Supplement of Geosci. Model Dev., 7, 1933–1943, 2014
http://www.geosci-model-dev.net/7/1933/2014/
doi:10.5194/gmd-7-1933-2014-supplement

Geoscientific Model Development
Open Access

Supplement of

Probabilistic calibration of a Greenland Ice Sheet model using spatially resolved synthetic observations: toward projections of ice mass loss with uncertainties

W. Chang et al.

Correspondence to: W. Chang (wonchang@psu.edu)

1. Gaussian process emulator for principal components

In this section, we outline our statistical approach for ice sheet model emulation using Gaussian process (GP) models and principal component (PC) analysis (often referred to as empirical orthogonal functions, EOFs). Our approach follows Chang et al. (2014) in that we summarize the ice sheet model runs as PCs and calibrate the ice sheet parameters based on GP emulators for PCs. Our description of methods below therefore also closely follows the notation and description in Chang et al. (2014). By decomposing spatial patterns into a small number of variables representing important characteristics of model runs, our approach drastically increases computational efficiency without causing significant information loss.

We denote the number of model runs by p and the number of spatial locations spatial locations by n. For the SICOPOLIS model output (from Applegate et al. 2012) we use here, $p = 99$ and $n = 264$. Note that the original ensemble in (Applegate et al. 2012) contains 100 model runs, but we leave one model run out to construct the synthetic truth. We let $Y(\boldsymbol{\theta}, \mathbf{s})$ denote the ice thickness from the ice sheet model at a parameter setting $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_5)^T$ and a spatial location $\mathbf{s}$. We let $\mathbf{s}_1, \ldots, \mathbf{s}_n$ be the spatial locations of the model grid points and $\mathbf{Y}(\boldsymbol{\theta}) = (Y(\boldsymbol{\theta}, \mathbf{s}_1), \ldots, Y(\boldsymbol{\theta}, \mathbf{s}_n))$ be the vector of model output at a parameter setting $\boldsymbol{\theta}$. Let $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p$ be the vectors of input parameters for our model. $\mathbf{Y}$ is an $n \times p$ matrix of the ice sheet model output where its rows correspond to spatial locations and columns to parameter settings, i.e.

$$\mathbf{Y} = \begin{pmatrix} Y(\boldsymbol{\theta}_1, \mathbf{s}_1), & Y(\boldsymbol{\theta}_2, \mathbf{s}_1), & \ldots, & Y(\boldsymbol{\theta}_p, \mathbf{s}_1) \\ Y(\boldsymbol{\theta}_1, \mathbf{s}_2), & Y(\boldsymbol{\theta}_2, \mathbf{s}_2), & \ldots, & Y(\boldsymbol{\theta}_p, \mathbf{s}_2) \\ \vdots, & \vdots, & \ddots, & \vdots \\ Y(\boldsymbol{\theta}_1, \mathbf{s}_n), & Y(\boldsymbol{\theta}_2, \mathbf{s}_p), & \ldots, & Y(\boldsymbol{\theta}_p, \mathbf{s}_n) \end{pmatrix}.$$

Similarly, $Z(\mathbf{s})$ denotes the observed ice sheet thickness at a location $\mathbf{s}$, and $\mathbf{Z} = (Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_n))^T$ is the $n \times 1$ vector of the observational data. For the matrices and the statistical parameters used in the following sections, the subscript y indicates that a symbol is used for the emulation model, while the subscript d shows that a symbol is for the discrepancy model.

2. Principal component analysis for model output

The first step is summarizing the model output by principal component analysis. Using principal components in emulation and calibration has the following advantages: First, the principal components are independent of each other under Gaussian assumption, and this independence allows a substantial gain in computational efficiency because it enables us to treat each of the principal components separately in the emulation stage. Second, the principal components are the "best" summary of the model runs in the sense that they show the clearest contrast between model runs, among all possible linear combinations of model runs (see below for more details).

Following the standard procedure of principal component analysis, the column means are subtracted from each element in the corresponding columns such that each column is centered on zero. We apply singular value decomposition to this centered output matrix to find the scaled principal basis vectors $\mathbf{k}_1 = \sqrt{\lambda_1}\mathbf{e}_1, \ldots, \mathbf{k}_p = \sqrt{\lambda_p}\mathbf{e}_p$, where $\lambda_1 > \lambda_2 > \cdots > \lambda_p$ and $\mathbf{e}_1, \ldots, \mathbf{e}_p$ are ordered eigenvalues and their eigenvectors respectively. Each eigenvalue represents the explained variation for the corresponding principal component. We keep only the first $J \ll p$ PCs with the largest explained variation (i.e. the largest eigenvalues) to minimize the information loss due to dimension reduction. The principal components for model output can be computed by

$$\mathbf{Y}^R = (\mathbf{K}_y^T\mathbf{K}_y)^{-1}\mathbf{K}_y^T\mathbf{Y} = (\mathbf{Y}_1^R \ldots \mathbf{Y}_J^R)^T$$

where $\mathbf{K}_y = (\mathbf{k}_1, \ldots, \mathbf{k}_J)$ is the principal basis matrix. $\mathbf{Y}_i^R = (Y_i^R(\boldsymbol{\theta}_1), \ldots, Y_i^R(\boldsymbol{\theta}_p))^T$ is the $p \times 1$ vector of the ith principal components, and $Y_i^R(\boldsymbol{\theta}_j)$ is the ith principal component at the parameter setting $\boldsymbol{\theta}_j$. The resulting matrix $\mathbf{Y}^R$ is the summarized output matrix with rows for PCs and columns for parameter settings . The procedure reduces the size of the data from $n \times p$ to $J \times p$.

3. Gaussian process emulator

We emulate the ice sheet model output using Gaussian processes (GP), a fast method for probabilistic interpolation between existing model runs (Sacks et al. 1989; Kennedy and O'Hagan 2001; Higdon et al. 2008; Drignei et al. 2008; Rougier 2008; Goldstein and Rougier 2009; Bhat et al. 2012; Holden et al. 2010; Lee et al. 2011; Olson et al. 2012, 2013; McNeall et al. 2013; Williamson et al. 2013). The GP emulator approach yields a flexible approximation without requiring detailed physical information on the ice sheet model, unlike linear regression-based emulators (cf. Piani et al. 2005). By interpolating existing model runs at different parameter settings, a GP emulator provides a reasonable approximation to the original model unless the model output abruptly changes in the input parameter space. Interpolation using GP emulator is essentially kriging in the input parameter space; the interpolator is a random process with (i) a mean term that is the optimal interpolation between ice sheet model runs in terms of the expected mean squared error and (ii) a variance term that quantifies the uncertainty of the interpolation. In this section we first describe the basics of the GP emulator approach and explain how to construct an emulator for principal components.

Before describing emulation for principal components, we illustrate the GP emulation approach with $n = 1$, where the ice model output at each parameter setting is a scalar. For ease of exposition we further simplify the case by assuming that the input parameter is also a scalar. The collection of the model output is simply a vector $\mathbf{Y} = (Y(\theta_1), \ldots, Y(\theta_p))^T$ at the parameter settings $\theta_1, \ldots, \theta_p$. The GP emulator for the model output is then given by

$$\mathbf{Y} \sim N(\mathbf{0}, \Sigma),$$

with a $p \times p$ covariance matrix Σ. While linear regression finds a trend that fits the data well using a mean function, the GP model interpolates the data using a covariance structure. Unlike the linear regression, which requires careful specification of the mean function along with various statistical assumptions when dealing with highly nonlinear processes such as ice

model outputs, the GP model can handle such processes using a relatively simple covariance function. Only required assumption for the GP model is that the model output is a smoothly varying curve in the parameter space without too many abrupt changes. A common choice for the covariance function for a GP emulator is the squared exponential covariance, which defines the (i, j)th element of Σ as

$$\mathrm{Cov}(Y(\theta_i), Y(\theta_j)|\zeta, \kappa, \phi) = \zeta 1(\theta_i = \theta_j) + \kappa \exp\left(-\left(\frac{\theta_i - \theta_j}{\phi}\right)^2\right),$$

with all positive ζ, κ and ϕ. The range parameter ϕ defines how fast the model output is changing as the value of the parameter changes. The partial sill specifies the overall magnitude of the process, and the nugget parameter ζ captures the variability caused by various sources other than input parameters, such as the effect of initial conditions.

We now turn our attention to emulation for the principal components. Because the principal components are uncorrelated with each other by construction, we can model each of them separately using independent GPs. Note that this procedure ignores the dependence between the principal components that is not captured by the covariances. However, according to our cross-validation experiments for various models including SICOPOLIS, the emulator based on this assumption usually provides an accurate approximation to the original model (see e.g. Chang et al. 2014, Figure 2). We model each $\mathbf{Y}_i^R$ using a GP with mean zero and covariance determined by the following squared exponential covariance function:

$$\mathrm{Cov}(Y_i^R(\boldsymbol{\theta}_j), Y_i^R(\boldsymbol{\theta}_k); \zeta_i, \kappa_{y,i}, \phi_i) = \zeta_i 1(\boldsymbol{\theta}_j = \boldsymbol{\theta}_k) + \kappa_{y,i} \exp\left(-\sum_{l=1}^{5}\left(\frac{\theta_{jl} - \theta_{kl}}{\phi_{il}}\right)^2\right),$$

where $\zeta_i, \kappa_{y,i}, \phi_{i1}, \ldots, \phi_{i5} > 0$ are covariance parameters, θ_{jl} is the lth element of $\boldsymbol{\theta}_j$, and $1(\cdot)$ is the index function. The covariance parameters $(\zeta_1, \kappa_{1,y}, \phi_{11}, \ldots, \phi_{15}), \ldots, (\zeta_J, \kappa_{y,J}, \phi_{J1}, \ldots, \phi_{J5})$ are estimated by maximum likelihood estimation (MLE). Our emulator, denoted by $J \times 1$ vector-valued function $\boldsymbol{\eta}(\boldsymbol{\theta}, \mathbf{Y}^R)$, is the predictive distribution of PCs at an untried parameter setting $\boldsymbol{\theta}$ defined by the fitted GPs. Using the PC emulator, we can also emulate the original model transect by computing $\mathbf{K}_y\boldsymbol{\eta}(\boldsymbol{\theta}, \mathbf{Y}^R)$.

Note that our approach allows significant improvements in computational efficiency. Without any dimension reduction, the computational cost for a single likelihood evaluation scales as $\mathcal{O}(n^3p^3)$, which corresponds to a few hours of computing time. Thus, application of any numerical methods requiring repeated evaluation of the likelihood function is computationally prohibitive if no dimensional reduction is performed. Our approach decreases the computational complexity to $\mathcal{O}(Jp^3)$, and this is a reduction from 3.18×10^{14} flops to 1.56×10^8 flops when using 10 principal components ($J = 10$). Using 10 principal components captures more than 90% of the variation in the model output, and we have confirmed that using more than 10 principal components does not significantly improve the emulation accuracy by cross-validation.

4. Model parameter calibration

In this section, we formulate the probability model for calibration using the PC emulator constructed above and explain the inference procedure for the model parameters using Markov chain Monte Carlo (MCMC). The main goal here is to estimate the input parameters using a probability model that combines the model for the emulator described above and a discrepancy term that detects the systematic model-observation discrepancy. The calibration model operates in a reduced dimensional space defined by the principal components computed above and the kernel convolution (Higdon et al. 2008, explained below) and allows us to avoid computational issues for dealing with high-dimensional data (Chang et al. 2014). Based on this model, one can efficiently sample from the posterior density of input parameters by MCMC.

We assume that the observational dataset is emulator output contaminated by model discrepancy and observational error;

$$\mathbf{Z} = \mathbf{K}_y\boldsymbol{\eta}(\boldsymbol{\theta}^*, \mathbf{Y}^R) + \mathbf{K}_d\boldsymbol{\nu} + \boldsymbol{\epsilon}, \tag{S1}$$

where $\boldsymbol{\theta}^*$ is the best fit input parameter setting (Bayarri et al. 2007; Rougier 2007) for the

observational data, and $\boldsymbol{\epsilon} \sim N(0, \sigma^2 \mathbf{I}_n)$ is the observational error with variance $\sigma^2 > 0$. $\mathbf{K}_d\boldsymbol{\nu}$ is the model-observation discrepancy picking up systematic differences between the model and the observations (cf. Bayarri et al. 2007; Bhat et al. 2012), where $\mathbf{K}_d$ is a kernel basis matrix relating the spatial locations $\mathbf{s}_1, \ldots, \mathbf{s}_n$ to J_d knot locations $\mathbf{a}_1, \ldots, \mathbf{a}_{J_d}$, and $\boldsymbol{\nu} \sim N(\mathbf{0}, \kappa_d \mathbf{I}_{J_d})$ is the vector of knot processes, a set of random variables assigned to each of the knot locations with variance $\kappa_d > 0$. Our choice for the kernel function is an exponential covariance given by

$$\{\mathbf{K}_d\}_{ij} = \exp\left(-\frac{|\mathbf{s}_i - \mathbf{a}_j|}{\phi_d}\right),$$

with $\phi_d > 0$. The variance parameter κ_d is subject to inference, and the correlation parameter ϕ_d is pre-specified by expert judgment. In our implementation, we choose ϕ_d as 5% of the maximum distance between the spatial locations on the model grid to yields a sufficiently flexible discrepancy pattern. Fixing the range parameter not only reduces the computational cost for likelihood computation but also improves the identifiability between the input parameters and the discrepancy process. Note that the kernel basis often needs to be substituted by its scaled principal basis (eigenvectors) to improve identifiability; see Chang et al. (2014) for a more detailed discussion. We used the 30 leading components for $\mathbf{K}_d$ in our implementation. We apply a similar dimension reduction described in the previous section to find $\mathbf{Z}^R$, a summary of the observed transect as follows:

$$\mathbf{Z}^R = (\mathbf{K}^T\mathbf{K})^{-1}\mathbf{K}^T\mathbf{Z}, \tag{S2}$$

and therefore the model for $\mathbf{Z}^R$ can be written as

$$\mathbf{Z}^R \sim N\left(\begin{pmatrix} \boldsymbol{\mu}_{\boldsymbol{\eta}} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \Sigma_{\boldsymbol{\eta}} & \mathbf{0} \\ \mathbf{0} & \kappa_d \mathbf{I}_{J_d} \end{pmatrix} + \sigma^2(\mathbf{K}^T\mathbf{K})^{-1}\right),$$

where $\mu_{\boldsymbol{\eta}}$ and $\Sigma_{\boldsymbol{\eta}}$ are the mean and covariance, respectively, of the emulator $\boldsymbol{\eta}(\boldsymbol{\theta}^*, \mathbf{Y}^R)$, and $\mathbf{K} = (\mathbf{K}_y\ \mathbf{K}_d)$.

The parameters to be estimated in the calibration model are the ice sheet model input parameters $\boldsymbol{\theta}^*$, the discrepancy parameter κ_d, and the observational error variance σ^2. We

also re-estimate the partial sill parameters $\boldsymbol{\kappa}_y = (\kappa_{y,1}, \ldots, \kappa_{y,J})$ for the emulator (Bayarri et al. 2007; Bhat et al. 2012; Chang et al. 2014). This allows the emulator process to be re-scaled to better match the observational data. We define the posterior density based on the likelihood function given by (S2) denoted by $\ell(\mathbf{Z}^R|\boldsymbol{\theta}^*, \boldsymbol{\kappa}_y, \kappa_d, \sigma^2, \mathbf{Y}^R)$ and some standard prior specifications denoted by $f(\boldsymbol{\theta}^*)$, $f(\boldsymbol{\kappa}_y)$, $f(\kappa_d)$, and $f(\sigma^2)$ (Higdon et al. 2008; Chang et al. 2014). Each of the input parameters in $\boldsymbol{\theta}^*$ receives a flat prior on a broad range determined by model ensemble design and physical knowledge. The observational error variance σ^2 and the variance for the discrepancy κ_d have non-informative inverse-gamma priors with small shape parameters. We specify somewhat informative priors for $\kappa_{y,1}, \ldots, \kappa_{y,J}$ by specifying a large shape parameter in order to avoid numerical instability and identifiability issues (Higdon et al. 2008). The posterior distribution resulting from the above model is

$$\pi(\boldsymbol{\theta}^*, \boldsymbol{\kappa}_y, \kappa_d, \sigma^2|\mathbf{Z}^R, \mathbf{Y}^R) \propto \ell(\mathbf{Z}^R|\boldsymbol{\theta}^*, \boldsymbol{\kappa}_y, \kappa_d, \sigma^2, \mathbf{Y}^R) f(\boldsymbol{\theta}^*) f(\boldsymbol{\kappa}_y) f(\kappa_d) f(\sigma^2),$$

where

$$\begin{aligned}
\ell(\mathbf{Z}^R|\boldsymbol{\theta}^*, \boldsymbol{\kappa}_y, \kappa_d, \sigma^2, \mathbf{Y}^R) &\propto \left|\Sigma_{\boldsymbol{\eta}} + \mathbf{K}^T\mathbf{K}\sigma^2\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\mathbf{Z}^{RT}\left(\Sigma_{\boldsymbol{\eta}} + \mathbf{K}^T\mathbf{K}\sigma^2\right)^{-1}\mathbf{Z}^R\right) \\
f(\boldsymbol{\theta}^*) &\propto 1(\boldsymbol{\theta}^* \in \Theta),\ \Theta \text{ represents the range of } \boldsymbol{\theta}, \\
f(\boldsymbol{\kappa}_y) &\propto \prod_{i=1}^{J} \kappa_{y,i}^{-a_{y,i}-1} \exp\left(-\frac{b_{y,i}}{\kappa_{y,i}}\right),\ a_{y,1}, \ldots, a_{y,J}, b_{y,1}, \ldots, b_{y,J} > 0 \\
f(\kappa_d) &\propto \kappa_d^{-a_d-1} \exp\left(-\frac{b_d}{\kappa_d}\right),\ a_d, b_d > 0 \\
f(\sigma^2) &\propto \sigma^{-2(a_\sigma+1)} \exp\left(-\frac{b_\sigma}{\sigma^2}\right),\ a_\sigma, b_\sigma > 0.
\end{aligned}$$

For each i, we set $a_{y,i} = 50$ and choose $b_{y,i}$ such that the mode of the prior density $b_{y,i}/(a_{y,i}+1)$ coincides with the MLE of $\kappa_{y,i}$ computed in the emulation stage. For other parameters, we impose vague priors by setting $a_d = 2$, $b_d = 3$, $a_\sigma = 2$, and $b_\sigma = 3$.

The synthetic observations used in our perfect model experiment are constructed by superimposing a random error generated from a Gaussian process model on the assumed true ice sheet status. To make our experiment more realistic, the discrepancy process is generated from a different model to the discrepancy term that we use in the equation (S1).

The covariance function that we use for the Gaussian process model for simulated discrepancy here is a squared exponential covariance having range of 2100 km, partial sill of 2500 m, and a nugget of 1 m. Our choice for the simulated discrepancy process is based on the following two general assumptions: (i) the discrepancy is statistically identifiable from the emulator process, and (ii) SICOPOLIS has an enough skill to reproduce the observed ice profile. (i) is related to the value of the range parameter, which controls the effective distance at which two spatial locations are uncorrelated. To ensure that the discrepancy process is identifiable from the emulator process, we set the range parameter to be very large (80% of the spatial range of the model output) so that the discrepancy operates in a different spatial scale to the emulator process. (ii) is related to the value of the partial sill, which defines the magnitude of the discrepancy. Here we let the value of the partial sill to be reasonably small to simulate the situation in which the structural error is not large and therefore SICOPOLIS can reproduce the observed ice profile reasonably well. Note that calibration based on any framework, including our approach, may yield misleading results if any of the assumptions are violated. For example, if the discrepancy process operates over a similar spatial scale to the emulator process (i.e. (i) does not hold), the discrepancy causes identifiability issues and hence introduces a significant bias in the calibration result. If the magnitude of the discrepancy is too large (i.e. (ii) does not hold) compared to the variation between model outputs, the calibration results will become essentially non-informative (i.e., the likelihood then has little effect on the posterior distribution). Note that these are common issues for calibration methods in general. The curves in Figure S1 show that realizations from the discrepancy process are clearly identifiable from the difference between model runs.

Based on the pseudo-observations, we infer the parameters using the MCMC sample from the above posterior distribution obtained by the Metropolis-Hastings algorithm (cf. Higdon et al. 2009). In particular, we infer the input parameters in $\boldsymbol{\theta}^*$ by investigating their marginal density $\pi(\boldsymbol{\theta}^*|\mathbf{Z}^R, \mathbf{Y}^R)$. In our perfect model experiment, we obtained 300,000 draws using block updating when estimating the full joint density of all five parameters. The

computing time takes about eight hours on a single high-performance core. For inference on individual input parameter, only 300,000 draws using block updating is sufficient. In both cases, we confirmed that the Monte Carlo chain is well-mixed by comparing the densities of the first half of the chain with the entire chain. We find the probability density of the input parameters via kernel density estimation for the MCMC sample. The estimated density can be easily plotted for visual analysis as shown in Figures 3 and 4. Note that ignoring the spatially correlated discrepancy results in a notably biased calibration results in our perfect model experiment. See Figure S2 for a comparison of posterior densities with and without the discrepancy term.

5. Ice volume change projection based on calibrated parameters

One important purpose of parameter calibration is making better projections for the future ice sheet mass loss. In our illustrative example, the variable that we want to project is the ice volume change from present to 2100 in meters of sea level equivalent. For each model run, we compute the ice volume change by subtracting the current ice volume from the future ice volume. Making future projections based on calibration results requires a function that relates input parameter values $\boldsymbol{\theta}^*$ to future changes in ice sheet volume. We construct such a function by interpolating the ice volume changes for the 100 input parameter settings in our ensemble. Using this function we convert the MCMC chain for the input parameters generated in the previous section into Monte Carlo sample for the future ice volume changes.

Among many possible choices for the interpolator, we use the Gaussian process emulator similar to the model described in 3. More specifically, we fit a Gaussian process model for the ice volume change over the input parameter space with zero-mean and the covariance

function

$$Cov(\Delta v(\boldsymbol{\theta}_j), \Delta v(\boldsymbol{\theta}_k); \zeta^{vol}, \kappa^{vol}, \phi^{vol}) = \zeta^{vol} 1(\boldsymbol{\theta}_j = \boldsymbol{\theta}_k) + \kappa^{vol} \exp\left(- \sum_{l=1}^{5} \frac{|\theta_{jl} - \theta_{kl}|}{\phi_l^{vol}} \right),$$

for any given design points $\boldsymbol{\theta}_j$ and $\boldsymbol{\theta}_k$ $(j, k = 1, \cdots, 100)$, where $\Delta v(\boldsymbol{\theta})$ is the volume change at a parameter setting $\boldsymbol{\theta}$, and $\zeta^{vol}.\kappa^{vol}, \phi_1^{vol}, \ldots, \phi_5^{vol} > 0$ are the covariance parameters that need to be estimated via MLE. The resulting function can predict ice volume change at any given value of $\boldsymbol{\theta}$ as the conditional mean given by the standard kriging approach (Cressie 1993). Figure S3 shows the marginal surface of the projection as a function of input parameters. To validate the emulator constructed here, we have conducted leave-5%-out cross-validation. The mean error rate, computed by dividing the RMS by the overall mean, is around 16%; the error rate is a little higher than the heuristic upper limit for the generally acceptable emulation error (10%) due to the irregular behavior of the volume change surface.

We obtain a Monte Carlo sample of ice volume projections by supplying the posterior sample of the calibrated parameters to the interpolation function. Each element of the posterior sample is converted to ice volume change. The predictive density of the ice volume projection can be found by applying kernel density estimation. We find the prior density of the projections in the same manner; we convert the design points of the existing model runs into the ice volume changes and compute the predictive density for it using kernel density estimation.

6. Cross-validation

To investigate (i) whether the perfect model experiment results shown in the main text are sensitive to the values of input parameters assumed as the synthetic truth, and (ii) whether the prediction intervals for ice volume projections generated from our method have the right coverage, we have conducted leave-one-out cross-validation across all input parameter settings in the ensemble. In other words, we have repeated the same perfect model experiment described in the previous sections for all 100 possible different synthetic truths.

We summarize the cross-validation results for emulation and calibrated projections in Figure S4 and Figure S5, respectively.

The results in Figure S4 show that our emulator can predict the model output reasonably well across all input parameter settings. The predicted ice volume thickness profiles are concentrated around the diagonal line that connects the lower left and the upper right corners of the plot, and hence the emulator can predict the model output reasonably well for most input parameter settings. Note that leave-one-out cross-validation is already rigorous enough in our case due to the sparsity of the design points (100 points in 5-dimensional space) for the input parameters in our ensemble. We have also conducted leave-10-out cross-validation for emulation and the results are essentially the same (not shown).

The plots in Figure S5 show that the prediction intervals generated from our approach achieve the nominal coverage level only when the modern ice volume generated by the synthetic truth is close enough to the observed volume (i.e. within 10% of the observed value). The width of the prediction interval also varies considerably across the different assumed truths. Therefore, consistent with the findings in McNeall et al. (2013), selection of the assumed truth affects the calibration performance. Another important observation is that including the discrepancy term reduces the overconfidence issue that occurs when the synthetic truths are outside of the 90-110% range. The prediction intervals are overconfident when the synthetic truth is outside of this range because the coverage is consistently less than 95%. Including the discrepancy term reduces this issue in some degree since it make the actual coverage closer to the nominal coverage when the synthetic truth yields the modern ice volume that is within at most 70% of the observed volume. However, this correction effect is not sufficient to make the prediction intervals achieve the nominal coverage.

The cross-validation results allow us to examine the interaction between input parameters across all possible choices of the synthetic truth. We have computed the rank correlations between the input parameters across all 100 ensemble members and summarized their distributions in Figure S6. From the shapes of the densities we can identify five pairs of parameters

that tend to be more negatively correlated: (i) the flow factor and the snow PDD factor, (ii) the flow factor and the geothermal heat flux, (iii) the basal sliding factor and the ice PDD factor, (iv) the geothermal heat flux and the ice PDD factor, and (v) the ice PDD factor and the snow PDD factor.

7. Summary

We describe an ice sheet model calibration approach based on PCs of the model output and the observational data. We build a GP emulator for the PCs of the model output as a fast approximation to the ice sheet model. The calibration model links the observed PCs with the input parameters using the GP emulator while taking the systematic discrepancy into account. We infer the input parameters along with other statistical parameters in the calibration model using MCMC. Combined with projections generated by the ice sheet model, the resulting posterior density of the parameters provide calibrated probabilistic projections of the future ice sheet volume changes. Our cross-validation results across all input parameter settings in the ensemble show that the probabilistic projections achieves the nominal coverage rate when the synthetic truth yields a modern ice volume that is close to the observed volume.

REFERENCES

Applegate, P., N. Kirchner, E. Stone, K. Keller, and R. Greve, 2012: An assessment of key model parametric uncertainties in projections of Greenland Ice Sheet behavior. *The Cryosphere*, **6 (3)**, 589–606.

Bayarri, M., et al., 2007: Computer model validation with functional output. *The Annals of Statistics*, **35 (5)**, 1874–1906.

Bhat, K., M. Haran, R. Olson, and K. Keller, 2012: Inferring likelihoods and climate system characteristics from climate models and multiple tracers. *Environmetrics*, **23 (4)**, 345–362.

Chang, W., M. Haran, R. Olson, and K. Keller, 2014: Fast dimension-reduced climate model calibration and the effect of data aggregation. *The Annals of Applied Statistics*, **8 (2)**, 649–673.

Cressie, N., 1993: *Statistics for Spatial Data*. Wiley, New York.

Drignei, D., C. Forest, and D. Nychka, 2008: Parameter estimation for computationally intensive nonlinear regression with an application to climate modeling. *The Annals of Applied Statistics*, **2 (4)**, 1217–1230.

Goldstein, M. and J. Rougier, 2009: Reified bayesian modelling and inference for physical systems. *Journal of Statistical Planning and Inference*, **139 (3)**, 1221–1239.

Higdon, D., J. Gattiker, B. Williams, and M. Rightley, 2008: Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, **103 (482)**, 570–583.

Higdon, D., C. Reese, J. Moulton, J. Vrugt, and C. Fox, 2009: Posterior exploration for computationally intensive forward models. *Handbook of Markov Chain Monte Carlo*, S. Brooks, A. Gelman, G. Jones, and X. Meng, Eds., CRC Press, New York, 401–418.

Holden, P. B., N. Edwards, K. Oliver, T. Lenton, and R. Wilkinson, 2010: A probabilistic calibration of climate sensitivity and terrestrial carbon change in genie-1. *Climate dynamics*, **35 (5)**, 785–806.

Kennedy, M. and A. O'Hagan, 2001: Bayesian calibration of computer models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, **63 (3)**, 425–464.

Lee, L., K. Carslaw, K. Pringle, G. Mann, and D. Spracklen, 2011: Emulation of a complex global aerosol model to quantify sensitivity to uncertain parameters. *Atmospheric Chemistry and Physics*, **11 (23)**, 12 253–12 273.

McNeall, D., P. Challenor, J. Gattiker, and E. Stone, 2013: The potential of an observational data set for calibration of a computationally expensive computer model. *Geoscientific Model Development*, **6 (5)**, 1715–1728.

Olson, R., R. Sriver, M. Goes, N. Urban, H. Matthews, M. Haran, and K. Keller, 2012: A climate sensitivity estimate using Bayesian fusion of instrumental observations and an Earth System model. *Journal of Geophysical Research: Atmospheres*, **117 (D4)**, D04 103, doi:10.1029/2011JD016 620.

Olson, R., R. Sriver, M. Haran, W. Chang, N. Urban, and K. Keller, 2013: What is the effect of unresolved internal climate variability on climate sensitivity estimates? *Journal of Geophysical Research: Atmospheres*, **118 (10)**, 4348–4358.

Piani, C., D. Frame, D. Stainforth, and M. Allen, 2005: Constraints on climate change from a multi-thousand member ensemble of simulations. *Geophysical Research Letters*, **32 (23)**.

321 Rougier, J., 2007: Probabilistic inference for future climate using an ensemble of climate
322 model evaluations. *Climatic Change*, **81 (3–4)**, 247–264.

323 Rougier, J., 2008: Efficient emulators for multivariate deterministic functions. *Journal of*
324 *Computational and Graphical Statistics*, **17 (4)**, 827–843.

325 Sacks, J., W. Welch, T. Mitchell, and H. Wynn, 1989: Design and analysis of computer
326 experiments. *Statistical Science*, **4 (4)**, 409–423.

327 Williamson, D., M. Goldstein, L. Allison, A. Blaker, P. Challenor, L. Jackson, and K. Ya-
328 mazaki, 2013: History matching for exploring and reducing climate model parameter space
329 using observations and a large perturbed physics ensemble. *Climate Dynamics*, **41 (7–8)**.

List of Tables

TABLE 1. Summary of notation

Symbol	Definition
p	number of model runs
n	number of spatial locations for model grid
θ_{il}	value of lth input parameter in $\boldsymbol{\theta}_i$ for ith model run
$\boldsymbol{\theta}_i$	input parameter setting for ith model run
$\mathbf{s}_j$	jth location on computer model grid
$Y(\boldsymbol{\theta}_i, \mathbf{s}_j)$	model output at location $\mathbf{s}_j$ for input parameter setting $\boldsymbol{\theta}_i$
$\mathbf{Y}$	$p \times n$ matrix of all model ensemble
J	number of principal components used in emulation
$\mathbf{Y}_i^R$	principal components for ith model run, $(Y_i^R(\boldsymbol{\theta}_1), \ldots, Y_i^R(\boldsymbol{\theta}_p))$
$\mathbf{Y}^R$	$p \times J$ matrix of all principal components, $(\mathbf{Y}_1^R, \ldots, \mathbf{Y}_p^R)^T$
$\mathbf{k}_j$	jth (scaled) principal component basis vector
$\mathbf{K}_y$	$n \times J$ principal component basis matrix $(\mathbf{k}_1, \ldots, \mathbf{k}_J)$
$\boldsymbol{\eta}(\boldsymbol{\theta}, \mathbf{Y}^R)$	emulator for principal components
$\boldsymbol{\theta}^*$	true or fitted value of computer model parameter for observational data
ζ_i	nugget for ith principal component emulator
$\kappa_{y,i}$	partial sill for ith principal component emulator
ϕ_{il}	range for ith principal component and lth input parameter
$\mathbf{K}_d$	kernel basis matrix for discrepancy
$\mathbf{a}_j$	jth knot location for kernel basis
$\boldsymbol{\nu}$	vector of knot processes
J_d	number of knot locations
κ_d	variance of knot processes in $\boldsymbol{\nu}$
ϕ_d	range parameter for discrepancy kernel basis in $\mathbf{K}_d$
$\mathbf{Z}$	vector of observational data
$\mathbf{Z}^R$	principal components for observational data
$\boldsymbol{\epsilon}$	vector of observational errors
σ^2	variance of observational errors
$\mathbf{K}$	basis matrix for observational data, $\mathbf{K} = (\mathbf{K}_y\ \mathbf{K}_d)$
$\Delta v(\boldsymbol{\theta})$	volume change projection for input parameter setting $\boldsymbol{\theta}$
ζ^{vol}	nugget for volume change emulator
κ^{vol}	partial sill for volume change emulator
ϕ_l^{vol}	lth range for volume change emulator

List of Figures

S4 Leave-one-out cross-validation results for the emulation performance. Each grey curve shows the comparison of zonal mean ice thickness transects from the model output and that from the emulator output for each parameter setting. Each boxplot shows the distribution of emulator output for each of the evenly spaced bins that span the range of true model output. In spite of the fact that our design points for parameter settings are quite sparse (100 runs in 5-dimensional space) most of the curves are concentrated around 1:1 line connecting the lower left and upper right corners of the plot, indicating that our emulator can reconstruct the original model output reasonably well across the input parameter settings. 24

S5 Leave-one-out cross-validation results for ice volume change projections across all 100 input parameter settings as the synthetic truth. The left penal shows 95% prediction intervals for ice volume change projections across all 100 perfect model experiments conducted for cross-validation. If the interval covers the 1:1 line connecting the lower left and upper right corners of the plot, the 95% prediction interval includes the ice volume projection given by the synthetic truth. The right penal shows the coverage of those prediction intervals as a function of allowed range for the ice volume in 2005 AD relative to the observed ice volume. As going from left to right, the synthetic truths used in computing the coverages include more "unrealistic" ones in terms of modern ice volume. The numbers above the solid black line show how many synthetic truths fall into the given ice volume range. The plot shows that (i) the credible intervals achieve the nominal coverage level only for the "realistic" synthetic truths with modern ice volume within 10% of the observed ice volume, and (ii) the discrepancy term reduces overconfidence for the synthetic truths that are not within the 10% range. 25

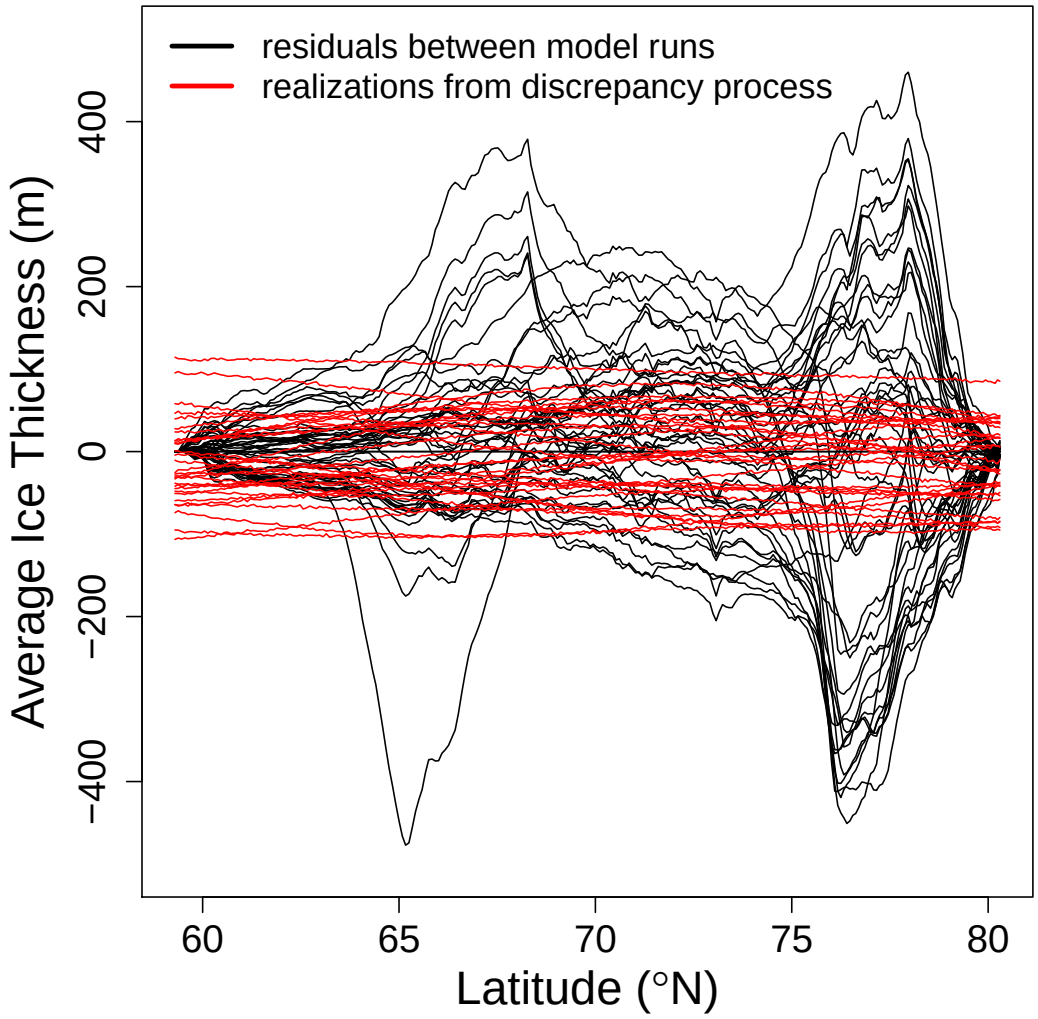

FIG. S1. Comparison between (i) residuals between the synthetic truth used in the main text (model run #67 in Applegate et al. 2012) and other model runs (black solid curves) and (ii) 30 different realizations from the model for the simulated discrepancy (red solid curves). The residuals are computed by subtracting the synthetic truth from each of the other model runs. For better display, we show only residual curves whose ranges are within (-500,500). It is easy to see that the black curves and red curves are generated from different processes, and therefore those two groups of curves can be separated by statistical inference (hence the discrepancy is identifiable). The magnitudes of the simulated discrepancy processes are well within the range covered by the model runs (hence the posterior density of input parameters does not show too large variation).

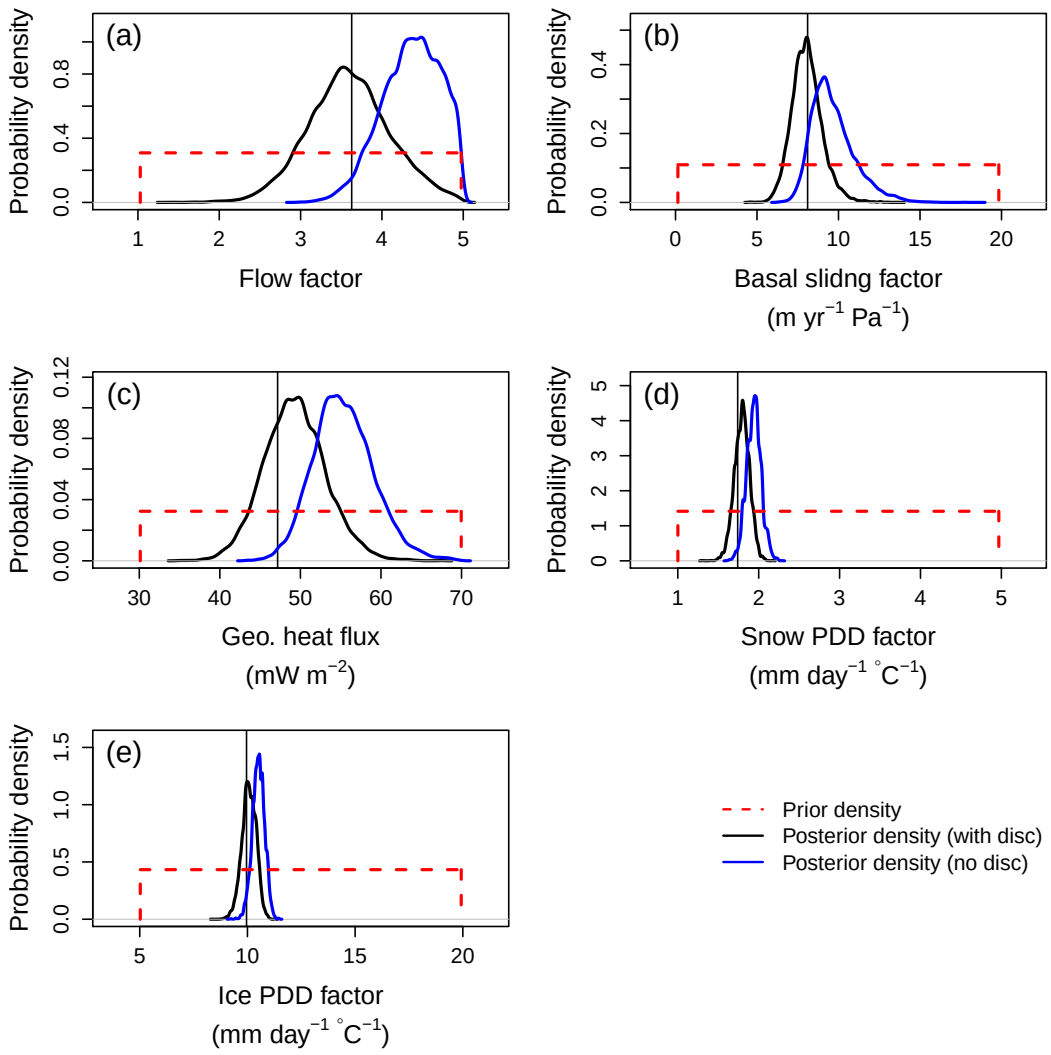

FIG. S2. Comparison between calibration results with and without the discrepancy term $\mathbf{K}_d\boldsymbol{\nu}$ in the calibration model in (S1). In each panel, we tried to learn each of the parameters while fixing the other parameters at their assumed-true values. The prior densities are assumed to be uniform over a broad range (dashed red lines). While the posterior densities computed by including the discrepancy term in the model (solid black curves) pick up the true parameter values without notable biases, the posterior densities without the discrepancy term (solid blue curves) cannot recover the true values.

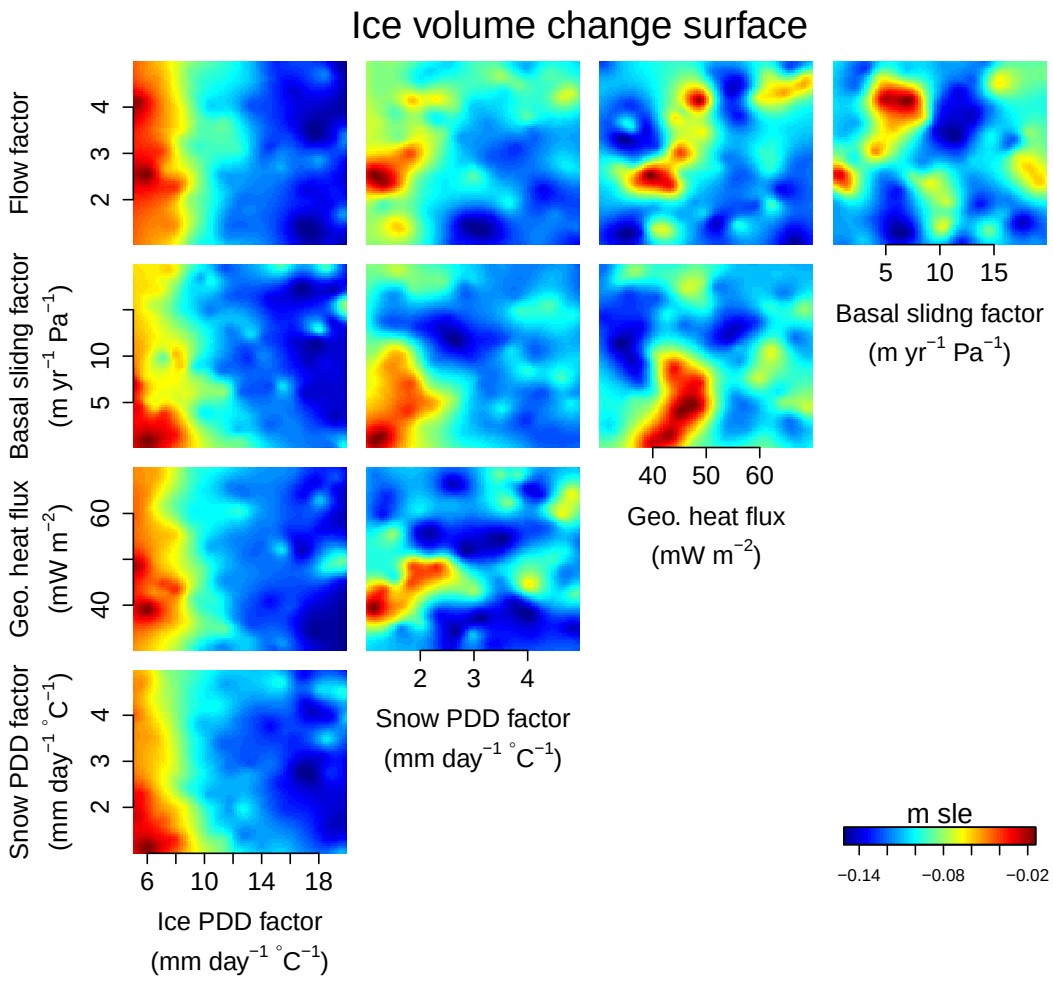

FIG. S3. Surfaces of ice volume change projections between 2005 and 2100 projected onto marginal spaces of all pairs of input parameters. Many local maxima and minima are scattered around the parameter space, indicating that the surfaces behave very irregularly and exhibit highly nonlinear relationship with the input parameters. m sle, meters of sea level equivalent.

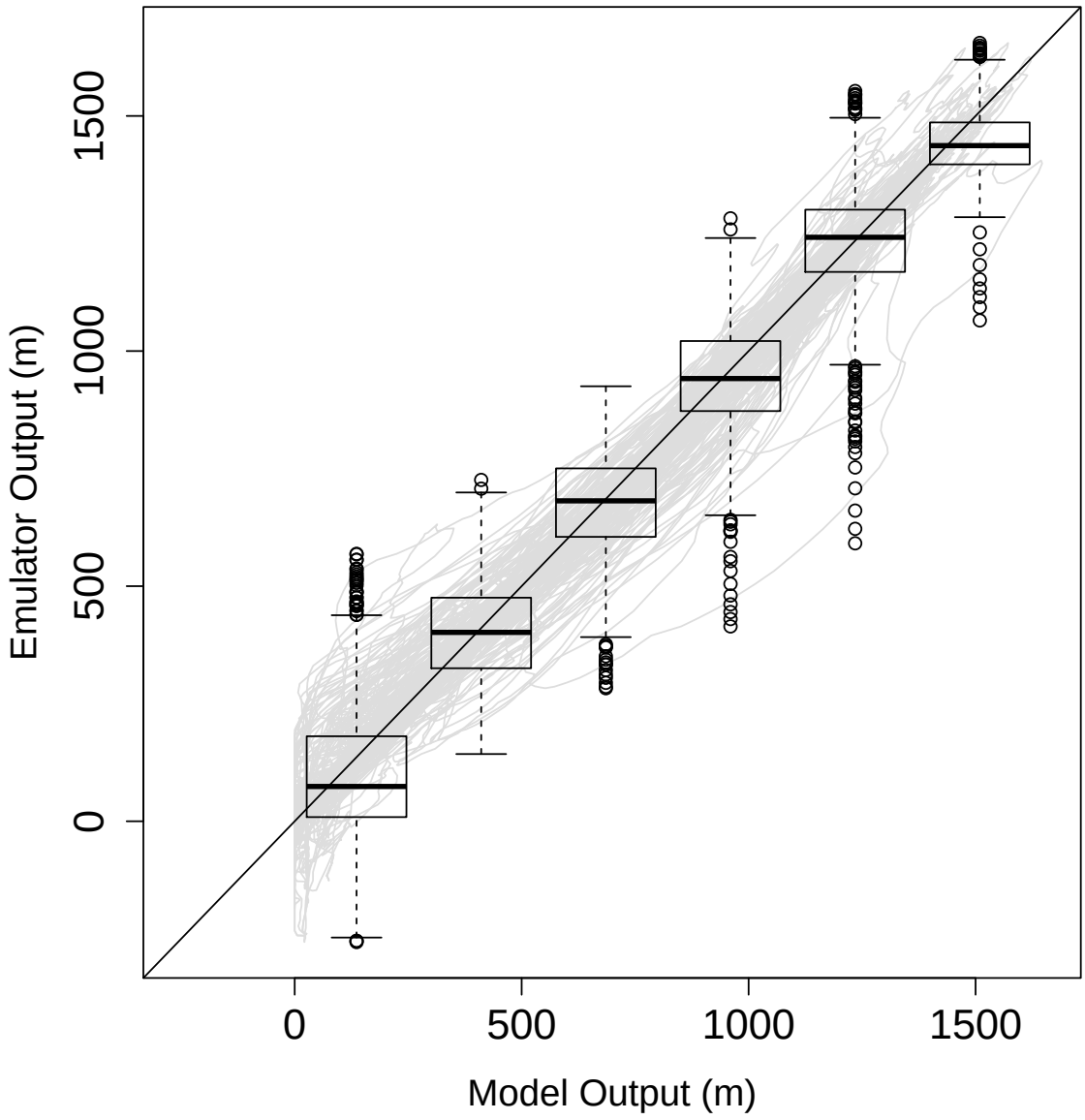

FIG. S4. Leave-one-out cross-validation results for the emulation performance. Each grey curve shows the comparison of zonal mean ice thickness transects from the model output and that from the emulator output for each parameter setting. Each boxplot shows the distribution of emulator output for each of the evenly spaced bins that span the range of true model output. In spite of the fact that our design points for parameter settings are quite sparse (100 runs in 5-dimensional space) most of the curves are concentrated around 1:1 line connecting the lower left and upper right corners of the plot, indicating that our emulator can reconstruct the original model output reasonably well across the input parameter settings.

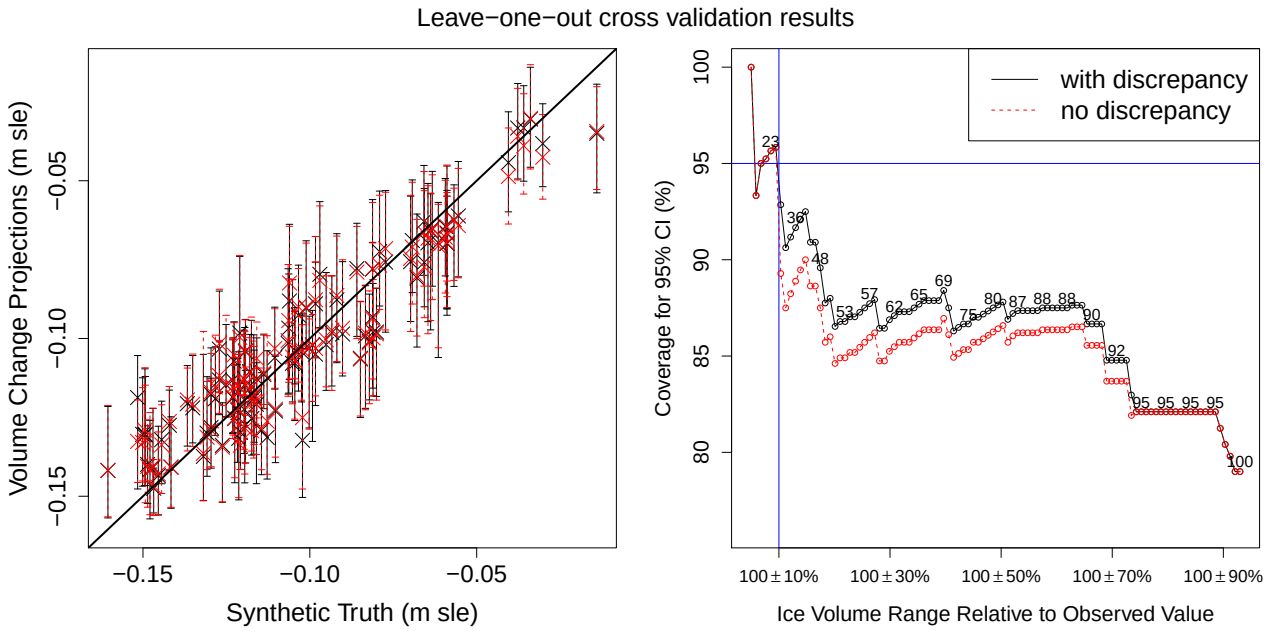

FIG. S5. Leave-one-out cross-validation results for ice volume change projections across all 100 input parameter settings as the synthetic truth. The left penal shows 95% prediction intervals for ice volume change projections across all 100 perfect model experiments conducted for cross-validation. If the interval covers the 1:1 line connecting the lower left and upper right corners of the plot, the 95% prediction interval includes the ice volume projection given by the synthetic truth. The right penal shows the coverage of those prediction intervals as a function of allowed range for the ice volume in 2005 AD relative to the observed ice volume. As going from left to right, the synthetic truths used in computing the coverages include more "unrealistic" ones in terms of modern ice volume. The numbers above the solid black line show how many synthetic truths fall into the given ice volume range. The plot shows that (i) the credible intervals achieve the nominal coverage level only for the "realistic" synthetic truths with modern ice volume within 10% of the observed ice volume, and (ii) the discrepancy term reduces overconfidence for the synthetic truths that are not within the 10% range.

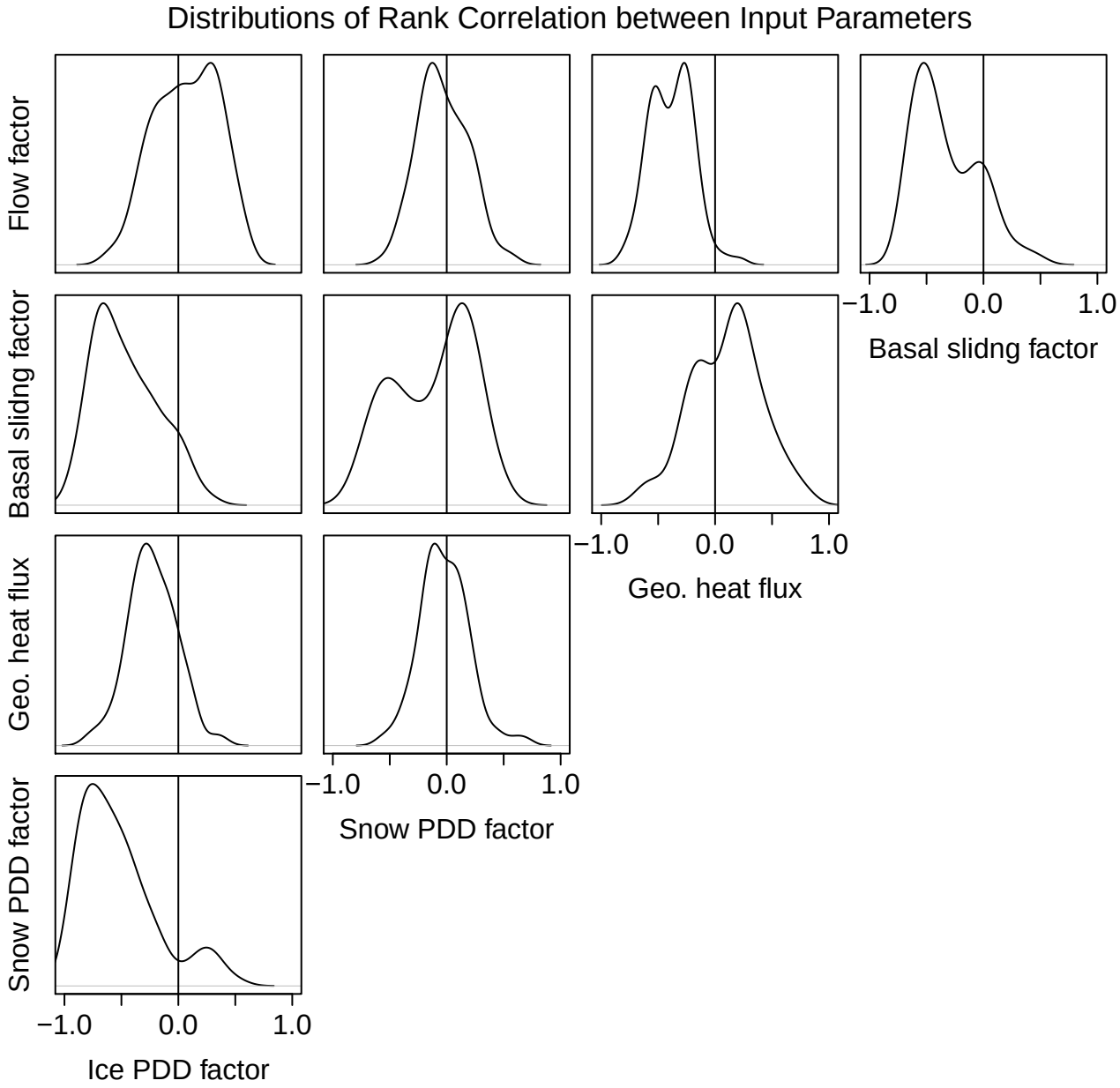

FIG. S6. Summary of interactions between input parameters computed from leave-one-out cross-validation. Each panel shows the distribution of the rank correlation between two input parameters across all synthetic truths in our leave-one-out cross-validation. Five pairs of input parameters, (i) the flow factor and the snow PDD factor, (ii) the flow factor and the geothermal heat flux, (iii) the basal sliding factor and the ice PDD factor, (iv) the geothermal heat flux and the ice PDD, and (v) the ice PDD factor and the snow PDD factor are tend to be more negatively correlated comparing to the other pairs of parameters.