Inferring Atmospheric Release Characteristics in a Large Computer Experiment using Bayesian Adaptive Splines

Devin Francom[1,2], Bruno Sansó[1], Vera Bulaevskaya[2], Donald Lucas[2]
[1]University of California Santa Cruz
[2]Lawrence Livermore National Laboratory

June 2, 2017

Abstract

An atmospheric release of hazardous material, whether accidental or intentional, can be catastrophic for those in the path of the plume. Predicting the path of a plume based on characteristics of the release (location, amount and duration) and meteorological conditions is an active research area highly relevant for emergency and long-term response to these releases. As a result, researchers have developed particle dispersion simulators to provide plume path predictions that incorporate release characteristics and meteorological conditions. However, since release characteristics and meteorological conditions are often unknown, the inverse problem is of great interest, that is, based on all the observations of the plume so far, what can be inferred about the release characteristics? This is the question we seek to answer using plume observations from a controlled release at the Diablo Canyon Nuclear Power Plant in Central California. With access to a large number of evaluations of an expensive particle dispersion simulator that includes continuous and categorical inputs and spatio-temporal output, building a fast statistical surrogate model (or emulator) presents many statistical challenges, but is an essential tool for inverse modeling. We achieve accurate emulation using Bayesian adaptive splines to model weights on empirical orthogonal functions. We use this emulator as well as appropriately identifiable simulator discrepancy and observational error models to calibrate the simulator, thus finding a posterior distribution of characteristics of the release. Since the release was controlled, these characteristics are known, making it possible to compare our findings to the truth.

Keywords: model calibration, inverse problem, multivariate emulation, categorical inputs, functional outputs, uncertainty quantification, atmospheric dispersion models

1 Introduction

Less than one year after California's Diablo Canyon Nuclear Power Plant became operational in 1985, a catastrophic nuclear accident occurred in Chernobyl, then part of the Soviet Union, resulting in large amounts of radioactive material being released into the atmosphere and proving fatal for many emergency responders and reactor staff. Radioactive plumes, which drifted over much of the western Soviet Union and Europe, necessitated the evacuation and long-term resettlement of many local people and have had lasting effects on public health (United Nations Scientific Committee on the Effects of Atomic Radiation, 2008).

Pacific Gas and Electric Company, which operates the Diablo Canyon Nuclear Power Plant, performed a series of experimental releases of sulfur hexafluoride, a benign gas, from the California plant soon after the 1986 Chernobyl accident. The purpose of these experiments was to gather concentration data at downwind observation sites after each release to be used for validation of established particle dispersion models in the presence of complex terrain (Thuillier, 1992). The inverse problem, to determine if the release characteristics can be inferred based on observations of the plume, was of less interest. However, the latest large-scale radioactive release that happened after an accident in Fukushima, Japan, has prompted renewed interest in the experimental release data from 1986 as a testbed for particle dispersion forward and inverse modeling (Lucas et al., 2017). Highly complex systems of this nature rarely allow for well controlled experiments, making the 1986 experimental release data a rare asset.

Particle dispersion models have improved in the past three decades, and statistical calibration of computer models has become a well developed field, especially following the work of Kennedy and O'Hagan (2001). Our interest lies in developing non-intrusive uncertainty quantification methods, specifically emulation and inverse modeling methods, suitable for use with the 1986 atmospheric release observations and many evaluations of a state-of-the-art particle dispersion model. This is a difficult task due to the complexity of both the observed and simulated data. The experimental release observations form a spatio-temporal field that includes measurement error, background noise and missing values. Each of our many evaluations of the simulator has both continuous and categorical inputs and outputs a spatio-temporal field, resulting in massive amounts of simulated data. We note that, in general, uncertainty quantification has become an essential step in much of modern scientific exploration, and many fields are consistently increasing their computational capacity, making analysis of large amounts of simulated data a relevant topic.

The primary innovation in this work that can be extended to other large computer experiments is our emulation methodology. An emulator is a fast statistical surrogate for the more complex simulator, which is a necessary tool when the simulator is expensive to evaluate. One of the most commonly used models for emulation is the Gaussian process (GP) (Sacks et al., 1989). However the GP is difficult to use because we employ a large number of evaluations of the simulator, each with spatio-temporal output. Scalability is not a strength of the GP, though more scalable versions are discussed in Kaufman et al. (2011) and Gramacy and Apley (2015). Instead, we use Bayesian multivariate adaptive regression splines (BMARS), recently used for emulation with small numbers of model runs in Chakraborty et al. (2013), Stripling et al. (2013) and Francom et al. (2016). While BMARS can be used on its own for functional emulation (Francom et al., 2016), we opt for a different approach in this application. We use BMARS to model weights on spatio-temporal empirical orthogonal functions (EOFs) that are linearly

combined to yield a spatio-temporal emulator. This has commonalities with the approach of Higdon et al. (2008), which emulates simulators with highly multivariate output by using GPs to model weights when linearly combining principle components. In our application, we also require an emulator that can handle categorical inputs. We develop a way for BMARS to incorporate categorical inputs that is true to its adaptive nature. The result is an emulator that (1) can flexibly and accurately model complicated functional response surfaces; (2) quantifies emulator uncertainty; and (3) is scalable in the number of inputs, the number of model runs and the dimension of the functional response. This is all fairly easy to reproduce via the R package BASS (Francom, 2017).

Inverse modeling, or calibration, for this particular dataset presents its own set of unique challenges. In addition to emulation for simulators that are expensive to evaluate, other essential parts of calibration are building the simulator discrepancy model and the observational error model. We combine these models with a modularized Bayesian approach similar to that of Liu et al. (2009) to ensure identifiability and to simplify computations. We use a BMARS model for the discrepancy. The scalability and flexibility of the combined framework could be valuable for other computer experiments with large amounts of observational and simulation data.

We introduce our data, methods and findings as follows. In Section 2, we introduce the experimental release data and our simulations. In Section 3, we detail the construction of our emulator and evaluate the fit. In Section 4, we describe our calibration framework, detailing simulator discrepancy and observational error models. We also demonstrate the inversion capability of the framework on synthetic data. In Section 5, we show the results of our calibration technique using the Diablo Canyon plume observations and discuss the accuracy. In Section 6, we summarize and detail future work.

2 Data

As is common in computer experiments, we have two sources of data: observations and simulations (evaluations of the simulator). In this section, we describe each of these.

2.1 Observations

The 1986 experimental release we analyze is one of eight releases performed on different days between August 31 and September 17. We focus on a release of 146 kilograms of sulfur hexafluoride (SF_6) on September 4. Starting at 8:00AM local time, the SF_6 gas was released continuously for eight hours from a location at the base of the southmost containment unit at the Diablo Canyon Nuclear Power Plant. Air sampling was performed automatically at 150 sites in the surrounding area. At each site, air was pumped into a tedlar bag over the course of one hour, at which point the bag was sealed and air was pumped into a new bag. Sampling was done from the hours of 7:00AM to 7:00PM, yielding 12 measurements at each site. The quantity of interest, the concentration of SF_6, is reported as an hourly average for each site. Roughly 24% of the samples are missing for unknown reasons.

The first sample at each site (the 7-8am average) was taken prior to the beginning of the experimental release, thus measuring the background level of SF_6. Investigation of the background level shows moderate amounts near the plant switchyard, where SF_6 is used for electrical insulation. A map of the background level is shown in Figure 1, along with the time series of

the period after the release for a few sites. The time series are noisy, show orders of magnitude in variation (even in the background level) and missing values. The large background level at sites near the switchyard complicate the task of determining the controlled release characteristics based on our plume observations, since this fugitive release plays a confounding role. To minimize this confounding, for the 12 measurements at each site we subtract the site background level (the first measurement). Any negative values are truncated to zero. This is an effort to remove the background, but it makes the assumptions that the background level (1) is constant over time, (2) is small enough that it has negligible effect on down-wind measurements (i.e. it mostly disperses before reaching other sites) and (3) is not subject to large measurement error. The latter two assumptions are likely to be valid, but the first is difficult to justify. Still, in the absence of unconfounded temporal data to characterize the switchyard release, we proceed under these assumptions. For sites that are missing a background measurement (shown with slightly smaller dots in Figure 1), we do not subtract a background value from the measurements. While we could try to impute the background level from neighboring sites, we assume that the background level does not exhibit strong spatial correlation. Further, there are no missing background readings that we believe would be significantly large (i.e., there are no missing values very near the switchyard). Because of the many orders of magnitude difference in observations, small background errors are unlikely to have much effect on our ability to identify release characteristics. Even the largest background measurements are orders of magnitude smaller than the largest post-release measurements later in the time series.

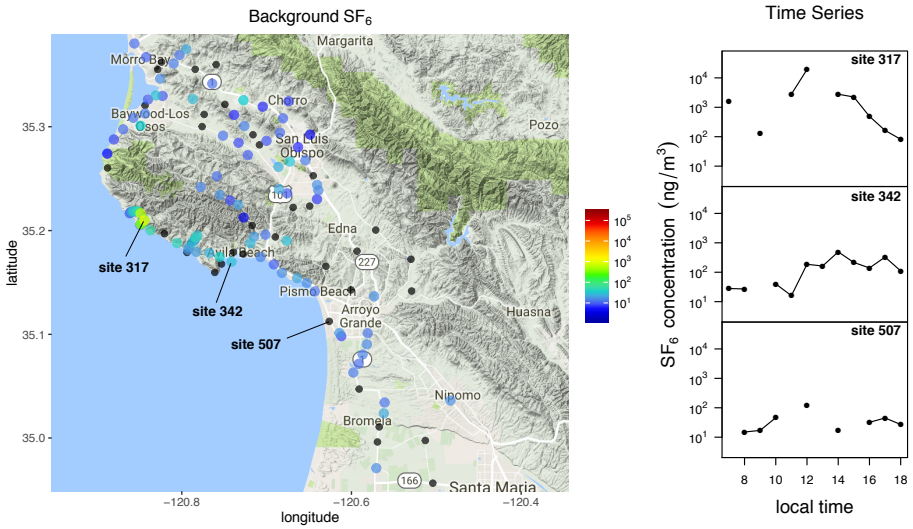


Figure 1: On the left, background SF_6 levels are shown (corresponding to 7:00 local time) for 137 of the 150 sites. Slightly smaller back dots are sites missing the 7:00 observation. On the right, observed time series for a few of the sites are shown. Lines connect adjacent observations in time, with missing values excluded.

2.2 Simulations

The simulator used in this work is a Lagrangian particle dispersion model called FLEXPART ("FLEXible PARTicle dispersion model") (Stohl et al., 2005). An in-depth discussion of the

Table 1: FLEXPART continuous parameter ranges

parameter	lower bound	upper bound	true value
latitude	35.1977	35.2250	35.2111
longitude	-120.8708	-120.8384	-120.8543
altitude (meters)	1	10	2
start time	7:00	9:00	8:00
duration (hours)	6	10	8
amount (kg)	10	1000	146.016

simulator and simulations can be found in Lucas et al. (2017), while we only introduce the simulations briefly here. The FLEXPART model requires six inputs detailing characteristics of the release (latitude, longitude, altitude, start time, duration, and amount) as well as a wind field. We use the Weather Research and Forecasting Model (WRF) to generate wind fields while varying only a few of WRF's many inputs. The inputs we vary are pre-release initialization time (9 or 15 hours), planetary boundary layer physics model used (YSU, MYJ TKE, or MYNN TKE), land surface model used (thermal diffusion, Noah, or RUC), FDDA nudging amount (none, low, or high), and type of reanalysis data used (NARR, ECMWF, or CFSR). We use a series of five nested domains for evaluating WRF, so that the innermost domain resolves winds to 300 meters. More about these parameters and nesting can be found in Skamarock et al. (2008). A combination of the five categorical variables yields a wind field, which, along with a setting of the six continuous variables detailing the release characteristics, yields a simulated plume.

We obtain an ensemble of 18000 evaluations of FLEXPART with the 11-dimensional inputs sampled from a Latin Hypercube using the ranges in Table 1 for the six continuous parameters. We note that 18000 model evaluations is uncommonly large in the computer experiment literature, though there are some recent applications with many evaluations (Gramacy and Apley, 2015; Kaufman et al., 2011). Our computational budget allowed for this large number of evaluations, though each is still expensive to obtain. As discussed above, this large number of simulations makes emulation difficult, since traditional emulation techniques do not scale well.

The output from one evaluation of the simulator is spatio-temporal on a grid of 400×400 spatial locations and 34 time points, as shown in Figure 2. The 34 time points are 30-minute averages starting at 6:00 local time, thus extending to 23:00. We use only a subset of the 160,000 spatial locations when building the emulator, though the methods we present are scalable to moderately large spatial grids. Specifically, we use the 137 spatial locations that correspond to the sites shown in Figure 1, which are sites where we collect measurements. We exclude 13 of the 150 measurement sites from the analysis because they fall outside the region considered in the simulations, and are far enough away from the release that they are likely to only measure background levels.

Both simulation and observation data are transformed to be on the $\log_{10}$ scale after adding a constant of 20 to all values. The log scale makes modeling the many orders of magnitude difference in concentrations easier. Adding 20, determined in consultation with field experts, minimizes the amount of attention we give to small (in this case unimportant) concentration values.

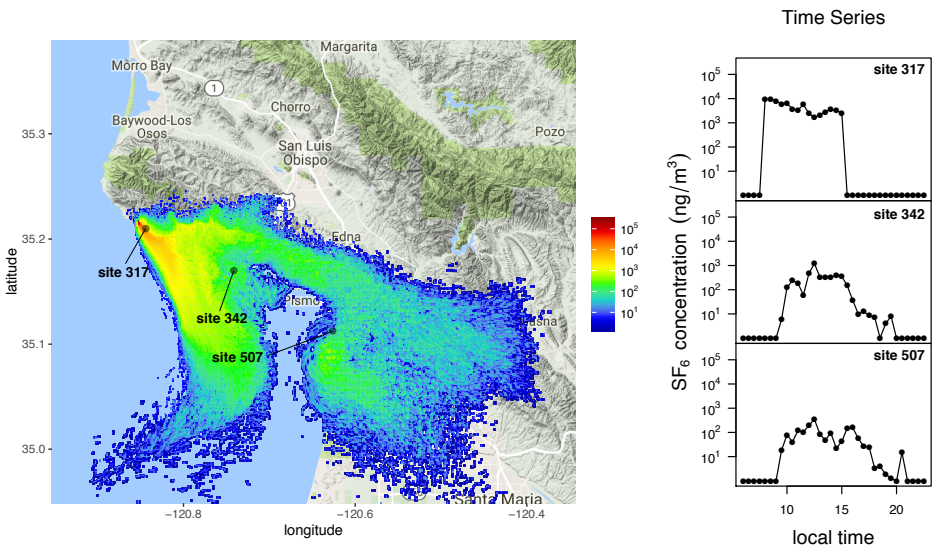


Figure 2: We show the simulated SF_6 plume for one of the 18000 simulations on the left (corresponding to 9:30 local time). On the right, we show simulation time series for a few of the sites.

3 Emulator

A usable emulator in this case needs to be able to produce a reasonably close approximation to the computer model for any possible combination of the 11 inputs. That is, given the characteristics of the release and the wind field characteristics, it should produce the spatio-temporal plume. Obtaining an estimate of emulation error is also a necessary task in order to allow us to propagate the error into calibration uncertainty. If this error is disregarded, we leave open the possibility of being too confident in our estimates of release and wind characteristics that could have generated the calibration data. This case also requires an emulator that can use the large number of model runs available.

Our model runs provide us with 18000 plumes in space and time. We use these plumes to obtain empirical orthogonal functions (EOFs) in space and time jointly, and we model the weights in this EOF decomposition using adaptive splines.

3.1 Empirical Orthogonal Functions

Let $y^c(s,t,\mathbf{x})$ denote computer model output at spatial location s, time t and 11-dimensional input setting $\mathbf{x}$. The model output is on a grid of n_s spatial locations and n_t times. We define the vector of model output for input setting $\mathbf{x}_j$ as

$$\mathbf{y}^c(\mathbf{x}_j) = (y^c(s_1,t_1,\mathbf{x}_j), y^c(s_2,t_1,\mathbf{x}_j), \ldots, y^c(s_{n_s},t_1,\mathbf{x}_j), y^c(s_1,t_2,\mathbf{x}_j), \ldots, y^c(s_{n_s},t_{n_t},\mathbf{x}_j))' \quad (1)$$

and define the matrix of model run output $\mathbf{Y}^c = [\mathbf{y}^c(\mathbf{x}_1), \ldots, \mathbf{y}^c(\mathbf{x}_{n_x})]$ for the n_x model runs, which has dimensions $n_s n_t \times n_x$. We obtain discretized EOFs using the singular value decomposition, yielding $\mathbf{Y}^c = \mathbf{UDV}'$. The matrix $\mathbf{U}$ is the $n_s n_t \times n_s n_t$ matrix that has EOFs as columns and $\mathbf{DV}'$ is the $n_s n_t \times n_x$ matrix of weights. To reduce the dimension in the problem, we use only

the first k EOFs, where $k < n_s n_t$, resulting in the truncated decomposition $\hat{\mathbf{Y}}^c = \hat{\mathbf{U}}\hat{\mathbf{D}}\hat{\mathbf{V}}'$ where $\hat{\mathbf{U}}$ has dimension $n_s n_t \times k$ and $\hat{\mathbf{D}}\hat{\mathbf{V}}'$ has dimension $k \times n_x$. For modeling purposes, we write the non-truncated EOF decomposition as $y^c(s,t,\mathbf{x}_j) = \sum_{i=1}^{n_s n_t} K_i(s,t) w_{ij}$ where $K_i(s,t)$ is the i^{th} EOF at spatial location s and time t and w_{ij} is the corresponding weight for $\mathbf{x}_j$, specifically $(\mathbf{D}\mathbf{V}')_{ij}$.

We specify our emulator as the truncated decomposition

$$y^c(s,t,\mathbf{x}) = \sum_{i=1}^{k} K_i(s,t) w_i(\mathbf{x}) + u(s,t) \tag{2}$$

where $u(s,t)$ is the truncation error. We replace w_{ij} with $w_i(\mathbf{x})$ in order to allow us to predict computer model output for $\mathbf{x}$ not in our training sample $\{\mathbf{x}_1, \ldots, \mathbf{x}_{n_x}\}$. We model the weight functions using adaptive splines with

$$w_i(\mathbf{x}) = \eta_i(\mathbf{x}) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma_i^2) \tag{3}$$

using the values of $\{w_{i1}, \ldots, w_{in_x}\}$ to train $w_i(\cdot)$. We discuss the form of the function $\eta_i(\cdot)$ in the next section. We assume that the weights on the EOFs are independent *a priori*.

Regarding the truncation, we need k, the number of EOFs used, to be sufficiently large to yield a suitable approximation, but small enough to be computationally feasible. Assuming a parametric distribution for the truncation error, $u(s,t)$, for which it is independent and identically distributed for all s and t is likely to improperly characterize the emulation uncertainty because of the large variation in plume characteristics in space and time. Instead we assume truncation error for a particular (s,t) combination comes from the distribution of $n_x = 18000$ truncation errors we have seen already,

$$u(s,t) \sim Unif \left\{ y^c(s,t,\mathbf{x}_j) - \sum_{i=1}^{k} K_i(s,t) w_{ij} \right\}_{j=1}^{n_x}. \tag{4}$$

Since the EOFs are fixed, there is little interest in the truncation error after we have chosen k, other than to make sure it is propagated during calibration. Hence, we see no value in trying to estimate a parametric distribution. The large value of n_x makes our discrete distribution for the truncation error accurate without making any limiting assumptions about distribution tails and symmetry. Further, since this distribution has no unknown parameters and because the weights $w_i(\mathbf{x})$ and $w_j(\mathbf{x})$ have no *a priori* correlation, the weights will also have no *a posteriori* correlation. This means that the adaptive spline models for $w_i(\mathbf{x})$ and $w_j(\mathbf{x})$ can be fit completely in parallel, which is a significant computational benefit.

3.2 Adaptive Splines

BMARS models are a Bayesian version of the more traditional MARS models introduced in Friedman (1991b). These models can be a powerful tool for emulation for a number of reasons (Denison et al., 1998; Chakraborty et al., 2013; Francom et al., 2016): (1) they are adaptive because knots and variables are only included in basis functions if they are necessary; (2) they perform implicit variable selection, as basis functions only include variables that are useful; (3) they scale well, especially when compared to Gaussian process models; (4) they can flexibly pick

up localized signal when the data suggest such signal exists; (5) they yield analytical Sobol' sensitivity indices; and (5) they can be used to emulate simulators with functional response.

The model for the adaptive spline mean function $\eta_i(\mathbf{x})$, used in Equation 3, is a linear combination of tensor product basis functions. We drop the index as we describe this model, since it is fit independently for $i = 1, \ldots, k$. Thus, $\eta(\mathbf{x}) = a_0 + \sum_{m=1}^{M} a_m B_m(\mathbf{x})$ where the basis function $B_m(\mathbf{x})$ is of the form

$$B_m(\mathbf{x}) = \begin{cases} \prod_{j=1}^{J_m} g_{jm}[s_{jm}(x_{v_{jm}} - t_{jm})]_+ & \text{if } J_m > 0 \\ 1 & \text{if } J_m = 0. \end{cases} \tag{5}$$

When $J_m > 0$, the basis function is a tensor product of polynomial splines where t_{jm} is a knot, v_{jm} indexes a variable, and $s_{jm} \in \{-1, 1\}$. The function $[\cdot]_+$ is defined as $\max(0, \cdot)$. The constant g_{jm} scales the basis function to have maximum of one, which helps with computational stability. Without loss of generality, if $x_{v_{jm}} \in [0, 1]$, then $g_{jm} = [(s_{jm} + 1)/2 - s_{jm}t_{jm}]^{-1}$. A basis function has J_m elements in the tensor product where each is required to involve a different variable. Hence, J_m is the degree of interaction for basis function m. The piecewise structure of the basis function will become important in the next section, where we will discuss the case when $J_m = 0$.

The unknowns associated with $\eta(\mathbf{x})$ are the number of basis functions, M, the basis function weights $\mathbf{a}$, and the basis function parameters. The value of σ^2 from Equation 3 is also unknown (again, we drop the index for notational simplicity). We assign priors to all of these parameters following the specifications of Francom et al. (2016). To keep the number of basis functions small, we use $M|\lambda \sim Poisson(\lambda)$ with $\lambda \sim Gamma(1, 1 \times 10^5)$. We also regularize the basis coefficients by using a g-prior (Zellner, 1986; Liang et al., 2008) such that $\mathbf{a}|\sigma^2, \tau, \mathbf{B} \sim N(\mathbf{0}, \sigma^2(\mathbf{B}'\mathbf{B})^{-1}/\tau)$ with $p(\sigma^2) \propto 1/\sigma^2$ and $\tau \sim Gamma(1, 1/n_x)$. Here, $\mathbf{B}$ is the $n_x \times (M + 1)$ matrix of basis functions (including an intercept) conditional on the number of basis functions and the basis function parameters $\{J_m, (t_{jm}, v_{jm}, s_{jm})_{j=1}^{J_m}\}_{m=1}^{M}$. These basis function parameters are given a discrete uniform prior with the constraint that each resulting basis function have at least 20 non-zero points to help prevent edge effects (Francom et al., 2016).

3.3 Categorical Predictors

As we have discussed, our emulator needs to be able to handle categorical inputs. This requires an extension of the traditional adaptive splines models that follows an approach similar to Friedman Friedman (1991a), but modified to fit into our Bayesian framework. Specifically, if the inputs to the simulator are $(\mathbf{x}, \mathbf{z})$ where $\mathbf{z}$ is a vector of categorical variables, we define the portion of the basis function due to the categorical variables as

$$B_m(\mathbf{z}) = \begin{cases} \prod_{l=1}^{L_m} 1(z_{u_{lm}} \in D_{lm}) & \text{if } L_m > 0 \\ 1 & \text{if } L_m = 0. \end{cases} \tag{6}$$

where u_{lm} indexes a categorical variable, D_{lm} is a proper subset of the categories of variable u_{lm}, and L_m is the degree of interaction of categorical variables. This approach to handling categorical variables is somewhat similar to Storlie et al. (2015) and Ma et al. (2015). We combine this portion of the basis function with Equation 5 so that the m^{th} basis function is

$B_m(\mathbf{x}, \mathbf{z}) = B_m(\mathbf{x})B_m(\mathbf{z})$. The purpose of allowing for $L_m = 0$ or $J_m = 0$ is to allow for basis functions that involve only the continuous predictors, or only the categorical predictors, respectively. In our case study, we permit up to third-order interactions of each variable type ($J_m = 3$ and $L_m = 3$) maximally resulting in six-way interactions if the data dictates that such interactions are useful. Allowing D_{lm} to be a subset of categories rather than a single category is useful for cases when two or more categories exhibit similar behavior, as it allows us to use fewer basis functions to describe the behavior. We again use a discrete uniform prior for the categorical basis function parameters $\{L_m, (u_{lm}, D_{lm})_{j=1}^{L_m}\}_{m=1}^{M}$, and we alter the constraint discussed above to apply to the new set of basis functions that include both categorical and continuous variables. That is, each resulting basis function must have at least 20 non-zero points.

Because the resulting posterior is transdimensional (since M is unknown), we use reversible jump Markov chain Monte Carlo (Green, 1995) to obtain samples of the posterior model space. We use the steps detailed in Denison et al. (1998) with the improvements of Nott et al. (2005) to add a basis function, delete a basis function, or make a small change to a basis function.

3.4 Performance

We demonstrate the performance of the emulator by comparing emulator and simulator output at input settings not used to fit the surrogate or to build the EOFs. We use the R package BASS (Francom, 2017) to fit the BMARS models. Figure 3 shows the performance at 15 of the 137 spatial locations for two different holdout simulator runs. In addition to the emulator mean, posterior predictive uncertainty is shown. The prediction uncertainty varies with space and time because (1) the homoscedastic BMARS error from Equation 3 is multiplied by the corresponding spatio-temporal EOF in Equation 2, thus producing spatio-temporal noise and (2) the truncation error varies in space and time. The models corresponding to the most important EOFs use around 200 BMARS basis functions to fit the data, while the least important EOFs, which correspond to higher frequencies, use around 50 basis functions.

These holdout predictions demonstrate reasonable emulator performance, especially considering the complexity of this simulator. Predictions of a number of other randomly chosen holdout simulations also show good performance. By using a large number of EOFs in our emulator, we are able to capture subtle variation in the shape of the time series for different input settings. For instance, the difference in the shapes of the two time series for the Whalers Island location (top right in Figure 3 (a) and (b)) could be attributed to "random" variation if a less accurate emulator was used. Instead, we can attribute the difference to parameter variation. Because the simulator is deterministic, there is technically no "random" variation, though there may be numerical variation that is difficult to attribute to any parameter.

Our efforts to build an emulator for a single location (using the 34 time points) resulted in poor emulation compared to those that utilize the spatio-temporal plume information. We also achieved better emulation performance using spatio-temporal EOFs than we did using EOFs that were separable in space and time. While separable EOFs have the benefit of easier interpretation, we have little interest in interpretation, and separability ends up being a poor assumption for this plume model.

The only case we foresee where we may be interested in interpretation of the EOFs is when doing sensitivity analysis. We can easily obtain the Sobol' decomposition for each EOF adaptive spline model, extending the work of Francom et al. (2016) to include the categorical framework

we have introduced. BMARS emulators are useful tools for global sensitivity analysis because the integrals necessary in the Sobol' decomposition can be obtained in closed form, yielding no Monte Carlo error. However, if the EOF lacks interpretability, so too does the sensitivity analysis. Because the first EOF is generally interpretable as the average and accounts for most of the variance, sensitivity analysis of that EOF model may be interesting (shown in Figure 4). That sensitivity analysis shows that variable three, which is the release altitude, plays no role in the weight for the first EOF. Hence, we would expect to learn very little about that parameter during calibration, since it is also unimportant in all of the other EOF models. We note that a more interpretable functional sensitivity analysis can be performed analytically using our emulator (i.e. the Sobol' decomposition integrals are analytical), but it bears a large computational burden and will not be further explored in this paper.

4 Calibration Model

Our strategy for calibration differs from that of Kennedy and O'Hagan (2001) in a few important ways. Rather than the GP emulator, we use BMARS. We also opt to fit the emulator before calibrating, independent of the observational data. While we do this primarily for computational reasons, it marks a general difference in strategy. A similar approach was used in Gramacy et al. (2015), with the justification that with a large number of simulations the observational data is not going to significantly influence the emulator. Others justify this strategy on philosophical grounds, because the emulator is meant only to replicate the computer model and thus should not be influenced by observational data (Liu et al., 2009).

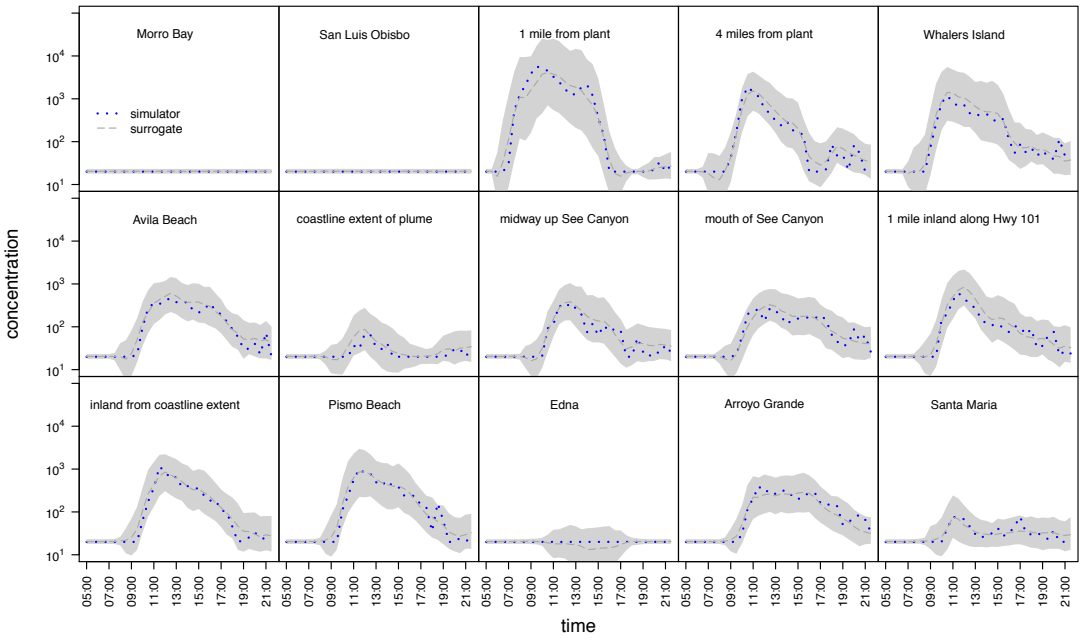
Let $y^F(s,t)$ denote the log concentration data gathered from sensors at location s and time t. Let $\zeta(s,t)$ denote the true concentration at the same location and time. Then we set up the calibration model as follows:

$$y^F(s,t) = \zeta(s,t) + \nu, \quad \nu \sim N(0, \sigma_F^2) \tag{7}$$

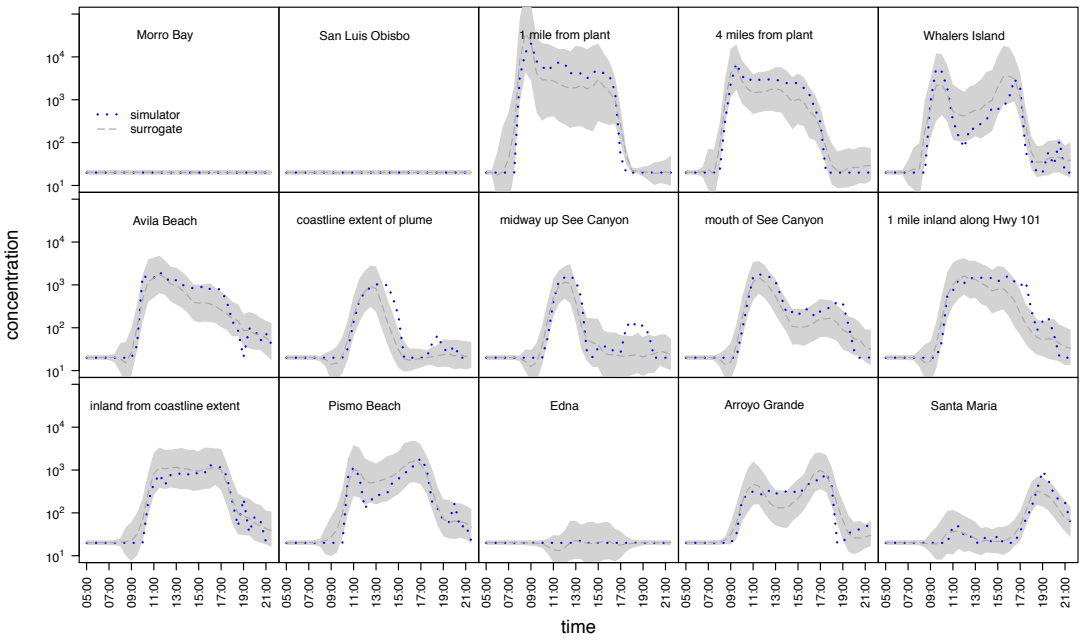
$$\zeta(s,t) = y^c(s,t,\boldsymbol{\theta}) + \delta(s,t) \tag{8}$$

where ν is the observation error, σ_F^2 is the observation error variance and $y^c(s,t,\boldsymbol{\theta})$ denotes the estimated computer model output at location s, time t and input setting $\boldsymbol{\theta}$. The $\delta(s,t)$ term denotes the systematic model discrepancy. Hence, $\boldsymbol{\theta}$ is the unknown setting of the simulator that best matches reality when jointly considered with simulator discrepancy and observational error. While the observations are hourly averages, the emulator gives 30-minute averages. Thus, we average 30-minute averages in the emulator output to match the observation time scale. We also exclude emulator and discrepancy values at (s,t) where the corresponding $y^F(s,t)$ is missing.

A potential problem with the introduced modeling framework is that the observations can be produced in a number of different ways. For instance, we might get good prediction if we get as close as we can to the observations by only altering $\boldsymbol{\theta}$, and then consider $\delta(\cdot)$ to be the leftover spatio-temporal structure in combination with ν, the observational error. However, we could achieve equally good prediction by fixing $\boldsymbol{\theta}$ at a particular value and only altering $\delta(\cdot)$. These two examples represent the extremes in overfitting, but we may attain equally misleading combinations of these. Hence, we need restrictions in order to make all the terms identifiable. The most satisfying restrictions we can introduce are in the form of an informative prior for



(a) Holdout Sample 1



(b) Holdout Sample 2

Figure 3: Demonstration of emulator fit at 15 locations. The top plots (a) show the time series simulator output (dotted lines) for a particular set of inputs not used to train the emulator. The bottom plots show a different input setting, to demonstrate variation in the model output shape. Emulator posterior predictive means (dashed lines) and pointwise 95% probability intervals (shaded region).

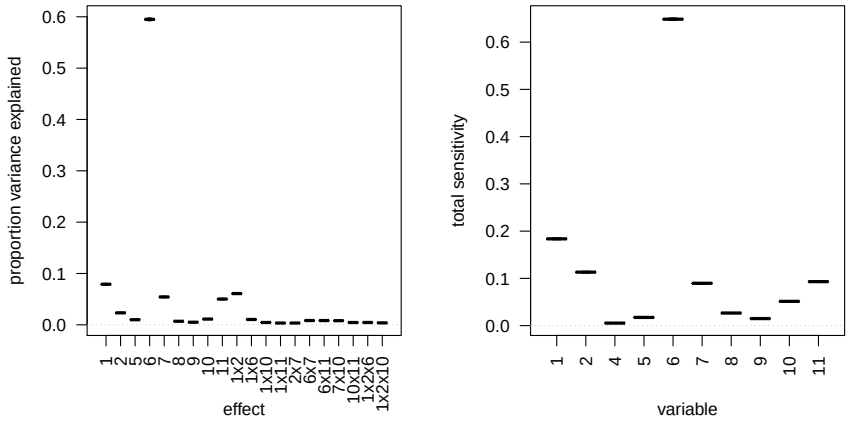


Figure 4: Sensitivity analysis for the adaptive spline model corresponding to the first EOF. Variable numbers 1-6 are the continuous inputs while variables 7-11 are the categorical inputs.

the shape of the discrepancy, rather than an informative prior for the parameters $\boldsymbol{\theta}$ (Liu et al., 2009). While there may be some applications where the shape of the discrepancy are somewhat understood, we do not have an informative prior for our discrepancy model.

Following Liu et al. (2009), we make the quantities of interest identifiable by modularizing the analysis further. Particularly, we fix the shape of the model discrepancy before trying to infer $\boldsymbol{\theta}$. We get an estimate of the model discrepancy by estimating the computer model output at the prior mean parameter settings (Bayarri et al., 2007) and subtracting that from the observations. We then fit a BMARS model to that discrepancy as a function of s and t, effectively smoothing it out. While we fix this discrepancy shape, we allow for its influence to vary by multiplying the discrepancy by a scale factor, γ. Our prior for γ is uniform between zero and two. If γ is near zero, the influence of the discrepancy is minimal. If it is near two, then the discrepancy plays a larger role. We limit the upper bound to two because anything larger would give the discrepancy similar magnitude to the simulator output, which we hope is not the case. While we introduce a scale parameter to the discrepancy, we do not introduce an intercept parameter because doing so would likely confound our ability to learn one of the simulator parameters (release amount), which explains much of the magnitude variation. Without *a priori* knowledge of systematic magnitude discrepancy, we refrain from including an intercept (or a multiplicative discrepancy term for $y^c(\cdot)$, included in Kennedy and O'Hagan (2001)). Thus, we alter Equation 8 to be

$$\zeta(s,t) = y^c(s,t,\boldsymbol{\theta}) + \gamma\delta(s,t).$$

While the functional forms of $y^c(\cdot)$ and $\delta(\cdot)$ are fixed in advance, we incorporate their uncertainties in calibration by sampling their posterior predictive distributions, rather than using their mean functions.

Under this calibration framework, our likelihood is $\mathbf{y}^F|\boldsymbol{\theta},\sigma^2 \sim N(\mathbf{y}^c(\boldsymbol{\theta}) + \gamma\boldsymbol{\delta}, \sigma_F^2\mathbf{I})$ and our

posterior is

$$\pi(\boldsymbol{\theta}, \sigma_F^2, \gamma | \mathbf{y}^F) \propto N(\mathbf{y}^F | \mathbf{y}^c(\boldsymbol{\theta}) + \gamma \boldsymbol{\delta}, \sigma_F^2 \mathbf{I}) IG(\sigma_F^2 | a, b) 1(\boldsymbol{\theta} \in D) 1(\gamma \in [0, 2]) \tag{9}$$

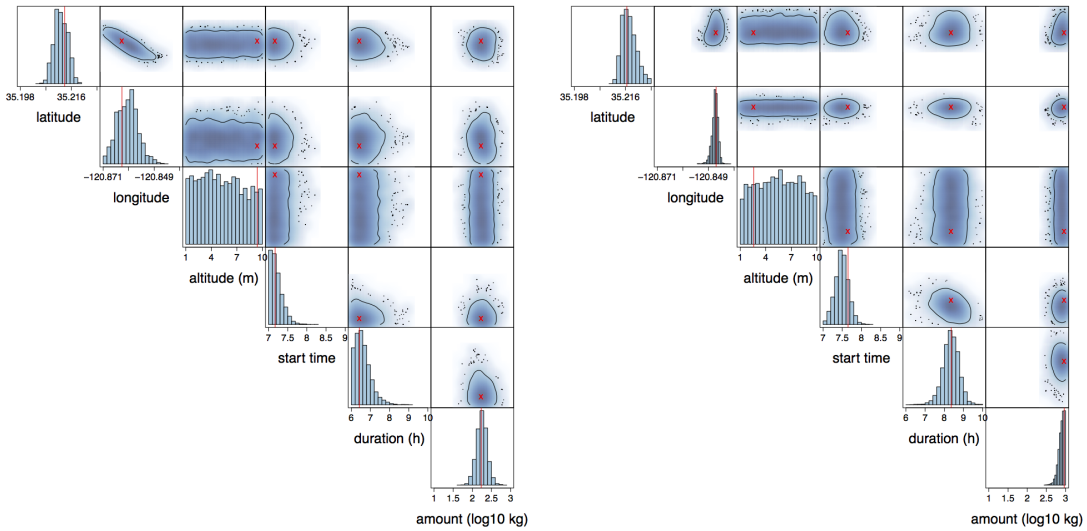
where D is the hypercube based on the prior ranges identified in Table 1 that also allows for all the discrete parameter combinations. We obtain samples from the posterior by using Markov chain Monte Carlo (MCMC) methods (Gelman et al., 2013). We sample σ_F^2 and γ from their Inverse Gamma and Truncated Normal full conditionals, respectively. We use the Metropolis-Hastings algorithm to sample the six continuous parameters in $\boldsymbol{\theta}$ from their joint full conditional. We sample the five categorical parameters jointly from their discrete full conditional. Specifically, there are 162 combinations of the categorical parameters. We take a sample from the emulator posterior predictive distribution for each of the 162 combinations and evaluate the posterior up to a constant. This is reweighted to produce a posterior (full conditional) probability for each setting of the categorical parameters. We then sample one of the 162 combinations according to that probability distribution.

The computational bottleneck to this algorithm is sampling the emulator posterior predictive distribution, as it requires building the BMARS basis functions for all EOF models. A single sample takes 0.2 seconds when the tasks for the 100 models are split between four cores. To obtain 162 samples, one for each categorical combination, a naive approach would require more than 30 seconds for each MCMC iteration. We overcome the bottleneck in sampling the posterior predictive 162 times by building all of the possible categorical basis functions in advance. For each EOF model and each emulator posterior sample, we obtain the basis functions used in each of the 162 predictive combinations. The resulting basis functions, which are combinations of ones and zeros, are multiplied (pointwise) by the portions of the basis functions from the continuous predictors. This allows us to obtain the 162 posterior predictive samples in less than 0.1 seconds on four cores, in contrast to the naive approach that takes more than 30 seconds. Though the categorical basis functions may seem like they would have a large memory footprint, they are combinations of ones and zeros and thus can be stored efficiently in memory.

4.1 Synthetic Calibration

To test our methods, we can calibrate to data where we know the truth. Particularly, we use two of the holdout model runs to perform two synthetic calibrations. That is, we treat each of the two holdout model runs as the true data. Hence, we exclude the simulator discrepancy portion of the model. The goal is to see if the parameters used to generate the model runs can be reasonably identified. One- and two-way marginal posterior distributions for the six continuous parameters are shown in Figure 5. These show that we are able to learn five of the six parameters well. However, we are unable to learn the altitude from these data. This is not surprising, given that our sensitivity analyses showed that altitude played little to no role in the emulator, so the output cannot constrain this particular input. Upon closer investigation, we found that the grid used in the simulations was too coarse to learn anything about the altitude parameter in the range that was specified *a priori* (see Table 1).

Regarding the five categorical parameters that are inputs to WRF, we are able to identify these well in the two synthetic examples (these results are not shown). Most posterior probability (86% and 87% in the two examples) is given to the particular combination of five variables that



(a) Synthetic Calibration 1

(b) Synthetic Calibration 2

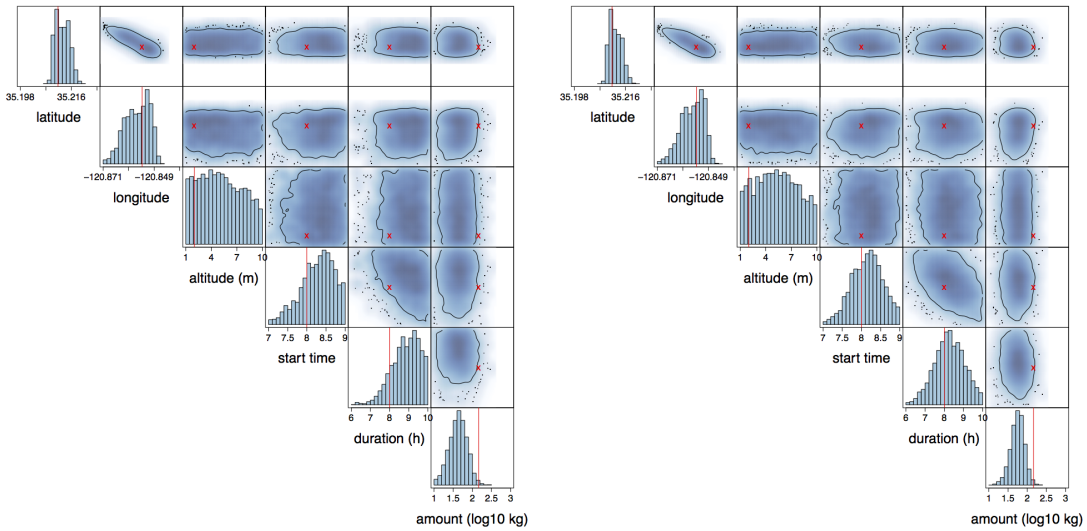
Figure 5: Marginal posterior distributions of continuous parameters in two synthetic calibration problems where we calibrate to model run output. True values are marked with vertical lines and X's. 95% contours are shown in the two-way marginal plots.

are the true settings. Other synthetic calibrations using holdout data have shown that the land surface model can be difficult to identify in some cases (results not shown).

5 Inferring the Source of a Diablo Canyon Release

In this section, we present the results of the case study. We first discuss calibration results under two different discrepancy settings: (1) assuming no discrepancy (i.e., $\delta = 0$) and (2) that the modularized discrepancy discussed above. Figure 6 shows the one- and two-dimensional marginal posterior distributions of the continuous parameters for the no discrepancy and modularized discrepancy cases. The true values of the parameters are also shown in Figure 6, from which we see that latitude and longitude are well identified under both discrepancy models. As can be seen from its nearly uniform posterior distribution, altitude is not well identified for the reason discussed in the previous section. The differences between these calibrated release location parameters are negligible under the two different discrepancy models. However, the release timing parameters (start time and duration) exhibit a fairly significant shift. The discrepancy model helps to reduce the bias of the timing parameters. While both models result in a biased amount parameter, the bias is reduced when a discrepancy model is included.

The bias in the calibrated amount parameter is most likely due to the fact that the amount is highly correlated with the maximum (over space and time) of a model run. The maximum is important because we are using the log scale, so other measurements may be orders of magnitude smaller and thus would not reflect much difference in the amount. Since the emulator is more smooth than the observations, the emulator maximum tends to be smaller than that seen in the observational data. Hence, we did not have a bias in our calibrated amount under the synthetic calibrations discussed earlier, as the synthetic data are more smooth, and maximum values tend



(a) Calibration Without Discrepancy

(b) Calibration With Discrepancy

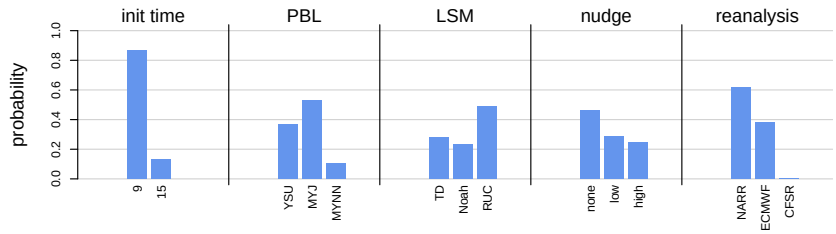
Figure 6: Marginal posterior distributions of continuous parameters obtained from calibrating to real data. True values are shown with vertical lines and X's. The left panel does not use discrepancy while the right panel does. 95% contours are shown in the two-way marginal plots.

to be well predicted by the emulator.

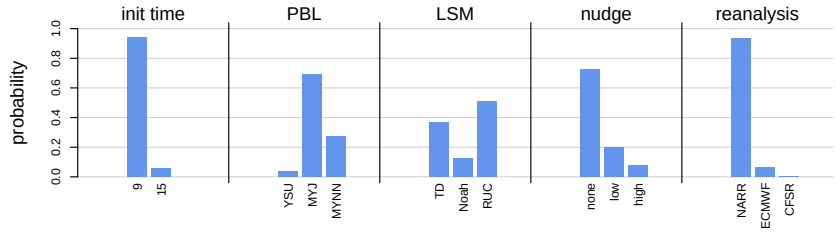
In Figure 7, the posterior marginal distributions of the categorical variables are shown for the two discrepancy cases. These distributions show significant change when a discrepancy model is used. The distribution of each parameter becomes less varied when the discrepancy is included. Notably, the reanalysis parameter strongly favors the North American model over the European model when the discrepancy is included. The distribution of the planetary boundary layer physics parameters is altered to give little mass to the YSU setting. There is also a much stronger preference for no nudging when discrepancy is included.

While the original goal of this analysis were to test the calibration methodology on an experimental release where many of the release characteristics were known, the field study was conducted over three decades ago at a time when global positioning satellites were not available. As a result, there is a discrepancy between the coordinates of the release location in the original field study documentation (UTM 695368 Easting, 3898440 Northing, and zone 10) and the qualitative description of the release site in the paper by Thuillier (1992). Our initial analysis showed that our estimate (posterior mean) of the location of the release (latitude and longitude) was biased by about 100 meters and more consistent with location described in Thuillier (1992), which indicates that the release occurred at the base of Containment Unit 2 (the southmost containment unit). The corrected release location is supported by our calibration analysis, indicating it is much more probable than the originally reported release location, as shown in Figure 8.

As discussed previously, our calibration approach fixed the shape of the model discrepancy before inferring the calibration parameters. The scale parameter γ, which would inflate or deflate the effect, preferred a deflated discrepancy (95% probability interval of (0.58, 1.02) for γ). The model discrepancy inferred from the data can be a useful tool in understanding what parts of the



(a) Calibration Without Discrepancy



(b) Calibration With Discrepancy

Figure 7: Marginal posterior distributions of categorical parameters obtained from calibrating to real data. The top panel does not use discrepancy while the bottom panel does. Unlike the continuous parameters, the true values of these parameters are unknown.

model could be explored further to improve predictive accuracy. We perform an exploratory data analysis of the discrepancy to try to determine its meaning. We see common shapes in groups of the discrepancy time series. When we cluster these time series, we partition our spatial field into areas where the discrepancy time series look similar. The clustered time series and the accompanying locations are shown in Figure 9. Clustering was performed using the fdakma R package (Parodi et al., 2015) with a K-means (but using the medians) type of algorithm that calculated the L2 distance between the first derivatives (approximated by differencing) of the time series. Thus, cluster membership was determined based on similarity of the shape, rather than amplitude of the curves. The number of clusters was selected based on visual assessment of the clustering results. The more interesting parts of our discrepancy are in clusters three and five, shown in Figure 9. Cluster three corresponds to locations that would be in the early path of the plume under the meteorological conditions we observe. The shape of the time series in cluster three seems to indicate a discrepancy in the timing of the plume reaching those locations, implying that the simulator's early concentrations are too high and late concentrations are too low. This matches the fact that our inference regarding the timing parameters was improved when we included this discrepancy model. As shown in Figure 6, excluding a discrepancy results in later inferred start time. Cluster five corresponds to the spatial locations closest to the release. These have largest discrepancy likely because of the high concentration of the plume just after the release. With only a slight perturbation of the wind conditions, the early plume predictions can be in completely different locations because the plume is so concentrated. Thus, if wind predictions are only slightly inaccurate, the early plume predictions yield a large model discrepancy.

Our calibrated predictions with and without discrepancy for the locations in cluster five

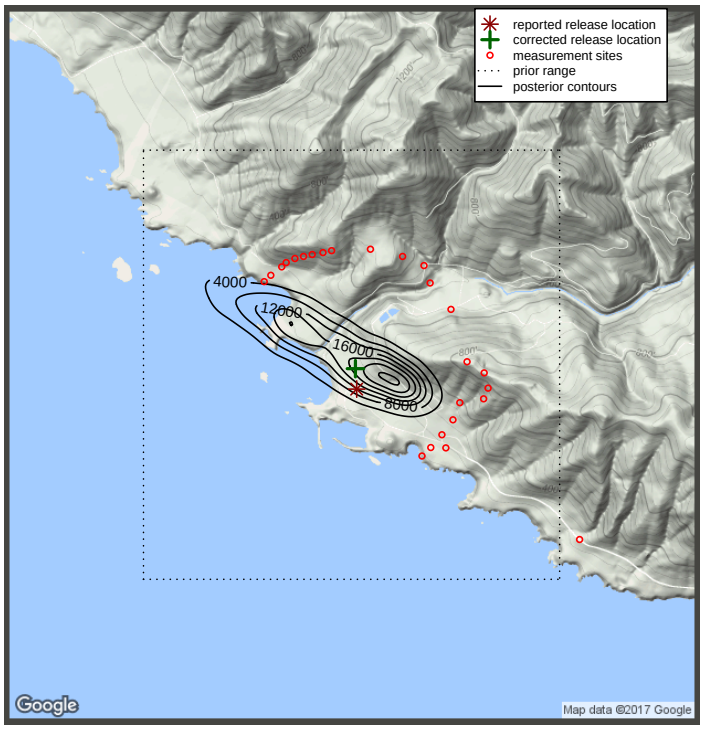


Figure 8: The marginal posterior distribution of the latitude and longitude parameters, as well as the reported location (see text for details).

are shown in Figure 10. These show that including discrepancy brings a significant benefit at those locations while not being overly complex in shape. These also show that the extrapolated predictions including discrepancy can be fairly inaccurate. Indeed, the curves shown in Figure 9 show that the temporal extrapolation is linear. As in all discrepancy functions that are motivated by data rather than scientific input, this extrapolation should not be trusted. Note that one of the locations, site 326, is missing all of the observations. Because site 326 is located such that it is not much of a spatial extrapolation (not shown), this prediction may be trustworthy.

To see the broader effect of including discrepancy on calibrated prediction, we show the predicted (posterior mean) versus observed data in Figure 11. This shows that the discrepancy model, while simple, generally improves prediction. The improvement is most significant for large values, corresponding to the location clusters closest to the release.

6 Conclusion

We have presented an analysis of a computer experiment with important applications to locating and assessing an atmospheric release. In the process, we have developed emulation methodology that can be scaled for use with large numbers of model runs when each has functional output. We have extended existing BMARS methodology to allow for categorical inputs and to model in a reduced dimension space. In building this emulator, we have detailed how to keep track of different sources of uncertainty. We have extended modular approaches to computer model

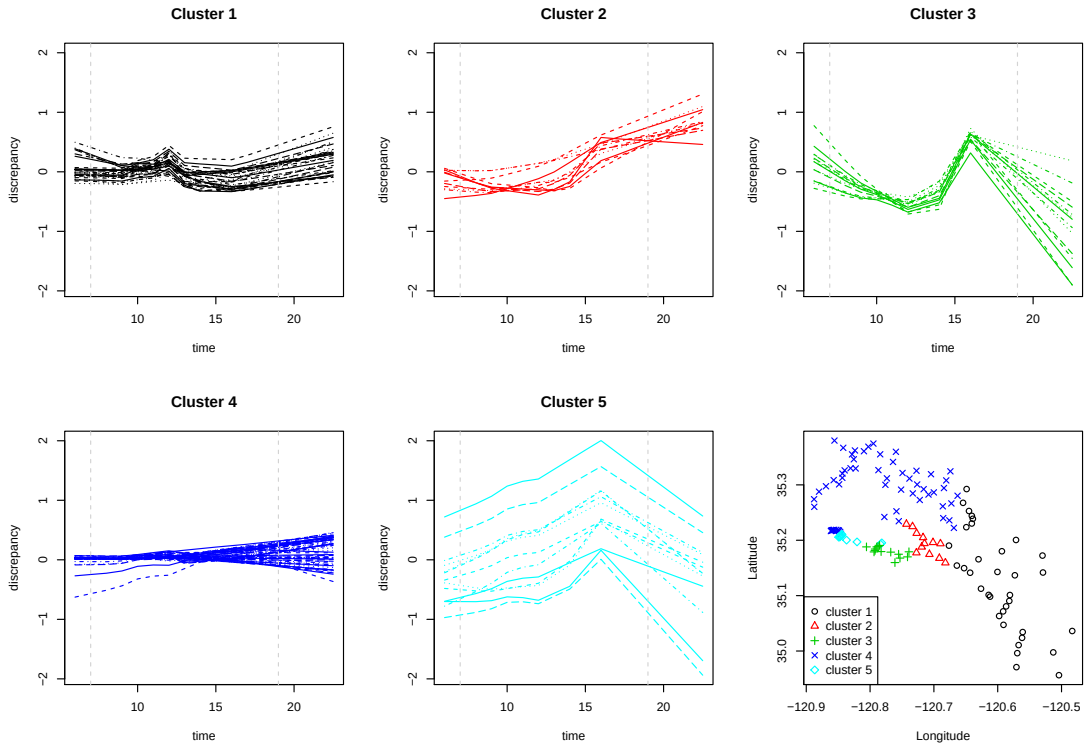
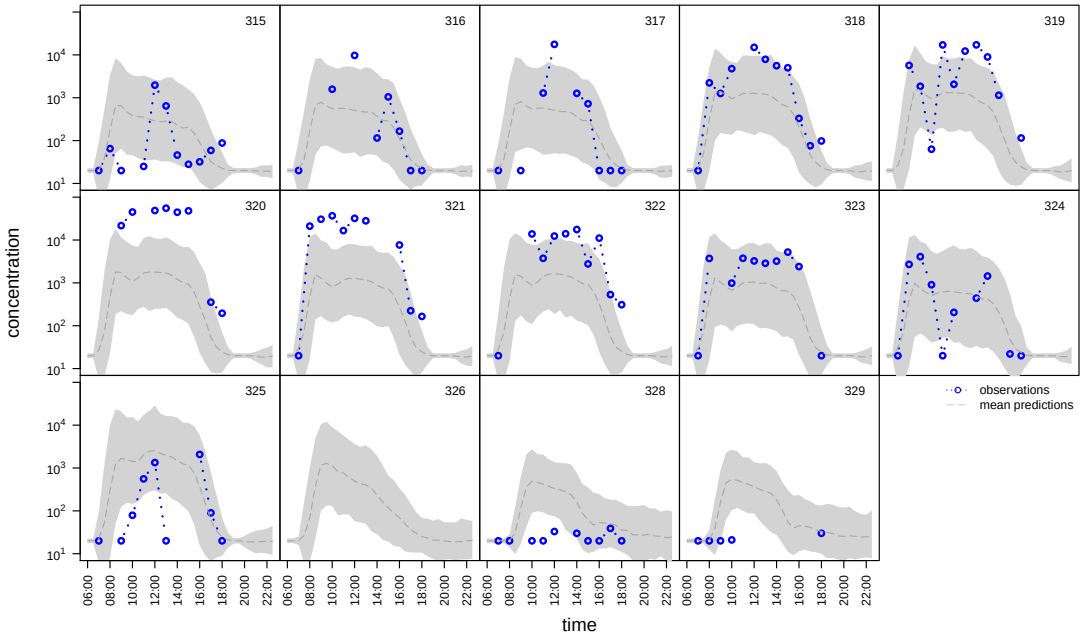


Figure 9: Clusters of discrepancy time series according to shape. The bottom right plot shows locations marked by cluster membership. The vertical dotted lines in the time series represent interval limits outside of which extrapolation begins.

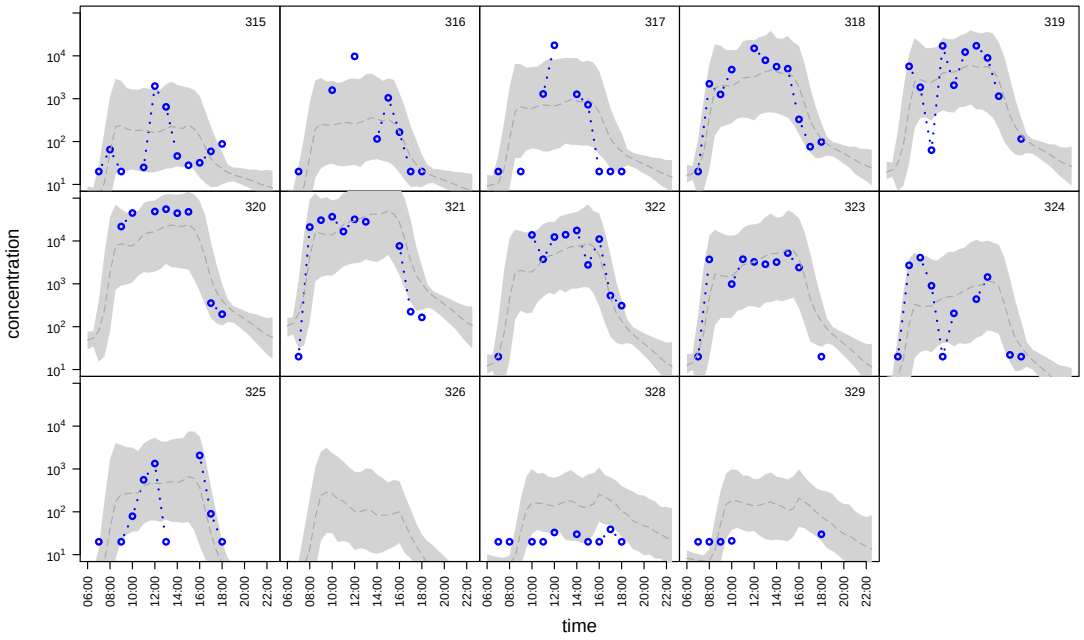
calibration for use with a BMARS emulator and a BMARS discrepancy function, paying special attention to identifiability.

The immediate applications of this work are for research and development purposes rather than emergency response. In an actual emergency, there would be limited time to run WRF (in forecast mode) to get the wind fields for use in FLEXPART. However, an approach that uses weather analogues (Delle Monache et al., 2011) rather than WRF runs may be feasible. An ensemble of FLEXPART simulations could be obtained in parallel, and emulation and calibration models could be fit thereafter. In order to be useful, emulation and calibration need to be done quickly and accurately. Hence, the emulation and calibration methodology outlined here has potential to be useful.

A possible extension of the proposed model would be to include functional input. That is, rather than parameterizing WRF with five categorical parameters, we could include the entire spatio-temporal wind field as input for the emulator. This may be possible by decomposing wind fields onto a set of basis functions and including the basis function weights as inputs to the BMARS emulator.



(a) Prediction Excluding Discrepancy



(b) Prediction Including Discrepancy

Figure 10: Calibrated prediction for the locations in cluster 5. The top panel shows prediction excluding discrepancy, while the bottom panel includes discrepancy. The 95% pointwise probability intervals do not include the observational error, which is Normal with posterior standard deviation near 0.39 (95% probability interval (0.35, 0.43) for σ_F, symmetry coincidental).

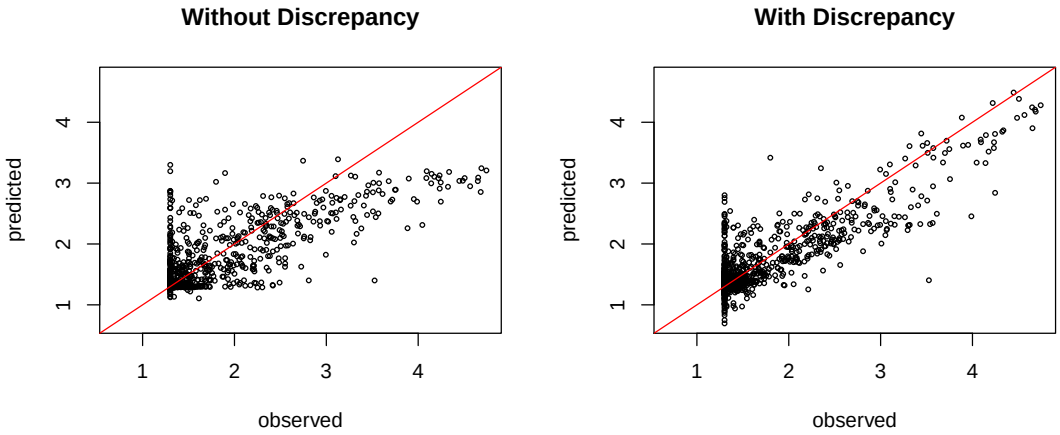


Figure 11: Calibrated prediction versus the observations, with and without the discrepancy in prediction.

7 Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344 and was funded by Laboratory Directed Research and Development at LLNL under project tracking code PLS-14ERD006. The manuscript is released under UCRL number LLNL-JRNL-732282. The authors thank PG&E for access to the Diablo Canyon measurement data. The authors also thank Matthew Simpson, Ronald Baskett and Philip Cameron-Smith from LLNL for helpful discussions about the simulations and measurement data.

References

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.

Chakraborty, A., Mallick, B. K., Mcclarren, R. G., Kuranz, C. C., Bingham, D., Grosskopf, M. J., Rutter, E. M., Stripling, H. F., and Drake, R. P. (2013), "Spline-based emulators for radiative shock experiments with measurement error," *Journal of the American Statistical Association*, 108, 411–428.

Delle Monache, L., Nipen, T., Liu, Y., Roux, G., and Stull, R. (2011), "Kalman filter and analog schemes to postprocess numerical weather predictions," *Monthly Weather Review*, 139, 3554–3570.

Denison, D. G., Mallick, B. K., and Smith, A. F. (1998), "Bayesian MARS," *Statistics and Computing*, 8, 337–346.

Francom, D. (2017), *BASS: Bayesian Adaptive Spline Surfaces*, R package version 0.2.2.

Francom, D., Sansó, B., Kupresanin, A., and Johannesson, G. (2016), "Sensitivity Analysis and Emulation for Functional Data using Bayesian Adaptive Splines," *Statistica Sinica*, in press.

Friedman, J. H. (1991a), "Estimating Functions of Mixed Ordinal and Categorical Variables Using Adaptive Splines," Tech. rep., DTIC Document.

—— (1991b), "Multivariate adaptive regression splines," *The annals of statistics*, 1–67.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013), *Bayesian data analysis*, CRC press.

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

Gramacy, R. B., Bingham, D., Holloway, J. P., Grosskopf, M. J., Kuranz, C. C., Rutter, E., Trantham, M., Drake, R. P., et al. (2015), "Calibrating a large computer experiment simulating radiative shock hydrodynamics," *The Annals of Applied Statistics*, 9, 1141–1168.

Green, P. J. (1995), "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, 82, 711–732.

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

Kaufman, C. G., Bingham, D., Habib, S., Heitmann, K., and Frieman, J. A. (2011), "Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology," *The Annals of Applied Statistics*, 2470–2492.

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

Liang, F., Paulo, R., Molina, G., Clyde, M. A., and Berger, J. O. (2008), "Mixtures of g priors for Bayesian variable selection," *Journal of the American Statistical Association*, 103.

Liu, F., Bayarri, M., Berger, J., et al. (2009), "Modularization in Bayesian analysis, with emphasis on analysis of computer models," *Bayesian Analysis*, 4, 119–150.

Lucas, D. D., Simpson, M. D., Cameron-Smith, P., and Baskett, R. L. (2017), "Bayesian inverse modeling of the atmospheric transport and emissions of a controlled tracer release from a nuclear power plant," *Atmospheric Chemistry and Physics Discussions*, 2017, 1–36.

Ma, S., Racine, J. S., and Yang, L. (2015), "Spline regression in the presence of categorical predictors," *Journal of Applied Econometrics*, 30, 705–717.

Nott, D. J., Kuk, A. Y., and Duc, H. (2005), "Efficient sampling schemes for Bayesian MARS models with many predictors," *Statistics and Computing*, 15, 93–101.

Parodi, A., Patriarca, M., Sangalli, L., Secchi, P., Vantini, S., and Vitelli, V. (2015), *fdakma: Functional Data Analysis: K-Mean Alignment*, R package version 1.2.1.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), "Design and analysis of computer experiments," *Statistical science*, 409–423.

Skamarock, W. C., Klemp, J. B., Dudhia, J., Gill, D. O., Barker, D. M., Wang, W., and Powers, J. G. (2008), "A description of the advanced research WRF version 3," Tech. Rep. NCAR/TN-475+STR, National Center For Atmospheric Research Boulder Co Mesoscale and Microscale Meteorology Div.

Stohl, A., Forster, C., Frank, A., Seibert, P., and Wotawa, G. (2005), "Technical note: The Lagrangian particle dispersion model FLEXPART version 6.2," *Atmospheric Chemistry and Physics*, 5, 2461–2474.

Storlie, C. B., Lane, W. A., Ryan, E. M., Gattiker, J. R., and Higdon, D. M. (2015), "Calibration of computational models with categorical parameters and correlated outputs via Bayesian smoothing spline ANOVA," *Journal of the American Statistical Association*, 110, 68–82.

Stripling, H., McClarren, R., Kuranz, C., Grosskopf, M., Rutter, E., and Torralva, B. (2013), "A calibration and data assimilation method using the Bayesian MARS emulator," *Annals of Nuclear Energy*, 52, 103–112.

Thuillier, R. H. (1992), "Evaluation of a puff dispersion model in complex terrain," *Journal of the Air & Waste Management Association*, 42, 290–297.

United Nations Scientific Committee on the Effects of Atomic Radiation (2008), "Sources and effects of ionizing radiation. UNSCEAR 2008 report to the General Assembly, with scientific annex," Volume II.

Zellner, A. (1986), "On assessing prior distributions and Bayesian regression analysis with g-prior distributions," *Bayesian inference and decision techniques: Essays in Honor of Bruno De Finetti*, 6, 233–243.