Comparing Emulation Methods for a High-resolution Storm Surge Model

Grant Hutchings, Bruno Sansó, James Gattiker, Devin Francom,
and Donatella Pasqualini

Abstract

The availability of powerful computing resources has led scientists to increasingly utilize simulation as a research tool. The statistical analysis of simulations, referred to as computer experiments, has similarly grown. Gaussian Process (GP) models have proven themselves exceptionally useful in this domain and have become a standard methodology for emulation of simulator response. However, with moderately large training data, GP's require careful implementation to scale appropriately. There are a number of reasonable emulation methods available from ready to use software packages. In this paper we compare four such models: BASS; BART; SEPIA; and RobustGaSP, by applying them to high-resolution hurricane inundation (flooding) data obtained from the Sea, Lake, and Overland Surges from Hurricanes (SLOSH) simulator. Both SEPIA and RobustGaSP are based on Gaussian Process modeling, while BASS implements a model based on adaptive splines, and BART is based on sums of regression trees. We will describe the modeling strategies implemented in these four packages, which run on R and Python, and then compare them in terms of computation time and a variety of predictive metrics. The four models included in this comparison study were chosen for their proven and distinct methodologies, their availability through easily accessible software, and their ability to quantify prediction uncertainty in the context of our application. The data in our case study form a large spatial grid with millions of response values. We find that SEPIA and RobustGaSP provide exceptional predictive power, but cannot scale to accommodate computer experiments as large as the one considered in this paper as effectively as BASS and BART.

1. Introduction.

1.1. Background. The study of complex physical systems is often limited by the acquisition of experimental data which can be expensive or even impossible to gather in many fields. As a result, scientists turn to simulation to supplement experimental data, to gain understanding, and to make predictions (Sacks et al., 1989). Aided by advances in computing, simulators based on mathematical models of physical processes have become a fundamental tool to obtain scientifically motivated representations of a system of interest. Simulators depend on a number of inputs (parameters) that control their behavior. We refer to the set

This manuscript is for review purposes only.

of all possible input values as the parameter space. To obtain a realistic description of the system, and a good understanding of the simulator's capabilities, we must analyze simulation output at collections of points in the parameter space.

Computer simulations do not completely remedy the challenges associated with experimental data. The information that can be obtained from simulations is limited by the feasibility of running the simulation at a given point in the parameter space. Depending on the system of interest, simulations can take hours or even days to run for a given combination of parameter values. This may make it impossible to do an exhaustive direct exploration of the space, a problem that is compounded as the parameter space increases in size or dimension. A variety of examples can be found in Sacks et al. (1989). Additionally, simulators are often deterministic, which typically means that the primary source of uncertainty when running the simulator is parameter uncertainty. Statistical models of these computer simulations, referred to here as emulators, are designed to solve these problems.

A seminal work in the literature of computer experiments (Sacks et al., 1989) showed how a Gaussian Process could be used to build a predictor with uncertainty quantification. GP's became and remain a common approach for emulating computer simulations with a vast literature encompassing a variety of approaches. The main purpose of an emulator is to provide predictions at untried parameter settings with an estimate of the associated uncertainty, and to do it much faster than running the actual computer simulation (Salter and Williamson, 2016a). GP's are therefore very desirable for their predictive power and straightforward uncertainty quantification. They are however not always practical; Gaussian Processes are limited by the computational bottleneck of covariance matrix inversion which limits applicability to large data. Many recent methods such as LaGP (Gramacy and Apley, 2015), TGP (Gramacy and Lee, 2008), GPvecchia (Katzfuss and Guinness, 2021), and RobustGaSP (Gu et al., 2017) aim to tackle this scalability issue. More recently, competitive alternatives to GP's have been proposed such as BASS (Francom and Sansó, 2020) which implements an adaptive spline model, and BART (Sparapani et al., 2021) which uses additive regression trees. These methods similarly provide accurate prediction with simple uncertainty quantification and often a smaller computational footprint.

The analysis presented here will compare four emulation methods on simulated hurricane induced flooding in the Delaware Bay. The simulator considered in this study allows researchers to learn about hurricane flood risk to critical infrastructure on an accelerated timeline, and explore different hurricane scenarios by changing the simulation parameters. The comparison here is motivated by the need for emulation in further analysis based on this model, as well as potential similar future models.

The goals of this study are to quantify the accuracy of predictions and understand the computational requirements of each method for a range of training set sizes. In doing so, we aim to understand how training set size effects predictions and run time. Additionally we

This manuscript is for review purposes only.

will compare the variable importance options given by each method. Investigating variable importance for hurricane flooding models helps researchers understand which qualities of a hurricane or a particular area are most influential in determining inland flooding. Some emulators allow for spatially resolved variable importance and variance-based assessment of importance (e.g., via the Sobol decomposition (Sobol, 2001)), which both benefit analyses involving highly multivariate emulators.

The remainder of the paper is structured as follows: In Section 1.2 we give a brief overview of the four methods included in our study and explain why they were chosen; Section 2 is an overview of the simulations from SLOSH; Section 3 describes of each of the four emulator formulations; Section 4 presents our comparison study, highlighting a variety of predictive metrics and scores; Section 5 gives an overview of the variable importance built into each package; and we conclude with a discussion of our findings and recommendations to the reader in Section 6.

1.2. Emulation Methods. The emulation methods we have chosen implement very different statistical models, all of which have proven themselves a reasonable choice for similarly structured spatial data. We will consider two GP based models, SEPIA (Gattiker et al., 2020b) and RobustGaSP; SEPIA fits a collection of independent GP models to coefficients of an orthogonal basis representation of the simulation response data, while RobustGaSP implements a Many Single approach, fitting an independent GP to each spatial location. We also include the two non-GP based models mentioned above; BASS and BART. These four models cover some diverse modeling strategies, but in no way cover the full spectrum of emulation methodologies. While recognizing the limitations of only considering four models, we would like to highlight the fact that this study customized implementation and computation appropriately for each method for the application, an approach that represents a significant investment in investigator and computational resources compared to a investigation based on relatively limited customization and tailored test problems. Emulator comparisons have been done in the past, often comparing on a host of test functions with relatively small amounts of data, or focusing on parameter calibration rather than strictly emulation (e.g. Salter and Williamson (2016b), Erickson et al. (2018)). The comparison here is motivated by the requirements of this application which poses particular problems that are relevant to spatial environmental modeling. What we present is a comparison which focuses only on a few models in greater detail, in an application driven big-data setting. This, to our knowledge, is not prevalent in the literature.

The first of the four methods that we consider in this paper is the Simulation Enabled Prediction Inference and Analysis (SEPIA) software that implements the Gaussian process model described in Higdon et al. (2008). This model was originally implemented at Los Alamos National Laboratory as the MATLAB code GPMSA (Gattiker et al., 2020a) and in 2020 was refurbished and translated to python as SEPIA. SEPIA makes use of a basis representation,

This manuscript is for review purposes only.

typically empirical orthogonal functions (EOF) (also known as principle components analysis), of the data to fit a Gaussian process to each of the basis coefficients. This is a tried and true methodology for spatial modeling that has seen much success in the literature and in applications.

Our implementation of Bayesian Adaptive Spline Surfaces (BASS) similarly makes use of a basis representation, but takes a wholly different approach to modeling basis coefficients by using adaptive splines. BASS has been recently applied to large spatial data from computer experiments and has shown great results (see, for example, Francom et al., 2019).

The implementation considered in this work of Bayesian Additive Regression Trees (BART) once again makes use of a basis representation where each basis coefficient is fit using an independent BART model. The BART package does not inherently manage multivariate response through basis representation (as in SEPIA and BASS), and so we extend the functionality by explicitly supplying an EOF basis. The BART model fits the EOF weights and the predictions are expanded into the native space. This allows a more direct comparison to other methods. We have explored this implementation in the past (Francom et al., 2020). Treed models have seen success in the literature for their speed and flexibility, and BART has proven to be effective in settings similar to the one considered in this paper, such as a recent analysis of airborne particulate data over California (Zhang et al., 2020). Preliminary comparisons of BASS and BART in Francom et al. (2019) showed that both approaches can be highly accurate and efficient.

The fourth method considered in this work consists of Robust Gaussian Stochastic Process Emulation (RobustGaSP) which handles multivariate response by fitting a GP to each point in space, rather than reducing the modeling dimension through a linear projection as the other methods in this comparison. This is made computationally feasible by both parallel computation, and the assumption of shared range parameters for all GP's. RobustGaSP does not make use of Markov-chain Monte Carlo (MCMC) for model fitting like the other three models. Instead parameters are fit using numerical optimization of marginal posterior distributions. These major model differences make this an interesting inclusion to our comparison study. RobustGaSP has also shown promising results on large scale computer model emulation of large volcanic flow simulations (Gu and Berger, 2016).

Additionally, we include a simple linear model on the coefficients of an orthogonal basis representation as a baseline to gauge the improvements provided by these complex models.

The models considered in this paper all show accurate predictions using quite different methodologies. We will give a more detailed description of each model in Section 3.

2. Simulator and Dataset.

The Sea, Lake, and Overland Surges from Hurricanes (SLOSH) simulator (Jelesnianski et al., 1992) is a computer code developed by the National Weather Service to estimate storm surge heights from hurricanes. Storm surge height is defined as the maximum water height due to a hurricane at any single location. Our data consists of an

ensemble of 4,000 runs from the SLOSH simulator, corresponding to 4,000 simulated storms. Each storm in the ensemble is defined by a unique set of five input parameters:

- sea level rise in the year 2100 (lower: -20; upper: 350; units: cm)
- heading of the eye of the storm when it made landfall (lower: 204.0349; upper: 384.0244; units: degrees, north is 0/360)
- velocity of the eye of the storm when it made landfall (lower: 0; upper: 40; units: knots)
- minimum air pressure of the storm when it made landfall (lower: 930; upper: 980; units: millibars)
- latitude of the eye of the storm when it made landfall (lower: 38.32527; upper: 39.26811; units: degrees)

Input parameters for the ensemble use a space-filling Latin hypercube design over our five dimensional parameter space. Models are trained on subsets of this ensemble and tested on storms outside of the training sets.

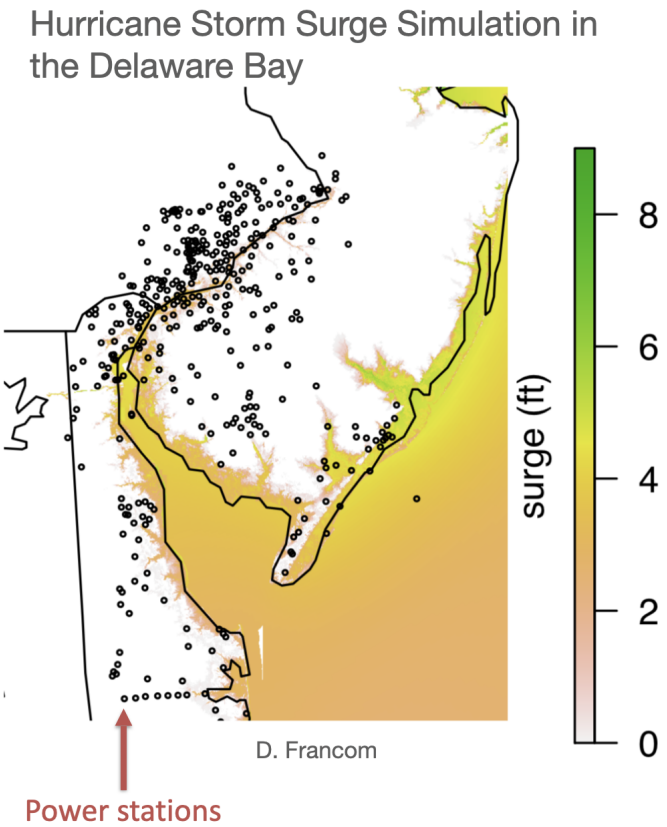


Figure 1: Surge output map from SLOSH

This manuscript is for review purposes only.

Our interest lies in prediction of hurricane-induced flooding in the Delaware Bay. Under our model setup, one output from SLOSH is a 4,520×5,115 grid of storm surge heights for each of the 23,119,800 locations. Figure 1 presents a spatial map of SLOSH output for a given combination of input parameters. This large number of spatial locations presents a formidable computational challenge which is fortunately eased by the fact that the majority of the points on the grid are far enough inland that there is no flooding for any of the 4,000 simulations. By modeling only cells which take non-zero values in at least one of the simulations we reduce the size of the field to 3,500,000 locations.

Accurate prediction of flooding is important for a variety of reasons including displacement of residents, and property/infrastructure damage. One area of specific interest for this project is possible damage to infrastructure, specifically power stations displayed as black dots in Figure 1. Power stations in this area are often fortified to handle four feet of flood water, any more can lead to catastrophic damage. We are therefore interested in the emulators' ability to accurately predict that a surge has reached four feet, as this information is very valuable for determining if an intervention (station shut down) is necessary due to an incoming storm. We will discuss predictions around this threshold of four feet in more detail in Section 4.

3. Model Formulation.

The emulation problem considered in this paper presents the challenge of building emulators that are able to handle 4,000 runs from SLOSH, each with $n_y = 3.5 \times 10^6$ response values. One very common approach to reduce the dimension of a problem like this is to decompose the data into principal components (PCs; Ramsay and Silverman (1997)) using a singular value decomposition (SVD). The output vector $\boldsymbol{y}(\boldsymbol{x}) \in \mathbb{R}^{n_y}$ from one SLOSH run, corresponding to inputs $\boldsymbol{x} \in \mathbb{R}^p$ can be represented on a set of orthogonal basis functions as $\sum_{j=1}^{\infty} w_j(\boldsymbol{x})\boldsymbol{b}_j$ where $\boldsymbol{b}_j \in \mathbb{R}^{n_y}$ captures the spatial variation. By stacking the output obtained from each of the m storms in the training set, we obtain the matrix $\boldsymbol{Y} \in \mathbb{R}^{m \times n_y}$, which we center by subtracting the mean storm. $\boldsymbol{Y}_{ik}$ then corresponds to the standardized output from storm i at location k. We compute $SVD(\boldsymbol{Y}) = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$ where $\boldsymbol{U}, \boldsymbol{V}$ are orthogonal matrices and $\boldsymbol{D}$ is a diagonal matrix of singular values. $\boldsymbol{V}^T$ and $\boldsymbol{U}\boldsymbol{D}$ store the empirical $w_j(\boldsymbol{x})$ and $\boldsymbol{b}_j$ respectively. We choose to truncate the sum at n_{pc} principal components, so that 99% of the variation in the data is captured by the basis representation. The number of principal components used varies by training set. The smallest set with only 50 storms requires just $n_{pc} = 14$ principal components while the largest set with 3,636 requires $n_{pc} = 24$. The power of this decomposition comes from the fact that, rather than fitting an emulator to all n_y response values, we only need to fit n_{pc} scalar response models to the coefficients $w_j(\boldsymbol{x})$, which results in drastic computational savings. We utilize the identical matrix decomposition when fitting BASS, BART, SEPIA, and the linear model. RobustGaSP does not make use of this representation, as discussed. In Subsections 3.1-3.3 we will suppress the subscript j for simplicity and refer to an arbitrary $w_j(\boldsymbol{x})$ as $w(\boldsymbol{x})$.

This manuscript is for review purposes only.

3.1. Simulation Enabled Prediction and Inference (SEPIA). SEPIA is a python code developed by Jim Gattiker, Natalie Klein, Grant Hutchings and Earl Lawrence at Los Alamos National Laboratory (Gattiker et al., 2020b) and implements the model described in Higdon et al. (2008), with extensions. Here we use the emulator component only, without SEPIA's full model calibration functionality. By utilizing the orthogonal basis representation described above, a Gaussian process is fit to each basis function coefficient $w(\boldsymbol{x})$.

$$w(\boldsymbol{x}) \sim GP(0, \boldsymbol{\Sigma});\ \boldsymbol{\Sigma} = \sigma_n^2 \boldsymbol{I} + \sigma_p^2 \boldsymbol{C} \tag{3.1}$$

where $\boldsymbol{C}_{kl} = \exp\{-\frac{1}{2}\sum_{i=1}^{p} \beta_i (\boldsymbol{x}_{ki} - \boldsymbol{x}_{li})^2\}$ is the matrix obtained by applying the negative exponential squared ("Normal kernel") correlation function to each pair of inputs, which is parameterized by length scale $\boldsymbol{\beta}$. $\boldsymbol{\Sigma}$ incorporates process variance σ_p^2 and includes a noise process with variance σ_n^2. This is a Bayesian model with priors on $\boldsymbol{\beta}, \sigma_p^2, \sigma_n^2$. For a full model specification including discussion of priors, refer to Higdon et al. (2008). The resulting posterior distributions are explored via MCMC.

3.2. Bayesian Adaptive Spline Surfaces (BASS). BASS is an R package to fit Bayesian adaptive spline surfaces (Francom and Sansó, 2020). It implements a Bayesian version of multivariate adaptive regression splines (Friedman, 1991). Similar to the approach we took with SEPIA, we make use of a basis representation for the SLOSH output. BASS models each $w(\boldsymbol{x})$ as

$$w(\boldsymbol{x}) = a_0 + \sum_{m=1}^{M} a_m Z_m(\boldsymbol{x}) + \epsilon(\boldsymbol{x}), \quad \epsilon(\boldsymbol{x}) \sim N(0, \sigma^2) \tag{3.2}$$

where $a_0, a_1, ..., a_M$ are constants and $Z_1, ..., Z_M$ are basis functions learned from the data. The basis functions have the form

$$Z_m(\boldsymbol{x}) = \prod_{k=1}^{K_m} g_{km} [s_{km} \max(0, x_{v_{km}} - t_{km})]^{\alpha} \tag{3.3}$$

where $s_{km} \in \{-1, 1\}$ is the sign, $t_{km} \in [0, 1]$ is a knot, v_{km} selects a covariate, K_m is the degree of interaction and $g_{km} = [(s_{km} + 1)/2 - s_{km} t_{km}]^{\alpha}$ is a constant that makes the basis function have a maximum of one. The exponent α defines the degree of the polynomial splines. Note that variables can only be used once in each basis function.

To fit this model we need to estimate $\boldsymbol{\theta} = \{\sigma^2, M, \boldsymbol{a}, \boldsymbol{K}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{v}\}$. This is done via a reversible jump MCMC (RJMCMC) algorithm. For specifics on priors and the RJMCMC algorithm see Francom and Sansó (2020).

3.3. Bayesian Additive Regression Trees (BART). BART is a treed model with strong predictive power for non-linear responses. A recent example is the use of BART for spatial modeling of ambient fine particulate matter pollution (PM_2.5) over California (Zhang et al.,

This manuscript is for review purposes only.

2020). As detailed in Chipman et al. (2010), BART is a sum of trees model where scalar output $w(\boldsymbol{x})$ is approximated as

$$w(\boldsymbol{x}) = \sum_{i=1}^{I} g(\boldsymbol{x}|T_i, M_i) + \epsilon, \quad \epsilon \sim N(0, \sigma^2) \tag{3.4}$$

where each T_i is a regression tree that can incorporate one or more of the p inputs, corresponding to main and interaction effects. A tree T utilizing $\boldsymbol{x}_t \subseteq \boldsymbol{x}$ consists of a set of interior nodes with binary decision rules, and a set of leaf nodes containing parameter estimates. Let $M = \{\mu_1, \ldots, \mu_b\}$ be the parameter estimates associated with the leaf nodes. The interior decision rules are binary splits of the predictor space, either $\boldsymbol{x}_t \in A$ or $\boldsymbol{x}_t \notin A$ where A is a subset of the range of $\boldsymbol{x}_t$. Then any fixed $\boldsymbol{x}_t^*$ is assigned a μ^* by the function $g(\boldsymbol{x}|T, M)$ based on the sequence of decision rules leading to a leaf node.

This additive structure endows BART with a high degree of flexibility when the number of trees is large. This does however come at the price of complexity. BART needs to estimate $\{(T_1, M_1), ..., (T_I, M_I), \sigma\}$ for I trees where T_i and M_i are not single parameters, but an entire tree structure fit with a set of decision rules, and a set of terminal nodes respectively. A backfitting MCMC algorithm is used for posterior sampling, which is designed to efficiently sample the many parameters in the additive tree structure. As a result, BART provides great flexibility with a relatively low computational cost. A key component of the model is a regularization prior which forces the effect from each tree to be small. This prevents individual tree effects from dominating the additive structure. Once posterior draws $(T_1^*, M_1^*), ..., (T_I^*, M_I^*)$ are available, predictions f^* can be obtained as

$$f^*(\cdot) = \sum_{i=1}^{I} g(\cdot|T_i^*, M_i^*) \tag{3.5}$$

(Sparapani et al., 2021).

3.4. Robust Gaussian Stochastic Process Emulation (RobustGaSP). RobustGaSP (Gu et al., 2017) is a GP-based method that avoids the use of the basis function representation that we have used for SEPIA, BASS and BART. Also, unlike the other three models the estimation procedure relies on marginal likelihood optimization rather than MCMC. This has its drawbacks when it comes to uncertainty quantification as confidence bounds must be estimated using distributional assumptions. On the other hand it avoids the iterative sampling involved in MCMC, which incurs relatively large computational cost and memory footprint.

RobustGaSP implements a computationally feasibly alternative to the Many Single (MS) emulation approaches (Conti and O'Hagan, 2010; Lee et al., 2011, 2012). Individual emulators are fit to each coordinate of the output, which, in the context of our case study, consists of n_y independent Gaussian process emulators. Each emulator has its own mean function and

This manuscript is for review purposes only.

variance, but they all share the same correlation parameters $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)$, which are estimated from the joint marginal likelihood of all emulators (Gu and Berger, 2016).

Let $i = 1, \ldots, n_y$ index the locations so that $y_i(\boldsymbol{x})$ denotes the scalar response at location i with inputs $\boldsymbol{x}$. $y_i(\boldsymbol{x})$ is modeled with the Gaussian Process

$$y_i(\boldsymbol{x}) \sim GP(\mu_i(\boldsymbol{x}), \sigma_i^2 c(\boldsymbol{x}, \boldsymbol{x}')),; \; i = 1, ..., k \tag{3.6}$$

where $\mu_i(\boldsymbol{x})$ is the location specific mean function, σ_i^2 the location specific variance, and $c(\boldsymbol{x}, \boldsymbol{x}')$, by default, is the product of p Matèrn 5/2 correlation functions, each with its own range parameter $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)$. Then for m runs of the simulator at inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ we have the multivariate likelihood

$$\big(y_i(\boldsymbol{x}_1), ..., y_i(\boldsymbol{x}_m) | \boldsymbol{\mu}_i, \sigma_i^2, \boldsymbol{\Sigma}\big) \sim \mathbf{MVN}\big((\mu_{i\boldsymbol{x}_1}, ..., \mu_{i\boldsymbol{x}_m}), \sigma_i^2 \boldsymbol{\Sigma}\big) \tag{3.7}$$

where $\boldsymbol{\Sigma}$ is the correlation matrix obtained by applying $c(\boldsymbol{x}, \boldsymbol{x'})$ to each pair of input vectors. The mean function is modelled using a linear regression, $\mu_i(\boldsymbol{x}) = \sum_{l=1}^{L} h_l(\boldsymbol{x})\theta_l$, with basis functions $\boldsymbol{h}_i(\boldsymbol{x}) = (h_{i1}(\boldsymbol{x}), \ldots, h_{iL}(\boldsymbol{x}))$ and unknown regression parameters $\boldsymbol{\theta}_i = (\theta_{i1}, \ldots, \theta_{iL})$. An important aspect of this approach is the definition of the prior for the model parameters. This consists of the product of a standard objective prior is for the mean and variance parameters (Gu and Berger, 2016),

$$\pi^R(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{n_y}, \sigma_1^2, \ldots, \sigma_{n_y}^2) \propto \frac{1}{\prod_{i=1}^{n_y} \sigma_i^2} \tag{3.8}$$

and a jointly robust (JR) prior applied to the correlation parameters $\boldsymbol{\gamma}$. This prior was introduced in Gu (2018) and is called jointly robust because is cannot be written as the product of marginal priors and its robust in marginal posterior mode estimation.

First consider reparameterizing to the inverse range parameters $\beta_j = 1/\gamma_j, j = 1, \ldots, p$. Then the JR prior is defined as

$$\pi^{JR}(\beta_1, ..., \beta_p) = C_0 \bigg(\sum_{l=1}^{p} C_l \beta_l \bigg)^{\alpha} \exp \bigg\{ - b \bigg(\sum_{l=1}^{p} C_l \beta_l \bigg) \bigg\}, \tag{3.9}$$

where $C_0 = \frac{(p-1)! b^{a+p} \prod_{l=1}^{p} C_l}{\Gamma(a+p)}$, $a > -(p+1)$, $b > 0$ and $C_l > 0$ are parameters. We use the default values for these parameters; $a = 0.2$, $b = n^{-1/p}(a+p)$. The default values for C_l are not clearly given in the documentation. As we will discuss in Section 5, this prior facilitates the form of variable importance provided by the package.

The posterior distribution resulting from this model formulation is marginally optimized to obtain parameter estimates.

3.5. Linear Model (LM). For a baseline comparison, we include a simple linear model on the EOF basis coefficients $w(\boldsymbol{x})$ with the form

$$w(\boldsymbol{x}) = \sum_{i=1}^{p} \beta_i x_i + \epsilon, \quad \epsilon \sim N(0, \sigma^2) \tag{3.10}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ are unknown regression coefficients which we determine using the function "lm()" from base R (R Core Team, 2020).

4. Comparison Study. This section presents assessment of the four different emulators on the basis of out-of-sample predictive accuracy and computational feasibility. Predictive accuracy is assessed using scores including root-mean-squared error (RMSE), energy score, and coverage. RMSE assesses the mean prediction, whereas the energy score and coverage assess the uncertainty associated with predictions. We will organize our comparison of predictive accuracy into two Subsections, one for assessing the accuracy of the mean, and the other considering estimates to be used in uncertainty quantification. Our results will show that in these metrics Gaussian Process based emulators (SEPIA and RobustGaSP) produce better mean predictions, however they appear less accurate in their predicted uncertainty.

We would like to be able to train our models with as few storms as needed for accuracy, while minimizing computation time and leaving more examples in the model test set. To examine the impact of training set size for each emulator we consider seven different training sets; 50, 100, 500, 1000, 1750, 2500, and 3636 storms. 3636 was chosen as the largest training set size because it is the largest number that permits a testing set size of 10% of the training set (364 testing storms). The largest training set was sampled randomly from the full 4000 storm ensemble, and subsequent training sets sampled randomly from this set of 3,636. While randomly subsampling a space-filling design is not optimal, the same selection is used for each emulator, affording fair comparison.

Our comparison study involves training each of the four emulators on each of the seven training sets, and computing all prediction metrics on the testing set. All models are tested on the same 364 storms. This allows estimation of the impact of training set size, and comparison of performance both within and between these training set sizes. Computation time is compared across training set sizes revealing the scaling properties of each algorithm. Our results underline an important and well known fact that Gaussian Process, while providing exceptional predictive power, becomes prohibitive with large data-sets. This is evident in that we were only able to fit SEPIA and RobustGaSP with a maximum of 1000 and 500 training storms respectively. We will discuss this further in Section 4.3.

BASS, BART, and SEPIA all make use of MCMC for parameter estimation. For each model we have chosen to collect 10,000 MCMC samples, and discard the first 9000 to eliminate transient state (so-called "burn-in"). Because of the size of the spatial field, we thin the remaining samples down to 50, driven by memory constraints on our computational resources.

This manuscript is for review purposes only.

To fully appreciate the memory challenge, recall that our testing set is 364 storms. To generate predictions using all 1000 posterior samples requires a double precision matrix of size (364 × 3,500,000 × 1000), which requires 10 terabytes of storage. We are limited on our platforms to a more modest 500 gigabyte matrix resulting from the use of 50 samples. This is one of the many challenges involving an application dataset of this size. We appreciate that given the relatively small number of initial samples (10,000) and even smaller number retained samples (50), there may be questions regarding the convergence and mixing of our initial chains, and of how well the 50 samples represent the posterior distributions. These software packages do not provide methods to quantify convergence or mixing, and it is infeasible for us (and in general practice) to tackle this problem for each combination of emulator, training set, and EOFs. The results should be viewed with the understanding that poor convergence/mixing and issues due to small sample set are potentially present in predictive metrics of accuracy and coverage. For a practitioner interested in assessing MCMC convergence, they may want to pursue a thorough analysis of chains which we do not consider here. For those who require this analysis, we would have to recommend reducing the computation by further reduction of the spatial data to make investigation tractable.

The following Subsection will present results for a variety of predictive metrics which can be used to compare the models.

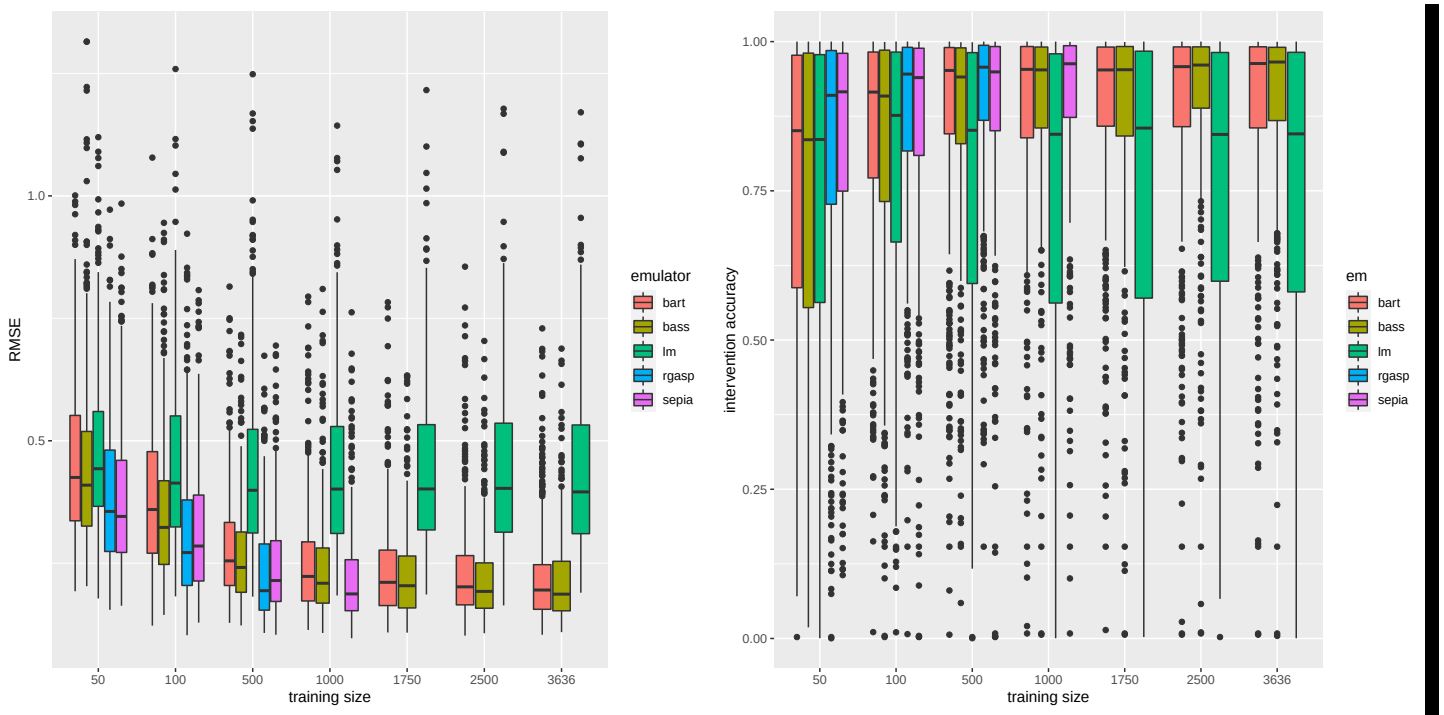
4.1. Predictive Accuracy: Mean. In this section we will access the accuracy of mean predictions from each emulator. For MCMC based models, this is the mean over our 50 posterior predictive samples and for RobustGaSP, the mean is returned to us by the package. Our assessment will consider RMSE, mean absolute error (MAE), and our own loss function designed specifically for flood risk analysis.

Figure 2a shows boxplots of RMSE for each emulation method and for each training set size at which they were run. Samples in each correspond to the 364 test storms dataset. As expected, RMSE is generally decreasing with training set size. The plots show diminishing returns, with a reasonable conclusion that a training set size greater than 1000 runs is unnecessary to achieve best performance in RMSE. Additionally, the figures show that that methods utilizing Gaussian process, SEPIA and RobustGaSP, tend to have the lowest RMSE at each training set size. Furthermore, they produce comparable RMSE to BASS trained on the full 3,636 storms. BASS and BART produce fairly similar results, lagging behind SEPIA and RobustGaSP, with BASS slightly outperforming BART. The results for MAE are very similar to RMSE and are reported in the supplementary material.

In Section 2 we noted that a flooding threshold of four feet is of special interest. This number has real implications in that many power stations are fortified to withstand this level of flood water. [1] Therefore, it is desirable for an emulator to correctly predict flood level above

[1] Different flood impact thresholds can be found in the literature. The four foot threshold is driven by our

This manuscript is for review purposes only.



(a) Root-Mean-Squared Error

(b) Intervention Accuracy

Figure 2: Predictive metrics for mean predictions by training set size.

four feet, a domain-relevant criterion for evaluation. We will evaluate this with standard emulation methods, rather than creating an emulator to satisfy the application-specific loss. To assess these emulators with respect to this feature, we consider the percentage of predictions that correctly indicate that an intervention is needed, which we call the intervention accuracy. To compute this metric for the mean prediction, we consider all cells in which the true SLOSH output is greater than four feet, and determine the percentage of cells in which the prediction is also greater than four feet. Figure 2b shows boxplots of our results where distributions are over the 364 testing storms. We see that SEPIA and RobustGaSP performance is better than BASS, BART, and the linear model at every training set size. Additionally, RobustGaSP with only 500 training storms is able to achieve indistinguishable performance to BASS with the full training set. SEPIA achieves a comparable performance with 1000 training storms. This is further evidence that GP-based methods are able to provide better mean predictions with less training data. This metric is especially interesting when viewed from a risk-management perspective; With SEPIA and RobustGaSP, we are less likely to miss an important interven-

application context of US infrastructure planning, and is indicative of threshold-based evaluation of emulators.

tion. There are a number of near zero values in Figure 2b which we found to be associated with storms for which an especially low number of locations reached the four foot threshold. One reason why this may result in low intervention accuracy is the smoothing associated with prediction. Fewer locations above the four foot threshold likely means those locations reside in smaller clusters which are more likely to be under-predicted due to smoothing effects.

4.2. Predictive Accuracy: Uncertainty. This section presents the results of predictive metrics which take uncertainty into consideration: *coverage probability*, *energy score*, and *interval score*.

4.2.1. Coverage. As all our emulators provide confidence intervals we are interested assessing in their level of coverage. In Figure 3a we present coverage probability distributions for a 95% interval over the 364 testing storms. Using the dashed red line at 0.95, we can see that the linear model, SEPIA, and RobustGaSP all consistently over-cover. BART tends to over-cover with small training sets and under-cover with larger training sets. BASS does the opposite, but seems to be consistently closest to the true 95% coverage. We will now extend our assessment of coverage by comparing the models using a score proposed in Gneiting and Raftery (2007), the interval score.

The interval score for confidence level α is defined as

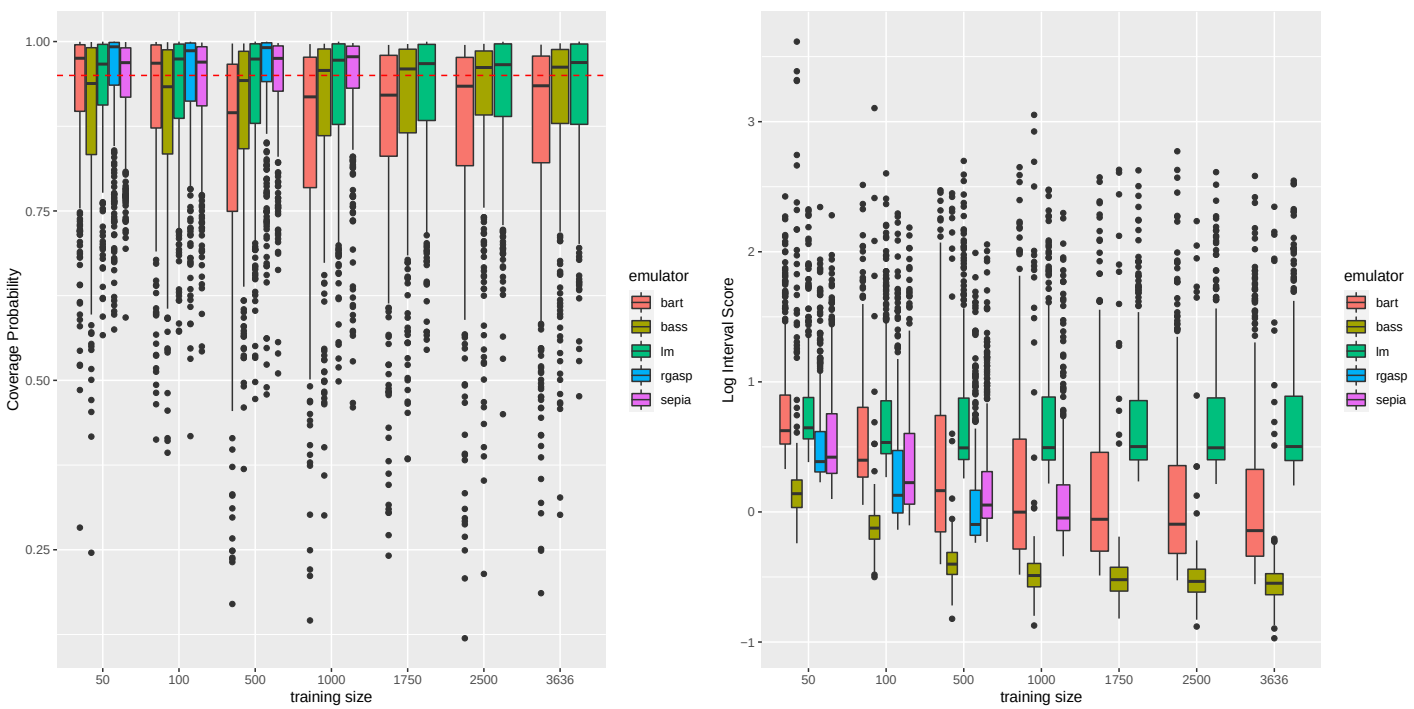
$$S_\alpha^{int}(l,u;x) = (u-l) + \frac{2}{\alpha}(l-x)\mathbb{1}\{x<l\} + \frac{2}{\alpha}(x-u)\mathbb{1}\{x>u\}. \tag{4.1}$$

Where l, u are the lower and upper bounds of the $1-\alpha$ confidence interval, and x is the true data value. This is a negative oriented score that is minimized at the width of the interval. The score then increases proportional to α if the true data value is outside the interval. This score provides more insight than coverage probability by consciously favoring models with the smallest possible intervals that still contain the data. In Figure 3b we present interval score distributions over the 364 testing storms where each storms score is an average over scores for each cell. BASS appears to be quite superior to the other emulators, while the linear model performs poorly in comparison. For small training sets, BART seems to do almost as poorly as the linear model, only catching up to SEPIA at 1000 training storms.

4.2.2. Energy Score. The energy score, a multivariate extension of the Continuous Rank Probability Score (CRPS) is proposed in Gneiting and Raftery (2007). This score takes into account not only the predictive accuracy of each sample from the posterior predictive distribution, but also the level of uncertainty in the distribution. For this reason, the CRPS and energy score have gained interest in recent literature as a model ranking mechanism (Heaton et al. (2018), Möller et al. (2013), Muniain and Ziel (2020)). With m draws from the posterior predictive distribution, $\tilde{Y} = \{\tilde{Y}_1, \tilde{Y}_1, ..., \tilde{Y}_m\}$, we compute the energy score as

$$es(Y,\tilde{Y}) = \frac{1}{m}\sum_{j=1}^{m}||\tilde{Y}-Y|| - \frac{1}{2m^2}\sum_{j=1}^{m}\sum_{k=1}^{m}||\tilde{Y}_j - \tilde{Y}_k||, \tag{4.2}$$

This manuscript is for review purposes only.



(a) Coverage probability, 95% interval.

(b) Interval scores for 95% confidence level.

Figure 3: Coverage Metrics

where Y is the true response.

Results from Subsection 4.1 indicate that Gaussian Process models might be superior in terms of mean predictions. Interestingly, the energy score, which uses predictive samples rather than the mean tends to favor the tree and spline based models over the GP based models. So, while GP's may provide very good mean predictions, results from this section indicate that they may not provide competitive uncertainty quantification to BASS.

4.3. Computational Feasibility. Computation time is an important aspect of any comparison of emulators especially on large data sets where some methods are simply not feasible. All of our models were built on a Los Alamos compute cluster 1.5TB node with 96 cores, 2 Xeon Platinum 8260 CPUs @ 2.40GHz, and 192GB of Dynamic RAM with the exception of RobustGaSP at 500 training storms and SEPIA at 1000 training storms, which were run on a similar but 3.4GHz node due to limits on clock run-time.

As expected, the baseline linear model is extremely fast and scales well but as shown above performs poorly. We can see that BASS remains relatively fast and scales well over the range of training set sizes, requiring a modest 1 minute of computation time to fit the full training

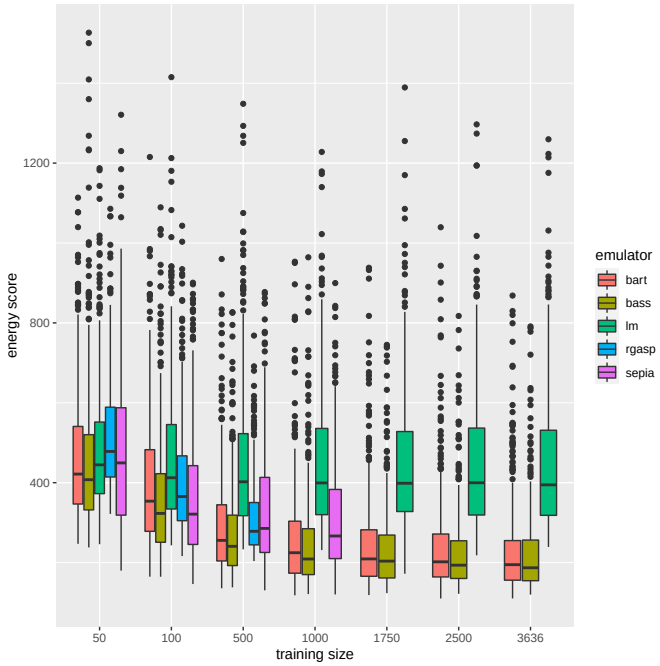


Figure 4: Energy score by training set size

set of 3,636 storms. BART scales similarly, requiring about 5.5 minutes for the full training set.

SEPIA and RobustGaSP scale relatively poorly. Both methods make use of Gaussian process which is inherently $O(n^3)$ scaling, so these methods quickly become infeasible. RobustGaSP is the slowest of the four emulators, perhaps not surprising given the scope of the optimization problem it is addressing, on the native response space.

Parallel MCMC chain approaches may be able reduce execution time for SEPIA by a fixed factor admitting somewhat larger problems, but will not change the inherent scaling. Fortunately for RobustGaSP and SEPIA, in this application we showed that 3,636 training storms is not necessary to achieve near optimal predictive performance. We have seen that RMSE for surge height, flood area, and flood volume all reach best performance with around 1000 training storms.

4.4. Application Specific Metrics. We also considered application-specific flooding and risk analysis related metrics which can be found in the supplementary materials. Specifically, we looked at predictions for the area and volume of catastrophic flooding, where area is defined as the number of land cells with greater than four feet of flood water, and volume is defined as the total water depth summed over all catastrophically flooded locations. We did not find our results to add a significant amount of information regarding the emulator methods

This manuscript is for review purposes only.

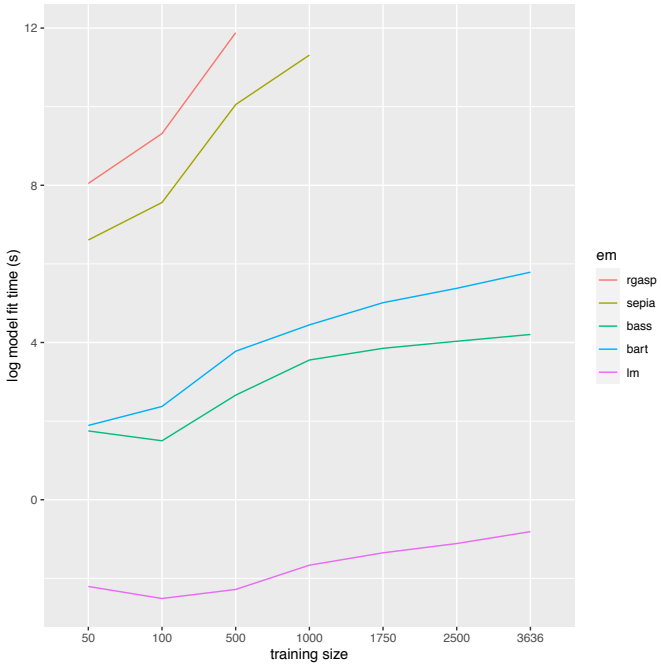


Figure 5: Model fit time

directly to our already rich comparison. To illustrate our point, we provide one example here in Figure 6 which shows the log RMSE for flood volume predictions. This shows little contrast to the information in Figure 2a. Additional results are available to the interested reader in the supplement.

There is also a description in the supplementary materials of an asymmetric loss function that we created to penalize emulators more heavily for under-prediction. This is of interest as a tunable metric that can express risk-aversion of decision-makers, especially surrounding the four foot threshold that results in power station damage. We applied this loss function to mean predictions and again found the results have no significant difference for the purposes of comparative evaluation, when compared to RMSE.

5. Variable Importance. Variable importance for computer models (often referred to as sensitivity analysis) consists of determining which inputs have the greatest (least) effect on the response. Validated emulators are useful for sensitivity analysis and variable importance calculations, as these operations typically require extensive evaluation of the response. Global sensitivity analysis consists of quantifying the percentage of the variability in the response due to each input, or combination of inputs, and is done through functional analysis of variance (ANOVA) (Gu, 2018). More specifically, practitioners often use Sobol indices computed using draws from the emulator posterior predictive distribution (Sobol, 2001). An additional,

This manuscript is for review purposes only.

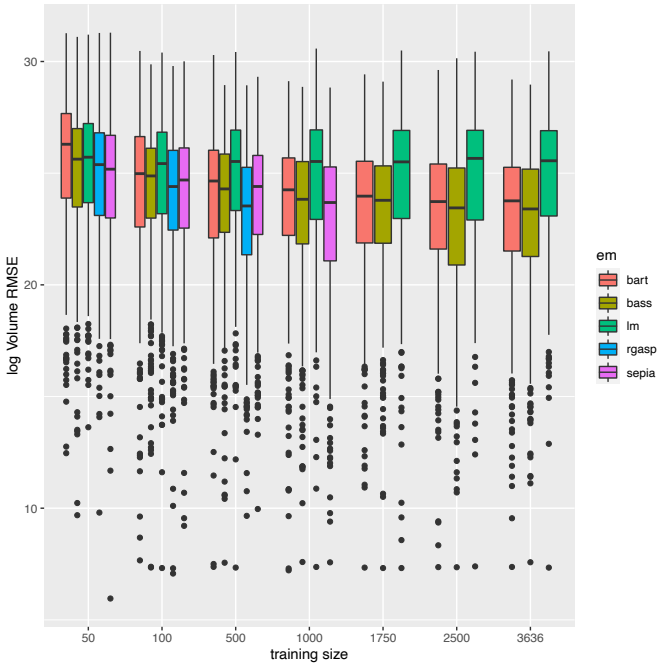


Figure 6: Flood volume log RMSE

very desirable property of Sobol indices is that different uncertainty distributions on the model inputs can be considered, and sensitivities can be compared across these distributional assumptions. This is very applicable to our case study as hurricane impacts are location specific, and there is no broad consensus on their spatial distributions (and the associated distributions in parameters).

SEPIA and BASS have built in functionality to compute Sobol indices, BART and RobustGaSP do not. Methods for computing Sobol indices have been generalized in the R package "sensitivity" (Iooss and Pujol, 2021), so in principle sensitivity indices is available through extensions of the packages. However, the Sobol analysis requires many predictions from the emulator at various input settings, compounded by distribution samples, which would entail considerable computation. For this reason, we will instead compare the variable importance metrics that RobustGaSP and BART provide natively, rather than using those emulators to obtain Monte Carlo based Sobol indices.

The variable importance measures significantly differ in their implications and presentation. This section is not a direct comparison of like quantities as above, but rather a presentation and qualitative comparison of the different information available from the methods to the user.

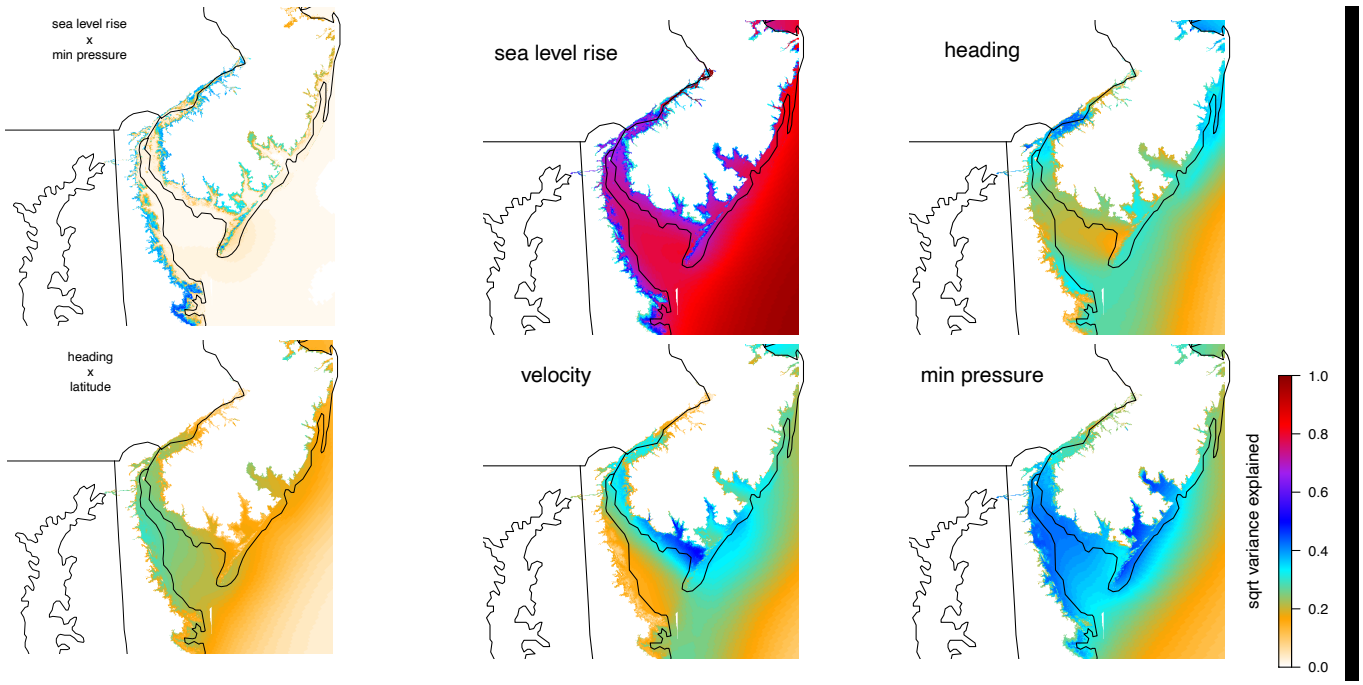


Figure 7: Bass Sobol indices, selected main and interaction effects.

5.1. BASS. BASS includes a closed-form technique for obtaining Sobol indices, facilitated by the underlying model form. The right four plots in Figure 7 show main effect Sobol indices colored by the square root of the explained variance. We can see that sea level rise explains most of the variation in the emulator and that velocity is most important at the northern opening to the bay, with a significant effect all along the northern coast.

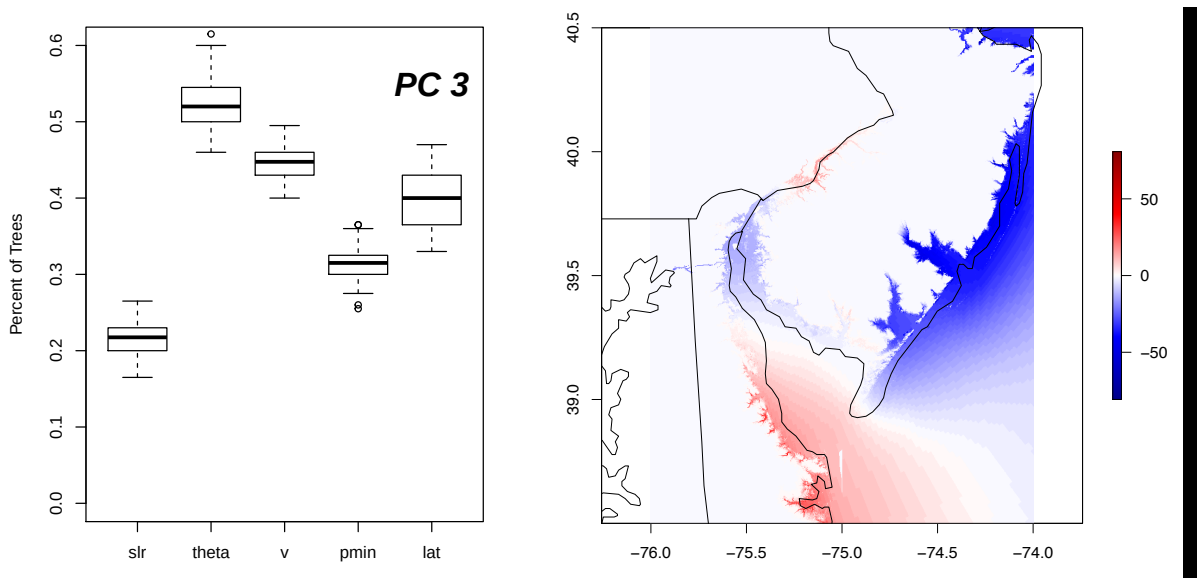
We can also get sensitivity indices for interaction effects, shown in the left two plots of Figure 7. We see that interactions between sea level rise and minimum pressure play an important role in the furthest inland flooding. Our goal here is not to analyze these sensitivities, but rather to demonstrate the information provided by the Sobol decomposition. These results were generated using simple uniform priors over the input parameters.

5.2. SEPIA. SEPIA also has built in functionality for computing Sobol indices which provides sensitivities for the original response, not just the basis coefficients. Unfortunately, we found data of this size infeasible in the current implementation.

5.3. BART. BART offers a unique form of variable importance (and hence, sensitivity analysis) by keeping track of the number of times each input variable is included in the regression trees. For every posterior predictive sample, we calculate the percentage of trees containing each input variable. This gives a distribution of percentages over posterior draws.

The drawback is that information is only available for individual models corresponding to a single basis coefficient and we cannot simply aggregate over components to get sensitivity for the original response.

Figure 8a shows these distributions for the third PC and we notice that heading (theta), velocity (v), and latitude (lat) appear to be the most important inputs. This plot is more informative when combined with a visualization of the principal component as seen in Figure 8b. Now we can see that these inputs explain variability mostly near the northern coast between 39 and 40 degrees latitude. Combining information from these figures gives us an idea of the locations in space where certain inputs are having an important effect. We show PC3 rather than another PC simply because it shows interesting structure and provides a good example of the results that are available from BART.



(a) Percentage of regression trees containing each input

(b) Principal Component 3 spatial map

Figure 8: BART variable importance

5.4. RobustGaSP. RobustGaSP determines if an input is believed to be inert, or contributes little to response variability. Inertness is decided through the estimated range parameters $\hat{\boldsymbol{\gamma}}$. This is really more of a variable selection technique introduced in (Linkletter et al., 2006), but can be considered a form of variable importance or sensitivity analysis. If γ_l is inert, $\hat{\gamma}_l \to \infty$ and has little effect response variability (Gu, 2018). The JR prior we described in Section 3.4 is required for this to work. The key is that this prior, unlike the reference prior,

makes sure the marginal posterior for $\boldsymbol{\gamma} > 0$ even if some $\hat{\gamma}_l \to \infty$. To decide whether a $\hat{\gamma}_l$ is sufficiently large to consider the associated input inert, we consider the normalized inverse

$$\hat{P}_l = \frac{C_l \hat{\beta}_l}{\sum_{i=1}^{p_x} C_i \hat{\beta}_i} \tag{5.1}$$

where $\hat{\beta}_l = 1/\hat{\gamma}_l$ and C_l is a normalization constant to account for the different scales of the inputs (Gu, 2018). We can then set a threshold (default of 0.1) below which an input is determined to be inert. Table 1 shows the results for our RobustGaSP model trained on 500 storms. We see that none of the inputs are found to be inert.

Table 1: Estimated normalized inverse range parameters

sea level rise	heading	velocity	min pressure	latitude
0.58	2.50	1.15	0.34	0.43

Albeit far less informative from a sensitivity analysis point of view than a Sobol decomposition, this is valuable information which comes for free as a byproduct of the model fit.

6. Discussion. Computer model emulation is most beneficial when applied to a simulator that is expensive to run. The SLOSH simulator is expensive enough to require emulation for analysis, but is not overly expensive; SLOSH's relative speed is what allowed us to generate a generous ensemble of 4000 runs making a training set size study possible. The insight gained from this study can provide guidance for studies with more complex storm surge simulators like ADCIRC (Luettich and Westerink, 2015), which incorporates more physics, as well as modeling at greater resolution. As a final note about SLOSH, it was created by the National Weather Service and has thus proven to be the simulator of choice for analysis by government agencies. Given that SLOSH is so widely used, this comparison may be interesting to a wide audience of not only statisticians, but applied scientists exploring uncertainty quantification methods.

Figures 2a and 2b indicate that, for our case study, GP based models produced the most accurate mean predictions. This however comes at a significant computational cost as seen in Figure 5. Additionally we see evidence that our GP based models do not perform as well as BASS in terms of uncertainty quantification in Figures 3a, 3b. Therefore, we recommend SEPIA or RobustGaSP when the size of the ensemble is relatively small with correspondingly tractable computational time, and when uncertainty quantification is not the over-riding emphasis. For most users, efficiency is likely to be very important and for these users we recommend BASS. BASS tends to outperform BART in our predictive metrics such as energy score, coverage probability, and interval score and it is relatively computationally tractable. Additionally BASS supplies intuitive variable importance analysis through Sobol indices, relevant

This manuscript is for review purposes only.

for this application.

In future work we would like to confront some of the questions and limitations that arose during this study. One of which is the outliers seen in all scores. It is clear that some storms are performing very poorly for our predictive metrics, and although we examined some of these, it is not clear whether or how these are systematic in the emulation application. An extension of this work could examine whether these storms have particular features, for example a particular region of the parameter space, and if outliers are consistent across methods. Another limitation that comes with data of this size is the storing of large matrices, which led us to use a relatively small number of posterior predictive samples. We would like to further investigate whether each model has sufficiently converged. For SEPIA, BASS, and BART this means analysis and diagnostics of the MCMC performance to ensure samples represent the model posterior distributions, and for RobustGaSP running the optimization with a number of different parameter initializations to ensure that we have not converged to a local mode. As discussed, these analyses come with heavy computational burden and time that would likely not be available in a typical applied analysis. Finally, in Section 4 we discussed the possibility of reducing the area of particular interest to the application context of power grid impacts, which would admit an effectively larger analysis within computational limitations.

References.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298.

Conti, S. and O'Hagan, A. (2010). Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140(3):640 – 651.

Erickson, C. B., Ankenman, B. E., and Sanchez, S. M. (2018). Comparison of Gaussian process modeling software. *European Journal of Operational Research*, 266(1):179–192.

Francom, D. and Sansó, B. (2020). BASS: An R package for fitting and performing sensitivity analysis of Bayesian adaptive spline surfaces. *Journal of Statistical Software*, 94(8):1–36.

Francom, D., Sansó, B., and Kupresanin, A. (2020). Landmark-based emulation for models with misaligned functional response. Technical report, Technical Report UCSC-SOE-19-09, University of California Santa Cruz.

Francom, D., Sansó, B., Bulaevskaya, V., Lucas, D., and Simpson, M. (2019). Inferring atmospheric release characteristics in a large computer experiment using Bayesian adaptive splines. *Journal of the American Statistical Association*, 114(528):1450–1465.

Friedman, J. (1991). Multivariate adaptive regression splines (with discussion). *Annals of Statistics*, 19:1–67.

Gattiker, J., Higdon, D., and Williams, B. (2020a). lanl/gpmsa. https://github.com/lanl/GPMSA.

Gattiker, J., Klein, N., Hutchings, G., and Lawrence, E. (2020b). lanl/sepia: v1.1. https://doi.org/10.5281/zenodo.4048801.

This manuscript is for review purposes only.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Gramacy, R. B. and Apley, D. W. (2015). Local Gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 24(2):561–578.

Gramacy, R. B. and Lee, H. K. H. (2008). Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130.

Gu, M. (2018). Jointly robust prior for Gaussian stochastic process in emulation, calibration and variable selection.

Gu, M. and Berger, J. O. (2016). Parallel partial Gaussian process emulation for computer models with massive output. *Ann. Appl. Stat.*, 10(3):1317–1347.

Gu, M., Wang, X., and Berger, J. O. (2017). Robust Gaussian stochastic process emulation.

Heaton, M., Datta, A., Finley, A., Furrer, R., Guinness, J., Guhaniyogi, R., Gerber, F., Gramacy, R., Hammerling, D., Katzfuss, M., Lindgren, F., Nychka, D., Sun, F., and Zammit-Mangion, A. (2018). A case study competition among methods for analyzing large spatial data. *Journal of Agricultural, Biological and Environmental Statistics*, 24.

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

Iooss, Sebastien Da Veiga, A. J. and Pujol, G. (2021). *sensitivity: Global Sensitivity Analysis of Model Outputs*. R package version 1.24.0.

Jelesnianski, C., Chen, J., and Shaffer, W. (1992). Slosh: Sea, lake, and overland surges from hurricanes.

Katzfuss, M. and Guinness, J. (2021). A general framework for vecchia approximations of Gaussian processes. *Statistical Science*, 36(1).

Lee, L., Carslaw, K., Pringle, K., and Mann, G. (2012). Mapping the uncertainty in global ccn using emulation. *ATMOSPHERIC CHEMISTRY AND PHYSICS*, 12:9739–9751.

Lee, L. A., Carslaw, K. S., Pringle, K. J., Mann, G. W., and Spracklen, D. V. (2011). Emulation of a complex global aerosol model to quantify sensitivity to uncertain parameters. *Atmospheric Chemistry and Physics*, 11(23):12253–12273.

Linkletter, C., Bingham, D., Hengartner, N., Higdon, D., and Ye, K. Q. (2006). Variable selection for Gaussian process models in computer experiments. *Technometrics*, 48:478 – 490.

Luettich, R. and Westerink, J. (2015). ADCIRC. http://www.adcirc.org. [Accessed: May 2021].

Muniain, P. and Ziel, F. (2020). Probabilistic forecasting in day-ahead electricity markets: Simulating peak and off-peak prices. *International Journal of Forecasting*, 36(4):1193–1210.

Möller, A., Lenkoski, A., and Thorarinsdottir, T. L. (2013). Multivariate probabilistic fore-

casting using ensemble Bayesian model averaging and copulas. *Quarterly Journal of the Royal Meteorological Society*, 139(673):982–991.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ramsay, J. and Silverman, B. W. (1997). *Functional Data Analysis*. Springer.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and Analysis of Computer Experiments. *Statistical Science*, 4(4):409 – 423.

Salter, J. M. and Williamson, D. (2016a). A comparison of statistical emulation methodologies for multi-wave calibration of environmental models. *Environmetrics*, 27(8):507–523.

Salter, J. M. and Williamson, D. (2016b). A comparison of statistical emulation methodologies for multi-wave calibration of environmental models. *Environmetrics*, 27(8):507–523.

Sobol, I. (2001). Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and Computers in Simulation*, 55(1):271–280. The Second IMACS Seminar on Monte Carlo Methods.

Sparapani, R., Spanbauer, C., and McCulloch, R. (2021). Nonparametric machine learning and efficient computation with Bayesian additive regression trees: The BART R package. *Journal of Statistical Software*, 97(1):1–66.

Zhang, T., Geng, G., Liu, Y., and Chang, H. H. (2020). Application of Bayesian additive regression trees for estimating daily concentrations of pm2.5 components. *Atmosphere*, 11(11).

This manuscript is for review purposes only.