SANDIA REPORT
2023-09749R
Printed September, 2023

Sandia National Laboratories

A spatially regularized detector for emergent/re-emergent disease outbreaks

Jaideep Ray, Cosmin Safta, Wyatt Bridgman, Maya Horii and Aidan Gould

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods

ABSTRACT

Early detection of outbreaks caused by emergent pathogens, using epidemiological surveillance data i.e., daily case counts, is difficult. This is because the data tend to be noisy during the early epoch of the outbreak. In contrast, the spread-rate of the disease tends to be well-behaved, as it depends only on the mixing patterns of the population and the characteristics of the pathogen, neither of which behave erratically in space-time. In this report, we explore whether the spread-rate can be used for epidemiological surveillance, conditional on case count data. Estimating the spread-rate from case count data allows us to exploit exogenous information, e.g., incubation period distributions etc., which can considerably smooth out any erratic temporal behavior. Further, epidemiological dynamics are spatially correlated, and if case counts are available for multiple areal units e.g., counties, these correlations could potentially be used to suppress noise in the early epoch data. These exogenous information and structure are not exploited by conventional syndromic surveillance detectors to extract a well-behaved latent variable for monitoring purposes. The technical challenge lies in the estimation of the spread-rate *field* jointly over a collection of areal units; further, the spread-rate varies over time. We develop a method based on *mean-field variational inference* to approximately estimate the spread-rate field, using a Gaussian Random Field Model for spatial regularization. The method is tested on the estimation of spread-rate in the thirty-three counties of New Mexico and detect the arrival of the Fall 2020 COVID-19 wave in Sepetember 2020. We find that the method is scalable, but underestimates the uncertainty in the estimated spread-rate field. We detect the arrival of the Fall 2020 wave a week ahead of conventional syndomic surveillance algorithms, but our simplistic detection algorithm, based on simple anomaly detection, suffers from a high false positive rate, similar to conventional detectors.

This page intentionally left blank.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In this report we document our investigation into developing a disease outbreak detector for emergent pathogens, using data available in the early epoch of the outbreak. This consists of daily case counts and knowledge of the incubation period of the disease. In the early epoch, the data tend to be noisy, with low case counts, and conventional outbreak detectors that are formulated as anomaly detection algorithms struggle to identify outbreaks until they are well-established. This is also partially due to the paucity of data and the inability of conventional outbreak detectors to exploit risk factors (e.g., socioeconomic data) as well as pathogen characterizations (e.g., incubation period distributions) which are often available during that time. Thus robust early detection of outbreak of novel pathogens have, to date, eluded us.

We explore an alternative detector that uses the (latent) spread-rate of a disease as the monitoring variable (as opposed to the raw case-counts). This is based on the hypothesis that the spread-rate (also known as the infection-rate) is a more stable monitoring variable (versus case counts), as it is fully determined by the mixing patterns of the population and the characteristics of the pathogen, neither of which change erratically day-to-day. The technical challenge lies in estimating this latent variable from the noisy case count data stably, and devising an anomaly detection problem with it to detect a change in the epidemiological dynamics (e.g., the arrival of a new wave of infection). To do so, we will develop an estimation algorithm based on mean-field Variational Inference (VI) and compare the performance of our outbreak detector versus a conventional one. We will test the method using data from the COVID-19 outbreak in the 33 counties of New Mexico; a conventional detector called the "RKI detector" will serve as the conventional equivalent for comparison [19].

We will build on our previous work [41, 7] that developed a technique to estimate a time-dependent spread-rate in a given areal unit, usually a state in the USA. The estimation is posed as a 6-dimensional Bayesian inverse problem and solved using Markov chain Monte Carlo (MCMC) methods. We will extend it to address the estimation of a spread-rate *field* defined over the 33 counties of NM, and devise a VI algorithm to perform the resultant high-dimensional estimation, along with a random field model to impose spatial correlations in the epidemiological dynamics. Thereafter, we will formulate an anomaly detection problem, conditioned on the estimated spread-rate, to detect the arrival of the Fall 2020 COVID-19 wave in NM, which occurred on September 15. We will compare its performance against the RKI detector.

The research questions are:

1. What exogenous (i.e., socioeconomic) covariates can serve as risk factors for COVID-19 in NM?

2. What is the spatial correlation model that could be used to regularize the estimation of the spread-rate field in NM?

3. How does one formulate an inverse problem for the spread-rate and embed the spatial regularization in it? Can this be solved with MCMC for a few NM counties? What is the predictive skill of the estimated spread-rate and can it be used to detect the arrival of the Fall 2020 wave? Does it detect it earlier than the RKI detector?

4. Can the inversion be scaled to all 33 NM counties using mean-field VI? What is the formulation of the inverse problem that is stable even when the data is very noisy and barely informative (a characteristic of the remote desert counties of NM)? Since VI is approximate, what are the consequences to the estimated spread-rate? Can the spread-rate be corrected for the approximation and used in an anomaly detector? Does it detect the arrival of the Fall 2020 wave?

In addition to the disease detector, the report also documents joint work with University of California, Berkeley, to construct a calibration algorithm for agent-based (disease) models (ABMs). ABMs are stochastic, which makes their calibration to data very challenging. We will pursue two methods, one based on Approximate Bayesian Computations (ABC) and the other involving MCMC with surrogate models, to do so.

Posing the anomaly detector in terms of a spread-rate field has a few key advantages. Public health interventions are invariably of a local (county) nature, which require us to estimate the spread-rate in a given areal unit. However, areal units like counties vary greatly in their population (implying a diversity in the case count magnitudes), their socioeconomic characteristics and their public health infrastructure, and consequently the quality of the data may not allow the estimation of spread-rates in individual areal units independently. Inferring the spread-rate field allows us to exploit spatial correlations to smooth over low quality data, and compute a spread-rate; if couched in Bayesian terms, the uncertainty in the estimate may also be captured. Our formulation is thus a way of ensuring robustness to poor data quality in the outbreak detector.

The report is structured as follows. In Chapter 2, we formulate and solve the inversion problem with MCMC, and also formulate and test a disease detector. In Chapter 3, we reformulate and solve the problem with VI. Chapter 4 contains our investigation into calibration of ABMs. We conclude in Chapter 5.

2. FORMULATION

2.1. Introduction

In this section we formulate the inverse problem for the spread-rate field (also called the infection-rate field) using COVID-19 data from the 33 counties of NM and devise an anomaly detection algorithm to detect the arrival of the Fall 2020 wave. This includes performing exploratory data data analysis of epidemiological covariates that partially explain the spatial correlations observed in the dynamics of case count data, as well as elucidating the spatial model.

2.2. Literature review

Our project builds on a previous one ("A Statistical Model for the Spread of SARS-CoV-2 in New Mexico", PI: Lyndsay Shand, ending in September 2020 and documented in Ref. [42]). In Ref. [42], the authors targeted the forecasting of case-counts in the counties of New Mexico, using data till July 2020. They did not estimate a latent spread-rate, but rather modeled the case counts using a SIR (Susceptible-Infectious-Recovered) model, with a multinomial model predicting the new daily (Infectious) case counts, and a binomial model for the Recovered. The probabilities in the multinomial model were modeled by linearly regressing to exogenous covariates (which captured most of the spatial patterns) while forecasting in time was performed using an AR1 time-series model. Random spatial effects (spatial correlation between counties) were modeling using a Gaussian Markov Random field, with the connectivity matrix being designed using roads and highway connections. Sparse PCA (Principal Component Analysis) was used to simplify the exogenous covariates, which found that the population and the availability of COVID-19 tests were the best predictor of case counts. Multiple models of different complexities were posed, fitted with the data available till July 2020, and then selected using Deviance Information Criterion and a few other measures of predictive skill of the model. They found that ignoring spatial effects (beyond what the exogenous covariates contributed) and removing unimportant covariates via sparse PCA yielded the best models.

The use of spatial statistics to impose spatial correlations as a prior has long been used in disease mapping [4, 49]. The case counts in an areal unit m is usually modeled as a Poisson distribution $Y_m \sim \text{Poisson}(\exp(S)E_m)$ where E_m is an expected value computed from covariates and S is a relative risks field used to impose any spatial patterns via $S \sim p(\cdot \mid \theta), \theta \sim \pi()$. The diversity lies in the prior $p(\cdot \mid \theta)$. Most simply, it may be a multivariate normal (i.e., a Gaussian Random Field, GRF), tying areal units together. It may also be modeled in a conditional manner, using Gaussian *Markov* Random Fields, of which the famous BYM model is an example [3]. If the relative risks evolve in time, one adds an auto-regressive (ARn) process to $\log(S)$; see Ref. [49] for spatiotemporal disease maps.

Our method is a modification of a spread-rate estimation technique described in Ref. [41], and extends it to multiple areal units, while simultaneously imposing a spatial regularization on the estimated spread-rate. The original method in Ref. [41] starts with a parameterization for the temporally evolving spread-rate in an areal unit, modeled using a Gamma function. This parameterized spread-rate profile is convolved with

an incubation period distribution for COVID-19, modeled as a log-normal distribution. The convolution accounts for the delay between infection and the appearance of symptoms, at which point people are detected and appear in the case count data. There are two other parameters to account for the magnitude of the spread-rate and the start of the outbreak. The formulation of the estimation problem is Bayesian, with the data - model mismatch being modeled as a zero-mean Gaussian with a standard deviation that is also parameterized to be proportional to the case counts on a given day. It is a six-dimensional inversion and was demonstrated on data from a number of US states and countries, i.e., the case-counts were large and the data generally of good quality with little noise. On estimating the posterior probability density of the parameters, we sampled it, created realizations of the spread-rate profile and proceeded to compute 15-day-ahead forecasts (i.e., over a time duration *not* included in the data used for estimating the spread-rate); the fan of forecasts usually bounded the observations. However, during the Fall 2020 wave, the observations fell outside the forecasts, indicating a change in epidemiological dynamics (and sparking our idea of using this technique for anomaly & outbreak detection). In Ref. [7], the authors proposed a one-wave and a two-wave model for the spread-rate and formulated a model-selection scheme, based on Akaike Information Criterion, to choose between the two; the 2-wave model naturally had more parameters to infer. This was successfully performed, indicating that our method could be used to detect the arrival of a wave. What was left unchecked was whether the model-selection would be useful in the absence of a prior belief regarding the existence of a second wave and whether the model-selection could be carried out with lower quality, noisy data, that would be characteristic of case counts from a smaller areal unit, such as a county.

In this report we will compare our ability to detect the arrival of the Fall 2020 COVID-19 wave in NM versus a conventional detector, colloquially called the “RKI" detector because it was demonstrated on Salmonella infection data from the Robert Koch Institute, Germany; see Ref. [19]. Conventional detectors are based on stochastic process control charts, where time-series data is used to learn a model, predict a distribution for future observations and then decide, on receipt of the data, the probability that they were drawn from the predicted distribution. In the RKI detector, the observed data is divided into two parts, a long “in-control” part that precedes the date when the outbreak is suspected to have started and a shorter “out-of-control” window after the suspected date where the data is believed to follow a different epidemiological dynamic. The data in both the parts are modeled as draws from a negative binomial distribution with means $\mu_0(t)$ and $\mu_1 = \kappa\mu_0(t)$, and the same dispersion α. $\mu_0(t)$ is learned from the “in-control” data, and can include exogenous covariates. κ is learned from previous outbreaks, by optimizing the false positive and false negative rates of the detector, an impossibility for an emergent pathogen. A change in epidemiological dynamics is detected via a generalized likelihood ratio test, posed by querying whether the “out-of-control” data is drawn from the distribution with mean μ_0 or μ_1. This also requires one to select a threshold for the likelihood ratio, which is also determined from past performance, or through simulated outbreaks (which may bear no resemblance to a new pathogen).

2.3. Exploratory data analysis

Ref. [42] develops a spread-rate model for NM using 79 exogenous covariates such as demographics and socioeconomic factors. These covariates exhibit spatial correlations and as such could explain some of the spatial patterning seen in epidemiological dynamics. Consequently, we model $c_i(t)$, the case counts at time t, summed over the previous 3 months, for each NM county m, as $\boldsymbol{c}(t) = \boldsymbol{X}\boldsymbol{w}(t)$, $\boldsymbol{c} = \{c_m\}, m = 1 \ldots M$, where $\boldsymbol{X}$, is a $M \times N_c$ matrix of $N_c = 79$ covariates, $M = 33$ counties and $\boldsymbol{w}(t)$ are the weights. In order to simplify the correlated covariates in $\boldsymbol{X}$, we perform a sparse PCA (Principal Component Analysis,

Ref. [15, 14]) so that the case counts could be modeled as

$$\frac{\boldsymbol{c}(t)}{\boldsymbol{P}} \approx C + \sum_{k=1}^{k=K} v_k(t)\phi_k, \tag{2.1}$$

where P_m is the population of county m, $\boldsymbol{P} = \{P_m\}$, ϕ_k are the principal components and C is a constant. This analysis is performed with COVID-19 case counts collected over April 1, 2020 to January 24, 2022, with the 90-day window being advanced by 30 days at a time. Fig. 2-1 plots $v_k(t)$ over the 2-year period. It indicates that the intercept C is the dominant term, while the others fluctuate, possibly due to numerical errors.

In order to fully explore the issue of whether any of the exogenous covariates play a significant role in deciding case counts, we take recourse to a sparse model. Since the sparse principal components ϕ_k are not orthogonal, we refit the coefficients $v_k(t)$ via LASSO. This analysis is performed with all the COVID-19 case counts over 2 years, pooled together. In Fig. 2-2a we plot the variation of $c_i(t)$ explained as we add principal components to the model; $K = 10$ captures 90% of the variation. In Fig. 2-2b, we plot the coefficients $v_k(t)$ over the 2-year dataset; we find that the intercept C is far larger than the rest of the coefficients, indicating that none of the covariates play much role in predicting $c_i(t)$. Thus the case counts are proportional to the population, and any temporal variation in $v_k(t)$ are due to changes in population mixing and changing characteristics of the disease itself.

Finally, we explore the spatial structure of $\boldsymbol{c}(t)/\boldsymbol{P}$. In Fig. 2-3, we plot the residual $r_m = c_m(t)/P_m - C$, the portion of the case counts that cannot be explained by the counties' population alone. r_m is standardized using its mean and standard deviation for plotting convenience. We see clear signs of spatial correlation, with the blue counties approximately along the Rio Grande Valley and yellow ones in the Northwest and Southeast. While the figure illustrates c_i as a sum of the case counts over the 2 years, the analysis was also repeated with sums over 90-day windows, with much the same results. This suggests the existence of spatial patterns of r_m values and we investigate that with Moran's I-test.

Moran's I-test (Chapter 9, Ref. [5]) for a spatial variable $q(x)$ over M areal units is performed in the following manner. We compute the I-statistic as

$$I = \frac{M}{\sum_{ij} w_{ij}} \frac{\sum_{i=1}^{i=M}\sum_{j=1}^{j=M} w_{ij}(q_i - \overline{q})(q_j - \overline{q})}{\sum_{i=1}^{i=M}(q_i - \overline{q})^2}$$

for the data q_i. Here w_{ij} are weights from an $M \times M$ adjacency matrix W for the counties, whose elements are set to 1 if two counties share a border, and zero otherwise. Thereafter W is row-normalized. Having computed I for the data, we proceed to compute a distribution for I under the assumption that q are uncorrelated random values i.e. a "white noise" field. Consequently $(q_i - \overline{q})$ are replaced with random draws from a standard normal and the I-statistic computed again. This is performed multiple times to compute the distribution of I commensurate with an uncorrelated random field and we perform a $p-$value test to check whether I resulting from the true q field differs from the I distribution (a normal distribution) resulting from the "white-noise" q. We find that the standard variate for I resulting from the true q field is 3.91 with a p-value of 4.7×10^{-5} i.e., q is far from being a "white noise" field, and has spatial structure in it.

Finally, we investigate the standard deviate of the I-statistic of the data, computed over 90-day windows over the 2 years of data. The statistic is computed using 3 different ways of determining w_{ij} - the row-normalized manner described above (called mode "W"), a similar version where the normalization is not done and the W matrix is binary (mode "B") and where the weights are inversely proportional to the

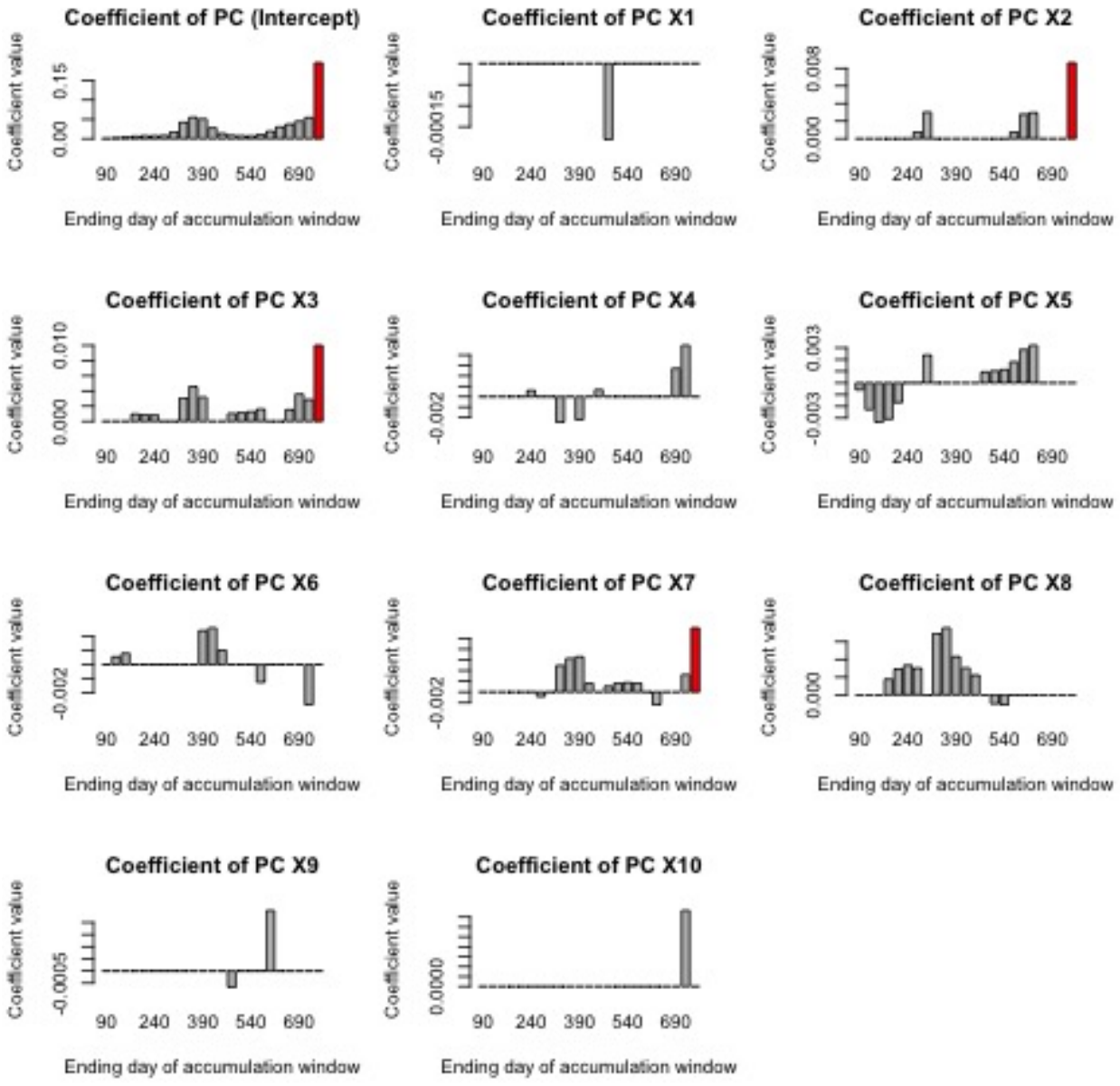

FIGURE 2-1. Coefficients $v_k(t)$, computed over 90-day windows which are advanced one month at a time, from April 1, 2020 to January 24, 2022, for all counties of NM. The last bar in red represents the coefficient computed by pooling the 2-year time-series together.

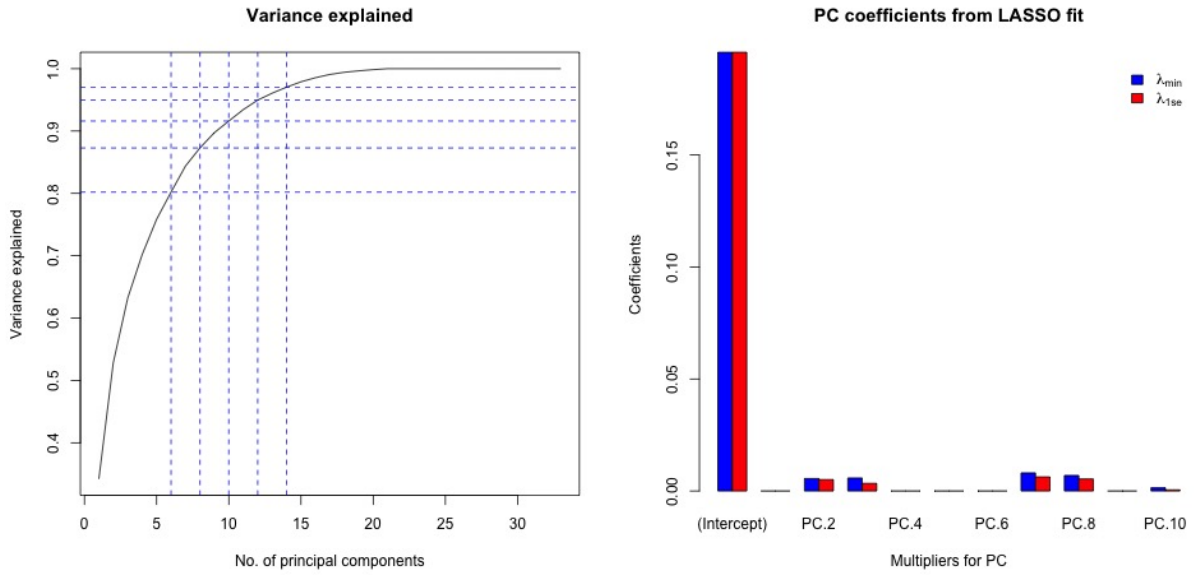

FIGURE 2-2. (a) Scree plot of the principal components. We will use $K = 10$ principal components. (b) Coefficients $v_k(t)$ fitted to all the data using LASSSO.

TABLE 2-1. Distribution of the (standard deviate) of the I-statistic, computed using different weighing models, over 90-day windows, for the two-year of data.

Weighing model	(Mean, standard deviation)
W	(2.849e+00, 1.430e+00)
B	(2.618e+00, 1.139e+00)
B-mod	(2.210e+00, 7.991e-01)

distance between the county seats of the adjoining counties (mode "B-mod"). For each type of weighing models, we get a set of (standard deviates of) I which are summarized by their mean and standard deviations in Table 2-1. It is clear that modes "W" and "B" perform better i.e., spatial patterns are clear and dominant when such a weighing model is used, and we will choose mode "B" for simplicity and because its the standard deviation is smaller. Not mentioned here is a test that we performed for an adjacency matrix where "2-hop" neighbors (neighbors of a county and their neighbors) were included when computing w_{ij}. The Moran I-statistic did not indicate any spatial structures.

2.4. Formulation of the inverse model

The formulation of the inverse problem for the spread-rate field, conditional on case count data from multiple areal units (counties of NM in our case), is an extension of the model, for one areal unit, in Ref. [41]. We first develop the inverse problem for a given areal unit. The epidemiological model combines

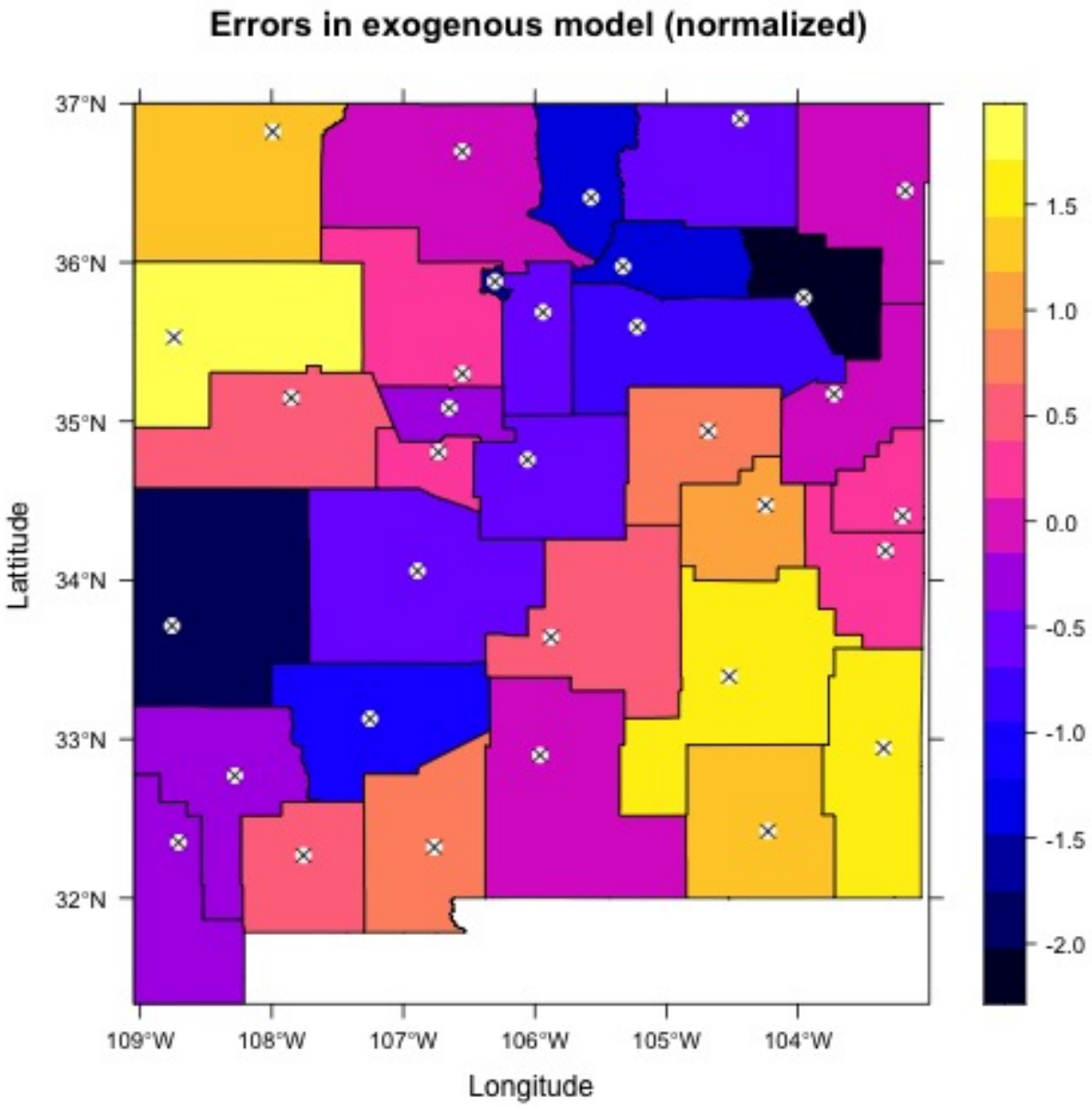

FIGURE 2-3. A plot of standardized r_m, computed as a sum over the two years

an infection-rate model and an incubation period distribution. The infection-rate is assumed to follow a Gamma distribution with a probability density function (pdf) given by

$$f_{inf}(t;k,\theta,t_0) = \theta^{-k}(t-t_0)^{k-1}\exp(-(t-t_0)/\theta)/\Gamma(k). \tag{2.2}$$

The infection-rate in Eq. (2.2) is controlled by two parameters, k (shape) and θ (scale), and is sufficiently flexible to capture a range of outbreaks. The third parameter, t_0, represents the start of the outbreak and will be inferred jointly with the infection rate parameters. For incubation we employ a model calibrated against early COVID-19 data [28]. This model follows a log-normal distribution with a cumulative distribution function (cdf) given by

$$F_{inc}(t;\mu,\sigma) = \frac{1}{2}\text{erfc}\left(-\frac{\log t-\mu}{\sigma\sqrt{2}}\right) \tag{2.3}$$

The mean μ is approximated as a Student's t distribution and σ is assumed to have a chi-square distribution. These choices results in confidence intervals that match the data in Ref [28].

The cumulative number of people that have turned symptomatic between time t_0 and time t_i is computed as a convolution between the infection rate and the CDF (Cumulative Distribution Function) of the incubation model

$$N_i = N\int_{t_0}^{t_i} f_{inf}(\tau-t_0;k,\theta)F_{inc}(t_i-\tau;\mu,\sigma)d\tau, \tag{2.4}$$

where N is the total number of people that will get infected (and counted) during the entire epidemic wave. This model assumes that a person shows symptoms once the virus incubation has completed. Furthermore, once symptoms are evident, it is also assumed that individuals have prompt access to medical services or otherwise self-report the COVID-19 infection, getting counted without delay.

The number of people that turn symptomatic over the time interval $[t_{i-1},t_i]$ is estimated as

$$n_i = y_r(i;t_0^r,N^r,k^r,\theta^r) \tag{2.5}$$

$$n_i = N_i - Ni - 1 = N\int_{t_0}^{t_i} f_{inf}(\tau-t_0;k,\theta)(F_{inc}(t_i-\tau;\mu,\sigma)-(F_{inc}(t_{i-1}-\tau;\mu,\sigma))d\,\tau \tag{2.6}$$

$$\approx N(t_i-t_{i-1})\int_{t_0}^{t_i} f_{inf}(\tau-t_0;k,\theta)f_{inc}(t_i-\tau;\mu,\sigma)d\tau \tag{2.7}$$

where f_{inc} is the PDF (probability density function) of the incubation model. $y_r(i;t_0^r,N^r,k^r,\theta^r)$ is an alternate variable used in Chp. 3.

In this report we focus on outbreak detection and for this purpose a model that follows a single wave, as above, is sufficient for the task. Given the assumptions above, these outbreak forecasts represent a lower bound on the actual number of people that are infected with COVID-19. A fraction of the population infected with a novel disease might also exhibit minor or no symptoms at all and might not seek medical advice, further contributing to lowering the predicted counts compared to the actual size of the epidemic.

We next extend the formulation to multiple areal units. Let Y_i^o be the vector of case counts observed in all M counties on day i. Define $\mathbf{m}_r^T = (t_0^r,N^r,k^r,\theta^r)$, for $r = 1,\ldots,N_{reg}$ as the region-specific spread-rate model parameters, with N_{reg} denoting the number of regions (or areal units). Let $\boldsymbol{p} = \{\mathbf{m}_r\}, r = 1,\cdots,N_{reg}$ be the complete set of spread-rate model parameters defined over all areal units/regions. Let $Y_i^{(p)}(\boldsymbol{p})$ be the predictions using the model Eq. 2.7, invoked N_{reg} times using parameters $\mathbf{m}_r$. We assume that the errors

$$Y_i^o - Y_i^{(p)}(\boldsymbol{p}) = \boldsymbol{\varepsilon_i}, \boldsymbol{\varepsilon_i} \sim N(0,\Sigma_i),$$

where the $M \times M$ covariance matrix Σ_i embodies the spatial correlations seen in Fig. 2-3. This makes $\boldsymbol{\varepsilon_i}$ a realization of a GRF. We define a parameter set $\boldsymbol{\theta}$ for all parameters in the inverse problem i.e.,

$$\boldsymbol{\theta} = \text{vec}(\boldsymbol{p}, \boldsymbol{\eta}) = \text{vec}\left(\begin{bmatrix}\mathbf{m}_1 & \cdots & \mathbf{m}_{N_{reg}} & \boldsymbol{\eta}\end{bmatrix}\right), \tag{2.8}$$

where $\boldsymbol{\eta} = (\tau_\Phi, \lambda_\Phi, \sigma_a, \sigma_m)$ are the global noise parameters. The likelihood expression for data collected over N_d days is then

$$\mathscr{L}_{\mathscr{D}} = \prod_{i=1}^{N_d} \frac{1}{(2\pi)^{N_r/2}\det\Sigma_i^{1/2}} \exp\left(-\frac{1}{2}(Y_i^{(o)} - Y_i^{(p)}(\boldsymbol{p}))\Sigma_i^{-1}(Y_i^{(o)} - Y_i^{(p)}(\boldsymbol{p}))^T\right) \tag{2.9}$$

where Σ_i is constructed as

$$\Sigma_i = \frac{\tau_\Phi^2}{1-\lambda_\Phi^2}\left[I + \lambda_\Phi W\right] + \text{diag}\left(\sigma_a + \sigma_m Y_i^{(p)}\right)^2. \tag{2.10}$$

Here, the entries of W are defined as

$$w_{kk} = 0 \,\text{and}\, w_{kl} = \begin{cases} 1 & \text{if regions } k \text{ and } l \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

The dimensionality of this inverse problem is $4N_r + 4$ and solving for all $M = 33$ counties constitutes a 136-dimensional inverse problem. Note that since $\boldsymbol{\eta}$ is shared across all M areal units, and $\{\tau_\Phi, \lambda_\Phi\}$ are parameters of the spatial parameterization, the inverse problem cannot be solved for each areal unit independently. Uniform priors are used for all the parameters. The inversion is performed using an adaptive Markov chain Monte Carlo method [16].

2.5. Inversion results

Due to MCMC's inability to solve very high-dimensional inverse problems, we limit ourselves to three counties of New Mexico *viz.* Bernalillo, Santa Fe, and Valencia. The 16-dimensional inverse problem is solved using case count data between June 1, 2020 and August 15, 2020. Thereafter, 100 samples of $\boldsymbol{\theta}$ are drawn from the posterior and used to predict the the case counts as well as forecast 15 days ahead. In order to facilitate a comparison, we also solve the inverse problem for each of the counties independently. In this case, $\Sigma_i = \text{diag}\left(\sigma_a + \sigma_m Y_i^{(p)}\right)^2$ and the problem is 6-dimensional. Since the Fall 2020 wave arrived a month later, the forecasts should be predictive.

In Fig. 2-4 we compare the marginalized posterior distributions from the joint (blue) and independent (red) estimation of the epidemiological model parameters $\mathbf{m_r}$ for the three NM counties. The prior is plotted with a black line. The parameters were estimated with data between June 1, 2020 and August 15, 2020. It is clear from the last column (that plots $\boldsymbol{\eta}$) that the parameters of the spatial model (τ_ϕ, λ) can be estimated since the PDF (probability density function) shows a clear peak. For all 3 counties, $\mathbf{m_r}$ are usually well estimated i.e., the difference between the prior and posterior distribution are quite clear. The blue and red posterior densities have differences, but the joint estimation always results in a narrower PDF, showing the constraining power of correlations (alternatively, the effect of borrowing information from neighboring counties). It is this ability to obtain crisp PDFs that led us to attempt joint estimation, despite the fact that it resulted in a high-dimensional inversion, with all its attendant difficulties. This gives us hope that when we attempt to estimate the spread-rate in all 33 counties of NM, the ability to "borrow information" from neighbors will help us stabilize the counties with very low quality data.

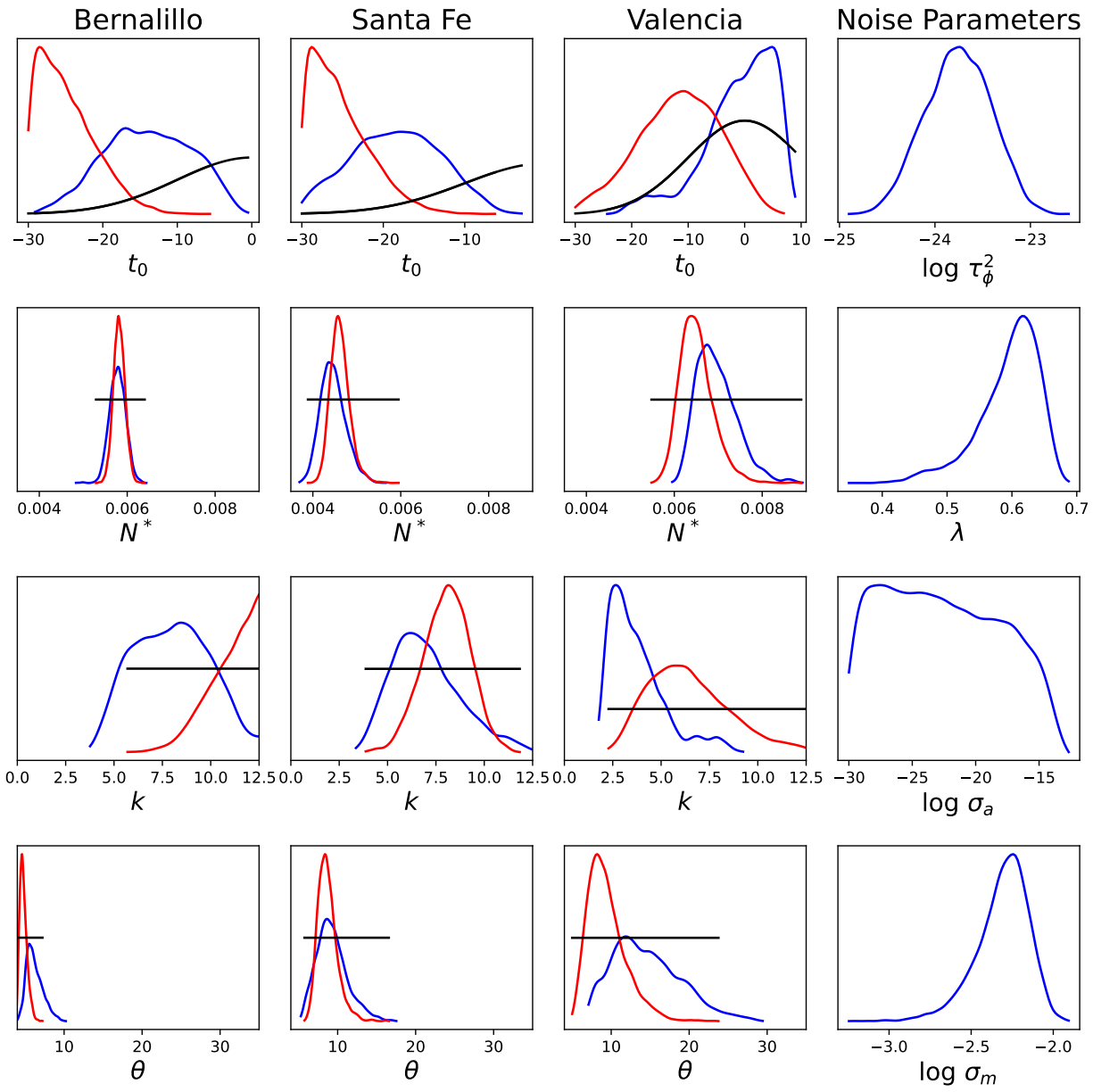

FIGURE 2-4. Marginalized posterior distributions for the parameters $\mathbf{m_r}$ for Bernalillo, Santa Fe, and Valencia. The black plot is the prior, the blue posterior distribution is from the joint estimation and the red one from the independent estimation. The last column has the "noise" parameters η. We see that the parameters of the spatial model can be estimated and the blue posteriors are usually narrower than the red ones, showing the constraining effect of the spatial correlations. N^* is the total size of the outbreak, normalized by the county's population.

In Fig. 2-5, we plot, on the left, the forecasts performed using samples drawn from the posterior distributions computed using joint (blue) and independent estimations of the counties. The dashed lines are the 95th and 5th percentiles of the forecasts and the thick line is the median prediction. There is hardly any difference in the two, showing that the joint inversion preserves the accuracy that one obtains when the counties' spread-rates are estimated independently, a much simpler and lower-dimensional inversion. Further, on most days beyond mid-August, the 95th and 5th percentile forecasts bound the observed case counts. On the right, we plot the daily infection-rate (or the spread-rate); again we see that the estimates from the joint and independent inversions are much the same.

2.6. Detecting the Fall 2020 wave

Finally, we explore the ability to detect the arrival of the Fall 2020 COVID-19 wave on September 15, 2020. We perform the estimation of spread-rate using MCMC for Bernalillo, Santa Fe and Valencia counties, and forecast 15 days beyond September 15, 2020. The 99th percentile is used as the "anomaly detection" boundary i.e., if the case count on a day exceeds the boundary, it is deemed an anomaly; three consecutive anomalies results in an "alarm" or a detection of a substantial departure of epidemiological dynamics from the model learned with past data. Similarly, with the RKI detector, we plot the anomaly boundary computed with the model, and use the same three-consecutive-anomalies as the rule for declaring an alarm.

In Fig. 2-6, we plot the results of the anomaly boundaries using the spread-rate i.e. the "expected" behavior of the COVID-19 outbreak in Bernalillo, Santa Fe and Valencia, as estimated from the joint posterior distribution of $\mathbf{m}_r$ and a conventional outbreak detector from RKI. We see that the Fall 2020 wave is detected by our detector within 3 days of September 15, for all three counties. For the RKI detector, the anomaly boundary is oscillatory and does not lead to an alarm even 15 days after the arrival of the Fall 2020 wave. The key reason is that the RKI detector, like all conventional ones, is purely statistical and does not exploit our prior knowledge of incubation periods or spatial correlations to smooth out the noise in the data.

In Fig. 2-7, we investigate the existence of false positives using our detector. We learn the spread-rate using data till August 15, 2020 and forecast ahead for 15 days. Since the Fall 2020 wave arrived on September 15, the detectors should not detect anything. We see that for Bernalillo and Santa Fe, the RKI detector detects a Fall 2020 wave within 3 days. Our spread-rate detector flags many anomalies, but none of them are consecutive and consequently no alarm is raised.

2.7. Summary of findings

In this chapter, we have formulated an inverse problem for the spread-rate in multiple areal units and solved it exactly using MCMC. Because of the lack of scalability of MCMC, we were limited to 3 adjacent counties. We find that joint and independent estimation of the spread-rate parameters $\mathbf{m_r}$ yield very similar forecasts. We fashioned the forecasts into a simple anomaly detector and proceeded to test for the arrival of the Fall 2020 wave in two scenarios - when the wave had arrived and before it had done so. Our detector behaved correctly. In contrast, the RKI detector failed in both scenarios, with a false negative and a false positive.

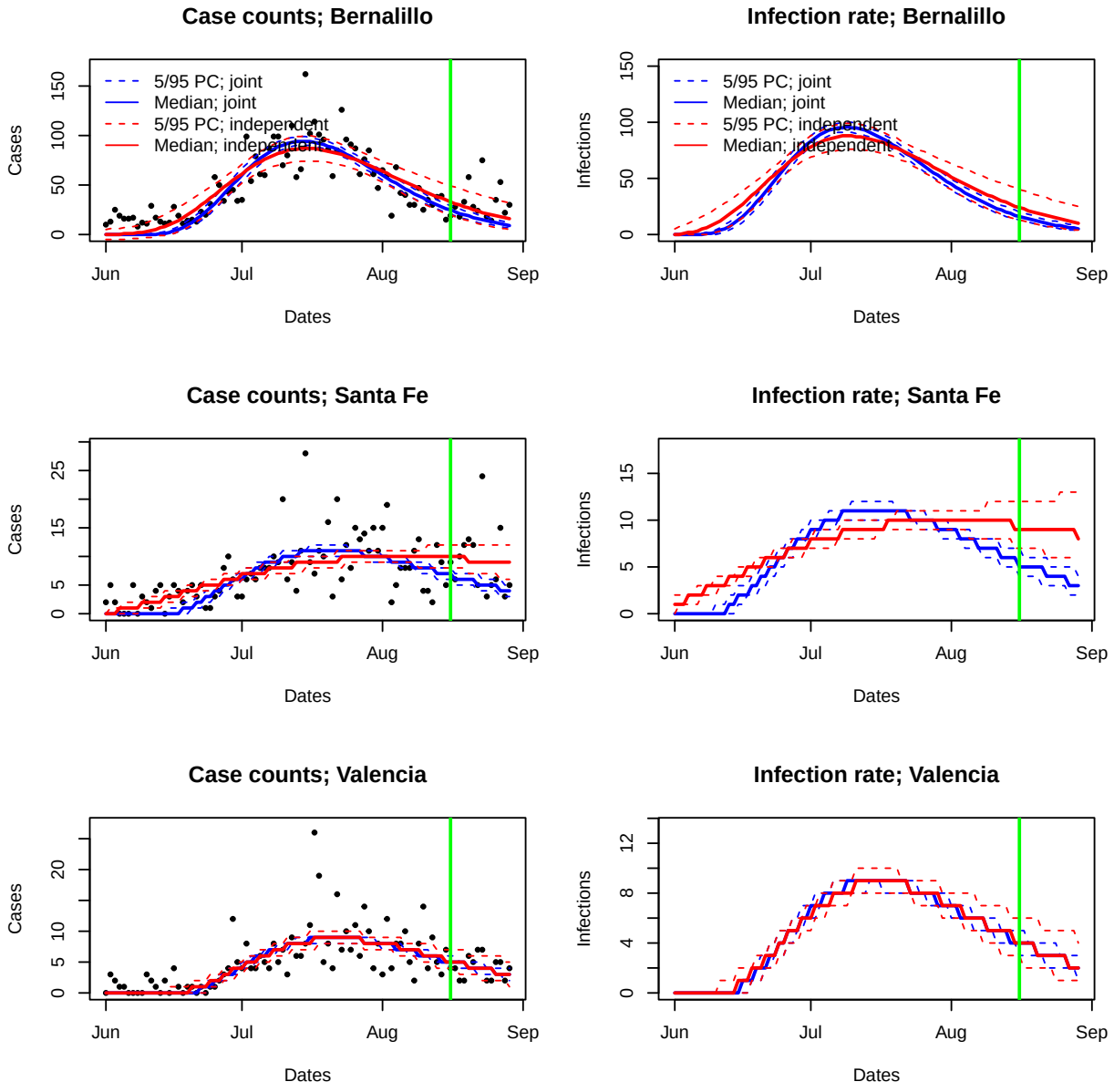

FIGURE 2-5. Left: Forecasts of the case counts, using data till August 15, 2020 (green line), for Bernalillo, Santa Fe and Valencia. Forecasts in blue are from the joint inversions whereas the ones in red are from inversions performed independently for each county. PC indicates "percentile"The reported case counts are plotted with symbols. Right: The corresponding spread rates.

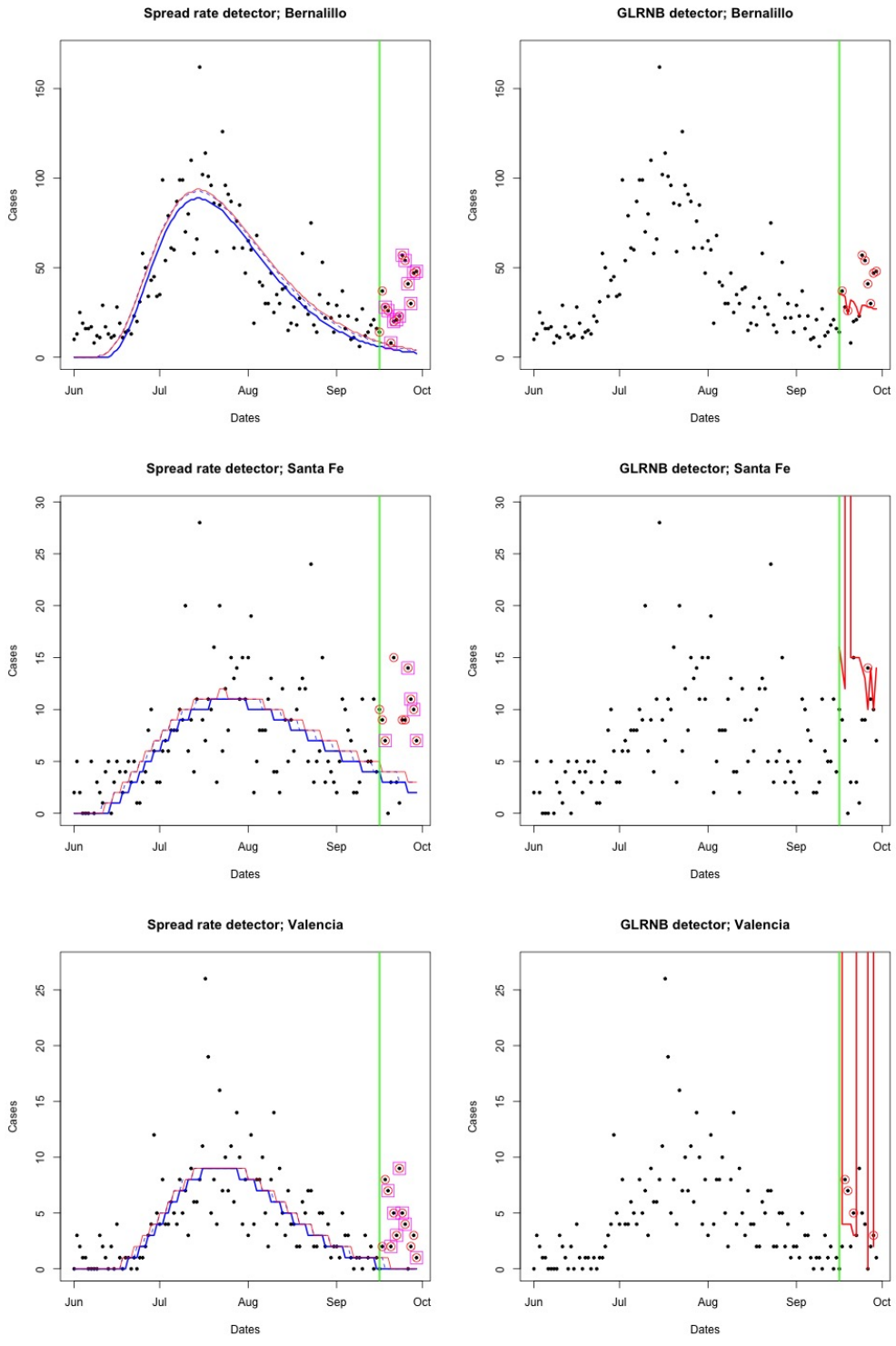

FIGURE 2-6. Anomaly boundaries (red line) for Bernalillo (top), Santa Fe (middle) and Valencia (bottom). On the left, the anomaly boundaries are computed using the spread-rate parameters $\mathbf{m}_r$ whereas on the right, they are computed using the RKI detector. Anomalies (i.e. case counts above the anomaly boundary) are circled in red and a magenta square indicates the third consecutive anomaly. The vertical green line denotes September 15. The GLRNB (Generalized Likelihood Ratio, Negative Binomial) detector is synonymous with the RKI detector.

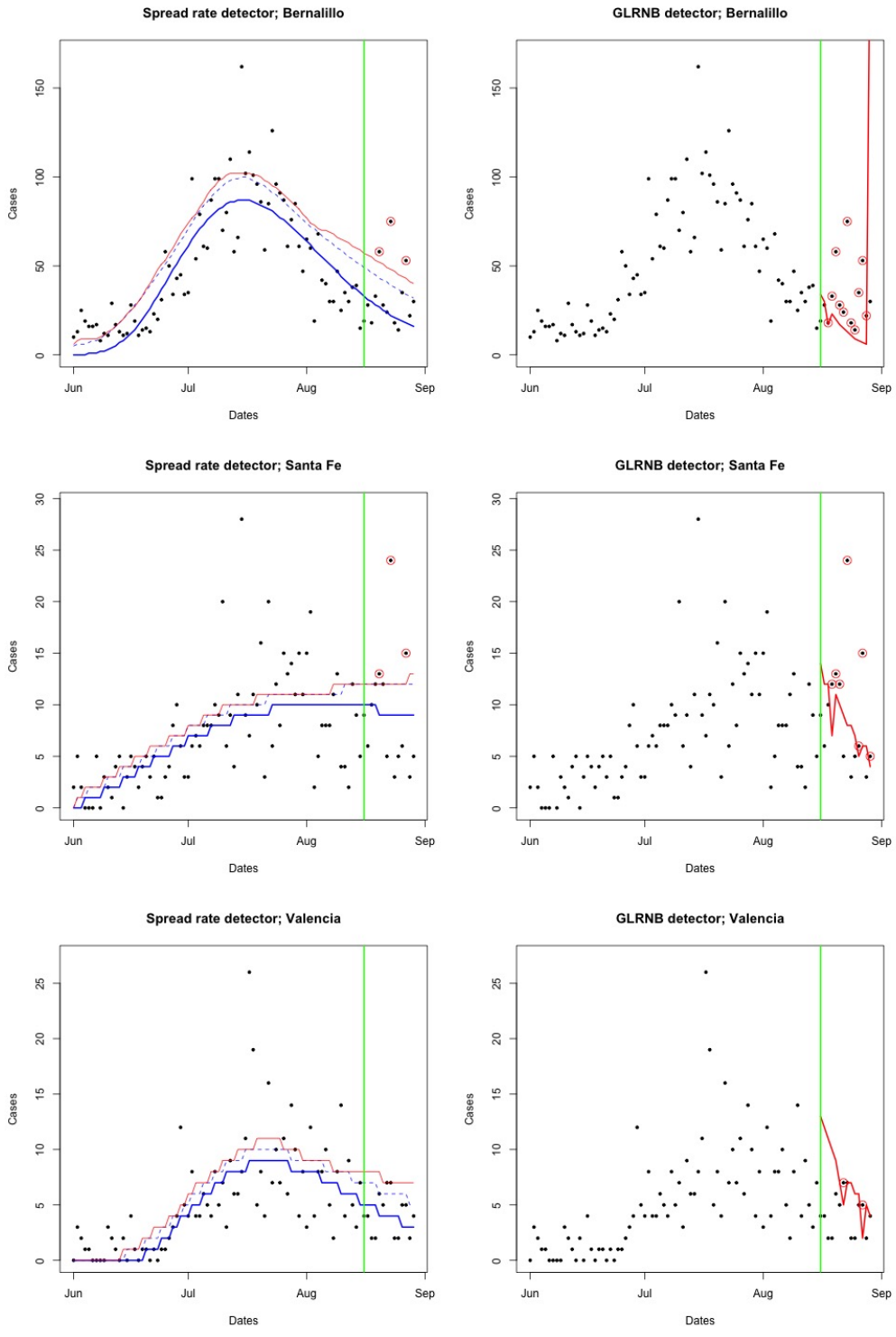

FIGURE 2-7. Anomaly boundaries (red line) for Bernalillo (top), Santa Fe (middle) and Valencia (bottom). On the left, the anomaly boundaries are computed using the spread-rate parameters $\mathbf{m}_r$ whereas on the right, they are computed using the RKI detector. Anomalies (i.e. case counts above the anomaly boundary) are circled in red and a magenta square indicates the third consecutive anomaly. The vertical green line denotes August 15. A Fall 2020 wave is detected in Bernalillo and Santa Fe by the RKI detector. The GLRNB (Generalized Likelihood Ratio, Negative Binomial) detector is synonymous with the RKI detector.

This page intentionally left blank

3. SCALING TO HIGHER DIMENSIONS

3.1. Introduction

In Chp. 2 we formulated an inverse problem for the spread-rate field and used it to estimate the spread-rate in 3 counties, due to lack of scalability of MCMC. The spread-rate was able to detect the arrival of the Fall 2020 COVID-19 wave. However, to fully exploit the power of spatial correlations to compensate for low-quality case-count data, and stress test the outbreak detector, we need to scale up to all 33 counties of NM. Many of the counties have far worse data than Bernalillo, Santa Fe and Valencia that we have investigated to date.

Scaling up to 33 counties, or 136 parameters, will be performed by mean-field Variational Inference. Variational inference (VI) [6] provides an alternative to sampling techniques where approximate inference is recast as seeking a member of a family of approximating densities which minimizes a discrepancy measure such as KL-divergence. Originally developed for probabilistic graphical models [22] where some degree of analytical tractability is maintained, it has more recently been extended to many-parameter models, such as those seen in deep learning, through gradient-based iterative schemes adapted to a probabilistic setting [8, 25, 17]. These techniques are often termed Stochastic Variational Inference (SVI) and can exploit the automatic differentiation available in large ML models. They offer significantly improved scalability over MCMC while potentially sacrificing some approximation quality depending on the set of approximating distributions used. VI has seen successful applications to a number high-dimensional inverse problems in areas including medical classification [47, 40] and segmentation [37, 32], computer vision and image processing [10], natural language processing [30, 20], and physics-based models [35, 51].

3.2. Formulation using VI

We start with the formulation of the inverse problem in Sec. 2.4, to obtain an approximation of the likelihood expression Eq. (2.9). We will compare the Bayesian posterior sampled with MCMC with posterior models obtained using mean-field VI (MFVI) which recasts approximate inference as an optimization problem. In particular, as the exact posterior is intractable, we consider a family of approximating densities $\mathscr{F} = \{q(\boldsymbol{\theta};\boldsymbol{\phi}) | \boldsymbol{\phi} \in \boldsymbol{\Phi} \subseteq \mathbb{R}^d\}$ and seek to find a density $q(\boldsymbol{\theta};\boldsymbol{\phi}^*)$ that minimizes the KL-divergence with respect to the posterior

$$\boldsymbol{\phi}^* = \text{argmin} \;\; D_{\text{KL}}\left(q(\boldsymbol{\theta};\boldsymbol{\phi}) \| p(\boldsymbol{\theta}|\mathscr{D})\right) \tag{3.1}$$

This can be re-expressed as minimizing the objective function $\mathscr{L}(\boldsymbol{\phi})$ based on the evidence lower bound (ELBO) [25]

$$\mathscr{L}(\boldsymbol{\phi}) = -\mathbb{H}[q(\boldsymbol{\theta};\boldsymbol{\phi})] - \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\phi})}[\log p(\mathscr{D}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})] \tag{3.2}$$

where the first term in Eq. (3.2) is the entropy of the surrogate posterior and the second, data-dependent term is an expectation with respect to the surrogate posterior that reflects both the expected data-fit and the

prior. Here we take $\mathscr{F}$ to be the set of mean-field Gaussian distributions, i.e.,

$$q(\boldsymbol{\theta};\boldsymbol{\phi}) = \prod_{i=1}^{d} q_i(\theta_i;\mu_i,\sigma_i) \tag{3.3}$$

where $q_i(\theta_i;\mu_i,\sigma_i) = \mathscr{N}(\theta_i;\mu_i,\sigma_i)$, $\boldsymbol{\phi} = (\boldsymbol{\mu},\boldsymbol{\sigma})$ and we arrive at an optimization problem over $2d$ parameters where d is the number of parameters defining the epidemiological model. To carry out the above minimization problem, we aim to use a gradient-based iterative scheme as the expectation in Eq. (3.2) cannot be evaluated explicitly due to the nonlinearity of the forward model. Furthermore, $\mathscr{L}(\boldsymbol{\phi})$ is potentially a non-convex objective. Note that the gradient and expectation operators do not commute, i.e., $\nabla_{\boldsymbol{\phi}}\mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\phi})}[\log p(\mathscr{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})] \neq \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\phi})}[\nabla_{\boldsymbol{\phi}} \log p(\mathscr{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})]$ so some care has to be taken to arrive at a Monte Carlo estimator for the gradient $\nabla_{\boldsymbol{\phi}}\mathscr{L}(\boldsymbol{\phi})$. Two widely used approaches are: (a) the Score function estimator, which forms the basis of black box MFVI and requires only evaluations of the log likelihood, and (b) the reparametrization approach which requires gradients of the log likelihood. The score function estimator typically displays much larger variance as seen in [26] where two orders of magnitude more samples were needed to arrive at the same variance as a reparametrization estimator. A similar trend was confirmed for the outbreak problem 3-3 suggesting that the reparametrization approach would lead to superior scalability. Reparametrization proceeds by expressing $\boldsymbol{\theta}$ as a differentiable transformation $\boldsymbol{\theta} = t(\boldsymbol{\varepsilon},\boldsymbol{\phi})$ of a $\boldsymbol{\phi}$-independent random variable $\boldsymbol{\varepsilon} \sim q(\boldsymbol{\varepsilon})$ such that $\boldsymbol{\theta}(\boldsymbol{\varepsilon},\boldsymbol{\phi}) \sim q(\boldsymbol{\theta},\boldsymbol{\phi})$. This allows the gradient to be expressed as

$$\nabla_{\boldsymbol{\phi}}\mathscr{L}(\boldsymbol{\phi}) = -\nabla_{\boldsymbol{\phi}}\mathbb{H}[q(\boldsymbol{\theta};\boldsymbol{\phi})] - \mathbb{E}_{q(\boldsymbol{\varepsilon})}[\nabla_{\boldsymbol{\phi}} \log p(\mathscr{D}|\boldsymbol{\theta}(\boldsymbol{\varepsilon},\boldsymbol{\phi})) + \nabla_{\boldsymbol{\phi}} \log p(\boldsymbol{\theta}(\boldsymbol{\varepsilon},\boldsymbol{\phi}))] \tag{3.4}$$

where gradients of the entropy term in Eq. (3.4) are available analytically for the Gaussian surrogate posterior and the second term can now be approximated with Monte Carlo given a method to compute the required gradients. For many machine learning models, automatic differentiation can be exploited to calculate the gradient of the log-likelihood with respect to parameters $\boldsymbol{\theta}$. Here, the objective function involves the log of the likelihood Eq. (2.9) where derivatives of matrix inverses and determinants with respect to parameters are required to compute the gradient. Gradients such as these are not available using most automatic differentiation libraries. Instead, matrix calculus and quadrature were used to compute the derivatives of the log likelihood with respect to model predictions $\mathbf{y}_i^{(p)}$ and to approximate the derivatives of the model predictions with respect to parameters, respectively. For details, see 3.6.

Note that some of the parameters comprising $\boldsymbol{\theta}$ are required to satisfy constraints for the noise and epidemiological models to be well-defined. For example, noise parameters τ_Φ, σ_a, and σ_m as well as model parameters N^r, k^r and θ^r for $r = 1,\ldots,R$ should be positive while λ_Φ should satisfy $0 \leq \lambda_\Phi \leq 1$. Sampling from the mean-field Gaussian Eq. (3.3) during the Monte Carlo estimation of the gradient Eq. (3.4) may result in violations of these constraints. To maintain the required properties without resorting to constrained optimization, we express a constrained parameter θ_i as an invertible, differentiable transformation $\theta_i = f_i(\hat{\theta}_i)$ of an unconstrained $\hat{\theta}_i$. Hence, the distribution governing θ_i is the push-forward density of $\mathscr{N}(\hat{\theta}_i;\mu_i,\sigma_i)$ through f_i, i.e., the components of the mean-field surrogate posterior Eq. (3.3) have modified probability densities

$$q_i(\theta_i;\mu_i,\sigma_i) = \mathscr{N}(\hat{\theta}_i;\mu_i,\sigma_i)|f_i'(\hat{\theta}_i)|^{-1}; \;\; 1 \leq i \leq d \tag{3.5}$$

where $\hat{\theta}_i = f_i^{-1}(\theta_i)$. This results in mean-field approximation where some of the factors are Gaussian and others non-Gaussian. Each factor is still defined by a μ_i and σ_i parameter.

3.2.1. Prior distribution

The COVID 19 case count data exhibits significant noise due to inaccurate case counting reported by hospitals. Furthermore, counties with small populations exhibit sparse data in the sense that not many positive daily case counts were reported. Hence, we expect the inverse problem to be ill-posed and require regularization in the form of a prior $p(\boldsymbol{\theta})$ over the parameters.

Because of push-forward formulation described by Eq. (3.5), a number of the parameters are already constrained by transformations $\theta_i = f_i(\hat{\theta}_i)$. In particular, the parameters N^r, k^r, θ^r for $r = 1, \ldots, R$ and each of the noise parameters comprising $\boldsymbol{\eta}$ are all constrained by transformations to take on values in some restricted interval. For example, λ_Φ is constrained to lie within $[0, 1-\varepsilon]$, for some $\varepsilon > 0$ so that Eq. (2.10) defines a valid covariance matrix, i.e., it remains symmetric, positive definite. The parameters t_0^r are the only unconstrained variables. Hence, we take Gaussian priors over t_0^r that incorporate diffuse assumptions about when it is reasonable for a wave to occur.

3.3. Results

In this section, we calibrate the coupled outbreak model across all 33 New Mexico counties using our formulation of MFVI. The convergence of the MVFI procedure for both the outbreak model and noise parameters is investigated and discussed. We then describe the final posterior approximation along with the Push-forward Posterior (PFP) distribution by pushing samples from the posterior through the forward model to examine the predictive uncertainty. Next, anomaly detection using the MFVI-calibrated outbreak model is carried out using COVID-19 spread-rates across all 33 counties in New Mexico using data from the summer of 2020 to detect the arrival of the Fall 2020 COVID-19 wave.

3.3.1. Independent vs. joint inversion of 3 counties

In Fig. 3-1, we plot the forecasts for Bernalillo, Santa Fe and Valencia counties, using spread-rates computed using MFVI. We see that the uncertainties are grossly underestimated - the "fan" of forecasts does not bound the measured case counts. Comparing the joint estimation (of all counties; bottom row of figures) with estimation done for each county independently, we see that there is hardly any difference i.e., the 138 parameter inversion does not lead to inaccuracies compared to the far simpler 6-dimensional inversion for a single county. In Fig. 3-2, we plot the spread-rates for the 3 counties, showing a peak that precedes the peak in case counts by about 5 days, the median of the incubation period distribution.

3.3.2. Joint inversion of all NM counties

The MFVI procedure is set up according to 3.2 where the stochastic gradient descent iteration is carried out using the Adaptive Moment Estimation (ADAM) algorithm [24]. MFVI is well-known to display mode-seeking behavior due properties of the KL-divergence. Hence, to improve the scalability and convergence of MFVI, we initialize the mean parameters for MFVI from a Maximum Likelihood Estimate (MLE) which is readily available using gradients of the log-likelihood. Calibration of the 33 county model using MFVI was then carried out where $N_s = 300$ samples were used in the Monte Carlo estimates of the ELBO gradient Eq. (3.4) at each iteration of ADAM. The convergence of MFVI is depicted in Figure 3-3 where the ELBO and the norm of its gradient are shown as a function of gradient descent iterations. The top of Figure 3-3 shows the ELBO and gradient for the 33-county calibration using the reparametrization formulation while the bottom provides a comparison to using black box MFVI for 1-county calibration of

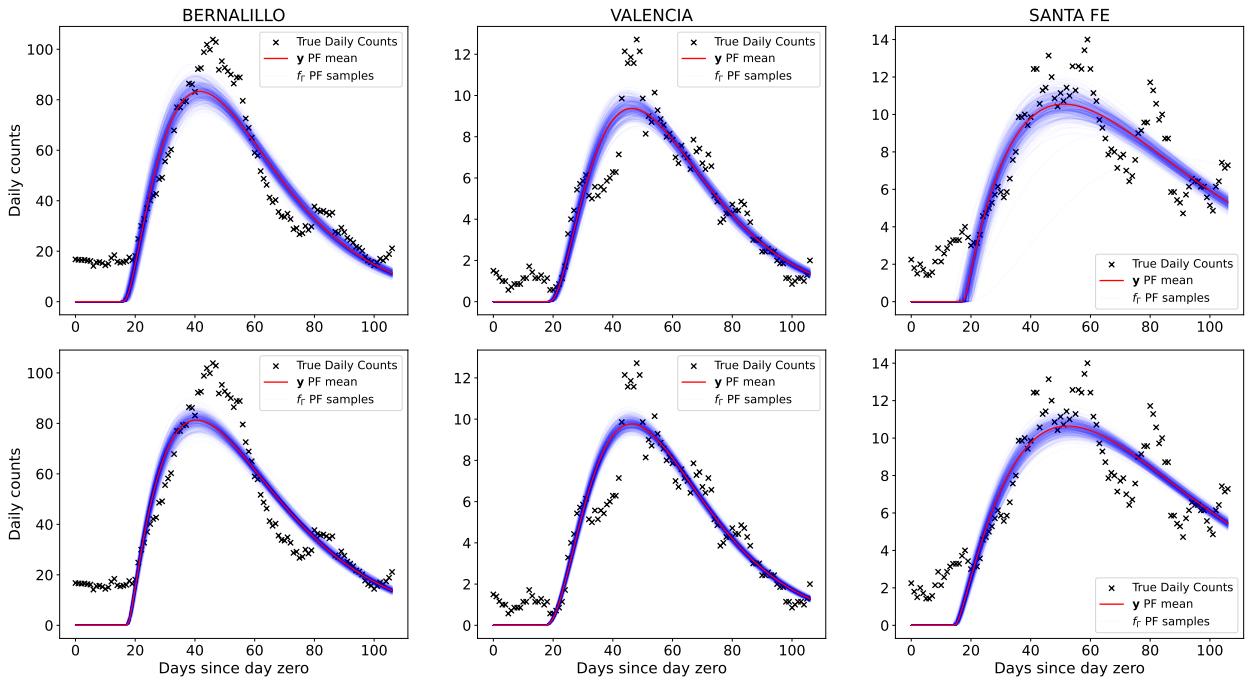

FIGURE 3-1. Comparison of the predictive distribution for the inversion of B, V, SF done independently (top) and jointly (bottom). Symbols denote the measured data, and the blue fan spans the 5^{th} and 95^{th} percentile bounds of the forecast. The red line is the median forecast.

Bernalillo. In both cases, $n = 300$ samples were used for the reparametrization and score function MC estimators of the gradient. Note that even for a single county, black box MFVI shows significantly higher variance in the gradient leading to poor convergence in comparison to the much larger 33-county problem calibrated with reparametrization. The convergence of the ELBO in the top row is also quite smooth suggesting that significant less samples could be used to obtain good estimates of the gradient with the reparametrization approach. Hence, it's clear that reparametrization is necessary to scale the calibration to the 33-county inversion despite the added complexity of complexity of obtaining gradients of the log likelihood.

In Figure 3-4, the convergence of MFVI is depicted in terms of the model predictions at the mean parameter values across the four counties Valencia, Santa Fe, Bernalillo, and Lea. The initial MLE prediction is shown in red with the final MFVI prediction in green. Intermediate values are shown in blue. Observe that while similar to the MLE, MFVI subtly expands the shape of the wave to better cover the tail of the outbreak. This is potentially an effect of the tendency of the KL-divergence to increase the overlap of the surrogate and true posterior distributions at some expense of the mean prediction fitting the data less accurately.

Exploiting spatial correlations across counties is a key aspect of the proposed inversion methodology. Hence, the convergence of the noise model parameters τ_Φ, λ_Φ, σ_a, and σ_m is of particular interest. In Figure 3-5, the convergence of the noise parameters during MLE calibration is shown as well as the subsequent evolution of the mean and standard deviations of the noise parameters during MFVI. The multiplicative noise σ_m convergences to roughly 0.1 during MLE, a relatively large value, while the additive noise σ_a tends to initially with the MLE and then significantly during MFVI. This suggests that the

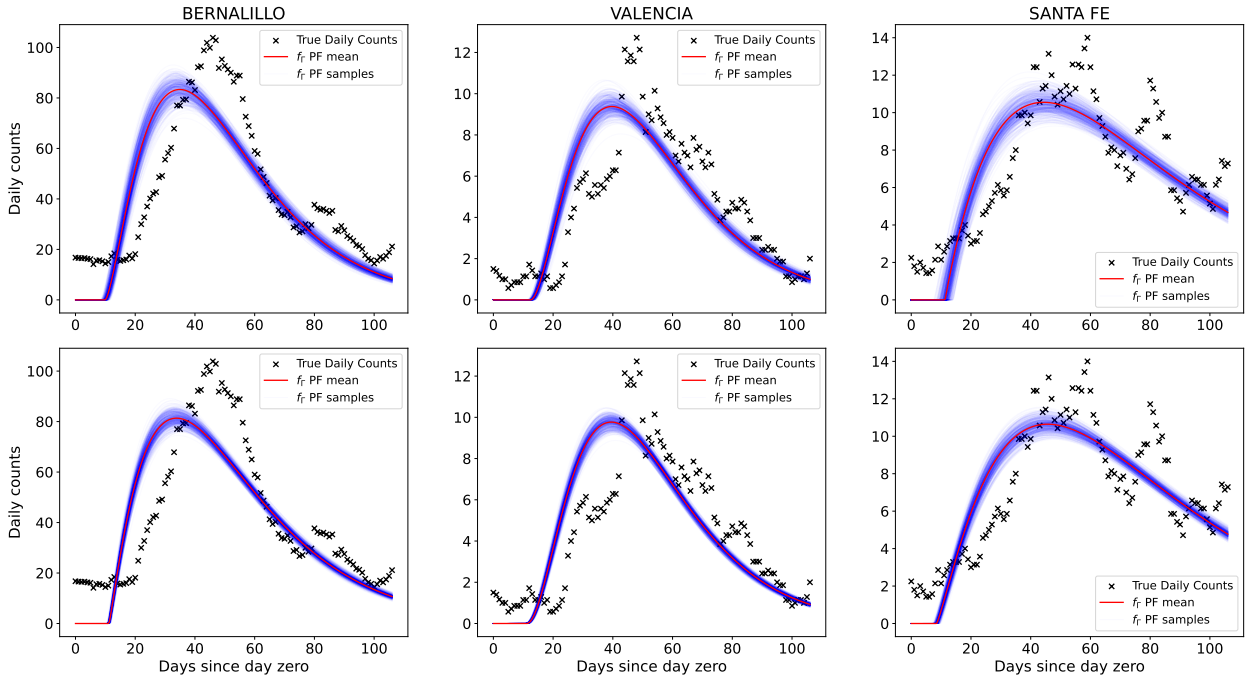

FIGURE 3-2. Comparison of the infection rate distribution for the inversion of B, V, SF done independently (top) and jointly (bottom). The red line is the median infection-rate and the blue fan its 5^{th} and 95^{th} percentile bounds. The measured case counts are also plotted to show how the spread-rate peaks about 5 days before the case counts do, commensurate with the incubation period distribution's median of 5 days.

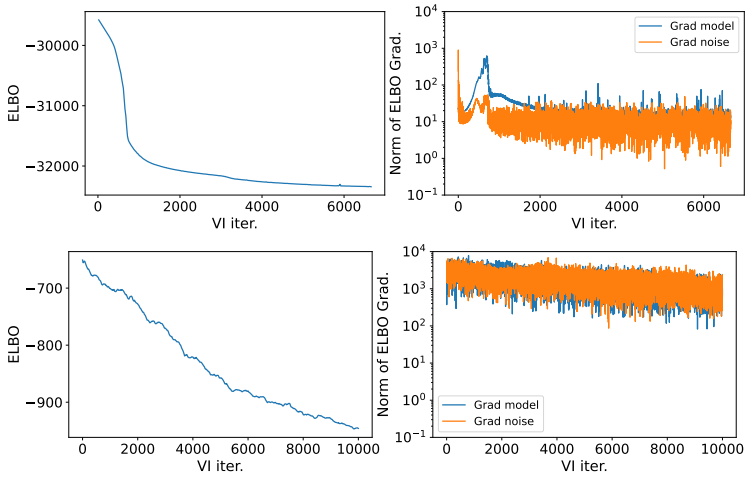

FIGURE 3-3. (Top) Convergence of the ELBO for the 33-county inversion along with the norm of the ELBO gradient as a function of gradient descent iterations for the reparametrized gradient formulation of MFVI. (Bottom) Convergence of the ELBO for a 1-county inverse problem along with the norm of the ELBO gradient for the black box formulation of MFVI. In both cases, $n = 300$ samples were used for the MC estimators of the gradient.

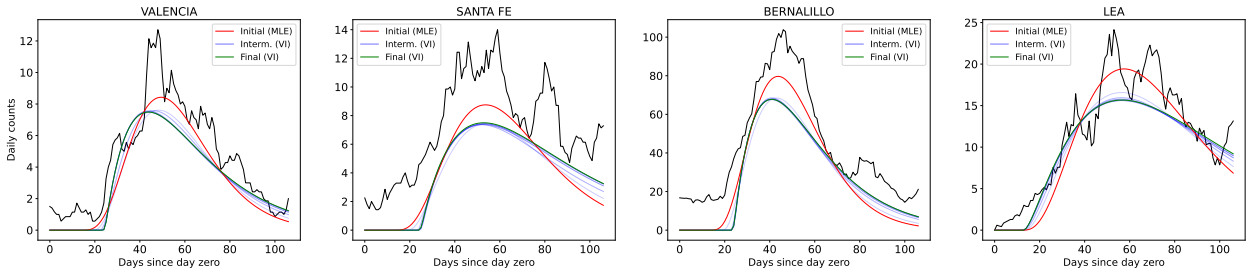

FIGURE 3-4. Visualization of the convergence of MFVI in terms of model predictions. The initial condition for MFVI is given by a MLE solution shown in red. Intermediate solutions are shown in blue along with the final solution in green.

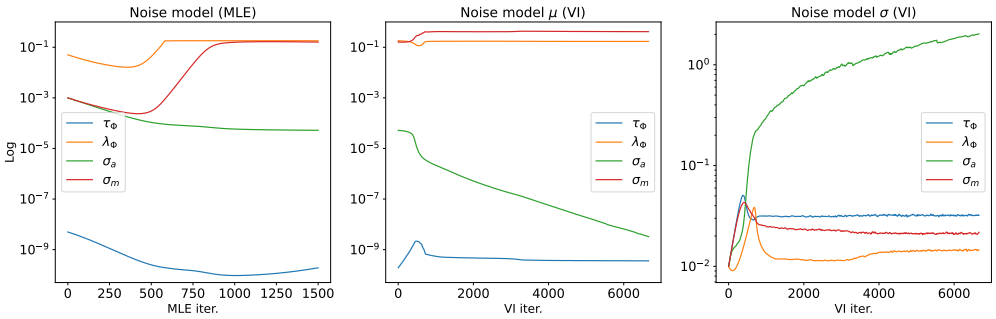

FIGURE 3-5. Convergence of MFVI for the noise model parameters τ_Φ, λ_Φ, σ_a, and σ_m. (Left) Converge of the MLE for the noise parameters. (Middle) Convergence of the mean parameters in MFVI. (Right) Convergence of the standard deviation parameters.

noise is more accurately described as count size dependent, as hypothesized, while a uniform additive noise over all counties is less able to capture the stochastic behavior of the data. For the GRF component of the noise model, recall that the parameter λ_Φ reflects the spatial correlation strength between adjacent counties. Observe that λ_Φ also converges to a relatively large value indicating that the spatial model is able to pick up on correlations. At first, it would seem that the small τ_Φ value would negate the effect of correlations by down-weighting the GRF term in 2.10. But the outbreak model was correlated on population-normalized daily case counts which are quite small in magnitude. Hence, the covariance structure ultimately exhibits non-trivial correlation strength across counties thus contributing significantly to the predictive model.

3.4. Anomaly detection using calibrated outbreak model

Comparing the forecasts of COVID-19 as obtained via MFVI (Fig. 3-1) and MCMC (Fig. 2-5), it is clear that the uncertainties are underestimated by MFVI. By the same token, we will have an alarm boundary that is too low, when computed using the MFVI solution. Consequently, using the MCMC-generated alarm boundaries for Bernalillo, Santa Fe and Valencia, we compute an “inflation factor" to adjust the MFVI alarm boundary for each day. The distribution of inflation factors is plotted in Fig. 3-7a. We choose the 75^{th} percentile of the inflation factors i.e., 1.2, to adjust the alarm boundaries obtained from a MFVI estimate of the spread-rate.

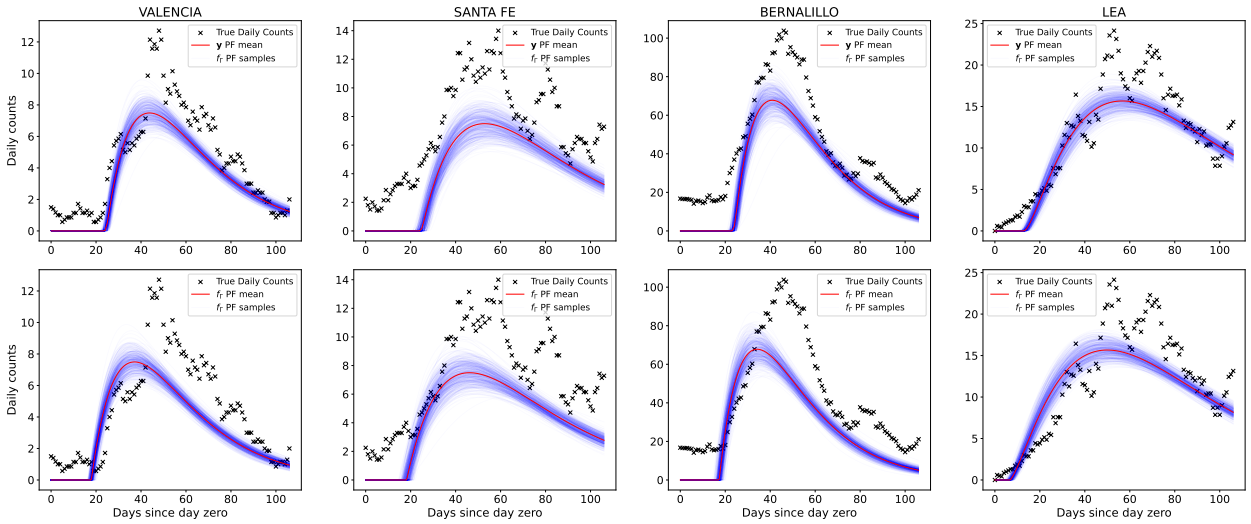

FIGURE 3-6. The push-forward of the MFVI posterior distribution through model predictions y (top) and the spread rate f_Γ (bottom). The mean is indicated by red the red curve and the variance is shown by plotting 500 samples from the push-forward distribution in blue.

In Fig. 3-7 (b), (c) and (d) we plot the arrival of the Fall 2020 wave in 3 counties. The spread-rate estimation is performed using data till September 15, 2020, followed by a 15-day-ahead forecast. The alarm boundary is considerably inflated compared to the 99^{th} percentile forecasts. The case counts are plotted and those within a magenta box are the third consecutive anomalous days (after September 15) which result in an alarm. We see that in all cases, we detect the arrival of the Fall 2020 wave. In Fig. 3-8, we repeat the detection using spread-rates computed from data available on August 15, a month before the arrival of the Fall 2020 wave. We see that the detector unfortunately records false positives in all cases.

Finally, using the case counts for each county and the “expected" number of cases provided by the alarm boundary, we perform a clustering using Kulldorff’s method [27]. The clusters are plotted in Fig. 3-9. We see the outbreak seed in Socorro on September 16, and by September 19, a cluster progressing along the Rio Grande Valley had formed.

3.5. Summary

We developed an approximate, but scalable, method for estimating the spread-rate field, a high-dimensional inversion involving 136 parameters; their posterior distributions were assumed to be independent Gaussians. Compared to MCMC, the uncertainties are under-estimated and we had to use an “inflation factor” to scale up the 99^{th} percentile of the forecast to serve as an alarm boundary. The method detects the Fall 2020 wave correctly within a week of September 15. Unfortunately, the method suffers from false positives and also detects a Fall 2020 wave a month before its arrival, on August 15th.

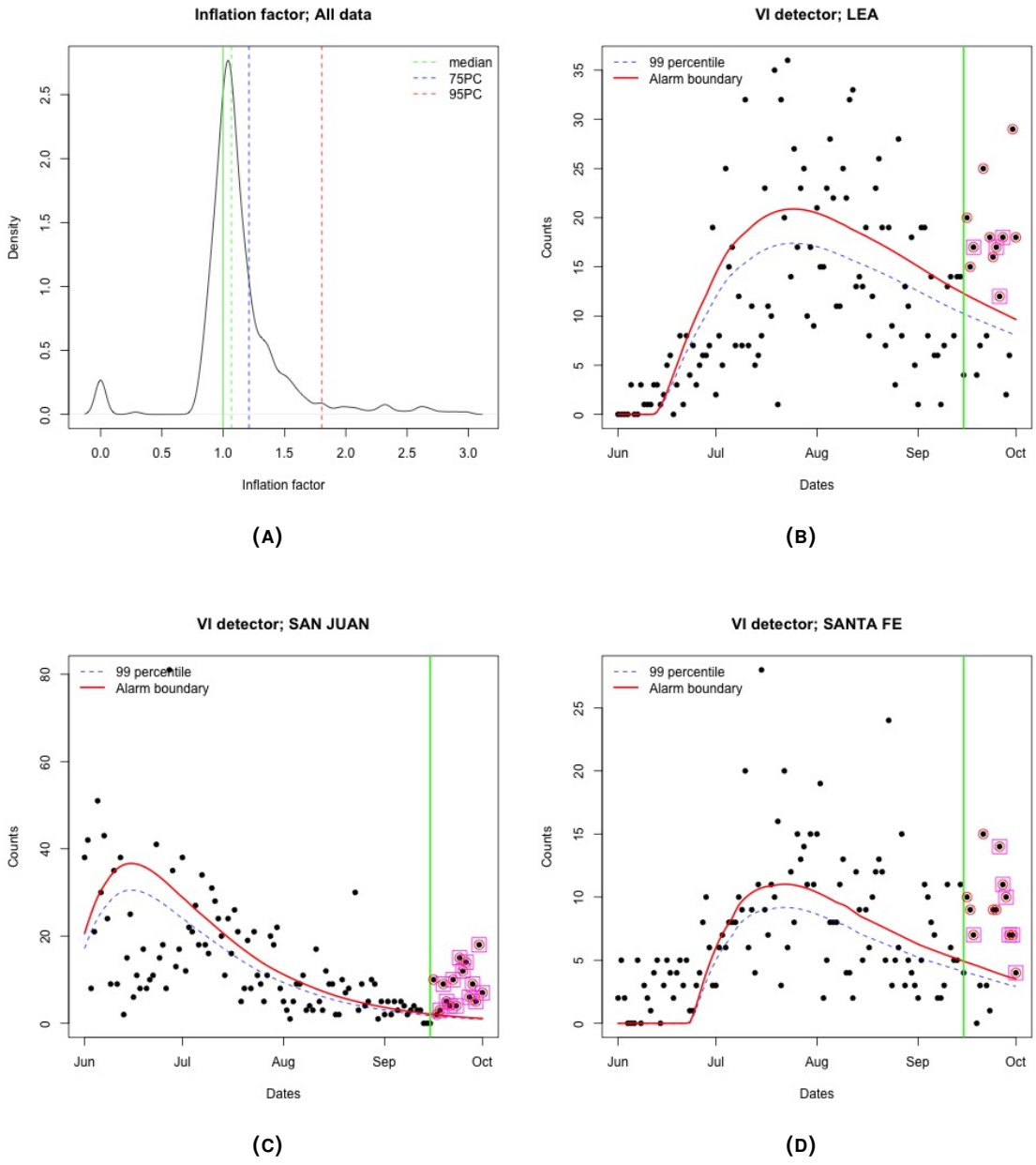

FIGURE 3-7. (a) Inflation factor distribution computed from Bernalillo, Santa Fe and Valencia forecasts. (b), (c) and (d) Forecasts and alarm boundary computed using a MFVI estimation of spread-rate in Lea, San Juan and Santa Fe counties. The magenta symbols denote an alarm. The September 15 arrival, denoted by a green line, was correctly recorded.

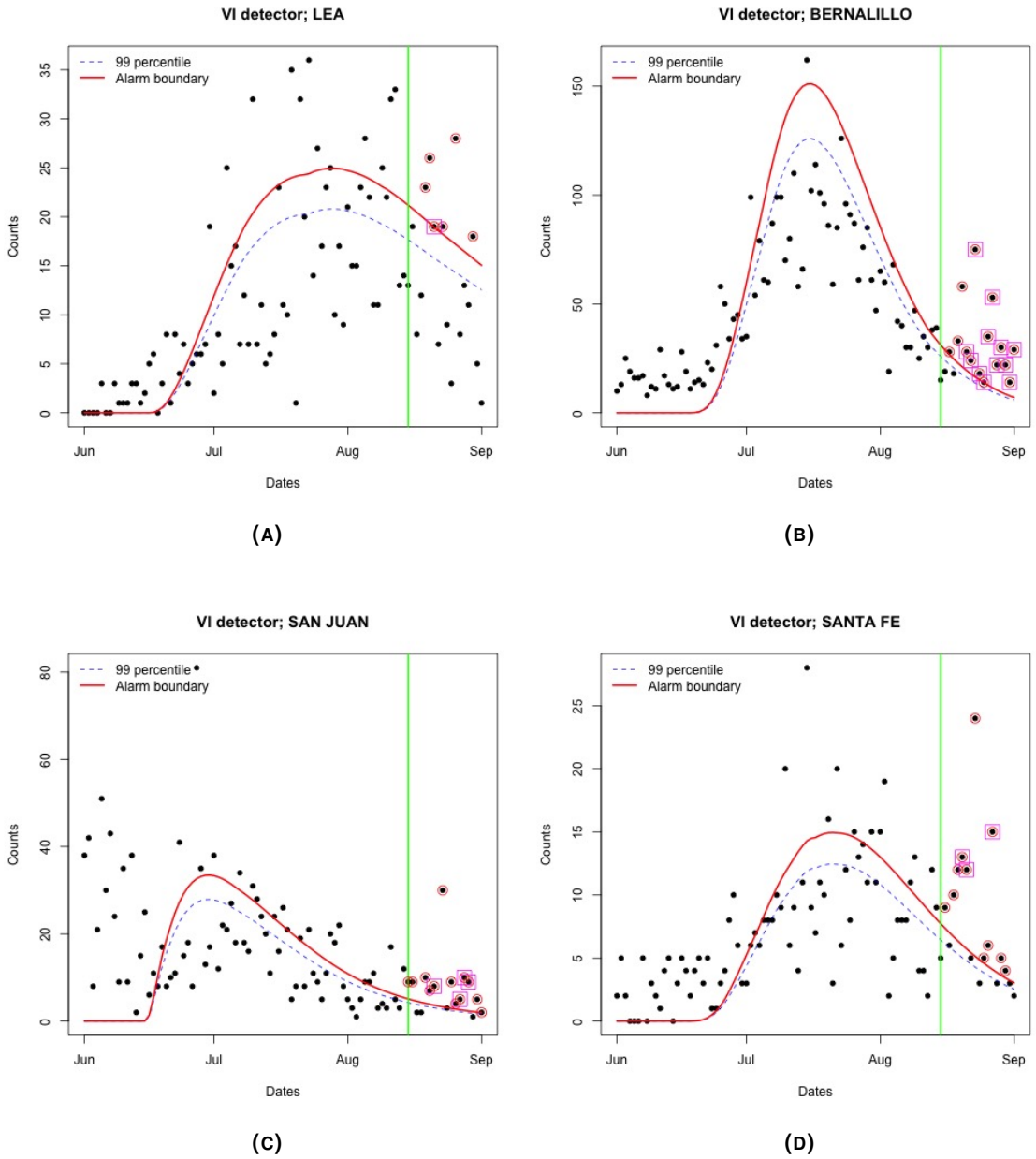

FIGURE 3-8. Forecasts and alarm boundary computed using a MFVI estimation of spread-rate in Lea, Bernalillo, San Juan and Santa Fe counties. The magenta symbols denote an alarm. Forecasts were started on August 15, 2020 (green vertical line), and the Fall 2020 wave was erroneously detected.

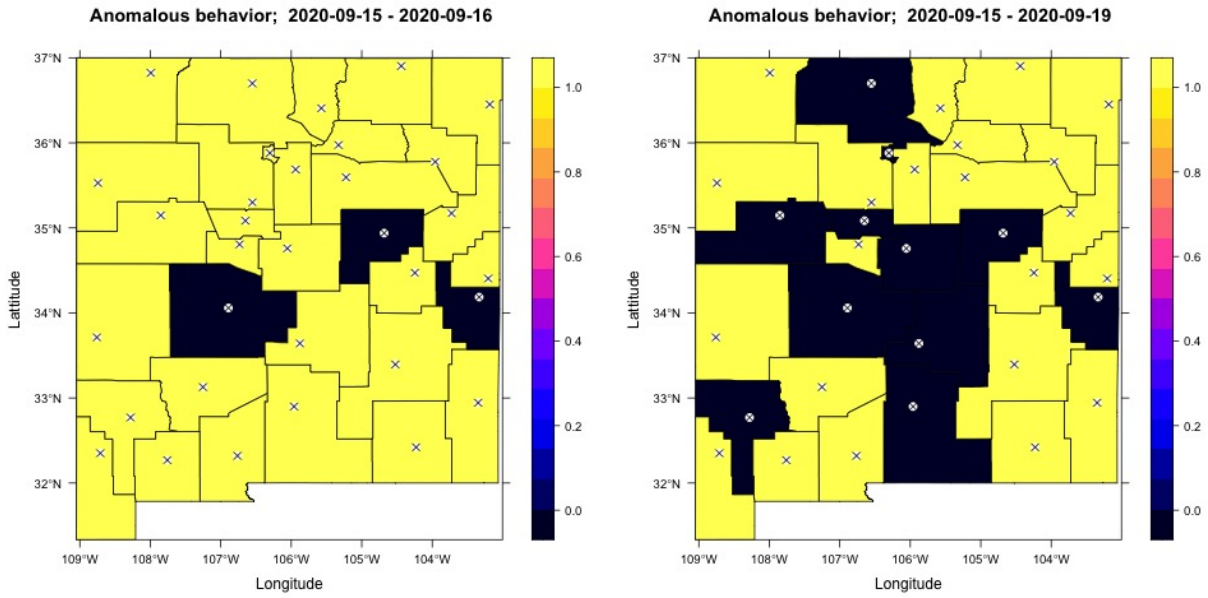

FIGURE 3-9. Anomalous *spatial* clusters starting on September 16 (left) and September 19 (right). The estimated spread-rate captures the spread of the Fall 2020 wave progressing down the Rio Grande Valley.

3.6. Appendix - Variational Inference

3.6.1. Reparametrization gradients of the ELBO

The likelihood and log likelihood are given by

$$p(\mathscr{D}|\boldsymbol{\theta}) = \prod_{i=1}^{N_d} 2\pi^{-N_r/2} \det(\blacksquare_i)^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} (\mathbf{y}_i^{(o)} - \mathbf{y}_i)\right) \tag{3.6}$$

$$l(\boldsymbol{\theta}) = -\frac{N_d N_r 2\pi}{2} - \frac{1}{2}\sum_{i=1}^{N_d} \log\det(\blacksquare_i) + (\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} (\mathbf{y}_i^{(o)} - \mathbf{y}_i) \tag{3.7}$$

Using the reparametrization trick, we can write the ELBO Eq. (3.2) and its gradient in the form

$$\mathscr{L}(\boldsymbol{\phi}) = -\mathbb{H}[q(\boldsymbol{\theta};\boldsymbol{\phi})] - \mathbb{E}_{q(\boldsymbol{\varepsilon})}[\log p(\mathscr{D}|\boldsymbol{\theta}(\boldsymbol{\varepsilon},\boldsymbol{\phi})) + \log p(\boldsymbol{\theta}(\boldsymbol{\varepsilon},\boldsymbol{\phi}))] \tag{3.8}$$

$$\nabla_{\boldsymbol{\phi}}\mathscr{L}(\boldsymbol{\phi}) = -\nabla_{\boldsymbol{\phi}}\mathbb{H}[q(\boldsymbol{\theta};\boldsymbol{\phi})] - \mathbb{E}_{q(\boldsymbol{\varepsilon})}[\nabla_{\boldsymbol{\phi}} \log p(\mathscr{D}|\boldsymbol{\theta}(\boldsymbol{\varepsilon},\boldsymbol{\phi})) + \nabla_{\boldsymbol{\phi}} \log p(\boldsymbol{\theta}(\boldsymbol{\varepsilon},\boldsymbol{\phi}))] \tag{3.9}$$

where $\boldsymbol{\phi} = (\boldsymbol{\mu}, \boldsymbol{\rho})$, $\boldsymbol{\theta} = \boldsymbol{\mu} + \boldsymbol{\sigma}(\boldsymbol{\rho}) \odot \boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon} \sim \mathscr{N}(\mathbf{0}, \mathbf{I})$. Here, σ is a positive transformation of the unconstrained variable $\boldsymbol{\rho}$ to ensure the variance is constrained to be positive. A Monte Carlo estimator of the gradient can then be written as

$$\nabla_{\boldsymbol{\phi}}\mathscr{L}(\boldsymbol{\phi}) \approx -\nabla_{\boldsymbol{\phi}}\mathbb{H}[q(\boldsymbol{\theta};\boldsymbol{\phi})] - \frac{1}{N_s}\sum_{i=1}^{N_s} \nabla_{\boldsymbol{\phi}} \log p(\mathscr{D}|\boldsymbol{\theta}(\boldsymbol{\varepsilon}_i,\boldsymbol{\phi})) + \nabla_{\boldsymbol{\phi}} \log p(\boldsymbol{\theta}(\boldsymbol{\varepsilon}_i,\boldsymbol{\phi})) \tag{3.10}$$

$$= -\nabla_{\boldsymbol{\phi}}\mathbb{H}[q(\boldsymbol{\theta};\boldsymbol{\phi})] - \frac{1}{N_s}\sum_{i=1}^{N_s} (\nabla_{\boldsymbol{\theta}} \log p(\mathscr{D}|\boldsymbol{\theta}(\boldsymbol{\varepsilon}_i,\boldsymbol{\phi})) + \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}(\boldsymbol{\varepsilon}_i,\boldsymbol{\phi}))) \odot \nabla_{\boldsymbol{\phi}}\boldsymbol{\theta}(\boldsymbol{\varepsilon}_i,\boldsymbol{\phi}) \tag{3.11}$$

where the last line is given by the chain rule and the fact that $\boldsymbol{\theta}$ is defined by an element-wise transformation of $\boldsymbol{\phi}$. Observe that

$$\nabla_{\boldsymbol{\mu}} \boldsymbol{\theta} = \mathbf{1} \tag{3.12}$$

$$\nabla_{\boldsymbol{\rho}} \boldsymbol{\theta} = \nabla_{\boldsymbol{\rho}} \boldsymbol{\sigma}(\boldsymbol{\rho}) \odot \boldsymbol{\varepsilon}_i \tag{3.13}$$

$$\nabla_{\boldsymbol{\mu}} \mathbb{H}[q(\boldsymbol{\theta};\boldsymbol{\phi})] = \mathbf{0} \tag{3.14}$$

$$\nabla_{\boldsymbol{\rho}} \mathbb{H}[q(\boldsymbol{\theta};\boldsymbol{\phi})] = \frac{1}{\boldsymbol{\sigma}(\boldsymbol{\rho})} \odot \nabla_{\boldsymbol{\rho}} \boldsymbol{\sigma}(\boldsymbol{\rho}) \tag{3.15}$$

so that it remains to compute the gradients of the log-likelihood and prior.

3.6.2. Gradients of the Log Likelihood

As the log likelihood factors independently across the data $i = 1, \ldots, N_d$, it suffices to compute the gradients $\nabla_{\boldsymbol{\theta}} \log\det \blacksquare_i$ and $\nabla_{\boldsymbol{\theta}} (\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} (\mathbf{y}_i^{(o)} - \mathbf{y}_i)$ for a particular day i. The differentials of the these two terms are computed using matrix calculus [39] as

$$\partial \log\det \blacksquare_i = \mathrm{Tr}(\blacksquare_i^{-1} \partial \blacksquare_i) \tag{3.16}$$

$$\partial (\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} (\mathbf{y}_i^{(o)} - \mathbf{y}_i) = -(\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} (\partial \blacksquare_i) \blacksquare_i^{-1} (\mathbf{y}_i^{(o)} - \mathbf{y}_i) - 2(\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} \partial \mathbf{y}_i \tag{3.17}$$

where the differential of $\blacksquare_i$ is

$$\partial \blacksquare_i = (\partial \tau_\phi)[\mathbf{I} - \lambda_\Phi \mathbf{W}]^{-1} - (\partial \lambda_\Phi) \tau \Phi [\mathbf{I} - \lambda_\Phi \mathbf{W}]^{-1} \mathbf{W} [\mathbf{I} - \lambda_\Phi \mathbf{W}]^{-1} \tag{3.18}$$

$$+ 2\mathrm{diag}(\sigma_a + \sigma_m \mathbf{y}_i)[(\partial \sigma_a)\mathbf{I} + (\partial \sigma_m)\mathrm{diag}(\mathbf{y}_i) + \sigma_m \mathrm{diag}(\partial \mathbf{y}_i)] \tag{3.19}$$

We have the following derivatives

- **Case:** $\hat{\mathbf{m}}_r^T = (\hat{t}_0^r, \hat{N}^r, \hat{k}^r, \hat{\theta}^r)$ are the unconstrained model variables for region r and $\hat{\theta}_r$ is a particular variable.

$$\partial_{\hat{\theta}_r} \mathbf{y}_i = (0, \ldots, \partial_{\theta_r} y_r(i; t_0^r, N_r, k^r, \theta^r) \theta_i'(\hat{\theta}_i), \ldots, 0) \tag{3.20}$$

$$\nabla_{\hat{\mathbf{m}}_r} \log\det \blacksquare_i = 2\sigma_m [\blacksquare_i^{-1}]_{ii} \nabla_{\hat{\mathbf{m}}_r} y_r(i; t_0^r, N^r, k^r, \theta^r) \tag{3.21}$$

$$\partial_{\hat{\theta}_r} (\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} (\mathbf{y}_i^{(o)} - \mathbf{y}_i) = -2\sigma_m (\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} \mathrm{diag}(\sigma_a + \sigma_m \mathbf{y}_i) \mathrm{diag}(\partial_{\hat{\theta}_r} \mathbf{y}_i) \blacksquare_i^{-1} (\mathbf{y}_i^{(o)} - \mathbf{y}_i) \tag{3.22}$$

$$- 2(\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} \partial_{\hat{\theta}_r} \mathbf{y}_i \tag{3.23}$$

- **Case:** $\hat{\theta}_r = \hat{\tau}_\Phi$

$$\partial_{\hat{\theta}_r} \mathbf{y}_i = \mathbf{0} \tag{3.24}$$

$$\partial_{\hat{\theta}_r} \log\det \blacksquare_i = \tau_\Phi'(\hat{\tau}_\Phi) \mathrm{Tr}(\blacksquare_i^{-1} [\mathbf{I} - \lambda_\Phi \mathbf{W}]^{-1}) \tag{3.25}$$

$$\partial_{\hat{\theta}_r} (\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} (\mathbf{y}_i^{(o)} - \mathbf{y}_i) = \tau_\Phi'(\hat{\tau}_\Phi) (\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} [\mathbf{I} - \lambda_\Phi \mathbf{W}]^{-1} \blacksquare_i^{-1} (\mathbf{y}_i^{(o)} - \mathbf{y}_i) \tag{3.26}$$

- **Case:** $\hat{\theta}_r = \hat{\lambda}_\Phi$

$$\partial_{\hat{\theta}_r} \mathbf{y}_i = \mathbf{0} \tag{3.27}$$

$$\partial_{\hat{\theta}_r} \log\det \blacksquare_i = \lambda_\Phi'(\hat{\lambda}_\Phi) \tau_\Phi \mathrm{Tr}(\blacksquare_i^{-1} [\mathbf{I} - \lambda_\Phi \mathbf{W}]^{-1} \mathbf{W} [\mathbf{I} - \lambda_\Phi \mathbf{W}]^{-1}) \tag{3.28}$$

$$\partial_{\hat{\theta}_r} (\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} (\mathbf{y}_i^{(o)} - \mathbf{y}_i) = \lambda_\Phi'(\hat{\lambda}_\Phi) \tau_\Phi (\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} [\mathbf{I} - \lambda_\Phi \mathbf{W}]^{-1} \mathbf{W} [\mathbf{I} - \lambda_\Phi \mathbf{W}]^{-1} \blacksquare_i^{-1} (\mathbf{y}_i^{(o)} - \mathbf{y}_i) \tag{3.29}$$

- **Case:** $\hat{\theta}_r = \hat{\sigma}_a$

$$\partial_{\hat{\theta}_r} \mathbf{y}_i = \mathbf{0} \tag{3.30}$$

$$\partial_{\hat{\theta}_r} \log\det \blacksquare_i = 2\sigma_a'(\hat{\sigma}_a)\mathrm{Tr}(\blacksquare_i^{-1}\mathrm{diag}(\sigma_a + \sigma_m \mathbf{y}_i)) \tag{3.31}$$

$$\partial_{\hat{\theta}_r} (\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} (\mathbf{y}_i^{(o)} - \mathbf{y}_i) = -2\sigma_a'(\hat{\sigma}_a)(\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} \mathrm{diag}(\sigma_a + \sigma_m \mathbf{y}_i) \blacksquare_i^{-1} (\mathbf{y}_i^{(o)} - \mathbf{y}_i) \tag{3.32}$$

- **Case:** $\hat{\theta}_r = \hat{\sigma}_m$

$$\partial_{\hat{\theta}_r} \mathbf{y}_i = \mathbf{0} \tag{3.33}$$

$$\partial_{\hat{\theta}_r} \log\det \blacksquare_i = 2\sigma_m'(\hat{\sigma}_m)\mathrm{Tr}(\blacksquare_i^{-1}\mathrm{diag}((\sigma_a + \sigma_m \mathbf{y}_i) \odot \mathbf{y}_i)) \tag{3.34}$$

$$\partial_{\hat{\theta}_r} (\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} (\mathbf{y}_i^{(o)} - \mathbf{y}_i) = -2\sigma_m'(\hat{\sigma}_m)(\mathbf{y}_i^{(o)} - \mathbf{y}_i)^T \blacksquare_i^{-1} \mathrm{diag}((\sigma_a + \sigma_m \mathbf{y}_i) \odot \mathbf{y}_i) \blacksquare_i^{-1} (\mathbf{y}_i^{(o)} - \mathbf{y}_i) \tag{3.35}$$

3.6.3. Approximation of model predictions and gradients via quadrature

The model predictions $\mathbf{y}_i$, given by Eq. (2.7), involve a convolution integral that cannot be expressed in closed form. Hence, we approximate the predictions by integral quadrature

$$y_r(i; t_0^r, N^r, k^r, \theta^r) = \int_{t_0^r}^{t_i} N_r f_{inf}(\tau - t_0^r; k^r, \theta^r)(F_{inc}(t_i - \tau; \mu, \sigma) - (F_{inc}(t_{i-1} - \tau; \mu, \sigma)) d\tau \tag{3.36}$$

$$\approx \sum_{j=1}^{n} N_r w_j f_{inf}(\tau_j - t_0^r; k^r, \theta^r)(F_{inc}(t_i - \tau_j; \mu, \sigma) - (F_{inc}(t_{i-1} - \tau_j; \mu, \sigma)) \tag{3.37}$$

where w_j, τ_j are quadrature weights and points given by a method such as Gaussian quadrature. As the function $(F_{inc}(t_i - \tau; \mu, \sigma) - (F_{inc}(t_{i-1} - \tau; \mu, \sigma))$ does not depend on parameters $\boldsymbol{\theta}$, we can write it as $\tilde{F}_{inc}(\tau)$ for simplicity of notation. By the Leibniz integral rule, we can write the derivatives of the model predictions as

$$\partial_{\hat{\theta}_r} \mathbf{y}_i = \int_{t_0^r}^{t_i} [\partial_{\hat{\theta}_r} N_r f_{inf}(\tau - t_0^r; k^r, \theta^r)] \tilde{F}_{inc}(\tau) d\tau \qquad \text{if } \hat{\theta}_r \neq t_0^r \tag{3.38}$$

$$\partial_{\hat{\theta}_r} \mathbf{y}_i = \int_{t_0^r}^{t_i} [\partial_{\hat{\theta}_r} N_r f_{inf}(\tau - t_0^r; k^r, \theta^r)] \tilde{F}_{inc}(\tau) d\tau - f_{inf}(0; k^r, \theta^r) \qquad \text{if } \hat{\theta}_r = t_0^r \tag{3.39}$$

This requires that the functions $N_r f_{inf}(\tau - t_0^r; k^r, \theta^r)]\tilde{F}_{inc}(\tau)$ and $\partial_{\hat{\theta}_r} N_r f_{inf}(\tau - t_0^r; k^r, \theta^r)]\tilde{F}_{inc}(\tau)$ are continuous in τ and $\hat{\theta}_r$ in a region of the τ-$\hat{\theta}_r$ plane including $t_0 \leq \tau \leq t_i$, a condition that's easily met by ensuring certain constraints are satisfied by the parameters $(t_0^r, N^r, k^r, \theta^r)$ for $r = 1, \ldots, N_r$. These constraints are enforced via the variable transformations in Eq. (3.5). Hence, we can approximate Eq. (3.38) and Eq. (3.39) via quadrature in the form

$$\partial_{\hat{\theta}_r} \mathbf{y}_i \approx \sum_{j=1}^{n} w_j [\partial_{\hat{\theta}_r} N_r f_{inf}(\tau_j - t_0^r; k^r, \theta^r)] \tilde{F}_{inc}(\tau_j) \qquad \text{if } \hat{\theta}_r \neq t_0^r \tag{3.40}$$

$$\partial_{\hat{\theta}_r} \mathbf{y}_i \approx \sum_{j=1}^{n} w_j [\partial_{\hat{\theta}_r} N_r f_{inf}(\tau_j - t_0^r; k^r, \theta^r)] \tilde{F}_{inc}(\tau_j) - f_{inf}(0; k^r, \theta^r) \qquad \text{if } \hat{\theta}_r = t_0^r \tag{3.41}$$

Hence, in this implementation of MFVI, gradients of the ELBO have a pseudo-analytic form where "outer" gradients of the log likelihood with respect to model predictions are exact and "inner" gradients of the model predictions with respect to parameters are approximated via quadrature. This allows for accurate gradient approximations that can be caculated efficiently leading to a scalable MFVI algorithm that can be applied to the high-dimensional inverse problem for the outbreak model.

4. CALIBRATION OF AGENT-BASED DISEASE MODELS

4.1. Introduction

Accurate predictive modeling of disease spread is critical for understanding the potential impacts of an outbreak and implementing effective interventions. For example, models such as Covasim [23], OpenABM-Covid19 [18], CityCovid [38], and many others [34] have been used to inform policy decisions and intervention strategies in the recent COVID-19 pandemic. Considering the significant impact that these models can have on policy and on public perception of risk, it is important that they provide realistic predictions and uncertainty estimations.

Inaccuracies in model results may come from simplifications and assumptions in model structure or from errors in calibration of parameters. In this paper, we will focus on the latter by testing parameter calibration methods on synthetic data generated by the same stochastic model we wish to calibrate. Since we know the parameters used to generate the synthetic data, we can directly compare the true and estimated parameter values, and since we are able to generate an arbitrary number of synthetic data sets, we can evaluate uncertainty predictions over repeated calibration tests.

We use an agent-based model (ABM), which is a commonly-used epidemiological model structure where individuals in a population are represented as unique agents, each with a set of characteristics and behavioral rules [21]. Compartmental ABMs categorize agents into compartments for disease status tracking — for example, we use compartments of susceptible (S), exposed (E), infected (I), and recovered (R), which together form an SEIR model. ABMs are generally stochastic via randomness in agent behavior, position, and/or infection mechanisms, and are well-suited to capturing the effects of heterogeneous population spread. This is in contrast to compartmental ordinary differential equation models, another widely-used model type which are generally deterministic and assume homogeneous mixing with a population [46].

Calibration methods for epidemiological models vary widely. First, they often aim to achieve different goals — some wish to establish parameter values and their uncertainty (whether by constructing a full posterior, calculating confidence intervals, or through some other method) [29, 36, 31, 1, 38, 44, 48], where some aim to find the best-fit parameters with no uncertainty predictions [48, 18, 1, 50]. Some papers aim to quantify uncertainty for only a subset of their parameters [1, 48]. Drake et al. creates a "plausible parameter set" of parameters for which real-world cumulative case data was within the range of repeated model simulation results, from which an ensemble can be formed [12]. Ozik et al. finds the Pareto set of best parameter combinations (in addition to Bayesian posterior construction) [38].

They also use varied methods to evaluate the goodness-of-fit between model results and observed data. Of those constructing a likelihood function for use in Bayesian inference, some split case incidence data into intervals and assume independent Poisson ([36]) or normal ([48]) distributions to calculate likelihoods, some use data augmentation to construct analytical likelihoods ([44]). Approximate Bayesian computation (ABC) can be used when likelihoods are difficult to calculate, allowing researchers to use an alternate measure of "similarity," such as an exponentially weighted error function [38]. For non-Bayesian

approaches, objective functions include L1 norms [48], mean squared error [18, 1, 12], etc.

Lastly, they use different sampling and search strategies to realize their goodness-of-fit scores into parameter characterizations. Sequential ABC can be used to generate parameter posterior estimates according to Bayesian inference principles [38], as can Markov chain Monte Carlo (MCMC) methods [36, 44], sequential Monte Carlo methods [31], or Latin hypercube sampling [48]. Latin hypercube sampling has also been used to enable plausible parameter set identification [12]. Bayesian optimization has been used to find the Pareto set of best parameters [38]. Grid search [18, 1] and Nelder-Mead direct search [48] methods have been used to identify best-fit parameters.

In this paper, we will aim to estimate parameter posteriors using two methods. In the first, we perform Bayesian inference using MCMC, with likelihoods constructed using an approximate empirical probability distribution function. In the second, we use ABC, where random sampling is used to construct an approximate posterior according to an ABC rejection algorithm.

4.2. Methodology

In this section, we describe our methodology, including the creation of our ABM, synthetic data generation, calibration methods, and credible interval evaluations (Fig. 4-1). We will run calibration tests for two scenarios: the first will only have one parameter to calibrate, the second will have two, and will be referred to as the "one-parameter case" and "two-parameter case" respectively.

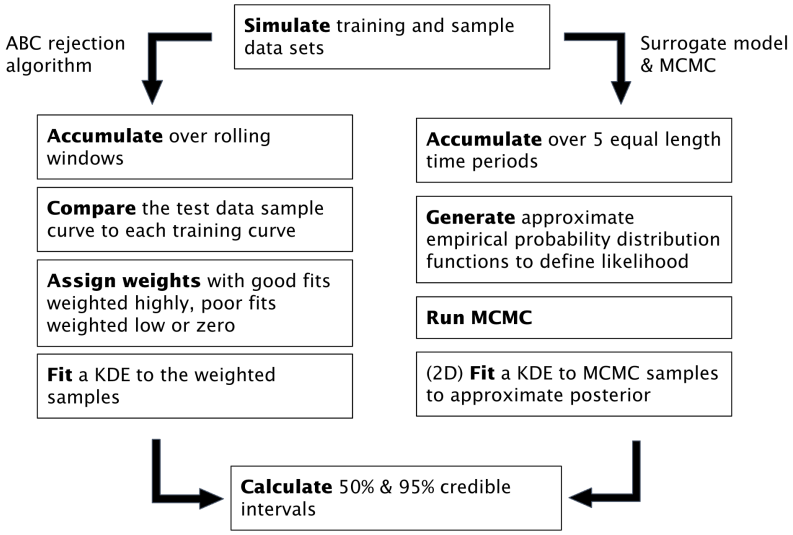

FIGURE 4-1. Flowchart of calibration method testing process using synthetic data.

4.3. Agent-Based Model Framework

Following a similar approach to Zohdi [52], we implement an agent-based modeling scheme in which each agent represents a member of the population. We used a 2D domain denoted by the Cartesian x and y directions. The domains for our studies were rectangular, defined by limits x_{low}, x_{high}, y_{low}, y_{high}. Each domain is a simple representation of a sub-population (e.g., a building or small community).

Agent motion

Agents move at each time step in a random direction with random velocity according to the following definitions:

$$x(t+\Delta t) = x(t) + M * \Delta t * \delta_x \tag{4.1}$$
$$y(t+\Delta t) = y(t) + M * \Delta t * \delta_y \tag{4.2}$$

where the agent's position is $\boldsymbol{r} = (x, y)$, Δt is the time step, M is the mobility parameter in units of distance per unit time, and δ_x and δ_y are random numbers sampled from a uniform distribution between -1 and 1 ($\delta_x \sim U(-1,1)$, $\delta_y \sim U(-1,1)$). Higher mobility parameters allow agents to move more quickly, which accelerates the rate of disease spread. The agents move independent of the motion of the other agents, and agents are not allowed to move outside of their domains. After implementing the movement, if an agent is outside its domain, it is reflected back according to Algorithm 1.

Algorithm 1 Agent bouncing algorithm

```
while x < x_low ||x > x_high do
    if x<x_low then
        x = 2*x_low-x
    else
        x = 2*x_high-x
    end if
end while
while y < y_low ||y > y_high do
    if y<y_low then
        y = 2*y_low-y
    else
        y = 2*y_high-y
    end if
end while
```

When there are multiple domains/sub-populations, agents are allowed to "jump" between domains, similar to a person visiting different buildings and interacting with the people within. At each step of the simulation, an agent will have a chance to jump based on a jumping probability parameter, J. At each time step, a random number between 0 and 1 is sampled, and if it is less than the jumping probability parameter J, the agent is moved to a random position within a random new domain. If it is higher, the agent remains in its current domain.

Agent States

To model the infectious states of the agents, our framework tracks whether an agent is Susceptible (S), Exposed (E), Infected (I), or Recovered (R). When an infected agent is within infection distance D_I of a susceptible agent, that susceptible agent becomes exposed (Eq. 4.3).

$$\|\boldsymbol{r}_i - \boldsymbol{r}_j\| < D_I \tag{4.3}$$

where $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ are the position vectors of agents i and j. An exposed agent becomes infected after time $t_{incubate}$, and an infected agent becomes recovered after time $t_{infection}$, at which point they no longer expose any nearby susceptible agents. The values of $t_{incubate}$ and $t_{infection}$ are sampled from gamma distributions and are unique to each agent – they are assigned at the beginning of the simulation, and are fixed throughout. All agent states are tracked throughout the simulation. This data is summarized as the total number of S, E, I, R, and new infected agents per sub-population at each time step. We track status counts based on an agent's original sub-population.

Simulation Framework and Workflow

The general algorithm of our ABM simulation is as follows:

1. Initialize sub-population(s):
    a) Generate random positions for each agent.
    b) Assign initial disease states (S, E, I, or R) based on initial inputted fractions.
    c) Assign values of $t_{incubate}$ and $t_{infection}$ to each agent.
2. Step forward in time until simulation duration, t_{Sim}, is reached. For each time step:
    a) If using multiple domains, update agent positions via jumping.
    b) Apply motion from Eqs. 4.1 and 4.2, and enforce domain boundaries using Algorithm 1.
    c) Update agent states based on time parameters $t_{incubate}$ and $t_{infection}$.
    d) Check distance between infected agents and susceptible agents, update states if they are within infection distance (Eq. 4.3).
    e) Save agent state count data.

Following the algorithm described above, we can then output our quantities of interest. In this case, we are investigating the number of new infections per time step.

The inputs to the simulation are summarized in Table 1. Two simulations are described, corresponding to the one- and two-parameter cases: Simulation 1 has a single sub-population, while Simulation 2 has two sub-populations. In Simulation 1, the parameter of interest for calibration is mobility, while in Simulation 2, we wish to calibrate both mobility and jumping probability. The values are nondimensional, having been normalized by their respective units of length, time, etc.

Table 1. Parameters used in ABM

Symbol	Parameter	Simulation 1 Input Values	Simulation 2 Input Values
Δt	Time step	0.1	0.1
T	Total length of simulation	300	300
m	Total number of sub-populations	1	2
$\boldsymbol{x}_{low} = (x^0_{low}, ..., x^m_{low})$	Lower x-direction domain boundaries	0	(0, 0.2)
$\boldsymbol{x}_{high} = (x^0_{high}, ..., x^m_{high})$	High x-direction domain boundaries	0.1	(0.1, 0.3)
$\boldsymbol{y}_{low} = (y^0_{low}, ..., y^m_{low})$	Lower y-direction domain boundaries	0	(0, 0)
$\boldsymbol{y}_{high} = (y^0_{high}, ..., y^m_{high})$	High y-direction domain boundaries	0.1	(0.1, 0.1)
$\boldsymbol{N}_{pop} = (N^0_{pop}, ..., N^m_{pop})$	Sub-population sizes	100	(100, 100)
$\boldsymbol{S_0} = (S^0_0, ..., S^m_0)$	Initial fractions of susceptible agents	0.99	(0.99, 1)
$\boldsymbol{E_0} = (E^0_0, ..., E^m_0)$	Initial fractions of exposed agents	0	(0, 0)
$\boldsymbol{I_0} = (I^0_0, ..., I^m_0)$	Initial fractions of infected agents	0.01	(0.01, 0)
$\boldsymbol{R_0} = (R^0_0, ..., R^m_0)$	Initial fractions of recovered agents	0	(0, 0)
D_I	Infection distance	0.005	0.005
	Incubation time mean	11.6[a]	11.6[a]
	Incubation time standard deviation	1.9[a]	1.9[a]
	Infection time mean	18.49[a]	18.49[a]
	Infection time standard deviation	3.71[a]	3.71[a]
M	Mobility	varies	varies
J	Jumping probability	0	varies

[a] Consistent with incubation and infection period (from fever to recovery) of smallpox as reported in [13], if the unit of time is days.

4.4. Bayesian inference with MCMC

Likelihood construction

We use an adaptive Metropolis-Hastings MCMC (AMCMC) algorithm to determine the posterior distribution of the parameters given an observed data set [16, 11]. Due to the large number of iterations required for MCMC, directly sampling from the model can be time-intensive and impractical. To circumvent this, we instead evaluate likelihood using approximate empirical probability distribution functions (PDFs) constructed from training data. This method also has the advantage of avoiding direct comparison between the observed data and a single sample from the model, which, due to stochasticity, may lead to MCMC getting "stuck" at a good match [36].

The approximate empirical PDFs were made using a training data set sampled along a grid of parameter values. Mobility M was sampled at 17 equally spaced points between $[0.005, 0.025]$. For the one-parameter case, jumping probability J was held constant at 0, while for the two-parameter case, jumping probability was sampled at 10 equally spaced points between $[0, 0.001]$. At each parameter combination, 5000 different random seeds were run. For each training data sample, $\tilde{x}$, the data consists of time series of new infections per time step for each sub-population. We represent the number of new infections at time step t and for sub-population k as $\tilde{x}_{t,k}$. Each vector $\tilde{\mathbf{x}}_{:,\mathbf{k}}$ was processed by summing the total number of new infections over $n = 5$ evenly spaced time intervals, resulting in a training data set of summary statistics $\tilde{s}$ containing new infection counts $\tilde{s}_{j,k}$, where $j = 1, ..., 5$ is the time interval (Fig 4-2). Likewise, test data time series $\mathbf{x}_{:,\mathbf{k}}$ are summed over time intervals to result in test data summary statistics, $s_{j,k}$.

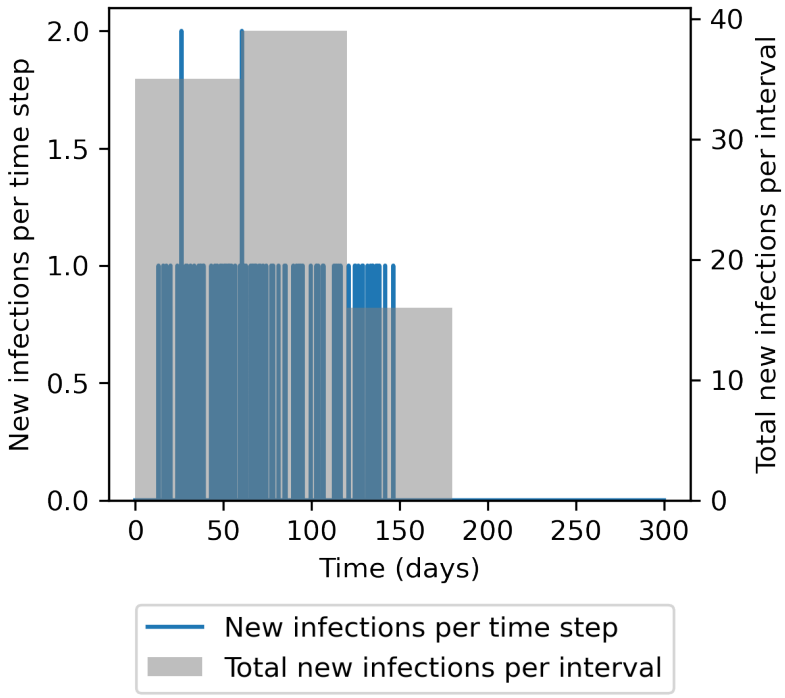

FIGURE 4-2. Visualization of data processing, illustrating summation of new infections per time step over five equally sized time intervals.

A PDF of new infections in a certain time interval and sub-population given a parameter set can be described using Gaussian kernel density estimation (KDE) on the training data. For a given parameter set, there are $n * m$ PDFs, where n is the number of intervals, and m is the number of sub-populations. These PDFs describe the approximate empirical likelihood of obtaining a certain number of new infections at a specified time interval and sub-population, and for a specified parameter set. To expand beyond the discrete parameter combinations contained in the training data set, PDFs can be interpolated [9]. The interpolation process is described in detail in Appendix A.

Each approximate empirical PDF is notated $f_{j,k}(s_{j,k}|\theta)$, where θ is the parameter vector.

Given a test data sample s, AMCMC is used to construct the posterior, $P(\theta|s)$. At each iteration, the log-likelihood $l(s|\theta_i)$ is evaluated according to Eq. 4.4 for the parameter set θ_i, where i is the iteration number. This likelihood construction assumes that data is independent across time intervals and sub-populations.

$$l(s|\theta_i) = \sum_{k=1}^{m} \sum_{j=1}^{n} ln(f_{j,k}(s_{j,k}|\theta_i)) \tag{4.4}$$

AMCMC sampling

We use PyUQTk for AMCMC [11]. This implementation uses Metropolis-Hastings sampling, where a parameter set θ is pulled from a proposal distribution at each iteration, i, and accepted or rejected based on

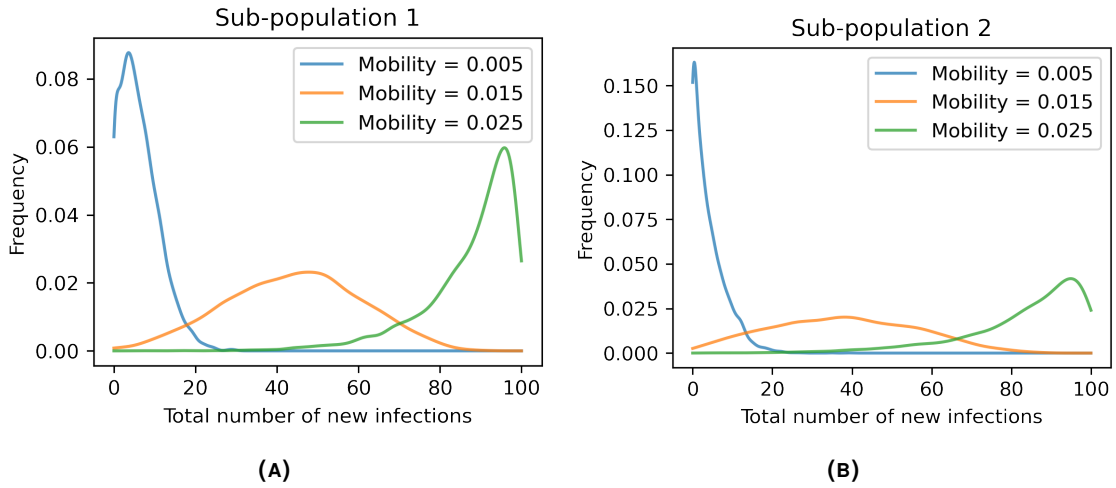

FIGURE 4-3. Approximate empirical PDFs for sub-population 1 (Fig. A) and 2 (Fig. B) at the first time interval generated via KDE from training data. Jumping probability is constant at $7.78e-4$.

the likelihood of the observed data at the parameters. To evaluate the quality of AMCMC samples, we calculate the effective sample size using PyUQTk [11].

The initial chain location for the one-parameter case was $\theta_0 = (M = 0.0151)$, while the initial chain location for the two-parameter case was $\theta_0 = (M = 0.0151, J = 0.00051)$. AMCMC is run for 75,000 iterations, with a non-adaptive period length $\nu = 5000$ and a burn-in of 25,000 iterations. Further details are described in Appendix B.

We use a uniform prior over our domain. In the 1-d case, this domain spans mobility values between 0.005 and 0.025. In the 2-d case, the domain is rectangular and spans mobility values between 0.005 and 0.025 and jumping probability values between 0 and 0.001.

4.5. Bayesian inference with an ABC rejection algorithm

ABC methods allow one to avoid direct likelihood calculations in favor of a chosen distance function, $\|\cdot\|$. In an ABC rejection algorithm, parameters are sampled from the prior, $\theta_i \sim p(\theta)$, and run through the model, outputting data $\tilde{x}$ [45]. The data are processed to result in summary statistics $\tilde{s}$, generally chosen to encapsulate important information while reducing dimensionality. Samples are weighted according to a weight function, K_ε, in which "closer" matches to the summary statistics of observed test data s are assigned higher weights, while poor matches are assigned small or zero weights. ε is a tolerance proxy parameter controlling the range of accepted samples. This process results in weighted samples distributed along $p_\varepsilon(\theta|s) = \int p_\varepsilon(\theta, \tilde{s}|s)d\tilde{s}$, which is approximately equal to the true desired posterior, $p(\theta|x)$ [33, 2].

$$p_\varepsilon(\theta, \tilde{s}|s) \propto K_\varepsilon(\|\tilde{s}, s\|)p(\tilde{s}|\theta)p(\theta) \tag{4.5}$$

To generate our training data samples $\tilde{x}$, we create a training data set with parameter values sampled from the priors. For the one-parameter case, mobility values M were sampled from a uniform prior distribution,

$M \sim U(0.005, 0.025)$, while jumping probability was held constant ($J = 0$). For the two-parameter case, mobility values were again sampled according to $M \sim U(0.005, 0.025)$ and jumping probabilities J were sampled from a uniform distribution, $J \sim U(0, 0.001)$. 85,000 total runs were generated for the one-parameter case (Simulation 1), while 850,000 were generated for the two-parameter case (Simulation 2). Within the two training data sets, each run used a different random seed. As before, the training data consists of time series of new infections per time step.

In our implementation, the distance function is the sample rank relative to the rest of the training data. The rank is determined by calculating scores $\rho(\tilde{s}, s)$. In the one-parameter case, scores are defined using an L2 norm: $\rho(\tilde{s}, s) = \|\tilde{\mathbf{s}}_{:,\mathbf{1}} - \mathbf{s}_{:,\mathbf{1}}\|_2$. In the two-parameter case, the data from sub-populations 1 and 2 are concatenated into a vector $(\tilde{\mathbf{s}}_{:,\mathbf{1}}, \tilde{\mathbf{s}}_{:,\mathbf{2}})$. The score is then calculated: $\rho(\tilde{s}, s) = \|(\tilde{\mathbf{s}}_{:,\mathbf{1}}, \tilde{\mathbf{s}}_{:,\mathbf{2}}) - (\mathbf{s}_{:,\mathbf{1}}, \mathbf{s}_{:,\mathbf{2}})\|_2$. For example, the training data sample with the lowest score $\rho(\tilde{s}, s)$ relative to other training data samples has rank 0, therefore its distance function is evaluated: $\|\tilde{s}, s\| = rank(\rho(\tilde{s}, s)) = 0$. A visualization of scoring in the one-parameter case is shown in Fig. 4-4.

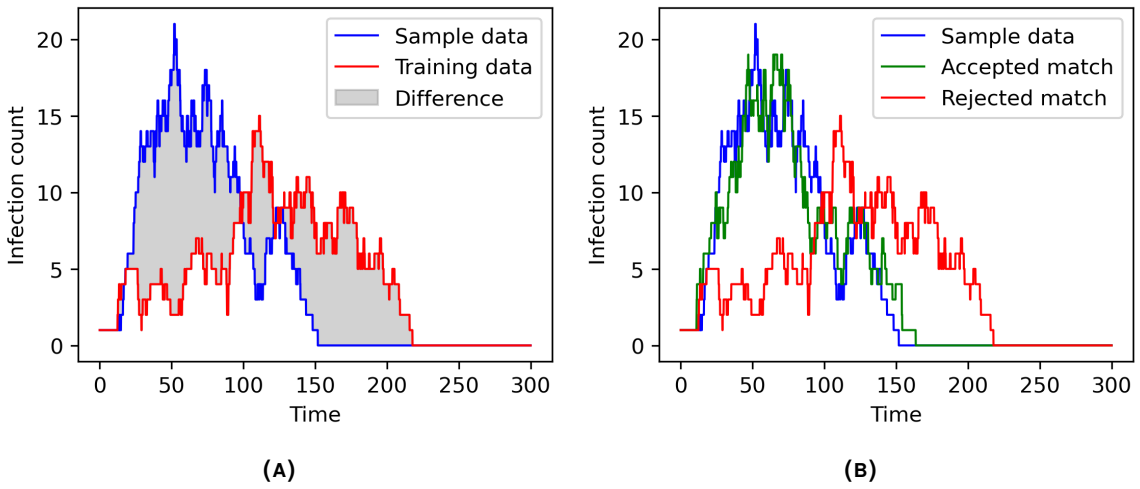

FIGURE 4-4. A Visualization of L2 norm score. B Example of a relative "good" and "bad" match to a sample curve based on L2 norm score.

To generate our summary statistics s, we processed the training data via accumulation over a rolling window (Fig. 4-5). The small population sizes and time steps in the simulations result in sparse data, with the number of new infections frequently zero. Unprocessed, this means that tolerance levels must be very low to find meaningfully similar matches — a new infection occurring a single time step apart in two data sets is penalized the same as a new infection occurring 100 time steps apart. This would require very large amounts of training data to obtain a significant number of similar samples. To address this, we accumulate the data over a rolling window, such that $s_{t,k} = \sum_{i=1}^{w} x_{t-w+i,k}$, where w is the number of time steps in the window, and any data at $t < 0$ is taken to be 0. In preliminary studies, a window of 14 (140 time steps) was effective. This allows all of the information in the original data set to be preserved (unlike in a binning process), which means our approximate posterior $p_\varepsilon(\theta|s)$ is exactly equal to $p_\varepsilon(\theta|x)$. This process does not decrease the dimensionality of the data, which most ABC implementations aim to do when creating summary statistics.

We test four weight functions to study the effect of weight function shape on performance (Fig. 4-6). The

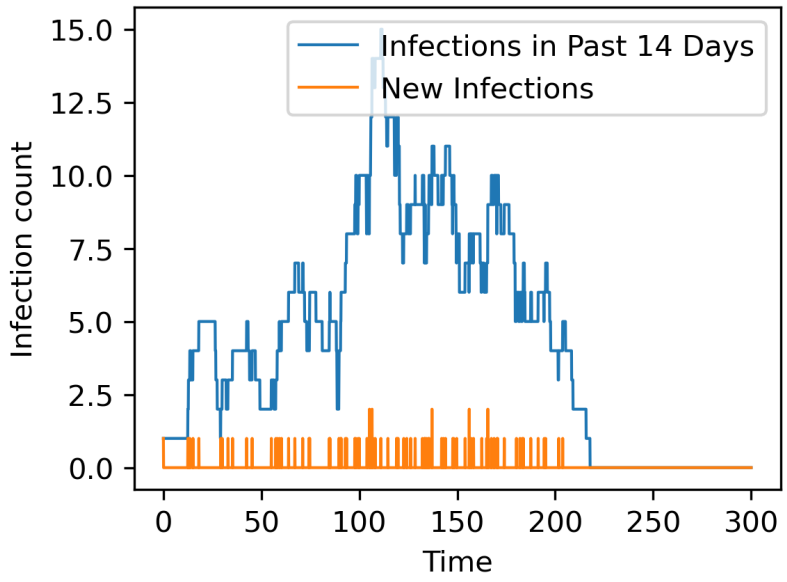

FIGURE 4-5. The time series of new infections (orange) is accumulated in rolling 14 time interval windows (blue). The value of the accumulated curve (blue) can be interpreted as the number of new infections in the last 14 time intervals.

first is a step function, as shown in Eq. 4.6, where c is a normalizing constant. Using a step function leads to an accept-reject algorithm with no intermediate weighting. The other three weight functions provide some level of smoothing: the Epanechnikov kernel (Eq. 4.8), a negative exponential (Eq. 4.9), and a linear piece-wise function (Eq. 4.7). The linear function and Epanechnikov kernel both have hard cut-offs, above which samples are rejected (weight is set to 0), while the negative exponential places some non-zero weight on every sample.

Each weight function contains a variable δ_ε, defined as the value at which the centroid of the weight function $K_\varepsilon(d)$ over positive values $d = [0, \infty)$ is equal to ε. Larger weight function centroid values cause more samples to have non-zero or high weights, while smaller centroid values restrict high weights to samples with very small distance measures. This allows us to directly compare weight function shapes while holding the tolerance to an equivalent standard.

$$K_\varepsilon^S(d) = \begin{cases} c, & \text{if } d \leq \delta_\varepsilon \\ 0, & \text{if } d > \delta_\varepsilon \end{cases} \tag{4.6}$$

$$K_\varepsilon^L(d) = \begin{cases} c(\delta_\varepsilon - d), & \text{if } d \leq \delta_\varepsilon \\ 0, & \text{if } d > \delta_\varepsilon \end{cases} \tag{4.7}$$

$$K_\varepsilon^E(d) = \begin{cases} c\delta_\varepsilon^{-1}(1-(d/\delta_\varepsilon)^2), & \text{if } d \leq \delta_\varepsilon \\ 0, & \text{if } d > \delta_\varepsilon \end{cases} \tag{4.8}$$

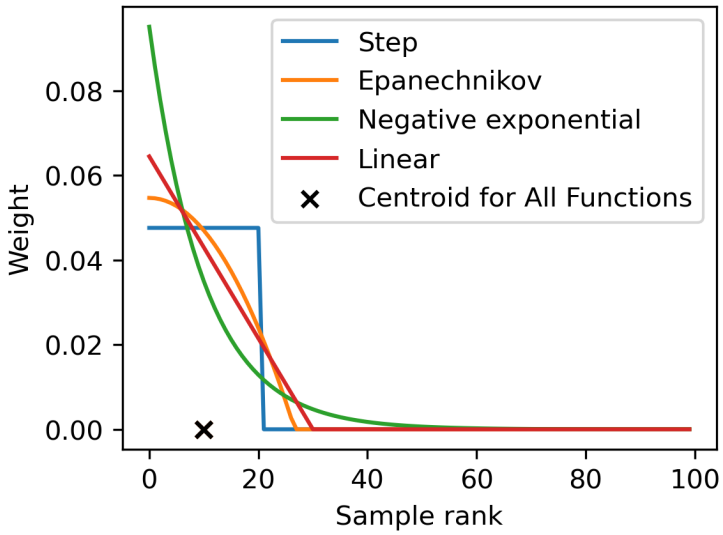

FIGURE 4-6. Rejection function shape varying with constant centroid at fraction 0.1 of sample rank

$$K_\varepsilon^N(d) = c\delta_\varepsilon^{-d} \tag{4.9}$$

4.6. Credible interval generation

We calculate the 50% and 95% credible intervals for all parameter predictions. In the one-parameter case, MCMC chain values can be converted into credible intervals by finding the 25 and 75 percentiles of the chain (50% CI bounds) and the 2.5 and 97.5 percentiles (95% CI bounds). Weighted samples from the ABC rejection algorithm are first converted into an approximate posterior using KDE, which is then integrated from both sides to obtain equal-tailed 50% and 95% credible intervals (Fig. 4-7).

In the two-parameter case, both MCMC samples and weighted ABC samples are processed into an approximate posterior using KDE. The posterior is sampled along a grid of 101 by 100 (mobility by jumping probability). To determine the credible interval bounds, we calculate the smallest area containing 50% or 95% of the integral under the posterior according to Algorithm 2. A parameter set is classified as being in the credible interval if the nearest sampling grid point is in the credible interval (i.e., point has a corresponding value of 1 in the `inside` matrix according to Algorithm 2).

The function can also be integrated continuously to form a cumulative distribution function (CDF). If parameter estimates for a previous outbreak are familiar to the policy makers, a line can be added to the plot to estimate the probability that the current outbreak under analysis is "worse" than the historical outbreak (Fig. 4-8).

4.7. Test data generation

For the one-parameter case, test data was generated by running the model with mobility values pulled from $M \sim U(0.01, 0.02)$ with jumping probability values held constant at 0. For the two-parameter case, test

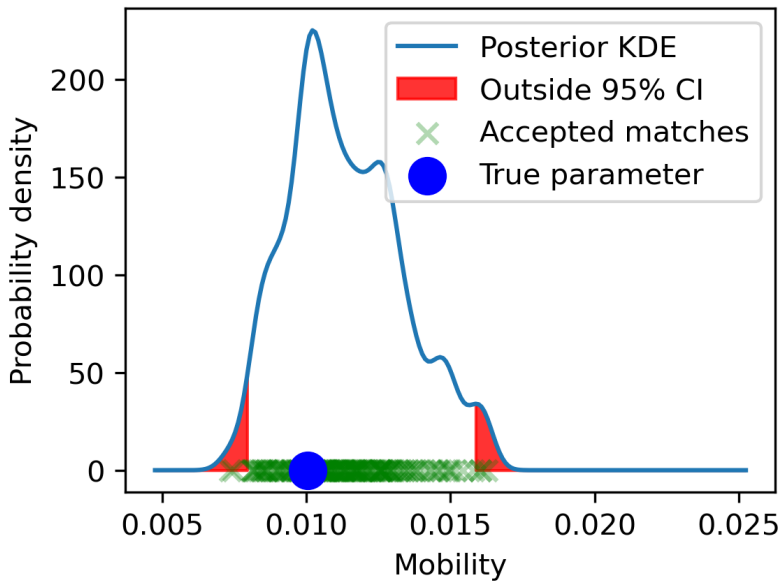

FIGURE 4-7. For the ABC method, the KDE is generated from a weighting of close matches. Integrating from both sides yields the confidence interval.

Algorithm 2 2D credible interval determination

Input: Vector `Z_vect` containing unraveled likelihood or posterior values, credible interval fraction `target`, number of points in x-direction `n_x`, number of points in y-direction `n_y`

```
Initialize vector of zeros inside_vect with length = length(Z_vect)
Z_vect_norm = Z_vect/(sum(Z_vect))
total = 0
i = 0
while total < target do
    total += Z_vect_norm[i]
    inside_vect[i] = 1
    i += 1
end while
inside ← reshape inside_vect to (n_x, n_y)
```

data was generated using mobility values pulled from $M \sim U(0.01, 0.02)$ and jumping probability values from $J \sim U(2E-4, 8E-4)$. 500 model runs were collected for each test data set.

4.8. Testing inference performance

We test the credible intervals generated using Bayesian inference for their frequentist properties. Under a frequentist paradigm, we expect that for repeated calibration tests on different test data, the 95% confidence interval will contain the true parameter value 95% of the time, and similarly that the 50% confidence interval will contain the true parameter value 50% of the time. We test if this holds true over 500 test data samples.

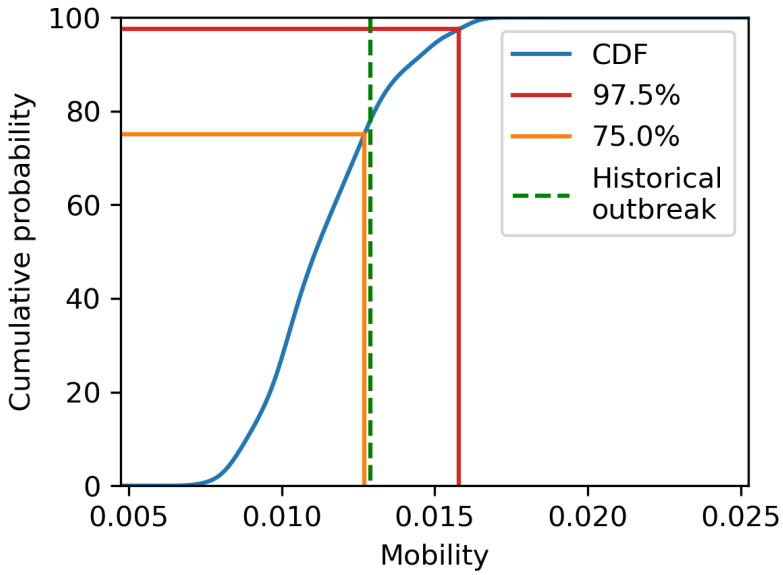

FIGURE 4-8. Sample cumulative distribution with probabilities highlighted. Historical outbreaks can be added to CDF plots to aid in understanding outbreak severity.

4.9. Results

For both MCMC and ABC, we test calibration on the 500 runs contained in the test data sets to assess performance. Additionally, for the template matching ABC method, we run a sensitivity analysis on several meta-parameters to determine their effect on calibration results.

MCMC calibration results

Since a uniform prior was used in generating the parameter posterior, a brute-force grid sampling of the likelihood function (Eq. 4.4) can be used to check for MCMC convergence, as the likelihood should be proportional to the posterior. Qualitatively, nearly all MCMC-constructed posteriors appeared to match the shape of the likelihood function, which indicates sufficient MCMC iterations (outlier cases are discussed later). For the one-parameter case, the effective sample size (ESS) had a mean value of 10642. For the two-parameter case, the ESS for mobility values had a mean of 5201, while the ESS for jumping probability values had a mean of 5191. We expect these ESS values to be sufficient in establishing a reliable posterior estimate.

One-parameter case results

Example calibration results are shown in Fig. 4-9. Though the true parameter value does not fall within the 50% and 95% credible intervals in all three examples, likelihood predictions qualitatively match MCMC histograms well. Overall, of the 500 test data samples in the one-parameter case, true parameter values fell within the 50% credible interval at a 22% rate and within the 95% credible interval at a 64.6% rate.

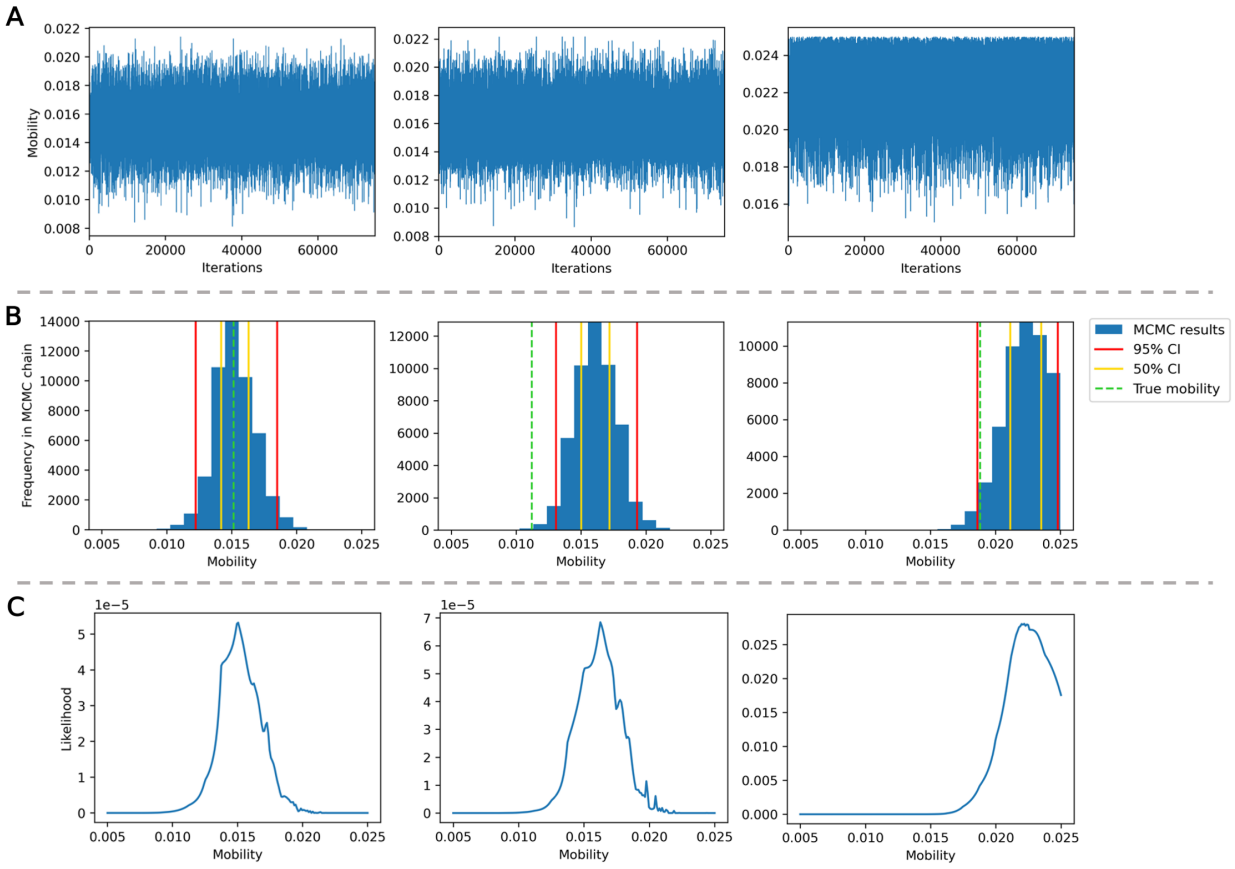

FIGURE 4-9. Each column represents the calibration results for a different test data sample: A MCMC trace plots. B Histograms of MCMC chain results, with 50% and 95% credible interval bounds marked. C Likelihood functions sampled over grid of mobility values.

Two-parameter case results

Example calibration results for the two-parameter case are shown in Fig. 4-10. Again, likelihood plots qualitatively match MCMC posterior plots well. For the two-parameter case, true parameter values fell within the 50% confidence interval at a 19.4% rate and within the 95% confidence interval at a 57.8% rate.

ABC rejection calibration results

Sensitivity studies

There are three primary meta-parameters in the ABC process: the weight function shape, the weight function centroid value (ε), and the kernel density estimate (KDE) bandwidth. The centroid value is used as the characteristic width of the weight function so that a meaningful comparison can be done between weight function shapes. For example, with 500 training data samples, using a step function with the centroid set at 0.02 would correspond to 4% of the samples (20 of them) being accepted at equal weight with the rest of the samples rejected. We perform a sensitivity analysis on these meta-parameters, evaluating performance by repeated calibration over 500 synthetic sample pandemics from the test data set. Sensitivity analyses are performed for the one-parameter case only. For each sensitivity analysis, one

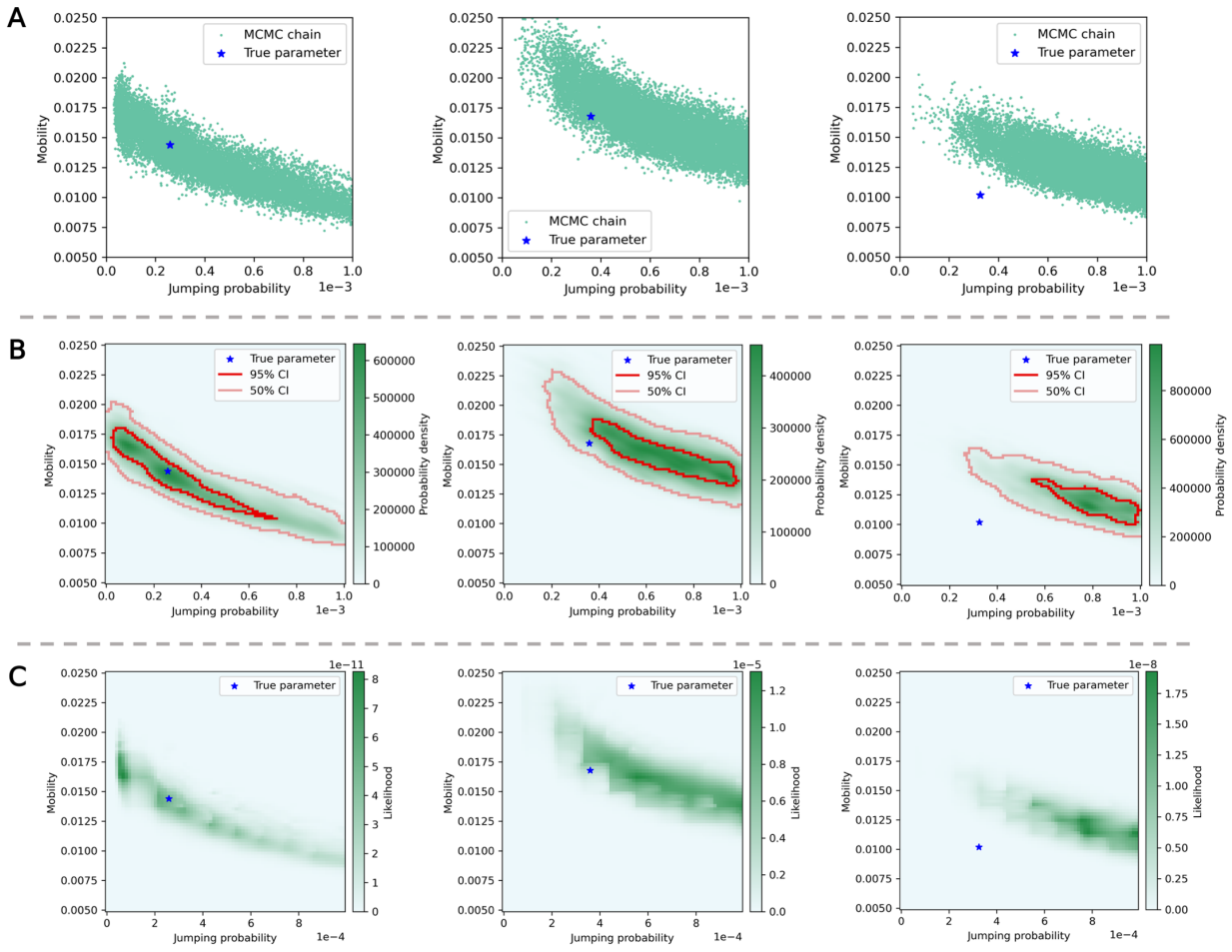

FIGURE 4-10. Each column represents the calibration results for a different test data sample: A Scatter plot of MCMC chain values. B Approximate posterior constructed using KDE, with 50% and 95% credible interval bounds marked. C Likelihood function sampled over grid of parameter values.

meta-parameter is varied while the other two remain fixed. Based on preliminary results, the fixed parameters to be used when the parameter is not the subject of the sensitivity analysis are the Epanechnikov kernel with centroid at 0.02 and a KDE bandwidth of 0.2.

Four weight functions (Fig. 4-6) are tested: step, Epanechnikov kernel, negative exponential, and linear. The results of varying the weight function while holding the other two meta-parameters constant as described above are shown in Fig. 4-11.

Next, we test the effect of changing the weight function centroid over a wide range of values from 0.0001 to 0.05 (Fig. 4-12). In the extreme limits, poor performance is expected. For example, consider the extreme where tolerance is very high, and nearly all of the training data set are accepted as good matches. This results in the posterior distribution simply recreating the prior distribution. In our case, this will result in overly wide confidence intervals, since our prior is uniform. On the other extreme, a very small rejection function centroid may result in very few training data set samples being given non-zero weights, which may be insufficient to generate an accurate posterior estimation, particularly given the stochastic nature of

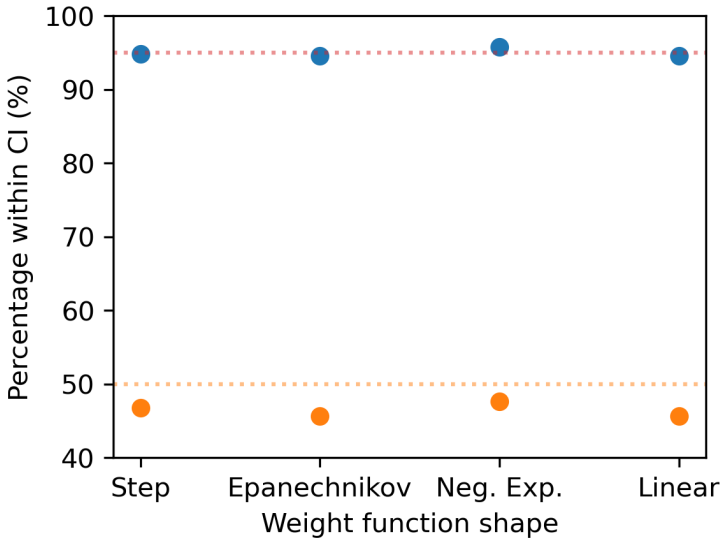

FIGURE 4-11. Confidence interval accuracy for 4 different weight function shapes

the ABM. An extremely small tolerance may result in artificially narrow posteriors or artificially multi-modal posteriors in some cases.

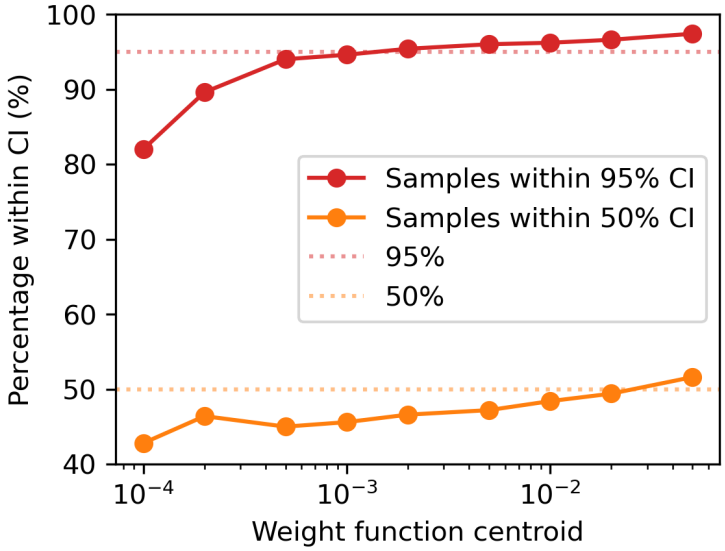

FIGURE 4-12. Confidence interval accuracy versus centroid location

Lastly, we test the sensitivity to KDE bandwidth by varying it 0.05 to 0.45. The Silverman bandwidth [43] recommendation is around 0.45 for the number of data points, which is included in the range. Fig. 4-13 shows the parameter posterior of a single sample case for KDE bandwiths across the range. Fig. 4-14 shows the sensitivity results to the KDE bandwidth changing.

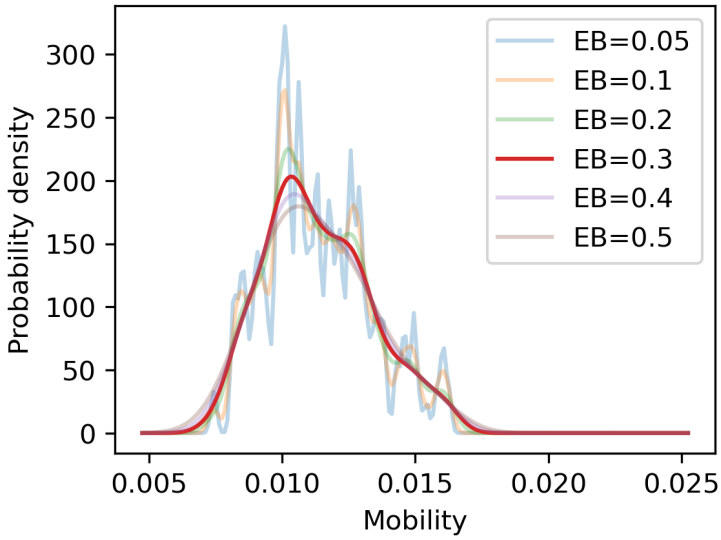

FIGURE 4-13. Posterior for a sample case with varying KDE bandwidths

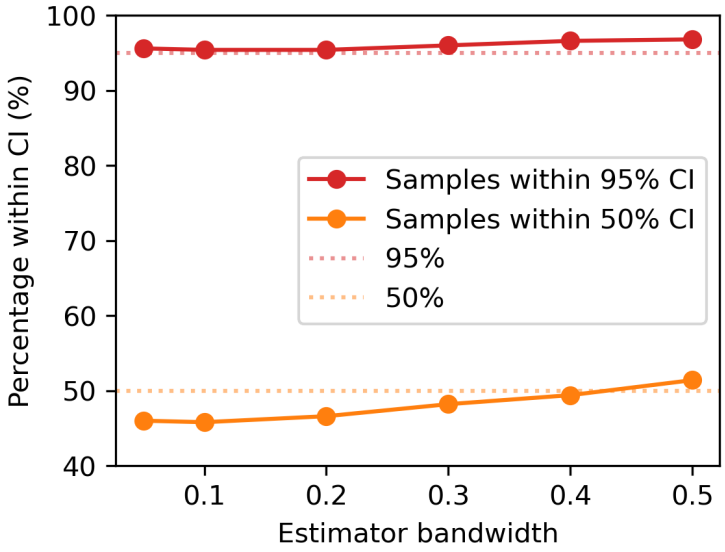

FIGURE 4-14. Confidence interval accuracy versus KDE estimator bandwidth

One-parameter case results

Between all three sensitivity studies, though minor differences in performance exist, the method is robust to large variations in the meta-parameters. Also, it is expected that some of the small differences may be explained by noise from the stochastic data set.

The shape of the weight function does not to have a significant impact on performance, so we will proceed with the Epanechnikov because it is between the step and linear options and has the added computational

benefit of setting many weights to zero unlike the negative exponential function.

The weight function centroid has a wide range of well performing values, from 0.0005 to 0.01. We will proceed with centroid set to 0.02 as a logarithmic midpoint.

For the KDE bandwidth, 0.3 is selected. This gives both the best confidence interval performance in figure 4-14 and qualitatively is the smallest bandwidth value to give reliably smooth distributions. For example, see the 0.3 bandwidth KDE curve in 4-13.

These meta-parameters result in the true parameter falling within the 50% credible interval 48.3% of the time and within the 95% interval and 96.0% of the time. More importantly, the method is robust to relatively large changes in the meta-parameters, meaning that only rough tuning is required for reliable results in analysing new data sets.

Two-parameter case results

The meta-parameter selection for the 1D study are used tested in 2D. An example of calibration results is shown in Fig. 4-15. In testing calibration for the 500 test data runs, 49.2% of the runs had true parameter values fall within the 50% confidence interval, while 96.8% had parameters fall within the 95% confidence interval.

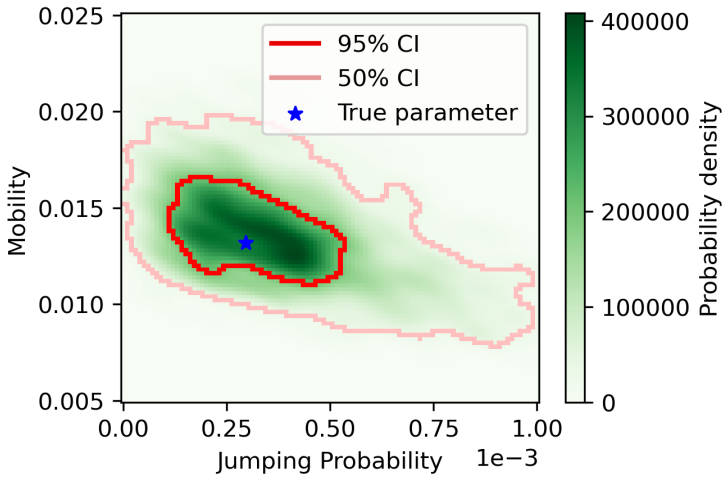

FIGURE 4-15. Sample 2D KDE

Table 2. Summary of rate of true parameters within predicted credible intervals

		50% CI	95% CI
One-parameter case	ABC	48.3%	96.0%
	MCMC	22%	64.6%
Two-parameter case	ABC	49.2%	96.8%
	MCMC	19.4%	57.8%

4.10. Discussion

In this work, we have evaluated the performance of calibration methods by testing the probability that true parameter values will be recovered within the 50% and 95% confidence intervals over repeated tests. Theoretically, if we were able to capture the true posterior $p(\theta|x)$ with our calibration methods, we would not necessarily have the property that the true parameter will be within the 95% credible interval 95% of the time, or that the true parameter will be within the 50% credible interval 50% of the time, because our test data was not sampled directly from the prior, and, in the two-parameter case, because this claim cannot be made over a multi-dimensional parameter space. However, this test still provides a valuable metric to evaluate performance with. It is reasonable to desire good coverage properties of the credible interval (in a frequentist sense) over a subset of parameters, and that is what this test represents. We evaluate the results keeping in mind that obtaining the true parameter within a 95% credible interval for 95% of test data samples is not strictly expected, but desired.

Our calibration results indicate that the MCMC method is not able to represent parameter uncertainty with desired coverage properties. Since we used a uniform prior, and seeing that the MCMC posteriors qualitatively matched the likelihood functions well, we suggest that this can be primarily attributed to an inaccurate underlying likelihood. Within the likelihood, some sources of error include the approximation inherent in constructing the approximate empirical probability density function due to finite sample availability and the use of KDE, as well as the approximations made in interpolation between PDFs. There is also some information lost when converting the raw data into summary statistics, since the summary statistics are not sufficient. However, we believe the primary source of error is the assumption of independence between time intervals, and for the two-parameter case, sub-populations, which is made in the formulation of the likelihood. This assumption is not technically correct, despite being used previously in the field [36, 48]. More importantly, these results show that it can lead to unfavorable aggregate results.

Beyond aggregate results, we can also identify particular test data samples which lead to obviously poor posterior predictions from the MCMC method. For example, the likelihood values for a test data sample with zero new infections across all time steps is shown in Fig. 4-16A. The likelihood is negligible throughout the domain apart from a very small section in the bottom-left corner, where jumping probability and mobility values are low. The posterior prediction constructed with MCMC is not shown because MCMC completely failed to converge for this sample, obtaining only a few unique chain values. Contrast this with the posterior results from the ABC rejection method, shown in Fig. 4-16B, where a much wider spread of values are given high probability. We can compare these likelihood and posterior predictions to the spread of parameter values that resulted in the same observed data (zero infections across all time steps) within the training data sets for MCMC (Fig. 4-16C) and ABC (Fig. 4-16D). In the case of the ABC training data, these parameter values represent iid samples from the true posterior [33].

Clearly, the ABC method reflects the true posterior much more closely than the MCMC method, despite both training data sets reflecting a similar spread of matches. We suggest that this is due to the assumption of independence between time intervals and sub-populations in the MCMC likelihood, which exacerbates the stratification of posterior values. This also causes poor MCMC convergence — all of the test data samples with ESS smaller than 2000 in the two-parameter case had similar data (≤ 2 total infections across the entire simulation) and similar likelihood plots.

We find that the ABC rejection method works very well in aggregate. The data processing associated with this method does not cause any information loss, unlike many ABC methods. Therefore, the only approximations made in this method are in using KDE to construct a posterior PDF, and in using a non-zero ε. As long as ε is sufficiently small, we expect good results from ABC. However, because the

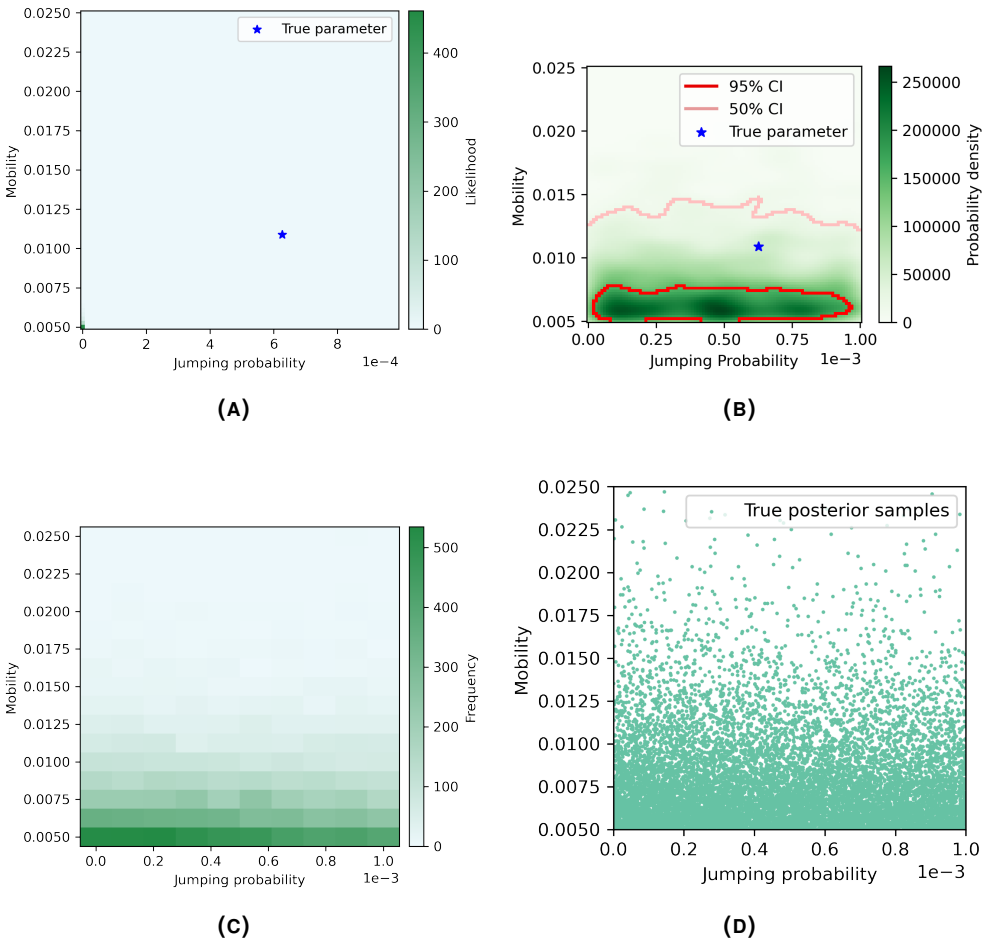

FIGURE 4-16. For calibration on a test data sample with zero new infections over the entire simulation: A Posterior prediction with MCMC method. B Posterior prediction with ABC rejection method. C Frequency of matching data across discretely sampled training data set for MCMC. D Scatter plot of parameter sets resulting in matching data across training data set for ABC.

distance function for ABC is based on the rank of L2 norms with respect to other data in the training data set rather than the objective value of the L2 norms, a test data sample with very few "good" matches will potentially have more biased posterior estimates than other test data samples. Since our performance test does not check the accuracy of individual posteriors or credible intervals, we may miss these effects. Under this method, data generation and distance function evaluations are embarrassingly parallel, and can be computed efficiently. However, we note that this ABC method is ill-suited to high-dimensional problems, since the amount of data and distance function evaluations required will increase very quickly.

The discrepancies that we have found between our MCMC calibration method's predictions and the desired posterior may not be found with typical validation methods in the epidemiology field. Often, when real-world data is used for validation, there are only one or a few data sets available. Some form of posterior predictive check is common: researchers will estimate a posterior, sample parameter values from the posterior, then generate data with those parameter samples [36, 44]. This allows the predicted data distribution $p(\tilde{x}|x)$ to account for both parameter uncertainty and model stochasticity, which can be compared to the observed data qualitatively or quantitatively (note that posterior predictive checking was not necessary for our tests since our training data generation model exactly matches our test data generation model). However, given that there are generally limited data sets available, it is difficult to guarantee model fit across a wide range of parameter values and observed data sets with posterior predictive checks.

Additionally, without a synthetic data test it is usually not possible to directly check if a method is capable of recovering the true values of parameters, since they are often not measureable. This is not of significant practical concern for models focused on prediction, but having interpretable parameters is valuable for qualitative understanding of outbreak dynamics [38]. In the cases where researchers wish to use parameter values directly in some way, it may be beneficial to use synthetic data tests to verify that under ideal conditions (perfect model match), the calibration method is able to accurately predict parameter posteriors. This is not a guarantee of success under real-world data tests, but can identify underlying issues that may be obscured in typical validation techniques.

Some limitations of our performance testing method include that we are only testing two credible interval levels: 50% and 95%. Additionally, this method doesn't test the correctness of individual test data samples' posterior predictions, it instead tests on average across many test data samples.

4.11. Conclusion

We used a synthetic data set-up to test two calibration methods for stochastic ABM models: an MCMC method and an ABC rejection algorithm. Both aim to predict the posterior of the parameter values given observed data. Performance evaluation was done by assessing the probability of true parameter values falling within the predicted 50% and 95% credible intervals across calibration tests on 500 different test data samples, with the desire that the probabilities match the respective credible interval levels. Importantly, we note that this performance test is not expected to be necessarily correct given perfect posterior predictions, but that it is the metric by which we are measuring *desired* performance. The MCMC method performs poorly by this aggregate testing metric, and also is shown to fail in some specific individual instances. We suggest that the primary cause of poor performance in the MCMC method is the assumption of independence between time intervals and sub-populations in the likelihood construction — an assumption that has a precedence of use in epidemiological model calibration. In contrast, the ABC method performs very well, and appears to be robust to changes in meta-parameters.

Our results suggest that calibration methods for epidemiological models may benefit from synthetic data

testing. Posterior predictive checks, common throughout the field, may fail to identify model fit issues across a wide range of parameters due to limited real-world data set availability. Additionally, validation with real-world data generally precludes direct validation of parameter values. In cases where parameter values may be used for interpretation of outbreak dynamics, it may be valuable to first verify that calibration methods are capable of accurate posterior prediction in a setting where the observed data and model data are both generated by the same process, in order to safeguard against underlying issues.

4.12. Appendix

4.12.1. Appendix A: PDF interpolation

To generate the approximate empirical PDFs, the raw training data (time series of new infections per time step) is first processed to sum the number of infections over each interval, resulting in a data set of summary statistics, $\tilde{s}$. We wish to determine the set of PDFs representing the probability of the number of new infections for each time segment and sub-population given a parameter set θ. We will denote the PDFs as $f_{j,k}(s_{j,k}|\theta)$, where j is the represented time interval, and k is the represented sub-population ($j = 1,...,n$, $k = 1,...,m$).

For a parameter set θ that is included in the training data, the PDF $f_{j,k}(s_{j,k}|\theta)$ is found by applying Gaussian KDE to the number of new infections in time interval j and sub-population k ($s_{j,k}$) across all recorded stochastic runs. This is repeated for each time interval and sub-population to create the set of PDFs. To facilitate later interpolation, we calculate and store the means ($\tilde{\mu}$) and variances ($\tilde{K}$) of each of these PDFs. However, at some parameter values, there may be no infections across all stochastic runs (for instance, when jumping probability is zero and an infection begins in sub-population 1, any other sub-populations would have zero new infections). This causes an error when trying to generate a KDE, as the covariance matrix of the data is singular. In this case, we replace the KDE with an approximation, as shown in Eq. 4.10,

$$f_{j,k}^{\sigma=0}(s_{j,k}|\theta) = \begin{cases} \frac{1-(\mu-s_{j,k})/\varepsilon}{\varepsilon}, & \text{if } \mu-\varepsilon \leq s_{j,k} \leq \mu \\ \frac{1-(s_{j,k}-\mu)/\varepsilon}{\varepsilon}, & \text{if } \mu < s_{j,k} \leq \mu+\varepsilon \\ 0, & \text{otherwise} \end{cases} \tag{4.10}$$

where μ is the mean new infections across the training data set for time interval j and sub-population k (in this case, since data is singular, it is the number of new infections seen in every training data sample for the given j and k), and ε is a chosen small value. This represents a PDF with a standard deviation of $\sigma = \sqrt{\varepsilon^2/6}$. This approximate PDF allows us to do the interpolation process as usual, preserving the validity of the PDFs and avoiding divide-by-zero errors when calculating transformed coordinates.

4.12.1.1. One-parameter case interpolation procedure

The following procedure follows Bursal [9]. Say we want to determine the set of PDFs corresponding to parameter set θ that is not contained within the training data set. In the one-parameter case, where θ is a scalar, we can interpolate for the PDFs we want using the training data at the two closest parameter values. Call the lower neighboring parameter $\theta^{(0)}$, and the higher neighboring parameter $\theta^{(1)}$. In general, values relating to the neighboring lower parameter will be denoted with the superscript $^{(0)}$, values relating to the

neighboring lower parameter with the superscript $^{(1)}$, and values relating to the final interpolated PDF will have no superscript.

Following [9], first calculate a normalized interpolation coordinate α according to:

$$\alpha = \frac{\theta - \theta^{(0)}}{\theta^{(1)} - \theta^{(0)}} \tag{4.11}$$

Calculate the approximate mean of the distribution at parameter θ by interpolating between the means of the distributions at neighboring parameters according to:

$$\mu = (1-\alpha) * \mu^{(0)} + \alpha * \mu^{(1)} \tag{4.12}$$

Calculate the approximate variance of the distribution at parameter θ by interpolating between the variances of the distributions at neighboring parameters according to Eq. 4.13. The variance values of $K^{(0)}$ and $K^{(1)}$ are pulled from the set of variances calculated directly from the training data, and therefore may include zeroes.

$$K = (1-\alpha) * K^{(0)} + \alpha * K^{(1)} \tag{4.13}$$

We establish a second representation of the variance:

$$\hat{K} = \begin{cases} \varepsilon^2/6, & \text{if } K = 0 \\ K, & \text{otherwise} \end{cases} \tag{4.14}$$

Accordingly, $\hat{\sigma} = \sqrt{\hat{K}}$.

We specify the coordinates at which to sample the interpolated PDF, s, dropping the subscript j,k throughout this section for simplicity. Then, we define the transformed coordinates $s^{(0)}$ and $s^{(0)}$, which are the values at which the distribution corresponding to the upper and lower neighbors will be sampled, respectively. If the coordinates are not transformed, and the values of the interpolated PDF are interpolated naively (such that $f_{j,k}(s|\theta) = (1-\alpha) f_{j,k}(s|\theta^{(0)}) + \alpha f_{j,k}(s|\theta^{(1)})$), it may cause issues: for example, two normal distributions with different means would interpolate to create a bimodal distribution, instead of a normal distribution with a mean somewhere between the original two.

In calculating the transformed coordinates in Eq. 4.15, $\hat{\sigma}$ is used in place of σ to avoid divide-by-zero errors when σ is zero. However, this is not of great importance, as the interpolated PDF in that case will be replaced with an approximate, as detailed in Eq. 4.16 below. Likewise, $\hat{\sigma}^{(0)}$ and $\hat{\sigma}^{(1)}$ are used to avoid multiplying by zero, causing all of the transformed coordinates to sample at a single point repeatedly ($\mu^{(0)}$ or $\mu^{(1)}$, respectively). The result of this is a flat curve, and is obviously not a good interpolation between the two neighboring PDFs.

$$\frac{s-\mu}{\hat{\sigma}} = \frac{s^{(0)} - \mu^{(0)}}{\hat{\sigma}^{(0)}} = \frac{s^{(1)} - \mu^{(1)}}{\hat{\sigma}^{(1)}} \tag{4.15}$$

Lastly, the interpolated PDF, $f_{j,k}$ can be found using:

$$f_{j,k}(s|\theta) = \begin{cases} f_{j,k}^{\sigma=0}(s|\theta), & \text{if } \sigma = 0 \\ (1-\alpha) * \frac{ds^{(0)}}{ds} f_{j,k}^{\sigma=0}(s^{(0)}) + \alpha * \frac{ds^{(1)}}{ds} f_{j,k}^{(1)}(s^{(1)}), & \text{if } \sigma^{(0)} = 0, \sigma^{(1)} \neq 0 \\ (1-\alpha) * \frac{ds^{(0)}}{ds} f_{j,k}^{(0)}(s^{(0)}) + \alpha * \frac{ds^{(1)}}{ds} f_{j,k}^{\sigma=0}(s^{(1)}), & \text{if } \sigma^{(1)} = 0, \sigma^{(0)} \neq 0 \\ (1-\alpha) * \frac{ds^{(0)}}{ds} f_{j,k}^{(0)}(s^{(0)}) + \alpha * \frac{ds^{(1)}}{ds} f_{j,k}^{(1)}(s^{(1)}), & \text{otherwise} \end{cases} \tag{4.16}$$

where $f_{j,k}^{\sigma=0}(s|\theta)$ is the approximate stand-in for a PDF with zero-standard deviation defined in Eq. 4.10, $ds^{(0)}/ds$ is equal to $\sigma^{(0)}/\sigma$, and $ds^{(1)}/ds$ is equal to $\sigma^{(1)}/\sigma$.

It can be verified that the final interpolated PDF is a valid PDF (all values are non-negative and the integral over $s = [-\infty, \infty]$ is unity). It is clear that the values will be non-negative, as they will be the result of linear interpolation between the values of $f_{j,k}^{(0)}(s|\theta)$ and $f_{j,k}^{(1)}(s|\theta)$, which by definition must be non-negative. We can also show that the integral over state space is unity using the definition in 4.16 for PDFs with non-zero σ [9]. We use fact (a): $\int_{-\infty}^{\infty} s^{(i)} f^{(i)}(s^{(i)}) ds^{(i)} = \mu^{(i)}$. We drop the subscript on $f_{j,k}$ for simplicity:

$$\int_{-\infty}^{\infty} f(s) ds = (1-\alpha) * \frac{ds^{(0)}}{ds} \int_{-\infty}^{\infty} f^{(0)}(s^{(0)}) ds + \alpha * \frac{ds^{(1)}}{ds} \int_{-\infty}^{\infty} f^{(1)}(s^{(1)}) ds \tag{4.17}$$

$$= (1-\alpha) * \int_{-\infty}^{\infty} f^{(0)}(s^{(0)}) ds^{(0)} + \alpha * \int_{-\infty}^{\infty} f^{(1)}(s^{(1)}) ds^{(1)} \tag{4.18}$$

$$= 1 - \alpha + \alpha = 1 \tag{4.19}$$

Additionally, we can verify that the mean is equal to μ. First, we can solve for s from 4.15:

$$s = \frac{\hat{\sigma}}{\hat{\sigma}^{(0)}} s^{(0)} + \frac{\mu \hat{\sigma}^{(0)} - \mu^{(0)} \hat{\sigma}}{\hat{\sigma}^{(0)}} \tag{4.20}$$

$$= \frac{\hat{\sigma}}{\hat{\sigma}^{(1)}} s^{(1)} + \frac{\mu \hat{\sigma}^{(1)} - \mu^{(1)} \hat{\sigma}}{\hat{\sigma}^{(1)}} \tag{4.21}$$

Then, we verify the mean of the interpolated PDF, using fact (a) and fact (b): $\int_{-\infty}^{\infty} f^{(i)}(s^{(i)}) ds^{(i)} = 1$. In the case that either $\sigma^{(0)}$ or $\sigma^{(1)}$ is 0, the approximate PDF $f_{j,k}^{\sigma=0}$ will be used in place of $f^{(0)}$ or $f^{(1)}$ respectively, consistent with Eq. 4.16. Facts (a) and (b) are still true under this substitution.

$$
\begin{aligned}
\int_{-\infty}^{\infty} sf(s)ds &= (1-\alpha)*\frac{ds^{(0)}}{ds}\int_{-\infty}^{\infty} sf^{(0)}(s^{(0)})ds + \alpha*\frac{ds^{(1)}}{ds}\int_{-\infty}^{\infty} sf^{(1)}(s^{(1)})ds \\
&= (1-\alpha)*\int_{-\infty}^{\infty} sf^{(0)}(s^{(0)})ds^{(0)} + \alpha*\int_{-\infty}^{\infty} sf^{(1)}(s^{(1)})ds^{(1)} \\
&= (1-\alpha)*\left[\frac{\hat{\sigma}}{\hat{\sigma}^{(0)}}\int_{-\infty}^{\infty} s^{(0)}f^{(0)}(s^{(0)})ds^{(0)}\right. \\
&\left.+\frac{\mu\hat{\sigma}^{(0)}-\mu^{(0)}\hat{\sigma}}{\hat{\sigma}^{(0)}}*\int_{-\infty}^{\infty} f^{(0)}(s^{(0)})ds^{(0)}\right] \\
&+\alpha*\left[\frac{\hat{\sigma}}{\hat{\sigma}^{(1)}}\int_{-\infty}^{\infty} s^{(1)}f^{(1)}(s^{(1)})ds^{(1)}\right. \\
&\left.+\frac{\mu\hat{\sigma}^{(1)}-\mu^{(1)}\hat{\sigma}}{\hat{\sigma}^{(1)}}*\int_{-\infty}^{\infty} f^{(1)}(s^{(1)})ds^{(1)}\right] \\
&= (1-\alpha)*(\frac{\hat{\sigma}\mu^{(0)}+\mu\hat{\sigma}^{(0)}-\mu^{(0)}\hat{\sigma}}{\hat{\sigma}^{(0)}})+\alpha*(\frac{\hat{\sigma}\mu^{(1)}+\mu\hat{\sigma}^{(1)}-\mu^{(1)}\hat{\sigma}}{\hat{\sigma}^{(1)}}) \\
&= \mu
\end{aligned}
\tag{4.22}
$$

We can verify the variance of the interpolated PDF similarly. Again, the approximate PDF $f_{j,k}^{\sigma=0}$ will be used in place of $f^{(0)}$ or $f^{(1)}$ if either $\sigma^{(0)}$ or $\sigma^{(1)}$ is 0. Therefore, we can use facts (a), (b), and (c): $\int_{-\infty}^{\infty} s^{(i)^2}f^{(i)}(s^{(i)})ds^{(i)} = \mu^{(i)^2}+\hat{K}^{(i)}$.

$$
\int_{-\infty}^{\infty} s^2 f(s)ds = (1-\alpha)*\int_{-\infty}^{\infty} s^2 f^{(0)}(s^{(0)})ds^{(0)} + \alpha*\int_{-\infty}^{\infty} s^2 f^{(1)}(s^{(1)})ds^{(1)} \tag{4.23}
$$

We will examine the first integral term: $\int_{-\infty}^{\infty} s^2 f^{(0)}(s^{(0)})ds^{(0)}$. Recall that $\hat{\sigma}^2 = \hat{K}$.

$$
\begin{aligned}
\int_{-\infty}^{\infty} s^2 f^{(0)}(s^{(0)})ds^{(0)} &= (\mu^2+\frac{\mu^{(0)^2}\hat{\sigma}^2}{\hat{\sigma}^{(0)^2}}-\frac{2\mu^{(0)}\mu\hat{\sigma}}{\hat{\sigma}^{(0)}})\int_{-\infty}^{\infty} f^{(0)}(s^{(0)})ds^{(0)} \\
&+(-\frac{2\mu^{(0)}\hat{\sigma}^2}{\hat{\sigma}^{(0)^2}}+\frac{2\mu\hat{\sigma}}{\hat{\sigma}^{(0)}})\int_{-\infty}^{\infty} s^{(0)}f^{(0)}(s^{(0)})ds^{(0)} \\
&+\frac{\hat{\sigma}^2}{\hat{\sigma}^{(0)^2}}\int_{-\infty}^{\infty} s^{(0)^2}f^{(0)}(s^{(0)})ds^{(0)} \\
&= \mu^2+\frac{\mu^{(0)^2}\hat{\sigma}^2}{\hat{\sigma}^{(0)^2}}-\frac{2\mu^{(0)}\mu\hat{\sigma}}{\hat{\sigma}^{(0)}}-\frac{2\mu^{(0)^2}\hat{\sigma}^2}{\hat{\sigma}^{(0)^2}}+\frac{2\mu^{(0)}\mu\hat{\sigma}}{\hat{\sigma}^{(0)}}+\frac{\mu^{(0)^2}\hat{\sigma}^2}{\hat{\sigma}^{(0)^2}}+\frac{\hat{\sigma}^2 S^{(0)}}{\hat{\sigma}^{(0)^2}} \\
&= \mu^2+\frac{\hat{\sigma}^2 S^{(0)}}{\hat{\sigma}^{(0)^2}} = \mu^2+\hat{S}
\end{aligned}
\tag{4.24}
$$

It can be similarly shown that $\int_{-\infty}^{\infty} s^2 f^{(1)}(s^{(1)})ds^{(1)} = \mu^2+\hat{S}$. Therefore,

$$\int_{-\infty}^{\infty} s^2 f(s) ds = (1-\alpha) * (\mu^2 + \hat{S}) + \alpha * (\mu^2 + \hat{S})$$
$$= \mu^2 + \hat{S} \tag{4.25}$$

Eqs. 4.22 and 4.25 show that the interpolated PDF is valid, and that the mean and variance of the interpolated PDF are consistent with the values calculated in Eq. 4.12 and Eq. 4.13 respectively, despite using the approximate PDF $f_{j,k}^{\sigma=0}$ during interpolation for cases when $\sigma^{(i)} = 0$. However, practically, we can only take a finite number of discrete samples when interpolating, not infinite samples as reflected in the above calculations.

We determine the PDF over the range $s \in [0, N]$, where N is the number of agents in the specified sub-population. Then, we use Eq. 4.15 to determine the transformed values of coordinates to sample from the high and low PDFs. Then we are able to use Eq. 4.16 to calculate the values of the interpolated PDF at the chosen grid values. The last step is to re-normalize the PDF over the interval $[0, N]$ to have an integral of 1. Since we use *Gaussian* KDE, the PDF is non-zero over the entire domain, but this is not reflective of what is possible in our simulations, and is corrected through this re-normalization.

The final interpolated PDF is a set of discretely sampled points. For its use in the likelihood function, we linearly interpolate between these samples.

Two-parameter case interpolation procedure

The two-parameter case follows a similar procedure as the one-parameter case. Define: $\theta = [M, J]$, where M is the mobility parameter, and J is the jumping probability parameter. We first identify the neighboring parameters, $M^{(0)}$, $M^{(1)}$, $J^{(0)}$, and $J^{(1)}$. With the set of four corresponding PDFs, we can interpolate along one parameter in state space twice to reduce the set to two PDFs, then once more to get a final PDF.

More concretely, the interpolation procedure can be described:

1. First, we interpolate along the mobility parameter at the lower neighboring jumping parameter (interpolating between $f_{j,k}^{M_{(0)},J_{(0)}}(s|\theta)$ and $f_{j,k}^{M_{(1)},J_{(0)}}(s|\theta)$ to get $f_{j,k}^{J_{(0)}}(s|\theta)$).
2. Then, interpolate along the mobility parameter at the higher neighboring jumping parameter (interpolating between $f_{j,k}^{M_{(0)},J_{(1)}}(s|\theta)$ and $f_{j,k}^{M_{(1)},J_{(1)}}(s|\theta)$ to get $f_{j,k}^{J_{(1)}}(s|\theta)$).
3. Lastly, we can interpolate between $f_{j,k}^{J_{(1)}}(s|\theta)$ and $f_{j,k}^{J_{(0)}}(s|\theta)$ along the jumping parameter to get the final interpolated PDF, $f_{j,k}(s|\theta)$.

Though the interpolated PDF theoretically has an integral of unity over the state space domain when sampled continuously (Eq. 4.19), we are limited to a finite number of samples, for which this may not hold. Similarly, the K and μ values calculated in steps 1 and 2 are correct over an infinitely fine sampling mesh, but are approximations of the true means and variances of the interpolated PDFs obtained in step 1 and 2 due to the finite, discrete sampling used to construct them. To correct for these two approximations, we first re-normalize the interpolated PDFs obtained in step 1 and 2, then re-evaluate the means and variances of the interpolated PDFs using a trapezoid rule approximation (Eq. 4.26, Eq. 4.27). Near $x = 0$, the grid points x must be along a finer mesh than the grid points s along which $f_{j,k}^{J_{(0)}}(s|\theta)$ and $f_{j,k}^{J_{(1)}}(s|\theta)$ are sampled

in order to accurately assess the variance. To achieve this with minimal extra computation time, we transform a linear sequence of grid points using a power function with $q = 1.5$ (Eq. 4.28), and sample along the resulting values. This concentrates mesh density around 0, which is where thin spikes are likely to occur. As shown in Eqs. 4.26 and 4.27, we do not reevaluate the mean and variance values of interpolated PDFs with variance K of 0, as these values are already known precisely. We use $\hat{K}$ instead of K to define the new variance value.

$$\mu_{new} = \begin{cases} \mu & \text{if } K = 0 \\ \sum_{k=1}^{N} \frac{[x_{k-1} * f_{j,k}(x_{k-1}|\theta)] + [x_k * f_{j,k}(x_k|\theta)]}{2}(x_k - x_{k-1}) & \text{otherwise} \end{cases} \tag{4.26}$$

$$K_{new} = \begin{cases} \hat{K} & \text{if } K = 0 \\ \sum_{k=1}^{N} \frac{[(x_{k-1} - \mu_{new})^2 * f_{j,k}(x_{k-1}|\theta)] + [(x_k - \mu_{new})^2) * f_{j,k}(x_k|\theta)]}{2}(x_k - x_{k-1}) & \text{otherwise} \end{cases} \tag{4.27}$$

$$x_k = \begin{cases} -|x_k|^q, & \text{if } x_k \leq 0 \\ (x_k)^q, & \text{otherwise} \end{cases} \tag{4.28}$$

In step 3, the mean and variance values of the interpolated PDF will be calculated according to Eq. 4.12 and Eq. 4.13, where $\mu^{(0)}$ is equal to μ_{new} of $f_{j,k}^{J_{(0)}}(s|\theta)$ (PDF from step 1), $\mu^{(1)}$ is equal to μ_{new} of $f_{j,k}^{J_{(1)}}(s|\theta)$ (PDF from step 2), $K^{(0)}$ is equal to K_{new} of $f_{j,k}^{J_{(0)}}(s|\theta)$ (PDF from step 1), $K^{(1)}$ is equal to K_{new} of $f_{j,k}^{J_{(1)}}(s|\theta)$ (PDF from step 2). The rest of step 3 proceeds as described in the one-parameter case interpolation section.

Due to the process of transforming the coordinates, the range of points needed for interpolation can be outside the region that is theoretically possible (e.g., if there are only 100 agents in a sub-population, the there is only a non-zero probability for values in the range s = [0,100]). After the first two interpolations are performed (1 & 2 above), we have two PDFs that have been constructed through discrete sampling, and can be evaluated via interpolation. Since we then need to interpolate again between these two PDFs, we need to ensure that the initial discrete samples are taken over a wide enough range to avoid extrapolation, which could lead to negative values, and in turn, invalid PDF values. We do this by sampling along an evenly spaced grid that covers the full range of possible values, along with a margin on both sides. Additionally, we then set the outermost sampling points to be arbitrarily large in magnitude, under the assumption that the points in that region will already be very close to 0 (example of sampling points: s = [-1,000,000, -49, -48, -47...147, 148, 149, 1,000,000]).

Appendix B: AMCMC details

The proposal distribution is a Gaussian centered at the current chain value with covariance C_i. The algorithm has two phases: During the non-adaptive period (prior to ν iterations), C_i is adjusted if the fraction of (rejections since last scaled)/(total samples since last scaled) is greater than 0.95 or less than 0.05, in which case the proposal standard deviation is scaled down or up respectively by a chosen scale factor value. During the adaptive period, C_i is updated intermittently (every n_a iterations) using a recursive update shown below in Eqs. 4.29 and 4.30,

$$\text{cov}(\theta_0, ..., \theta_i) = \frac{i-1}{i}\text{cov}(\theta_0, ..., \theta_{i-1}) + \frac{i+1}{i^2}(\theta_i - \bar{\theta}_i)(\theta_i - \bar{\theta}_i)^T \tag{4.29}$$

$$C_i = \begin{cases} s_d \text{cov}(\theta_0, ..., \theta_{i-1}), & \text{if } i \geq \nu \text{ AND cov}(\theta_0, ..., \theta_{i-1}) \text{ is non-singular} \\ s_d \text{cov}(\theta_0, ..., \theta_{i-1}) + s_d \varepsilon_c I_d, & \text{if } i \geq \nu \end{cases} \tag{4.30}$$

where s_d is a parameter, I_d is the identity matrix with dimension d, and ε_c is a chosen small value.

We use $s_d = 2.4^2/d$ [16] and $\varepsilon_c = 1 \times 10^{-10}$. Initial covariance for the one-parameter case was $C_0 = 0.001$, and for the two-parameter case was $C_0 = 0.001 I_d$. The covariance is updated every $n_a = 100$ iterations during the adaptive period.

This page intentionally left blank

5. CONCLUSIONS

In this report, we develop a new detector for outbreaks caused by emergent/re-emergent pathogens, using the type of noisy case count data that is available in the early epoch of the outbreak. The detector is meant to provide an early warning of a change in the epidemiological dynamics that is usually the harbinger of a new wave of infections. The algorithm is demonstrated on COVID-19 data from the 33 counties of New Mexico and is tested by checking whether it can detect the Fall 2020 wave of COVID19 infections earlier than a conventional detector from Robert Koch Institute in Germany.

Conventional detectors are generally developed for endemic diseases and rely on long historical dataset to detect anomalous changes in the daily case counts obtained from surveillance activities. They do not use exogenous information e.g., knowledge of the incubation period of the disease, to perform the early detection; instead they rely on learning spatiotemporal patterns from the large surveillance datasets. In the early epoch of an outbreak with an *emergent* pathogen, large training datasets do not exist and conventional detectors do not perform well. Consequently early warning of an outbreak of an emergent pathogen remains elusive.

Our approach is based on the hypothesis that the spread-rate of a disease is a more robust monitoring variable for detecting changes in epidemiological dynamics compared to case count data. This is because the spread-rate of a disease is dependent on human mixing patterns and pathogen characteristics, neither of which change erratically day-to-day. The technical challenge lies in estimating the spread-rate (or infection rate) from noisy case count data. Further, since epidemiological actions like countermeasures are often taken at a local (e.g., county) level, the spread-rate has to be estimated locally. This is a challenge for remote, sparsely populated and often low income areas, with less than satisfactory epidemiological surveillance and reporting. Further, due to small populations, their data may also suffer from high-variance Poisson noise.

In this report we have developed a Bayesian method to estimate a spread-rate *field* defined over multiple areal units, the counties of NM in our case. The estimation method involves spatial statistics, specifically Gaussian Markov Random Fields, to learn and impose spatial correlations in the spread-rate field. The spread-rate in each areal unit is also allowed to vary in time, in a parametric fashion. This spatiotemporal estimation leads to a high-dimensional inverse problem, 136 parameters in total for NM.

We solve the inverse problem formulation over 3 adjacent NM counties - Bernalillo, Santa Fe and Valencia - using Markov Chain Monte Carlo (MCMC). We also solve each county independently. We find that the spread-rates and 2-week-ahead forecasts computed using the two methods are virtually identical, implying that the high-dimensional (16-dimensional) joint inversion could be solved just as well as the low-dimensional (6-dimensional) inversions for each county. We formulated an anomaly detection scheme, based on the forecasts generated by the learned spread-rate, to detect the Fall 2020 wave. The anomaly detector successfully detected the arrival Fall 2020 wave, and did so about a week before the RKI detector. In some cases, the RKI detector could not detect the Fall 2020 wave within a 2-week window from its arrival on September 15, 2020. Further, when tested on August 15, 2020, before the arrival of the Fall 2020 wave, our detector provided a true negative; in contrast, the RKI detector flagged a false positive for some

counties.

MCMC is not scalable, and in order to scale to all the NM counties, we developed a scalable, but approximate, method to estimate the spread-rate field. It is based on mean-field Variation Inference (VI), with a particular formulation to stabilize it against the low quality data obtained from about 10 desert counties with small populations. While the VI worked stably and obtained useful estimates of spread-rates, it under-estimated the uncertainty in the estimates. Consequently, when performing anomaly detection, the anomaly boundary (or threshold) had to be inflated with a factor that was calibrated to the results obtained in Bernalillo, Santa Fe and Valencia. This inflation factor was then applied to all NM counties and used to detect the arrival of the Fall 2020 wave. It was successful in most cases. However, when tested on August 15, before the arrival of the Fall 2020 wave, it resulted in false positives on most cases. This is almost certainly due to the simplistic manner in which our detector combines detected anomalies - three consecutive anomalous days are regarded as a definitive change in epidemiological dynamics. Further, having access to a *field* of anomalies allowed us to use Kulldorff's clustering algorithm with the inferences performed around September 15 to track the spread of the Fall 2020 wave through the Rio Grande Valley. The progress was evident with 4 days of data.

Finally, we investigated the calibration of agent-based disease models (ABMs) which are used to design epidemiological countermeasures that involve changes in the behavior of humans e.g., vaccinations, lock-downs etc. ABMs are stochastic models, which makes their calibration difficult, especially with high-variance Poisson-like noise which are dominant in the early epoch. We pursued two methods, one based on Approximate Bayesian Computations (ABC) and the other based on surrogate modeling and MCMC. For ABC, we designed a new scoring function that quantifies the match between observed data and model results. For the second method, we designed stochastic surrogate models and an approximate likelihood function that ignored temporal correlations in the data. The two methods were tested with synthetic data, and a 2-dimensional parameter estimation problem. The ABC method proved superior; the approximate likelihood function proved to be the cause of the failure of the MCMC method.

The most useful tangible product of this investigation is the VI scheme to infer a field from measurements combining, as it does, a random field model and a stabilized Evidence Lower Bound formulation. It will find multiple applications, ranging from high-dimensional calibration problems to estimation of model-form errors/corrections in many scientific and engineering problems.

Our detector needs to be enhanced to be useful. The enhancements are needed to compensate for VI's tendency to under-estimate the uncertainty in the spread-rate. A couple of enhancements come to mind. First, the arrival of a new wave of infection will result in anomalies that increase in degree over time - this temporal behavior is not exploited in the current detector which simply relies on consecutive anomalies over time. Secondly, in the absence of a wave of infections, anomalous case counts should resemble noise (for low quality data). Being able to characterize this noise and exploit it in the detector holds the most promise for early detection in sparsely populated areal units with poor epidemiological reporting.

BIBLIOGRAPHY

[1] Matthew Abueg, Robert Hinch, Neo Wu, Luyang Liu, William Probert, Austin Wu, Paul Eastham, Yusef Shafi, Matt Rosencrantz, Michael Dikovsky, Zhao Cheng, Anel Nurtay, Lucie Abeler-Dörner, David Bonsall, Michael V. McConnell, Shawn O'Banion, and Christophe Fraser. Modeling the effect of exposure notification and non-pharmaceutical interventions on COVID-19 transmission in Washington state. *npj Digital Medicine*, 4(1):49, March 2021.

[2] Mark A Beaumont, Wenyang Zhang, and David J Balding. Approximate Bayesian Computation in Population Genetics. *Genetics*, 162(4):2025–2035, December 2002.

[3] Julian Besag, Jeremy York, and Annie Mollié. Bayesian image restoration, with two applications in spatial statistics. *Annals of the institute of statistical mathematics*, 43:1–20, 1991.

[4] Nicky Best, Sylvia Richardson, and Andrew Thomson. A comparison of bayesian spatial models for disease mapping. *Statistical methods in medical research*, 14(1):35–59, 2005.

[5] Roger S. Bivand, Edzer Pebesma, and Virgilio Gomez-Rubio. *Modeling Areal Data*, chapter 9. Springer, New York, 2013.

[6] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[7] P. Blonigan, J. Ray, and C. Safta. Forecasting multi-wave epidemics through bayesian inference. *Archives of Computatationa Methods in Engineering*, 28:4169–4183, 2021.

[8] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

[9] Faruk H. Bursal. On interpolating between probability distributions. *Applied Mathematics and Computation*, 77(2):213–244, 1996.

[10] Eduardo DC Carvalho, Ronald Clark, Andrea Nicastro, and Paul HJ Kelly. Scalable uncertainty for computer vision with functional variational inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12003–12013, 2020.

[11] B. Debusschere, K. Sargsyan, C. Safta, and K. Chowdhary. The uncertainty quantification toolkit (uqtk). In R. Ghanem, D. Higdon, and H. Owhadi, editors, *Handbook of Uncertainty Quantification*, pages 1807–1827. Springer, 2017.

[12] John M. Drake, RajReni B. Kaul, Laura W. Alexander, Suzanne M. O'Regan, Andrew M. Kramer, J. Tomlin Pulliam, Matthew J. Ferrari, and Andrew W. Park. Ebola Cases and Health System Demand in Liberia. *PLOS Biology*, 13(1):e1002056, January 2015.

[13] M. Eichner. Transmission Potential of Smallpox: Estimates Based on Detailed Data from an Outbreak. *American Journal of Epidemiology*, 158(2):110–117, July 2003.

[14] N. Benjamin Erichson, Peng Zheng, and Sasha Aravkin. *sparsepca: Sparse Principal Component Analysis (SPCA)*, 2018. R package version 0.1.2.

[15] N. Benjamin Erichson, Peng Zheng, Krithika Manohar, Steven L. Brunton, J. Nathan Kutz, and Aleksandr Y. Aravkin. Sparse principal component analysis via variable projection. *SIAM Journal on Applied Mathematics*, 80(2):977–1002, 2020.

[16] Heikki Haario, Eero Saksman, and Johanna Tamminen. An adaptive metropolis algorithm. *Bernoulli*, pages 223–242, 2001.

[17] José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International conference on machine learning*, pages 1861–1869. PMLR, 2015.

[18] Robert Hinch, William J. M. Probert, Anel Nurtay, Michelle Kendall, Chris Wymant, Matthew Hall, Katrina Lythgoe, Ana Bulas Cruz, Lele Zhao, Andrea Stewart, Luca Ferretti, Daniel Montero, James Warren, Nicole Mather, Matthew Abueg, Neo Wu, Olivier Legat, Katie Bentley, Thomas Mead, Kelvin Van-Vuuren, Dylan Feldner-Busztin, Tommaso Ristori, Anthony Finkelstein, David G. Bonsall, Lucie Abeler-Dörner, and Christophe Fraser. OpenABM-Covid19—An agent-based model for non-pharmaceutical interventions against COVID-19 including contact tracing. *PLOS Computational Biology*, 17(7):e1009146, July 2021.

[19] Michael Hohle and Michaela Paul. Count data regression charts for the monitoring of surveillance time series. *Computational Statistics and Data Analysis*, 52(9):4357–4368, 2008.

[20] Shoukang Hu, Xurong Xie, Shansong Liu, Jianwei Yu, Zi Ye, Mengzhe Geng, Xunying Liu, and Helen Meng. Bayesian learning of lf-mmi trained time delay neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1514–1529, 2021.

[21] Elizabeth Hunter, Brian Mac Namee, and John D. Kelleher. A taxonomy for agent-based models in human infectious disease epidemiology. *Journal of Artificial Societies and Social Simulation*, 20(3):2, 2017.

[22] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[23] Cliff C. Kerr, Robyn M. Stuart, Dina Mistry, Romesh G. Abeysuriya, Katherine Rosenfeld, Gregory R. Hart, Rafael C. Núñez, Jamie A. Cohen, Prashanth Selvaraj, Brittany Hagedorn, Lauren George, Michał Jastrzębski, Amanda S. Izzo, Greer Fowler, Anna Palmer, Dominic Delport, Nick Scott, Sherrie L. Kelly, Caroline S. Bennette, Bradley G. Wagner, Stewart T. Chang, Assaf P. Oron, Edward A. Wenger, Jasmina Panovska-Griffiths, Michael Famulare, and Daniel J. Klein. Covasim: An agent-based model of COVID-19 dynamics and interventions. *PLOS Computational Biology*, 17(7):e1009149, July 2021.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[25] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[26] Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. Automatic differentiation variational inference. *Journal of machine learning research*, 2017.

[27] Martin Kulldorff and Neville Nagarwalla. Spatial disease clusters: Detection and inference. *Statistics in Medicine*, 14(8):799–810, 1995.

[28] Stephen A. Lauer, Kyra H. Grantz, Qifang Bi, Forrest K. Jones, Qulu Zheng, Hannah R. Meredith, Andrew S. Azman, Nicholas G. Reich, and Justin Lessler. The Incubation Period of Coronavirus Disease 2019 (COVID-19) From Publicly Reported Confirmed Cases: Estimation and Application. *Annals of Internal Medicine*, 2020.

[29] J. Legrand, R. F. Grais, P. Y. Boelle, A. J. Valleron, and A. Flahault. Understanding the dynamics of Ebola epidemics. *Epidemiology and Infection*, 135(4):610–621, May 2007.

[30] Percy Liang, Slav Petrov, Michael I Jordan, and Dan Klein. The infinite pcfg using hierarchical dirichlet processes. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 688–697, 2007.

[31] Luyang Liu, Sharad Vikram, Junpeng Lao, Xue Ben, Alexander D'Amour, Shawn O'Banion, Mark Sandler, Rif A. Saurous, and Matthew D. Hoffman. Estimating the Changing Infection Rate of COVID-19 Using Bayesian Models of Mobility. preprint, Epidemiology, August 2020.

[32] Gongning Luo, Suyu Dong, Wei Wang, Kuanquan Wang, Shaodong Cao, Clara Tam, Henggui Zhang, Joanne Howey, Pavlo Ohorodnyk, and Shuo Li. Commensal correlation network between segmentation and direct area estimation for bi-ventricle quantification. *Medical image analysis*, 59:101591, 2020.

[33] Jean-Michel Marin, Pierre Pudlo, Christian P. Robert, and Robin J. Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, November 2012.

[34] Emma S. McBryde, Michael T. Meehan, Oyelola A. Adegboye, Adeshina I. Adekunle, Jamie M. Caldwell, Anton Pak, Diana P. Rojas, Bridget M. Williams, and James M. Trauer. Role of modelling in COVID-19 policy development. *Paediatric Respiratory Reviews*, 35:57–60, September 2020.

[35] Xuhui Meng, Hessam Babaee, and George Em Karniadakis. Multi-fidelity bayesian neural networks: Algorithms and applications. *Journal of Computational Physics*, 438:110361, 2021.

[36] Stefano Merler, Marco Ajelli, Laura Fumanelli, Marcelo F C Gomes, Ana Pastore y Piontti, Luca Rossi, Dennis L Chao, Ira M Longini, M Elizabeth Halloran, and Alessandro Vespignani. Spatiotemporal spread of the 2014 outbreak of Ebola virus disease in Liberia and the effectiveness of non-pharmaceutical interventions: a computational modelling analysis. *The Lancet Infectious Diseases*, 15(2):204–211, February 2015.

[37] Onur Ozdemir, Benjamin Woodward, and Andrew A Berlin. Propagating uncertainty in multi-stage bayesian convolutional neural networks with application to pulmonary nodule detection. *arXiv preprint arXiv:1712.00497*, 2017.

[38] Jonathan Ozik, Justin M Wozniak, Nicholson Collier, Charles M Macal, and Mickaël Binois. A population data-driven workflow for COVID-19 modeling and learning. *The International Journal of High Performance Computing Applications*, 35(5):483–499, September 2021.

[39] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

[40] Łukasz Rączkowski, Marcin Możejko, Joanna Zambonelli, and Ewa Szczurek. Ara: accurate, reliable and active histopathological image classification framework with bayesian deep learning. *Scientific reports*, 9(1):14347, 2019.

[41] C. Safta, J. Ray, and K. Sargsyan. Characterization of partially observed epidemics through bayesian inference: Application to covid-19. *Computational Mechanics*, 66:1109–1129, 2020.

[42] Lyndsay Shand, Alex Foss, Adah Zhang, J. Derek Tucker, and Gabriel Huerta. A statistical model for the spread of sars-cov-2 in new mexico. Technical Report SAND2020-10080, Sandia National Laboraories, Livermore, CA, September 2020.

[43] B.W. Silverman. Density estimation for statistics and data analysis. *Monographs on Statistics and Applied Probability*, 26, 1986.

[44] Jessica E. Stockdale, Theodore Kypraios, and Philip D. O'Neill. Modelling and Bayesian analysis of the Abakaliki smallpox data. *Epidemics*, 19:13–23, June 2017.

[45] Mikael Sunnåker, Alberto Giovanni Busetto, Elina Numminen, Jukka Corander, Matthieu Foll, and Christophe Dessimoz. Approximate Bayesian Computation. *PLoS Computational Biology*, 9(1):e1002803, January 2013.

[46] Lu Tang, Yiwang Zhou, Lili Wang, Soumik Purkayastha, Leyao Zhang, Jie He, Fei Wang, and Peter X-K Song. A review of multi-compartment infectious disease models. *International Statistical Review*, 88(2):462–513, 2020.

[47] Ryutaro Tanno, Daniel E Worrall, Aurobrata Ghosh, Enrico Kaden, Stamatios N Sotiropoulos, Antonio Criminisi, and Daniel C Alexander. Bayesian image quality transfer with cnns: exploring uncertainty in dmri super-resolution. In *Medical Image Computing and Computer Assisted Intervention- MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part I 20*, pages 611–619. Springer, 2017.

[48] Srinivasan Venkatramanan, Bryan Lewis, Jiangzhuo Chen, Dave Higdon, Anil Vullikanti, and Madhav Marathe. Using data-driven agent-based models for forecasting emerging infectious diseases. *Epidemics*, 22:43–49, March 2018.

[49] Lance Waller and Brad Carlin. Disease mapping. In Alan E. Gelfand, Peter J. Diggle, Montserrat Fuentes, and Peter Guttorp, editors, *Handbook of Spatial Statistics*, chapter 14. Chapman & Hall / CRC Press, 2010.

[50] Joshua S. Weitz and Jonathan Dushoff. Modeling Post-death Transmission of Ebola: Challenges for Inference and Opportunities for Control. *Scientific Reports*, 5(1):8751, March 2015.

[51] Liu Yang, Xuhui Meng, and George Em Karniadakis. B-pinns: Bayesian physics-informed neural

networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.

[52] T. I. Zohdi. An agent-based computational framework for simulation of global pandemic and social response on planet x. *Computational Mechanics*, 66(5):1195–1209, 2020.

DISTRIBUTION

Hardcopy—Internal

Number of Copies	Name	Org.	Mailstop
1	D. Chavez, LDRD Office	1911	0359

Email—External (encrypt for OUO)

Name	Company Email Address	Company Name
Maya Horii	mjhorii@berkeley.edu	Univ. of California, Berkeley
Aidan Gould	atgould@berkeley.edu	Univ. of California, Berkeley

Email—Internal (encrypt for OUO)

Name	Org.	Sandia Email Address
Wyatt Bridgman	08734	whbridg@sandia.gov
Technical Library	01177	libref@sandia.gov

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.