Surrogate-assisted Bayesian inversion for landscape and basin evolution models

Rohitash Chandra\textsuperscript{1,2}, Danial Azam\textsuperscript{2}, Arpit Kapoor\textsuperscript{3}, and R. Dietmar Müller\textsuperscript{2}

\textsuperscript{1}School of Mathematics and Statistics, University of New South Wales, Sydney, NSW 2052, Australia
\textsuperscript{2}EarthByte Group, School of Geosciences, University of Sydney, Sydney, NSW 2006, Australia
\textsuperscript{3}Department of Computer Science and Engineering, SRM Institute of Science and Technology, Tamil Nadu, India

Correspondence: Rohitash Chandra (rohitash.chandra@unsw.edu.au)

Received: 9 December 2018 – Discussion started: 20 February 2019
Revised: 30 March 2020 – Accepted: 6 June 2020 – Published: 8 July 2020

Abstract. The complex and computationally expensive nature of landscape evolution models poses significant challenges to the inference and optimization of unknown model parameters. Bayesian inference provides a methodology for estimation and uncertainty quantification of unknown model parameters. In our previous work, we developed parallel tempering Bayeslands as a framework for parameter estimation and uncertainty quantification for the Badlands landscape evolution model. Parallel tempering Bayeslands features high-performance computing that can feature dozens of processing cores running in parallel to enhance computational efficiency. Nevertheless, the procedure remains computationally challenging since thousands of samples need to be drawn and evaluated. In large-scale landscape evolution problems, a single model evaluation can take from several minutes to hours and in some instances, even days or weeks. Surrogate-assisted optimization has been used for several computationally expensive engineering problems which motivate its use in optimization and inference of complex geoscientific models. The use of surrogate models can speed up parallel tempering Bayeslands by developing computationally inexpensive models to mimic expensive ones. In this paper, we apply surrogate-assisted parallel tempering where the surrogate mimics a landscape evolution model by estimating the likelihood function from the model. We employ a neural-network-based surrogate model that learns from the history of samples generated. The entire framework is developed in a parallel computing infrastructure to take advantage of parallelism. The results show that the proposed methodology is effective in lowering the computational cost significantly while retaining the quality of model predictions.

1 Introduction

The Bayesian methodology provides a probabilistic approach for the estimation of unknown parameters in complex models (Sambridge, 1999; Neal, 1996; Chandra et al., 2019b). We can view a deterministic geophysical forward model as a probabilistic model via Bayesian inference, which is also known as Bayesian inversion, which has been used for landscape evolution (Chandra et al., 2019a, c), geological reef evolution models (Pall et al., 2020), and other geoscientific models (Sambridge, 1999, 2013; Scalzo et al., 2019; Olierook et al., 2020). Markov chain Monte Carlo (MCMC) sampling is typically used to implement Bayesian inference that involves the estimation and uncertainty quantification of unknown parameters (Hastings, 1970; Metropolis et al., 1953; Neal, 2012, 1996). Parallel tempering MCMC (Marinari and Parisi, 1992; Geyer and Thompson, 1995) features multiple replicas to provide a balance between exploration and exploitation, which makes them suitable for irregular and multimodal distributions (Patriksson and van der Spoel, 2008; Hukushima and Nemoto, 1996). In contrast to canonical sampling methods, we can implement parallel tempering more easily in a parallel computing architecture (Lamport, 1986).

Our previous work presented parallel tempering Bayeslands for parameter estimation and uncertainty quantification for landscape evolution models (LEMs) (Chandra et al., 2019c). Parallel tempering Bayeslands features parallel computing to enhance computational efficiency of inference for the Badlands LEM. Although we used parallel computing, the procedure was computationally challenging since thousands of samples were drawn and
evaluated (Chandra et al., 2019c). In large-scale LEMs, running a single model can take several hours, to days or weeks, and usually thousands of model runs are required for inference of unknown model parameters. Hence, it is important to enhance parallel tempering Bayeslands, which can also be applicable for other complex geoscientific models. One of the ways to address this problem is through surrogate-assisted estimation.

Surrogate-assisted optimization refers to the use of statistical and machine learning models for developing approximate simulation or surrogate of the actual model (Jin, 2011). Since typically optimization methods lack a rigorous approach for uncertainty quantification, Bayesian inversion becomes as an alternative choice particularly for complex geophysical numerical models (Sambridge, 2013, 1999). The major advantage of a surrogate model is its computational efficiency when compared to the equivalent numerical physical forward model (Ong et al., 2003; Zhou et al., 2007). In the optimization literature, surrogate utilization is also known as response surface methodology (Montgomery and Vernong M. Bettencourt, 1977; Letisinger et al., 1996) and applicable for a wide range of engineering problems (Tandjiria et al., 2000; Ong et al., 2005) such as aerodynamic wing design (Ong et al., 2003). Several approaches have been used to improve the way surrogates are utilized. Zhou et al. (2007) combined global and local surrogate models to accelerate evolutionary optimization. Lim et al. (2010) presented a generalized surrogate-assisted evolutionary computation framework to unify diverse surrogate models during optimization and taking into account uncertainty in estimation. Jin (2011) reviewed a range of problems such as single, multi-objective, dynamic, constrained, and multimodal optimization problems (Díaz-Manriquez et al., 2016). In the Earth sciences, examples for surrogate-assisted approaches include modelling water resources (Razavi et al., 2012; Asher et al., 2015), atmospheric general circulation models (Scher, 2018), computational oceanography (van der Merwe et al., 2007), carbon-dioxide (CO₂) storage and oil recovery (Ampomah et al., 2017), and debris flow models (Navarro et al., 2018).

Given that Bayeslands is implemented using parallel computing, the challenge is in implementing surrogates across different processing cores. Recently, we developed surrogate-assisted parallel tempering for Bayesian neural networks, which used a global–local surrogate framework to execute surrogate training in the master processing core that manages the replicas running in parallel (Chandra et al., 2020). The global surrogate refers to the main surrogate model that features training data combined from different replicas running in parallel cores. Local surrogate model refers to the surrogate model in the given replica that incorporates knowledge from the global surrogate to make a prediction given new input parameters. Note that the training only takes place in the global surrogate, and the prediction or estimation for pseudo-likelihood only takes place in the local surrogates. The method gives promising results where prediction performance is maintained while lowering computational time using surrogates.

In this paper, we present an application of surrogate-assisted parallel tempering (Chandra et al., 2020) for Bayesian inversion of LEMs using parallel computing infrastructure. We use the Badlands LEM model (Salles et al., 2018) as a case study to demonstrate the framework. Overall, the framework features the surrogate model, which mimics the Badlands model and estimates the likelihood function to evaluate the proposed parameters. We employ a neural network model as the surrogate that learns from the history of samples from the parallel tempering MCMC. We apply the method to several selected benchmark landscape evolution and sediment transport/deposition problems and show the quality of the estimation of the likelihood given by the surrogate when compared to the actual Badlands model.

2 Background and related work

2.1 Bayesian inference

Bayesian inference is typically implemented by employing MCMC sampling methods that update the probability for a hypothesis as more information becomes available. The hypothesis is given by a prior probability distribution (also known as the prior) that expresses one’s belief about a quantity (or free parameter in a model) before some data are taken into account. Therefore, MCMC methods provide a probabilistic approach for estimation of free parameters in a wide range of models (Kass et al., 1998; van Ravenzwaaij et al., 2016). The likelihood function is a way to evaluate the sampled parameters for a model with given observed data. In order to evaluate the likelihood function, one would need to run the given model, which in our case is the Badlands model. The likelihood function is used with the Metropolis criteria to either accept or reject a proposal. When accepted, the proposal becomes part of the posterior distribution, which essentially provides the estimation of the free parameter with uncertainties. The sampling process is iterative and requires that thousands of samples are drawn until convergence. In our case, convergence is defined by a predefined number of samples or until the likelihood function has reached a specific value.

2.2 Badlands model and Bayeslands framework

LEMs incorporate different driving forces such as tectonics or climate variability (Whipple and Tucker, 2002; Tucker and Hancock, 2010; Salles et al., 2018; Campforts et al., 2017; Adams et al., 2017) and combine empirical data and conceptual methods into a set of mathematical equations. Badlands (basin and landscape dynamics) (Salles et al., 2018; Salles and Hardiman, 2016) is an example of such a model that can be used to reconstruct landscape evolution and associated sediment fluxes (Howard et al., 1994; Hobley et al., 2011).
Figure 1. Location of (a) Continental-Margin problem shown taken from Te Waipounamu/South Island of Aotearoa/New Zealand. (b) Tasmania, Australia, with latitude and longitude information shown in degrees.

Table 1. In the given landscape evolution problems, the run time represents approximately the duration for one model to run on a single CPU. The length and width are given in kilometres (km), which are represented by the specified number of points (pts) as defined by the resolution (Res.) factor.

<table>
<thead>
<tr>
<th>Topography</th>
<th>Evo. Length</th>
<th>Width</th>
<th>Res.</th>
<th>Run-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continental-Margin</td>
<td>1 000 000</td>
<td>136.0</td>
<td>123.0</td>
<td>3.0</td>
</tr>
<tr>
<td>Synthetic-Mountain</td>
<td>1 000 000</td>
<td>202.0</td>
<td>102.0</td>
<td>5.0</td>
</tr>
<tr>
<td>Tasmania</td>
<td>1 000 000</td>
<td>523.0</td>
<td>554.0</td>
<td>71.3</td>
</tr>
</tbody>
</table>

Baylands LEM model (Salles et al., 2018) simulates landscape evolution and sediment transport/deposition with given parameters such as the precipitation rate and rock erodibility coefficient. The Badlands LEM simulates landscape dynamics, which requires an initial topography exposed to climate and geological factors over time.

Bayeslands essentially provides the estimation of unknown Badlands parameters with Bayesian inference via MCMC sampling (Chandra et al., 2019c). We use the final or present-day topography at time $T$ and expected sediment deposits at selected intervals to evaluate the quality of proposals during sampling. In this way, we constrain the set of unknown parameters ($\theta$) using ground-truth data ($D$). The prior distribution (also known as prior) refers to one’s belief in the distribution of the parameter without taking into account the evidence or data. Bayeslands estimates $\theta$ so that the simulated topography by Badlands can resemble the ground-truth topography $D$ to some degree. Bayeslands samples the posterior distribution $p(\theta|D)$ using principles of Bayes’ rule

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{P(D)},$$

where, $p(D|\theta)$ is the likelihood of the data given the parameters, $p(\theta)$ is the prior, and $P(D)$ is a normalizing constant and equal to $\int p(D|\theta)p(\theta)d\theta$. We note that the prior ratio cancels out since we use a uniform distribution for the priors.

3 Methodology

3.1 Benchmark landscape evolution problems

We select two benchmark landscape problems from parallel tempering Bayeslands (Chandra et al., 2019c) that are adapted from earlier work (Chandra et al., 2019a). These include Continental-Margin (CM) and Synthetic-Mountain (SM), which are chosen due to the computational time taken for running a single model since they use less than 5 s to run a single model on a single central processing unit (CPU). These problems are well suited for a parameter evaluation for the proposed surrogate-assisted Bayesian inversion framework. In order to demonstrate an application which is computationally expensive, we introduce another problem, which features the landscape evolution of Tasmania in Australia for a million years that features the region shown in Fig. 1b. The Synthetic-Mountain landscape evolution is a synthetic problem, while the Continental-Margin problem is a real-world problem based on the topography of a region.
Figure 2. Synthetic-Mountain: initial and eroded ground-truth topography after a million years of evolution. Continental-Margin: initial and eroded ground-truth topography and sediment after 1 million years. The erosion–deposition that forms sediment deposition after 1 million years is also shown. Note that $x$ axis represents the latitude; $y$ axis represents the longitude, and that aligns with Fig. 1a. The elevation in metres (m) is given by the $z$ axis, which is further shown as a colour bar. The Synthetic-Mountain problem does not align with actual landscape.
Figure 3. Tasmania: initial and eroded ground-truth topography along with erosion–deposition that shows sediment deposition after 1 million years evolution. Note that $x$ axis represents the latitude; $y$ axis represents the longitude, and that aligns with Fig. 1b for the Tasmania problem. The elevation in metres (m) is given by the $z$ axis, which is further shown as a colour bar.

along the eastern margin of Te Waipounamu/South Island of Aotearoa/New Zealand as shown in Fig. 1a. We use Badlands to evolve the initial landscape with parameter settings given in Tables 1 and 2 and create the respective problems synthetic ground-truth topography.

The initial and synthetic ground-truth topographies along with erosion/deposition for these problems appear in Figs. 2 and 3, respectively. Note that the figure shows that the Synthetic-Mountain is flat in the beginning, then given a constant uplift rate, along with weathering with constant precipitation rate, which creates the mountain topography. We use present-day topography as the initial topography in the Continental-Margin and Tasmania problems, whereas we use a synthetic flat region for Synthetic-Mountain initial topography. The problems involve an erosion–deposition model history that is used to generate synthetic ground-truth data for
which has been adapted from previous work (Chandra et al., 2019c) and is given as follows. The initial topography is denoted by $D_0$ with $D_0 = (D_{0,s_1}, \ldots, D_{0,s_n})$, where $s_i$ corresponds to site $s_i$, with the coordinates given by the latitude $\nu_i$ and longitude $\nu_i$.

We assume an inverse gamma (IG) prior $\tau^2 \sim \text{IG}(v/2, 2/v)$ and integrate it so that the likelihood for the topography at time $t = T$ is

$$L_1(\theta) \propto \prod_{i=1}^{n} \left( 1 + \frac{(D_{0,s_i} - f_{0,s_i}(\theta))^2}{\nu} \right)^{-\frac{v+1}{2}} ,$$

(1)

where $\nu$ is the number of observations, and the subscript “1” in $L_1(\theta)$ denotes that it is the landscape likelihood to distinguish it from a sediment likelihood.

Although Badlands produces successive time-dependent topographies, only the final topography $D_T$ is used for the calculation of the elevation likelihood since little ground-truth information is available for the detailed evolution of surface topography. In contrast, the time-dependence of sedimentation can be used to ground-truth the time-dependent evolution of surface process models that include sediment transportation and deposition. The sediment erosion/deposition values at time $T$ are simulated (predicted) by the Badlands model given set of parameters, $\theta$, plus some Gaussian noise as follows:

$$z_{s_j,t} = g_{s_j,t}(\theta) + \eta_{s_j,t} \quad \text{with} \quad \eta_{s_j,t} \sim (0, \chi^2).$$

(2)

The sediment likelihood $L_s(\theta)$, after integrating out $\chi^2$, becomes

$$L_s(\theta) \propto \prod_{i=1}^{J} \prod_{j=1}^{T} \left( 1 + \frac{(z_{s_j,t} - g_{s_j,t}(\theta))^2}{\nu} \right)^{-\frac{v+1}{2}} .$$

(3)

The combined likelihood takes both elevation and sediment/deposition into account

$$L(\theta) = L_1(\theta) \times L_s(\theta)$$

(4)
3.3 Surrogate-assisted Bayeslands

The surrogate model learns from the relationship between the set of input parameters and the response given by the true (Badlands) model. The input is the set of proposals by the respective replica samplers in the parallel tempering MCMC sampling algorithm. We refer to the likelihood estimation by the surrogate model as the *pseudo-likelihood*.

We need to take into account the cost of inter-process communication in parallel computing environment to avoid computational overhead. As given in our previous implementa-
tion (Chandra et al., 2019c), the swap interval refers to the
number of iterations after which each replica pauses and can
undergo a replica transition. After the swap proposal is ac-
cepted or rejected, the respective replica sampling is resumed
while undergoing Metropolis transition in between the swap
intervals. We incorporate the surrogate-assisted estimation
into the multicore parallel tempering algorithm. Our previous
work (Chandra et al., 2020) used a surrogate interval that
determines the frequency of training by collecting the history of
past samples with their likelihood from the respective repli-
cas. We need a swap interval of several samples when dealing
with small-scale models that take a few seconds to run; how-
ever for large models, we recommend having a swap interval
of 1.

Taking into account that the true model is represented as
\( y = f(x) \), the surrogate model provides an approximation in
the form \( \hat{y} = f(x) \); such that \( y = \hat{y} + e \), where \( e \) represents
the difference or error. The task of the surrogate model is to
provide an estimate for the pseudo-likelihood by training from
the history of proposals, which is given by the set of input
\( x_s \) and likelihood \( y_s \), where “s” represents the sample and
“r” represents the replica. Hence, we create the training dataset
\( \Phi \) for the surrogate by fusion of \( x_{r,s} \) across all the replica
for a given surrogate interval \( \psi \), which can be formu-
lated as follows:

\[
\Phi = \{x_{1,s}, \ldots, x_{1,s+\psi}, \ldots, x_{M,s}, \ldots, x_{M,s+\psi}\}
\]

\[
\lambda = (y_{1,s}, \ldots, y_{1,s+\psi}, \ldots, y_{M,s}, \ldots, y_{M,s+\psi}),
\]

(5)

where, \( x_{r,s} \) represents the set of parameters proposed at sam-
ple “s”, \( y_{r,s} = \log(p(y|x_{r,s})) \) is the likelihood, which is de-
pendent on data and the Badlands model, and \( M \) is the total
number of replicas. \( \Theta \) denotes the training surrogate dataset,
which features input \( \Phi \) and response \( \lambda \) at the end of every
surrogate interval denoted by \( s + \psi \). Therefore, we give the
pseudo likelihood as \( \hat{y} = \hat{f}(\Theta) \), where \( \hat{f} \) is the prediction
from the surrogate model. The likelihood in training data is
altered, with respect of the temperature, since it has been
changed by taking \( L_{local}/T_r \) for given replica “r”. We undo
this change by multiplying the likelihood by the respective
replica temperature level taken from the geometric tempera-
ture ladder.

We present surrogate-assisted Bayeslands in Algorithm 1,
which features parallel processing of the ensemble of repli-
cas. The highlighted region in the colour pink of the Algo-

R. Chandra et al.: Surrogate-assisted Bayeslands

Due to multiple parallel processing replicas, it is not straight-
forward to implement when to terminate sampling. Hence,
the termination condition waits for all the replica processes
to end as it monitors the number of active or alive replica pro-
cesses in the manager process. We begin by setting the num-
ber of alive replicas in the ensemble (alive = \( M \)) and then the
replicas that sample \( \theta_s \) are assigned values using a uniform
distribution \([-\alpha, \alpha]\); where \( \alpha \) defines the range of the respec-
tive parameters. We then assign the user-defined parameters,
which include the number of replica samples \( R_{\text{max}} \), swap-
interval \( R_{\text{swap}} \), surrogate interval, \( \psi \), and surrogate proba-

The samples that cover the first surrogate interval makes
up the initial surrogate training data \( \Theta \), which feature all
the replicas. We then train the surrogate to estimate the pseudo-
likelihood when required according to the surrogate proba-

tility. Figure 4 shows how the manager processing unit con-

To enable better estimation for the pseudo-likelihood, we
retrain the surrogate model for remaining surrogate interval
blocks until the maximum time \( (R_{\text{max}}) \). We train the surro-

gate model only in the manager process and the algorithm
passes the surrogate model copy with the trained parameters
to the ensemble of replica processes for predicting or estimat-

The samples associated with the true-likelihood only becomes
part of the surrogate training dataset. In Stage 1.4 of Algo-

Stage 1.5 calculates the likelihood moving average of past
three likelihood values, \( L_{\text{past}} = \text{mean}(L_{s-1}, L_{s-1}, L_{s-2}) \). In
Stage 1.6, we combine the moving average likelihood with
the pseudo-likelihood to give a prediction that considers the
present replica proposal and taking into account the past,
\( L_{\text{local}} = (0.5 \times L_{\text{surrogate}}) + 0.5 \times L_{\text{past}} \). The surrogate
training can consume a significant portion of time, which is de-
pendent on the size of the problem in terms of the number of
parameters and also the type of surrogate model used, along
with the training algorithm. We evaluate the trade-off be-

We validate the quality of estimation from the surrogate
model by the root-mean-squared error (RMSE), which con-
siders the difference between the true likelihood and the
pseudo-likelihood. This can be seen as a regression prob-

Table 4. Neural network architecture for the different problems.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Input</th>
<th>Output</th>
<th>Train size</th>
<th>Test size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continental-Margin</td>
<td>6</td>
<td>1</td>
<td>8073</td>
<td>879</td>
</tr>
<tr>
<td>Synthetic-Mountain</td>
<td>5</td>
<td>1</td>
<td>8073</td>
<td>879</td>
</tr>
</tbody>
</table>
### Table 5. Evaluation of surrogate training accuracy.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Batch ratio</th>
<th>Transfer and train</th>
<th>Train from scratch</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SGD</td>
<td>Adam</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSE Time(s)</td>
<td>MSE Time(s)</td>
</tr>
<tr>
<td>Continental-Margin</td>
<td>0.1</td>
<td>0.0198 19.40</td>
<td>0.0209 31.23</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.0197 26.95</td>
<td>0.0211 56.84</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td>0.0199 25.53</td>
<td>0.0212 61.41</td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>0.0195 70.42</td>
<td>0.0193 48.28</td>
</tr>
<tr>
<td>Synthetic-Mountain</td>
<td>0.1</td>
<td>0.0161 40.38</td>
<td>0.0097 54.45</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.0134 52.87</td>
<td>0.007 70.65</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td>0.0129 65.105</td>
<td>0.0088 73.035</td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>0.0164 50.14</td>
<td>0.0048 87.67</td>
</tr>
</tbody>
</table>

### Table 6. Convergence diagnosis (PSRF score) for Continental-Margin problem.

<table>
<thead>
<tr>
<th>Proposal</th>
<th>Method</th>
<th>Precip.</th>
<th>Erod.</th>
<th>m value</th>
<th>n value</th>
<th>c-marine</th>
<th>c-surface</th>
<th>Mean R score</th>
</tr>
</thead>
<tbody>
<tr>
<td>RW</td>
<td>PT-Bayeslands</td>
<td>1.50</td>
<td>1.6</td>
<td>1.14</td>
<td>4.82</td>
<td>2.62</td>
<td>1.56</td>
<td>2.21</td>
</tr>
<tr>
<td>ARW</td>
<td>PT-Bayeslands</td>
<td>1.26</td>
<td>1.55</td>
<td>1.26</td>
<td>1.63</td>
<td>1.38</td>
<td>1.13</td>
<td>1.37</td>
</tr>
<tr>
<td>RW</td>
<td>SAPT-Bayeslands</td>
<td>4.06</td>
<td>1.70</td>
<td>6.57</td>
<td>1.51</td>
<td>1.46</td>
<td>1.49</td>
<td>2.80</td>
</tr>
<tr>
<td>ARW</td>
<td>SAPT-Bayeslands</td>
<td>1.33</td>
<td>2.88</td>
<td>1.22</td>
<td>2.46</td>
<td>1.03</td>
<td>1.30</td>
<td>1.70</td>
</tr>
</tbody>
</table>

We evaluate the prediction performance by comparing the predicted/simulated Badlands landscape with the ground-truth data using the root-mean-squared error (RMSE). We compute the RMSE for the elevation (elev) and sediment erosion/deposition (sed) at each iteration of the sampling scheme using

\[
\text{RMSE}_{\text{elev}} = \sqrt{\frac{1}{n \times m} \sum_{i=1}^{n} \sum_{j=1}^{m} \left( g(\hat{\theta}_{T,i,j}) - g(T_{i,j}(\theta)) \right)^2}
\]

\[
\text{RMSE}_{\text{sed}} = \sqrt{\frac{1}{n \times v} \sum_{i=1}^{n} \sum_{j=1}^{m} \left( f(\hat{\theta}_{t,i,j}) - f(\theta_{t,i,j}) \right)^2}
\]

where \( \hat{\theta} \) is an estimated value of \( \theta \), and \( \theta \) is the true value representing the synthetic ground truth. \( f(.) \) and \( g(.) \) represent the outputs of the Badlands model, while \( m \) and \( n \) represent the size of the selected topography. \( v \) is the number of selected points from sediment erosion/deposition over the selected time frame, \( n_t \).

### 3.4 Surrogate model

To choose a particular surrogate model, we need to consider the computational resources for training the model during the sampling process. The literature review showed that Gaussian process models, neural networks, and radial basis functions (Broomhead and Lowe, 1988) are popular choices for surrogate models. We note that Badlands LEM features about a dozen free parameters in one of the simplest cases; this increases when taking into account spatial and temporal dependencies. For instance, the precipitation rate for a million years can be represented by a single parameter or by 10 different parameters that capture every 100,000 years for 10
Table 7. Evaluation for Continental-Margin problem.

<table>
<thead>
<tr>
<th>Method</th>
<th>$S_{\text{prob}}$</th>
<th>$\psi$</th>
<th>RMSE$_{\text{elev}}$ (mean)</th>
<th>RMSE$_{\text{elev}}$ (SD)</th>
<th>RMSE$_{\text{sed}}$ (mean)</th>
<th>RMSE$_{\text{sed}}$ (SD)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PT-Bayeslands</td>
<td>N/A</td>
<td>N/A</td>
<td>78.80</td>
<td>10.03</td>
<td>35.91</td>
<td>11.36</td>
<td>3243.30</td>
</tr>
<tr>
<td>SAPT-Bayeslands</td>
<td>0.20</td>
<td>0.05</td>
<td>75.53</td>
<td>9.89</td>
<td>35.68</td>
<td>10.93</td>
<td>3082.53</td>
</tr>
<tr>
<td>SAPT-Bayeslands</td>
<td>0.40</td>
<td>0.05</td>
<td>80.22</td>
<td>15.63</td>
<td>44.72</td>
<td>16.52</td>
<td>2450.77</td>
</tr>
<tr>
<td>SAPT-Bayeslands</td>
<td>0.60</td>
<td>0.05</td>
<td>82.04</td>
<td>8.23</td>
<td>44.33</td>
<td>13.37</td>
<td>1859.52</td>
</tr>
<tr>
<td>SAPT-Bayeslands</td>
<td>0.80</td>
<td>0.05</td>
<td>79.30</td>
<td>26.70</td>
<td>43.29</td>
<td>18.68</td>
<td>1149.63</td>
</tr>
<tr>
<td>SAPT-Bayeslands</td>
<td>0.20</td>
<td>0.10</td>
<td>76.92</td>
<td>11.59</td>
<td>48.19</td>
<td>11.46</td>
<td>3075.31</td>
</tr>
<tr>
<td>SAPT-Bayeslands</td>
<td>0.40</td>
<td>0.10</td>
<td>82.43</td>
<td>11.58</td>
<td>46.47</td>
<td>12.55</td>
<td>2494.13</td>
</tr>
<tr>
<td>SAPT-Bayeslands</td>
<td>0.60</td>
<td>0.10</td>
<td>80.12</td>
<td>12.08</td>
<td>47.80</td>
<td>19.05</td>
<td>1934.34</td>
</tr>
<tr>
<td>SAPT-Bayeslands</td>
<td>0.80</td>
<td>0.10</td>
<td>88.81</td>
<td>20.61</td>
<td>51.12</td>
<td>14.26</td>
<td>1148.80</td>
</tr>
<tr>
<td>SAPT-Bayeslands</td>
<td>0.20</td>
<td>0.15</td>
<td>44.90</td>
<td>33.54</td>
<td>23.95</td>
<td>19.86</td>
<td>2914.06</td>
</tr>
<tr>
<td>SAPT-Bayeslands</td>
<td>0.40</td>
<td>0.15</td>
<td>73.64</td>
<td>8.05</td>
<td>38.53</td>
<td>10.02</td>
<td>2495.56</td>
</tr>
<tr>
<td>SAPT-Bayeslands</td>
<td>0.60</td>
<td>0.15</td>
<td>83.38</td>
<td>8.45</td>
<td>51.15</td>
<td>19.07</td>
<td>1986.51</td>
</tr>
<tr>
<td>SAPT-Bayeslands</td>
<td>0.80</td>
<td>0.15</td>
<td>84.73</td>
<td>10.04</td>
<td>39.78</td>
<td>14.44</td>
<td>1294.64</td>
</tr>
</tbody>
</table>

Table 8. Performance comparison for respective problems and methods. N/A: not applicable.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Method</th>
<th>$S_{\text{prob}}$</th>
<th>$\psi$</th>
<th>RMSE$_{\text{elev}}$ (mean)</th>
<th>RMSE$_{\text{elev}}$ (SD)</th>
<th>RMSE$_{\text{sed}}$ (mean)</th>
<th>RMSE$_{\text{sed}}$ (SD)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continental-Margin</td>
<td>PT-Bayeslands</td>
<td>N/A</td>
<td>N/A</td>
<td>78.80</td>
<td>10.03</td>
<td>35.91</td>
<td>11.36</td>
<td>3243.30</td>
</tr>
<tr>
<td></td>
<td>SAPT-Bayeslands</td>
<td>0.60</td>
<td>0.05</td>
<td>82.0</td>
<td>8.23</td>
<td>44.33</td>
<td>13.37</td>
<td>1859.52</td>
</tr>
<tr>
<td>Synthetic-Mountain</td>
<td>PT-Bayeslands</td>
<td>N/A</td>
<td>N/A</td>
<td>106.10</td>
<td>48.24</td>
<td>20.34</td>
<td>24.02</td>
<td>8474.67</td>
</tr>
<tr>
<td></td>
<td>SAPT-Bayeslands</td>
<td>0.60</td>
<td>0.05</td>
<td>104.88</td>
<td>5.51</td>
<td>11.87</td>
<td>8.69</td>
<td>4161.43</td>
</tr>
<tr>
<td>Tasmania</td>
<td>PT-Bayeslands</td>
<td>N/A</td>
<td>N/A</td>
<td>172.64</td>
<td>10.74</td>
<td>3.90</td>
<td>0.50</td>
<td>600 293.61</td>
</tr>
<tr>
<td></td>
<td>SAPT-Bayeslands</td>
<td>0.60</td>
<td>0.05</td>
<td>179.67</td>
<td>19.71</td>
<td>3.91</td>
<td>0.10</td>
<td>221 942.41</td>
</tr>
</tbody>
</table>

different regions, which can account for 1000 parameters instead of 1. Considering hundreds or thousands of unknown Badlands model parameters, the surrogate model needs to be efficiently trained without taking lots of computational resources. The flexibility of the model to have incremental training is also needed, and hence, we rule out Gaussian process models since they have limitations in training when the size of the dataset increases to a certain level (Rasmussen, 2004). Therefore, we use neural networks as the choice of the surrogate model, and the training data and neural network model is formulated as follows.

We denote the surrogate model training data by $\Phi$ and $\lambda$, which is shown in Eq. (5), where $\Phi$ is the input, and $\lambda$ is the desired output of the model. The prediction of the model is denoted by $\hat{\lambda}$. We use a feedforward neural network as the surrogate model. Given input $x_t$, $f(x_t)$ is computed by the feedforward neural network with one hidden layer defined by the function

$$f(x_t) = g\left(\delta_o + \sum_{h=1}^{H} v_j g\left(\delta_h + \sum_{d=1}^{I} w_{dh} x_t\right)\right),$$

where $\delta_o$ and $\delta_h$ are the bias weights for the output $o$ and hidden $h$ layer, respectively. $v_j$ is the weight which maps the hidden layer $h$ to the output layer, $w_{dh}$ is the weight which maps $x_t$ to the hidden layer $h$, and $g(.)$ is the activation function for the hidden and output layer units. We use ReLU (rectified linear unitary function) as the activation function. The learning or optimization task then is to iteratively update the weights and biases to minimize the cross-entropy loss $J(W, B)$. This can be done using gradient update of weights using the Adam (adaptive moment estimation) learning algorithm (Kingma and Ba, 2014) and stochastic gradient descent (Bottou, 1991, 2010). We experimentally evaluate them for training the feedforward network for the surrogate model in the next section.

3.5 Proposal distribution

Bayeslands features random-walk (RW) and adaptive-random-walk (ARW) proposal distributions which will be evaluated further for surrogate-assisted Bayeslands in our experiments. In our previous work (Chandra et al., 2019a), ARW showed better convergence properties when com-
pared to RW proposal distribution. The RW proposal distribution features $\Sigma$ as the diagonal matrix, so that $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_P^2)$, where $\sigma_j$ is the step size of the $j$th element of the parameter vector $\theta$. The step size for $\theta_j$ is a combination of a fixed step size $\phi$, which is common to all parameters, multiplied by the range of possible values for parameter $\theta_j$; hence $\sigma_j = (a_j - b_j) \times \phi$, where $a_j$ and $b_j$ represent the maximum and minimum limits of the prior for $\theta_j$ given in Table 2. In our experiments, the RW proposal distribution employs fixed step size, $\phi = 0.05$.

The ARW proposal distribution features adaptation of the diagonal matrix $\Sigma$ at every $K$ interval of within-replica sampling. It allows for the dependency between elements of $\theta$ and adapts during sampling (Haario et al., 2001). We adapt the elements of $\Sigma$ for the posterior distribution using the sample covariance of the current chain history $\Sigma = \text{cov}((\theta^{[0]}, \ldots, \theta^{[i-1]})) + \text{diag}(\lambda_1^2, \ldots, \lambda_P^2)$, where $\theta^{[i]}$ is the $i$th iterate of $\theta$ in the chain, and $\lambda_j$ is the minimum allowed step sizes for each parameter $\theta_j$.

3.6 Design of experiments

We demonstrate effectiveness of surrogate-assisted parallel tempering (SAPT-Bayeslands) framework for selected Badlands LEMs taken from our previous study (Chandra et al., 2019c).

We first investigate the effects of different surrogate training procedures and parameter evaluation for SAPT-Bayeslands using smaller synthetic problems. Afterwards, we apply the methodology to a larger landscape evolution problem, which is Tasmania, Australia. We design the experiments as follows.

1. We generate a dataset for training and testing the surrogate for the Synthetic-Mountain and Continental-Margin landscape evolution problems. We use the neural network model for the surrogate and evaluate different training techniques.

2. We evaluate if the transfer of knowledge from previous surrogate interval is better than no transfer of knowledge for Synthetic-Mountain and Continental-Margin problems. Note this is done only with the data generated from the previous step.

3. We provide convergence diagnosis for the RW and ARW proposal distributions in PT-Bayeslands and SAPT-Bayeslands.
4. We integrate the surrogate model into Bayeslands and evaluate the effectiveness of the surrogate in terms of estimation of the likelihood and computational time. Due to the computational requirements, we only consider the Continental-Margin problem.

5. We then apply SAPT-Bayeslands to all the given problems and compare with PT-Bayeslands.

We use Keras neural networks library (Gulli and Pal, 2017) for implementation of the surrogate. We provide the open-source software package that implements Algorithm 1 along with benchmark problems and experimental results \(^1\).

We use a geometric temperature ladder with a maximum temperature of \(T_{\text{max}} = 2\) for determining the temperature level for each of the replicas. In trial experiments, the selection of these parameters depended on the performance in terms of the number of accepted samples and prediction accuracy of elevation and sediment/deposition. We use replica-exchange or swap interval value; \(R_{\text{swap}} = 3\) samples that determine when to check whether to swap with the neighbouring replica. In previous work (Chandra et al., 2019c), we observed that increasing the number of replicas up to a certain point does not necessarily mean that we get better performance in terms of the computational time or prediction accuracy. In this work, we limit the number of replicas to \(R_{\text{num}} = 8\) for all experiments with maximum of 5000 samples.

We use a 50 % burn in, which discards the portion of samples in the parallel tempering MCMC stage as done in our previous work (Chandra et al., 2019a).

4 Results

4.1 Surrogate accuracy

To implement the surrogate model, we need to evaluate the training algorithm, such as Adam and stochastic gradient descent (SGD). Furthermore, we also evaluate specific parameters, such as the size of the surrogate interval (batch ratio), the neural network topology for the surrogate, and the effectiveness of either training from scratch or utilizing previous knowledge for surrogate training (transfer and train). We create a training dataset from the cases where the true likelihood was used, which compromises the history of the set of pa-
Figure 7. Topography cross section and erosion–deposition prediction for 10 chosen points (selected coordinates denoted by location identifier, ID, number) for Tasmania problem from results summarized in Table 8.

Figure 8. Surrogate likelihood vs. true likelihood estimation for Continental-Margin problem (RMSE_{surf} = 3605).

Figure 9. Surrogate likelihood vs. true likelihood estimation for Synthetic-Mountain problem (RMSE_{surf} = 9917).
Algorithm 1: Surrogate-assisted Bayeslands

Data: Ground-truth topography dataset
Result: Posterior distribution of unknown parameters $\theta$ \{precipitation and erodibility\}

i. Initialize $M$ replicates, $\theta_1, \theta_2, \ldots, \theta_M$ with corresponding temperature values $T_1, T_2, \ldots, T_M$

ii. Set all replicates in ensemble as $alive$; $alive = M$

iii. Define the surrogate interval ($\psi$), surrogate probability $S_{prob}$, and maximum number of samples for each replica ($R_{max}$).

(We highlight the manager process in colour blue and ensemble of replica processes running in parallel in colour pink.)

while ($alive \neq 0$)

<table>
<thead>
<tr>
<th>Stage 0: Prepare manager process to execute each replica in parallel cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>for each replica $r$ in $M$ do</td>
</tr>
<tr>
<td>while ($i &lt; R_{max}$) do</td>
</tr>
<tr>
<td>Stage 1.0: Metropolis Transition</td>
</tr>
<tr>
<td>for each $s$ in $\psi$ do</td>
</tr>
<tr>
<td>1.1 Random-walk, $\theta_s^* = \theta_s + \epsilon$</td>
</tr>
<tr>
<td>1.2 $L_{local}$ calculate:</td>
</tr>
<tr>
<td>1.2.1 Draw $k$ from a Uniform distribution [0,1]</td>
</tr>
<tr>
<td>if $k &lt; S_{prob}$ and $s &gt; \psi$ then</td>
</tr>
<tr>
<td>1.2.2 Estimate $L_{local}$ from local surrogate's prediction, $L_{surrogate}$</td>
</tr>
<tr>
<td>1.2.3 Copy global surrogate knowledge to local surrogate</td>
</tr>
<tr>
<td>1.2.4 Predict $L_{surrogate}$ value with the proposed $\theta_s^*$</td>
</tr>
<tr>
<td>1.2.5 $L_{past} = \text{mean}(L_{s-1}, L_{s-1}, L_{s-2})$</td>
</tr>
<tr>
<td>1.2.6 Assign $L_{local} = (0.5 \times L_{surrogate}) + 0.5 \times L_{past}$</td>
</tr>
<tr>
<td>1.2.7 Save $L_s = L_{local}$</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>$L_{local} = \text{true-likelihood}, \text{given by the Likelihood function in Equation 4}$</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>1.8 Draw $\alpha$ from uniform distribution [0,1]</td>
</tr>
<tr>
<td>if $\alpha \leq L_{local}(\theta_s \rightarrow \theta_s^*)$ then</td>
</tr>
<tr>
<td>1.9 Update replica state, $\theta_s \leftarrow \theta_s^*$</td>
</tr>
<tr>
<td>end</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stage 2.0: Replica Transition;</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Draw $\beta$ from a Uniform distribution [0,1]</td>
</tr>
<tr>
<td>if $\beta \leq P(\theta_s \leftrightarrow \theta_{s+1})$ then</td>
</tr>
<tr>
<td>2.2 Signal() manager process</td>
</tr>
<tr>
<td>2.3 Exchange neighboring Replica, $\theta_s \leftrightarrow \theta_{s+1}$</td>
</tr>
<tr>
<td>end</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stage 3.0: Check when to end the process</th>
</tr>
</thead>
<tbody>
<tr>
<td>if $i == R_{max} - 1$ then</td>
</tr>
<tr>
<td>3.1 Signal() manager process</td>
</tr>
<tr>
<td>3.2 decrement number of replica processes $alive$</td>
</tr>
<tr>
<td>end</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stage 4.0: Signal() manager process</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Set $\Theta$ which features history of proposals $\Phi(\theta)$ and response $\lambda(L_{local})$ from Stage 1.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stage 5.0: Global Surrogate Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>for each replica do</td>
</tr>
<tr>
<td>5.1 Get $\Theta$ which features history of proposals $\Phi(\theta)$ and response $\lambda(L_{local})$</td>
</tr>
<tr>
<td>5.2 Append proposal list to $X$</td>
</tr>
<tr>
<td>5.3 Append likelihood list to $Y$</td>
</tr>
<tr>
<td>end</td>
</tr>
</tbody>
</table>

| 5.4 Train global surrogate model with input $X$ and output $Y$ |

| 5.5 Save global surrogate model parameters |

end

| Stage 6: Combine predictions and posterior from respective replicas in the ensemble. |
rameters proposed with the corresponding likelihood. This is done for standalone evaluation of the surrogate model, which further ensures that the experiments are reproducible since different experimental runs create different datasets depending on the exploration during sampling. We then evaluate the neural network model designated for the surrogate using two major training algorithms which featured the Adam optimizer and stochastic gradient descent. The parameters that define the neural network surrogate model used for the experiments are given in Table 4. Note that the train size in Table 4 refers to the maximum size of the dataset. The training is done in batches where the batch ratio determines the training dataset size, as shown in Table 5.

Table 5 presents the results for the experiments that took account of the training data collected during sampling for two benchmark problems (Continental-Margin and Synthetic-Mountain). Note that we report the mean value of the mean-squared-error (MSE) for the given batch ratio from 10 experiments. The batch ratio is taken, in relation to the maximum number of samples across the chains ($R_{\text{max}}/R_{\text{num}}$). We normalize the likelihood values (outcomes) in the dataset to the range $[0,1]$. In most cases, the accuracy of the neural network is slightly better when training from scratch with combined data; however, there is a considerable trade-off with the time required to train the network. The results show that the transfer and train methodology, in general, requires much lower computational time when compared to training from scratch with combined data. Moreover, in comparison to SGD and Adam training algorithms, we observe that SGD achieves slightly better accuracy than Adam for Continental-Margin problem. However, Adam, having an adaptive learning rate, outperforms SGD in terms of the time required to train the network. Thus, we can summarize that transfer and train method is better since it saves significant computation time with a minor trade-off with accuracy.

### 4.2 Convergence diagnosis

The Gelman–Rubin diagnostic (Gelman and Rubin, 1992) is one of the popular methods used for evaluating convergence by analysing the behaviour of multiple Markov chains. The assessment is done by comparing the estimated between-chain and within-chain variances for each parameter, where large differences between the variances indicate non-convergence. The diagnosis reports the potential scale reduction factor (PSRF), which gives the ratio of the current variance in the posterior variance for each parameter compared to that being sampled, and the values for the PSRF near 1 indicates convergence. We analyse five experiments for each case using different initial values for 5000 samples for each problem configuration.

Table 6 presents the convergence diagnosis using the PSRF score for RW and ARW proposal distributions for PT-Bayeslands and SAPT-Bayeslands. We notice that ARW has a lower PSRF score (mean) when compared to the RW proposal distribution, which indicates better convergence. We also notice that the ARW SAPT-Bayeslands maintains convergence with a PSRF score close to ARW PT-Bayeslands when compared to rest of the configurations. This suggests that although we use surrogates, convergence can be maintained up to a certain level, which is better than RW PT-Bayeslands.

### 4.3 Surrogate-assisted Bayeslands

We investigate the effect of the surrogate probability ($S_{\text{prob}}$) and surrogate interval ($\psi$) on the prediction accuracy (RMSE$_{\text{elev}}$ and RMSE$_{\text{sed}}$) and computational time. Note that we report the prediction accuracy mean and standard deviation (mean and SD) of accepted samples over the sampling time after removing the burn-out period. We report the computational time in seconds (s). Table 7 presents the performance of the respective methods (PT-Bayeslands and SAPT-Bayeslands) with respective parameter settings for the Continental-Margin problem. In SAPT-Bayeslands, we observe that there is not a major difference in the accuracy of elevation or erosion/deposition given different values of $S_{\text{prob}}$. Nevertheless, there is a significant difference in terms of the computational time where higher values of $S_{\text{prob}}$ save computational time. Furthermore, we notice that there is not a significant difference in the prediction accuracy given different values of $\psi$, which suggests that the selected values are sufficient.

We select a suitable combination of the set of parameters evaluated in the previous experiment ($S_{\text{prob}} = 0.6$ and $\psi = 0.05$) and apply them to rest of the problems. Table 8 gives a comparison of performance for Continental-Margin and Synthetic-Mountain problems, along with the Tasmania one, which is a bigger and more computationally expensive problem. We notice that the performance of SAPT-Bayeslands is similar to PT-Bayeslands, while a significant portion of computational time is saved.

Figures 5, 6, and 7 provide a visualization of the elevation prediction accuracy when compared to actual ground truth between the given methods from results given in Table 8. We also provide the prediction accuracy of erosion/deposition for 10 chosen points taken at selected locations. Although both methods provide erosion/deposition prediction for four successive time intervals, we only show the final time interval. In both the Continental-Margin and Synthetic-Mountain problems, we notice that the prediction accuracy of PT-Bayeslands is very similar to SAPT-Bayeslands, and the Badlands prediction of the topography is close to ground truth, within the credible interval. This indicates that the use of surrogates has been beneficial where no major loss in accuracy in prediction is given. In the case of the Tasmania problem, there is a loss in Badlands prediction accuracy, which could be due to the size of the problem. Nevertheless, this loss is not that clear from results in Table 8. It could be that the topog-

https://doi.org/10.5194/gmd-13-2959-2020
Topography prediction is mostly inconsistent at the cross section where it features mountainous regions.

Figures 8 and 9 show the true likelihood and prediction by the surrogate for the Continental-Margin and Synthetic-Mountain problems, respectively. We notice that at certain intervals given in Fig. 8, given by different replica, there is inconsistency in the predictions. Moreover, Fig. 9 shows that the log-likelihood is very chaotic, and hence there is difficulty in providing robust prediction at certain points in the time given by samples for the respective replica.

4.4 Discussion

We observe that the surrogate probability is directly related to the computational performance; this is obvious since computational time depends on how often we use the surrogate. Our concern is the prediction performance, especially while increasing the use of the surrogate as it could lower the accuracy, which can result in a poor estimation of the parameters. According to the results, the accuracy is well retained given a higher probability of using surrogates. In the cross section presented in the results for Continental-Margin and Synthetic-Mountain problems, we find that there is not much difference in the accuracy given in prediction by the SAPT-Bayeslands when compared to PT-Bayeslands. Moreover, in the application to a more computationally intensive problem (Tasmania), we find that a significant reduction in computational time is achieved. Although we demonstrated the method using small-scale models that run within a few seconds to minutes, the computational costs of continental-scale Badlands models are extensive. For instance, the computational time for a 5 km resolution for the Australian continent Badlands model for 149 million years is about 72 h; hence, in the case when thousands of samples are required, the use of surrogates can be beneficial. We note that improved efficiency of the surrogate-assisted Bayeslands comes at the cost of accuracy for some problems (in case of the Tasmania problem), and there is a trade-off between accuracy and computational time.

In future work, rather than a global surrogate model, we could use the local surrogate model on its own, where the training only takes place in the local surrogates by relying on the history of the likelihood and hence taking a univariate time series prediction approach using neural networks. Our primary contribution is in terms of the parallel-computing-based open-source software and the proposed underlying framework for incorporating surrogates, taking into account complex issues such as inter-process communication. This opens the road to using different types of surrogate models while using the underlying framework and open-source software. Given that the sediment erosion/deposition is temporal, other ways of formulating the likelihood could be possible; for instance, we could have a hierarchical Bayesian model with two stages for MCMC sampling (Chib and Carlin, 1999; Wikle et al., 1998).

The initial evaluation for the setup surrogate model shows that it is best to use a transfer learning approach where the knowledge from the past surrogate interval is utilized and refined with new surrogate data. This consumes much less time than accumulating data and training the surrogate from scratch at every surrogate interval. We note that in the case when we use the surrogate model for pseudo-likelihood, there is no prediction given by the surrogate model. The prediction (elevation topography and erosion–deposition) during sampling are gathered only from the true Badlands model evaluation rather than the surrogate. In this way, one could argue that the surrogate model is not mimicking the true model; however, we are guiding the sampling algorithm towards forming better proposals without evaluation of the true model. A direction forward is in incorporating other forms of surrogates, such as running a low-resolution Badlands model as the surrogate, which would be computationally faster in evaluating the proposals; however, limitations in terms of the effect of resolution setting on Badlands topography simulation may exist.

Furthermore, computationally efficient implementations of landscape evolution models that only feature landscape evolution (Braun and Willett, 2013) could be used as the surrogate, while we could use Badlands model that features both landscape evolution and erosion/deposition as the true model. We could also use computationally efficient implementations of landscape evolution models that consider parallel processing (Hassan et al., 2018) in the Bayeslands framework. In this case, the challenge would be in allocating specialized processing cores for Badlands and others for parallel tempering MCMC.

We adapted the surrogate framework developed for machine learning (Chandra et al., 2020) with a different proposal distribution instead of using gradient-based proposals. Gradient-based parameter estimation has been very popular in machine learning due to availability of gradient information. Due to the complexity in geological or geophysical numerical forward models, it is challenging to obtain gradients, which has been the case for the Badlands landscape evolution model. We used random-walk and adaptive-random-walk proposal distributions which have limitations; hence, we need to incorporate advanced meta-heuristic techniques to form non-gradient-based proposals for efficient search. Our study is limited to a relatively small set of free parameters, and a significant challenge would be to develop surrogate models with an increased set of parameters.

5 Conclusions

We presented a novel application of surrogate-assisted parallel tempering that features parallel computing for landscape evolution models using Badlands. Initially, we experimented with two different approaches for training the surrogate model, where we found that a transfer learning-based
approach is beneficial and could help reduce the computational time of the surrogate. Using this approach, we presented the experiments that featured evaluating certain key parameters of the surrogate-based framework. In general, we observed that the proposed framework lowers the computational time significantly while maintaining the required quality in parameter estimation and uncertainty quantification.

In future work, we envision applying the proposed framework to more complex applications such as the evolution of continental-scale landscapes and basins over millions of years. We could use the approach for other forward models such as those that feature geological reef development or lithospheric deformation. Furthermore, the posterior distribution of our parameters requires multimodal sampling methods; hence, a combination of meta-heuristics for proposals with surrogate-assisted parallel tempering could improve exploration features and also help in lowering the computational costs.
Appendix A: Parallel tempering MCMC

Parallel tempering MCMC features massive parallelism with enhanced exploration capabilities. It features several replicas with slight variations in the acceptance criteria through relaxation of the likelihood with a temperature ladder that affects the replica sampling acceptance criterion. The replicas associated with higher temperature levels have more chance in accepting weaker proposals, which could help in escaping a local minimum. Given an ensemble of \( M \) replicas defined by a temperature ladder, we define the state by \( X = x_1, x_2, \ldots, x_M \), where \( x_i \) is the replica at temperature level \( T_i \).

We construct a Markov chain to sample proposal \( x_i^* \) from a proposal distribution \( q_i(\cdot|x_i) \). The Metropolis–Hastings ratio at temperature level \( T_i \) is given by

\[
L_{\text{local}}(x_i \rightarrow x_i^*) = \exp \left( -\frac{1}{T_i} \left( L(x_i^*) - L(x_i) \right) \right), \tag{A1}
\]

where \( L \) represents the likelihood at the local replica. We accept the new state with probability, \( \min(1, L_{\text{local}}(x_i \rightarrow x_i^*)) \). The detailed balance condition holds for each MCMC replica; therefore, it holds for the ensemble system (Calderhead, 2014).

In the replica transition phase, we consider the exchange of the current state between two neighbouring replicas based on the Metropolis–Hastings acceptance criteria. Hence, given a probability \( \alpha \), we exchange a pair of replica defined by two neighbouring temperature levels, \( T_i \) and \( T_{i+1} \).

\[
x_i \leftrightarrow x_{i+1} \tag{A2}
\]

The exchange of neighbouring replicas provides an efficient balance between local and global exploration (Sambridge, 2013). The temperature ladder and replica exchange have been of focus of investigation in the past (Calvo, 2005; Liu et al., 2005; Bittner et al., 2008; Patriksson and van der Spoel, 2008), and there is a consensus that they need to be tailored for different types of problems given by their likelihood landscape. In this paper, the selection of temperature spacing between the replicas is carried out using a geometric spacing methodology (Voudsen et al., 2015), given as follows:

\[
T_i = T_{\text{max}}^{(i-1)/(M-1)}, \tag{A3}
\]

where \( i = 1, \ldots, M \) and \( T_{\text{max}} \) is maximum temperature, which is user defined and dependent on the problem.

A1 Training the neural network surrogate model

We note that stochastic gradient descent maintains a single learning rate for all weight updates, and typically the learning rate does not change during the training. Adam (adaptive moment estimation) learning algorithm (Kingma and Ba, 2014) differs from classical stochastic gradient descent, as the learning rate is maintained for each network weight and separately adapted as learning unfolds. Adam computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. Adam features the strengths of root mean square propagation (RMSprop) and adaptive gradient algorithm (AdaGrad) (Kingma and Ba, 2014; Duchi et al., 2011). Adam has shown better results when compared to stochastic gradient descent, RMSprop, and AdaGrad. Hence, we consider Adam as the designated algorithm for the neural-network-based surrogate model. We formulate the learning procedure through weight update for iteration number \( t \) for weights \( W \) and biases \( b \) by

\[
\Theta_{t-1} = [W_{t-1}, b_{t-1}]
\]

\[
g_t = \nabla_{\Theta} J_t(\Theta_{t-1})
\]

\[
m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t
\]

\[
v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2
\]

\[
\hat{m}_t = m_t / (1 - \beta_1^t)
\]

\[
\hat{v}_t = v_t / (1 - \beta_2^t)
\]

\[
\Theta_t = \Theta_{t-1} - \alpha \cdot \hat{m}_t / \left( \sqrt{\hat{v}_t} + \epsilon \right), \tag{A4}
\]

where \( m_t \) and \( v_t \) are the, respectively, first and second moment vectors for iteration \( t \); \( \beta_1 \) and \( \beta_2 \) are constants \( \in [0, 1] \); \( \alpha \) is the learning rate, and \( \epsilon \) is a close to zero constant.
Code availability. We provide open-source code along with data and sample results to motivate further work in this area: https://github.com/intelligentEarth/surrogateBayeslands (last access: 6 July 2020, intelligentEarth, 2020); https://doi.org/10.5281/zenodo.3892277 (Chandra, 2020).

Author contributions. RC led the project and contributed to writing the paper and designing the experiments. DA contributed by running experiments and providing documentation of the results. AK contributed in terms of programming, running experiments, and providing documentation of the results. RDM contributed by managing the project, writing the paper, and providing analysis of the results.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. We would like to thank Konark Jain for technical support. R. Dietmar Müller and Danial Azam were supported by the Australian Research Council (grant no. IH130200012). We sincerely thank the reviewers for their comments that helped us in improving the paper.

Financial support. This research has been supported by the University of Sydney (grant no. SREI Grant 2017-2018) and the Australian Research Council (grant no. IH130200012).

Review statement. This paper was edited by Richard Neale and reviewed by two anonymous referees.

References


https://doi.org/10.5194/gmd-13-2959-2020

Geosci. Model Dev., 13, 2959–2979, 2020


Geosci. Model Dev., 13, 2959–2979, 2020

R. Chandra et al.: Surrogate-assisted Bayeslands

https://doi.org/10.5194/gmd-13-2959-2020