```
from bstpp.main import LGCP_Model, Hawkes_Model,
load_Chicago_Shootings, load_Boko_Haram
import numpyro.distributions as dist
import numpy as np
np.random.seed(16)
```

Chicago Shootings Dataset

Reported shootings from the city of Chicago for the years 2022 and 2023. Data provided includes:

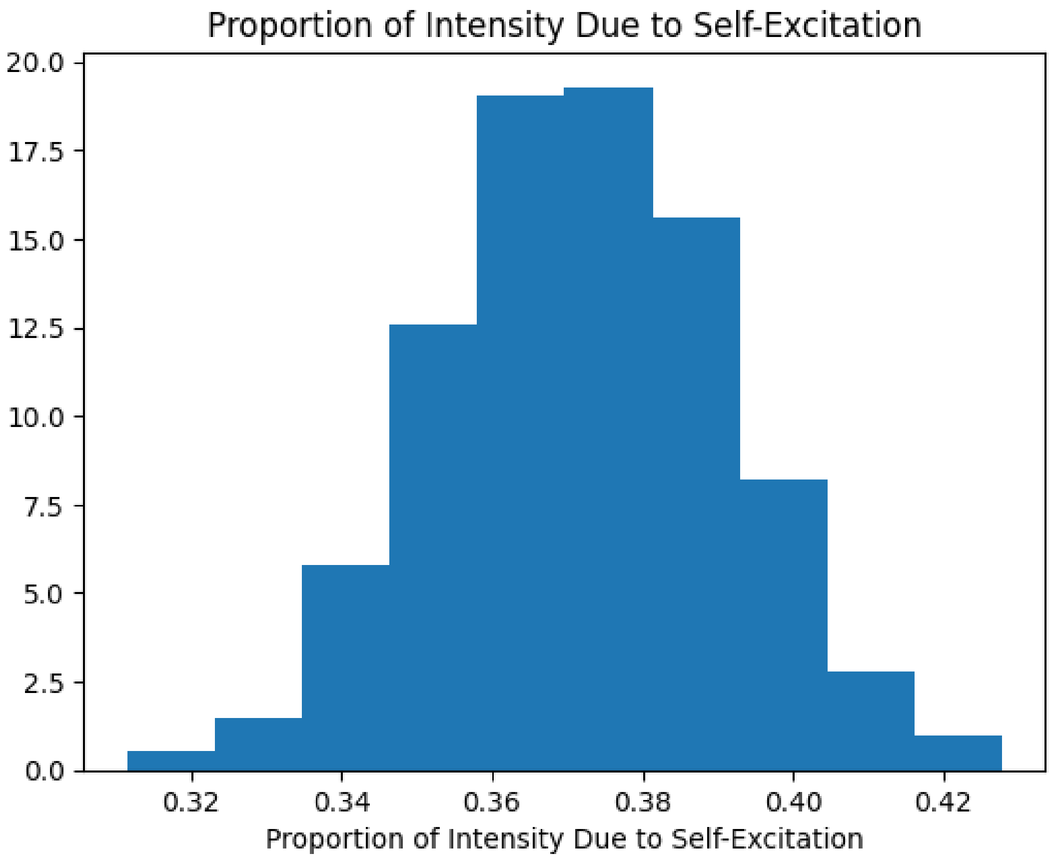
- latitude, longitude, and time of shooting
- demographic covariates defined on the 74 community areas of Chicago
- Polygon definition of the city of Chicago

```
#load Chicago Shooting data
data = load_Chicago_Shootings()
```

Cox Hawkes Model

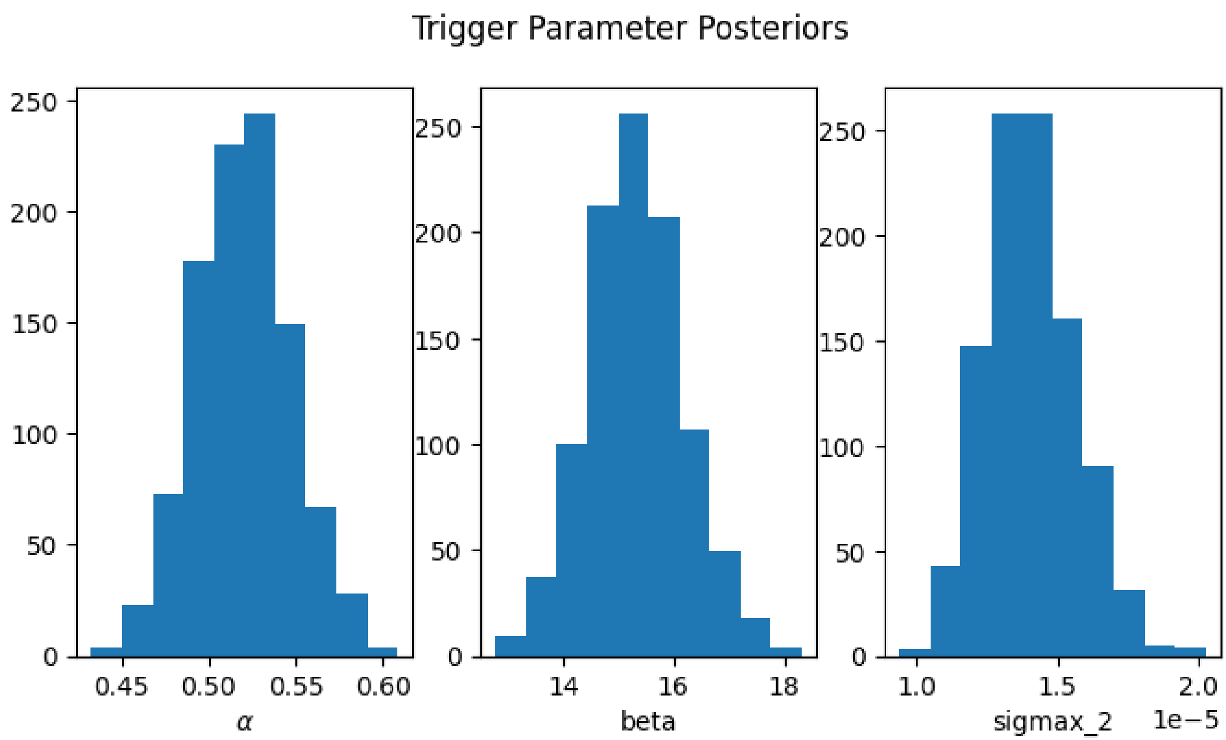
```
column_names =
['UNEMP_DENS','MEDINC','MED_HV','assoc_plus','VACANT_DEN',
       'VAC_HU_pct','HCUND20K_L','POP_DENS','CT_SP_WCHI']
model = Hawkes_Model(data['events_2022'],#spatiotemporal points
                     data['boundaries'],#Chicago boundaries
                     365,#Time frame (1 yr)
                     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)
                    )

WARNING:jax._src.lib.xla_bridge:No GPU/TPU found, falling back to CPU.
(Set TF_CPP_MIN_LOG_LEVEL=0 and rerun for more info.)
/home/imanring/PointProcess/Cox_Hawkes_Cov/bstpp/main.py:113:
UserWarning: Geometry is in a geographic CRS. Results from 'area' are
likely incorrect. Use 'GeoSeries.to_crs()' to re-project geometries to
a projected CRS before this operation.

  args['A_area'] = A.area.sum()/((A_[0,1]-A_[0,0])*(A_[1,1]-A_[1,0]))
/home/imanring/PointProcess/Cox_Hawkes_Cov/bstpp/main.py:213:
UserWarning: Geometry is in a geographic CRS. Results from 'area' are
likely incorrect. Use 'GeoSeries.to_crs()' to re-project geometries to
```

```
a projected CRS before this operation.

  intersect['area'] = intersect.area/((A_[0,1]-A_[0,0])*(A_[1,1]-
A_[1,0]))

model.run_svi(lr=0.02,num_steps=15000)

 94%|███████████████ | 14151/15000 [16:08<00:58, 14.62it/s, init loss:
-8490.0225, avg. loss [12751-13500]: -18347.2578]

model

<bstpp.main.Hawkes_Model at 0x7f8d2b633100>

model.save_rslts('output/Chicago_Shootings/cox_hawkes/output.pkl')

model.load_rslts('output/Chicago_Shootings/cox_hawkes/output.pkl')

model.log_expected_likelihood(data['events_2023'])

7509.26025390625

model.expected_AIC()

-18406.12109375

model.plot_prop_excitation()

0.3720119297504425
```

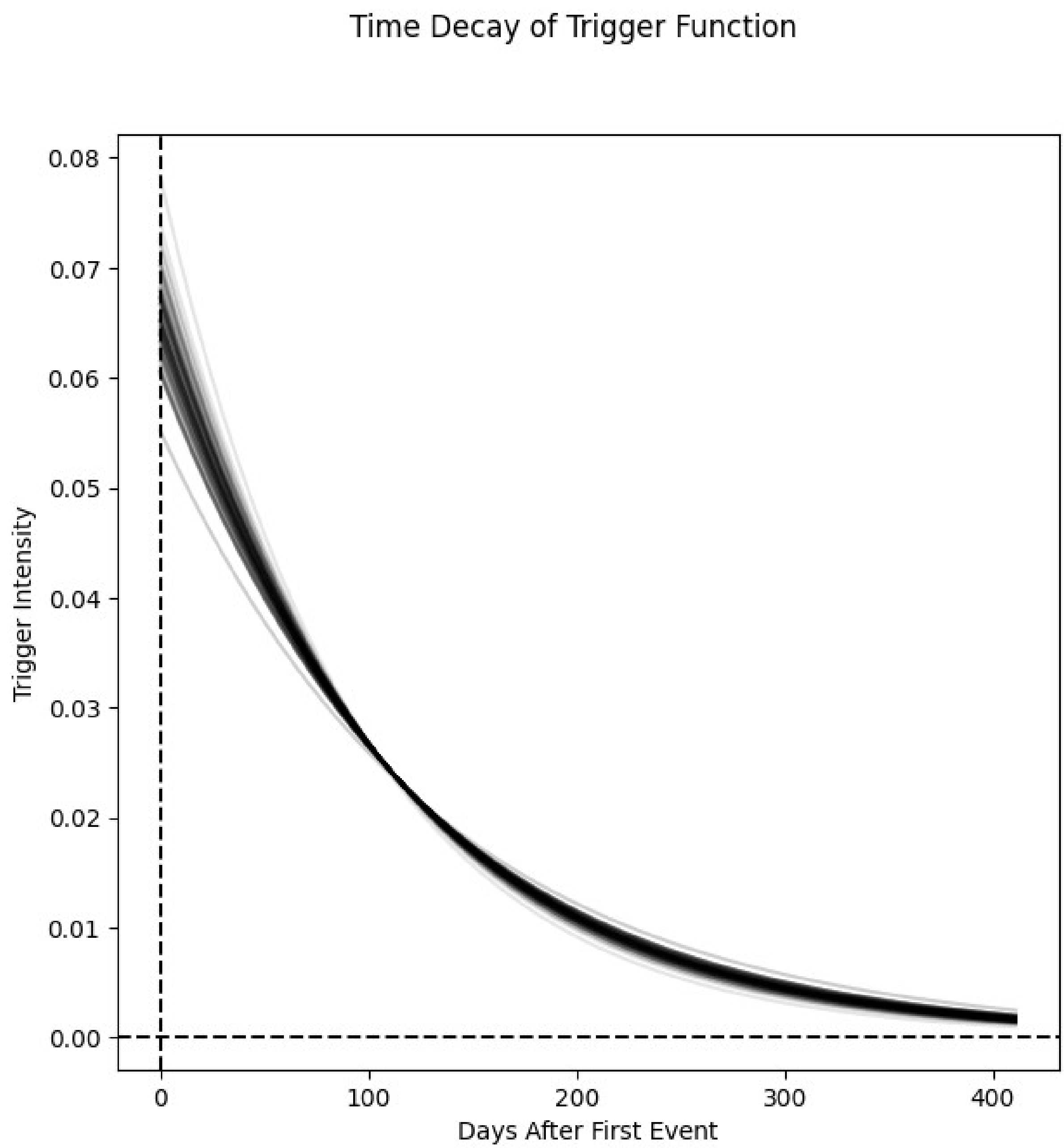



```
model.plot_trigger_posterior(trace=False)
```

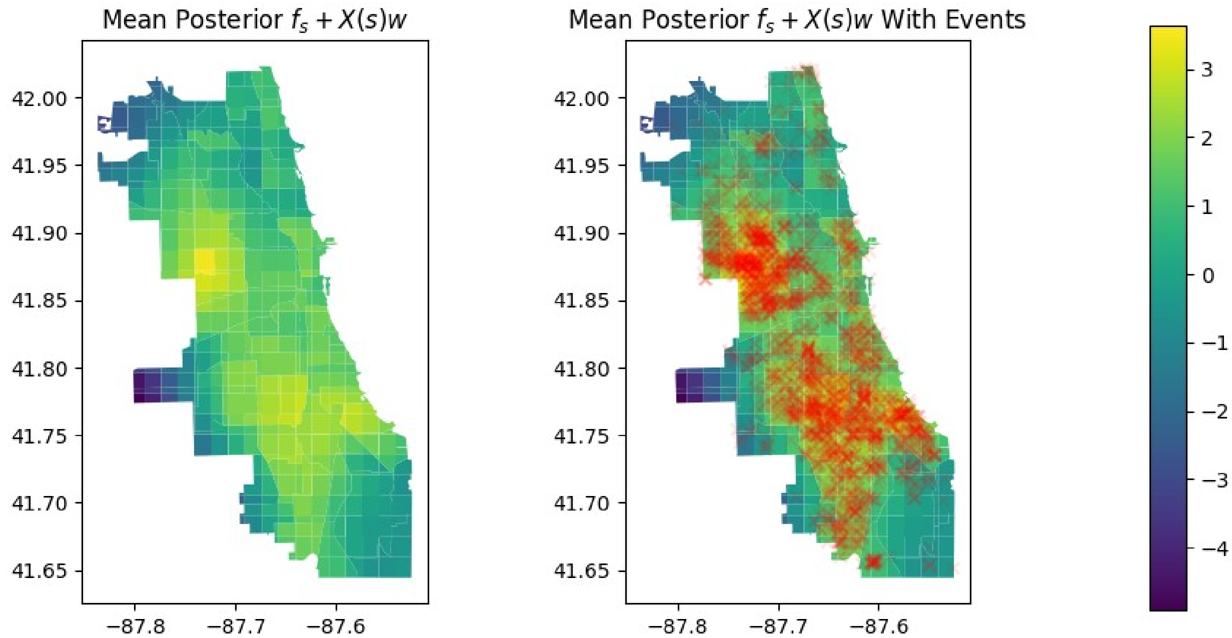
	Post Mean	Post Std	P(w>0)	[0.025	0.975]
alpha	0.519663	0.027255	1.0	0.467356	0.575680
beta	15.322455	0.869912	1.0	13.696313	17.139059
sigmax_2	0.000014	0.000002	1.0	0.000011	0.000017



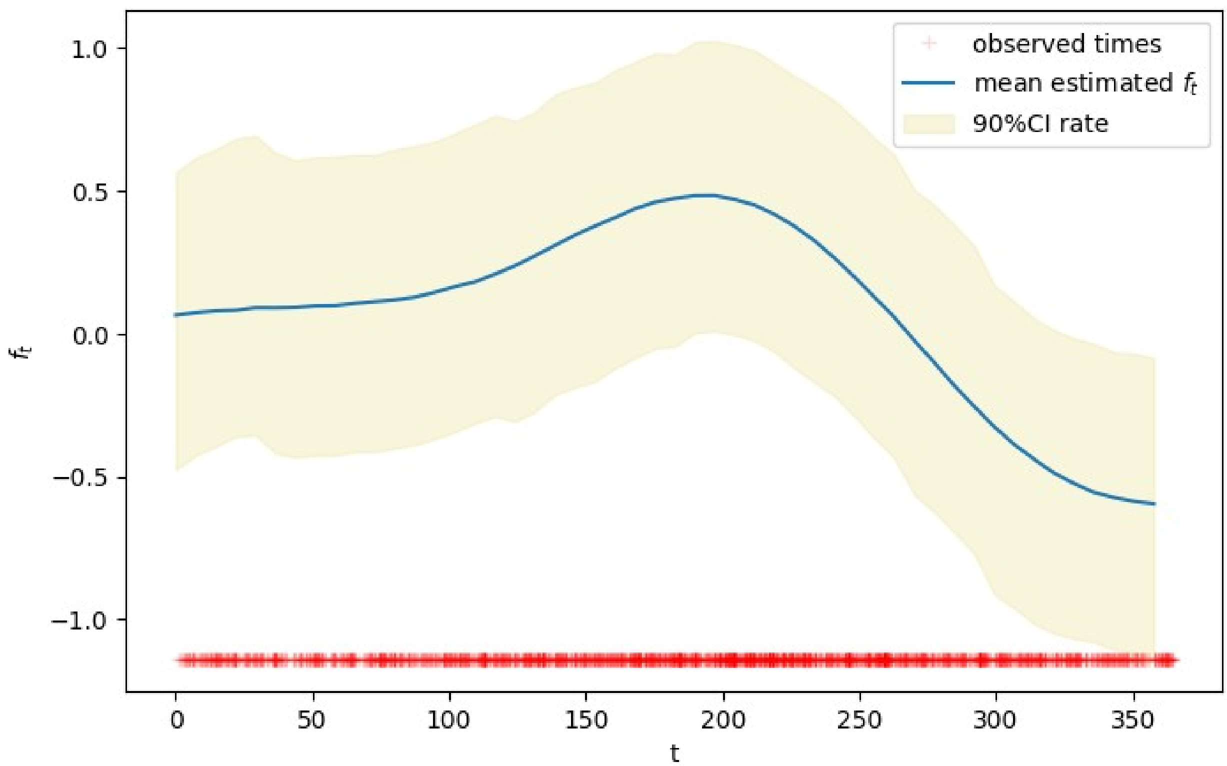

```
model.plot_trigger_time_decay()
```




```
model.plot_spatial(include_cov=True)
```



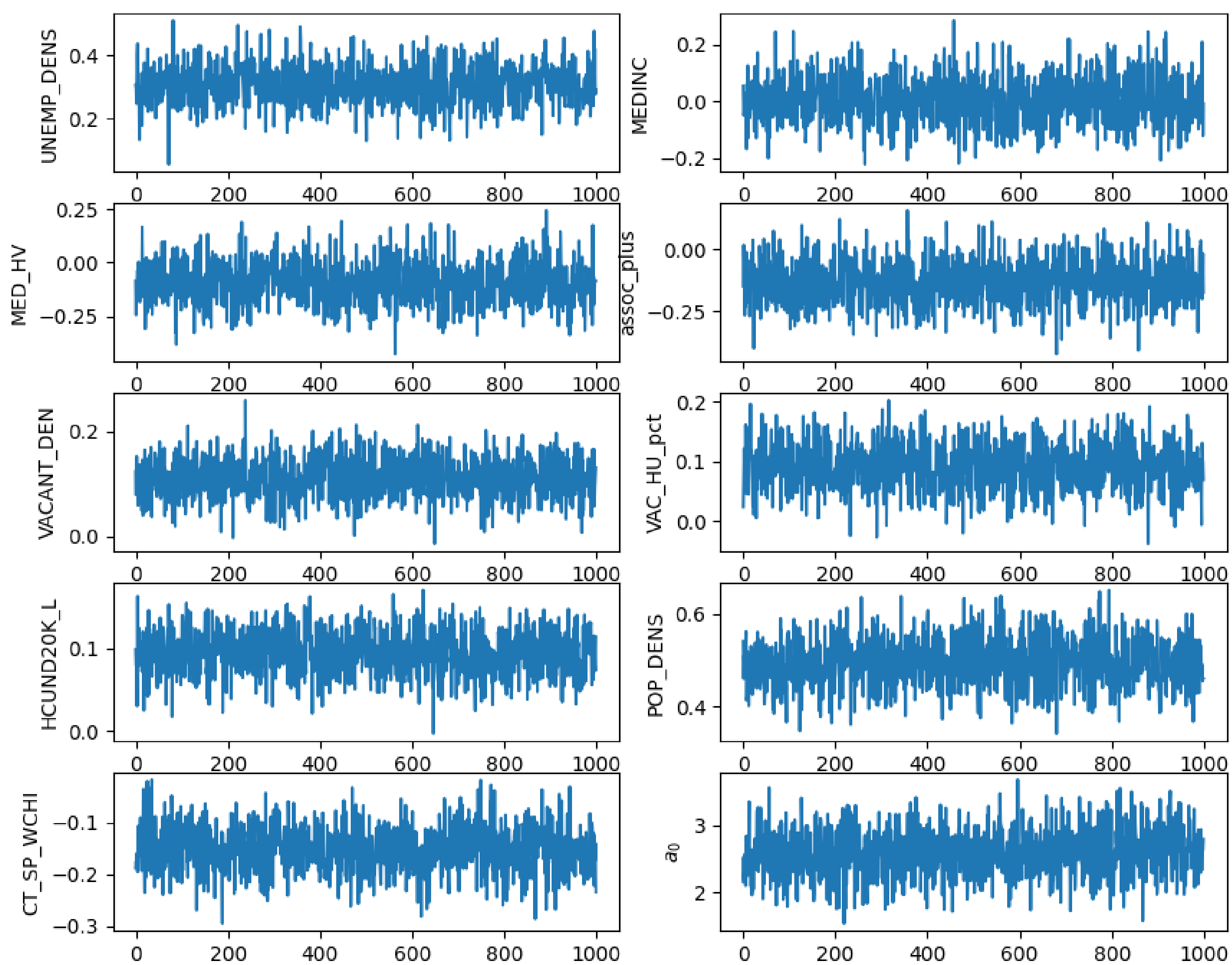

```
model.plot_temporal()
```




```
model.cov_weight_post_summary(trace=True)
```

	Post Mean	Post Std	P(w>0)	[0.025	0.975]
UNEMP_DENS	0.306386	0.062248	1.000	0.189508	0.432044
MEDINC	0.006433	0.081728	0.527	-0.153571	0.174928
MED_HV	-0.090982	0.098998	0.180	-0.282343	0.116833
assoc_plus	-0.132583	0.089921	0.077	-0.305087	0.041783
VACANT_DEN	0.112044	0.039022	0.998	0.037453	0.182883
VAC_HU_pct	0.091259	0.039587	0.989	0.016829	0.165188
HCUND20K_L	0.094737	0.026913	0.999	0.040015	0.144901
POP_DENS	0.494057	0.051886	1.000	0.394119	0.595436
CT_SP_WCHI	-0.150708	0.045120	0.000	-0.237306	-0.063454
a_0	2.590553	0.360308	1.000	1.879139	3.297548

Covariate Weights



Trigger function Extension

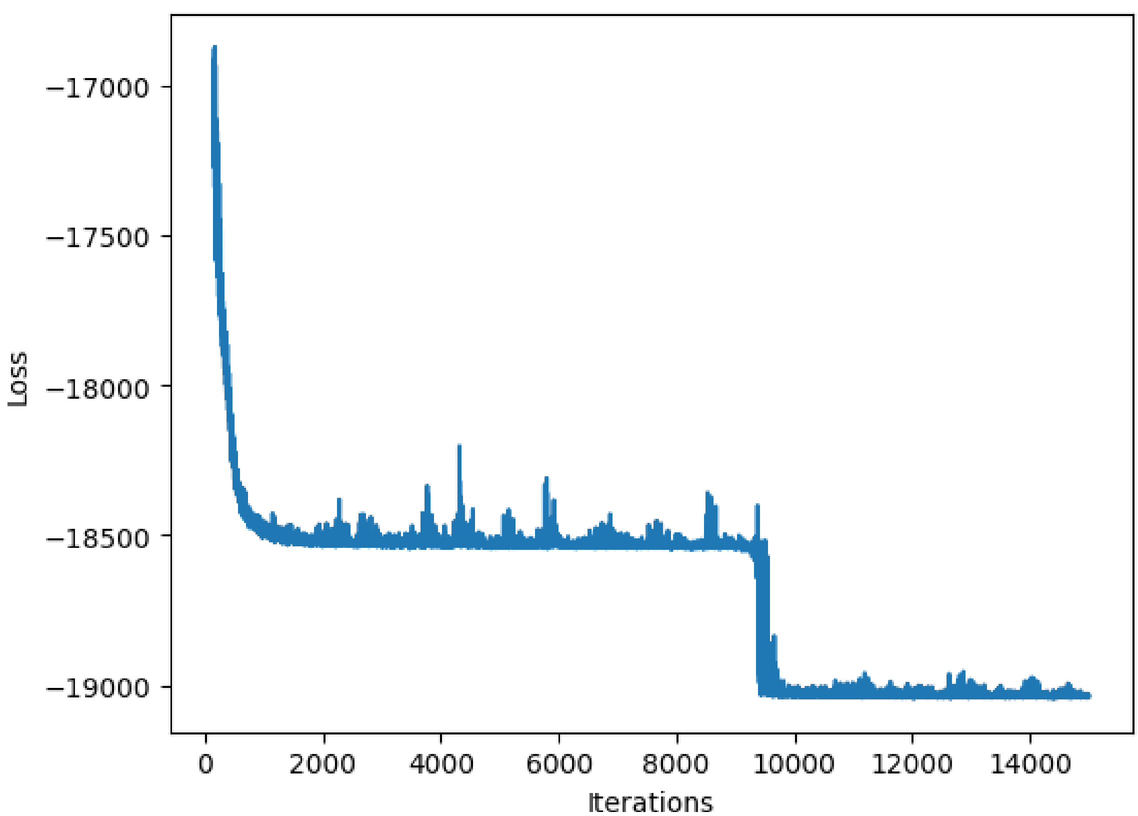
Here we define a spatial trigger function for an independent spatial double exponential distribution. The trigger is assumed to be a pdf and the reproduction rate is coded separately. The required methods to implement are:

- `compute_trigger`: compute the trigger function (pdf)
- `compute_integral`: compute the integral of the trigger function given limits (cdf)
- `get_par_names`: returns a list of the parameter names used in the trigger function

`simulate_trigger` is used only if a user wishes to simulate from the trigger function.

```
from bstpp.trigger import Trigger
import jax.numpy as jnp

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']

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
                   )

/home/imanring/PointProcess/Cox_Hawkes_Cov/bstpp/main.py:113:
UserWarning: Geometry is in a geographic CRS. Results from 'area' are
likely incorrect. Use 'GeoSeries.to_crs()' to re-project geometries to
a projected CRS before this operation.

  args['A_area'] = A.area.sum()/((A_[0,1]-A_[0,0])*(A_[1,1]-A_[1,0]))
/home/imanring/PointProcess/Cox_Hawkes_Cov/bstpp/main.py:213:
UserWarning: Geometry is in a geographic CRS. Results from 'area' are
likely incorrect. Use 'GeoSeries.to_crs()' to re-project geometries to
a projected CRS before this operation.

  intersect['area'] = intersect.area/((A_[0,1]-A_[0,0])*(A_[1,1]-
A_[1,0]))

model.run_svi(lr=0.02,num_steps=15000)

100%|████████████████| 15000/15000 [22:50<00:00, 10.95it/s, init loss:
-8969.3467, avg. loss [14251-15000]: -19033.2070]

Sampling Posterior...

SVI elapsed time: 1381.4155144691467
```



```
model.save_rslts('output/Chicago_Shootings/cox_hawkes/
output_double_exp.pkl')
```

```
model.load_rslts('output/Chicago_Shootings/cox_hawkes/
output_double_exp.pkl')
```

```
model.expected_AIC()
```

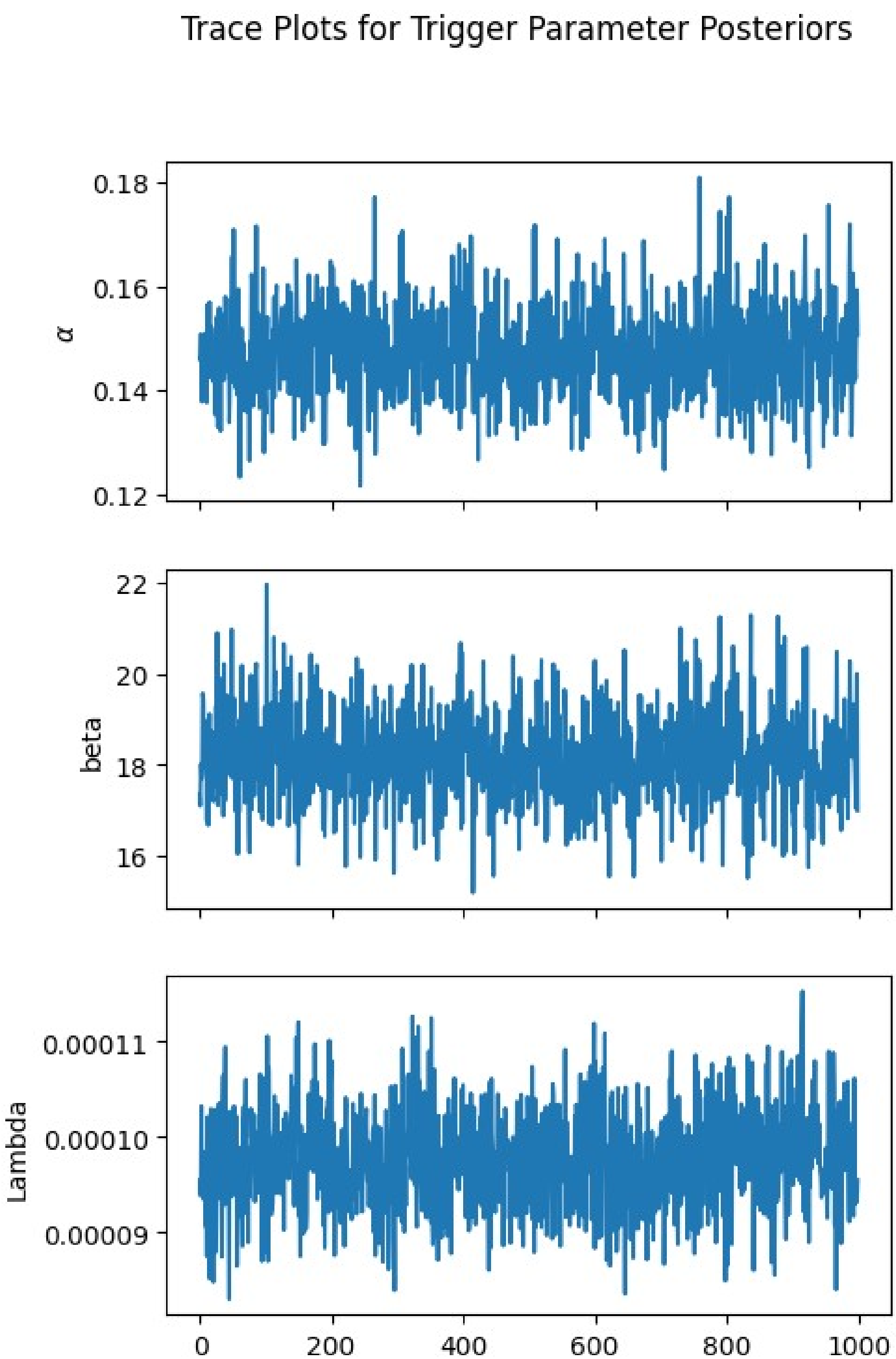
```
-19106.455078125
```

```
model.log_expected_likelihood(data['events_2023'])
```

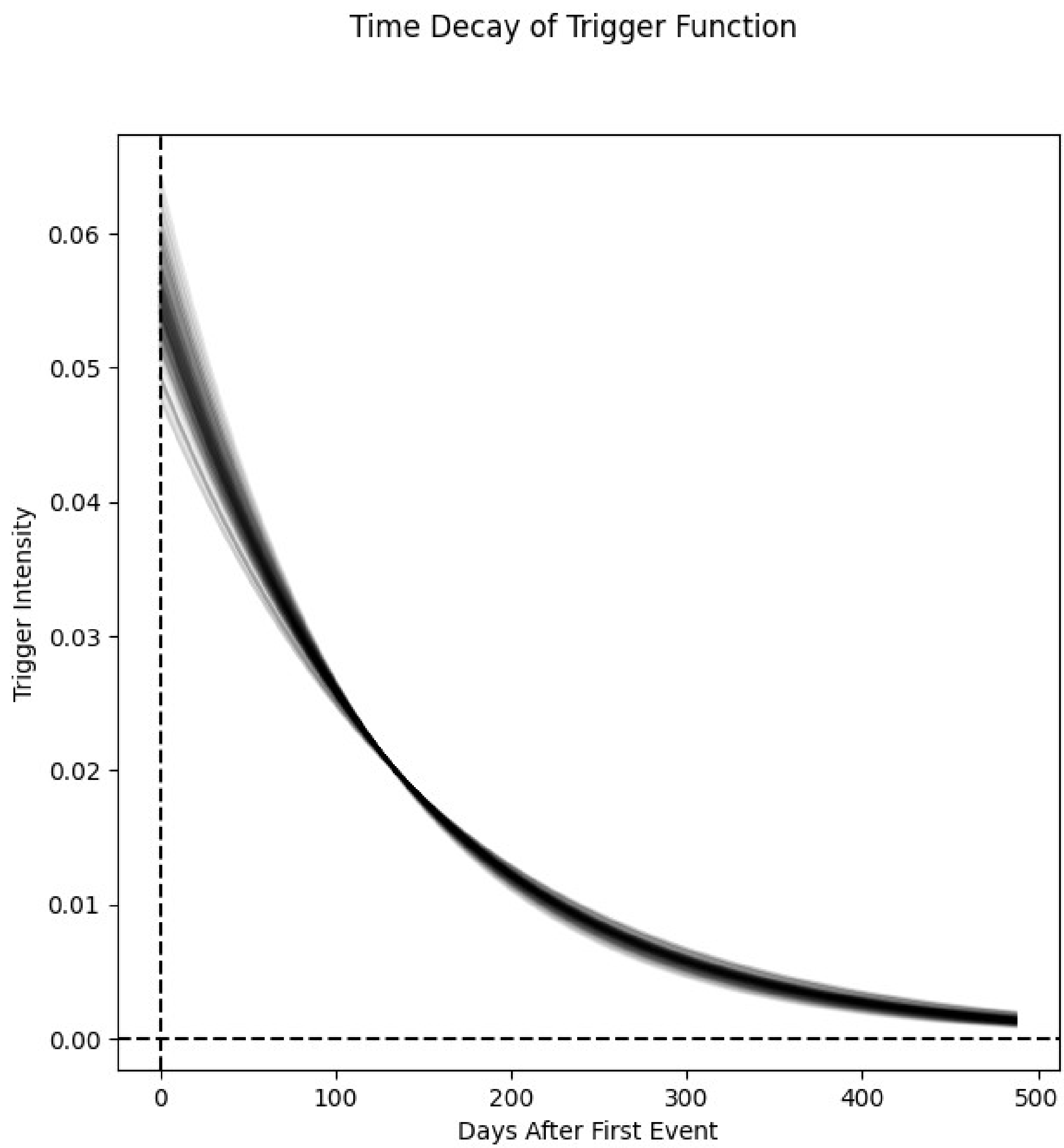
```
7807.57177734375
```

```
model.plot_trigger_posterior(trace=True)
```

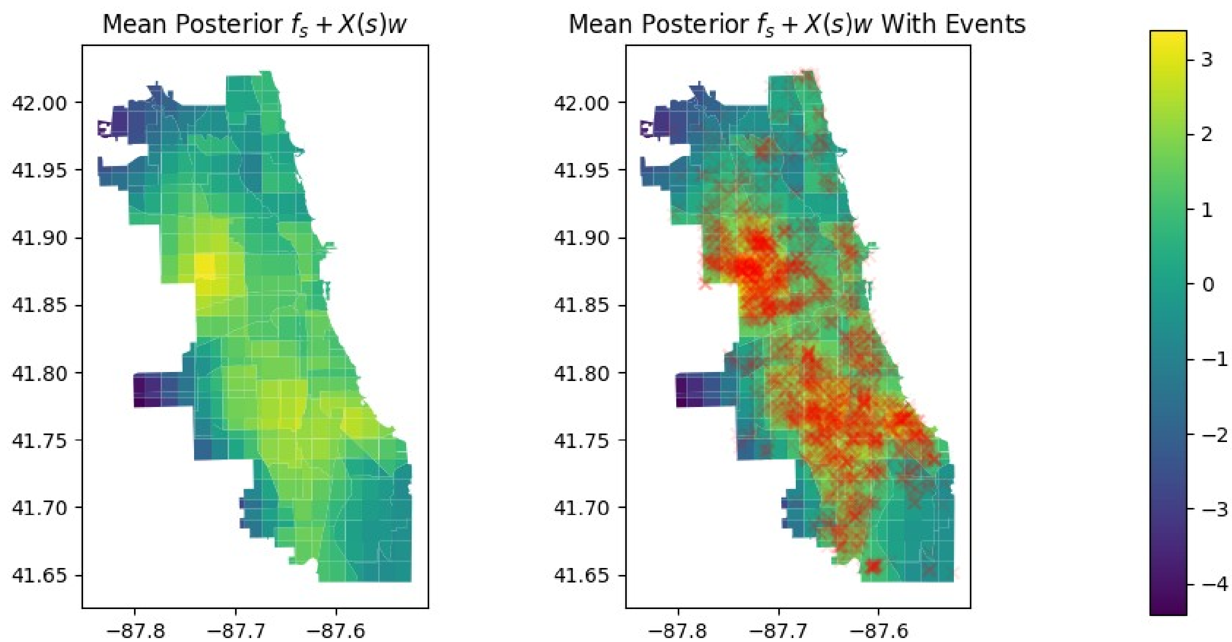
	Post Mean	Post Std	P(w>0)	[0.025	0.975]
alpha	0.147088	0.008850	1.0	0.130867	0.165129
beta	18.174726	1.008495	1.0	16.244011	20.302648
Lambda	0.000098	0.000005	1.0	0.000088	0.000109



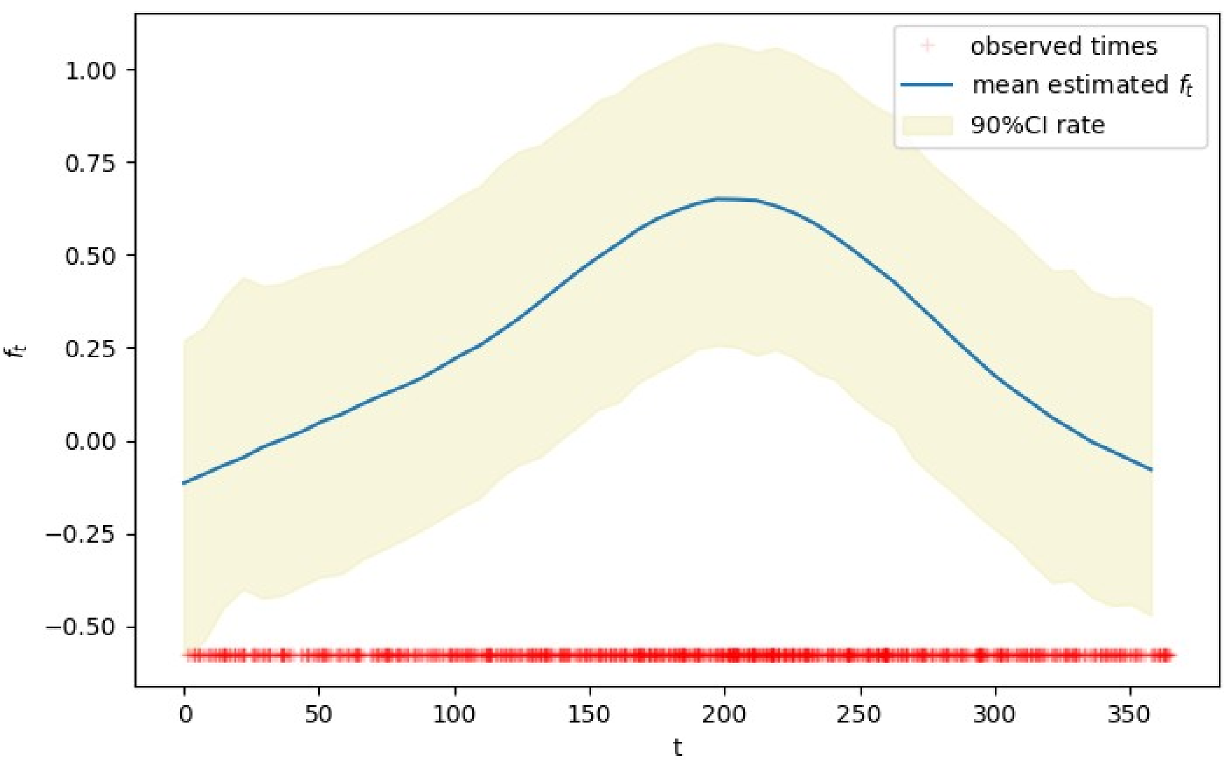

```
model.plot_trigger_time_decay()
```




```
model.plot_spatial(include_cov=True)
```

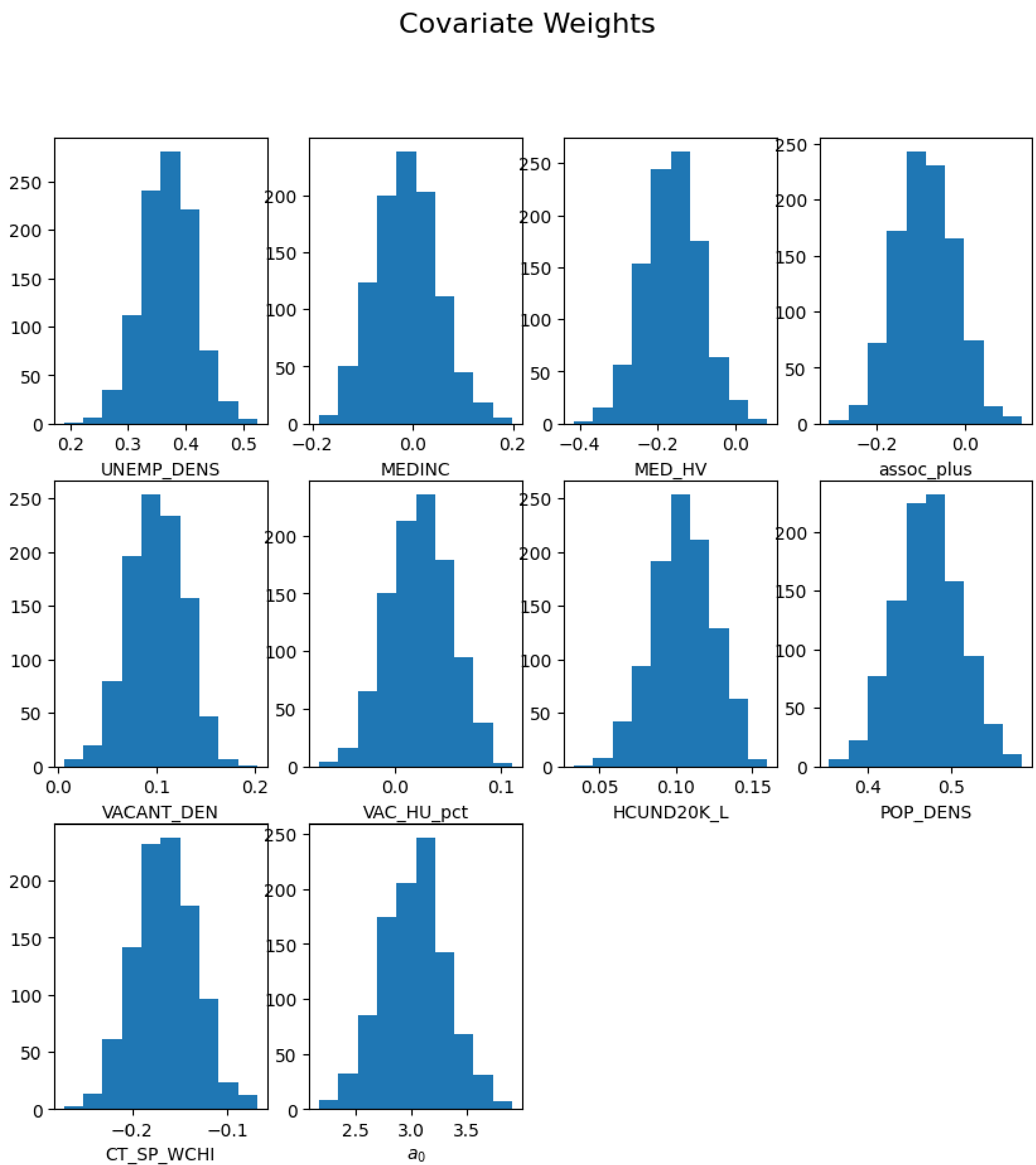



```
model.plot_temporal()
```




```
model.cov_weight_post_summary()
```

	Post Mean	Post Std	P(w>0)	[0.025	0.975]
UNEMP_DENS	0.368023	0.045239	1.000	0.281966	0.457796
MEDINC	-0.011645	0.062551	0.424	-0.133912	0.119011
MED_HV	-0.162242	0.073201	0.022	-0.301816	-0.011218
assoc_plus	-0.088143	0.067171	0.096	-0.213867	0.041958
VACANT_DEN	0.099696	0.027945	1.000	0.045608	0.149519
VAC_HU_pct	0.023184	0.029211	0.772	-0.031448	0.077794
HCUND20K_L	0.105068	0.019782	1.000	0.065802	0.142814
POP_DENS	0.472448	0.038975	1.000	0.397075	0.547998
CT_SP_WCHI	-0.165509	0.031963	0.000	-0.227017	-0.104208
a_0	3.025383	0.290694	1.000	2.451419	3.595780



No Covariates

See results when there are no covariates. All performance metrics decline.

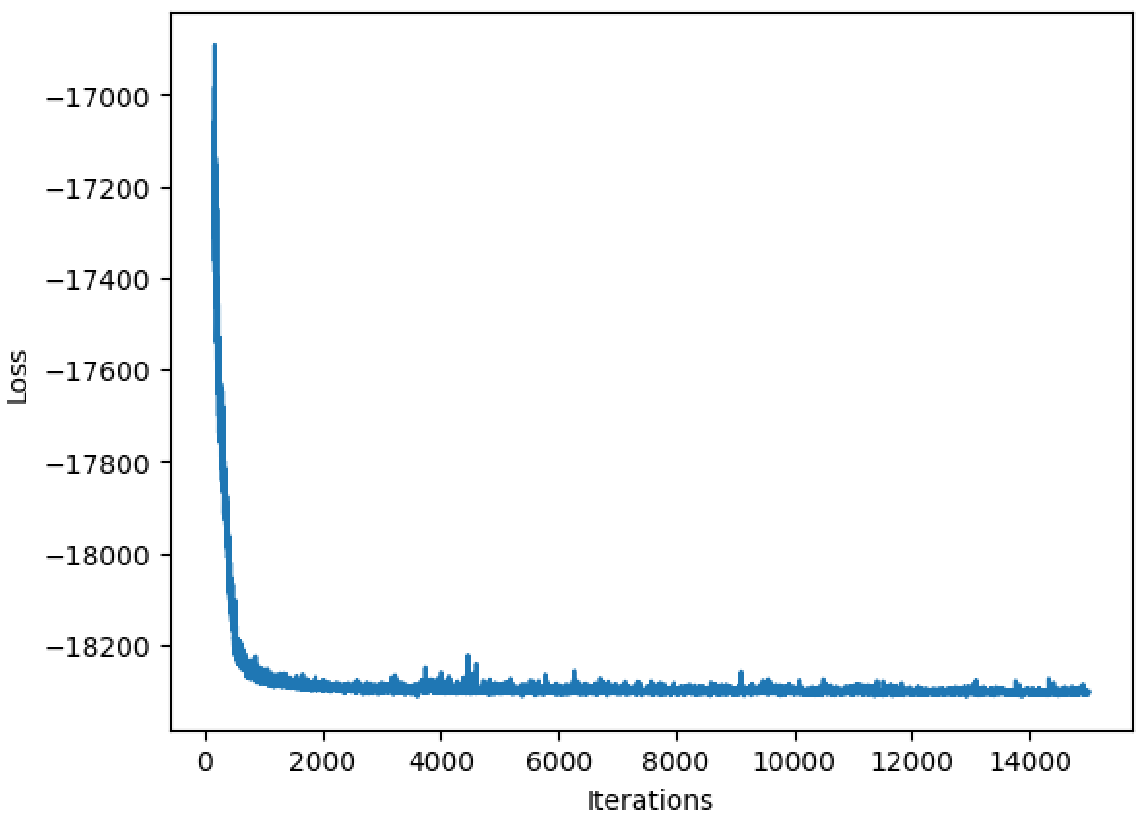
```
model = Hawkes_Model(data['events_2022'],#spatiotemporal points
                     data['boundaries'],#Chicago boundaries
                     365,#Time frame (1 yr)
```

```
                     True,#use Cox as background
                     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
                    )
```

```
/home/imanring/PointProcess/Cox_Hawkes_Cov/bstpp/main.py:113:
UserWarning: Geometry is in a geographic CRS. Results from 'area' are
likely incorrect. Use 'GeoSeries.to_crs()' to re-project geometries to
a projected CRS before this operation.

  args['A_area'] = A.area.sum()/((A_[0,1]-A_[0,0])*(A_[1,1]-A_[1,0]))
```

```
model.run_svi(lr=0.02,num_steps=15000)
```

```
100%|████████████████| 15000/15000 [22:20<00:00, 11.19it/s, init loss:
-9537.4941, avg. loss [14251-15000]: -18303.0938]


SVI elapsed time: 1360.987762928009
```




```
model.save_rslts('output/Chicago_Shootings/cox_hawkes/
output_double_exp_no_cov.pkl')
```

```
model.load_rslts('output/Chicago_Shootings/cox_hawkes/
output_double_exp_no_cov.pkl')
```

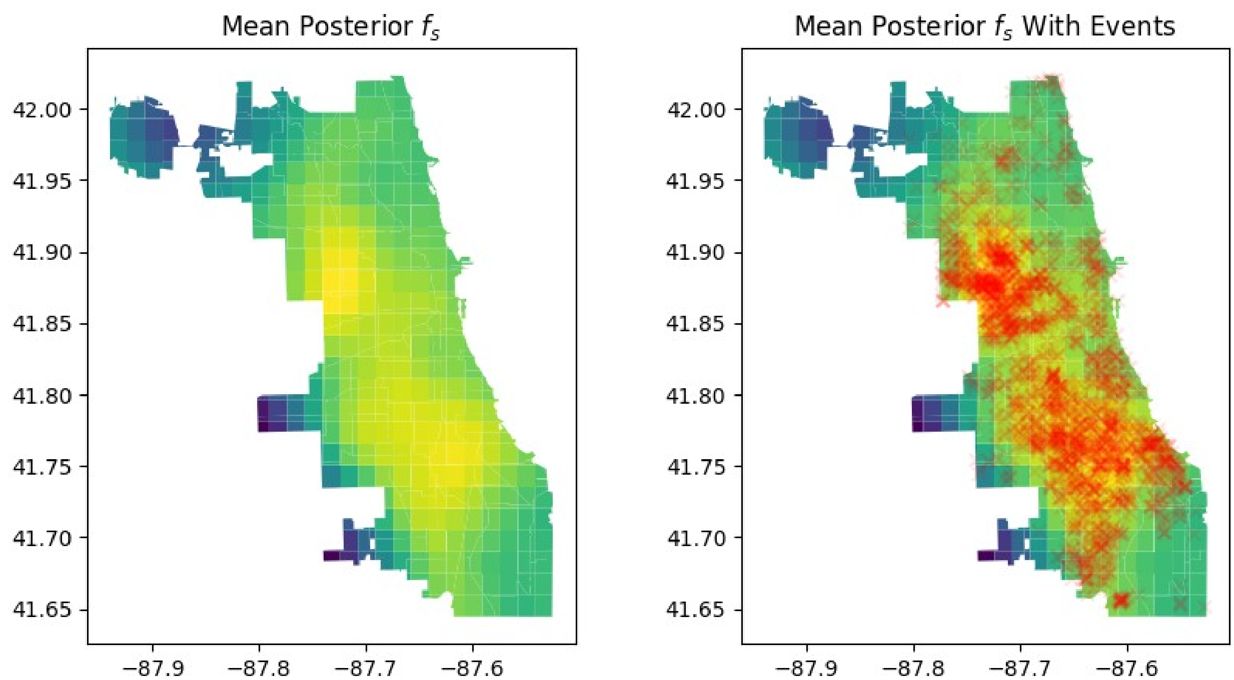
```
model.expected_AIC()
```

```
-18368.54296875
```

```
model.log_expected_likelihood(data['events_2023'])
```

```
7476.47998046875
```

```
model.plot_spatial(include_cov=False)
```

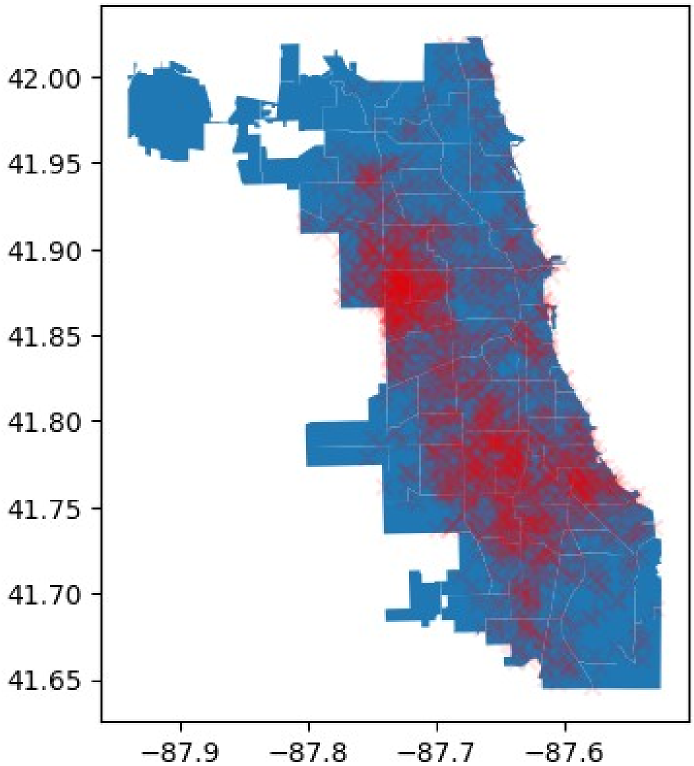


Simulation

Use the function `simulate` to simulate a new realization from the posterior mean parameters.

```
sample = model.simulate()
```

```
import matplotlib.pyplot as plt
ax = model.A.plot()
sample.plot(ax=ax,color='red',marker='x',alpha=0.1)
plt.show()
```



Boko Haram Dataset

Conflict events in Nigeria involving Boko Haram. Spatial covariates from prio-grid.

Log Gaussian Cox Process Model

```
#load Boko Haram conflicts from Nigeria
data = load_Boko_Haram()
column_names = ['droughtstart_speibase', 'urban_ih_log',
'droughtyr_speigdm',
       'herb_gc', 'capdist', 'grass_ih_log', 'nlights_sd_log',
'water_gc_log',
       'pop_gpw_sd_log', 'pasture_ih']
lgcp_bh_model = LGCP_Model(data['events'],#event data
                           data['boundaries'],#boundary of events
                           data['events']['T'].max(),
                           spatial_cov=data['covariates'],#covariate
matrix
                           cov_grid_size=(0.5,0.5),#grid cell width
and height to construct spatial covariate grid
                           cov_names = column_names,#columns to use
from covariates
                           a_0=dist.Normal(1,10)#set prior
                          )
```

```
lgcp_bh_model.run_mcmc()
```

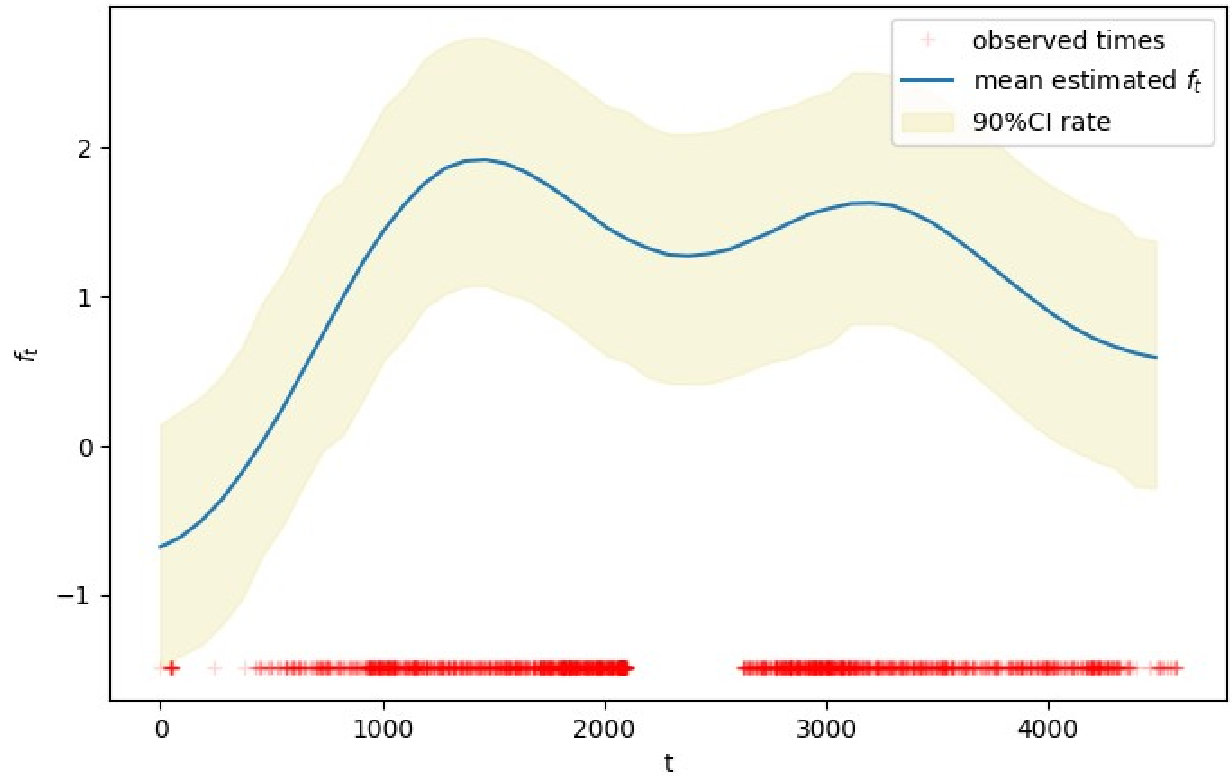
```
sample: 100%|██████████████████████████████████| 1500/1500 [00:36<00:00,
41.30it/s, 511 steps of size 1.25e-02. acc. prob=0.95]


                    mean       std    median      5.0%     95.0%
n_eff     r_hat
           a_0     -3.18      0.61     -3.19     -4.13     -2.16
491.08      1.00
          w[0]     -0.39      0.11     -0.39     -0.59     -0.22
658.23      1.00
          w[1]     -0.07      0.05     -0.07     -0.16     -0.00
651.68      1.00
          w[2]     -0.05      0.03     -0.05     -0.09      0.00
837.27      1.00
          w[3]      0.13      0.07      0.13      0.02      0.24
867.39      1.00
          w[4]      0.63      0.08      0.63      0.51      0.75
1055.48      1.00
          w[5]      0.18      0.04      0.18      0.12      0.23
906.50      1.00
          w[6]      0.78      0.04      0.78      0.72      0.86
927.72      1.00
          w[7]     -0.12      0.04     -0.12     -0.19     -0.05
994.49      1.00
          w[8]      0.78      0.08      0.78      0.67      0.93
661.91      1.00
          w[9]      0.58      0.05      0.58      0.50      0.67
873.24      1.00
  z_spatial[0]      0.04      0.21      0.03     -0.30      0.40
547.16      1.00
  z_spatial[1]     -2.45      0.30     -2.44     -2.93     -1.94
517.54      1.00
  z_spatial[2]      0.09      0.14      0.09     -0.14      0.31
656.15      1.00
  z_spatial[3]      0.55      0.13      0.55      0.35      0.76
533.83      1.00
  z_spatial[4]     -3.04      0.21     -3.03     -3.38     -2.70
350.69      1.00
  z_spatial[5]      1.84      0.20      1.83      1.50      2.15
275.58      1.00
  z_spatial[6]     -1.66      0.16     -1.67     -1.95     -1.43
293.41      1.00
  z_spatial[7]      2.07      0.22      2.08      1.71      2.41
491.37      1.00
  z_spatial[8]     -3.93      0.27     -3.92     -4.34     -3.48
407.93      1.00
  z_spatial[9]      1.96      0.23      1.95      1.58      2.32
263.13      1.00
```

```
 z_spatial[10]      3.27      0.30      3.29      2.76      3.73
313.58      1.00
 z_spatial[11]     -1.40      0.20     -1.40     -1.69     -1.05
535.95      1.00
 z_spatial[12]     -1.17      0.17     -1.17     -1.45     -0.91
347.78      1.00
 z_spatial[13]     -0.67      0.16     -0.66     -0.93     -0.41
369.06      1.00
 z_spatial[14]      1.52      0.12      1.51      1.32      1.70
673.69      1.00
 z_spatial[15]     -1.44      0.18     -1.44     -1.76     -1.18
344.81      1.00
 z_spatial[16]     -0.56      0.20     -0.56     -0.87     -0.23
532.21      1.00
 z_spatial[17]     -1.21      0.14     -1.20     -1.44     -0.99
531.65      1.00
 z_spatial[18]      1.72      0.20      1.72      1.41      2.06
404.75      1.00
 z_spatial[19]      0.37      0.10      0.36      0.22      0.53
769.36      1.00
 z_temporal[0]     -1.03      0.08     -1.03     -1.16     -0.89
1113.75      1.00
 z_temporal[1]      0.02      0.93      0.04     -1.45      1.53
1563.80      1.00
 z_temporal[2]     -1.03      0.08     -1.04     -1.17     -0.90
484.04      1.00
 z_temporal[3]     -0.23      0.22     -0.23     -0.54      0.16
1222.19      1.00
 z_temporal[4]     -1.95      0.88     -1.94     -3.35     -0.52
465.63      1.00
 z_temporal[5]     -2.15      0.44     -2.13     -2.89     -1.46
819.75      1.00
 z_temporal[6]      0.04      1.02      0.06     -1.61      1.74
1246.55      1.00
 z_temporal[7]      1.94      0.23      1.93      1.56      2.27
464.75      1.00
 z_temporal[8]     -0.01      0.97     -0.02     -1.50      1.65
1306.60      1.00
 z_temporal[9]     -0.18      0.20     -0.17     -0.49      0.14
452.22      1.00
z_temporal[10]     -0.01      1.00     -0.00     -1.76      1.52
1462.34      1.00

Number of divergences: 0

MCMC elapsed time: 43.3479950428009
```

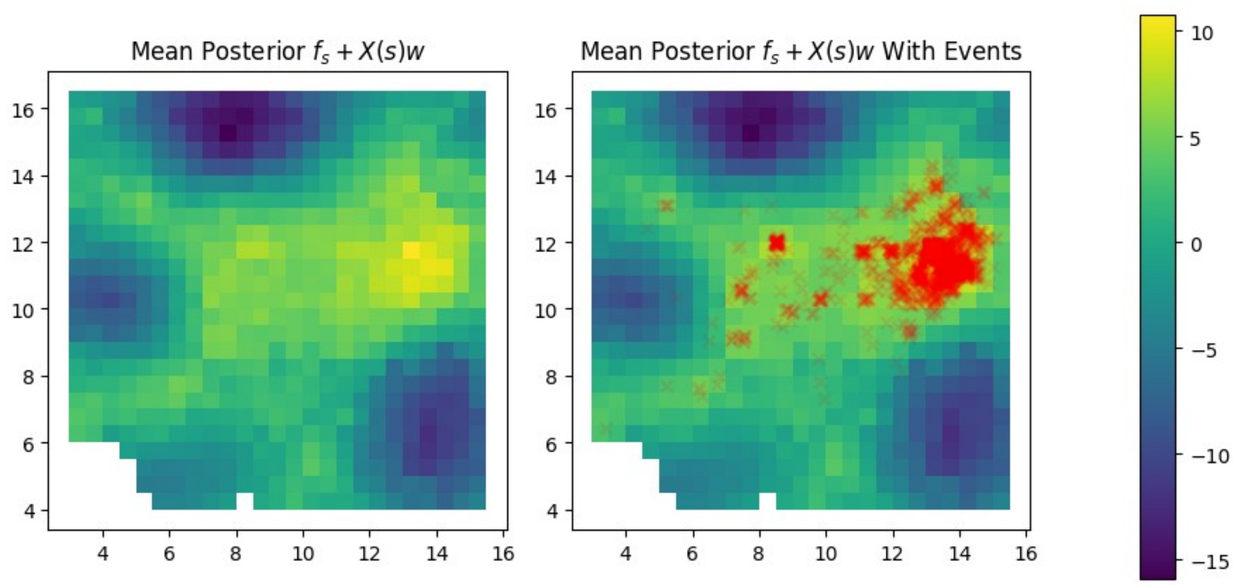
```
lgcp_bh_model.expected_AIC()
```

```
-29747.21875
```

```
lgcp_bh_model.plot_temporal()
```

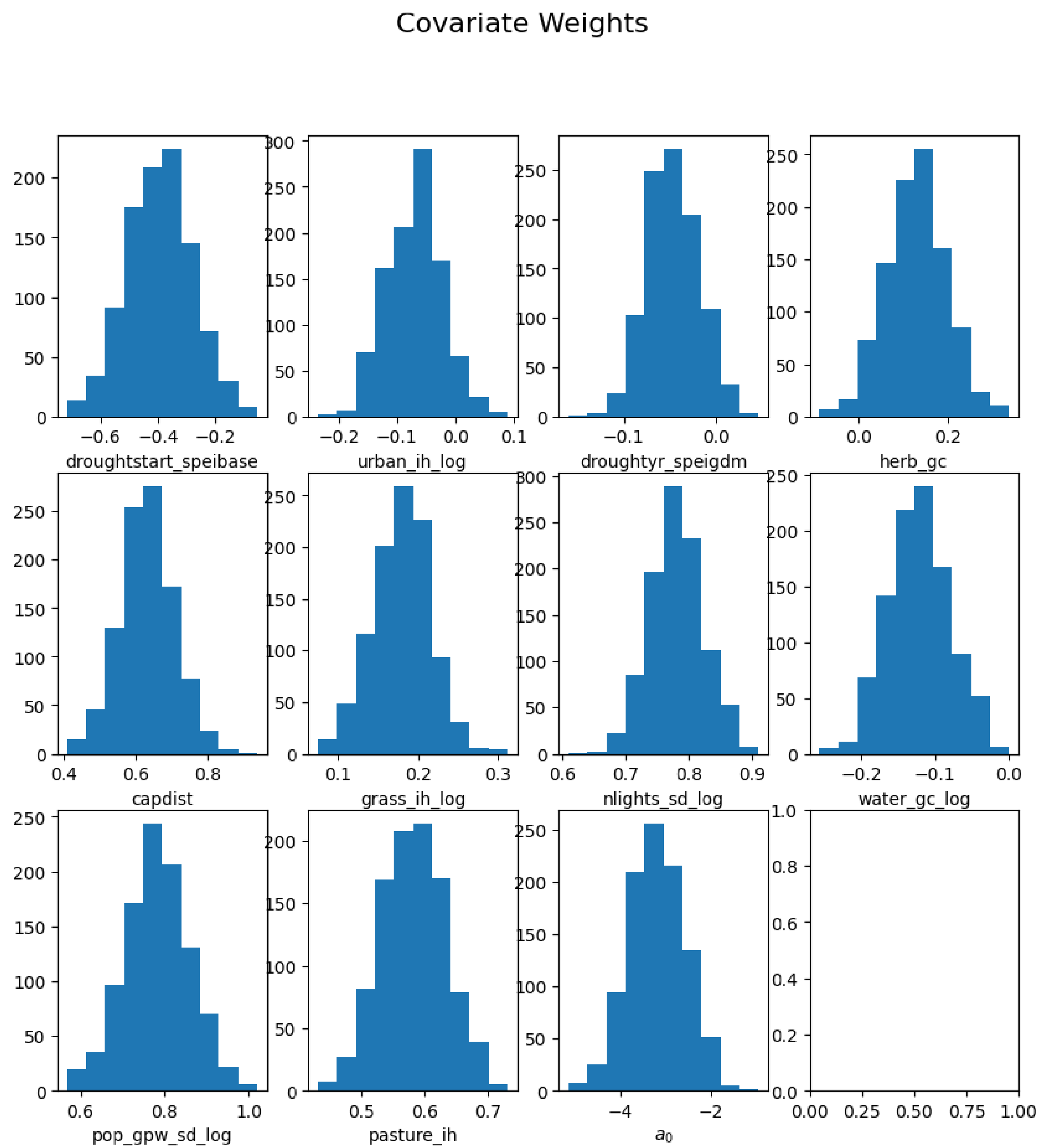



```
lgcp_bh_model.plot_spatial(include_cov=True)
```




```
lgcp_bh_model.cov_weight_post_summary()
```

```
                       Post Mean  Post Std          z         P>|z|
[0.025  \
droughtstart_speibase  -0.393704  0.113655  -3.464037  5.321333e-04 -
0.616811
urban_ih_log           -0.068996  0.047387  -1.455993  1.453947e-01 -
0.157801
droughtyr_speigdm      -0.046708  0.028325  -1.649012  9.914511e-02 -
0.098879
herb_gc                 0.128528  0.066626   1.929096  5.371897e-02 -
0.001818
capdist                 0.631130  0.075291   8.382559  0.000000e+00
0.480287
grass_ih_log            0.179572  0.036071   4.978258  6.415900e-07
0.108358
nlights_sd_log          0.780560  0.042008  18.581089  0.000000e+00
0.700161
water_gc_log           -0.122108  0.041723  -2.926630  3.426567e-03 -
0.198359
pop_gpw_sd_log          0.782923  0.076487  10.236053  0.000000e+00
0.627051
pasture_ih              0.582839  0.050643  11.508811  0.000000e+00
0.487036
a_0                    -3.179106  0.611577  -5.198213  2.012132e-07 -
4.366732

                          0.975]
droughtstart_speibase -0.171760
urban_ih_log           0.025017
droughtyr_speigdm      0.011444
herb_gc                0.256239
capdist                0.783103
grass_ih_log           0.247056
nlights_sd_log         0.859844
water_gc_log          -0.039363
pop_gpw_sd_log         0.931252
pasture_ih             0.681689
a_0                   -2.014276
```

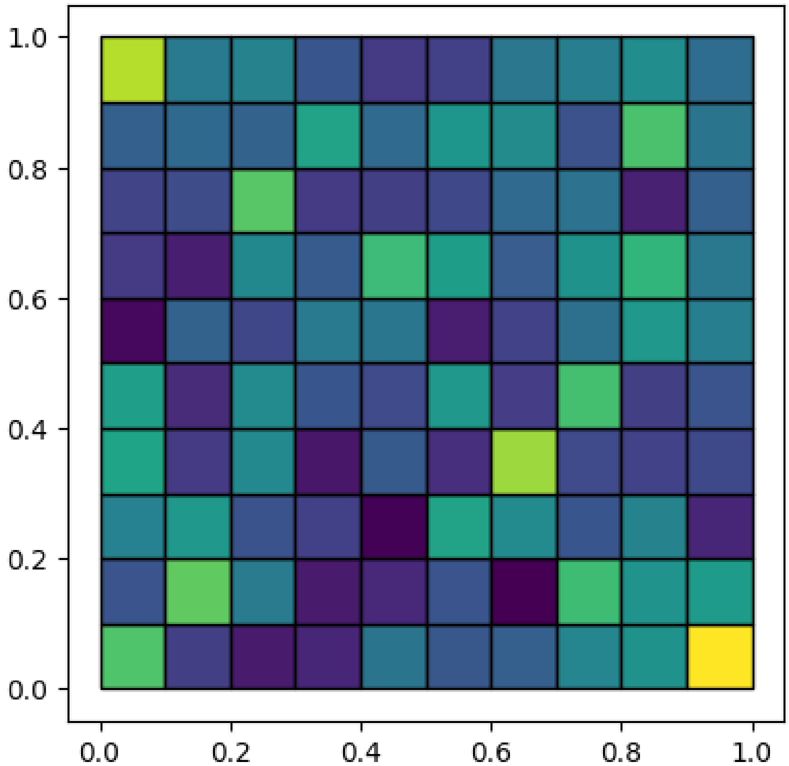


Simulation

Simulate Cox Hawkes process with covariates and perform inference to regain original parameters. You can provide a dictionary of parameters to the `simulate` function and it will simulate a realization from those parameters.

```python
from bstpp.main import Hawkes_Model
import numpyro.distributions as dist
import geopandas as gpd
from shapely.geometry import Polygon
import numpy as np
import pandas as pd

np.random.seed(16)

length = .1
wide = .1

cols = list(np.arange(0, 1 + wide, wide))
rows = list(np.arange(0, 1 + length, length))

polygons = []
for x in cols[:-1]:
    for y in rows[:-1]:
        polygons.append(Polygon([(x,y), (x+wide, y), (x+wide,
y+length), (x, y+length)]))

w = np.random.normal(scale = 0.3, size = 3)
X = np.random.normal(size=(len(polygons),3))
sp_cov = gpd.GeoDataFrame(data=X,geometry=polygons)

sp_cov['int'] = np.exp(X @ w)
sp_cov.plot('int',edgecolor='black')
```

```
<AxesSubplot:>
```

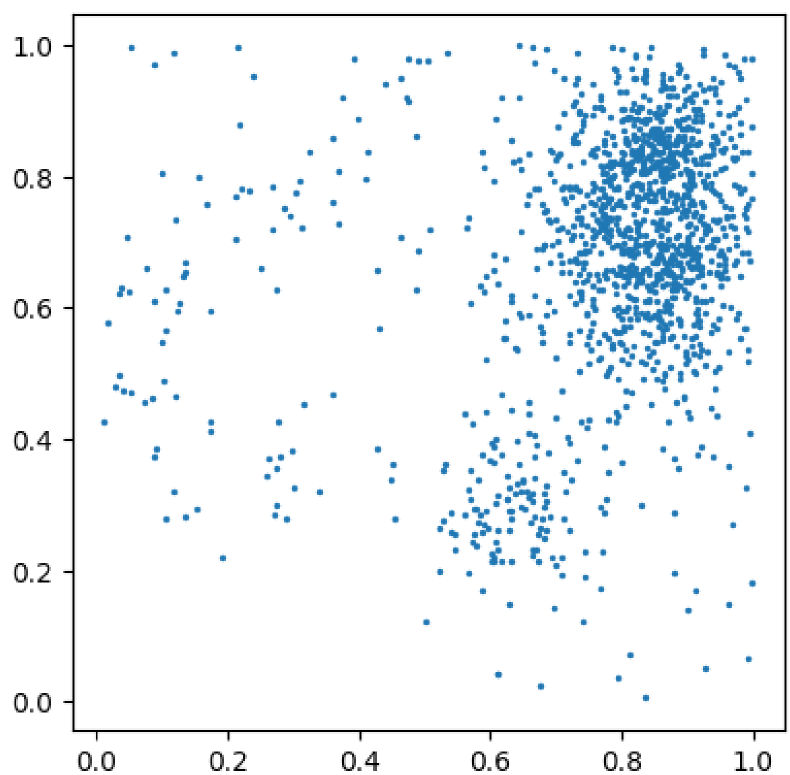



```
column_names = [0,1,2]
model = Hawkes_Model(pd.DataFrame({'X':[1],'Y':[1],'T':
[1]}),#spatiotemporal points
                     sp_cov,#Chicago boundaries
                     50,#Time frame (1 yr)
                     True,#use Cox as background
                     spatial_cov=sp_cov,#spatial 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)
                    )

model.args['sp_var_mu'] = 1.
par = {'alpha':0.25,'beta':2.,'a_0':1.0,'sigmax_2':0.05**2,

'z_spatial':np.random.normal(size=20),'z_temporal':np.random.normal(si
ze=11),
       'w':w
      }
sample = model.simulate(par)

sample
```

```
                X          Y          T                   geometry
0     0.135396  0.653549   0.070058  POINT (0.13540 0.65355)
913   0.135168  0.668185  45.468541  POINT (0.13517 0.66819)
963   0.126994  0.607680  47.384735  POINT (0.12699 0.60768)
1104  0.131486  0.648233  26.080482  POINT (0.13149 0.64823)
1185  0.105198  0.627285  34.357803  POINT (0.10520 0.62728)
...        ...       ...        ...                        ...
1175  0.584982  0.633698  31.465032  POINT (0.58498 0.63370)
1317  0.568688  0.607004  34.504676  POINT (0.56869 0.60700)
1235  0.275743  0.428338  46.538023  POINT (0.27574 0.42834)
1280  0.116611  0.320302  49.406639  POINT (0.11661 0.32030)
1296  0.593966  0.520073  21.007639  POINT (0.59397 0.52007)

[1349 rows x 4 columns]
```

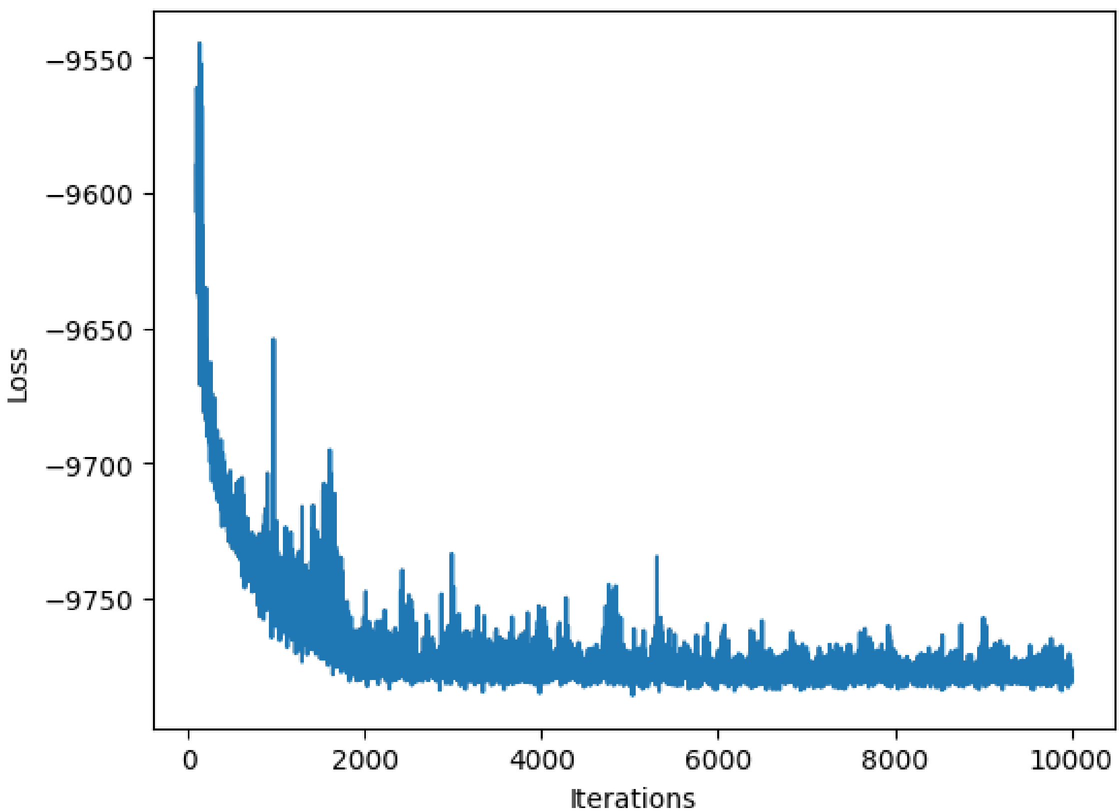
```
sample.plot(markersize=2)
```

```
<AxesSubplot:>
```

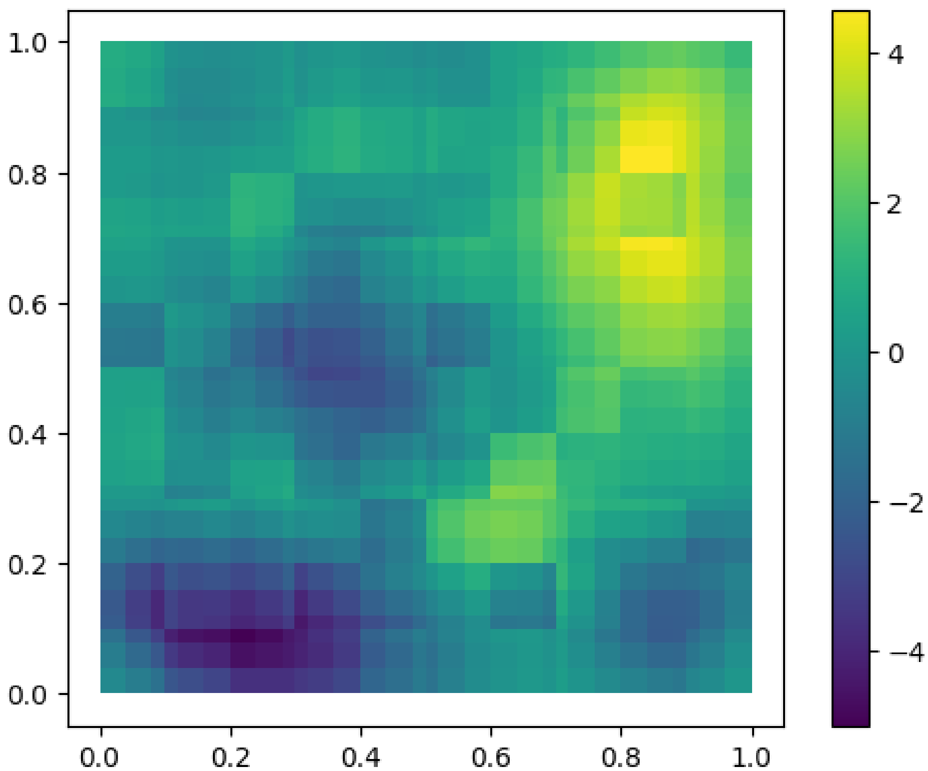



```
model =
Hawkes_Model(sample[['X','Y','T']].sort_values('T'),#spatiotemporal
points
                     sp_cov,#Chicago boundaries
                     50,#Time frame (1 yr)
                     True,#use Cox as background
```

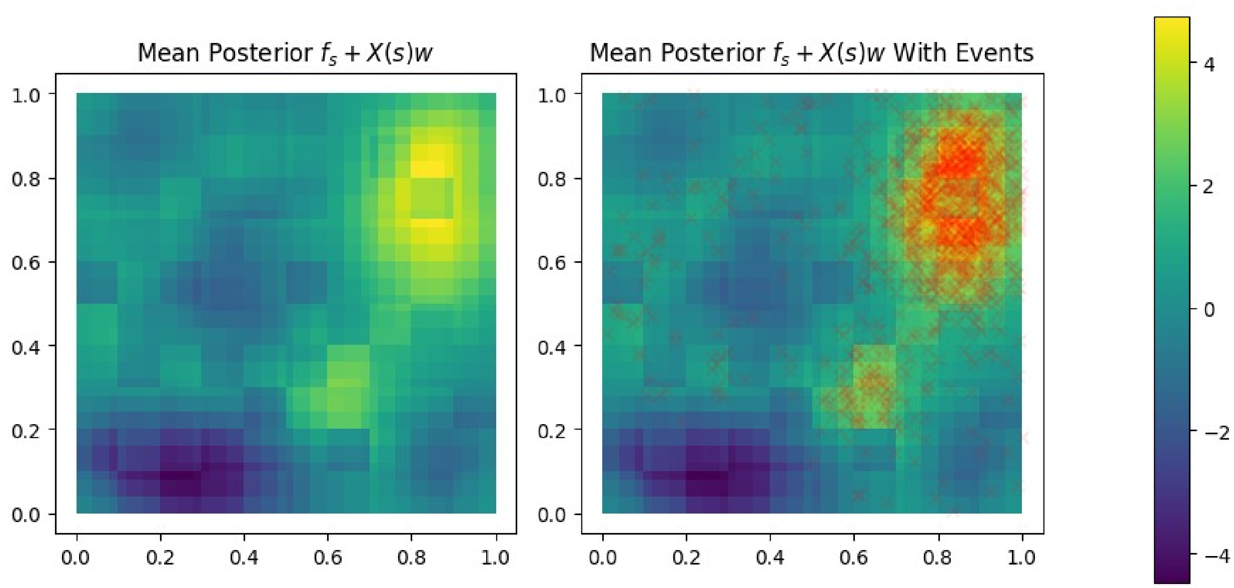
```
                        spatial_cov=sp_cov,#spatial covariate matrix
                        cov_names = column_names,#columns to use from
covariates
                        a_0=dist.Normal(1,10), alpha =
dist.Beta(20,60),#set priors

beta=dist.HalfNormal(2.0),sigmax_2=dist.HalfNormal(0.25)
                       )
model.args['sp_var_mu'] = 1.

model.run_svi(lr=0.02,num_steps=10000)

100%|██████████████████| 10000/10000 [04:25<00:00, 37.63it/s, init
loss: -5939.0361, avg. loss [9501-10000]: -9777.6250]


SVI elapsed time: 280.2311267852783
```




```
geo_df = model.args['int_df'].copy()
geo_df['spatial_log_intensity'] = (par['b_0']
[geo_df['cov_ind'].values] +
                           par['f_xy'][geo_df['comp_grid_id'].values])
geo_df.plot('spatial_log_intensity',legend=True)

<AxesSubplot:>
```

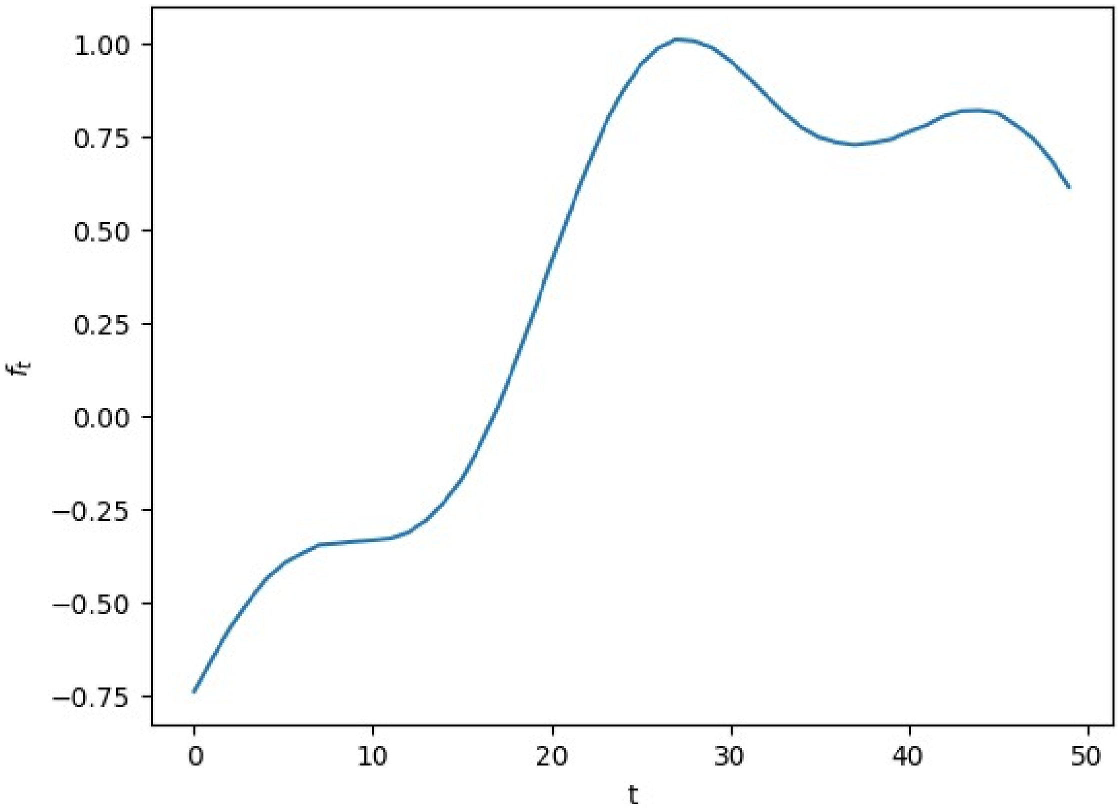



```
model.plot_spatial(include_cov=True,alpha=0.1)
```

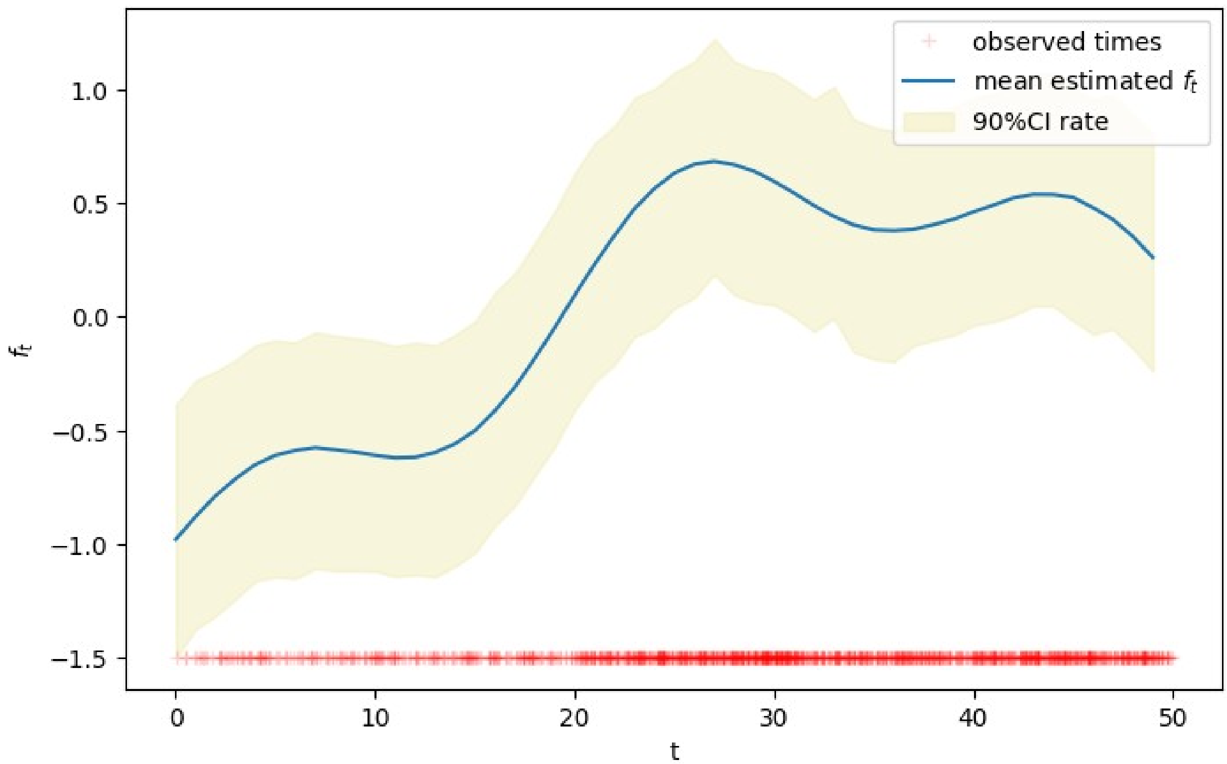



```
import matplotlib.pyplot as plt
plt.plot(par['f_t'])
plt.xlabel("t")
```

```
plt.ylabel("$f_t$")
plt.show()
```




```
model.plot_temporal()
```

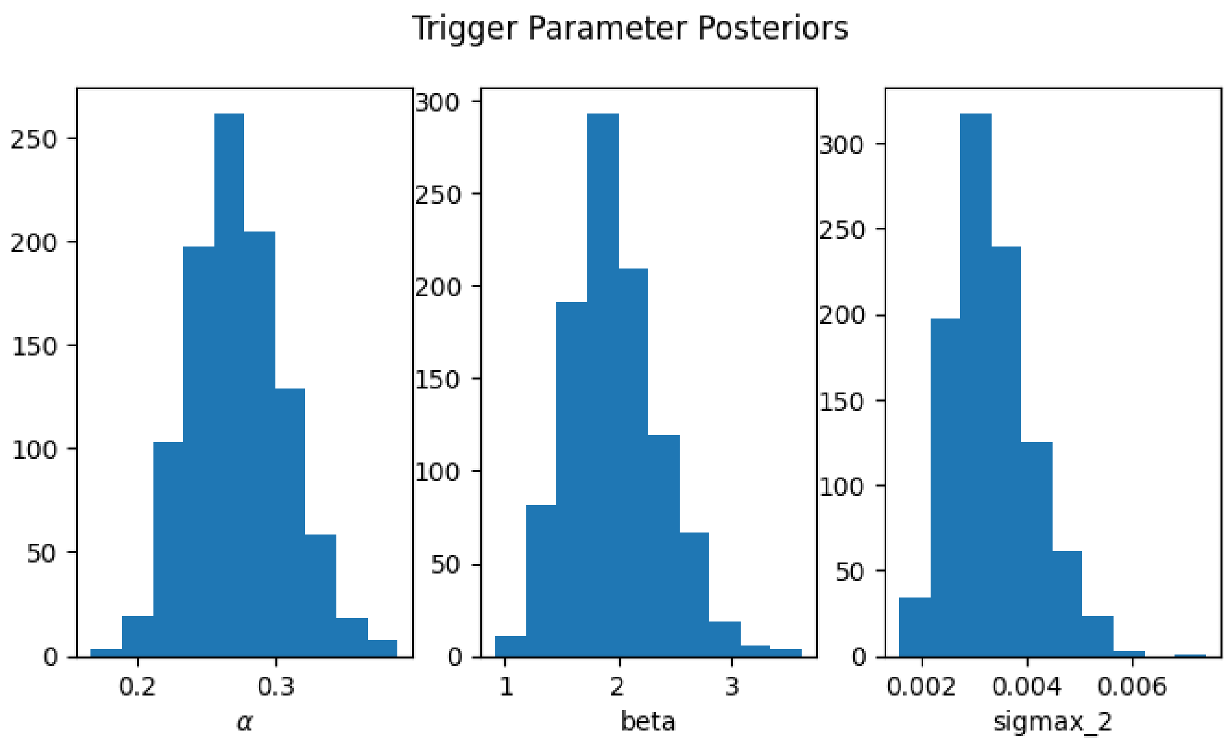



```
par['alpha'],par['beta'],par['sigmax_2']
```

```
(0.25, 2.0, 0.0025000000000000005)
```

```
model.plot_trigger_posterior()
```

	Post Mean	Post Std	P(w>0)	[0.025	0.975]
alpha	0.272750	0.034365	1.0	0.211776	0.344508
beta	1.966590	0.405064	1.0	1.286605	2.812737
sigmax_2	0.003343	0.000772	1.0	0.002135	0.005107




```
par['w']
```

```
array([ 0.03838462, -0.45854204, -0.17834039])
```

```
model.cov_weight_post_summary()
```

```
     Post Mean  Post Std  P(w>0)    [0.025    0.975]
0     0.108145  0.034223   1.000  0.043403  0.180031
1    -0.365996  0.037616   0.000 -0.439750 -0.293214
2    -0.182522  0.029363   0.000 -0.240313 -0.126062
a_0   1.105342  0.341331   0.999  0.438356  1.774482
```

Covariate Weights