JOURNAL OF APPLIED STATISTICS
https://doi.org/10.1080/02664763.2025.2462969

Taylor & Francis
Taylor & Francis Group

APPLICATION NOTE

BSTPP: a python package for Bayesian spatiotemporal point processes

Isaac Manring [a], Honglang Wang [a], George Mohler [b] and Xenia Miscouridou [c,d]

[a]Department of Mathematics, Indiana University Indianapolis, Indianapolis, USA; [b]Department of Computer Science, Boston College, Boston, USA; [c]Department of Mathematics and Statistics, University of Cyprus, Nicosia, Cyprus; [d]Department of Mathematics, Imperial College London, London, UK

ABSTRACT

Spatiotemporal point process models have a rich history of effectively modeling event data in space and time. However, they are sometimes neglected due to the difficulty of implementing them. There is a lack of packages with the ability to perform inference for these models, particularly in python. Thus we present BSTPP a python package for Bayesian inference on spatiotemporal point processes. It offers three different kinds of models: space-time separable Log Gaussian Cox, Hawkes, and Cox Hawkes. Users may employ the pre-defined trigger parameterizations for the Hawkes models, or they may implement their own trigger functions with the extendable Trigger module. For the Cox models, posterior inference on the Gaussian processes is sped up with a pre-trained Variational Auto Encoder (VAE). The package includes a new flexible pre-trained VAE. We validate the model through simulation studies and then explore it by applying it to shooting data in Chicago.

ARTICLE HISTORY
Received 15 June 2024
Accepted 29 January 2025

KEYWORDS
Bayesian; spatiotemporal point process; Log Gaussian Cox; Hawkes; Cox Hawkes; variational auto encoder

MATHEMATICS SUBJECT CLASSIFICATION
60G55

1. Introduction

Spatio-temporal event data analysis is a critical field of study focused on understanding patterns, dependencies, and dynamics of events distributed across space and time [5,8,11,29]. This type of analysis combines spatial and temporal dimensions to explore phenomena such as earthquake occurrences, crime incidents, disease outbreaks, and social interactions. By modeling the interactions between where and when events occur, researchers can uncover underlying processes driving the events, predict future occurrences, and inform decision-making in fields like urban planning, public health, and environmental management [5,8]. Spatio-temporal event data presents unique challenges due to its complexity, requiring specialized methodologies and computational tools that integrate spatial statistics, temporal modeling, and machine learning [29]. These approaches aim to provide actionable insights into complex systems by capturing intricate spatio-temporal dependencies and offering explanations for observed phenomena [2,24].

CONTACT Isaac Manring isaacamanring@gmail.com

Supplemental data for this article can be accessed online at http://dx.doi.org/10.1080/02664763.2025.2462969.

However, these methods are not always employed because of their complexity and lack of practical implementations. For example, Spatial Event Aggregation (SEA) methods are commonly used to analyze conflicts in space and time. Such methods, in which events are aggregated on spatial units and then regression is performed on the resulting counts, may lead to erroneous conclusions because of the simplicity of the assumed mechanism [19,31]. One of the main problems identified with the methodology is that it ignores spatial diffusion and correlation of violent events, and it also does not account for non-uniform sizes of spatial units. Self-exciting point processes address both of these concerns, but are more difficult to implement due to a lack of available software packages and knowledge on the topic.

The availability of Python packages for spatio-temporal point process analysis remains significantly limited, reflecting a gap in the current ecosystem of data science tools. While several libraries such as tick and hawkeslib offer functionalities for temporal point processes, they lack the capability to model spatial dependencies [1]. Similarly, spatial analytics libraries like pySal and its pointpat module focus exclusively on spatial point patterns, leaving the temporal dimension unaddressed [30]. This lack of integrated tools poses challenges for researchers and practitioners aiming to analyze complex phenomena where both spatial and temporal interactions are critical, such as the spread of diseases, urban crime, or ecological dynamics. The absence of dedicated Python libraries that seamlessly handle spatio-temporal point processes limits the ability to develop, test, and apply advanced models in this field, necessitating either custom implementations or reliance on less accessible software platforms. Addressing this gap would significantly enhance the accessibility and scalability of spatio-temporal point process analysis across diverse application domains.

There are more R packages for spatiotemporal point data. The R package stpp is useful for spatiotemporal point process analysis, but it does not do inference for any models [10]. stpphawkes does Bayesian inference but only for temporal processes with spatial marks [36]. The package starve models spatiotemporal point data using a nearest-neighbor Gaussian process, but does not model self-excitation [18].The R package stopp can fit spatiotemporal Log Gaussian Cox Processes and a particular spatiotemporal Hawkes process known as ETAS [7]. stlnpp provides first and second order summary statistics and intensity estimates for spatiotemporal point processes on linear networks [25]. The most similar package to BSTPP is an R package called stelfi [15]. It has the capability to fit the ETAS model with Gaussian random field background and a Log Gaussian Cox Process with covariates and marks. However, the self-exciting model does not allow covariates and the trigger function is strictly set as exponential in time and Gaussian in space. As will be seen later, the trigger function can make a big difference in the quality of the fit. Further, it performs maximum likelihood estimation rather than Bayesian inference.

Because of the scarcity of software packages, it can be tempting to go the easier route and perform spatially aggregated Poisson regression. BSTPP[1] fills this gap by providing an interface for Bayesian inference. Now available on PyPi, it has many advantages, including

- Easy to use interface for inference and simulation on three different models: Hawkes, space-time separable Log Gaussian Cox, and Cox Hawkes.
- Two different methods of posterior sampling (MCMC and SVI for speed).
- High quality plots and model interpretation.
- Spatial covariates for the Cox Hawkes model (a novel addition to this model).

- User-defined trigger functions.
- Ability to use geospatial boundaries or simply set boundaries via an array.
- Powerful pre-trained VAE for fast inference.

BSTPP is built on top of numpyro [3,27] for posterior sampling, GeoPandas [16] for accommodation of spatial data, and matplotlib [14] for visualization.

The paper is organized as follows: first we describe the mathematical details of the three models covered in BSTPP, then we show its success by a simulation study, and finally analyze its application to a dataset involving shootings in Chicago.

2. Models

Point process models describe and predict events occurring in a continuum [35]. The events are idealized as points in that continuum. In this paper, we are interested in the space-time continuum, so the points are space-time points. A spatio-temporal point process is a mathematical model used to describe events that occur randomly in both space and time. Formally, it is defined as a random set of points $\{(s_i, t_i) : i \in \mathbb{N}\}$, where $s_i \in \mathbb{R}^d$ represents the spatial location of the ith event in a d-dimensional space, and $t_i \in \mathbb{R}^+$ represents its occurrence time. The process is typically described by its conditional intensity function $\lambda(s, t \mid \mathcal{H}_t)$, which quantifies the instantaneous rate of events occurring at location s and time t, given the history of events up to time t, denoted as $\mathcal{H}_t$ [5,8].

Mathematically, the conditional intensity function is expressed as:

$$\lambda(s, t \mid \mathcal{H}_t) = \lim_{\Delta s, \Delta t \to 0} \frac{\mathbb{P}(\text{An event occurs in } [s, s + \Delta s) \times [t, t + \Delta t) \mid \mathcal{H}_t)}{\Delta s \cdot \Delta t}.$$

where s is a given spatial point, Δs is an infinitesimal change in space, t is a given temporal point, Δt is an infinitesimal change in time, $\mathbb{P}(\cdot \mid \mathcal{H}_t)$ is the conditional probability given the historical events $\mathcal{H}_t$. $\mathcal{H}_t$ is only included if the point process is self-exciting, meaning that previous event occurrences impact future event occurrences. We will omit the $\mathcal{H}_t$ for the rest of the paper for notational simplicity. See [29] for a more full explanation of self-exciting spatiotemporal point processes.

The simplest case is the homogeneous Poisson process, where $\lambda(s, t)$ is constant, implying that events occur independently and uniformly over space and time. Extensions include *inhomogeneous Poisson processes*, where $\lambda(s, t)$ varies with s and t, and *self-exciting processes* like the Hawkes process, where the occurrence of one event increases the likelihood of future events in its spatio-temporal vicinity [2,29].

Applications of spatio-temporal point processes include seismic modeling (e.g. ETAS models) [26], crime pattern analysis [24], and the spread of infectious diseases [21]. These models enable the estimation of event likelihood, identification of clusters, and prediction of future occurrences based on historical data, providing invaluable tools for both theoretical research and practical decision-making.

BSTPP is capable of doing inference using three different kinds of point process models: Log Gaussian Cox processes, Hawkes processes, and Cox Hawkes processes. We also included a discussion of SAE models for comparison even though it is not included in the package.

2.1. Log Gaussian Cox process

The Log Gaussian Cox Process is a doubly stochastic model [9]. The spatiotemporal data is generated by an inhomogeneous Poisson process with intensity $\lambda(s,t)$. For a full explanation of Poisson Processes see [35, Chapter 2]. λ itself, however, is stochastic. It is determined by the exponential of a realization of a Gaussian process, $f_{st}(s,t)$. The intensity of the process is shown in the following equation,

$$\lambda(s,t) = \exp\left(f_{st}(s,t)\right). \tag{1}$$

In the particular model used in this package, the spatial and temporal parts of the Gaussian process are assumed to be independent and hence we make f_{st} separable in space and time. Often researchers are interested in the effects of covariates on the likelihood of an event to occur. For example, we would like to answer questions like, 'Does high median income in a community area in Chicago make it less likely for a shooting to occur?' To answer these kinds of questions and also give additional explanatory power to the model, we included spatial covariates in the model with a similar setup to regression.

$$\lambda(s,t) = \exp\left(f_s(s) + f_t(t) + X^{\mathsf{T}}(s)w + a_0\right). \tag{2}$$

Here f_t and f_s are zero mean Gaussian processes of one and two dimensions respectively,[2] $X(s)$ is the spatial covariate vector of dimension p, w is a vector of learnable weights of length p that models the linear effect of the spatial covariates on the intensity, while a_0 is a constant bias term (intercept) that captures the magnitude of the intensity independent of time and location.

In practice, the Gaussian processes are approximated by piecewise constant Gaussian random variables on a computational grid because Gaussian processes are theoretically infinite-dimensional. We refer the reader to [28, Chapter 2] for details on Gaussian processes and their properties. For the spatial dimension, this is a two-dimensional computational grid, while the temporal dimension only needs a one-dimensional vector.

2.2. Hawkes process

A spatiotemporal Hawkes process is a self-exciting point process which is fully defined by its intensity which admits the following form,

$$\lambda(s,t) = \mu(s,t) + \sum_{i:t_i<t} g(s - s_i, t - t_i). \tag{3}$$

where s_i is the spatial point for the ith observation, t_i is the temporal point for the ith observation, and μ and g are functions. μ is called the background function and g is called the trigger function. The interpretation of the intensity form is that events arise either from the background or from the triggering kernel. The initial event comes from the background. Subsequent events are generated from either the background or from the triggering kernel. Background events are called immigrant events whereas triggered events are called offspring [29]. The names arise from a common analogy in which, a Hawkes process describes the population of a country. Background points correspond to immigrants, while triggered points correspond to the descendants of immigrants. The expected number of offspring generated by an individual is called the reproduction rate. It is important to note that a reproduction rate greater than 1 would result in uncontrolled growth. Restricting the reproduction rate to be below 1 would ensure non-explosivity of the process. The default trigger function for the package is parameterized as follows,

$$g(s,t) = \alpha f(t;\beta)\varphi(s;\sigma). \tag{4}$$

where α is the reproduction rate, f is the exponential probability density function with rate β, and φ is the Gaussian probability density function with mean 0 and standard deviation σ. Thus the time between parent and triggered events is exponentially distributed, while the triggered point is normally distributed in space centered around the parent point.

BSTPP allows other trigger parameterizations also. The power law temporal decay is included in the package: $f(t;\beta,\gamma) = \beta\gamma^{\beta}(\gamma + t)^{-\beta-1}$, where β and γ can range from 0 to ∞. The package also includes the ability for the user to define their own trigger functions by extending the Trigger module. At this time, all user-defined trigger functions must be separable in space and time. The general form is $g(s,t) = \alpha f(t;\theta_1)h(s;\theta_2)$, where θ_1 and θ_2 are the parameters for the temporal and spatial components of the trigger intensity, respectively.

For the background rate, we use a spatial regression similar to the one in the previous section.

$$\mu(s,t) = X^{\mathsf{T}}(s)w + a_0. \tag{5}$$

2.3. Cox Hawkes

The final model is a combination of the previous two models: the Hawkes model with Log Gaussian Cox as the background process as initially introduced by [22].

$$\lambda(s,t) = \exp\left(a_0 + X^{\mathsf{T}}(s)w + f_s(s) + f_t(t)\right) + \sum_{i:t_i<t} \alpha f(t-t_i;\beta)\varphi(s-s_i;\sigma). \tag{6}$$

Again, the full version of the model includes spatial covariate regression. The Hawkes part models the triggering of events, while the Cox background models the clustering of events. This is the most powerful model in BSTPP.

To illustrate this model we will use the example data of shootings in Chicago. Some areas are more likely to have shootings because of various factors such as gang presence and population density. This is the clustering effect and can be modeled by the Log Gaussian Cox process with spatial covariates. Retaliatory shootings on the other hand are more likely to occur shortly after and near a recent shooting. This is the triggering effect modeled by the Hawkes process. Neither of these models separately can account for both effects, which is the motivation for including the Cox Hawkes model.

2.4. Spatial event aggregation model

Although spatial event aggregation (SEA) regression is not a part of BSTPP, it is discussed for comparison because of its popularity. The SEA regression model is a statistical framework used for analyzing and modeling aggregated event data over a spatial domain. This type of model is particularly useful in scenarios where the spatial locations of events are observed, but the events are aggregated into regions (e.g. grid cells, administrative units) rather than being recorded as precise point locations.

SEA models the relationship between the aggregated counts of events and a set of explanatory variables (covariates), such as demographic, environmental, or economic factors, that vary across the spatial domain. SEA regression typically uses a generalized linear model (GLM) framework. For count data, a Poisson regression or Negative Binomial regression is often employed. The expected event count is linked to the covariates through a log-linear or similar link function.

Spatially aggregated Poisson regression may be regarded as a point process model. It is an inhomogeneous piece-wise constant Poisson process. This method ignores all spatial correlation between the spatial regions. It also ignores any self-excitation present in the data and assumes temporal homogeneity. If T is the length of the time over which the data were observed, then the spatiotemporal intensity for the SEA model can be written as,

$$\lambda(s,t) = \exp\left(X^{\intercal}(s)w + a_0\right)/T. \tag{7}$$

3. Bayesian inference

The user may define the prior for each parameter by using numpyro distributions. Priors are specified at model instantiation. For example,

```
model = Hawkes_Model(data['events_2022'],
                     data['boundaries'], cox_background=True,
                     alpha = dist.Beta(20,60),
                     beta = dist.HalfNormal(2.0),
                     sigmax_2 = dist.HalfNormal(0.25),
                     a_0 = dist.Normal(1,10)
                    )
```

sets the priors for α, β, σ^2, and a_0. There is a default prior for w: $w \sim N(\mathbf{0}, \mathbf{1})$. As with any Hawkes process, inference can be computationally complex. For n observed data points over the spatial region of A with model parameters θ, the log likelihood of a point process model is given by,

$$\ell(\theta) = \sum_{i=1}^{n} \log\left(\lambda\left(s_i, t_i; \theta\right)\right) - \int_0^T \int_A \lambda\left(s, t; \theta\right)\, \mathrm{d}s\, \mathrm{d}t. \tag{8}$$

For a Hawkes model, evaluating the likelihood is typically an $O(n^2)$ operation because evaluating the intensity function at any given point is $O(n)$ due to the summation and the intensity is evaluated n times. A Cox model has cubic complexity due to matrix inversions.

In order to speed up inference for the Log Gaussian Cox and Cox Hawkes models, we employ pre-trained Variational Auto-Encoders (VAE) for the prior on the Gaussian processes (f_s and f_t) (see [33] for more details on the method as well as [23]). Following [22], we used a 25×25 computational grid for the spatial process and a length 50 computational vector for the temporal process.

Semenova *et al.* [33] provided a pre-trained VAE to approximate a Gaussian process prior. The spatial Gaussian process was trained to encode hyperpriors on length and scale for a Gaussian kernel for the covariance (the mean is set to zero). These hyperpriors were not expressive enough, however. The hyperprior for the scale parameter, which determines how flat the Gaussian process will be, was Log Normal with a mean of 0 and a variance of 0.1. To check whether this is reasonable, we ran a kernel density estimate of the spatial intensity (ignoring time) for Chicago shootings with bandwidth chosen by Silverman's rule of thumb [34] for simplicity. For more sophisticated methods of bandwidth selection for point process learning, please refer to a nice recent work [6]. This showed the log intensity had a range of approximately 11. The given hyperprior for scale simply could not account for this amount of spatial variance in the intensity. Further, the hyperprior for the length parameter ($l \sim InverseGamma(4, 1)$) was not expressive enough: it would oversmooth the intensity function causing the background intensity to be under-fit. The result was a very high α for the Cox Hawkes model, meaning that the background could not be fully captured by the VAE and that some of the inhomogeneity in the intensity was falsely attributed to self-excitation. To remedy this, we created a larger VAE with more expressive hyperpriors: $l \sim InverseGamma(15, 1)$ and $\sigma^2 \sim LogNormal(2, 0.5)$. The pretrained model is included in the package. If a user desires to train their own VAE, a training script is also included.

In our implementation, evaluating the log likelihood is $O(n^2)$ in memory. It is important to keep this in mind when modeling large datasets.

Posterior inference is performed by numpyro. We offer two kinds of inference for the model. MCMC sampling is performed by the state of the art Hamiltonian sampler NUTS [13]. The second kind of posterior inference, Stochastic Variational Inference or SVI, is an approximation. SVI proposes a distribution, called the guide, and then optimizes the parameters of the guide to match the true posterior as well as possible [12]. The advantage is that solving this optimization problem is much faster than actually doing MCMC sampling. We chose a multivariate Normal distribution as the guide. In our experience, once SVI is fully converged for these models, the results are almost indistinguishable from MCMC sampling. In order to stay consistent with MCMC sampling, we draw samples from the guide after training. These samples are then used in the posterior plots.

4. Simulation study

In this section we discuss the simulation capabilities of BSTPP and use them to test the most powerful model in BSTPP: Cox Hawkes. We validate the methodology through the comparison of posterior results from simulated data and known parameters. Then we demonstrate the usefulness of the model by comparing the posterior sampled data with the actual data.

For the methodology validation, we simulated a Cox Hawkes process with spatial covariates. The parameters were set as follows: reproduction rate at $\alpha = 0.25$, temporal exponential decay parameter at $\beta = 2.0$, and spatial trigger variance parameter at $\sigma^2 = 0.0025$. The three spatial covariates were randomly generated from a standard Normal distribution on a 10×10 grid and three corresponding weights were also generated from a Normal distribution with standard deviation of 0.3. We also randomly generated the latent variables according to a standard Normal distribution to feed into the VAEs for the spatial and temporal Gaussian processes. 1349 events were generated in total.

We then did inference on the data using SVI. The results are shown in Figures 1, 2, 3, and 4 for comparison between the true values and the posterior distributions of those parameters. The posterior distributions of the trigger function parameters were very close

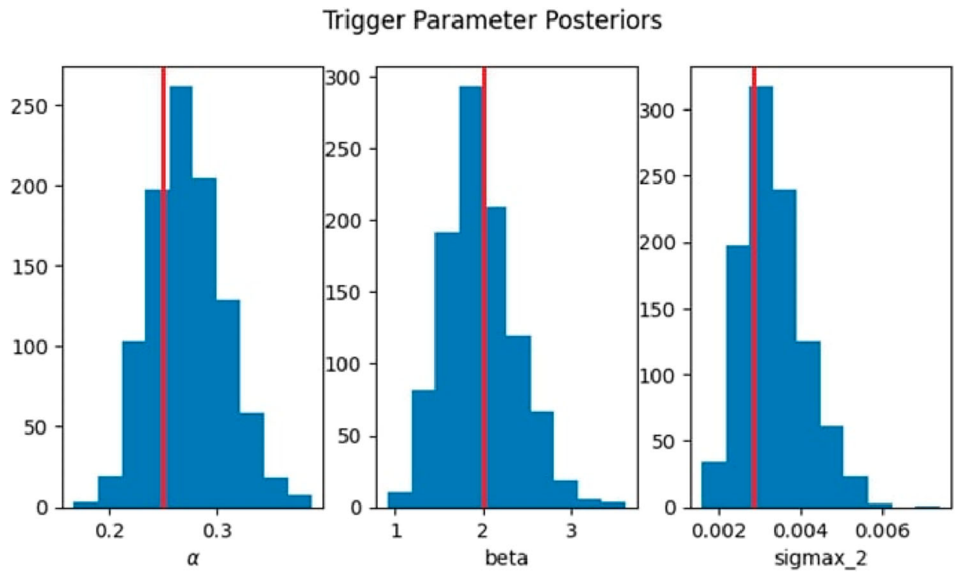


Figure 1. Posteriors for trigger parameters for the Cox Hawkes simulation with covariates. The red lines represent the true parameter values.

to centered on the true value (Figure 1). In addition, the temporal background intensity matched the true temporal background (Figure 3). The same can be said for the spatial background (Figure 4). Lastly, the spatial covariate weights, while not 100% correct every time always showed the correct direction (Figure 2). Furthermore, the true values were included within the bounds of the posterior distributions each time.

All models are wrong, but some are useful. To check the usefulness of the Cox Hawkes model, we generated data from the posterior mean of the fit on the Chicago shooting data (see details in the next section about real data fitting using the model) to see if it looked similar to the observed data (compare Figure 5 with Figure 6(a) with actual spatial data).

Note that BSTPP enables simulation for all models in its purview. While the Gaussian processes in the Cox and Cox Hawkes models are theoretically continuous, in practice they are piece-wise constant grids. Therefore simulation involves simulating the number of points occurring on the space-time region of interest (Poisson according to the intensity on the region) and then generating those points uniformly on the region. Self-exciting points are then generated using the immigration interpretation of the Hawkes process. In any of the models, the command to simulate a realization is model.simulate(). If no parameters are given for the simulation, the parameters for the simulation are set to the posterior mean (this assumes inference has been done).

5. Illustration using Chicago shootings data

We further examined the package by applying it to reported shootings in Chicago in 2022.[3] We used demographic information for the community areas in Chicago as spatial covariates.[4] We will illustrate the basic fitting features, model metrics, trigger function options, spatial covariates, and finally the visualization capabilities of the package.

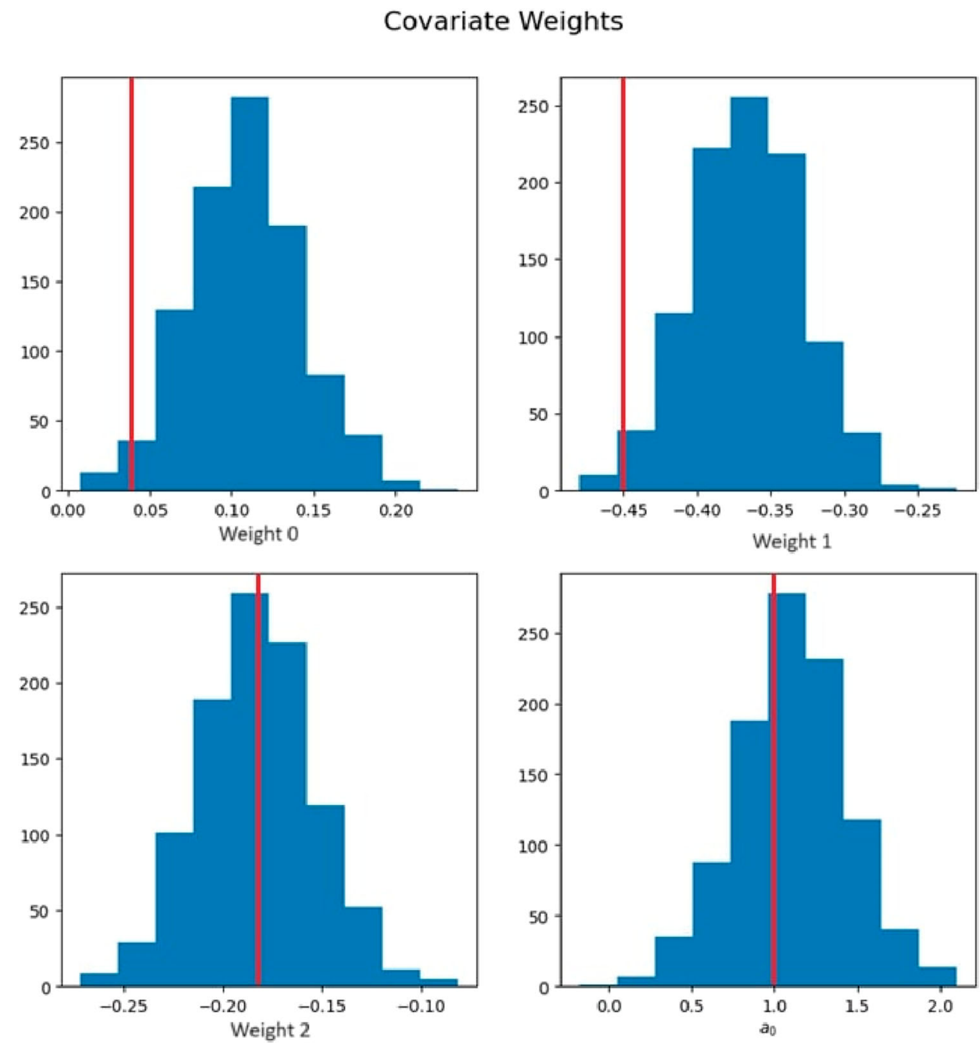
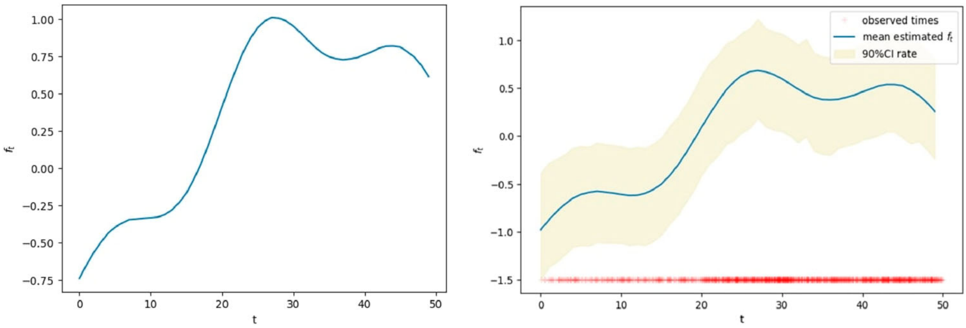


Figure 2. Posteriors for covariate weights for the Cox Hawkes simulation with covariates. The red lines represent the true parameter values.

5.1. Basic model fitting

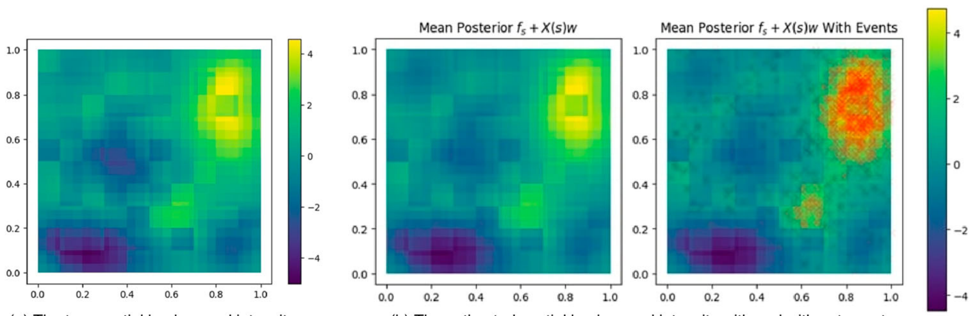
To sample the posterior using numpyro's NUTS sampler, we provided the function model.run_mcmc(). There is also the ability to use model.run_svi() using numpyro's SVI optimization with a multivariate Normal distribution as a guide. Details on the optional parameters of these functions can be seen in the API documentation and numpyro documentation.

A simple script running the model on Chicago shooting data is shown at the end of this session. The event data (contained in data['events_2022']) is a pandas DataFrame with location (X and Y) and time (T) columns. The boundaries data (data['boundaries'] is a GeoPandas GeoDataFrame, which contains the boundaries of the city of Chicago. The



(a) True background temporal function ft (b) Estimated background temporal function ft

Figure 3. Temporal background intensity for the Cox Hawkes simulation. The true function $f_t(t)$ is shown in the left plot. Right plot shows the mean mean posterior predictive for $f_t(t)$ in blue and the 90% credible interval in the yellow shaded area, with true time stamps of the events on the x-axis. (a) True background temporal function f_t and (b) Estimated background temporal function f_t.



(a) The true spatial background intensity (b) The estimated spatial background intensity with and with out events overlayed

Figure 4. This figure shows the spatial background intensity for the Cox Hawkes simulation with spatial covariates. The true and estimated and spatial intensities are almost indistinguishable. (a) The true spatial background intensity and (b) The estimated spatial background intensity with and with out events overlayed.

covariate data (data['covariates'] is also a GeoDataFrame and includes all columns specified in column_names. The rest of the arguments in the Hawkes_Model are priors for the various parameters defined using numpyro distributions.

```
data = load_Chicago_Shootings()
column_names = ['UNEMP_DENS','MEDINC','MED_HV','assoc_plus','VACANT_DEN',
        'VAC_HU_pct','HCUND20K_L','POP_DENS','CT_SP_WCHI']
model = Hawkes_Model(data['events_2022'],
                        data['boundaries'],True,
                        spatial_cov=data['covariates'],cov_names = column_names,
                        a_0=dist.Normal(1,10), alpha = dist.Beta(20,60),
                        beta=dist.HalfNormal(2.0)
                       )
model.run_svi(lr=0.02,num_steps=10000)
```

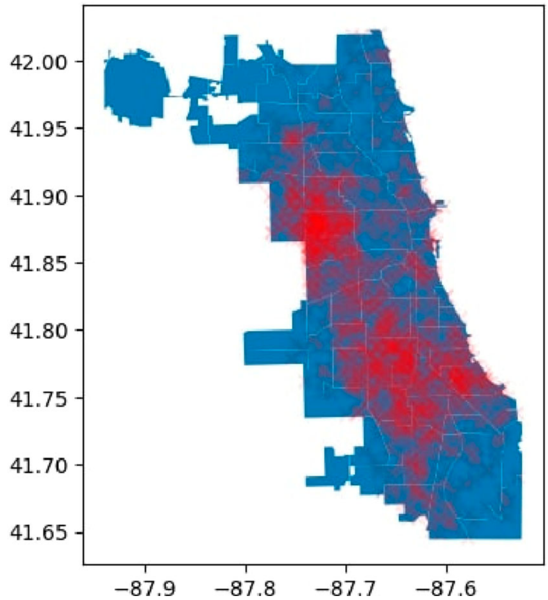
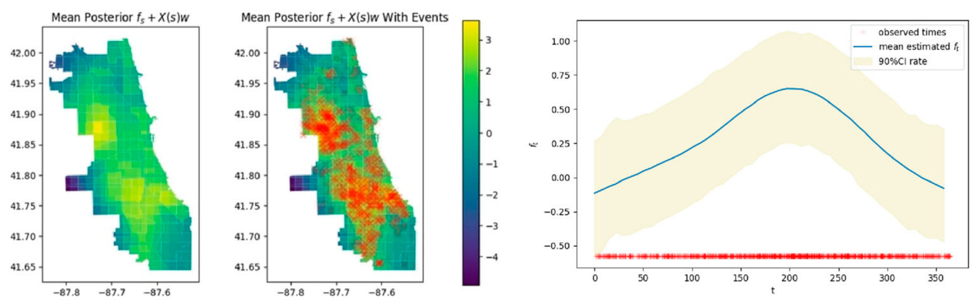


Figure 5. Simulation of Chicago shooting data from the posterior mean. The red markers are the simulated locations of the gunfire. The simulation looks reasonably similar to the actual data, so there are no red flags from this check.



(a) Spatial background intensity for Chicago shooting reports with actual event locations plotted on the right

(b) Temporal background intensity with 90% credible interval for Chicago shooting data. t is the day of year in 2022

Figure 6. Cox Hawkes background intensity visualized. (a) Spatial background intensity for Chicago shooting reports with actual event locations plotted on the right and (b) Temporal background intensity with 90% credible interval for Chicago shooting data. *t* is the day of year in 2022.

5.2. Model metrics

Comparing models can be difficult where temporal inhomogeneity is modeled. The temporal Gaussian process cannot be trivially extended outside the study time without additional assumptions. Further, cross-validation is not possible due to the self-excitation, which relies on past events. Because of this, our default is to rely on internal validation. We chose expected AIC. For notational simplicity, let θ denote the parameters of the model, and k

Table 1. Model performance on 2023 shooting data.

	Cox Hawkes	Hawkes	Cox	SEA regression
Log Expected Likelihood	7510.5	7446.2	7226.5	5213.3

be the number of parameters, then

$$\text{Expected AIC} = E_\theta\left[2k - 2\ell(\hat{\theta})\right]. \tag{9}$$

Since we cannot compute this expectation exactly, we estimate it by Monte Carlo methods according to the following equation where S is the number of samples drawn and θ^s is a particular sample,

$$\text{Expected AIC} \approx \frac{1}{S}\sum_{s=1}^{S} 2k - 2\ell(\theta^s). \tag{10}$$

You can get the expected AIC by calling model.expected_AIC().

If the temporal process is assumed to have a seasonal pattern to it, the learned Gaussian process may be used to apply to next cyclic period for testing. This is true in the case of Chicago shootings data. We assume that the temporal Gaussian process represents seasonal changes in shooting frequency. We tested our models on the next year of shootings in Chicago (2023). To evaluate their predictive performance we used the metric Log Expected Likelihood. Let y denote the training data, and $\tilde{y}$ denote the test data.

$$\text{Log Expected Likelihood} = \log\left(\int p\left(\tilde{y} \mid \theta\right) p\left(\theta \mid y\right) \mathrm{d}\theta\right) \tag{11}$$

In a similar manner to expected AIC, we estimate Log Expected Likelihood by a sample mean,

$$\text{Log Expected Likelihood} \approx \log\left(\frac{1}{S}\sum_{s=1}^{S} p\left(\tilde{y} \mid \theta^s\right)\right) \tag{12}$$

This can be calculated by model.log_expected_likelihood(data['events_2023']). The results are shown in Table 1. The Cox Hawkes model performed the best due to its expressiveness. The SEA model had very limited flexibility and therefore performed poorly.

5.3. *Trigger function*

The form of the trigger function impacts the modeling ability of a Hawkes model significantly. Because of this it is important to allow flexibility in its form. The BSTPP package provides some basic trigger functions, but also allows users to define their own by extending the Trigger class. They are required to implement the following class methods when extending the Trigger class: compute_trigger which simply computes the trigger function, compute_integral which computes the definite integral of the trigger given bounds, and get_par_names returns the names of the parameters for the trigger. Users may have as many parameters as necessary so long as they set priors for them in the model constructor and provide a list of their names. Code for the trigger computations must use jax.numpy. Spatial

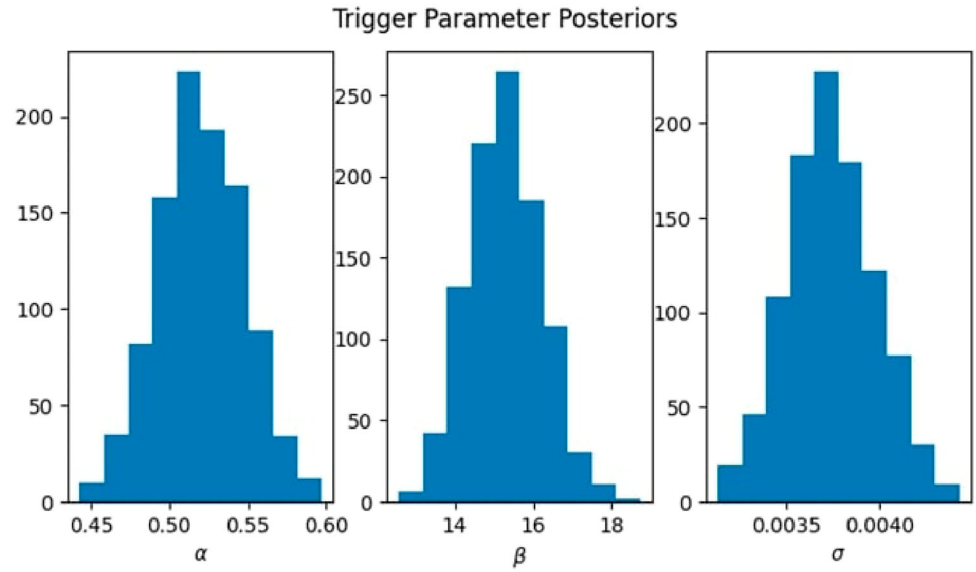


Figure 7. Trigger parameter posterior distributions for Cox Hawkes model on Chicago shooting data.

and temporal parts of the trigger function must be separable and also must be well defined probability density functions. To illustrate a user defined trigger function, we coded class spatial_double_exp for the spatial trigger function for the Chicago shootings data. This trigger mechanism significantly improved model performance. The Log Expected Likelihood increased to 7807.6. Additionally, the reproduction rate decreased to approximately 0.15, which is very similar to results in [4] for the city of Chicago.

BSTPP provides several plotting capabilities for the trigger function. To plot the posteriors for the trigger parameters and get a tabular summary of the outputs use, model.plot_trigger_posterior(). This will provide a histogram (Figure 7) or a trace plot. model.plot_trigger_time_decay() will visualize the decay of the trigger with time by plotting several curves sampled from the posterior temporal trigger parameters. model.plot_prop_excitation() will plot a histogram of the posterior of the proportion of the intensity due to self-excitation.

5.4. Spatial covariates

BSTPP also has plotting capabilities for spatial covariates. model.cov_weight_post_summary() works similarly to model.plot_trigger_posterior() (see Figure 8). These weights indicate how a covariate influences the likelihood of events occurring.

Covariate weights vary significantly depending on whether spatially correlated errors are part of the model. Because of this, SAE models cannot be relied upon when spatially correlated errors are an appropriate assumption. Table 2 summarizes the covariate weights for several different models. The SEA model is most similar to the Hawkes model, because neither assumes any spatially correlated error in the regression. The Cox Hawkes and Cox models have more similar regression coefficients. The Gaussian processes seem to have a regularizing effect on the regression. There are some highly correlated variables (median

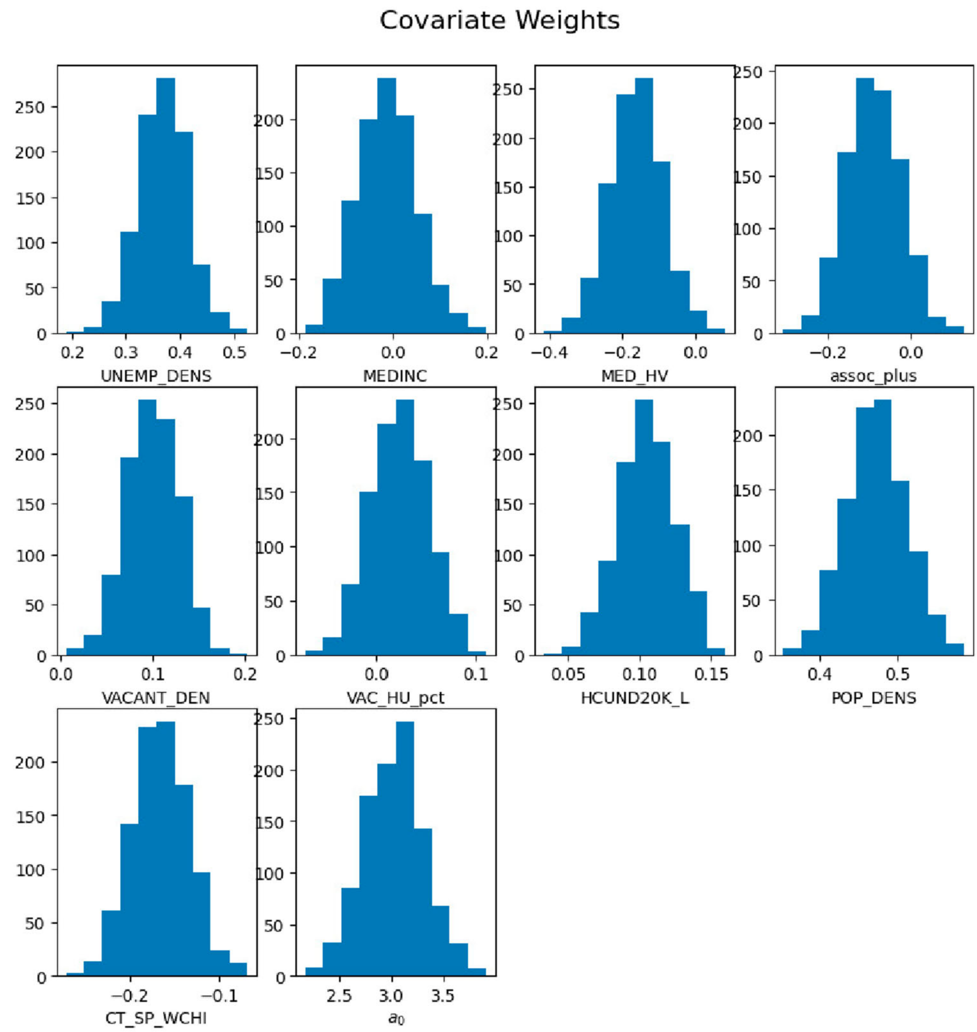


Figure 8. Covariate weight posterior distributions for Cox Hawkes model on Chicago shooting data with the given covariate names as titles. The covariates correspond to the covariates in the table in order.

income and median house value, and house and land vacancy) that have opposite signs of weight values in the Hawkes and SEA regression models, but for the most part the same sign with lower magnitude in the Cox and Cox Hawkes models (Table 2).

We tried the model with and without spatial covariates for our model with the best trigger function (see Section 5.3) and compared the performance with the test data (see Section 5.2). The model with spatial covariates performed better: 7807.6 to 7476.5 log expected likelihood on 2023 data, and −19,106.5 to −18,368.5 expected AIC for 2022 data.

5.5. Temporal and spatial visualization

model.plot_temporal() will plot the posterior mean of the temporal Gaussian process and the 90% confidence interval. Figure 6(b) clearly displays a higher intensity in the summer

Table 2. Posterior mean of covariate weights to the second decimal place.

	Cox Hawkes	Hawkes	Cox	SEA regression
Unemployment density	0.31	0.40	0.38	0.29
Median income	0.0	−0.38	0.01	−0.44
Median house value	−0.08	0.46	−0.17	0.45
Percentage of population with associates degrees or higher	−0.13	−0.27	−0.09	−0.26
Vacant land density	0.11	−0.10	0.09	−0.05
Percent of housing units vacant	0.09	0.47	0.03	0.41
Households with less than 20% of income allocated to housing	0.10	0.07	0.12	0.09
Population density	0.49	0.39	0.46	0.37
Single parent with child homes density	−0.15	−0.06	−0.15	0.04

months. This is in accordance with the well established fact that the violent crime rate is higher in hotter weather [17,20]. A model with constant temporal intensity (like the SEA model), cannot capture this insight.

model.plot_spatial() will plot the posterior mean of the spatial background. It also overlays the plot with observed data (see Figure 6(a)).

6. Conclusion

Spatiotemporal point process models are a powerful family of models, boasting much more flexibility than the commonly used SEA method, to explain the occurrence of events in space and time. However, their complexity makes it difficult and time consuming to work with, in the absence of software libraries designed for this purpose. BSTPP provides a flexible user-friendly interface for working with these models. The expressive pre-trained VAE included in the package makes inference much faster.

Even so, there are several features which are worth considering in future work. Currently, spatial and temporal parts of the trigger function must be separable. Also, changing the hyperpriors on the Gaussian processes requires retraining the VAEs, which could be improved by using a CVAE [32]. Additionally, because shootings and crimes in general often occur on the street or near the street, it may be interesting to consider the shootings in Chicago as occurring on a linear network (roads in this case). This would allow us to bring in additional information about the street and increase the spatial accuracy. Future research could apply and extend the R package stlnpp for the Chicago shootings dataset [25].

Notes

1. Code is hosted at https://github.com/imanring/BSTPP
2. Note that it is straightforward to consider a three dimensional (or higher) spatial Gaussian process if desired.
3. https://data.cityofchicago.org/Public-Safety/Chicago-Shootings/fsku-dr7m
4. https://datahub.cmap.illinois.gov/maps/2a0b0316dc2c4ecfa40a171c635503f8/about

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research was supported by NSF grants SCC-2125319, DMS-2212928, and ATD-2124313 and AFOSR MURI grant FA9550-22-1-0380; U.S. Department of Defense.

ORCID

Isaac Manring http://orcid.org/0009-0003-8294-6528

References

[1] E. Bacry, M. Bompaire, S. Gaïffas, and S. Poulsen, *Tick: A python library for statistical learning, with a particular emphasis on time-dependent modelling* (2018).

[2] E. Bacry, I. Mastromatteo, and J.F. Muzy, *Hawkes processes in finance*, Market Microstructure Liquidity 1 (2015), p. 1550005.

[3] E. Bingham, J.P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P.A. Szerlip, P. Horsfall, and N.D. Goodman, *Pyro: Deep universal probabilistic programming*, J. Mach. Learn. Res. 20 (2019), pp. 28:1–28:6. Available at http://jmlr.org/papers/v20/18-403.html.

[4] P.J. Brantingham, J. Carter, J. MacDonald, C. Melde, and G. Mohler, *Is the recent surge in violence in American cities due to contagion?* J. Crim. Justice 76 (2021), p. 101848. Available at https://www.sciencedirect.com/science/article/pii/S0047235221000684.

[5] N. Cressie and C.K. Wikle, *Statistics for Spatio-Temporal Data*, John Wiley & Sons, Boboken, NJ, 2011.

[6] O. Cronie, M. Moradi, and C.A. Biscio, *A cross-validation-based statistical theory for point processes*, Biometrika 111 (2024), pp. 625–641.

[7] N. D'Angelo and G. Adelfio, *stopp: Methods for spatio-temporal point pattern analysis, simulation, model fitting, diagnostics, and local analyses* (2023). Available at https://arxiv.org/abs/2302.13684.

[8] P.J. Diggle, *Statistical Analysis of Spatial and Spatiotemporal Point Patterns*, 3rd ed., CRC Press Taylor and Francis Group, New York, 2014.

[9] P.J. Diggle, P. Moraga, B. Rowlingson, and B.M. Taylor, *Spatial and spatio-temporal Log-Gaussian Cox processes: Extending the geostatistical paradigm*, Stat. Sci. 28, pp. 542–563 (2013). Available at http://dx.doi.org/10.1214/13-STS441.

[10] E. Gabriel, B.S. Rowlingson, and P.J. Diggle, *stpp: An R package for plotting, simulating and analyzing spatio-temporal point patterns*, J. Stat. Softw. 53 (2013), pp. 1–29. Available at https://www.jstatsoft.org/index.php/jss/article/view/v053i02.

[11] J.A. González, F.J. Rodríguez-Cortés, O. Cronie, and J. Mateu, *Spatio-temporal point process statistics: A review*, Spat. Stat. 18 (2016), pp. 505–544.

[12] M.D. Hoffman, D.M. Blei, C. Wang, and J. Paisley, *Stochastic variational inference*, J. Mach. Learn. Res. 14 (2013), pp. 1303–1347.

[13] M.D. Homan and A. Gelman, *The no-u-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo*, J. Mach. Learn. Res. 15 (2014), pp. 1593–1623.

[14] J.D. Hunter, *Matplotlib: A 2D graphics environment*, Comput. Sci. Eng. 9 (2007), pp. 90–95.

[15] C.M. Jones-Todd and A.B.M. van Helsdingen, *stelfi: An R package for fitting Hawkes and Log-Gaussian Cox point process models*, Ecol. Evol. 14 (2024), p. e11005. Available at https://onlinelibrary.wiley.com/doi/abs/10.1002/ece3.11005, e11005 ECE-2023-10-01795.R1.

[16] K. Jordahl, J.V. den Bossche, M. Fleischmann, J. Wasserman, J. McBride, J. Gerard, J. Tratner, M. Perry, A.G. Badaracco, C. Farmer, G.A. Hjelle, A.D. Snow, M. Cochran, S. Gillies, L. Culbertson, M. Bartos, N. Eubank, A. Bilogur, S. Rey, C. Ren, D. Arribas-Bel, L. Wasser, L.J. Wolf, M. Journois, J. Wilson, A. Greenhall, C. Holdgraf, Filipe, and F. Leblanc, *geopandas/geopandas: v0.8.1* (2020). Available at https://doi.org/10.5281/zenodo.3946761.

[17] J.L. Lauritsen and N. White, *Seasonal patterns in criminal victimization trends* (2014).
[18] E. Lawler, C. Field, and J. Mills Flemming, *starve: An R package for spatio-temporal analysis of research survey data using nearest-neighbour Gaussian processes*, Methods Ecol. Evol. 14 (2023), pp. 817–830. Available at https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.14053.
[19] C. Luo and A.L. Yuille, *Grouped spatial-temporal aggregation for efficient action recognition*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, Korea (South), 2019, pp. 5512–5521.
[20] S. Mavroudeas, J. Papastamatelou, A. Unger, G. Tsouvelas, G. Konstantakopoulos, and O. Giotakos, *(Summer) time for murder: Is there a link between increased temperature and homicides?* Dialogues Clin. Neurosci. Mental Health 1 (2018), pp. 128–137.
[21] S. Meyer and L. Held, *A space-time conditional intensity model for infectious disease spread*, Biometrics68 (2012), pp. 548–557.
[22] X. Miscouridou, S. Bhatt, G. Mohler, S. Flaxman, and S. Mishra, *Cox-Hawkes: Doubly stochastic spatiotemporal poisson processes*, Trans. Mach. Learn. Res. 2023 (2023). Available at https://openreview.net/forum?id=xzCDD9i4IZ.
[23] S. Mishra, S. Flaxman, T. Berah, M. Pakkanen, H. Zhu, and S. Bhatt, *π vae: Encoding stochastic process priors with variational autoencoders*, Stat. Comput. 32 (2022), pp. 96.
[24] G. Mohler, *Marked point process hotspot maps for homicide and gun crime prediction in Chicago*, Int. J. Forecast. 30 (2014), pp. 491–497.
[25] M. Moradi, O. Cronie, and J. Mateu, *stlnpp: Spatio-temporal analysis of point patterns on linear networks* (2024).
[26] Y. Ogata, *Space-time point-process models for earthquake occurrences*, Ann. Inst. Stat. Math. 50 (1998), pp. 379–402.
[27] D. Phan, N. Pradhan, and M. Jankowiak, *Composable effects for flexible and accelerated probabilistic programming in NumPyro*, preprint (2019). Available at arXiv:1912.11554.
[28] C.E. Rasmussen and C.K.I. Williams, *Regression*, in *Gaussian Processes for Machine Learning*, chap. 2, The MIT Press, Cambridge, 2005.
[29] A. Reinhart, *A review of self-exciting spatio-temporal point processes and their applications*, Stat. Sci. 33 (2018), pp. 299–318. Available at http://dx.doi.org/10.1214/17-STS629.
[30] S.J. Rey and L. Anselin, *PySAL: A python library of spatial analytical methods*, Rev. Reg. Stud. 37 (2007), pp. 5–27.
[31] S. Schutte and C. Kelling, *A Monte Carlo analysis of false inference in spatial conflict event studies*, PLoS ONE 17 (2022), pp. 1–22. Available at https://doi.org/10.1371/journal.pone.0266010.
[32] E. Semenova, P. Verma, M. Cairney-Leeming, A. Solin, S. Bhatt, and S. Flaxman, *Priorcvae: Scalable mcmc parameter inference with bayesian deep generative modelling* (2023).
[33] E. Semenova, Y. Xu, A. Howes, T. Rashid, S. Bhatt, S. Mishra, and S. Flaxman, *Priorvae: Encoding spatial priors with variational autoencoders for small-area estimation*, J. R. Soc. Interface 19 (2022), pp. 20220094. Available at http://dx.doi.org/10.1098/rsif.2022.0094.
[34] B.W. Silverman, *Density Estimation for Statistics and Data Analysis*, Routledge, London, 1998.
[35] D.L. Snyder and M.I. Miller, *Random Point Processes in Time and Space*, 2nd ed., Springer-Verlag, New York, 1991.
[36] D. Tucker, L. Shand, and J.R. Lewis, *Handling missing data in self-exciting point process models*, Spat. Stat. 29 (2019), pp. 160–176. Available at https://www.sciencedirect.com/science/article/pii/S2211675318301726.

Appendices

Appendix 1. Chicago shootings model results

The results for the Hawkes and Log Gaussian Cox models are shown in this appendix. The Hawkes model assumes a constant temporal background/immigrant process, so there is no plotting functionality enabled for that.

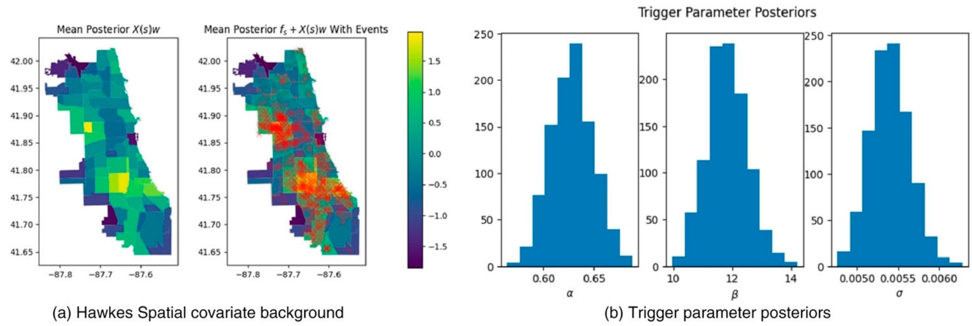


Figure A1. Hawkes model results for Chicago shooting data. (a) Hawkes Spatial covariate background and (b) Trigger parameter posteriors.

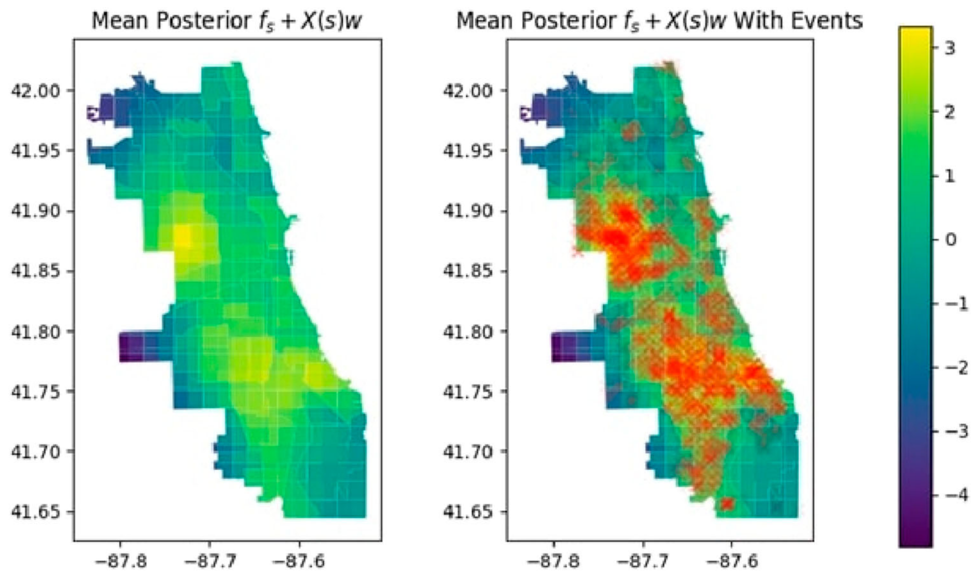


Figure A2. Log Gaussian Cox Process spatial intensity for Chicago shootings data. Events are added to the right plot.

It is clear to see that the Hawkes model provides a much less smooth background intensity: compare Figures A1(a) and A2. The high reproduction rate helps account for temporal and spatial inhomogeneity.

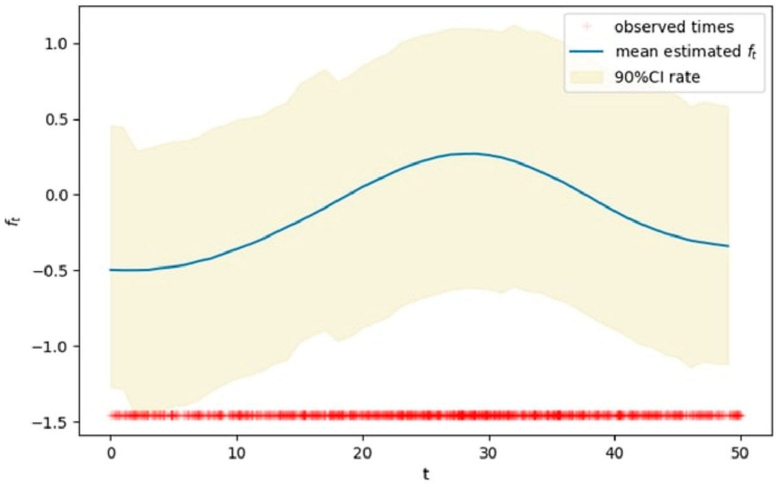


Figure A3. Log Gaussian Cox Process temporal intensity with 90% credible interval. t is the day of the year in 2022. Event times are plotted in red at the bottom of the figure.

Appendix 2. Demo code

```
from bstpp.main import LGCP_Model, Hawkes_Model, load_Chicago_Shootings
import numpyro.distributions as dist
import numpy as np
import matplotlib.pyplot as plt
#set seed for reproducability
np.random.seed(16)
data = load_Chicago_Shootings()
column_names = ['UNEMP_DENS','MEDINC','MED_HV','assoc_plus','VACANT_DEN',
      'VAC_HU_pct','HCUND20K_L','POP_DENS','CT_SP_WCHI']
# Will produce a warning because we are using latitude, longitude coordinates instead
# of geometrically projected coordinates. The area that we are looking at is small
# enough that it doesn't matter if we are using the geometric projection though.
model = Hawkes_Model(data['events_2022'],#spatiotemporal points
                  data['boundaries'],#Chicago boundaries
                  365,#Time frame (1 yr)
                  True,#use Cox as background
                  spatial_cov=data['covariates'],#spatial covariate matrix
                  cov_names = column_names,#columns to use from covariates
                  a_0=dist.Normal(1,10), alpha = dist.Beta(20,60),#set priors
                  beta=dist.HalfNormal(2.0),sigmax_2=dist.HalfNormal(0.25)
                 )
# trains the model using Stochastic Variational Inference
# !! IMPORTANT !! run_svi first trains the variational distribution and then
# samples the variational distribution to be consistent with mcmc methods.
model.run_svi(lr=0.02,num_steps=15000)
print("Log_Expected_Likelihood:",
   model.log_expected_likelihood(data['events_2023']))
print("Expected_AIC:",model.expected_AIC())
model.plot_prop_excitation()
plt.show()
model.plot_trigger_posterior(trace=False)
plt.show()
model.plot_trigger_time_decay()
plt.show()
model.plot_spatial(include_cov=True)
plt.show()
model.cov_weight_post_summary(trace=True)
```

```
plt.show()
from bstpp.trigger import Trigger
import jax.numpy as jnp
# Defining a custom trigger function to illustrate the process
class spatial_double_exp(Trigger):
   def compute_trigger(self,pars,dif_mat):
       return jnp.exp(-jnp.abs(dif_mat).sum(axis=0)/
             pars['Lambda'])/(2*pars['Lambda'])**2
   def compute_integral(self,pars,limits):
      x_limits = limits[0] #shape [2,n]
      y_limits = limits[1] #shape [2,n]
      x_int = 1-0.5*jnp.exp(-jnp.abs(x_limits[0]/pars['Lambda'])) - \
          0.5*jnp.exp(-jnp.abs(x_limits[1]/pars['Lambda']))
      y_int = 1-0.5*jnp.exp(-jnp.abs(y_limits[0]/pars['Lambda'])) - \
          0.5*jnp.exp(-jnp.abs(y_limits[1]/pars['Lambda']))
      return x_int*y_int
   def simulate_trigger(self,pars):
      return np.random.laplace(size=2,scale=pars['Lambda'])
   def get_par_names(self):
      return ['Lambda']
# same as before except with new trigger!
model = Hawkes_Model(data['events_2022'],#spatiotemporal points
                 data['boundaries'],#Chicago boundaries
                 365,#Time frame (1 yr)
                 True,#use Cox as background
                 spatial_cov=data['covariates'],#spatial covariate matrix
                 cov_names = column_names,#columns to use from covariates
                 a_0=dist.Normal(1,10), alpha = dist.Beta(20,60),#set priors
                 beta=dist.HalfNormal(2.0),Lambda=dist.HalfNormal(0.5),
                 spatial_trig=spatial_double_exp
                )
model.run_svi(lr=0.02,num_steps=15000)
lel = model.log_expected_likelihood(data['events_2023'])
print(f"Log␣Expected␣Likelihood:␣{lel}")
eaic = model.expected_AIC()
print(f"Expected␣AIC␣{eaic}")
model.plot_trigger_posterior(trace=True)
plt.show()
model.plot_trigger_time_decay()
plt.show()
model.plot_spatial(include_cov=True)
plt.show()
model.plot_temporal()
plt.show()
model.cov_weight_post_summary()
plt.show()
```