

# Supplemental Information for “PDFlood: Probabilistic Downscaling for Flood Hazard Models”

## 1 Additional details on LISFLOOD-FP

We set the boundary conditions using ground elevation data describing the floodplain topography, channel bathymetry information (river width, depth, and shape), and inflow to the modeling domain. To apply LISFLOOD-FP, we use the subgrid-scale hydrodynamic scheme of Neal et al. (2012) to solve the momentum and continuity equations for both channel and floodplain flow. The scheme operates on a rectangular grid mesh of the same resolution as the input elevation map, i.e. digital elevation model (DEM). The cells’ water depths are updated using mass fluxes between cells, ensuring mass conservation.

## 2 Details on Bayesian emulation-calibration

We calibrate a single parameter, Manning’s roughness coefficient for the channel ( $n_{ch}$ ). This parameter approximates the resistance to flow in the channel (Arcement and Schneider, 1989). Flood models are generally very sensitive to channel roughness (Pappenberger et al., 2008; Savage et al., 2016; Alipour et al., 2022). We set the range of plausible values for channel roughness as  $n_{ch} \in (0.01, 0.1)$  based on prior work (Alipour et al., 2022; Pappenberger et al., 2008).

In calibration we infer computer model parameters by comparing the associated outputs to observational data (e.g. Kennedy and O’Hagan, 2001). In our case the computer model outputs are spatial flood projections. A flood projection is a collection of flood heights at all locations for a given parameter setting. The observational data are simulated flood heights at the 3 m resolution at the locations of local businesses. The computer model calibration framework accounts for sources of uncertainty stemming from uncertain input parameters, observational errors, and model-observation discrepancies. Because we simulate observations at a higher spatial resolution than the flood projections used for calibration, we still need to estimate the model-observation discrepancy.

We define  $Y(\theta, s)$  as the computer model output at location  $s \in \mathcal{S}$  and parameter setting  $\theta \in \Theta$ . In our case  $\mathcal{S} \subset \mathbb{R}^2$  is the spatial domain of interest consisting of latitude and longitude pairs. We denote the  $m$  locations defining the center of each grid cell of the computer model output as  $\mathbf{s}^M = (s_1^M, \dots, s_m^M)$ . Since we only infer channel roughness, the parameter space  $\Theta = (0.01, 0.1) \subset \mathbb{R}^1$ . We have access to computer model runs at  $p$  design points  $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)$ . Each flood projection  $\mathbf{Y}(\theta_i, \mathbf{s}^M) = (Y(\theta_i, s_1^M), \dots, Y(\theta_i, s_m^M))^T \in \mathbb{R}^m$  at parameter setting  $\theta_i$  is a spatial process. The observations  $\mathbf{Z} = (Z(s_1^O), \dots, Z(s_n^O))^T \in \mathbb{R}^n$  occur at locations of local businesses  $\mathbf{s}^O = (s_1^O, \dots, s_n^O)$ . We treat the computer model as a ‘black box.’ In other words, we only change the model inputs.

The Bayesian calibration framework calls for a computer model run at each iteration of the MCMC algorithm. This can require tens of thousands of sequential iterations. This approach can be computationally prohibitive for computer models with even moderately long run times on the order of minutes. This includes the 10 m version of LISFLOOD-FP which would take about 41 days of wall time to produce 10000 model evaluations. We mitigate this problem by constructing a Gaussian process emulator based on a training set of computer model runs.

We calibrate LISFLOOD using a two-stage emulation-calibration approach. We first fit an emulator to the computer model (Equation (1)) and then calibrate the resulting emulator using observations (Equation (2)). Single-stage methods (Higdon et al., 2008)

combine the two stages (emulation and calibration) into a single inferential step. However, two-stage approaches have several advantages over single-stage methods. Fitting the emulator in a separate step using only computer model output prevents contamination by the calibration model parameters (e.g. Liu et al., 2009; Bhat et al., 2010; Bayarri et al., 2007). This allows for easier emulator evaluation and reduces identifiability issues. Two stage emulation calibration is also more computationally efficient.

We provide an overview of the two-stage emulation-calibration framework for a spatial computer model. We also provide details of how we apply this framework in our study. In the emulation step we fit a Gaussian process emulator to training data. The training data is comprised of concatenated spatial computer model outputs  $\mathbf{Y} = (\mathbf{Y}(\theta_1, \mathbf{s}^O), \dots, \mathbf{Y}(\theta_p, \mathbf{s}^O))^T$  evaluated at the  $p$  parameter settings and  $n$  observation locations. We construct the Gaussian process emulator  $\boldsymbol{\eta}(\theta, \mathbf{Y})$  by fitting the model:

$$\mathbf{Y} \sim \mathcal{N}(h(\mathbf{X})\boldsymbol{\beta}, \Sigma_{\boldsymbol{\xi}}(\mathbf{X}) + \sigma^2\mathbf{I}). \quad (1)$$

Here  $\mathbf{X}$  is a  $np \times b$  matrix of covariates including the spatial locations and computer model parameter settings.  $h(\mathbf{X})$  is a function  $h(\cdot)$  of the matrix  $\mathbf{X}$ . We also use  $\mathbf{X}$  to compute the covariance matrix  $\Sigma_{\boldsymbol{\xi}}(\mathbf{X})$ . We estimate the emulator parameters including the vector of covariance parameters  $\boldsymbol{\xi}$ , the regression coefficients  $\boldsymbol{\beta} \in \mathbb{R}^b$ , and the nugget parameter  $\sigma^2 \in \mathbb{R}$ . We estimate  $\boldsymbol{\beta}$ ,  $\boldsymbol{\xi}$ , and  $\sigma^2$  by maximizing the likelihood to fit a Gaussian random field to  $\mathbf{Y}$ . This field gives a probability model for the computer model run at any parameter setting  $\theta \in \Theta$  and any location  $s \in \mathcal{S}$ . The Gaussian process model gives a predictive distribution for  $\mathbf{Y}(\theta^0, s^0)$  at unobserved  $\theta^0$  and  $s^0$  given the runs used for training  $\mathbf{Y}$ . The emulator  $\boldsymbol{\eta}(\mathbf{Y}, \theta)$  is the resulting interpolated process.

In our study we centered  $\mathbf{Y}$  before emulation and set  $h(\mathbf{X}) = 0$ , eliminating the need to estimate  $\boldsymbol{\beta}$ . We fit a Gaussian process emulator at each observation location independently, because we only need computer model output approximated at the observation locations. We fit the Gaussian processes in parallel using an unconstrained quasi-Newton

Method optimizer (Dennis et al., 1981) to maximize the likelihood. Our approach results in the following  $n$  emulators:

$$\mathbf{Y}(\boldsymbol{\theta}, s_i^O) \sim \mathcal{N}(\mathbf{0}, \Sigma_{\boldsymbol{\xi}_i}(\boldsymbol{\theta}) + \sigma_i^2 \mathbf{I}), \text{ for } i = 1, \dots, n.$$

In our case,  $\boldsymbol{\eta}(\mathbf{Y}, \theta)$  is the collection of independent emulators  $\boldsymbol{\eta}(\mathbf{Y}(s_i^O), \theta)$  at all observation locations  $\mathbf{s}^O$ .  $\boldsymbol{\eta}(\mathbf{Y}, \theta)$  gives a predictive distribution for  $\mathbf{Y}(\theta^0, s_i^O)$  at unobserved  $\theta^0$  and any  $s_i^O \in \mathbf{s}^O$  given the runs used for training at the same location  $\mathbf{Y}(\boldsymbol{\theta}, s_i^O)$ .

In the calibration step, we model the observed data  $\mathbf{Z}$  with respect to the emulator  $\boldsymbol{\eta}(\mathbf{Y}, \theta)$  as follows:

$$\begin{aligned} \mathbf{Z} &= \boldsymbol{\eta}(\mathbf{Y}, \theta) + \boldsymbol{\delta} + \boldsymbol{\epsilon}, \\ \boldsymbol{\delta} &\sim \mathcal{N}(\mathbf{0}, \Sigma_{\boldsymbol{\gamma}}(\mathbf{s})), \text{ and} \\ \boldsymbol{\epsilon} &\sim \mathcal{N}(\mathbf{0}, \sigma_{\epsilon}^2 \mathbf{I}). \end{aligned} \tag{2}$$

The discrepancy term  $\boldsymbol{\delta}$  captures the systematic differences between emulator projections and observations (e.g. Chang et al., 2014).  $\Sigma_{\boldsymbol{\gamma}}(\mathbf{s})$  is a spatial covariance matrix.  $\boldsymbol{\epsilon}$  represents the independent and identically distributed observational errors (e.g. Chang et al., 2014). We typically infer  $\theta$ ,  $\boldsymbol{\delta}$  and  $\sigma_{\epsilon}^2$  by sampling from the posterior distribution  $\pi(\theta, \boldsymbol{\delta}, \sigma_{\epsilon}^2 | \mathbf{Z})$  via MCMC.

In our study we use the following simplified approach to estimate  $\boldsymbol{\delta}$ . First, we find the mean absolute error (MAE) between  $\mathbf{Y}(\theta_j, \mathbf{s}^O)$  and  $\mathbf{Z}$  for all  $j = 1, \dots, p$ . For each observation location  $s_i^O$ , we compute the average difference between  $\mathbf{Y}(\theta_j, s_i^O)$  and  $\mathbf{Z}(s_i^O)$  over the three  $\theta_j$ s with the smallest MAEs. We denote the vector of average differences  $\hat{\boldsymbol{\delta}} = (\hat{\delta}(s_1^O), \dots, \hat{\delta}(s_n^O))^T$  and use it in place of  $\boldsymbol{\delta}$  in calibration. We use this simplification to simplify calibration and save computational costs. We then take the posterior mean of  $\theta$ ,  $\theta^*$ , to be a point estimate of the channel roughness value that gives us the low resolution flood projection most similar to the observational data. We find that our simplified

calibration procedure is able to closely replicate the observational data considering the difference in resolution between our observation and our model projections. Thus our approach neglects uncertainty surrounding  $\theta^*$  and  $\delta$ .

### 3 Details on modeling flooding probability with elevation

We produce Figure 1 by repeating these steps for each spatial resolution. First we identify the minimum elevation for which any 10 m resolution cell is not flooded (21.9 m) and the maximum elevation for which any 10 m resolution cell is flooded (25.1 m). We then divide the cells with elevations between these two values into eight groups so that each group has an elevation range of 0.4 m. For each group, we compute the proportion of cells that are flooded and use that to estimate flooding probability. We chose to group cells in this way so that there were at least 100 cells in each group used to estimate the sample proportion. This way, the relationship between flooding probability and elevation would be strictly decreasing to reflect physics-based common knowledge. Figure 1 plots the midpoint value of each elevation group against the proportion of cells that are flooded in that elevation group. Using our group proportions and corresponding midpoints, we fit a Gaussian process to predict flood probability at elevations in between the group midpoints. We select a squared exponential covariance function fix a small nugget parameter to reflect the smoothness of the relationship.

### 4 Additional details on cost distance analysis

The cost-of-passage for each cell characterizes the difficulty of movement through the cell. Factors influencing cost-of-passage may include surface type and what is passing through the cell, which in our case is flood water. Cost-of-passage can depend on the direction of travel through the cell. If this is the case, then a cost-of-passage grid for each direction

of travel may be appropriate. Direction of travel may influence cost-of-passage due to factors like slope and wind direction.

We can use the grid of cost-of-passage cells to calculate the total cost of travel starting from any cell on the grid and finishing at any cell on the grid. For example, assume there are  $N$  cells to travel through to get to a given destination cell with cell 1 being the source cell and cell  $N$  being the destination cell. Let  $c_{i,j}$  be the cost of moving from cell  $i$  to cell  $j$  given that no other cells must be crossed to get from cell  $i$  to  $j$ . Then the accumulated cost  $C_{1,N}$  at the destination cell is  $C_{1,N} = \sum_{n=1}^{N-1} c_{n,n+1}$ . We assume that for two adjacent cells,  $c_{n,n+1} = \frac{c_n + c_{n+1}}{2}$  and for two cells diagonal from each other  $c_{n,n+1} = \frac{\sqrt{2}(c_n + c_{n+1})}{2}$ . Here we assume direction of travel does not matter and  $c_i$  is the cost-of-passage through cell  $i$ .

## 5 Additional results

The calibrated projections from LISFLOOD-FP run at the high spatial resolution (5 m) are similar to the recorded high water marks for Hurricane Ida (Figure S2). The calibrated high resolution projection slightly underestimates four out of five of the high water marks and slightly overestimates the deepest high water mark. Given the scale of the data and the limited availability of high water marks, we deem the 5 m resolution model performance (MAE = 0.2 m) to be satisfactory.

CostGrow performs slightly better in terms of accurately identifying flooded cells, but PDFlood better in terms of accurately identifying non-flooded cells (Table S1). The difference in performance on non-flooded cells is greater than the difference in performance on flooded cells. PDFlood accurately classifies 93% of flooded high resolution grid cells. CostGrow classifies 95% of flooded high resolution grid cells. However, PDFlood accurately classifies 99.9% of non-flooded high resolution grid cells, and CostGrow accurately classifies 92% of non-flooded high resolution grid cells.

For the other flood events considered, PDFlood still correctly identifies a very high

percentage of both flooded and non-flooded cells (Table S2). PDFlood correctly identifies practically all non-flooded cells up to a rounding error. PDFlood also correctly identifies at least 95% of flooded cells for all other flood events considered. PDFlood’s performance on flood events that we do not have high water marks for is at least as good as its performance on Hurricane Ida in terms of these metrics. Compared to CostGrow, PDFlood is still slightly better at identifying non-flooded cells and slightly worse at identifying flooded cells. However, CostGrow never identifies more than 2% more of the flooded cells, and PDFlood consistently identifies 4% more of the non-flooded cells.

## 6 Additional caveats related to calibration

We configure LISFLOOD-FP by treating channel roughness and floodplain roughness as spatially constant. For the small reach of the Schuylkill river we focus on, we find that our assumption of spatially constant channel roughness is reasonable. In addition, previous studies show that flood hazard projections from LISFLOOD-FP are rather insensitive to floodplain roughness (Roth et al., 2023; Hosseini-Shakib et al., 2024).

## References

- Alipour, A., Jafarzadegan, K., and Moradkhani, H. (2022). Global sensitivity analysis in hydrodynamic modeling and flood inundation mapping. *Environmental Modelling & Software*, 152:105398.
- Arcement, G. J. and Schneider, V. R. (1989). Guide for selecting manning’s roughness coefficients for natural channels and flood plains. US Geological Survey Water-Supply Paper 2339.
- Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C.-H., and Tu, J. (2007). A framework for validation of computer models. *Technometrics*, 49(2):138–154.

- Bhat, K. S., Haran, M., and Goes, M. (2010). Computer model calibration with multivariate spatial output: A case study. *Frontiers of Statistical Decision Making and Bayesian Analysis*, 111:168–184.
- Chang, W., Haran, M., Olson, R., and Keller, K. (2014). Fast dimension-reduced climate model calibration and the effect of data aggregation. *The Annals of Applied Statistics*, 8:649–673.
- Dennis, J. E., Gay, D. M., and Walsh, R. E. (1981). An adaptive nonlinear least-squares algorithm. *ACM Trans. Math. Softw.*, 7(3):348–368.
- Higdon, D. A., Gattiker, J., Williams, B., and Rightley, M. (2008). Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103:570–583.
- Hosseini-Shakib, I., Alipour, A., Lee, B. S., Srikrishnan, V., Nicholas, R. E., Keller, K., and Sharma, S. (2024). What drives uncertainty surrounding riverine flood risks? *Journal of Hydrology*, 634:131055.
- Kennedy, M. C. and O’Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464.
- Liu, F., Bayarri, M., and Berger, J. (2009). Modularization in bayesian analysis, with emphasis on analysis of computer models. *Bayesian Analysis*, 4:119–150.
- Neal, J., Schumann, G., and Bates, P. D. (2012). A subgrid channel model for simulating river hydraulics and floodplain inundation over large and data sparse areas. *Water Resources Research*, 48(11).
- Pappenberger, F., Beven, K. J., Ratto, M., and Matgen, P. (2008). Multi-method global sensitivity analysis of flood inundation models. *Advances in Water Resources*, 31(1):1–14.



- Roth, S. M., Lee, B. S., Sharma, S., Hosseini-Shakib, I., Keller, K., and Haran, M. (2023). Flood hazard model calibration using multiresolution model output. *Environmetrics*, 34(2):e2769.
- Savage, J. T. S., Pianosi, F., Bates, P., Freer, J., and Wagener, T. (2016). Quantifying the importance of spatial resolution and other factors through global sensitivity analysis of a flood inundation model. *Water Resources Research*, 52(11):9146–9163.

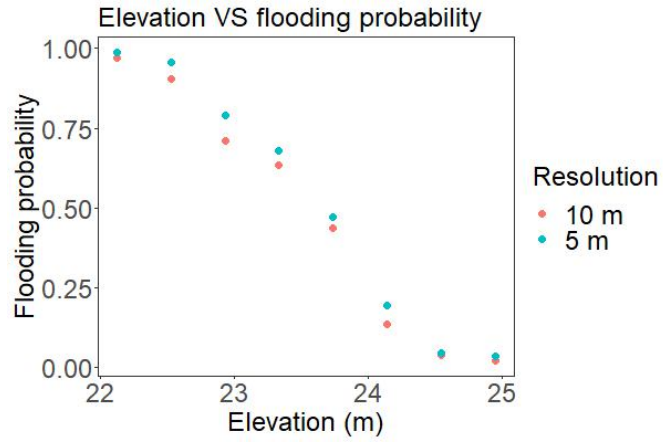


Figure 1: We plot midpoint of each elevation group (x axis) against the proportion of cells that are flooded in said elevation group (y axis). The relationship between elevation and flooding probability is similar whether the cells are 10 m or 5 m resolution.

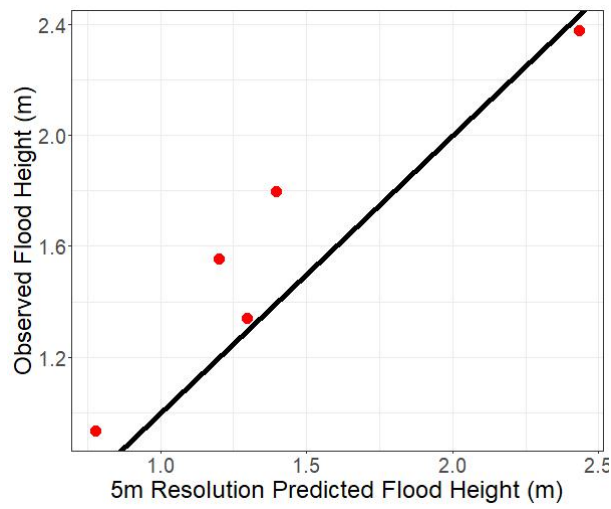


Figure 2: We plot the predicted flood height according to the calibrated 5m resolution model at the observation locations (x axis) against the observations, i.e. high water marks (y axis). For reference, we display a black line representing what perfect model performance would look like.

Table 1: Comparing the performance of PDFlood to CostGrow in terms of how well downscaled projections approximate 5 m resolution projections. Satisfactory values of percent of flooded cells identified and percent of non-flooded cells identified are both greater than 90%. Bold font indicates the better value of the metric between the two approaches. \* indicates that the value has been rounded up to 100%.

	PDFlood	CostGrow
Percent of flooded cells identified	93%	<b>95%</b>
Percent of non-flooded cells identified	<b>100%*</b>	92%

Table 2: Evaluating the performance of PDFlood compared to CostGrow in terms of how well downscaled projections approximate 5 m resolution projections for two historical and one potential future flood event. Satisfactory values of percent of flooded cells identified and percent of non-flooded cells identified are both greater than 90%. Bold font indicates the better value of the metric between the two approaches. \* indicates that the value has been rounded up to 100%.

	PDFlood	CostGrow
2014 flood event		
Flooded cells correctly identified	95%	<b>97%</b>
Non-flooded cells correctly identified	<b>100%*</b>	96%
2020 flood event		
Flooded cells correctly identified	95%	<b>97%</b>
Non-flooded cells correctly identified	<b>100%*</b>	96%
Possible future flood event		
Flooded cells correctly identified	97%	<b>99%</b>
Non-flooded cells correctly identified	<b>100%*</b>	96%