Speed-up of posterior inference of highly-parameterized environmental models from a Kalman proposal distribution: DREAM$_{(KZS)}$

Jiangjiang Zhang[1], Jasper A. Vrugt[2,3], Xiaoqing Shi[4], Guang Lin[5], Laosheng Wu[6], and Lingzao Zeng[1*]

[1] Zhejiang Provincial Key Laboratory of Agricultural Resources and Environment, Institute of Soil and Water Resources and Environmental Science, College of Environmental and Resource Sciences, Zhejiang University, Hangzhou, 310058, China,

[2] Department of Civil and Environmental Engineering, University of California Irvine, Irvine, CA, 92697, USA,

[3] Department of Earth System Science, University of California Irvine, Irvine, CA, 92697, USA,

[4] Key Laboratory of Surficial Geochemistry of Ministry of Education, School of Earth Sciences and Engineering, Nanjing University, Nanjing, 210023, China,

[5] Department of Mathematics & School of Mechanical Engineering, Purdue University, West Lafayette, IN, 47907, USA,

[6] Department of Environmental Sciences, University of California Riverside, Riverside, CA, 92521, USA.

*Correspondence to:

L. Zeng,
lingzao@zju.edu.cn

Abstract

Markov chain Monte Carlo (MCMC) simulation methods are widely used to generate samples from a target distribution. In posterior inference of highly-parameterized environmental models, the convergence speed of MCMC methods may be disturbingly low, even with the state-of-the-art algorithms, such as DREAM$_{(ZS)}$ (differential evolution adaptive Metropolis). At each iteration, DREAM$_{(ZS)}$ generates the proposal distributions with a mix of parallel direction jump and snooker jump that are only based on the information about the model parameters in the thinned chain history. In this study, to speed up the convergence of DREAM$_{(ZS)}$, we introduce a Kalman proposal distribution that utilizes the information contained in the covariance structure of the model parameters, the measurements and the model outputs. Compared with the parallel direction jump and the snooker jump, the Kalman jump can generate a more directional update of the model parameters. As the Kalman jump cannot maintain detailed balance, we restrict it only to the "burn-in" period and use the other two jumps with diminishing adaptation afterwards. The modified algorithm is called DREAM$_{(KZS)}$ as it uses the three jumps simultaneously with pre-defined probabilities. Numerical experiments demonstrate that DREAM$_{(KZS)}$ converges to the same posterior distribution as DREAM$_{(ZS)}$ but with much lower computational budget. Specifically, in problems with about 100 unknown model parameters, the saving can be as big as 20 times.

1. Introduction

Environmental modeling is an important tool for analyzing and predicting the behavior of a system that integrates hydrological, ecological and biogeochemical processes, etc. [*Schnoor*, 1996]. However, there are many sources of uncertainty that would hamper an accurate prediction of the system behavior of concern. These uncertainties are usually originated from model conceptualization and data collection, which include errors in the model structure, initial and boundary conditions, model parameters and measurement data [*Clark et al.*, 2011; *Refsgaard et al.*, 2012; *Vrugt*, 2016; *Wagener and Gupta*, 2005]. In many cases, model parameters cannot be measured directly, or one can only obtain very sparse measurements of some spatially distributed parameters [*Vereecken et al.*, 2016]. In this situation, it is common practice to calibrate the model parameters against the measurements, where Bayesian methods are ideal tools as they can handle uncertainties in the model parameters and outputs in a coherent and consistent manner.

Let's assume that the observation process of an arbitrary environmental system can be expressed as:

$$\mathbf{d} = f(\mathbf{m}) + \boldsymbol{\varepsilon}, \tag{1}$$

where $\mathbf{d} = \{d_1, \dots, d_n\}$ is a n-vector for the measurements, $f(\cdot)$ is a numerical model for the environmental system, $\mathbf{m} = \{m_1, \dots, m_k\}$ is a k-vector for the unknown model parameters, and $\boldsymbol{\varepsilon} = \{\varepsilon_1, \dots, \varepsilon_n\}$ is a n-vector for the measurement errors, respectively. In this work, $\boldsymbol{\varepsilon}$ is assumed to follow a zero-mean multivariate Gaussian distribution with the covariance matrix $\boldsymbol{\Sigma}$, i.e., $\boldsymbol{\varepsilon} \sim \mathcal{N}_n(\mathbf{0}, \boldsymbol{\Sigma})$.

In the Bayesian framework, we simulate all the quantities as random variables. Before assimilating any measurement data, our knowledge about the model parameters is represented by the prior distribution. When the measurements are available, we can evaluate the goodness of fit between the simulated model outputs and the measurements with the likelihood function. Conditioned on the measurements, our knowledge about the model parameters is updated and represented by the posterior distribution. According to Bayes' theorem, the posterior distribution is expressed as:

$$p(\mathbf{m}|\mathbf{d}) = \frac{p(\mathbf{m})p(\mathbf{d}|\mathbf{m})}{p(\mathbf{d})}, \tag{2}$$

where $p(\mathbf{m})$ and $p(\mathbf{m}|\mathbf{d})$ signify the prior and posterior probability density functions (pdfs), $L(\mathbf{m}|\mathbf{d}) \equiv p(\mathbf{d}|\mathbf{m})$ is the likelihood function, and $p(\mathbf{d}) = \int p(\mathbf{d}|\mathbf{m})p(\mathbf{m})d\mathbf{m}$ is the evidence, which is a normalizing constant. In practice, we usually don't need to evaluate the evidence $p(\mathbf{d})$, then we can rewrite equation (2) as:

$$p(\mathbf{m}|\mathbf{d}) \propto p(\mathbf{m})L(\mathbf{m}|\mathbf{d}). \tag{3}$$

When the measurement errors are modeled as zero-mean Gaussian random variables, the likelihood function can be expressed as:

$$L(\mathbf{m}|\mathbf{d}) = \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}[\mathbf{d} - f(\mathbf{m})]^{\mathrm{T}}\mathbf{\Sigma}^{-1}[\mathbf{d} - f(\mathbf{m})]\right\}. \tag{4}$$

For complex, nonlinear systems, analytical forms of the posterior distribution are nonexistent. To obtain a numerical approximation, one has to resort to Monte Carlo simulation methods. Over the past decades, Markov chain Monte Carlo (MCMC) simulation has become increasingly popular in the inference of the posterior statistics. MCMC simulation works by constructing a Markov chain to explore the parameter space and successively draw samples from the posterior distribution. Before the Markov chain reaches its stationary regime, it has to sufficiently explore the parameter space, i.e., it requires a "burn-in" period.

MCMC was first introduced by *Metropolis et al.* [1953] with the random walk Metropolis (RWM) algorithm. Then *Hastings* [1970] extended the RWM algorithm to more general cases by considering non-symmetrical jumping distributions, which is well known as the Metropolis-Hastings (M-H) algorithm. In the past decades, considerable effort has been devoted to improving the efficiency of the original RWM and M-H algorithms by adaptively tuning the proposal distribution. One famous example is the differential evolution adaptive Metropolis (DREAM) algorithm developed by *Vrugt et al.* [2009b]. The DREAM algorithm is based on the differential evolution Markov chain algorithm [*Ter Braak,* 2006] but uses randomized subspace sampling and outlier chain correction. Based on DREAM, some extensions have also been made, e.g., DREAM$_{(ZS)}$ and MT-DREAM$_{(ZS)}$ that are more suitable for high-

dimensional inverse problems [*Laloy and Vrugt*, 2012; *Laloy et al.*, 2013], DREAM$_{(D)}$ for problems with discrete and combinatorial posterior distributions [*Vrugt and Ter Braak*, 2011] and DREAM$_{(ABC)}$ for diagnostic model evaluation [*Sadegh and Vrugt*, 2014], etc. Due to the high efficiency, the family of the DREAM algorithms have found very wide applications in different fields like geophysics, hydrology and soil science [*Bikowski et al.*, 2012; *Keating et al.*, 2010; *Linde and Vrugt*, 2013; *Lochbühler et al.*, 2014; *Muleta et al.*, 2012; *Ramin et al.*, 2014; *Shi et al.*, 2014; *Wöhling and Vrugt*, 2011; *Xu et al.*, 2017b; *Ying et al.*, 2017; *Zeng et al.*, 2016; *Zhang et al.*, 2013]. For more details, one can refer to [*Vrugt*, 2016] and the references therein.

In this work, our focus is the DREAM$_{(ZS)}$ algorithm, which is more efficient than the original DREAM algorithm and does not require the forceful treatment of outlier chains used in DREAM. However, in posterior inference of highly-parameterized, CPU-intensive environmental models, DREAM$_{(ZS)}$ may not necessarily be efficient enough. For example, if the number of unknown model parameters is more than 100, then one may need hundreds of thousands of model evaluations to sample the target distribution [*Laloy and Vrugt*, 2012; *Laloy et al.*, 2013]. In DREAM$_{(ZS)}$, two ways of generating proposal distributions, i.e., the parallel direction jump and the snooker jump, are used. However, the two jumps are solely based on the information about the model parameters in the thinned chain history, which may not be very efficient for high-dimensional problems.

Here, we introduce a Kalman proposal distribution that is especially designed to speed up the convergence of DREAM$_{(ZS)}$ in exploring high-dimensional target distributions. The new proposal distribution is based on the updating scheme in the ensemble smoother (ES) [*Evensen*, 2007] that only updates the model parameters by assimilating all historical measurements. In the Kalman jump, the information contained in the covariance structure of the model parameters, the measurements and the model outputs is utilized. Thus the Kalman jump can generate a more directional update of the model parameters than the parallel direction jump and the snooker jump. The Kalman proposal was also suggested by *Vrugt et al.* [2013] to update samples of a particle-MCMC filter, but this idea has not been implemented in [*Vrugt et al.*, 2013].

As the Kalman jump cannot maintain detailed balance, we will restrict it to the "burn-in" period. After that, we will use a mix of the parallel direction jump and the snooker jump to sample the posterior, which can satisfy detailed balance with diminishing adaptation. As all the three jumps will be used in the modified algorithm, we call it the DREAM$_{(KZS)}$ algorithm in this paper, where K stands for the Kalman jump, Z stands for the parallel direction jump and S stands for the snooker jump.

The remainder of this paper is organized as follows. We first formulate the DREAM$_{(KZS)}$ algorithm in Section 2. Then in Section 3, we compare the performance of DREAM$_{(KZS)}$ with DREAM$_{(ZS)}$ in numerical case studies involving non-linearity and high-dimensionality. Finally, some conclusions and discussions are provided in Section 4.

2. Methods

2.1. DREAM$_{(ZS)}$: the basic algorithm

In most cases, analytical forms of the posterior, $p(\mathbf{m}|\mathbf{d})$ are nonexistent. In this situation, we can resort to an MCMC algorithm, e.g., DREAM$_{(ZS)}$, to sample the posterior distribution. MCMC explores the parameter space by constructing a Markov chain that gradually converges to its equilibrium distribution, i.e., the posterior distribution. The Markov chain evolves so that more effort will be spent in the posterior regions. This is realized by moving the Markov chain from the current state, $\mathbf{m}_{t-1}$ to a candidate state, $\mathbf{m}_{\mathrm{p}}$ that mimics a sample drawn from the posterior distribution [*Andrieu et al.*, 2003]. The probability of accepting $\mathbf{m}_{\mathrm{p}}$ as the next state of the Markov chain is calculated as:

$$p_{\mathrm{acc}}\left(\mathbf{m}_{t-1} \rightarrow \mathbf{m}_{\mathrm{p}}\right) = \min\left[1, \frac{p\left(\mathbf{m}_{\mathrm{p}}\right) q\left(\mathbf{m}_{\mathrm{p}} \rightarrow \mathbf{m}_{t-1}\right)}{p\left(\mathbf{m}_{t-1}\right) q\left(\mathbf{m}_{t-1} \rightarrow \mathbf{m}_{\mathrm{p}}\right)}\right], \tag{5}$$

where $p(\mathbf{m}_{\mathrm{p}})$ and $p(\mathbf{m}_{t-1})$ are the posterior densities of $\mathbf{m}_{\mathrm{p}}$ and $\mathbf{m}_{t-1}$, which are proportional to the product of the corresponding prior density times the likelihood; $q(\mathbf{m}_{\mathrm{p}} \rightarrow \mathbf{m}_{t-1})$ and $q(\mathbf{m}_{t-1} \rightarrow \mathbf{m}_{\mathrm{p}})$ are the conditional probabilities of trail moves

from $\mathbf{m}_{\mathrm{p}}$ to $\mathbf{m}_{t-1}$ and from $\mathbf{m}_{t-1}$ to $\mathbf{m}_{\mathrm{p}}$, respectively. In DREAM$_{(ZS)}$, a symmetric jumping distribution is used, i.e., $q(\mathbf{m}_{\mathrm{p}} \rightarrow \mathbf{m}_{t-1}) = q(\mathbf{m}_{t-1} \rightarrow \mathbf{m}_{\mathrm{p}})$. Thus equation (5) simplifies to :

$$p_{\mathrm{acc}}(\mathbf{m}_{t-1} \rightarrow \mathbf{m}_{\mathrm{p}}) = \min\left[1, \frac{p(\mathbf{m}_{\mathrm{p}})}{p(\mathbf{m}_{t-1})}\right]. \tag{6}$$

If $p_{\mathrm{acc}}(\mathbf{m}_{t-1} \rightarrow \mathbf{m}_{\mathrm{p}})$ is larger than a random draw, u from the standard uniform distribution, $u \sim \mathcal{U}(0,1)$, we accept the candidate state, i.e., $\mathbf{m}_t = \mathbf{m}_{\mathrm{p}}$; Otherwise, the Markov chain will stay at the previous state, i.e., $\mathbf{m}_t = \mathbf{m}_{t-1}$. After a number of iterations, the Markov chain will converge to its stationary regime and then the states in the Markov chain can be viewed as samples drawn from the posterior distribution.

The efficiency of an MCMC algorithm is largely determined by the choice of the proposal distribution that generates the candidate state. In DREAM$_{(ZS)}$, N Markov chains are evolving simultaneously. The candidate state for the ith chain ($i = 1, \ldots, N$) is generated using differential evolution based on an archive of past states:

$$\mathbf{m}_{\mathrm{p}}^{i} = \mathbf{m}_{t-1}^{i} + \Delta\mathbf{m}^{i}, \tag{7}$$

where $\Delta\mathbf{m}^{i}$ is the jump distance from the current state, $\mathbf{m}_{t-1}^{i}$ to the candidate state, $\mathbf{m}_{\mathrm{p}}^{i}$, which is calculated as:

$$\begin{aligned} &\Delta\mathbf{m}_{A}^{i} = \zeta_{d^*} + (1_{d^*} + \lambda_{d^*})\gamma_{(\delta,d^*)} \sum_{j=1}^{\delta} \left(\mathbf{Z}_{A}^{\mathbf{a}_j} - \mathbf{Z}_{A}^{\mathbf{b}_j}\right), \\ &\Delta\mathbf{m}_{\neq A}^{i} = 0. \end{aligned} \tag{8}$$

In the above equation, A is a d^*-dimensional subset of the original k-dimensional parameter space. Equation (8) means that we only update d^* randomly selected parameters in $\mathbf{m}_{t-1}^{i}$ and keep the rest $k - d^*$ parameters unchanged; ζ_{d^*} and λ_{d^*} are random samples drawn from a multivariate normal distribution, $\mathcal{N}_{d^*}(0, e_*)$ and a multivariate uniform distribution, $\mathcal{U}_{d^*}(-e, e)$, respectively. The default values for e_* and e are 10^{-12} and 0.05, respectively; $\gamma_{(\delta,d^*)} = 2.38/\sqrt{2\delta d^*}$ is the jump rate and δ is the number of chain pairs that are used to generate the jump. In DREAM$_{(ZS)}$, the value of γ is periodically (e.g., with a 20% chance) set as 1, which facilitates jumps

from one mode to another in the posterior distribution; $\mathbf{Z}$ is a $k \times z$ matrix for the thinned chain history with z parameter samples; $\mathbf{a}$ and $\mathbf{b}$ are δ integers drawn from $\{1, \dots, z\}$ without replacement. It is noted here that the length of the archive $\mathbf{Z}$ increases linearly with the iteration number. Thus, z is not a fixed value.

The above way of generating candidate states is called the parallel direction jump. To enhance the diversity of the proposals, another way to generate candidate states, i.e., the snooker jump [*Ter Braak and Vrugt*, 2008], is also used in DREAM$_{(ZS)}$. The distance of the snooker jump is calculated as:

$$\Delta \mathbf{m}^i = \zeta_k + \gamma\left(\mathbf{Z}_\perp^b - \mathbf{Z}_\perp^c\right), \tag{9}$$

where ζ_k are random samples drawn from the multivariate normal distribution, $\mathcal{N}_k(0, e_*)$; γ is the jump rate of the snooker update, which is randomly drawn from the uniform distribution, $\mathcal{U}(1.2, 2.2)$; $\mathbf{Z}_\perp^b$ and $\mathbf{Z}_\perp^c$ are projection points of $\mathbf{Z}^b$ and $\mathbf{Z}^c$ onto the direction from $\mathbf{m}_{t-1}^i$ to $\mathbf{Z}^a$, respectively; a, b and c are integers drawn from $\{1, \dots, z\}$ without replacement.

In DREAM$_{(ZS)}$, the relative growth of the archive decreases over time, this diminishing adaptation can guarantee that the Markov chains are ergodic and can converge to the posterior distribution [*Roberts and Rosenthal*, 2007]. Here, the probabilities of using the parallel direction jump and the snooker jump are p_{Z} and p_{S}, where $p_{\mathrm{Z}} \geq 0$, $p_{\mathrm{S}} \geq 0$ and $p_{\mathrm{Z}} + p_{\mathrm{S}} = 1$.

2.2. The Kalman jump

From equations (8) and (9) we can see that, the parallel direction jump and the snooker jump are solely based on the information about the model parameters in the thinned chain history. If the information contained in the covariance structure of the model parameters, the measurements and the model outputs is utilized to generate the proposals, a more directional jump can be obtained. One method that fits this situation is the ensemble smoother [*Evensen*, 2007], which can update the model parameters and model states in the optimal direction by making best use of the measurements. Here, we adopt a modified scheme of the ensemble smoother that only updates the model parameters by assimilating all historical measurements to generate the jump for $\mathbf{m}_{t-1}^i$:

$$\begin{aligned}\Delta\mathbf{m}^i &= \mathbf{C}_{\mathrm{MD}}(\mathbf{C}_{\mathrm{DD}}+\boldsymbol{\Sigma})^{-1}\left[\mathbf{d}+\boldsymbol{\varepsilon}_{t-1}^i - f\left(\mathbf{m}_{t-1}^i\right)\right] \\ &= \mathbf{K}_{t-1}^i\boldsymbol{\varepsilon}_{t-1}^i + \mathbf{K}_{t-1}^i\mathbf{r}_{t-1}^i,\end{aligned} \tag{10}$$

where $\mathbf{C}_{\mathrm{MD}}$ is the $k \times n$ cross-covariance matrix between the model parameters and model outputs; $\mathbf{C}_{\mathrm{DD}}$ is the $n \times n$ auto-covariance matrix of the model outputs; $\boldsymbol{\Sigma}$ is the $n \times n$ covariance matrix of the measurement errors, and $\boldsymbol{\varepsilon}_{t-1}^i$ is one set of randomly simulated measurement noise drawn from the multivariate normal distribution, $\mathcal{N}_n(\mathbf{0},\boldsymbol{\Sigma})$; $\mathbf{K}_{t-1}^i = \mathbf{C}_{\mathrm{MD}}(\mathbf{C}_{\mathrm{DD}}+\boldsymbol{\Sigma})^{-1}$ is the Kalman gain and it is calculated from an archive of past states in τ chains. Here, the chain indices are τ integers randomly drawn from $\{1,\dots,i-1,i+1,\dots,N\}$ without replacement; $\mathbf{r}_{t-1}^i = \mathbf{d} - f\left(\mathbf{m}_{t-1}^i\right)$ is the residual vector. Then the proposal state $\mathbf{m}_{\mathrm{p}}^i$ can be obtained according to equations (7) and (10).

Obviously, the Kalman jump is asymmetric, i.e., $q\left(\mathbf{m}_{t-1}^i \to \mathbf{m}_{\mathrm{p}}^i\right) \neq q\left(\mathbf{m}_{\mathrm{p}}^i \to \mathbf{m}_{t-1}^i\right)$. As symmetric jumping distribution is considered in DREAM$_{(ZS)}$, here we introduce a backward jump of the Kalman proposal distribution to satisfy symmetry. The backward jump is implemented as follows. If the candidate state $\mathbf{m}_{\mathrm{p}}^i$ is accepted, i.e., $\mathbf{m}_t^i = \mathbf{m}_{\mathrm{p}}^i$, the jump distance for $\mathbf{m}_t^i$ at the next iteration is directly calculated as:

$$\Delta\mathbf{m}^i = -\mathbf{K}_{t-1}^i\boldsymbol{\varepsilon}_t^i - \mathbf{K}_{t-1}^i\mathbf{r}_{t-1}^i, \tag{11}$$

where $\boldsymbol{\varepsilon}_t^i$ is another random sample drawn from $\mathcal{N}_n(\mathbf{0},\boldsymbol{\Sigma})$. This treatment can guarantee that $q\left(\mathbf{m}_{t-1}^i \to \mathbf{m}_{\mathrm{p}}^i\right) = q\left(\mathbf{m}_{\mathrm{p}}^i \to \mathbf{m}_{t-1}^i\right)$ and enable us to use equation (6) to calculate the acceptance rate directly.

It should be noted here that in many cases the distribution of the measurement errors is not available. In this situation, we can estimate the measurement error parameters together with the unknown model parameters. Then in the Kalman jump described in equation (10), the measurement error covariance, $\boldsymbol{\Sigma}$ and the measurement error realization, $\boldsymbol{\varepsilon}_{t-1}^i$ are all generated from the measurement error parameters at iteration $t-1$ in the ith chain.

2.3. DREAM$_{(KZS)}$

Here, we augment DREAM$_{(ZS)}$ with the Kalman jump to enhance the diversity of the proposals. This extension is referred to as DREAM$_{(KZS)}$, as we use the three jumps simultaneously with pre-defined probabilities. As the Kalman jump can generate a rather directional update of the model parameters, the forward jump described in equation (10) will be very likely accepted and the backward jump described in equation (11) will be very likely rejected. At this point, the Kalman jump cannot maintain detailed balance. To address this issue, we can restrict the Kalman jump only to the "burn-in" period. Afterwards we will use the parallel direction jump and the snooker jump to sample the posterior distribution and the detailed balance is satisfied with diminishing adaptation. It is expected that introducing the Kalman jump in DREAM$_{(ZS)}$ can speed up the convergence of MCMC simulations.

The implementation detail of DREAM$_{(KZS)}$ is shown in Algorithm 1. Here z_0 $(z_0 > N)$ is the initial size of the archive $\mathbf{Z}$. From this initial archive, we randomly draw N samples as the starting states of the N Markov chains. This initial archive is also used to generate the candidate states for the N starting states; p_{K}, p_{Z} and p_{S} are the probabilities of using the Kalman jump, the parallel direction jump and the snooker jump during the "burn-in" period. Here $p_{\mathrm{K}} \geq 0, p_{\mathrm{Z}} \geq 0, p_{\mathrm{S}} \geq 0$, and $p_{\mathrm{K}} + p_{\mathrm{Z}} + p_{\mathrm{S}} = 1$. For example, we can set the "burn-in" period, $T_{\mathrm{burn-in}}$ as 20% of the total iteration number, T, i.e., $T_{\mathrm{burn-in}} = 0.2 \times T$; p'_{Z} and p'_{S} are the probabilities of using the parallel direction jump and the snooker jump after the "burn-in" period. Here $p'_{\mathrm{Z}} \geq 0, p'_{\mathrm{S}} \geq 0$, and $p'_{\mathrm{Z}} + p'_{\mathrm{S}} = 1$. Every T_{thin} generations, we will append the current N states in the Markov chains to the archive $\mathbf{Z}$, thus the number of samples in the archive $\mathbf{Z}$ will grow from z to $z + N$.

Algorithm 1 The DREAM$_{(KZS)}$ algorithm.

1. Draw z_0 samples from the prior distribution to create the initial archive **Z**.
2. **for** $t = 2, \ldots, T$ **do**
    for $i = 1, \ldots, N$ **do**
        Draw a random sample u from the uniform distribution, $\mathcal{U}(0,1)$.
        if $t \leq T_{\mathrm{burn-in}}$ **then**
            Generate a candidate state $\mathbf{m}_{\mathrm{p}}^{i}$ using the Kalman jump, the parallel direction jump or the snooker jump with the selection probabilities of p_{K}, p_{Z} and p_{S}.
        else
            Generate a candidate state $\mathbf{m}_{\mathrm{p}}^{i}$ using the parallel direction jump or the snooker jump with the selection probabilities of p'_{Z} and p'_{S}.
        end if
        Evaluate the prior density times the likelihood for $\mathbf{m}_{t-1}^{i}$ and $\mathbf{m}_{\mathrm{p}}^{i}$, calculate the acceptance rate $p_{\mathrm{acc}}\left(\mathbf{m}_{t-1}^{i} \rightarrow \mathbf{m}_{\mathrm{p}}^{i}\right)$ according to equation (6).
        if $u \leq p_{\mathrm{acc}}\left(\mathbf{m}_{t-1}^{i} \rightarrow \mathbf{m}_{\mathrm{p}}^{i}\right)$ **then** Accept $\mathbf{m}_{\mathrm{p}}^{i}$ and set $\mathbf{m}_{t}^{i} = \mathbf{m}_{\mathrm{p}}^{i}$.
        else Reject $\mathbf{m}_{\mathrm{p}}^{i}$ and set $\mathbf{m}_{t}^{i} = \mathbf{m}_{t-1}^{i}$.
        end if
    end for
    Append the current N states in the Markov chains to the archive **Z** when $\mathrm{mod}(t, T_{\mathrm{thin}}) = 0$.
end for

3. Illustrative examples

In this section, we illustrate the performance of the DREAM$_{(KZS)}$ algorithm by application to three different case studies involving the simulation of surface and subsurface hydrological processes. Compared with the original DREAM$_{(ZS)}$ algorithm, a significant improvement in the convergence speed will be observed through introducing the Kalman proposal distribution, especially in high-dimensional inverse problems.

3.1. Example 1: A rainfall-runoff model

We first test a 7-parameter model, known as the hmodel, that describes the rainfall-runoff process at the Guadalupe River basin at Spring Branch, Texas [*Duan et al.*, 2006]. In this model, precipitation that falls in the watershed undergoes various processes including evaporation, runoff generation, interception, percolation, and distribution between fast and slow reservoirs, then it discharges from the outlet of the watershed through surface and subsurface routings. The 7 parameters and the corresponding prior

ranges are listed in Table 1. For a detailed description of the model structure and underling processes, one can refer to [*Schoups and Vrugt*, 2010]. This model is also included in the DREAM software package as a benchmark test for diagnostic model evaluation [*Vrugt*, 2016].

[Table 1]

To infer the 7 unknown model parameters, we run DREAM$_{(ZS)}$ and DREAM$_{(KZS)}$ respectively conditioned on the measured streamflow data. The measurements are generated from one set of true model parameters $\mathbf{m}^*$ (Table 1) perturbed with additive measurement errors that follow $\mathcal{N}_n(\mathbf{0}, \boldsymbol{\sigma}^2)$, where $n = 1827$ is the number of measurement data, $\boldsymbol{\sigma} = 0.05 \times f(\mathbf{m}^*)$ is the standard deviation of the measurement error, and $f(\cdot)$ signifies the hmodel. In both algorithms, there are $N = 4$ parallel chains evolving simultaneously with a total chain length of 6000, which means 24,000 model evaluations in total. In DREAM$_{(ZS)}$, the probabilities of using the parallel direction jump and the snooker jump are $p_{\mathrm{Z}} = 90\%$ and $p_{\mathrm{S}} = 10\%$, respectively. In the first 20% simulation time of DREAM$_{(KZS)}$, the probabilities of using the Kalman jump, the parallel direction jump and the snooker jump are $p_{\mathrm{K}} = 20\%$, $p_{\mathrm{Z}} = 72\%$ and $p_{\mathrm{S}} = 8\%$, while in the remaining 80% simulation, $p_{\mathrm{K}} = 0\%$, $p_{\mathrm{Z}} = 90\%$ and $p_{\mathrm{S}} = 10\%$. The convergence of the two algorithms is monitored with the $\hat{R}$-statistic proposed by *Gelman and Rubin* [1992]. Evolutions of the $\hat{R}$-statistics of the 7 parameters in DREAM$_{(ZS)}$ and DREAM$_{(KZS)}$ are shown in Figure 1(a) and Figure 1(b), respectively. When the $\hat{R}$-statistics of all the 7 parameters are below 1.2 (the red dashed lines in Figure 1), we can declare that the Markov chains converge to the equilibrium distribution. As shown in Figure 1, DREAM$_{(KZS)}$ converges faster than DREAM$_{(ZS)}$, which signifies an improved efficiency by introducing the Kalman proposal distribution in DREAM$_{(ZS)}$.

[Figure 1]

In Figure 2, We compare the marginal posterior pdfs obtained by DREAM$_{(ZS)}$ (red curves) and DREAM$_{(KZS)}$ (blue dashed curves) estimated via a Gaussian kernel using

the last 4000 samples in the Markov chains. It is found that, the two algorithms can obtain almost identical results, which indicates the estimation accuracy of DREAM$_{(KZS)}$.

[Figure 2]

To further convince the performance of DREAM$_{(KZS)}$, we run some extra simulations with different sets of true model parameters and observe similar results. In Figure 3, we compare the multivariate $\hat{R}$-statistics obtained by DREAM$_{(ZS)}$ (magenta curves) and DREAM$_{(KZS)}$ (blue curves) given 4 different sets of true model parameters. In each subplot, 4 curves with the same color signify 4 repeated simulations with DREAM$_{(ZS)}$ or DREAM$_{(KZS)}$ given the same set of measurements generated from the same set of true model parameters. It should be noted here that, in Figure 1 we draw the evolution of the $\hat{R}$-statistics of the 7 parameters separately, while in Figure 3 we monitor the overall convergence with the multivariate $\hat{R}$-statistics. From Figure 3, it is found that DREAM$_{(KZS)}$ generally converges faster than DREAM$_{(ZS)}$ (it is noted here that, when doing more simulation tests of this example, we observe that occasionally DREAM$_{(KZS)}$ performs similarly as DREAM$_{(ZS)}$ in convergence speed).

[Figure 3]

In the above simulations, the distribution of the measurement errors is assumed to be known, i.e., the mean is zero and the standard deviation is 5% of the true model response. While in many cases, the measurement error distribution is unknown. In this situation, we can estimate the parameters that describe the measurement error distribution together with the unknown model parameters using DREAM$_{(KZS)}$. For example, we can define a model, $\boldsymbol{\sigma} = a + b * \mathbf{d}$ for the standard deviation of the measurement error if it is linearly dependent on the measurement. Here a and b are two extra unknown parameters for the measurement error distribution, whose prior distributions are $\mathcal{U}(0,1)$ and $\mathcal{U}(0,1)$, respectively. At iteration $t-1$, the current state in the ith chain is $\left[\mathbf{m}_{t-1}^{i};\ a_{t-1}^{i};\ b_{t-1}^{i}\right]$. Then we can use the corresponding model outputs, $f(\mathbf{m}_{t-1}^{i})$ and the estimated standard deviation of the measurement error, $\boldsymbol{\sigma}_{t-1}^{i} = a_{t-1}^{i} + b_{t-1}^{i} * \mathbf{d}$ to evaluate the likelihood function defined in equation (4).

Similarly, we can calculate the likelihood of the proposal, $\left[\mathbf{m}_p^i;\ a_p^i;\ b_p^i\right]$ and decide whether to accept it according to equation (6). Using the same settings for DREAM$_{(KZS)}$ as in the previous simulations, the two types of unknown parameters can be well identified. As shown in Figure 4, the estimated posterior pdfs (blue curves) can cover the true values (black vertical lines) of the model parameters and the parameters for the measurement error distribution.

[Figure 4]

3.2. Example 2: High-dimensional contaminant source identification

When dealing with high-dimensional inverse problems, MCMC usually requires a very large number of model evaluations, which generally means a prohibitive computational cost. In the following example, we will show that when the number of unknown model parameters is large, DREAM$_{(KZS)}$ can converge much faster than DREAM$_{(ZS)}$, thus save a lot of computational cost.

[Figure 5]

Here we consider the process of contaminant transport in a steady-state saturated groundwater flow system. As shown in Figure 5, the flow domain is $20[L] \times 10[L]$ with no-flow conditions at the upper and lower boundaries, constant-head conditions at the left and right boundaries, respectively. The conductivity ($K[LT^{-1}]$) field is heterogeneous and its log-transformed values ($Y = \ln K$) are assumed to be spatially correlated in the following way:

$$C_Y(x_1, y_1; x_2, y_2) = \sigma_Y^2 \exp\left(-\frac{|x_1 - x_2|}{\lambda_x} - \frac{|y_1 - y_2|}{\lambda_y}\right), \tag{12}$$

where (x_1, y_1) and (x_2, y_2) are two arbitrary locations in the flow domain, $\sigma_Y^2 = 1$ is the variance of the *Y* field, $\lambda_x = 10[L]$ and $\lambda_y = 5[L]$ are the correlation lengths along the *x* and *y* directions, respectively. To reduce the parametric dimensionality, we represent the *Y* field with the Karhunen-Loève (KL) expansion [*Zhang and Lu*, 2004]:

$$Y(\mathbf{x}) \approx \bar{Y}(\mathbf{x}) + \sum_{i=1}^{N_{\mathrm{KL}}} \sqrt{\tau_i} s_i(\mathbf{x}) \xi_i, \tag{13}$$

where $\bar{Y}(\mathbf{x}) = 2$ is the mean value of the *Y* field; $s_i(\mathbf{x})$ and τ_i are the eigenfunctions and eigenvalues of the covariance kernel defined in equation (12); ξ_i are i.i.d. Gaussian random variables, hereafter referred to as KL terms, that fit $\xi_i \sim \mathcal{N}(0,1), i = 1, \ldots, N_{\mathrm{KL}}$; $N_{\mathrm{KL}} = 100$ is the number of truncated KL terms that can preserve about 95% of the field variance, i.e., $\sum_{i=1}^{N_{\mathrm{KL}}} \tau_i / \sum_{i=1}^{\infty} \tau_i \approx 0.95$. Then we can obtain steady-state hydraulic head, $h[L]$ and pore water velocity, $v[LT^{-1}]$ by numerically solving the following equations with MODFLOW [*Harbaugh et al.*, 2000]:

$$\frac{\partial}{\partial x_i}\left(K_i \frac{\partial h}{\partial x_i}\right) = 0, \tag{14}$$

and

$$v_i = -\frac{K_i}{\theta}\frac{\partial h}{\partial x_i}, \tag{15}$$

where the subscript i demotes the component along the respective coordinate axis $(i = 1, 2)$, $\theta = 0.25[-]$ is the porosity of the aquifer.

In the steady-state flow field, some amount of contaminant is released from an unknown point source. The contaminant source is located somewhere in the area denoted by the red dashed rectangle in Figure 5. Its releasing strength (mass-loading rate, $[MT^{-1}]$) varies with time and can be described by 6 parameters in 6 time segments, i.e., $S_i[MT^{-1}]$ during $t_i = i[T]: (i+1)[T]$, for $i = 1, \ldots, 6$. Concentration of the contaminant, $C[ML^{-3}]$ at different times and locations can be obtained by numerically solving the following advection dispersion equation with MT3DMS [*Zheng and Wang*, 1999]:

$$\frac{\partial(\theta C)}{\partial t} = \frac{\partial}{\partial x_i}\left(\theta D_{ij}\frac{\partial C}{\partial x_j}\right) - \frac{\partial}{\partial x_i}(\theta v_i C) + q_{\mathrm{s}} C_{\mathrm{s}}, \tag{16}$$

where $q_{\mathrm{s}}[T^{-1}]$ is the volumetric flow rate per unit volume of the aquifer; $C_{\mathrm{s}}[ML^{-3}]$ is the concentration of the contaminant source; $D_{ij}[L^2T^{-1}]$ is the hydrodynamic dispersion tensor, which has the following component forms:

$$\begin{cases} D_{xx} = (\alpha_L v_x^2 + \alpha_T v_y^2)/|\boldsymbol{v}|, \\ D_{yy} = (\alpha_L v_y^2 + \alpha_T v_x^2)/|\boldsymbol{v}|, \\ D_{xy} = D_{yx} = (\alpha_L - \alpha_T) v_x v_y/|\boldsymbol{v}|, \end{cases} \tag{17}$$

where D_{xx} and D_{yy} are the principal components of the dispersion tensor, $[L^2T^{-1}]$, D_{xy} and D_{yx} are the cross terms, $[L^2T^{-1}]$, $\alpha_L = 0.3[L]$ is the longitudinal dispersivity, $\alpha_T = 0.03[L]$ is the transverse dispersivity, v_x and v_y are the components of the velocity $\boldsymbol{v}$ along the *x* and *y* directions, $|\boldsymbol{v}| = \sqrt{v_x^2 + v_y^2}$ is the magnitude of the velocity, respectively.

In this example, there are 108 unknown model parameters, i.e., the 100 KL terms, $\xi_i (i = 1, \dots, 100)$, that parameterize the log conductivity field; the location of the contaminant source, $(x_s, y_s)[L]$, and the time-varying source strengths, $(S_1, \dots, S_6)[MT^{-1}]$. Here we assume that the 8 source parameters are uniformly distributed, whose ranges are listed in Table 2.

[Table 2]

To infer the 108 unknown model parameters, we collect measurements of the hydraulic head and concentration at 15 wells denoted by the blue dots in Figure 5. As the flow field is steady, we collect the head measurements only once, while the concentration measurements are collected every $1[T]$ from $4[T]$ to $12[T]$. The errors for the head and concentration measurements are assumed to be independent and Gaussian with zero means and standard deviations of $0.005[L]$ and $0.005[ML^{-3}]$, respectively. To better explore the high-dimensional parameter space, we run $N = 20$ parallel chains simultaneously in both DREAM$_{(ZS)}$ and DREAM$_{(KZS)}$. Here the chain lengths of DREAM$_{(ZS)}$ and DREAM$_{(KZS)}$ are set as 40,000 and 2000, which mean that the total numbers of model evaluations are 800,000 and 40,000, respectively. The probabilities of using the different jumps in DREAM$_{(ZS)}$ and DREAM$_{(KZS)}$ are the same as those in example 1. In DREAM$_{(KZS)}$, the "burn-in" period is set as the first 70% simulation time, which is much longer than that in example 1. It is expected that using

the Kalman jump over a longer period is beneficial to better estimate unknown model parameters in high-dimensional, nonlinear problems.

[Figure 6]

In Figure 6, we draw trace plots of the 8 contaminant source parameters obtained by DREAM$_{(ZS)}$ as well as the true values. It is found that after about 200,000 model evaluations, the variance of the samples in the Markov chains is greatly reduced, while the mean values are still slightly deviated from the true values. Then the Markov chains spend a very long time (about 600,000 model evaluations) approaching to the true values. In this simulation, DREAM$_{(ZS)}$ consumes 800,000 model evaluations, which imposes a very high computational burden. However, when we introduce the Kalman proposal distribution in DREAM$_{(ZS)}$, the convergence can be greatly accelerated. As shown in Figure 7, the chains of DREAM$_{(KZS)}$ approach very close to the true values of the contaminant source parameters within 30,000 model evaluations, which is less than 1/20 the computational cost needed by DREAM$_{(ZS)}$.

[Figure 7]

Using the last 10,000 samples in the Markov chains of DREAM$_{(ZS)}$ and DREAM$_{(KZS)}$, we can obtain the mean estimates of the *Y* field. From Figure 8, it is found that the estimated *Y* fields from both DREAM$_{(ZS)}$ and DREAM$_{(KZS)}$ can resemble the reference field to some extent. For a better estimation, we can collect more measurement data and use longer chains in the MCMC simulations. Inevitably, the cost in both data collection and computation will be increased.

[Figure 8]

3.3. Example 3: A 3D groundwater model

Here we further test the performance of DREAM$_{(KZS)}$ in a 3D groundwater model presented in [*Fienen et al.*, 2013]. This model has 3 layers, 40 rows and 35 columns. From top to bottom, the thicknesses of the 3 layers are 1.8 meters (*m*), 1.4 *m* and 1.8 *m*, respectively. The spacing for each row is 2.0 *m*, and the spacing for each column is 1.5

m. On all sides, the boundaries are constant with the head of 60 *m*. At row 18, column 17, there is a single well pumping water from each layer with a constant rate of 0.01 liter per minute. We assume that the *Y* (i.e., the log-transformed conductivity) fields in the three layers fit the correlation function defined in equation (12), where $\lambda_x = 37.5\,m$, $\lambda_y = 60\,m$, $\bar{Y}(\mathbf{x}) = -6.5$, and $\sigma_Y^2 = 0.5$. To reduce the parametric dimensionality, we use 40 KL terms to represent the *Y* field in each layer, which can preserve about 94% of the field variance. Then steady-state fields of hydraulic head are obtained by running the numerical solver of MODFLOW [*Harbaugh et al.*, 2000].

In this example, there are 120 unknown model parameters for the *Y* fields in the three layers. To reduce the parametric uncertainty, we collect 243 head measurements in the 3 layers at 81 wells and then assimilate them in the MCMC simulations. The 81 wells are located every 4 rows from row 3 to row 35, every 3 columns from column 5 to column 29. The measurement errors are assumed to be independent and Gaussian with zero mean and standard deviation of 0.01 *m*.

[Figure 9]

To infer the 120 unknown model parameters, we run both DREAM$_{(ZS)}$ and DREAM$_{(KZS)}$ with $N = 20$ parallel chains and a chain length of 4000. The probabilities of using the different jumps in DREAM$_{(ZS)}$ and DREAM$_{(KZS)}$ are the same as example 1. Using the last 10,000 samples in the Markov chains, we can obtain mean estimates of the *Y* fields in the 3 layers. As shown in Figure 9, both algorithms can capture the main patterns of the reference fields quite well. In Figure 10, we plot the root-mean-square error (RMSE) between the simulated model outputs in the chains and the measurements for both algorithms. It is again observed that introducing the Kalman proposal distribution can significantly speed up (about 20 times) the convergence of DREAM$_{(ZS)}$.

[Figure 10]

4. Conclusions and discussions

In this paper, we try to speed up the convergence of DREAM$_{(ZS)}$ by introducing a Kalman proposal distribution. In MCMC simulations, we use three jumps, i.e., the Kalman jump, the parallel direction jump and the snooker jump simultaneously with pre-defined probabilities to generate the proposal distributions. The new algorithm is thus called DREAM$_{(KZS)}$. As the Kalman jump cannot maintain detailed balance, we restrict it to the "burn-in" period. After that, we will use a mix of the parallel direction jump and the snooker jump to sample the posterior, which can satisfy detailed balance with diminishing adaptation.

Illustrated with three numerical cases, we find that DREAM$_{(KZS)}$ generally converges faster than DREAM$_{(ZS)}$. Specifically, in problems with about 100 unknown model parameters, the saving in computational budget can be as big as 20 times. It is because the Kalman jump utilizes the information contained in the covariance structure of the model parameters, the measurements and the model outputs, which can generate a more directional update of the model parameters than the other two jumps that only utilize the information about the model parameters. The Kalman filter (KF)-based methods, such as the ensemble Kalman filter and the ensemble smoother, have been adopted to solve high-dimensional inverse problems [*Chen and Zhang*, 2006; *Crestani et al.*, 2013; *Zhou et al.*, 2014]. However, they are not guaranteed to obtain a correct estimation of the posterior distribution. DREAM$_{(KZS)}$ combines the strengths of MCMC and the KF-based ensemble smoother while it remedies the weaknesses of the two methods. Thus, it enjoys both accuracy and efficiency in exploring the posterior distribution.

Nevertheless, there are many issues that deserve further consideration. For example, we adopt the backward jump of the Kalman proposal distribution to realize symmetry, which is rather crude. To address this issue, more theoretical development is needed. Moreover, we assume that the measurement errors fit Gaussian distribution. In practice, we usually have to consider non-Gaussian errors and infer the error statistics along with the model parameters [*Schoups and Vrugt*, 2010; *Vrugt et al.*, 2009a]. Another issue that should be concerned is the model structure error, which is also not considered in the present study. To address this issue, we can adopt the framework of

Bayesian model averaging [*Rojas et al.*, 2008; *Vrugt et al.*, 2008; *Ye et al.*, 2004], or quantify the model structure uncertainty together with other sources of uncertainties [*Xu and Valocchi*, 2015; *Xu et al.*, 2017a], etc. These issues will be addressed in our future work.

Acknowledgments

Computer codes and data used are available upon request to the corresponding author. This work is supported by the National Natural Science Foundation of China (Grants 41371237 and 41271470). G. Lin would like to acknowledge the support from National Science Foundation (DMS-1555072 and DMS-1736364).

References

Andrieu, C., N. De Freitas, A. Doucet, and M. I. Jordan (2003), An introduction to MCMC for machine learning, *Mach. Learn.*, *50*(1), 5-43, doi: 10.1023/A:1020281327116.

Bikowski, J., J. A. Huisman, J. A. Vrugt, H. Vereecken, and J. van der Kruk (2012), Integrated analysis of waveguide dispersed GPR pulses using deterministic and Bayesian inversion methods, *Near Surf. Geophys.*, *10*(6), 641-652, doi: 10.3997/1873-0604.2012041.

Chen, Y., and D. Zhang (2006), Data assimilation for transient flow in geologic formations via ensemble Kalman filter, *Adv. Water Resour.*, *29*(8), 1107-1122, doi: 10.1016/j.advwatres.2005.09.007.

Clark, M. P., D. Kavetski, and F. Fenicia (2011), Pursuing the method of multiple working hypotheses for hydrological modeling, *Water Resour. Res.*, *47*(9), W09301, doi: 10.1029/2010WR009827.

Crestani, E., M. Camporese, D. Baú, and P. Salandin (2013), Ensemble Kalman filter versus ensemble smoother for assessing hydraulic conductivity via tracer test data assimilation, *Hydrol. Earth Syst. Sci.*, *17*(4), 1517-1531, doi: 10.5194/hess-17-1517-2013.

Duan, Q., J. Schaake, V. Andreassian, S. Franks, G. Goteti, H. V. Gupta, Y. Gusev, F. Habets, A. Hall, and L. Hay (2006), Model Parameter Estimation Experiment (MOPEX): An overview of science strategy and major results from the second and third workshops, *J. Hydrol.* , *320*(1), 3-17, doi: 10.1016/j.jhydrol.2005.07.031.

Evensen, G. (2007), *Data assimilation: the ensemble Kalman filter*, Springer-Verlag, Berlin

Fienen, M. N., M. D'Oria, J. E. Doherty, and R. J. Hunt (2013), Approaches in highly parameterized inversion: bgaPEST, a Bayesian geostatistical approach implementation with PEST: documentation and instructions*.* 69 pp, U.S. Geol. Surv., Reston, Va.

Gelman, A., and D. B. Rubin (1992), Inference from iterative simulation using multiple sequences, *Stat. Sci.*, *7*, 457-472.

Harbaugh, A. W., E. R. Banta, M. C. Hill, and M. G. McDonald (2000), MODFLOW-2000, The U. S. Geological Survey modular ground-water model-user guide to modularization concepts and the ground-water flow process*.* U.S. Geol. Surv., Reston, Va. [Available at https://pubs.usgs.gov/of/2000/0092/report.pdf].

Hastings, W. K. (1970), Monte Carlo sampling methods using Markov chains and their applications, *Biometrika*, *57*(1), 97-109, doi: 10.1093/biomet/57.1.97.

Keating, E. H., J. Doherty, J. A. Vrugt, and Q. Kang (2010), Optimization and uncertainty assessment of strongly nonlinear groundwater models with high parameter dimensionality, *Water Resour. Res.*, *46*(10), W10517, doi: 10.1029/2009WR008584.

Laloy, E., and J. A. Vrugt (2012), High-dimensional posterior exploration of hydrologic models using multiple-try DREAM(ZS) and high-performance computing, *Water Resour. Res.*, *48*(1), W01526, doi: 10.1029/2011WR010608.

Laloy, E., B. Rogiers, J. A. Vrugt, D. Mallants, and D. Jacques (2013), Efficient posterior exploration of a high-dimensional groundwater model from two-stage Markov chain Monte Carlo simulation and polynomial chaos expansion, *Water Resour. Res.*, *49*(5), 2664-2682, doi: 10.1002/wrcr.20226.

Linde, N., and J. A. Vrugt (2013), Distributed soil moisture from crosshole ground-penetrating radar travel times using stochastic inversion, *Vadose Zone J.*, *12*(1), doi: 10.2136/vzj2012.0101.

Lochbühler, T., S. J. Breen, R. L. Detwiler, J. A. Vrugt, and N. Linde (2014), Probabilistic electrical resistivity tomography of a CO2 sequestration analog, *J. Appl. Geophys.*, *107*, 80-92, doi:

10.1016/j.jappgeo.2014.05.013.
Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953), Equation of state calculations by fast computing machines, *J. Chem. Phys.*, *21*(6), 1087-1092, doi: 10.1063/1.1699114.
Muleta, M. K., J. McMillan, G. G. Amenu, and S. J. Burian (2012), Bayesian approach for uncertainty analysis of an urban storm water model and its application to a heavily urbanized watershed, *J. Hydrol. Eng.*, *18*(10), 1360-1371, doi: 10.1061/(ASCE)HE.1943-5584.0000705.
Ramin, E., D. S. Wágner, L. Yde, P. J. Binning, M. R. Rasmussen, P. S. Mikkelsen, and B. G. Plósz (2014), A new settling velocity model to describe secondary sedimentation, *Water Res.*, *66*, 447-458, doi: 10.1016/j.watres.2014.08.034.
Refsgaard, J. C., S. Christensen, T. O. Sonnenborg, D. Seifert, A. L. Højberg, and L. Troldborg (2012), Review of strategies for handling geological uncertainty in groundwater flow and transport modeling, *Adv. Water Resour.*, *36*, 36-50, doi: 10.1016/j.advwatres.2011.04.006.
Roberts, G. O., and J. S. Rosenthal (2007), Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms, *J. Appl. Probab.*, *44*(2), 458-475, doi: 10.1017/S0021900200117954.
Rojas, R., L. Feyen, and A. Dassargues (2008), Conceptual model uncertainty in groundwater modeling: Combining generalized likelihood uncertainty estimation and Bayesian model averaging, *Water Resour. Res.*, *44*(12), W12418, doi: 10.1029/2008WR006908.
Sadegh, M., and J. A. Vrugt (2014), Approximate bayesian computation using Markov chain Monte Carlo simulation: DREAM(ABC), *Water Resour. Res.*, *50*(8), 6767-6787, doi: 10.1002/2014WR015386.
Schnoor, J. L. (1996), *Environmental modeling: fate and transport of pollutants in water, air, and soil*, John Wiley and Sons, N.Y.
Schoups, G., and J. A. Vrugt (2010), A formal likelihood function for parameter and predictive inference of hydrologic models with correlated, heteroscedastic, and non-Gaussian errors, *Water Resour. Res.*, *46*(10), W10531, doi: 10.1029/2009WR008933.
Shi, X., M. Ye, G. P. Curtis, G. L. Miller, P. D. Meyer, M. Kohler, S. Yabusaki, and J. Wu (2014), Assessment of parametric uncertainty for groundwater reactive transport modeling, *Water Resour. Res.*, *50*(5), 4416-4439, doi: 10.1002/2013WR013755.
ter Braak, C. J. (2006), A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces, *Stat. Comput.*, *16*(3), 239-249, doi: 10.1007/s11222-006-8769-1.
ter Braak, C. J., and J. A. Vrugt (2008), Differential evolution Markov chain with snooker updater and fewer chains, *Stat. Comput.*, *18*(4), 435-446, doi: 10.1007/s11222-008-9104-9.
Vereecken, H., et al. (2016), Modeling soil processes: Review, key challenges, and new perspectives, *Vadose Zone J.*, *15*(5), doi: 10.2136/vzj2015.09.0131.
Vrugt, J. A. (2016), Markov chain Monte Carlo simulation using the DREAM software package: Theory, concepts, and MATLAB implementation, *Environ. Modell. Softw.*, *75*, 273-316, doi: 10.1016/j.envsoft.2015.08.013.
Vrugt, J. A., and C. Ter Braak (2011), DREAM(D): an adaptive Markov Chain Monte Carlo simulation algorithm to solve discrete, noncontinuous, and combinatorial posterior parameter estimation problems, *Hydrol. Earth Syst. Sci.*, *15*(12), 3701-3713, doi: 10.5194/hess-15-3701-2011.
Vrugt, J. A., C. G. Diks, and M. P. Clark (2008), Ensemble Bayesian model averaging using Markov chain Monte Carlo sampling, *Environ. Fluid Mech.*, *8*(5-6), 579-595, doi: 10.1007/s10652-008-9106-

3.
Vrugt, J. A., C. J. Ter Braak, H. V. Gupta, and B. A. Robinson (2009a), Equifinality of formal (DREAM) and informal (GLUE) Bayesian approaches in hydrologic modeling?, *Stoch. Env. Res. Risk A.*, *23*(7), 1011-1026, doi: 10.1007/s00477-008-0274-y.
Vrugt, J. A., C. J. ter Braak, C. G. Diks, and G. Schoups (2013), Hydrologic data assimilation using particle Markov chain Monte Carlo simulation: Theory, concepts and applications, *Adv. Water Resour.*, *51*, 457-478, doi: 10.1016/j.advwatres.2012.04.002.
Vrugt, J. A., C. Ter Braak, C. Diks, B. A. Robinson, J. M. Hyman, and D. Higdon (2009b), Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling, *Int. J. Nonlinear Sci. Numer. Simul.* , *10*(3), 273-290, doi: 10.1515/IJNSNS.2009.10.3.273.
Wöhling, T., and J. A. Vrugt (2011), Multiresponse multilayer vadose zone model calibration using Markov chain Monte Carlo simulation and field water retention data, *Water Resour. Res.*, *47*(4), W04510, doi: 10.1029/2010WR009265.
Wagener, T., and H. V. Gupta (2005), Model identification for hydrological forecasting under uncertainty, *Stoch. Environ. Res. Risk Assess.*, *19*(6), 378-387, doi: 10.1007/s00477-005-0006-5.
Xu, T., and A. J. Valocchi (2015), A Bayesian approach to improved calibration and prediction of groundwater models with structural error, *Water Resour. Res.*, *51*(11), 9290-9311, doi: 10.1002/2015WR017912.
Xu, T., A. J. Valocchi, M. Ye, and F. Liang (2017a), Quantifying model structural error: Efficient Bayesian calibration of a regional groundwater flow model using surrogates and a data-driven error model, *Water Resour. Res.*, *53*(5), 4084-4105, doi: 10.1002/2016WR019831.
Xu, T., A. J. Valocchi, M. Ye, F. Liang, and Y. F. Lin (2017b), Bayesian calibration of groundwater models with input data uncertainty, *Water Resour. Res.*, *53*(4), 3224-3245, doi: 10.1002/2016WR019512.
Ye, M., S. P. Neuman, and P. D. Meyer (2004), Maximum likelihood Bayesian averaging of spatial variability models in unsaturated fractured tuff, *Water Resour. Res.*, *40*(5), W05113, doi: 10.1029/2003WR002557.
Ying, S., J. Zhang, L. Zeng, J. Shi, and L. Wu (2017), Bayesian inference for kinetic models of biotransformation using a generalized rate equation, *Sci. Total. Environ.*, *590-591*, 287-296, doi: 10.1016/j.scitotenv.2017.03.003.
Zeng, X., J. Wu, D. Wang, and X. Zhu (2016), Assessing the pollution risk of a groundwater source field at western Laizhou Bay under seawater intrusion, *Environ. Res.*, *148*, 586-594, doi: 10.1016/j.envres.2015.11.022.
Zhang, D., and Z. Lu (2004), An efficient, high-order perturbation approach for flow in random porous media via Karhunen-Loève and polynomial expansions, *J. Comput. Phys.*, *194*(2), 773-794, doi: 10.1016/j.jcp.2003.09.015.
Zhang, G., D. Lu, M. Ye, M. Gunzburger, and C. Webster (2013), An adaptive sparse-grid high-order stochastic collocation method for Bayesian inference in groundwater reactive transport modeling, *Water Resour. Res.*, *49*(10), 6871-6892, doi: 10.1002/wrcr.20467.
Zheng, C., and P. P. Wang (1999), MT3DMS: A modular three-dimensional multispecies transport model for simulation of advection, dispersion, and chemical reactions of contaminants in groundwater systems; documentation and user's guide, Contract Rep. SERDP-99-1, U.S. Army Eng. Res. and Dev. Cent., Vicksburg, Miss. [Available at

http://www.geology.wisc.edu/courses/g727/mt3dmanual.pdf].
Zhou, H., J. J. Gómez-Hernández, and L. Li (2014), Inverse methods in hydrogeology: Evolution and recent trends, *Adv. Water Resour.*, *63*, 22-37, doi: 10.1016/j.advwatres.2013.10.014.

Tables

Table 1. Prior ranges and true values of the 7 parameters in the hmodel

Parameter [*unit*]	Symbol	Prior range	True value
Maximum interception [*mm*]	$I_{\max}$	[0.5 10]	3.601
Soil water storage capacity [*mm*]	$S_{\max}$	[10 1000]	704.566
Maximum percolation rate [*mm* d^{-1}]	$Q_{\max}$	[0 100]	99.124
Evaporation parameter [-]	α_{E}	[1×10^{-6} 100]	43.568
Runoff parameter [-]	α_{F}	[-10 10]	4.161
Time constant, fast reservoir [*days*]	K_{F}	[0 10]	9.051
Time constant, slow reservoir [*days*]	K_{S}	[0 150]	14.780

Table 2. Prior ranges and true values of the contaminant source parameters

Parameter [*unit*]	Prior range	True value
x_s [L]	[3 5]	3.520
y_s [L]	[4 6]	4.437
S_1 [MT^{-1}]	[0 8]	5.692
S_2 [MT^{-1}]	[0 8]	7.883
S_3 [MT^{-1}]	[0 8]	6.306
S_4 [MT^{-1}]	[0 8]	1.485
S_5 [MT^{-1}]	[0 8]	6.872
S_6 [MT^{-1}]	[0 8]	5.552

Figures

Figure 1. Evolution of the $\hat{R}$-statistics of the 7 model parameters in (a) DREAM$_{(ZS)}$ and (b) DREAM$_{(KZS)}$. The threshold of 1.2 for convergence diagnosis is represented by the red dashed lines.

Figure 2. Marginal posterior pdfs of the 7 model parameters obtained by DREAM$_{(ZS)}$ (red curves) and DREAM$_{(KZS)}$ (blue dashed curves), respectively. The true values of the model parameters are represented by the black vertical lines.

Figure 3. Given 4 different sets of measurement data generated from 4 different sets of true model parameters, the multivariate $\hat{R}$-statistics obtained by DREAM$_{(ZS)}$ (magenta curves) and DREAM$_{(KZS)}$ (blue curves). In each subplot, the 4 curves with the same color signify 4 repeated simulations with DREAM$_{(ZS)}$ or DREAM$_{(KZS)}$ given the same set of measurement data.

Figure 4. Marginal posterior pdfs of the 7 model parameters and the 2 parameters for the measurement error distribution obtained by DREAM$_{(KZS)}$ (blue curves). The true values of the two types of parameters are represented by the black vertical lines.

Figure 5. Flow domain of example 2.

Figure 6. Trace plots of contaminant source parameters obtained by DREAM$_{(ZS)}$. Here the true values are represented by the red crosses.

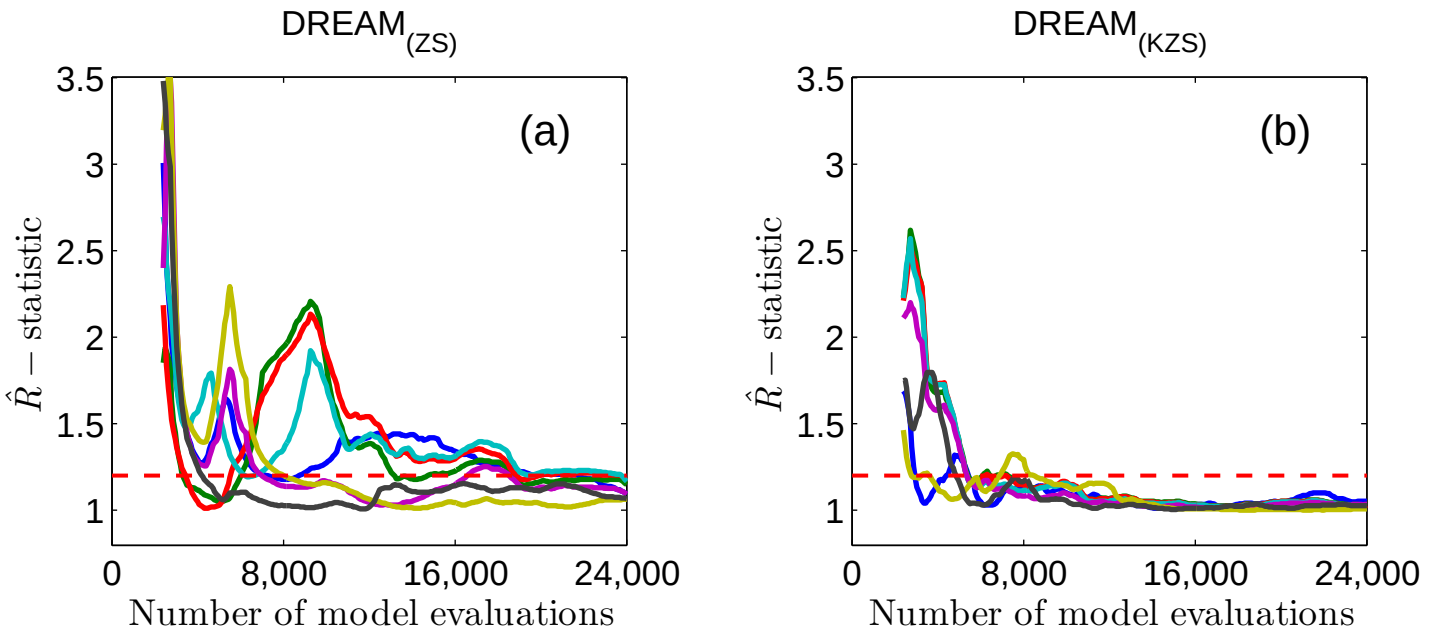
Figure 7. Trace plots of contaminant source parameters obtained by DREAM$_{(KZS)}$. Here the true values are represented by the red crosses.

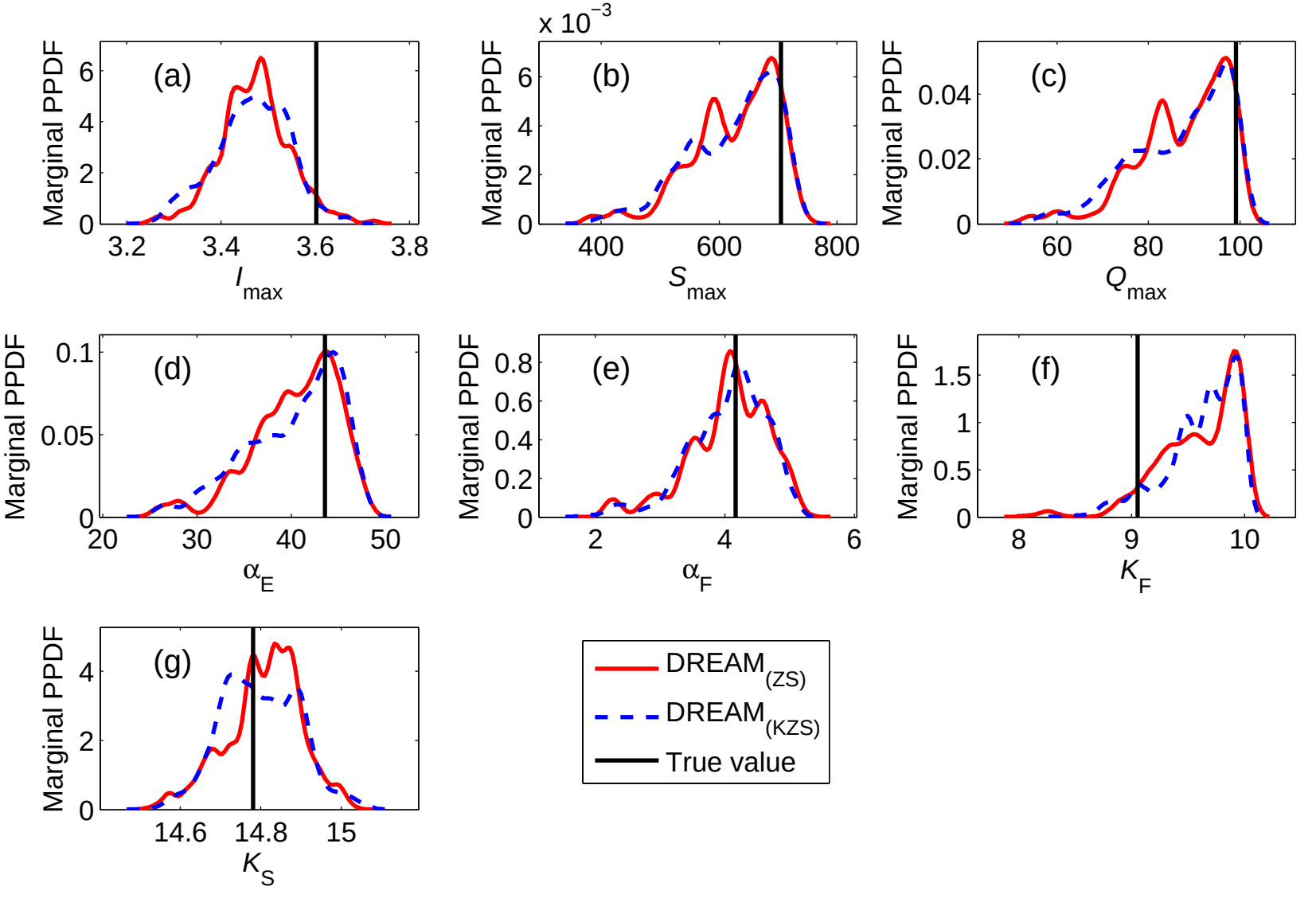
Figure 8. (a) The reference *Y* field and the mean estimates of the *Y* field obtained by (b) DREAM$_{(ZS)}$ and (c) DREAM$_{(KZS)}$.

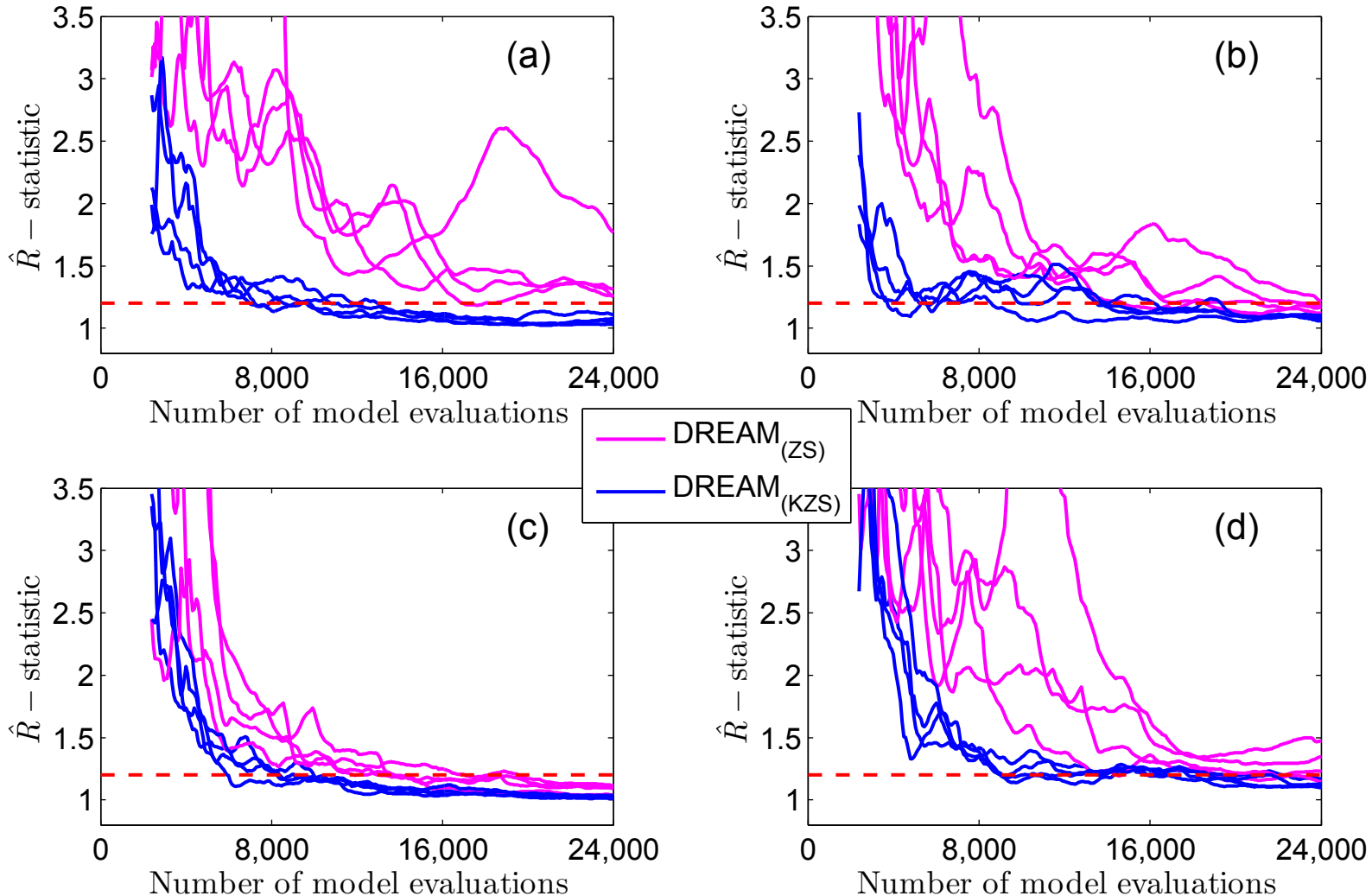
Figure 9. The first column: the reference *Y* fields in the 3 layers; The second column: the estimated *Y* fields obtained by DREAM$_{(ZS)}$; The third column: the estimated *Y* fields obtained by DREAM$_{(KZS)}$.

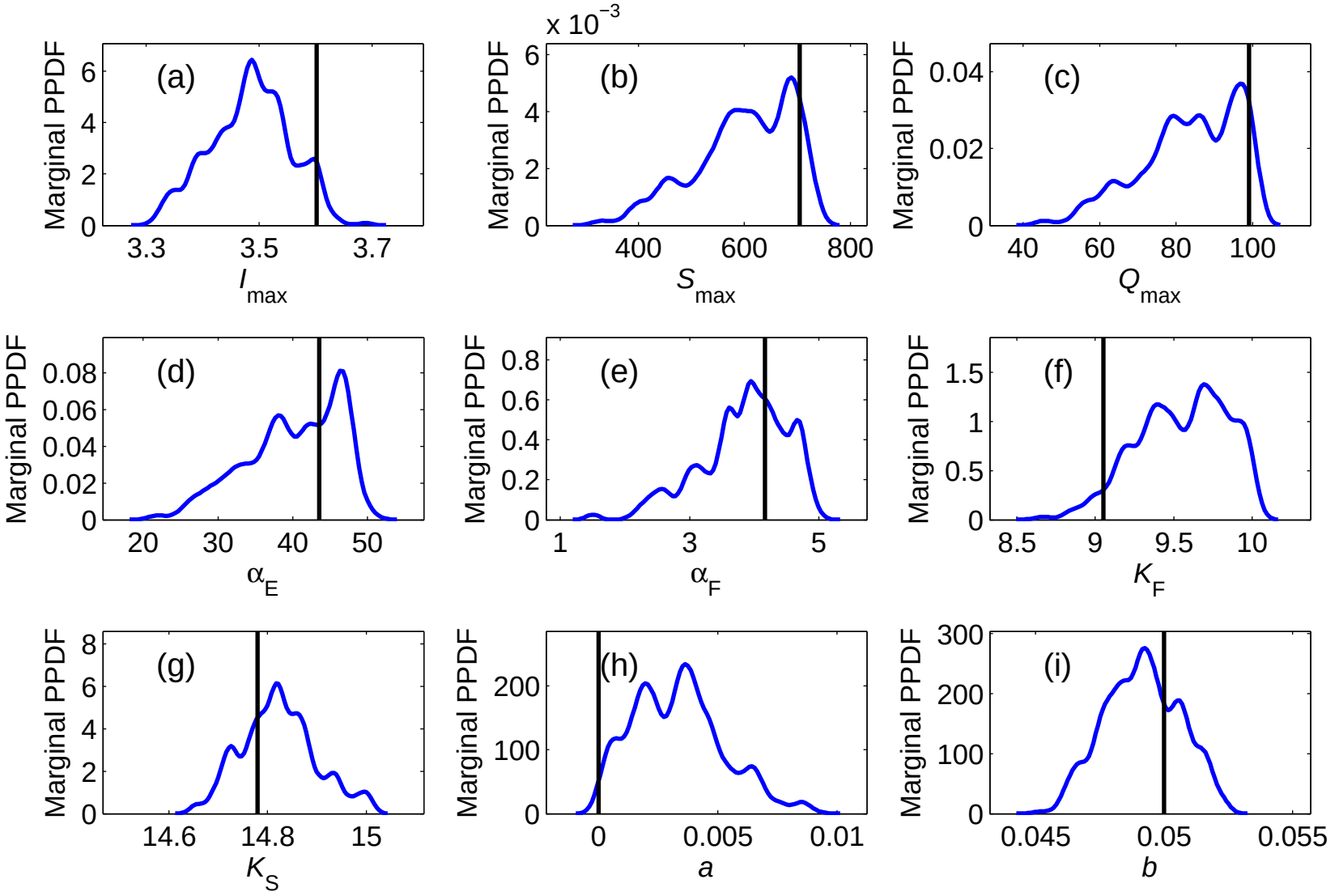
Figure 10. RMSE values between the simulated model outputs in the Markov chains

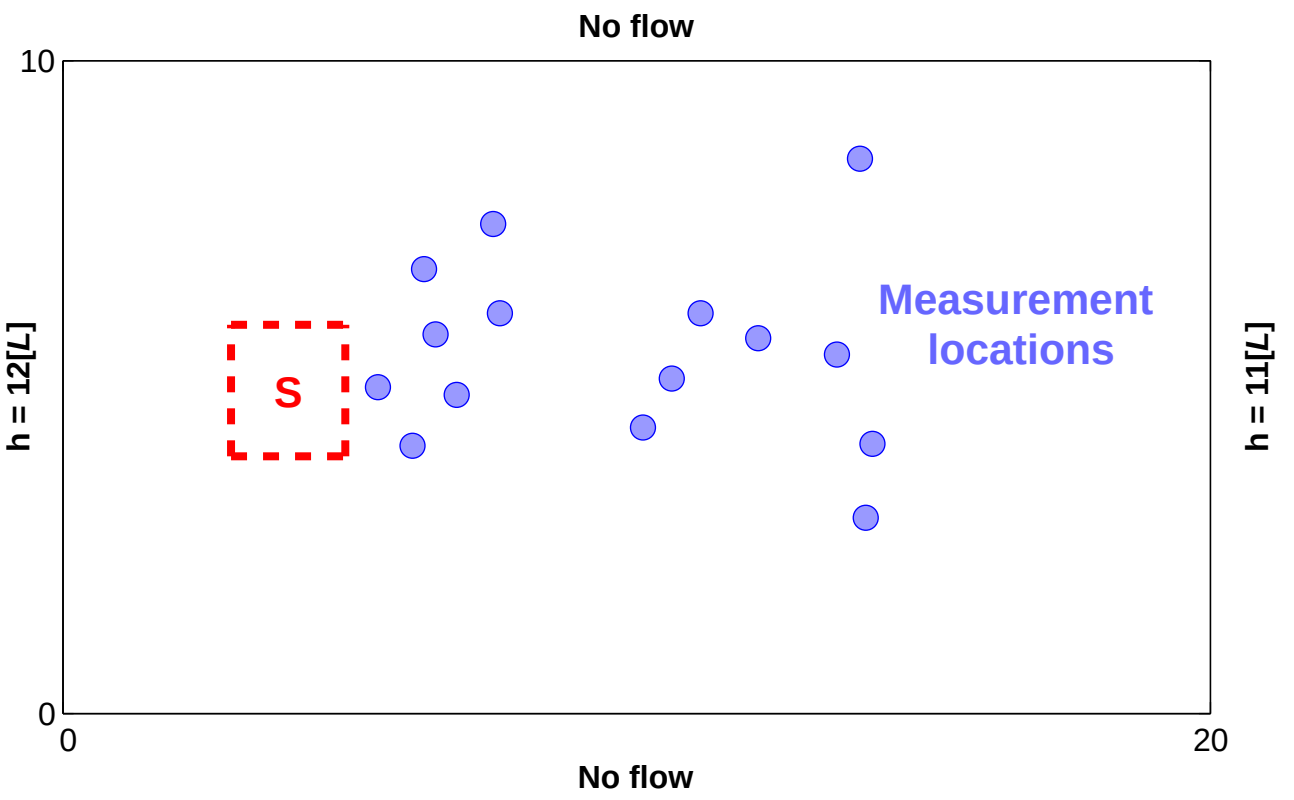
and the measurements. Here the blue dots are for the results obtained by DREAM$_{(ZS)}$, and the red dots are for the results obtained by DREAM$_{(KZS)}$.

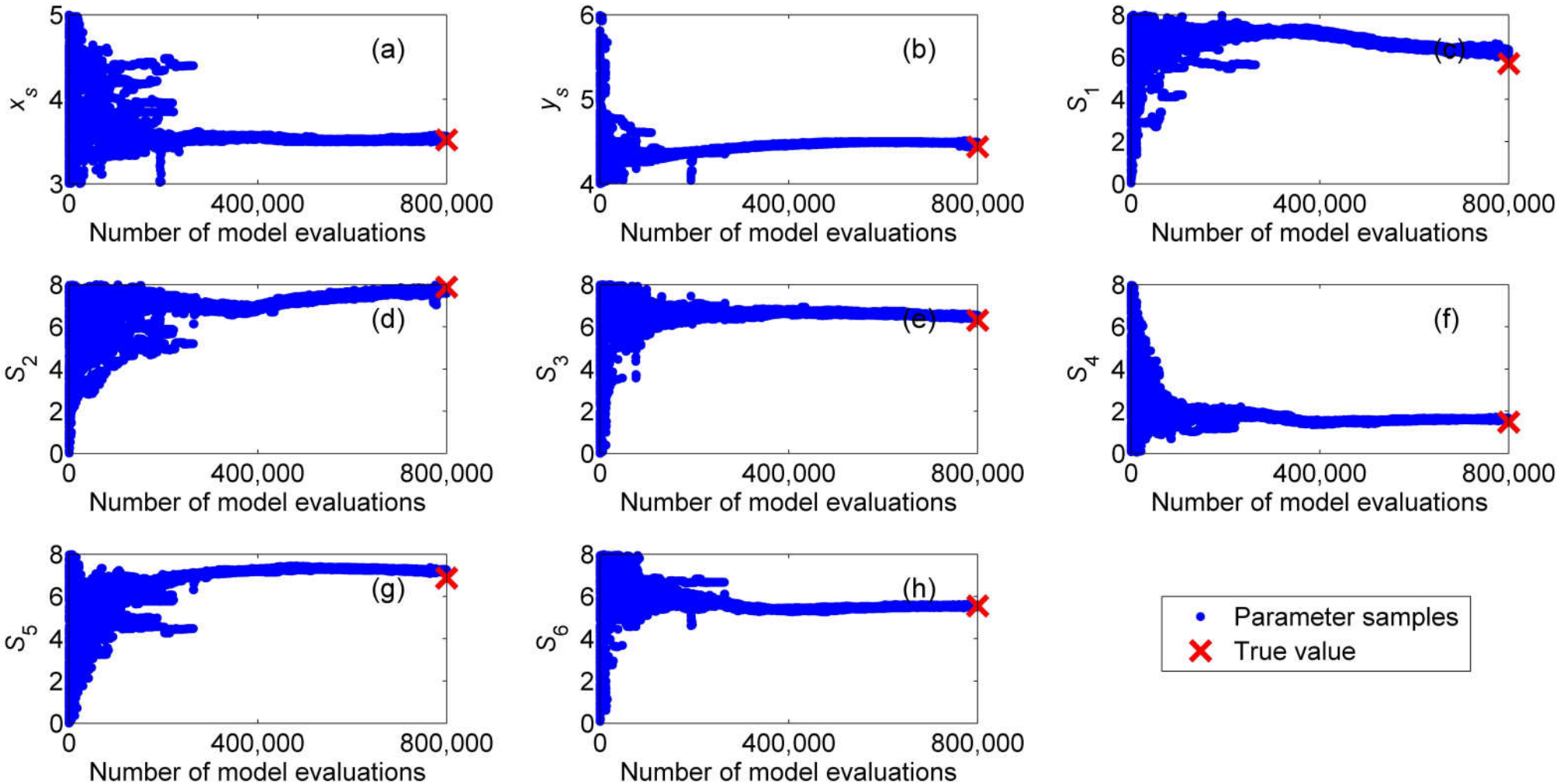

DREAM(ZS)
(a)
$\hat{R}$ − statistic
3.5
3
2.5
2
1.5
1
0
8,000
16,000
24,000
Number of model evaluations
DREAM(KZS)
(b)
$\hat{R}$ − statistic
3.5
3
2.5
2
1.5
1
0
8,000
16,000
24,000
Number of model evaluations

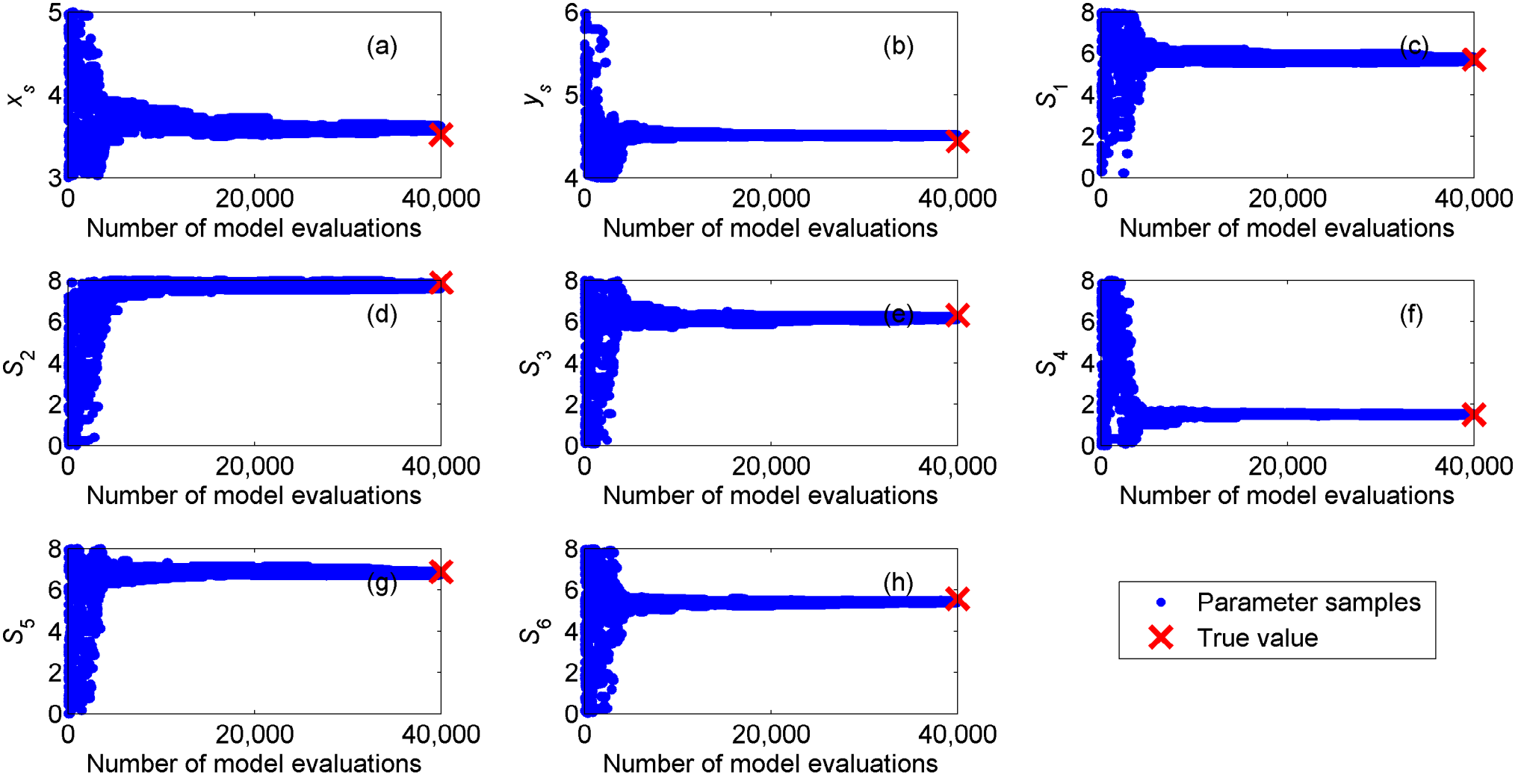

(a)
(b)
(c)
(d)
(e)
(f)
(g)
Marginal PPDF
x 10^-3
I_{max}
S_{max}
Q_{max}
α_E
α_F
K_F
K_S
DREAM(ZS)
DREAM(KZS)
True value

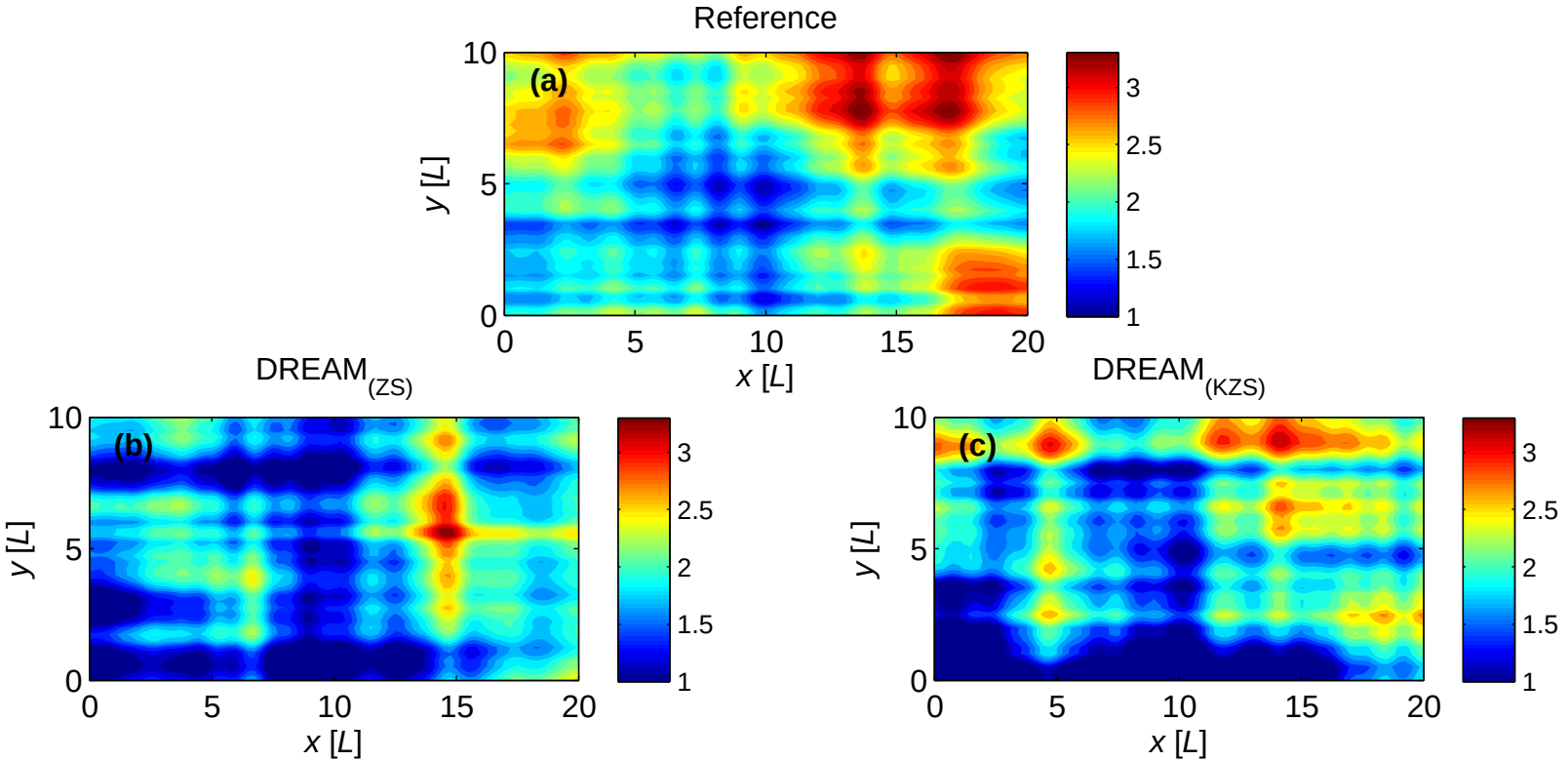

(a)
(b)
(c)
(d)
$\hat{R}$ − statistic
Number of model evaluations
3.5
3
2.5
2
1.5
1
0
8,000
16,000
24,000
DREAM(ZS)
DREAM(KZS)

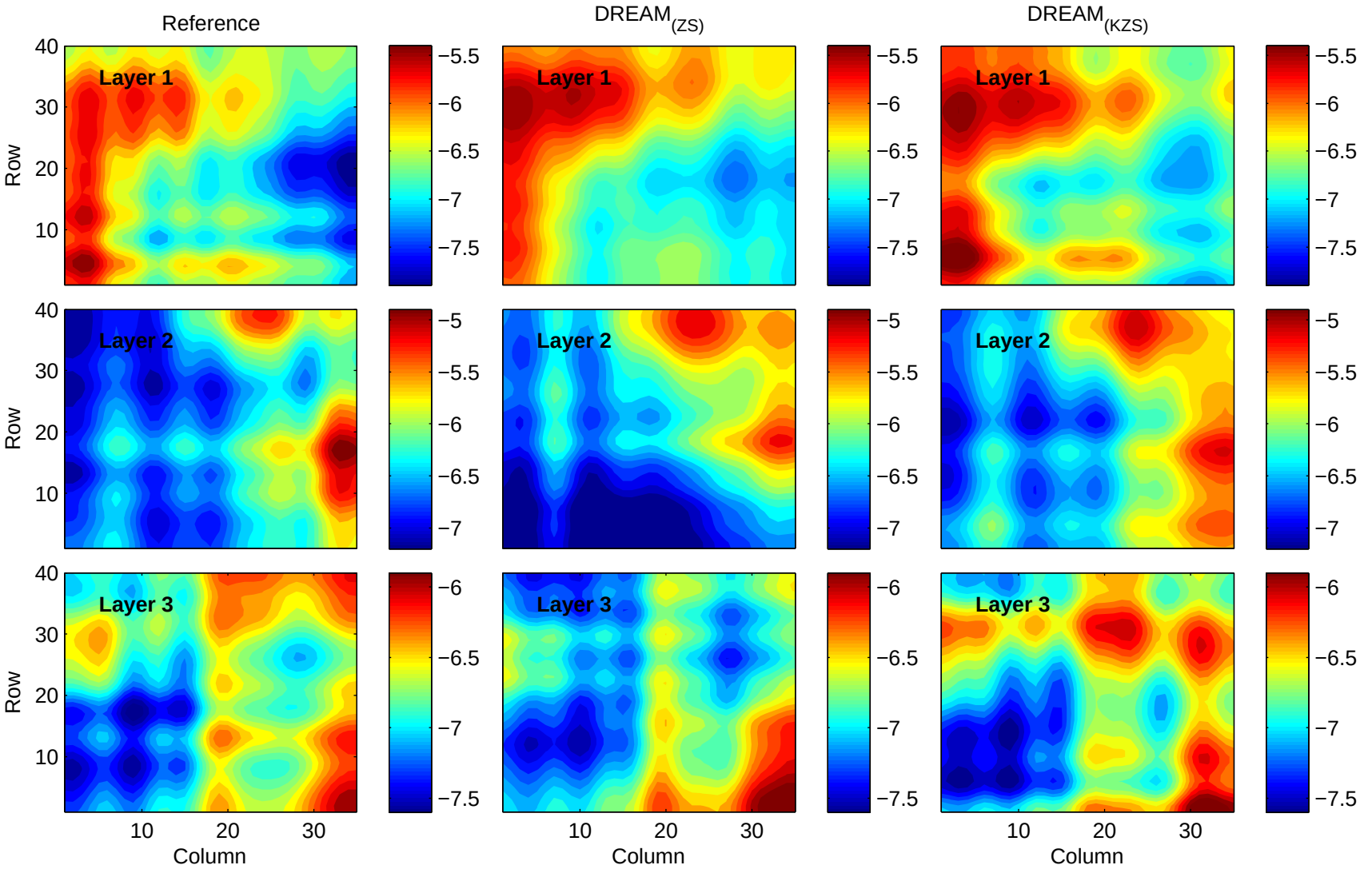

(a)
Marginal PPDF
I_{max}
(b)
x 10^{-3}
S_{max}
(c)
Q_{max}
(d)
α_E
(e)
α_F
(f)
K_F
(g)
K_S
(h)
a
(i)
b

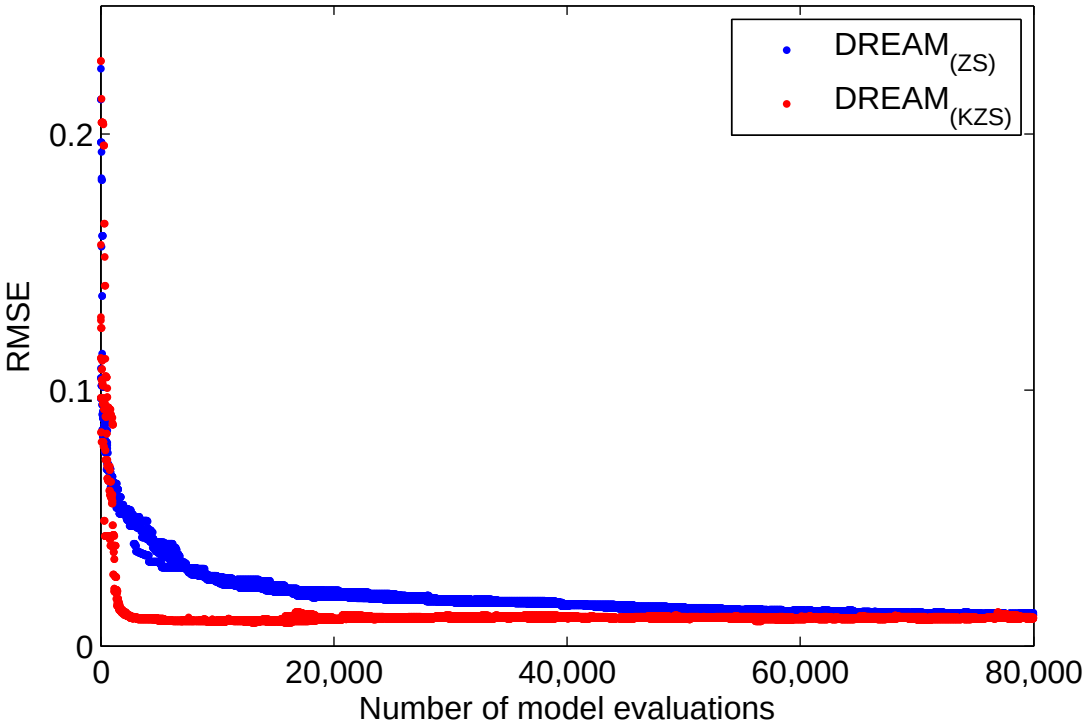

No flow
10
h = 12[L]
S
Measurement locations
h = 11[L]
0
0
20
No flow


(a)
x_s
(b)
y_s
(c)
S_1
(d)
S_2
(e)
S_3
(f)
S_4
(g)
S_5
(h)
S_6
0
400,000
800,000
Number of model evaluations
Parameter samples
True value


(a)
x_s
(b)
y_s
(c)
S_1
(d)
S_2
(e)
S_3
(f)
S_4
(g)
S_5
(h)
S_6
0
20,000
40,000
Number of model evaluations
Parameter samples
True value


Reference
(a)
y [L]
x [L]
DREAM(ZS)
(b)
DREAM(KZS)
(c)


Reference
DREAM(ZS)
DREAM(KZS)
Layer 1
Layer 2
Layer 3
Row
Column
40
30
20
10
−5.5
−6
−6.5
−7
−7.5
−5


DREAM(ZS)
DREAM(KZS)
RMSE
0.2
0.1
0
0
20,000
40,000
60,000
80,000
Number of model evaluations