employed in the hydrological sciences and other disciplines, the interrelations and similarities between these algorithms can be unclear. A taxonomy of estimation methods is provided in Figure 1, providing a categorization of frequently used estimation algorithms across machine learning and hydrology. Within machine learning, the adjective "deep" often refers to large numbers of variables with feed-forward connections, leading to highly nested composite functional forms and varying levels of abstraction. In our application, "depth" is achieved by employing a dynamical system that induces cross-time step dependencies and similarly requires back-propagation of gradients through upward of 10^4 nested function evaluations. However, we only consider a generative process with a limited number of structural parameters by using a conceptual rainfall-runoff model as opposed to a deep neural network with thousands or even millions of parameters. We are able to calculate the gradients required because the chosen rainfall-runoff model, GR4J, is mostly continuous with only a few point discontinuities. As inverse problems in the Earth and environmental sciences in general and hydrology in particular are frequently ill-posed with more unknown quantities of interest than observed data, we make use of the fact that the hydrological model structure imposed by GR4J can be considered to be a very strong form of prior. This restricts the possible pairings of input rainfall volumes and output streamflow values to a space that is sufficiently small to allow for inverse estimation.

In this work, we address the problem of inverse modeling of precipitation conditional on streamflow observations to illustrate the advantages of using gradient-based estimation methods. Inverse modeling typically requires exploring a high-dimensional solution space in which each system input or initial condition induces an extraparameter. We note that the generality of our approach is not unique to inverse modeling, but we have chosen this application, as it allows us to designate an arbitrarily large number of unknown variables to be estimated and it has been considered in past studies as a difficult test case for estimation algorithms (Vrugt et al., 2008). We apply automatic differentiation variational inference (ADVI; Kucukelbir et al., 2016) and a variant of Hamiltonian Monte Carlo (HMC; Duane et al., 1987), the No-U-Turn-Sampler (Hoffman & Gelman, 2014), two Bayesian gradient-based optimization methods, to the task of recovering unknown parameters in both a synthetic data case study and also apply ADVI to inverse estimation of a multidecadal observational record of streamflow. Our goal is not to specifically emphasize the usefulness of any of the inference algorithms studied but rather to offer an initial investigation into gradient-based estimation methods in general. Within this work, we focus on Bayesian parameter estimation, building upon past work in uncertainty quantification and optimization in hydrology (Kingston et al., 2008; Kuczera & Parent, 1998; Pathiraja et al., 2018; Renard et al., 2010; Smith et al., 2015; Smith & Marshall, 2008) in order to provide a coherent probabilistic framework. The main research questions of this work are as follows:

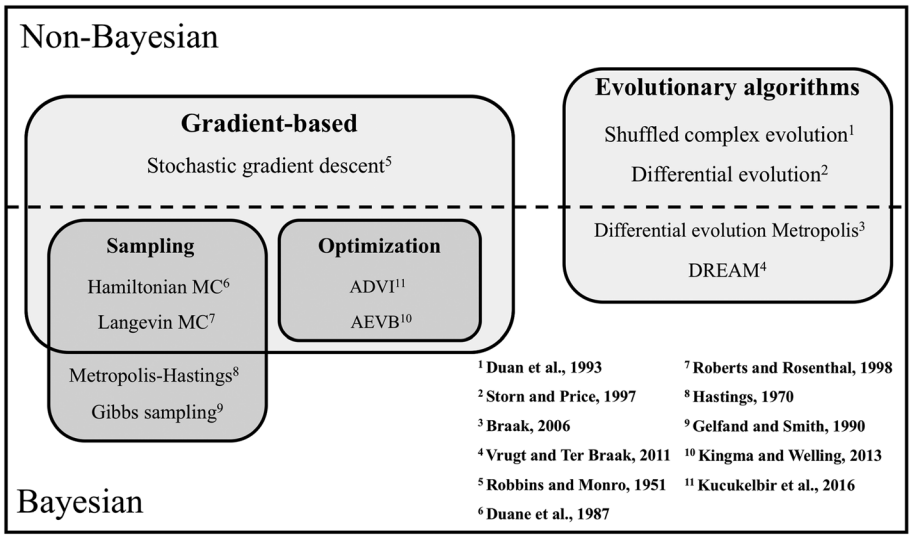

Figure 1. Taxonomy of estimation algorithms. The inference methods listed here may belong to multiple classes, some of which overlap. For example, both Bayesian and non-Bayesian versions of differential evolution exist. Our criterion for an algorithm to be Bayesian is that it assumes a joint probability distribution over model parameters. Methods for estimating parameters in deep neural networks such as stochastic gradient descent are readily adapted to work on Bayesian estimation problems in the form of sampling approaches such as Hamiltonian Monte Carlo or optimization approaches such as automatic differentiation variational inference (ADVI). This is not a comprehensive listing of estimation algorithms employed in hydrology but covers those that are closest to the methods employed in this study.

1. Do deep learning frameworks provide suitable functionality for inverse estimation in conceptual hydrological models?
2. How do gradient-based estimation methods compare with existing inference algorithms applied in hydrology in terms of scalability and accuracy?
3. Are gradient-based methods effective for inverse modeling in scenarios involving large ($>10^3$) parameter sets?

We note that although this work explores the utility of ADVI and HMC/NUTS as applied to inverse rainfall-runoff modeling, we do not single these out as especially well suited to hydrology parameter estimation. Instead, we regard them as exemplars of a very broad class of probabilistic and nonprobabilistic methods that are only feasible when the hydrological model is programmed in an AD enabled framework. Furthermore, we have selected case studies displaying applications in inverse modeling, but the advantages of our modeling strategy may hold in a much broader range of scenarios such as parameter estimation for overparameterized distributed hydrological models. We conducted this study as a proof-of-concept displaying integration of machine learning optimization methods with hydrology-specific model forms. This stands in contrast to applications of machine learning to hydrology, which do not incorporate system-specific processes such as neural networks and random forests. Additionally, we make use of community-standard implementations of ADVI and NUTS from PyMC3 (Salvatier et al., 2016) in order to dramatically simplify the model development and parameter estimation workflow. PyMC3 is one of several statistical programming frameworks that provides a flexible and extensive set of modular building blocks for stochastic model definition and Bayesian parameter estimation. Alternative software platforms that incorporate similar functionality include Stan (Carpenter et al., 2017) and Edward (Tran et al., 2016). We hope that this will provide a roadmap to researchers interested in solving difficult parameter estimation problems in an efficient, reproducible way.

This work is structured as follows: Section 2 describes our methodology involving implementation of conceptual rainfall-runoff model GR4J (Perrin et al., 2003) in Theano, a gradient-aware AD framework. This section also provides an overview of the algorithms that we considered. The data and problem formulations for two case studies are described in section 3 while results and discussion are given in sections 4 and 5, respectively. Relevant computing codes of our implementation of GR4J within Theano are available at GitHub (github.com/ckrapu/gr4j_theano).

2. Methods

To assess the effectiveness of the four algorithms considered, we designed multiple experiments in which we attempted to estimate the precipitation that was used to force a hydrology model, conditional on either simulated streamflow values (section 3.1) or real streamflow data (section 3.2). Section 2.1 describes the hydrological model used, and section 2.2 outlines the prior distribution assumed for the precipitation values.

2.1. Conceptual Rainfall-Runoff Model

We selected GR4J as our hydrological model, as it is a compact, parsimonious representation of runoff generation requiring only daily time series of precipitation and potential evapotranspiration (PET). Several studies have employed GR4J in a Bayesian estimation framework

(Evin et al., 2014; Thyer et al., 2009). Comparable conceptual hydrological models include HBV (Lindström et al., 1997), HyMOD (Boyle et al., 2003), and IHACRES (Jakeman et al., 1990). GR4J has only four parameters and is built around a dual reservoir structure; if daily rainfall P_t exceeds PET E_t, the net rainfall $P_t - E_t$ is added to the storage reservoir, which is parameterized by a capacity parameter x_1 in units of millimeters of water. This storage reservoir may experience deep losses, which are parameterized by x_2 in units of millimeters per day. Evaporative losses from the storage reservoir are determined by E_t. The storage reservoir transfers water to a second reservoir with capacity x_3 (mm), which is routed and converted into a stream hydrograph according to a process controlled by parameter x_4, which is in units of days. Full details of the hydrological model are given in Perrin et al. (2003). In all applications studied within this work, we treat the 4 GR4J parameters as fixed, known quantities. We will refer to these as the structural parameters for GR4J to disambiguate them from the inverse rainfall estimates, which can also be considered parameters of our model in a statistical sense. For the synthetic data case studies (section 3.1), the inverse estimates of precipitation were conditioned on true values used for generating simulated discharge. The forward simulation of streamflow time series was done with a daily time step, and we did not include a more robust numerical integration scheme (Kavetski & Clark, 2010) or a continuous-time representation (Santos et al., 2018) of GR4J. In our real-world case study (section 3.2), we first performed estimation of the GR4J structural parameters and initial conditions with a single year of data of precipitation, PET, and streamflow. This was not a held-out subset of the data, as we needed the initial conditions for the assessment period to be the same as the initial conditions for the training period. We then treat the posterior mean of these estimates as ground truth for an inverse estimation procedure over that year and the following 24 years. We adopted this procedure for the real-world case study in order to disambiguate estimation issues occurring due to GR4J model structure from deficiencies in the parameter estimation algorithms used; as this study is not intended to critically evaluate the performance of GR4J, we did not regard joint estimation of structural parameters $X = \{x_1, x_2, x_3, x_4\}$ and inverse parameters $\{P_1, ..., P_T\}$ as an objective of this work. Consequently, the inverse estimation problem studied in this work can be understood as the estimation of the posterior density $p(P_1, ..., P_T | X, E_1, ..., E_T, Q_1, ..., Q_T)$ in terms of the likelihood $p(Q_1, ..., Q_T | X, E_1, ...E_T)$ and the prior distribution $p(P_1, ..., P_T)$. The next section discusses the prior distribution in greater detail.

2.2. Stochastic Rainfall Model

Rainfall has been studied extensively from a stochastic or probability-centric point of view (Guttorp, 1996; Rodriguez-Iturbe et al., 1987; Waymire & Gupta, 1981) with multiple distributional forms suggested for rainfall volumes (Cannon, 2008; Hanson & Vogel, 2008) and cross-day correlations for the presence and absence of rainfall (Holsclaw et al., 2016). With regard to the presence and absence of rainfall, two different formulations were considered in this study. In the first, our goal was to estimate precipitation inputs given knowledge of which days had rainfall; that is, the binary indicator variable $r_t = I_{P_t>0}$ is observed and is not a quantity to be estimated. In the second formulation, we treat r_t as a Bernoulli random variable parameterized by p_r, the probability of rain on any given day such that $\Pr(r_t = 1) = p_r$. For the latter case, we fixed the hyperparameter p_r to be equal to the fraction of days that rain was observed over the span of the data used. It is important to note that since p_r is a discrete random variable, the gradient of the model likelihood with regard to r_t is not well defined. We employ a reparameterization of r_t as a deterministic transformation of a continuous variable $r_t = I_{U_t>1-p_r}$, where $U_t \sim Uniform(0,1)$ to allow application of ADVI and HMC. Rainfall volumes are parameterized as $v_t \sim Lognormal(\mu, \sigma^2)$. μ and σ^2 both have their own weak lognormal prior distributions: $\mu \sim Lognormal(m_\mu = 2.0, s_\mu = 3.0)$ and $\sigma \sim Lognormal(m_\sigma = 0.5, s_\sigma = 1.0)$. The

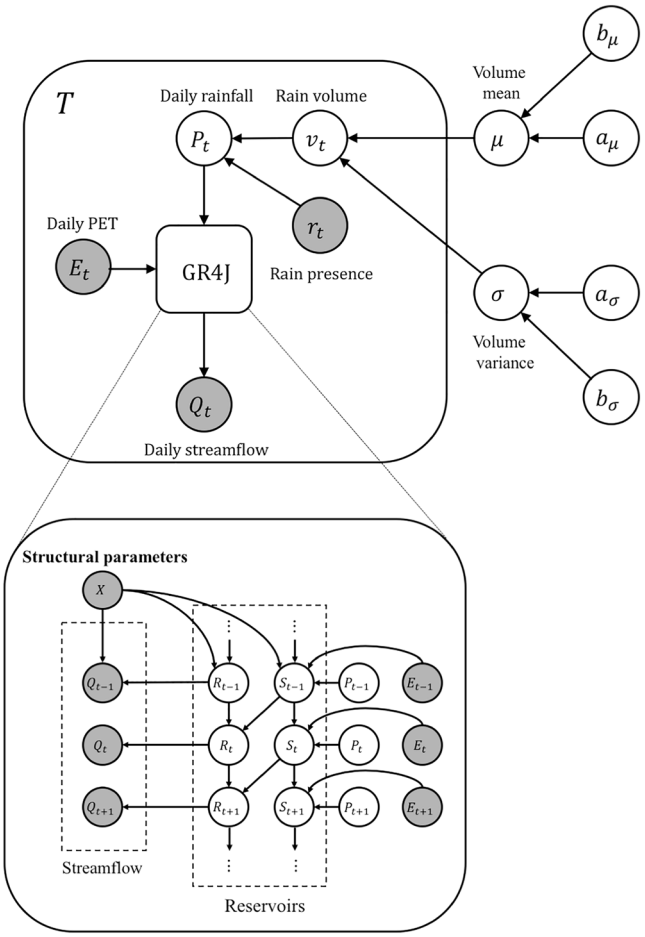

Figure 2. Combined stochastic rainfall-runoff model. Daily rainfall volumes are represented as draws from a lognormal distribution parameterized by mean μ and standard deviation σ. These are then used as inputs in GR4J, along with potential evapotranspiration (E_t). In this diagram we have assumed that the presence/absence of rain on any given day is a known variable, though we also consider the case in which it is unobserved.

hyperparameters for these prior distributions were chosen to provide a weak prior that roughly matches the observed marginal distribution of rainfall volumes across all the used data sets. Daily time series of precipitation are therefore defined as $P_t = v_t \cdot r_t$ under this model. Figure 2 depicts the combined GR4J-stochastic rainfall model in graphical form. We refer to the case where the presence of rainfall is observed as the lognormal rainfall model (LNR; Cho et al., 2004; Kedem & Chiu, 1987). The case in which the presence of rainfall is latent will be referred to as the Bernoulli-lognormal rainfall (B-LNR) model.

2.3. Parameter Estimation

We compared four algorithms for inverse estimation of P_t and r_t: random walk Metropolis (RWM; Haario et al., 2001), differential evolution Metropolis (DEM; Braak, 2006), the No-U-Turn Sampler (NUTS; Hoffman & Gelman, 2014), and ADVI (Kucukelbir et al., 2016). RWM, DEM, and NUTS are all Markov chain Monte Carlo (MCMC) methods, which require drawing large numbers of samples from the posterior distribution in order to compute the posterior density, while ADVI is a Bayesian approximation method, which does not draw samples from the true posterior. MCMC has been used extensively in hydrology for estimation of model structural parameters such as $X = \{x_1, ..., x_4\}$ in GR4J as well as statistical parameters such as the variance and autocorrelation of error processes (Bates & Campbell, 2001) and parameters for input error models (Kavetski et al., 2006). However, drawing sufficient numbers of samples to adequately estimate the posterior can be prohibitively slow for large data sets and models as each drawn sample requires evaluation of the likelihood $p(\boldsymbol{x}|\theta)$, a potentially expensive calculation. Each MCMC algorithm described here shares several common steps for generating samples of the parameter vector θ. A Markov chain $S = \boldsymbol{\theta_1}, \boldsymbol{\theta_2}, ..., \boldsymbol{\theta_n}$ with $\boldsymbol{n}$ samples is created by first initializing $\boldsymbol{\theta_1}$ to a random value and proposing a new candidate value for $\boldsymbol{\theta_2}$ given some proposal function. The key difference between RWM, DEM, and the NUTS is the mechanism by which new proposals are generated. Each of these three algorithms also includes a Metropolis step in which the posterior density of the proposed new value of $\boldsymbol{\theta_{t+1}}$ is compared to the posterior density of the current parameter value $\boldsymbol{\theta_t}$. If the posterior density is higher for the new value, then the Markov chain moves deterministically to that new value. If the posterior density is higher for the current parameter value, then the chain moves to the new candidate $\boldsymbol{\theta_{t+1}}$ with some probability less than one. Once the chain S is long enough, summaries of the posterior are generated from the samples drawn in S. For example, the posterior mean can be estimated by simply taking the mean of the values saved in the chain S.

An alternative strategy, commonly referred to as a variational Bayes approach, is to identify a parametric approximating distribution $q(\boldsymbol{\phi}, \theta)$ for the posterior $p(\theta|\boldsymbol{x})$ such that the parameters $\boldsymbol{\phi}$ are easily optimized to better match the posterior. However, identifying the form of the variational objective function usually requires a detailed analysis of the model likelihood, which is sufficiently difficult to be infeasible for many nonstatisticians. ADVI is a variational Bayesian algorithm that employs AD to derive an efficient formula for optimizing $\boldsymbol{q}$ with regard to the variational parameters $\boldsymbol{\phi}$ without human intervention. Additional details are given for each method in the following subsections.

2.3.1. Random Walk Metropolis

The RWM algorithm is the simplest of the methods we employed in this study. The proposal distribution used to generate candidate values of θ_{t+1} conditional on θ_t is a multivariate normal centered at θ_t and with a covariance matrix Σ that is updated during the tuning phase to be proportional to the covariance matrix of past samples. This enables the RWM sampler to make proposals that are rescaled to match the shape of the posterior distribution. This approach can be applied in virtually any Bayesian model fitting context but can

suffer from much longer convergence times for models with large numbers of parameters than methods that makes use of the gradient of the posterior density. In particular, statistical theory suggests that gradient-based methods such as NUTS and other HMC-based algorithms require less than $O\left(d^{\frac{1}{3}}\right)$ samples to converge to the posterior while RWM requires $O(d)$ samples (Mattingly et al., 2012) where d denotes the number of unknown parameters to be estimated. For both the RWM and NUTS algorithms, each sample requires at least one evaluation of the likelihood function. For the settings considered in this work, this evaluation involves running the entire forward model and therefore involves $O(d)$ operations. Thus, the computational effort involved in drawing samples that converge to the posterior is $O(d^2)$.

2.3.2. Differential Evolution Metropolis

DEM is a version of evolutionary MCMC in which K different MCMC chains $S_1,...,S_K$ are run in parallel. This is a probabilistic version of the widely used differential evolution optimization method (Storn & Price, 1997), which is a genetic algorithm incorporating mutation and crossover rules. At each sampling iteration, new candidates are generated for the kth chain by taking the most recent sample, that is, the vector θ_{tk} and adding to this vector a term proportional to the difference between the most recent samples from two other chains k_1,k_2 chosen at random from the population of K chains. Then, this new value is perturbed with additive noise ϵ drawn from a normal distribution (equation (1)). The parameter γ governs the trade-off between large and small jumps across the parameter space. We follow the recommendation of ter Braak (2006) and set $\gamma = 2.38 \cdot \sqrt{2d}$ where d is the dimensionality of the proposal distribution.

$$\theta_{t+1,k} = \theta_{t,k} + \gamma\left(\theta_{t,k_1} - \theta_{t,k_2}\right) + \epsilon \tag{1}$$

The number of chains used by DEM is a hyperparameter that must be set at the beginning of the estimation process. The recommendation offered by ter Braak (2006) is to use at least $K = 2 \cdot d$ chains though in the applications we consider here, the number of chains required by this rule would exceed 1,500 for some of our longer streamflow records. This precludes allocating a single chain to each processor. While refinements in evolutionary MCMC such as differential evolution adaptive Metropolis (Laloy & Vrugt, 2012; Vrugt, 2016; Vrugt & Ter Braak, 2011) appear to reduce this number to $K \approx d$, this linear scaling appears to make application of differential evolution adaptive Metropolis or a similar evolutionary MCMC method unattractive for estimation with desktop computers in very large inverse problems of the sort considered in section 3.2.

2.3.3. No-U-Turn Sampler

While the previous two methods make extensive use of random walk style methods in which exploration of the possible parameter space is achieved using random jumps, the NUTS is a variant of HMC (Neal, 2012), a type of MCMC that maps the model's posterior density to a potential energy surface $U(\theta) = e^{-f(\theta|x)}$. This energy surface is used to simulate physics-like dynamics with the goal of allowing the sampler to rapidly move across the posterior, favoring zones of high probability but with the ability to occasionally visit low-probability regions. As a result, trajectories of the sampler in parameter space tend to move toward regions of higher posterior density. This approach requires calculating the gradient of the energy function with regard to the model parameters but is known to be more effective than nongradient-based methods when used to estimate posteriors with high-dimensional parameter sets or cross-parameter correlations. HMC is highly attractive for high-dimensional Bayesian estimation as the number of samples required to converge to the posterior is $O\left(d^{\frac{1}{4}}\right)$ (Beskos et al., 2013) compared to $O(d)$ samples for the RWM algorithm. The NUTS (Hoffman & Gelman, 2014) is a variant of HMC in which the trajectories are not allowed to double back on themselves in order to reduce the amount of time spent sampling in regions of the posterior, which have already been explored. HMC is designed to work on difficult statistical modeling problems with large numbers of parameters such as Bayesian neural networks (Neal, 1996). For more information on HMC and NUTS, we refer the reader to a review and tutorial paper by Betancourt (2017). A major shortcoming of HMC and other MCMC methods relative to ADVI is that the MCMC scales poorly to large data sets and becomes prohibitively expensive in terms of compute time (Blei et al., 2017). Moreover, HMC is not currently applicable to models with discrete variables.

2.3.4. Automatic Differentiation Variational Inference

Bayesian parameter estimation for hydrological models can be difficult and time-consuming when the complexity of the model or length of the streamflow record is sufficiently long that drawing a single

sample of the parameter values takes more than a few seconds, as each MCMC method typically requires on the order of 10^2 to 10^5 samples to be drawn before convergence. This is especially true for estimation methods requiring the gradient of the model posterior density, as this requires more computation time than simply evaluating the likelihood. Nongradient-based MCMC methods may draw each sample relatively quickly but typically involve random walk exploration of the posterior without knowledge of its geometry. In high-dimensional parameter spaces, this is highly inefficient (Girolami & Calderhead, 2011), and it may take an unreasonably long time for the Markov chain estimates of the posterior to converge. Ideally, information from the gradient could be incorporated in a way that does not require large numbers of expensive samples to be drawn.

In statistics and machine learning, variational Bayes approximations (Fox & Roberts, 2012) are frequently used to calculate approximate the posterior distribution. We note that this is a qualitatively different kind of approximation from MCMC; while samples drawn using MCMC are guaranteed to converge to the true posterior, variational inference (VI) methods do not carry the same guarantee. A variational approach requires (1) making simplifying assumptions about the approximate posterior density $q(\boldsymbol{\phi}, \boldsymbol{\theta})$ that simplify calculations and (2) deriving a formula for the variational updates for the parameters $\boldsymbol{\phi}$ governing the approximate posterior. These updates iteratively decrease the distance between the true posterior and the approximating posterior. A common approach used for Step 1 is to assume that the probability distribution q factorizes as a product of marginal densities, that is, $q(\boldsymbol{\phi}, \boldsymbol{\theta}) = \prod_i q_i(\boldsymbol{\phi}_i, \boldsymbol{\theta}_i)$. Step 2 usually requires extensive manipulation of the model posterior density and is likely impossible to achieve with pen-and-paper calculations for hydrological models, which exhibit cross-time step dependencies and nonlinear dynamics. However, by employing AD, a formula for the variational update equations can be automatically derived (Kucukelbir et al., 2016). This enables practical usage of VI by users outside of the statistics and machine learning research communities. For a comprehensive overview of variational methods for parameter estimation, see Blei et al. (2017). After the optimization, samples can be drawn from the approximate posterior estimated by ADVI, though we note that no MCMC is involved in this step. In every application of ADVI mentioned in this study, 500 samples were drawn from the approximate posterior in order to calculate approximate posterior estimates of the inverse parameters.

2.3.5. Comparative Advantages and Disadvantages

Each of the methods described in this section have unique advantages and disadvantages; the RWM algorithm can draw samples relatively quickly but can require unreasonably large numbers of samples to estimate high-dimensional posteriors. Population sampling methods like DEM can perform better than RWM with larger parameter sets but require running an increasing number of chains with posterior dimension. NUTS and other HMC-based methods do well at estimating high-dimensional posterior distributions with a single chain but can require excessive amounts of time to draw individual samples. Theoretical analyses of MCMC convergence rates (Beskos et al., 2013; Mattingly et al., 2012; Pillai et al., 2012) indicate that MCMC methods using the gradient of the log likelihood require $O\left(d^{\frac{1}{3}}\right)$ or $O\left(d^{\frac{1}{4}}\right)$ samples to converge to the posterior, as opposed to $O(d)$ samples for RWM. When taking into account the computational cost of evaluating the likelihood function, which is typically $O(d)$, the overall complexity of NUTS and RWM is predicted to be $O\left(d^{\frac{5}{4}}\right)$ and $O(d^2)$, respectively. This comparative advantage is achieved by avoiding the random walk behavior exhibited by nongradient-based methods in high-dimensional settings (Mattingly et al., 2012). Furthermore, the posterior probability density in high-dimensional problems is known to concentrate in a small region of the parameter space, and navigation of this high-probability region is greatly enhanced by posterior curvature information provided by the gradient (Betancourt et al., 2014). In the next section, we describe a case study designed to evaluate the relative strengths and weaknesses of each approach for inverse parameter estimation problems in hydrology.

3. Data and Case Studies

We compared all the algorithms from the previous section as applied to synthetic data sets up to 12,000 time steps long (section 3.1). We included a case study employing real data over 25 years of daily observations to assess which methods could be employed for a very large estimation problem (section 3.2.1). An analysis of the suitability of these methods for uncertainty quantification using a single year of real data was also conducted (section 3.2.2)

3.1. Assessment With Simulated Data

We used simulated streamflow generated with GR4J with no obfuscating noise to test the ability and efficiency of each algorithm in an idealized setting. We obtained estimates of precipitation and PET spanning 50 days across the period 1 January 1948 to 19 February 1948 from the watershed of the Strawberry River in Arkansas, USA, via the MOPEX data set (Schaake et al., 2006). We set the model parameters to $x_1 = 100, x_2 = 0.5, x_3 = 15, x_4 = 2.5$ with initial conditions of $S_0 = 100$ and $R_0 = 8$. S_0 and R_0 are the initial water levels in the two reservoirs described in section 2.1.We then generated streamflow sequences of length T where T was allowed to vary over 20 values ranging from 25 to 12,000 time steps by repeating the first 50 days of precipitation and PET for all cases in which $T > 50$ for simplicity and to ensure temporal stationarity for the inputs. No noise was added to the simulated streamflow, but a normal error distribution with $\sigma = 0.1$ mm/day was assumed to induce a valid likelihood function. For each T, we attempted to recover the original precipitation inputs using the LNR model. We note that the number of days of streamflow T is not identical to the number of parameters d, which must be estimated. For all of the case studies in this paper, the number of days with nonzero rainfall volumes is between 50% and 60% of all days and therefore $d \approx 0.5T$ or $d \approx 0.6T$.

To compare performance of each algorithm with regard to convergence to the true posterior, we calculated the number of iterations and length of computation time required before a minimum accuracy threshold was reached. We required that the estimates had to reach a minimum Bayesian R^2 (Gelman et al., 2017) of 0.90. This time is abbreviated as τ_{90}, and a low value signifies that the inverse precipitation estimates are highly correlated with the true precipitation values with a small amount of time required for computation. These estimates were obtained by using the most recent samples or iterations; we used the preceding 10 samples for NUTS and the preceding 100 samples for RWM and DEM, as the latter two methods showed substantially higher sample autocorrelation. For ADVI, we used the approximate posterior of the single most recent optimization state. However, for increasing values of T the accuracy threshold was not reached within a reasonable amount of time for some algorithms, so we allowed for a maximum run time of 6 hr. Each of the sampling algorithms (RWM, DEM, and NUTS) was restarted three times to allow us to examine the influence of variation due to random initialization while only a single repetition was used for ADVI. For each repetition, 4 RWM chains, 10 DEM chains, 1 NUTS chain, and 1 ADVI optimization were run. In the case of RWM and NUTS, the chains operate independently of each other and thus repeating an experiment in triplicate with four chains in equivalent to running 12 chains sequentially. However, for DEM, the 10 chains constitute a population among which cross-chain updates are made at every iteration. We restricted all algorithms to 6 hr of run time; many of the DEM and RWM chains failed to converge within the time alotted for $d > 100$. We also estimated the scaling properties for each algorithm by estimating trends in $\log\tau_{90}$ in relation to $\log T$. We took the median τ_{90} value for each combination of algorithm and T and conducted a linear regression using ordinary least squares with the five largest problem sizes for which convergence was achieved for each algorithm. In the case of RWM, this meant using $T \in \{600,800,1,200,1,600,2,000\}$ while for ADVI and NUTS, $T \in \{5,000,\ 6,000,\ 8,000,\ 10,000,\ 12,000\}$.

3.2. Assessment With Real Data

3.2.1. Long-Term Point Estimates

The ability of each algorithm to reconstruct precipitation over several decades was also assessed using long-term records of streamflow, precipitation, and PET for three locations: the previously mentioned Strawberry River in Arkansas, USA (USGS gauge ID 07074000); the Leaf River in Mississippi, USA (02742000); and the French Broad River (03443000). We note that the latter two watersheds were studied in the context of inverse hydrological modeling in Vrugt et al. (2008). The data span the dates 1 January 1948 to 23 December 1977, and several summary statistics are listed in Figure 3. We attempted to apply all four estimation algorithms, though it was found that the time required to compute a minimum number of samples (in the case of the MCMC methods) to achieve convergence to the posterior was sufficiently large to be infeasible for this study. For RWM and DEM, acceptance rates under this model were too low to provide a reasonable estimate of sample autocorrelation with 6 hr of computation time. For NUTS, results obtained with 6 hr of computation produced multiple accepted proposals in the Markov chain but with a resulting Bayesian R^2 of less than 0.01. Consequently, in this section we focused on the abilities of ADVI to provide useful inverse estimates. Both the B-LNR and LNR models were used with an autoregressive lag-1 error process assumed for the

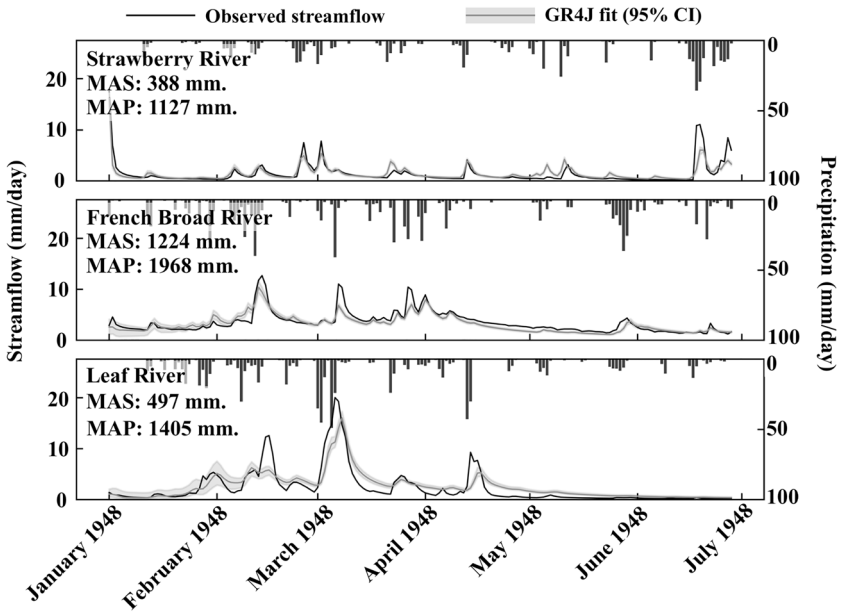

Figure 3. Streamflow and precipitation. These plots show the first 6 months of streamflow and precipitation used in section 3.2. Annual means of precipitation and streamflow are listed in the top left corner of each plot. The confidence intervals shown as gray shading reflect the posterior distribution over the GR4J structural parameters and do not incorporate a noise distribution.

observations. The GR4J structural parameters as well as standard deviation of the AR(1) increments and the autocorrelation were estimated from the first single year of data available for each catchment using four NUTS chains, which were allowed to run to convergence. Similar assessment criteria to the previous section were applied, and results from this assessment are discussed in section 4.2. We used 40,000 iterations of ADVI in each case, and this was a sufficiently large number of iterations to reach convergence for the ADVI loss function. In all three scenarios, estimation took less than the alotted 6 hr using a single core on a standard desktop computer.

3.2.2. Posterior Uncertainty Quantification

A major advantage to performing parameter estimation within a Bayesian framework is the availability of credible intervals and posterior uncertainty quantification for all estimated quantities. To determine whether or not the algorithms considered provided useful uncertainty estimates, we conducted an analysis of the posterior coverage properties of our precipitation estimates. We modified the analysis performed in the previous section to consist of a single year of data from each of the three watersheds with the LNR model and ran both NUTS and ADVI until convergence, as gauged by $\widehat{R}$ and the variational loss function, respectively. This required drawing 1,000 samples with NUTS and running ADVI for 40,000 iterations. We then calculated the fraction of times that the posterior credible intervals for the estimated precipitation amount contained the true value. We assumed a nominal coverage probability of 90%, indicating that we counted the number of instances in which the 90% credible interval included the true value. Actual coverage probabilities less than 90% indicate biased or overconfident estimates while coverage probabilities over 90% typically suggest inflated posterior variance.

4. Results

4.1. Simulated Data

Figure 4 displays a comparison of computation time and number of unknown variables. ADVI and NUTS were uniformly faster to reach convergence for all sequences of length greater than 300 days, and no DEM or RWM chains reached the accuracy threshold for any T > 2,000. Furthermore, most MCMC chains for RWM failed for $1{,}000 < T < 2{,}000$, with less than 3 out of 12 chains reaching the accuracy threshold in the allotted time. For ADVI, τ_{90} appeared to scale linearly with d and T while all of the MCMC method appear to

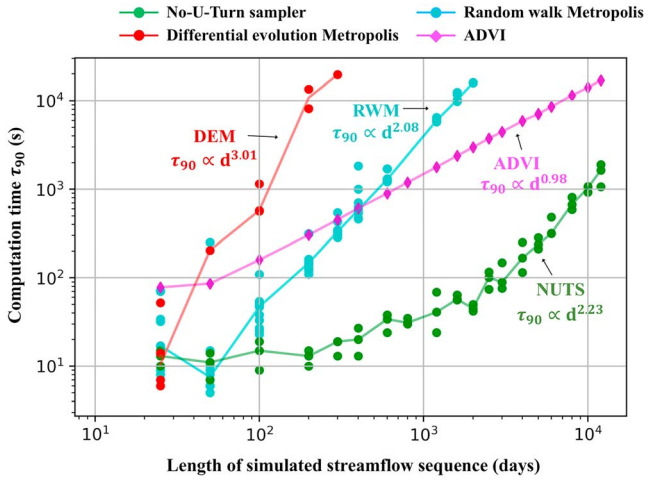

Figure 4. Comparison of estimation methods as measured by log computation time versus log T. For the Markov chain Monte Carlo methods, each marker indicates an independent chain (in the case of random walk Metropolis [RWM] or No-U-Turn Sampler [NUTS]) or population of chains (in the case of differential evolution Metropolis [DEM]). No RWM or DEM chain reached the requisite accuracy threshold for values of $T > 1{,}500$. ADVI = automatic differentiation variational inference

require time of $O(d^2)$ or greater. While NUTS was uniformly faster than RWM for every problem considered in this section, it appears that the $T_{90} \propto d^{\frac{5}{4}}$ scaling property of NUTS suggested by theory was not achieved in our experiments. Furthermore, although NUTS exhibited lower τ_{90} values in all cases with $T > 100$, the difference in scalability between ADVI and NUTS suggests that ADVI would become more efficient for $>10^5$ variables 4.

4.2. Multidecadal Estimation for Real Data

4.2.1. Long-Term Point Estimates

Inverse precipitation estimates for both the BLN-R and LN-R model using ADVI across all three catchments are shown in Figure 5. We computed Bayesian R^2 values for the 1-day totals, 10-day sums, and 30-day sums. In all cases, R^2 fell in the range 0.38–0.75. The Bernoulli-lognormal model had uniformly worse performance across every combination of summation time (i.e., 1, 10, or 30 days) and catchment. This is unsurprising given that estimation of the presence and absence of rain adds considerably to the difficulty of the problem. On the basis of comparison across 1-day precipitation estimates, the LNR model at the Strawberry River exhibited the highest predictive performance with $R^2 = 0.57$, and the worst case was the B-LNR model at the same site with $R^2 = 0.38$. The marginal distributions of precipitation volumes were also assessed for each site and are shown in figure 6. While both the LNR and B-LNR models underestimate the frequency of small rainfall events (< 1 mm in equivalent watershed depth), the upper tails of the true distribution appear to match the estimated posterior. We note that this is due to the distributional assumptions made in the stochastic model definition; including a more realistic distribution for daily rainfall volumes such as the Pearson Type III (Hanson & Vogel, 2008) could potentially lead to more accurate estimates. Two of the watersheds analyzed in this section were also the focus of a Bayesian inverse modeling study (Vrugt et al., 2008). However, a comparison of effectiveness with that work is not possible here, as the former assumed $d \approx 60$ while we have used $d > 6{,}000$, and there were also major differences in hydrological model and prior assumptions.

4.2.2. Posterior Uncertainty Quantification

Across the Strawberry River, French Broad River, and Leaf River data sets, the posterior coverage probabilities were 80%, 80%, and 82% for NUTS and 81%, 83%, and 81% for ADVI, respectively. Their departure from the nominal value of 90% indicates that the true posterior has not been captured completely by posterior estimates obtained using either NUTS or ADVI. Potential reasons for this discrepancy include structural model misspecification, inappropriate distributional assumptions for the error process, or bias due to an overly strong prior. As the prior distribution assumed over precipitation inputs was relatively diffuse, it appears that the former two causes are likely to be responsible. While variational Bayes methods are known to underestimate the posterior variance under certain conditions (Blei et al., 2017), the similar values for NUTS and ADVI coverage probabilities do not suggest that ADVI is performing worse due to this phenomenon.

5. Discussion

5.1. Comparison of Methods

One of the primary objectives of this work was to ascertain whether or not it is possible to write a hydrological model in a deep learning framework so as to perform model-based analyses such as inverse parameter estimation. We have found that it is straightforward and differs from the normal scientific programming procedure at a few points, mostly related to control flow such as *if* statements and *for* loops, which, in Theano, are replaced with *switch* and *scan* statements, which serve the same purpose, respectively. We have made our code available at github.com/ckrapu/gr4j_theano for any users wishing to extend or replicate the modeling conducted in this study. Additionally, we have found that the gradient information propagated through the model via AD does indeed appear to enable more advanced inference techniques. The model comparison performed in section 3.1 indicates that gradient-based methods for estimation

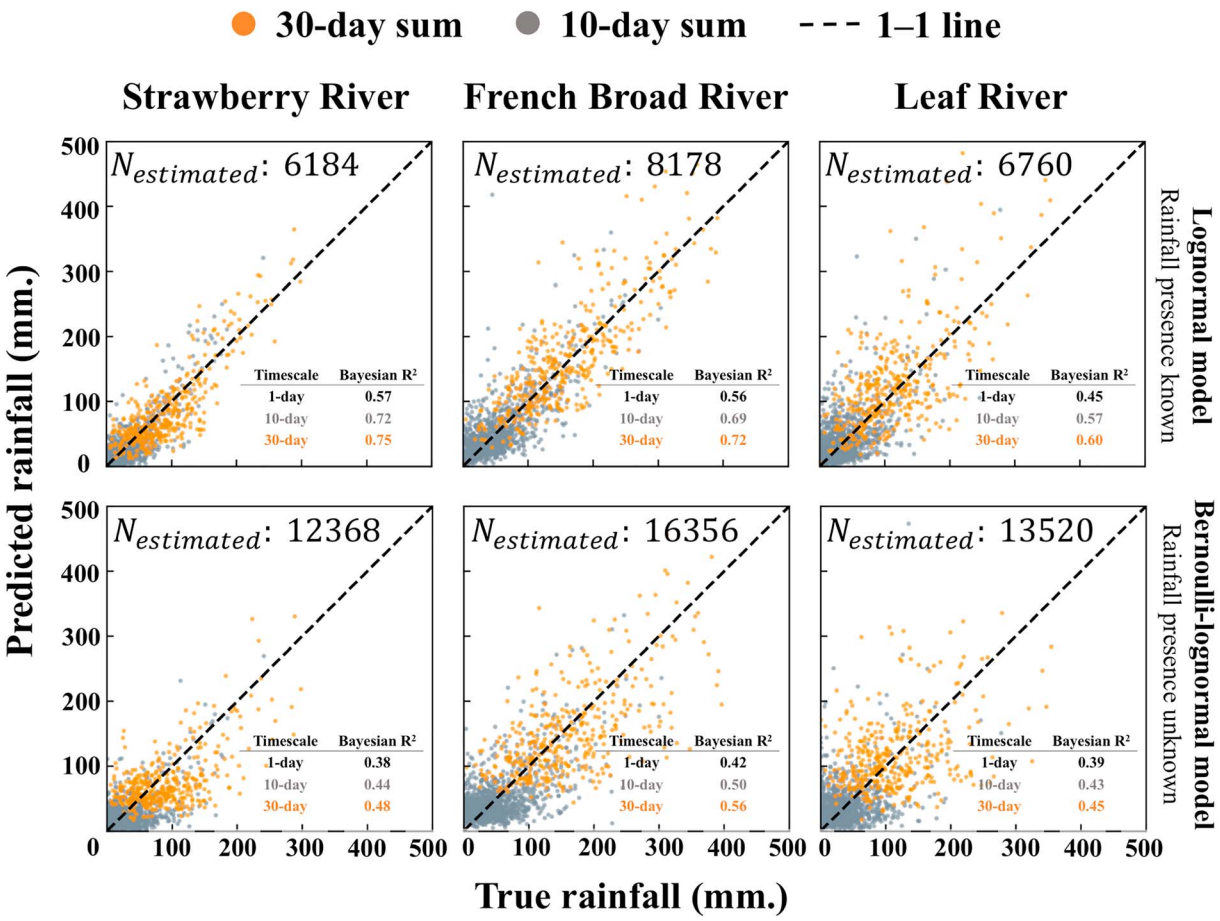

Figure 5. Inverse estimates of precipitation using real data. The 10- and 30-day sums were calculated for a single draw from the approximate posterior estimated by automatic differentiation variational inference. The Bayesian R^2 values reflect the accuracy of the inverse model evaluated over all 500 draws from the approximate posterior.

such as NUTS and ADVI outperform methods that do not incorporate this information. However, this comparison is not comprehensive, and it is possible that for relatively low-dimensional (d < 1,000) problems, implementing a hydrological model in an AD-enabled framework does not lead to major gains in sampling or estimation efficiency.

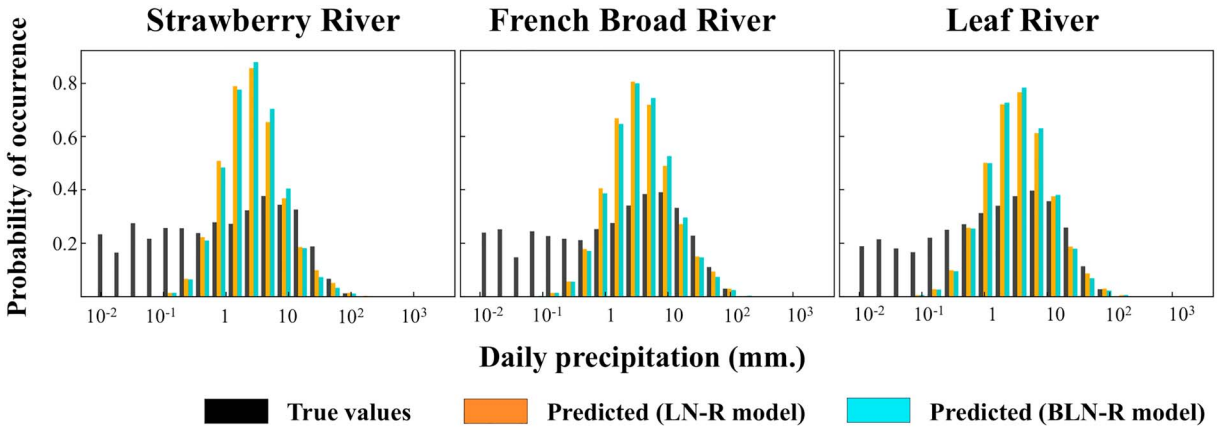

Figure 6. Marginal distribution of precipitation volumes. Our stochastic rainfall model differs from the true rainfall distribution for small events but accurately captures the right tail of the volume distribution. LNR = lognormal rainfall; B-LNR = Bernoulli-lognormal rainfall.

While the results presented in the comparative analysis (section 4.2) show relatively poor performance from random walk and DEM with increasing problem size as compared to NUTS and ADVI, in some cases, it may be feasible to simply draw many more samples until the chains have converged. It is also possible to use a combination of sampling methods to realize specific advantages of each one on different model subcomponents. For example, HMC could be used for numerous low-level parameters, which only influence a small portion of the model (Neal, 2012) such as individual rainfall volumes, while a simpler method such as the Metropolis algorithm could be used for high-level parameters such as the mean and variance of the prior distribution over rainfall volumes. Combining multiple MCMC methods often results in another valid sampling algorithm (Andrieu et al., 2003), and this approach removes the requirement for simulating Hamiltonian dynamics over parameters that influence many different parts of the model and therefore have computationally expensive gradients. Integration of MCMC and variational methods is also a potential option (Salimans et al., 2015), though algorithms incorporating this approach are relatively new and not yet available in commonly used parameter estimation frameworks. While we have shown results for estimation assuming a simple rainfall-runoff model, the general approach of embedding a hydrological model into a deep learning framework may also be applicable to more sophisticated models with multiple spatial units and potentially even physically based models as well (Schenck & Fox, 2018).

A surprising finding from this study is that ADVI adheres closely to $O(d)$ scaling of the time required for convergence, while NUTS exceeds the $O\left(d^{\frac{5}{4}}\right)$ convergence times predicted in the statistical theory literature for HMC (Neal, 2011). This may be due to additional computational overhead attributable to a suboptimal implementation of either the model or gradient calculation procedure or violations of the assumptions underlying existing studies of convergence rates on HMC. Additionally, we found that while NUTS was capable of rapidly converging toward the posterior in the synthetic data case for $d > 10^3$, similar performance could not be achieved with the real data set. This suggests that NUTS is sensitive to model misspecification and prior assumptions. In the real data set, ADVI was the only algorithm able to obtain precipitation estimates with any correspondence to the true values.

5.2. Implications for Hydrolgical Modeling

While this study has focused on showing a proof of concept for Bayesian inverse estimation with models on the order of 10^3–10^4 parameters, the methods used are known to scale to models with 10^6–10^7 parameters (Tran et al., 2018) and beyond. The strategy of combining a continuous data-generating process with a probability distribution over the inputs and the error model is generic and amenable to standard Bayesian inferential techniques such as MCMC and scalable methods such as variational inference. As virtually all hydrological models are composed of continuous functions, the suitability of gradient-based optimization for their calibration and inverse estimation is guaranteed. We note that this also allows for a more balanced class of models between process-centric and purely empirical approaches; for example, joint estimation of the parameters of a model comprised of the sum of a process-based hydrological model and an empirical component such as an ARIMA or a recurrent neural network-based error process is now possible. Alternatively, subcomponents of process models corresponding to poorly understood or underconstrained mechanisms could be replaced with empirical representations to improve predictive accuracy or provide statistical uncertainty quantification. While we have assumed in this work that the structural model parameters are known, it is straightforward to treat them as additional unknown quantities to be estimated jointly with the model inputs. It may even be possible to then use the inverse estimation procedure as a sort of validation check; a hydrological model that is sufficiently constrained and identified in parameter space could be assessed via the accuracy of its inverse estimates as well as the accuracy of its forward simulation.

Previously, the calibration or parameter estimation techniques relied on in hydrology were not feasible for large empirical or statistical models; this work shows that techniques designed for the latter are applicable to process-based models in principle. Much remaining work must be done to determine whether hydrological models with more advanced ODE solvers or a high spatial resolution (Kollet & Maxwell, 2006; Kumar et al., 2009; Shen & Phanikumar, 2010) can be accommodated in this framework. Fortunately, the problem of using gradient-based methods to estimate ODE parameters has already attracted substantial interest in the machine learning community (Chen et al., 2018), and it is possible that the solutions and insights gleaned from that research will translate to more efficient parameter estimation for environmental models.

Inverse modeling is a problem of interest across the natural sciences where models are often both ill-posed and high dimensional. While the former issue can be addressed with adequate Bayesian priors, effective methods for model inversion in Bayesian setting with many parameters have only recently been available. We note that our work parallels observations from researchers in geophysics (Fichtner et al., 2019) who have also noted the potential for inverse modeling with HMC. While this study has focused on inverse modeling across long time periods for a single site, it is possible that this same approach would be effective for spatially distributed inverse modeling. Previous studies on this topic have explored using two stage estimation/optimization approches (Grundmann et al., 2019) or regularized inversion (Kretzschmar et al., 2016), and it remains to be seen whether the approach advocated in this work compares favorably with existing methods.

6. Conclusions

Currently, there is significant attention paid to assessing the suitability and effectiveness of deep neural networks in hydrological modeling. It is also possible to use deep learning-style estimation methods for hydrology-specific model forms such as rainfall-runoff models. This approach is highly effective when the number of parameters is large, though not all gradient-based estimation algorithms scale effectively to large data sets. Our comparison of estimation methods in a synthetic data set shows that ADVI and the NUTS are much more viable methods for parameter estimation when the dimensionality of the parameter space exceeds 10^3 than previously used Bayesian estimation algorithms. We found that the total computational effort required for ADVI was $O(d)$, which compares favorable to $>O(d^2)$ time required by RWM and DEM. This work should encourage further careful consideration of the interplay between machine learning and hydrology and the optimization methods employed in each discipline.

Acknowledgments
The authors would like to acknowledge support from NASA via the Earth and Space Sciences Graduate Fellowship and the NSF via an IGERT graduate traineeship and Grants EAR-1331846 and EAR-1454983. Additional financial assistance and equipment has been provided by the Duke University Wetland Center and by Nvidia and Amazon via an equipment and server credit grant, respectively. All data used in this study are available from the National Weather Service website (www.nws.noaa.gov/ohd/mopex/mo_datasets.htm).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous distributed systems 19.

Andrieu, C., de Freitas, N., Doucet, A., & Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine Learning*, *39*.

Bates, B. C., & Campbell, E. P. (2001). A Markov chain Monte Carlo scheme for parameter estimation and inference in conceptual rainfall-runoff modeling. *Water Resources Research*, *37*(4), 937–947. https://doi.org/10.1029/2000WR900363

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research*, *18*(153).

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., et al. (2010). Theano: A CPU and GPU math compiler in Python 7.

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., & Stuart, A. (2013). Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, *19*(5A), 1501–1534. https://doi.org/10.3150/12-BEJ414

Betancourt, M., 2017. A conceptual introduction to Hamiltonian Monte Carlo. arXiv preprint:1701.02434 60.

Betancourt, M.J., Byrne, S., Livingstone, S., Girolami, M., 2014. The geometric foundations of Hamiltonian Monte Carlo. arXiv:1410.5110 [stat].

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, *112*(518), 859–877. https://doi.org/10.1080/01621459.2017.1285773

Boyle, D. P., Gupta, H. V., & Sorooshian, S. (2003). Multicriteria calibration of hydrologic models. In Q. Duan, H. V. Gupta, S. Sorooshian, A. N. Rousseau, & R. Turcotte (Eds.), *Water Science and Application*, (pp. 185–196). Washington, D. C: American Geophysical Union. https://doi.org/10.1029/WS006p0185

Braak, C. J. F. T. (2006). A Markov chain Monte Carlo version of the genetic algorithm Differential Evolution: Easy Bayesian computing for real parameter spaces. *Statistics and Computing*, *16*(3), 239–249. https://doi.org/10.1007/s11222-006-8769-1

Cannon, A. J. (2008). Probabilistic multisite precipitation downscaling by an expanded Bernoulli–Gamma density network. *Journal of Hydrometeorology*, *9*(6), 1284–1300. https://doi.org/10.1175/2008JHM960.1

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

Chen, R.T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D., 2018. Neural ordinary differential equations. arXiv:1806.07366 [cs, stat].

Cho, H.-K., Bowman, K. P., & North, G. R. (2004). A comparison of gamma and lognormal distributions for characterizing satellite rain rates from the tropical rainfall measuring mission. *Journal of Applied Meteorology*, *43*(11), 1586–1597. https://doi.org/10.1175/JAM2165.1

Dawson, C. W., & Wilby, R. L. (2001). Hydrological modelling using artificial neural networks. *Progress in Physical Geography: Earth and Environment*, *25*(1), 80–108. https://doi.org/10.1177/030913330102500104

Duane, S., Kennedy, A. D., Pendleton, B. J., & Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, *195*(2), 216–222. https://doi.org/10.1016/0370-2693(87)91197-X

Evin, G., Thyer, M., Kavetski, D., McInerney, D., & Kuczera, G. (2014). Comparison of joint versus postprocessor approaches for hydrological uncertainty estimation accounting for error autocorrelation and heteroscedasticity. *Water Resources Research*, *50*, 2350–2375. https://doi.org/10.1002/2013WR014185

Fichtner, A., Zunino, A., & Gebraad, L. (2019). Hamiltonian Monte Carlo solution of tomographic inverse problems. *Geophysical Journal International*, *216*(2), 1344–1363. https://doi.org/10.1093/gji/ggy496

Fox, C. W., & Roberts, S. J. (2012). A tutorial on variational Bayesian inference. *Artificial Intelligence Review*, *38*(2), 85–95. https://doi.org/10.1007/s10462-011-9236-8

Gelman, A., Goodrich, B., Gabry, J., Ali, I., 2017. R-squared for Bayesian regression models*.

Girolami, M., & Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods: Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73*(2), 123–214. https://doi.org/10.1111/j.1467-9868.2010.00765.x

Govindaraju, R. S. (2000). Artificial neural networks in hydrology. II: Hydrologic applications. *Journal of Hydrologic Engineering*, *5*, 124–137. https://doi.org/10.1061/(ASCE)1084-0699(2000)5:2(124)

Grundmann, J., Hörning, S., & Bárdossy, A. (2019). Stochastic reconstruction of spatio-temporal rainfall patterns by inverse hydrologic modelling. *Hydrology and Earth System Sciences*, *23*(1), 225–237. https://doi.org/10.5194/hess-23-225-2019

Guttorp, P., 1996. Stochastic modeling of rainfall, in: Environmental Studies, The IMA Volumes in Mathematics and Its Applications, DOI: https://doi.org/10.1007/978-1-4613-8492-2_7.

Haario, H., Saksman, E., & Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, *7*(2), 223. https://doi.org/10.2307/3318737

Hanson, L. S., & Vogel, R. (2008). The probability distribution of daily rainfall in the United States. In *World Environmental and Water Resources Congress 2008. Presented at the World Environmental and Water Resources Congress 2008*, (pp. 1–10). Honolulu, Hawaii, United States: American Society of Civil Engineers. https://doi.org/10.1061/40976(316)585

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*, 1593–1623.

Holsclaw, T., Greene, A. M., Robertson, A. W., & Smyth, P. (2016). A Bayesian hidden Markov model of daily precipitation over South and East Asia. *Journal of Hydrometeorology*, *17*(1), 3–25. https://doi.org/10.1175/JHM-D-14-0142.1

Iri, M., Tsuchiya, T., & Hoshi, M. (1988). Automatic computation of partial derivatives and rounding error estimates with applications to large-scale systems of nonlinear equations. *Journal of Computational and Applied Mathematics*, *24*(3), 365–392. https://doi.org/10.1016/0377-0427(88)90298-1

Jakeman, A. J., Littlewood, I. G., & Whitehead, P. G. (1990). Computation of the instantaneous unit hydrograph and identifiable component flows with application to two small upland catchments. *Journal of Hydrology*, *117*(1-4), 275–300. https://doi.org/10.1016/0022-1694(90)90097-H

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., et al. (2014). Caffe: Convolutional architecture for fast feature embedding. arXiv:1408.5093 [cs].

Kavetski, D., & Clark, M. P. (2010). Ancient numerical daemons of conceptual hydrological modeling: 2. Impact of time stepping schemes on model analysis and prediction: Numerical daemons of hydrological modeling, 2. *Water Resources Research*, *46*, W10511. https://doi.org/10.1029/2009WR008896

Kavetski, D., Kuczera, G., & Franks, S. W. (2006). Bayesian analysis of input uncertainty in hydrological modeling. *Water Resources Research*, *42*, W03407. https://doi.org/10.1029/2005WR004368

Kedem, B., & Chiu, L. S. (1987). On the lognormality of rain rate. *Proceedings of the National Academy of Sciences*, *84*(4), 901–905. https://doi.org/10.1073/pnas.84.4.901

Kingston, G. B., Maier, H. R., & Lambert, M. F. (2008). Bayesian model selection applied to artificial neural networks used for water resources modeling: BMS OF ANNS IN WATER RESOURCES MODELING. *Water Resources Research*, *44*, W04419. https://doi.org/10.1029/2007WR006155

Kollet, S. J., & Maxwell, R. M. (2006). Integrated surface–groundwater flow modeling: A free-surface overland flow boundary condition in a parallel groundwater flow model. *Advances in Water Resources*, *29*(7), 945–958. https://doi.org/10.1016/j.advwatres.2005.08.006

Kretzschmar, A., Tych, W., Chappell, N., & Beven, K. (2016). What really happens at the end of the rainbow?—Paying the price for reducing uncertainty (using reverse hydrology models). *Procedia Engineering*, *154*, 1333–1340. https://doi.org/10.1016/j.proeng.2016.07.485

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., Blei, D.M. (2016). Automatic differentiation variational inference. arXiv:1603.00788 [cs, stat].

Kuczera, G., & Parent, E. (1998). Monte Carlo assessment of parameter uncertainty in conceptual catchment models: the Metropolis algorithm. *Journal of Hydrology*, *211*(1-4), 69–85. https://doi.org/10.1016/S0022-1694(98)00198-X

Kumar, M., Duffy, C. J., & Salvage, K. M. (2009). A second-order accurate, finite volume–based, integrated hydrologic modeling (FIHM) framework for simulation of surface and subsurface flow. *Vadose Zone Journal*, *8*(4), 873. https://doi.org/10.2136/vzj2009.0014

Laloy, E., & Vrugt, J. A. (2012). High-dimensional posterior exploration of hydrologic models using multiple-try DREAM-ZS. *Water Resources Research*, *48*, W01526. https://doi.org/10.1029/2011WR010608

Lindström, G., Johansson, B., Persson, M., Gardelin, M., & Bergström, S. (1997). Development and test of the distributed HBV-96 hydrological model. *Journal of Hydrology*, *201*(1-4), 272–288. https://doi.org/10.1016/S0022-1694(97)00041-3

Marçais, J., & de Dreuzy, J.-R. (2017). Prospective interest of deep learning for hydrological inference: J. Marçais and J.-R. de Dreuzy Groundwater xx, no. x: xx-xx. *Groundwater*, *55*(5), 688–692. https://doi.org/10.1111/gwat.12557

Mattingly, J. C., Pillai, N. S., & Stuart, A. M. (2012). Diffusion limits of the random walk Metropolis algorithm in high dimensions. *The Annals of Applied Probability*, *22*(3), 881–930. https://doi.org/10.1214/10-AAP754

Neal, R. (2011). MCMC Using Hamiltonian Dynamics. In S. Brooks, A. Gelman, G. Jones, & X.-L. Meng (Eds.), *Handbook of Markov chain Monte Carlo*. Chapman and Hall/CRC (Chap. 5, pp. 113–162). London. https://doi.org/10.1201/b10905-6

Neal, R. M. (1996). *Bayesian learning for neural networks, lecture notes in statistics*. New York, NY: Springer New York. https://doi.org/10.1007/978-1-4612-0745-0

Neal, R.M. (2012). MCMC using Hamiltonian dynamics. arXiv:1206.1901 [physics, stat].

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., et al. (2017). Automatic differentiation in PyTorch. Presented at the 31st Conference on Neural Information Processing Systems, p. 4.

Pathiraja, S., Moradkhani, H., Marshall, L., Sharma, A., & Geenens, G. (2018). Data-driven model uncertainty estimation in hydrologic data assimilation. *Water Resources Research*, *54*, 1252–1280. https://doi.org/10.1002/2018WR022627

Perrin, C., Michel, C., & Andréassian, V. (2003). Improvement of a parsimonious model for streamflow simulation. *Journal of Hydrology*, *279*(1-4), 275–289. https://doi.org/10.1016/S0022-1694(03)00225-7

Pillai, N. S., Stuart, A. M., & Thiéry, A. H. (2012). Optimal scaling and diffusion limits for the Langevin algorithm in high dimensions. *The Annals of Applied Probability*, *22*(6), 2320–2356. https://doi.org/10.1214/11-AAP828

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

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, *22*(3), 400–407. https://doi.org/10.1214/aoms/1177729586

Rodriguez-Iturbe, I., Cox, D. R., & Isham, V. (1987). Some models for rainfall based on stochastic point processes. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *410*(1839), 269–288. https://doi.org/10.1098/rspa.1987.0039

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0

Salimans, T., Kingma, D. P., & Welling, M. (2015). Markov chain Monte Carlo and variational inference: Bridging the gap. *Journal of Machine Learning Research*, *9*.

Salvatier, J., Wiecki, T. V., & Fonnesbeck, C. (2016). Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, *2*, e55. https://doi.org/10.7717/peerj-cs.55

Santos, L., Thirel, G., & Perrin, C. (2018). Continuous state-space representation of a bucket-type rainfall-runoff model: A case study with the GR4 model using state-space GR4 (version 1.0). *Geoscientific Model Development*, *11*(4), 1591–1605. https://doi.org/10.5194/gmd-11-1591-2018

Schaake, J., Cong, S., Duan, Q., 2006. The US Mopex data set (No. IAHS Publication 308).

Schenck, C., Fox, D., 2018. SPNets: Differentiable fluid dynamics for deep neural networks. Arxiv 19.

Shen, C., & Phanikumar, M. S. (2010). A process-based, distributed hydrologic model based on a large-scale method for surface–subsurface coupling. *Advances in Water Resources*, *33*(12), 1524–1541. https://doi.org/10.1016/j.advwatres.2010.09.002

Smith, T., Marshall, L., & Sharma, A. (2015). Modeling residual hydrologic errors with Bayesian inference. *Journal of Hydrology*, *528*, 29–37. https://doi.org/10.1016/j.jhydrol.2015.05.051

Smith, T. J., & Marshall, L. A. (2008). Bayesian methods in hydrologic modeling: A study of recent advancements in Markov chain Monte Carlo techniques. *Water Resources Research*, *44*, W00B05. https://doi.org/10.1029/2007WR006705

Storn, R., & Price, R. (1997). Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359. https://doi.org/10.1023/A:1008202821328

Thyer, M., Renard, B., Kavetski, D., Kuczera, G., Franks, S. W., & Srikanthan, S. (2009). Critical evaluation of parameter consistency and predictive uncertainty in hydrological modeling: A case study using Bayesian total error analysis. *Water Resources Research*, *45*, W00B14. https://doi.org/10.1029/2008WR006825

Tokar, A., & Johnson, P. (1999). Rainfall-runoff modeling using artificial neural networks. *Journal of Hydrologic Engineering*, *4*(3), 232–239. https://doi.org/10.1061/(ASCE)1084-0699(1999)4:3(232)

Tran, D., Hoffman, M., Moore, D., Suter, C., Vasudevan, S., Radul, A., et al. (2018). Simple, distributed, and accelerated probabilistic programming. arXiv:1811.02091 [cs, stat].

Tran, D., Kucukelbir, A., Dieng, A.B., Rudolph, M., Liang, D., Blei, D.M. (2016). Edward: A library for probabilistic modeling, inference, and criticism. arXiv:1610.09787 [cs, stat].

Vrugt, J. A. (2016). Markov chain Monte Carlo simulation using the DREAM software package: Theory, concepts, and MATLAB implementation. *Environmental Modelling & Software*, *75*, 273–316. https://doi.org/10.1016/j.envsoft.2015.08.013

Vrugt, J. A., & Ter Braak, C. J. F. (2011). DREAM: an adaptive Markov Chain Monte Carlo simulation algorithm to solve discrete, non-continuous, and combinatorial posterior parameter estimation problems. *Hydrology and Earth System Sciences*, *15*(12), 3701–3713. https://doi.org/10.5194/hess-15-3701-2011

Vrugt, J. A., ter Braak, C. J. F., Clark, M. P., Hyman, J. M., & Robinson, B. A. (2008). Treatment of input uncertainty in hydrologic modeling: Doing hydrology backward with Markov chain Monte Carlo simulation. *Water Resources Research*, *44*, W00B09. https://doi.org/10.1029/2007WR006720

Waymire, E., & Gupta, V. K. (1981). The mathematical structure of rainfall representations: 1. A review of the stochastic rainfall models. *Water Resources Research*, *17*(5), 1261–1272. https://doi.org/10.1029/WR017i005p01261

Zhang, J., Zhu, Y., Zhang, X., Ye, M., & Yang, J. (2018). Developing a long short-term memory (LSTM) based model for predicting water table depth in agricultural areas. *Journal of Hydrology*, *561*, 918–929. https://doi.org/10.1016/j.jhydrol.2018.04.065