Interpretable Deep Generative Spatio-Temporal Point Processes

Shixiang Zhu
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332
shixiang.zhu@gatech.edu

Shuang Li
Department of Statistics
Harvard University
Cambridge, MA 02138
shuangli@fas.harvard.edu

Zhigang Peng
School of Earth and Atmospheric Sciences
Georgia Institute of Technology
Atlanta, GA 30332
zpeng@gatech.edu

Yao Xie
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332
yao.xie@isye.gatech.edu

Abstract

We present a novel Neural Embedding Spatio-Temporal (NEST) point process model for spatio-temporal discrete event data and develop an efficient imitation learning (a type of reinforcement learning) based approach for model fitting. Despite the rapid development of one-dimensional temporal point processes for discrete event data, the study of spatial-temporal aspects of such data is relatively scarce. Our model captures complex spatio-temporal dependence between discrete events by carefully design a mixture of heterogeneous Gaussian diffusion kernels, whose parameters are parameterized by neural networks. This is the key that our model can capture intricate spatial dependence patterns and yet still lead to interpretable results as we examine maps of Gaussian diffusion kernel parameters. Furthermore, the likelihood function under our model enjoys tractable expression due to Gaussian kernel parameterization. Experiments based on real data show our method's good performance relative to the state-of-the-art and the good interpretability of NEST's result.

1 Introduction

Spatio-temporal event data has become ubiquitous, emerging from various applications. Studying generative models for discrete events data has become a hot area in machine learning and statistics: it reveals of pattern in the data, helps us to understand the data dynamic and information diffusion, as well as serves as an important step to enable subsequent machine learning tasks. Point process models (see [17] for an overview) have become a standard choice for generative models of discrete event data. In particular, the self and mutual exciting processes, also known as the Hawkes processes, are popular since they can capture past events' influence on future events over time, space, and networks.

AI for Earth Sciences Workshop at NeurIPS 2020.

Despite the rapid development of one-dimensional temporal point processes models for discrete event data, the study focusing on *spatial-temporal* aspects of such data is relatively scarce. The original works of [15, 16] develop the so-called ETAS model, which is still widely used, suggesting an exponential decaying diffusion kernel function. This model captures the seismic activities' mechanism and is convenient to fit, as the kernel function is homogeneous at all locations with the same oval shape. However, these classical models for spatio-temporal event data (usually statistical models in nature) tend to make strong parametric assumptions on the conditional intensity.

However, in specific scenarios, the simplifying spatio-temporal model based on ETAS may lack flexibility. It does not capture the anisotropic spatial influence and cannot capture the complex spatial dependence structure (see a motivating example in Appendix A). On the other hand, when developing spatio-temporal models, we typically want to generate some statistical interpretations (e.g., temporal correlation, spatial correlation), which may not be easily derived from a complete neural network model. Thus, generative model based on specifying conditional intensity of point process models is a popular approach. For example, recent works [2, 12, 11, 18, 20, 19, 24] has achieved many successes in modeling temporal event data (some with marks) which are correlated in time. It remains an open question on extending this type of approach to include the spatio-temporal point processes. One challenge is how to address the computational challenge associated with evaluating the log-likelihood function. This can be intractable for the general model without a carefully crafted structure since it requires the integration of the conditional intensity function in a continuous spatial and time-space.

In this paper, we present a novel point-process based model for spatio-temporal discrete event data. Our proposed NEST model tackles flexible representation for complex spatial dependence, interpretability, and computational efficiency, through meticulously designed neural networks with embedding capturing spatial information. We generalize the idea of using a Gaussian diffusion kernel to model spatial correlation by introducing the more flexible heterogeneous mixture of Gaussian diffusion kernels with shifts, rotations, and non-isotropic shapes. Such a model can still be efficiently represented using a handful of parameters (compared with a full neural network model such as convolutional neural networks (CNN) over space). The Gaussian diffusion kernels are parameterized by neural networks, which allows the kernels to vary continuously over locations. This is the key that our model can capture intricate spatial dependence patterns and yet still lead to interpretable results as we examine maps of Gaussian diffusion kernel parameters. As shown in Figure 6 in Appendix A, our model is able to represent arbitrary diffusion shape at different locations in contrast to ETAS developed by [14, 15, 16]. Moreover, the likelihood function under our model enjoys tractable expression due to Gaussian kernel parameterization. Experiments based on real data show our method's good performance relative to the state-of-the-art and NEST results' interpretability.

2 Proposed model

To capture the complex and heterogenous spatial dependence in discrete events, we present a novel *continuous-time and continuous-space* point process model (see Appendix B for a basic introduction in point processes), called the Neural Embedding Spatio-Temporal (NEST) model. The NEST uses the flexible neural network structure to represent the conditional intensity's spatial heterogeneity while retaining interpretability as a semi-parametric statistical model.

Spatially heterogeneous Gaussian diffusion kernel We start by specifying the conditional probability of the point process model, as it will uniquely specify the joint distribution of a sequence of events. First, to obtain a similar interpretation as the ETAS model [15], we start from a similar parametric form for the conditional intensity function

$$\lambda^*(t, s) = \lambda_0 + \sum_{j:t_j<t} \nu(t, t_j, s, s_j), \tag{1}$$

where $\lambda_0 > 0$ is a constant background rate, ν is the kernel function that captures the influence of the past events $\mathcal{H}_t$. The form of the kernel function ν determines the profile of the spatio-temporal dependence of events.

We assume the kernel function takes the form of a standard Gaussian diffusion kernel over space and decays exponential over time. To enhance the spatial expressiveness, we adopt a mixture of generalized Gaussian diffusion kernels, which is location dependent. Thus, it can capture more

complicated spatio-nonhomogeneous structure. Given all past events $\mathcal{H}_t$, we define

$$\nu(t, t', s, s') = \sum_{k=1}^{K} \phi_{s'}^{(k)} \cdot g(t, t', s, s' | \Sigma_{s'}^{(k)}, \mu_{s'}^{(k)}), \qquad \forall t' < t, s \in \mathcal{S}, \tag{2}$$

where $\{\mu_{s'}^{(k)}, \Sigma_{s'}^{(k)}\}$ are the mean and covariance matrix parameters (which we will specify later), K is the hyper-parameter that defines the number of components of the Gaussian mixture; $\phi_{s'}^{(k)} : \mathcal{S} \to \mathbb{R}$ (form specified later) is the weight for the k-th Gaussian component that satisfies $\sum_{k=1}^{K} \phi_{s'}^{(k)} = 1$, $\forall s' \in \mathcal{S}$. In the following discussions, we omit the superscript k for the notational simplicity.

Now each Gaussian diffusion kernel is defined as

$$g(t, t', s, s' | \Sigma_{s'}, \mu_{s'}) = \frac{C e^{-\beta(t-t')}}{2\pi\sqrt{|\Sigma_{s'}|}(t-t')} \cdot \exp\left\{ - \frac{(s - s' - \mu_{s'})^T \Sigma_{s'}^{-1} (s - s' - \mu_{s'})}{2(t-t')} \right\}, \tag{3}$$

where $\beta > 0$ controls the temporal decay rate, $C > 0$ is constant to control the magnitude, $\mu_s = [\mu_x(s), \mu_y(s)]^T$, and Σ_s denote the mean and covariance parameters of the diffusion kernel (which may vary over time t and "source" locations $s \in \mathcal{S}$); $|\cdot|$ denotes the determinant of a covariance matrix; Σ_s is defined as a positive semi-definite matrix

$$\Sigma_s = \begin{pmatrix} \sigma_x^2(s) & \rho(s)\sigma_x(s)\sigma_y(s) \\ \rho(s)\sigma_x(s)\sigma_y(s) & \sigma_y^2(s) \end{pmatrix}.$$

The parameters μ_s and Σ_s control the shift, rotation and shape of each Gaussian component. As shown in Figure 1, parameters $\sigma_x(s), \sigma_y(s), \rho(s)$ may vary according to the location s and jointly control the spatial structure of diffusion at s. The $\mu_x(s), \mu_y(s)$ define the offset of the center of the diffusion from the location s. Note that $\mu_x : \mathcal{S} \to \mathbb{R}$, $\mu_y : \mathcal{S} \to \mathbb{R}$, $\sigma_x : \mathcal{S} \to \mathbb{R}^+$, $\sigma_y : \mathcal{S} \to \mathbb{R}^+$, $\rho : \mathcal{S} \to (-1, 1)$ are the non-linear mappings that project location s to the parameters. To capture intricate spatial dependence, we represent such non-linear mappings from location to the parameters of Gaussian components (defined by (3)) using neural networks. An illustration for our neural network architecture has been shown in Figure 2, and the detailed description has been elaborated in Appendix C.

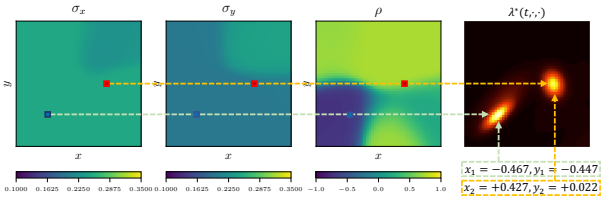


Figure 1: An example of kernel used in the NEST model: σ_x, σ_y, ρ defines a Gaussian component in the heterogeneous Gaussian diffusion kernel. The right hand side is the conditional intensity at time t, where two points occurred at location (x_1, y_1) and (x_2, y_2) have triggered the two diffusions (the bright spots) with different shapes.

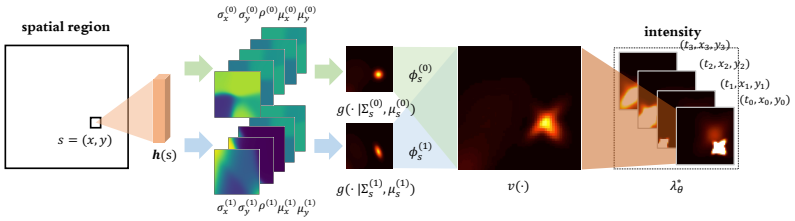


Figure 2: An illustration for NEST's neural network architecture based on a mixture of heterogeneous Gaussian diffusion kernel. Note that each Gaussian kernel is specified by neural networks, which can be viewed as summarizing the latent embedding information from data.

Comparison with ETAS model In the standard ETAS model, the kernel function is defined as a *single component whose parameters do not vary over space and time*: the kernel function (2) is simplified to $\nu(t, t', s, s') = g(t, t', s, s' | \Sigma, 0)$, where the spatial and temporal parameters are invariant $\Sigma \equiv \text{diag}\{\sigma_x^2, \sigma_y^2\}$ and $\mu_s \equiv 0$. Compared with the standard Gaussian diffusion kernel used in ETAS, here we introduce additional parameters ρ, μ_x, μ_y that allows the diffusion to shift, rotate, or stretch in the space. An example of the comparison of the spatio-temporal kernels between ETAS and NEST models is presented in Figure 6, Appendix A.

3 Computationally efficient learning

The model parameters can be estimated via maximum likelihood estimate (MLE) since we have the explicit form of the conditional intensity function. Given a sequence of events $\boldsymbol{a} = \{a_0, a_1, \ldots, a_n\}$ occurred on $(0, T] \times \mathcal{S}$ with length n, where $a_i = (t_i, s_i)$, the log-likelihood is given by

$$\ell(\theta) = \left(\sum_{i=1}^{n} \lambda_\theta^*(t_i, s_i) \right) - \int_0^T \int_{\mathcal{S}} \lambda_\theta^*(\tau, r) dr d\tau. \tag{4}$$

A crucial step to tackle the computational challenge is to evaluate the integral in (4). Here, we can obtain a closed-form expression for the likelihood function, using the following proposition. This can reduce the integral to an analytical form, which can be evaluated directly without numerical integral.

Proposition 1 (Integral of conditional intensity function)**.** *Given ordered event times* $0 = t_0 < t_1 < \cdots < t_n < t_{n+1} = T$*, for* $i = 0, \ldots, n$*,*

$$\int_{t_i}^{t_{i+1}} \int_{\mathcal{S}} \lambda_\theta^*(\tau, r) dr d\tau = \lambda_0 (t_{i+1} - t_i) |\mathcal{S}| + (1 - \epsilon) \frac{C}{\beta} \sum_{j: t_j < t_i} C_j \left(e^{-\beta(t_i - t_j)} - e^{-\beta(t_{i+1} - t_j)} \right),$$

where $C_j = \sum_{k=1}^{K} \phi_{s_j}^{(k)} (\sigma_x^{(k)}(s_j) \sigma_y^{(k)}(s_j) / |\Sigma_{s_j}^{(k)}|^{1/2})$*, and the constant*

$$\epsilon = \max_{j: t_j < t_{i+1}} \frac{\int_{t_i}^{t_{i+1}} \int_{\mathcal{S}} g(\tau, t_j, r, s_j) dr d\tau}{\int_{t_i}^{t_{i+1}} \int_{\mathbb{R}^2} g(\tau, t_j, r, s_j) dr d\tau}.$$

Since spatially, the kernel g is a Gaussian concentrated around s, when $\mathcal{S}$ is chosen sufficiently large, and most events s_i locates in the relatively interior of $\mathcal{S}$, we can ignore the marginal effect and ϵ can become a number much smaller than 1. Due to the decreased activity in the region's edges, the boundary effect is usually negligible [16]. Define $t_0 = 0$ and $t_{n+1} = T$. Since $\int_0^T \int_{\mathcal{S}} \lambda_\theta^*(\tau, r) dr d\tau = \sum_{i=0}^{n+1} \int_{t_i}^{t_{i+1}} \int_{\mathcal{S}} \lambda_\theta^*(\tau, r) dr d\tau$, using Proposition 1 we can write down the integral in the log-likelihood function in closed-form expression. Finally, the optimal parameters trained the maximum-likelihood is thus obtained by $\hat{\theta} = \text{argmax}_\theta \log \ell(\theta)$. Due to the non-convex nature of this problem, we solve the problem by stochastic gradient descent. In addition to MLE, we also introduce a more flexible, imitation learning framework for model fitting as described in Appendix D. The major benefit of this approach is that it does not rely on the likelihood function model and thus is more robust to model misspecification.

4 Numerical results

We describe our experimental setting and the real data we have used in Appendix E. More simulation studies can be found in Appendix F. We first quantitatively compare our `NEST+MLE` and `ETAS` by evaluating Mean Squared Error (MSE) of one-step-ahead prediction. The results has been summarized in Figure 3, which shows that both our methods `NEST+IL` and `NEST+MLE` outperform the state-of-the-art (`ETAS`) regarding two metrics. In addition, we also show that our method has a competitive performance even without considering spatial information in contrast to `RLPP`.

Figure 3: MSE for five methods on two real data sets.

DATA SET	RANDOM	ETAS	NEST+IL	NEST+MLE	RLPP
ROBBERY (SPACE-TIME)	.6323	.1425	**.0503**	.0649	N/A
SEISMIC (SPACE-TIME)	.2645	.0221	**.0119**	.0153	N/A
ROBBERY (TIME ONLY)	.4783	.0857	.0104	**.0094**	.0183
SEISMIC (TIME ONLY)	.1266	.0173	**.0045**	.0150	.0122

Interpretable conditional intensity To interpret the spatial dependence learned using our model, we plot the conditional intensity as a heatmap over the space at a specific time frame. For the state-of-the-art, as shown in Figure 4, we can see that the `ETAS` (the third column) captured the general pattern of the conditional intensity over the space, where regions with more events tend to have higher intensity. Comparing with the result shown in first two columns of Figure 4, our `NEST` is able to capture complex spatial pattern at different locations and the shape of the captured diffusion also differ from application to application. For 911 calls-for-service data, shown in the first row of Figure 4, the spatial influence of some robbery events diffuse to the surrounding streets and the community blocks unevenly. For seismic data, shown in the second row of Figure 4, the spatial influence of some events majorly diffuses along the earthquake fault lines.

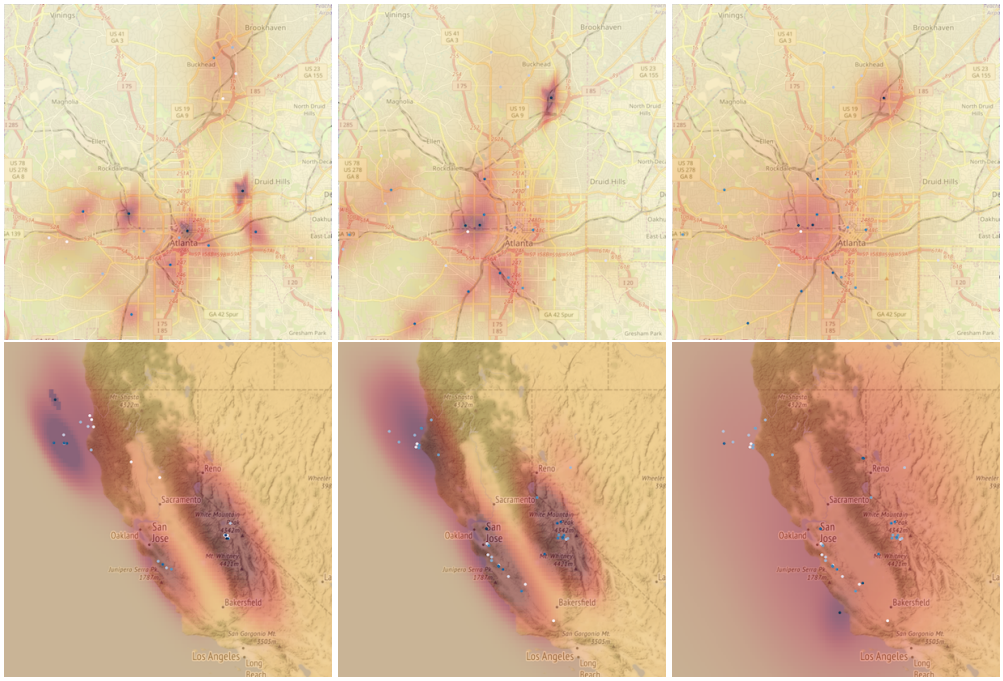


Figure 4: Snapshots of the conditional intensity for two real data sequences (crime events in Atlanta and seismic events): First and second row show snapshots of the conditional intensity for a series of robberies in Atlanta and a series of earthquakes in North of California, respectively. First two columns are generated by `NEST+MLE` ($K = 5$) and the third column is generated by `ETAS`. The color depth indicate the value of intensity. The region in darker red has higher risk to have next event happened again.

References

[1] D. J. Daley and D. Vere-Jones. *An introduction to the theory of point processes. Vol. II*. Probability and its Applications (New York). Springer, New York, second edition, 2008. General theory and structure.

[2] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1555–1564, New York, NY, USA, 2016. ACM.

[3] Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, pages 258–267, Arlington, Virginia, United States, 2015. AUAI Press.

[4] Eric W. Fox, Martin B. Short, Frederic P. Schoenberg, Kathryn D. Coronges, and Andrea L. Bertozzi. Modeling e-mail networks and inferring leadership using self-exciting point processes. *Journal of the American Statistical Association*, 111(514):564–584, 2016.

[5] Edith Gabriel, Barry Rowlingson, and Peter Diggle. stpp: An r package for plotting, simulating and analyzing spatio-temporal point patterns. *Journal of Statistical Software*, 53:1–29, 04 2013.

[6] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J. Smola. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems 19*, pages 513–520. MIT Press, 2007.

[7] ALAN G. HAWKES. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 04 1971.

[8] Beomjoon Kim and Joelle Pineau. Maximum mean discrepancy imitation learning. In *Robotics: Science and Systems*, 2013.

[9] Northern California Earthquake Data Center. UC Berkeley Seismological Laboratory. NCEDC, 2014.

[10] Erik Lewis, George Mohler, P Jeffrey Brantingham, and Andrea L Bertozzi. Self-exciting point process models of civilian deaths in iraq. *Security Journal*, 25(3):244–264, Jul 2012.

[11] Shuang Li, Shuai Xiao, Shixiang Zhu, Nan Du, Yao Xie, and Le Song. Learning temporal point processes via reinforcement learning. In *Advances in Neural Information Processing Systems 31*, pages 10781–10791. Curran Associates, Inc., 2018.

[12] Hongyuan Mei and Jason Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6757–6767, USA, 2017. Curran Associates Inc.

[13] F. Musmeci and D. Vere-Jones. A space-time clustering model for historical earthquakes. *Annals of the Institute of Statistical Mathematics*, 44(1):1–11, Mar 1992.

[14] Yosihiko Ogata. On lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31, January 1981.

[15] Yosihiko Ogata. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical Association*, 83(401):9–27, 1988.

[16] Yosihiko Ogata. Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, 50(2):379–402, 1998.

[17] Alex Reinhart. A review of self-exciting spatio-temporal point processes and their applications. *Statist. Sci.*, 33(3):299–318, 08 2018.

[18] Utkarsh Upadhyay, Abir De, and Manuel Gomez-Rodriguez. Deep reinforcement learning of marked temporal point processes. In *Advances in Neural Information Processing Systems 31*, 2018.

[19] Shuai Xiao, Mehrdad Farajtabar, Xiaojing Ye, Junchi Yan, Le Song, and Hongyuan Zha. Wasserstein learning of deep generative point process models. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 3250–3259. Curran Associates Inc., 2017.

[20] Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen M. Chu. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 1597–1603. AAAI Press, 2017.

[21] Shixiang Zhu and Yao Xie. Crime event embedding with unsupervised feature selection, 2018.

[22] Shixiang Zhu and Yao Xie. Crime incidents embedding using restricted boltzmann machines. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2376–2380, 2018.

[23] Shixiang Zhu and Yao Xie. Spatial-temporal-textual point processes with applications in crime linkage detection, 2019.

[24] Shixiang Zhu, Henry Shaowu Yuchi, and Yao Xie. Adversarial anomaly detection for marked spatio-temporal streaming data, 2019.

[25] Joseph R. Zipkin, Frederic P. Schoenberg, Kathryn Coronges, and Andrea L. Bertozzi. Point-process models of social network interactions: Parameter estimation and missing data recovery. *European Journal of Applied Mathematics*, 27(3):502–529, 2016.

A Motivating example

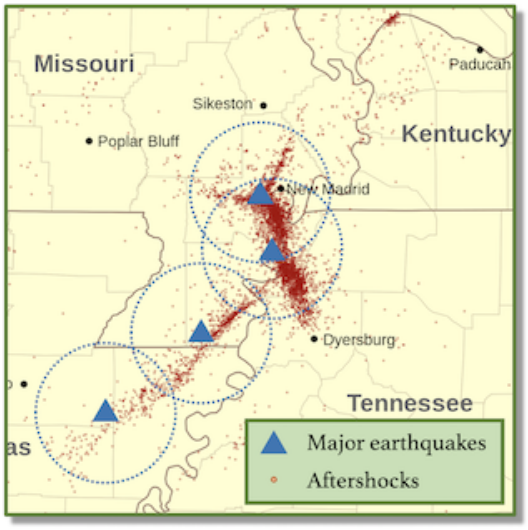


Figure 5: A motivating example of seismic activities: four major earthquakes and their aftershocks occurred in New Madrid, MO., in the United States since 1811. The blue triangles represent the major earthquakes, and the dotted circles represent the estimated aftershock regions suggested by ETAS. The small red dots represent the actual aftershocks caused by the major earthquakes. We can observe that the locations of actual aftershocks are related to the geologic structure of faults, and the vanilla ETAS model may not sufficiently capture such complex spatial dependence patterns over time.

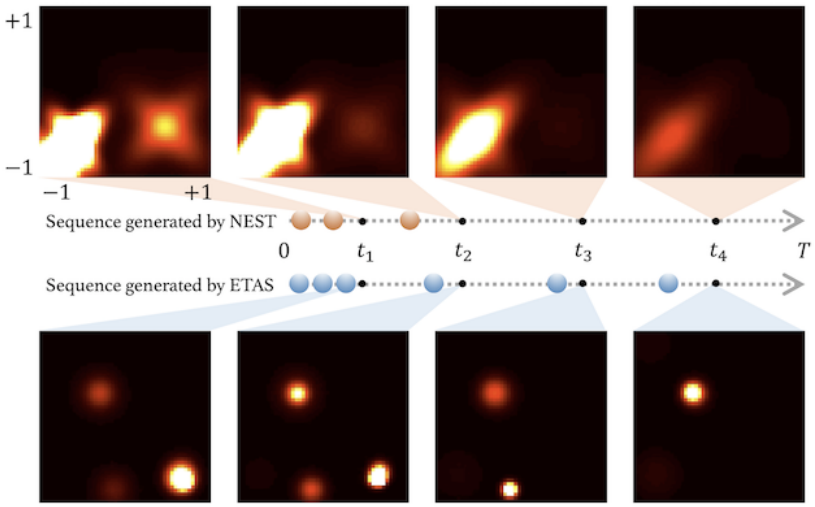


Figure 6: To illustrate the difference in the nature of ETAS and proposed NEST models, we show two series of events sequentially generated in a square region $\mathcal{S} = [-1, +1] \times [-1, +1]$ within the time horizon $[0, T]$ by ETAS and NEST, respectively. The colored balls represent events generated by the corresponding model. Snapshots were taken at time t_1, t_2, t_3, t_4 indicated by the black smaller dots to show the progression of the spatial intensity $\lambda^*(t, s), s \in \mathcal{S}$ through time for these two sequences. The brighter region in the snapshots represents a higher intensity value and is more likely to generate the next event. As self-exciting spatio-temporal point processes, the occurrence of a new event will raise the intensity in the local region instantaneously. Then the intensity of this region will decay and diffuse to the surrounding region over time.

Take earthquake event data as an example, consisting of a sequence of records of seismic activities: their times and locations. The aftershocks are minor seismic activities that are trigger by the major

earthquakes. According to the study[1], it has been shown that most of the recent seismic events occurred in New Madrid, MO, are aftershocks of four earthquakes of magnitude 7.5 in 1811. As shown in Figure 5, the distribution of the minor seismic activities is in a complex shape (clearly not "circles" or anisotropic), and the spatial correlation between seismic activities is related to the geologic structure of faults through the complex physical mechanism and usually exhibits a heterogeneous conditional intensity. For instance, most aftershocks either occur along the fault plane or along other faults within the volume affected by the mainshock's strain. This creates a spatial profile of the intensity function that we would like to capture through the model, such as the direction and shape of the intensity function at different locations, to provide useful information to geophysicists' scientific study.

B Background

In this section, we revisit the definitions of the spatio-temporal point processes (STPP) and the commonly used STPP models.

B.1 Spatio-temporal point processes (STPP)

STPP consists of an ordered sequence of events in time and location space. Assume our observation duration is T and the data is given by $\{a_1, a_2, \ldots, a_{N(T)}\}$, which sequences of events ordered in time. Each a_i is a spatio-temporal tuple $a_i = (t_i, s_i)$, where $t_i \in [0, T)$ is the time of the event and $s_i \in \mathcal{S} \subseteq \mathbb{R}^2$ is the associated location of the i-th event. We denote by $N(T)$ the number of the events in the sequence between time $[0, T)$ and in the region $\mathcal{S}$.

The joint distribution of a STPP is completely characterized by its conditional intensity function $\lambda(t, s|\mathcal{H}_t)$. Given the event history $\mathcal{H}_t = \{(t_i, s_i)|t_i < t\}$, the intensity corresponds to the probability of observing an event in an infinitesimal around (t, s):

$$\lambda(t, s|\mathcal{H}_t) = \lim_{\Delta t, \Delta s \to 0} \frac{\mathbb{E}\left[N([t, t + \Delta t) \times B(s, \Delta s))|\mathcal{H}_t\right]}{\Delta t \times |B(s, \Delta s)|},$$

where $N(A)$ is the counting measure of events over the set $A \subseteq [0, T) \times \mathcal{S}$, $B(s, \Delta s)$ denotes a Euclidean ball centered at s with radius Δs, $|\cdot|$ is the Lebesgue measure. Below, for notational simplicity, we denote the conditional intensity function $\lambda(t, s|\mathcal{H}_t)$ as $\lambda^*(t, s)$.

For instance, a type of self-exciting point processes, Hawkes processes [7] has been widely used to capture the mutual excitation among temporal events. Assuming that influence from past events are linearly additive for the current event, the conditional intensity function of a Hawkes process is defined as

$$\lambda(t|\mathcal{H}_t) = \lambda_0 + \sum_{t_i < t} \nu(t - t_i),$$

where $\lambda_0 \geq 0$ is the background intensity of events, $\nu(\cdot) \geq 0$ is the *triggering function* that captures temporal dependencies of the past events. The triggering function can be chosen in advance, for instance, in the one-dimensional case $\nu(t - t_i) = \alpha \exp\{-\beta(t - t_i)\}$.

B.2 ETAS model

The most commonly used kernel function for spatio-temporal point processes is the standard *diffusion kernel* function proposed by the Epidemic Type Aftershock-Sequences (ETAS) modeling [13], which was originally introduced to model the earthquake events, but now widely used in many other applications [15, 16, 23, 4, 10, 25]. ETAS model assumes that the influence over time and space decouples, and the influence decays exponentially over time, and over space decay only depends on distance (thus, it is a spherical model). Thus, *ETAS model does not capture the anisotropic shape of kernel.* This is a simplification and may not capture complex spatial dependence. ETAS model can also deal with scalar-valued marks (e.g., the magnitude of earthquakes), which we will not discuss here while focusing on capturing spatio-temporal interactions. One of the reasons that ETAS is a popular model is due to its interpretability.

[1] `https://www.nature.com/news/2009/091104/full/news.2009.1062.html`

C Deep neural network representation

Recall that parameters in each Gaussian component are determined by a set of non-linear mappings $\{\rho(s), \sigma_x(s), \sigma_y(s), \mu_x(s), \mu_y(s)\}$. We capture these non-linear spatial dependencies using a deep neural network through a latent embedding, which we explain below.

Assume the spatial structure at one location s is summarized by an latent embedding vector $\boldsymbol{h}(s) \in \mathbb{R}^d$, where d is the dimension of the embedding. The parameters of a Gaussian component at location s can be represented by a single-layer neural network where the input of the network is the latent embedding $\boldsymbol{h}(s)$. This layer is specified as follows. The mean parameters are specified by

$$\mu_x(s) = C_x \cdot \left(\text{sigm}(\boldsymbol{h}(s)^T W_{\mu_x} + b_{\mu_x}) - 1/2\right),$$
$$\mu_y(s) = C_y \cdot \left(\text{sigm}(\boldsymbol{h}(s)^T W_{\mu_y} + b_{\mu_y}) - 1/2\right),$$

where C_x, C_y are preset constants that control the shift of the center of Gaussian components from location s, $\text{sigm}(x) = 1/(1 + e^{-x})$ is the sigmoid function which gives an output in the range [0, 1]. The variance and the correlation parameters are specified by

$$\sigma_x(s) = \text{softplus}(\boldsymbol{h}(s)^T W_{\sigma_x} + b_{\sigma_x}),$$
$$\sigma_y(s) = \text{softplus}(\boldsymbol{h}(s)^T W_{\sigma_y} + b_{\sigma_y}),$$
$$\rho(s) = 2 \cdot \text{sigm}(\boldsymbol{h}(s)^T W_{\rho} + b_{\rho}) - 1,$$

where $\text{softplus} = \log(1 + e^x)$ is a smooth approximation of the ReLU function. The parameters in the network $\theta_w = \{W_{\sigma_x}, W_{\sigma_y}, W_{\mu_x}, W_{\mu_y}, W_{\rho}\}$ and $\theta_b = \{b_{\sigma_x}, b_{\sigma_y}, b_{\mu_x}, b_{\mu_y}, b_{\rho}\}$ are weight-vectors and biases in the output layer of the Gaussian component. Note that we omitted the superscript k of each parameter in the discussion above for the notational simplicity, but it should be understood that each Gaussian component will have its own set of parameters. Finally, the weight of each component is given by $\phi_s^{(k)}$, which is defined through the soft-max function

$$\phi_s^{(k)} = e^{\boldsymbol{h}(s)^T W_{\phi}^{(k)}} \Big/ \sum_{\kappa=1}^{K} e^{\boldsymbol{h}(s)^T W_{\phi}^{(\kappa)}}.$$

where $W_{\phi}^{(k)} \in \mathbb{R}^d$ is a weight vector to be learned. The latent embedding $\boldsymbol{h}(s)$ is characterized by another neural network defined as $\boldsymbol{h}(s) = \psi(s|\theta_h)$, where $\psi(\cdot) : \mathbb{R}^2 \to \mathbb{R}^d$ is a fully-connected multi-layer neural network function taking spatial location s as input, θ_h contains the parameters in this neural network. In our experiments later, we typically use three-layer neural networks where the width of each layer is 64.

In summary, the NEST with heterogeneous Gaussian mixture diffusion kernel is jointly parameterized by $\theta = \{\beta, \theta_h, \{\theta_w^{(k)}, \theta_b^{(k)}\}_{k=1,\dots,K}\}$. The architecture is summarized in Figure 2. In the following, we denote the conditional intensity as $\lambda_\theta^*(s, t)$ defined in (1), to make the dependence on the parameter more explicit. Note that the Gaussian diffusion kernels's parameters vary *continuously* over location, and are represented by flexible neural networks; this is the key that our model can capture complex spatial dependence patterns.

D Imitation learning approach

We now present a more flexible, imitation learning framework for model fitting. The major benefit of this approach is that it does not rely on the likelihood function model and thus is more robust to model misspecification. The model's performance is evaluated by the maximum mean discrepancy (MMD) statistic, which is a non-parametric metric that measures the divergence between two empirical distributions.

The setting of imitation learning is as follows. Assume a learner takes actions $a := (t, s) \in [0, T) \times \mathcal{S}$ sequentially in an environment according to certain *policy*, and the environment gives feedbacks (using a reward function) to the learner via observing the discrepancy between the learner actions and the demonstrations (training data) provided by an expert. In our setting, both the learner's actions and the demonstrations are over continuous-time and continuous space, which is a distinct feature of our problem.

D.1 Policy parameterization

Our desired learner policy is a probability density function of possible actions given the history $\mathcal{H}_t$. We define such function as $\pi(t, s) : [0, T) \times \mathcal{S} \to [0, 1]$, which assigns a probability to the next event at any possible location and time. Let the last event time before T be t_n, and thus the next possible event is denoted as (t_{n+1}, s_{n+1}). The definition of the policy is

$$\pi(t, s) = \mathbb{P}((t_{n+1}, s_{n+1}) \in [t, t + dt) \times B(s, \Delta s)|\mathcal{H}_t).$$

We will show that the policy can be explicitly related to the the conditional intensity function of the point process.

Lemma 1. *A spatial temporal point process which generate new samples according to $\pi(t, s)$ has the corresponding conditional intensity function*

$$\lambda^*_\theta(t, s) = \frac{\pi(t, s)}{1 - \int_0^t \int_{\mathcal{S}} \pi(\tau, r) d\tau dr}. \tag{5}$$

From Lemma 1, we can obtain the learner policy as the following proposition:

Proposition 2 (Learner policy related to conditional intensity)**.** *Given the conditional intensity function $\lambda^*_\theta(t, s)$, the learner policy of a STPP on $[0, T) \times \mathcal{S}$ is given by*

$$\pi_\theta(t, s) = \lambda^*_\theta(t, s) \cdot \exp\left\{ - \int_{t_n}^{t} \int_{\mathcal{S}} \lambda^*_\theta(\tau, r) d\tau dr \right\}.$$

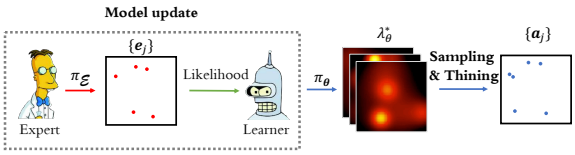
Thus, this naturally gives us a policy parameterization in a principled fashion: the learner policy π_θ is parameterized by θ based on the proposed heterogeneous Gaussian mixture diffusion kernel in Section 2. Note that using Proposition 1, which gives an explicit formula for the integral required in the exponent, the policy can be exactly evaluated even a deep neural network is included in the model.

D.2 Imitation learning objective

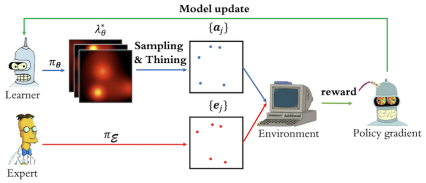
Now assume the training data is generated by an expert policy π_E, where the subscript E denotes "expert". Given a reward function $r(\cdot)$, the goal is to find an optimal policy that maximize the expected reward

$$\max_\theta J(\theta) := \mathbb{E}_{\boldsymbol{a} \sim \pi_\theta} \left[\sum_{i=1}^{n_a} r(a_i) \right], \tag{6}$$

where $\boldsymbol{a} = \{a_1, \ldots, a_{n_\alpha}\}$ is one sampled roll-out from policy π_θ. Note that n_a can be different for different roll-out samples.



(a) Maximum likelihood estimation (MLE) for NEST



(b) Imitation learning (IL) for NEST

Figure 7: Comparison between the maximum likelihood and the imitation learning approaches. The main difference is that the MLE measure the "likelihood" of the data under a model and thus can be more susceptible to model misspecification, whereas our IL approach measures the actual divergence between the training data and the sequence generated from the model using MMD statistic without relying on model assumptions.

D.3 Reward function

Consider the minimax formulation of imitation learning, which chooses the worst-case reward function that will give the maximum divergence between the rewards earned by the expert policy and the best learner policy:

$$\max_{r\in\mathcal{F}}\left(\mathbb{E}_{\epsilon\sim\pi_{\mathcal{E}}}\left[\sum_{i=1}^{n_e} r(e_i)\right] - \max_{\pi_\theta\in\mathcal{G}} \mathbb{E}_{\alpha\sim\pi_\theta}\left[\sum_{i=1}^{n_a} r(a_i)\right]\right),$$

where $\mathcal{G}$ is the family of all candidate policies π_θ and $\mathcal{F}$ is the family class for reward function r in reproducing kernel Hilbert space (RKHS).

We adopt a data-driven approach to solve the optimization problem and find the worst-case reward. This is related to inverse reinforcement learning; we borrow the idea of MMD reward in [8, 6, 3, 11] and generalize it from a simple one-dimensional temporal point process to the more complex spatio-temporal setting. Suppose we are given training samples $\{\boldsymbol{e}_j\}$, $j = 1, 2, \ldots, M_E$, which are the demonstrations provided by the expert π_E, where each $\boldsymbol{e}_j = \{e_0^{(j)}, e_1^{(j)}, \ldots, e_{n_j}^{(j)}\}$ denotes a single demonstration. Also, we are given samples generated by the learner: let the trajectories generated by the learner π_θ denoted by $\{\boldsymbol{a}_j\}$, $j = 1, 2, \ldots, M_L$, where each trajectory $\boldsymbol{a}_j = \{a_0^{(j)}, a_1^{(j)}, \ldots, a_{n_j}^{(j)}\}$ denotes a single action trajectory. We will discuss how to generate samples in the following sub-section.

Using a similar argument as proving Theorem 1 in [11] and based on kernel embedding, we can obtain analytical expression for the worst-case reward function based on samples, which is given by

$$\hat{r}(a) \propto \frac{1}{M_E}\sum_{j=1}^{M_E}\sum_{i=1}^{n_j} k(e_i^{(j)}, a) - \frac{1}{M_L}\sum_{k=1}^{M_L}\sum_{l=1}^{n_k} k(a_l^{(k)}, a). \tag{7}$$

where $k(\cdot,\cdot)$ is a reproducing kernel Hilbert space (RKHS) kernel. Here we use Gaussian kernel function, which achieves good experimental results in both synthetic data and real data.

Using samples generated from learner policy, the gradient of $J(\theta)$ with respect to θ can be computed by using policy gradient with variance reduction [11],

$$\nabla_\theta J(\theta) \approx \frac{1}{M_E}\sum_{j=1}^{M_E}\left[\sum_{i=1}^{n_j}\left(\nabla_\theta \log \pi_\theta(a_i)\cdot \hat{r}(a_i)\right)\right].$$

The gradient of policy $\nabla_\theta \log \pi_\theta(a_i)$ can be computed analytically in closed-form, since it is specified by the conditional intensity in Proposition 2, and the conditional intensity is fully specified by the neural networks architecture of NEST as we discussed in the Section 2.

D.4 Sampling from STPP using thinning algorithm

An important step in the imitation learning approach is to generate samples from our proposed model, i.e., $a \sim \pi_\theta$, given the history $\mathcal{H}_t$. Here, we develop an efficient sampling strategy to achieve good computational efficiency. We need to sample a point tuple $a = (t, s)$ according to the conditional intensity defined by (1). A default way to simulate point processes is to use the thinning algorithm [5, 1]. However, the vanilla thinning algorithm suffers from low sampling efficiency as it needs to sample in the space $|\mathcal{S}| \times [0, T)$ uniformly with the upper limit of the conditional intensity $\bar{\lambda}$ and only very few of candidate points will be retained in the end. In particular, given the parameter θ, the procedure's computing complexity increases exponentially with the size of the sampling space. To improve sampling efficiency, we propose an efficient thinning algorithm summarized in Algorithm 1. The "proposal" density is a non-homogeneous STPP, whose intensity function is defined from the previous iterations. This analogous to the idea of importance sampling [14].

E Experimental settings

E.1 Baselines and evaluation metrics

In this section, we compare our proposed Neural Embedding Spatio-Temporal (`NEST`) with a benchmark and several state-of-the-art methods in the field. These include (1) `Random uniform` that

Algorithm 1: Efficient thinning algorithm for STPP

input $\theta, \lambda_0, \beta, T, \mathcal{S}$;
output A set of events α ordered by time.;
Initialize $\alpha = \emptyset$, $t = 0$, $s \sim \texttt{uniform}(\mathcal{S})$.;
while $t < T$ **do**
  $u, D \sim \texttt{uniform}(0, 1)$; $s \sim \texttt{uniform}(\mathcal{S})$;
  $\bar{\lambda} \leftarrow \lambda_0 + \sum_{(\tau,r)\in\alpha} \nu(t, \tau, s_n, r)$;
  $t \leftarrow t - \ln u / \bar{\lambda}$;
  Compute $\lambda^*_\theta(t, s)$ from (5);
  if $D\bar{\lambda} > \lambda^*_\theta(t, s)$ **then**
    $\alpha \leftarrow \alpha \cup \{(t, s)\}$; $s_n \leftarrow s$;
  end
end

randomly makes actions in the action space; (2) Epidemic Type Aftershock-Sequences (`ETAS`) with standard diffusion kernel, which is currently the most widely used approach in spatio-temporal event data modeling. For ETAS, the parameters are estimated by maximum likelihood estimate (MLE); (3) reinforcement learning point processes model (`RLPP`) [11] is for modeling temporal point process only, which cannot be easily generalized to spatio-temporal models. (4) `NEST+IL` is our approach using imitation learning; (5) `NEST+MLE` is our approach where the parameters are estimated by MLE.

To evaluate the performance of algorithms (i.e., various generative models), we adopt two performance metrics: (1) the average Mean Square Error (MSE) of the *one-step ahead prediction* (which, thus, is a prediction performance metric). The *one-step ahead prediction* is obtained by performing Algorithm 1 given the current intensity $\lambda^*(t, s)$ and the past events; (2) the maximum mean discrepancy (MMD) metric between the real observed sequences and the generated sequences from the models, as specified in Section D (thus, this measures how good the generative model give a specific training method). For synthetic data, we also compare the recovered parameters against the true parameters, which is used for constructing the generator and generating the corresponding synthetic data.

E.2 Real data description

We test our approaches on two real-world data sets: Atlanta 911 calls-for-services data (provide by the Atlanta Police Department to us under a data agreement) and Northern California seismic data [9]. For the ease of comparison, we normalize the space region of both data sets to the same space $T \times \mathcal{S}$ where $T = (0, 10]$ and $\mathcal{S} = [-1, 1] \times [-1, 1]$. The detailed description of two data sets is as follows.

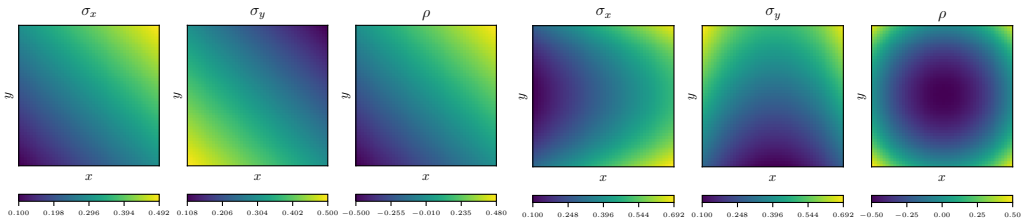
Atlanta 911 calls-for-service data. The 911 calls-for-service data in Atlanta from the end of 2015 to 2017 is provided by the Atlanta Police Department (this data set has been previously used to validate the crime linkage detection algorithm [21, 22, 23]). We extract 7,831 reported robbery from the data set since robbers usually follow particular *modus operandi* (M.O.), where criminal spots and times tend to have a causal relationship with each other. Each robbery report is associated with the time (accurate to the second) and the geolocation (latitude and longitude) indicating when and where the robbery occurred. We consider each series of robbery as a sequence.

Northern California seismic data. The Northern California Earthquake Data Center (NCEDC) provides public time series data [9] that comes from broadband, short period, strong motion seismic sensors, and GPS, and other geophysical sensors. We extract 16,401 seismic records that have a magnitude larger than 3.0 from 1978 to 2018 in Northern California and partition the data into multiple sequences every quarter. To test our model, we only take advantage of time and geolocation in the record.

F Simulation study

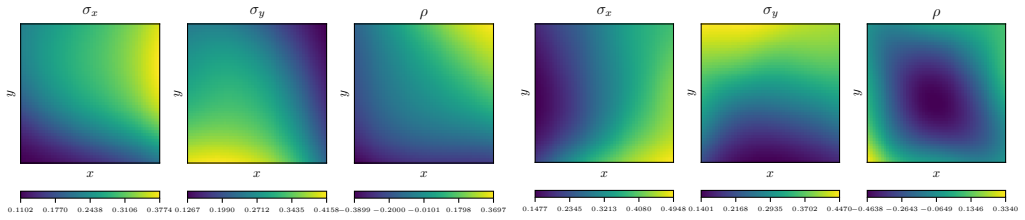
We evaluate our method on two synthetic data sets. These two data sets are generated by `NEST` with artificial parameters shown in Figure 8a and 8b, respectively, where parameters in Figure 8a are

linearly related to their spatial locations and parameters in Figure 8b are non-linearly related their spatial locations. In both data sets, there are 5,000 sequences with an average length of 191, and 80%, 20% sequences are used for training and testing. The time and location of events have been normalized to $T = 10$ and $\mathcal{S} = [-1, +1] \times [-1, +1]$. As an ablation study, we specify the model with only one Gaussian component and three hidden layers in the neural network. Each layer contains 64 neurons, and randomly takes 40 sequences as a batch. The model is trained by both MLE, and IL approaches.



(a) ground truth for synthetic data set 1

(b) ground truth for synthetic data set 2



(c) recovered result for synthetic data set 1

(d) recovered result for synthetic data set 2

Figure 8: Simulation results on two sets of synthetic data. (8a): The ground truth of the Gaussian parameter σ_x, σ_y, ρ in the synthetic data set 1; (8b): The ground truth of the Gaussian parameter σ_x, σ_y, ρ in the synthetic data set 2; (8c): The recovered Gaussian parameters using the synthetic data set 1; (8d): The recovered Gaussian parameters using the synthetic data set 2.

Table 1: MSE for five methods on two synthetic data sets.

DATA SET	RANDOM	ETAS	NEST+IL	NEST+MLE	RLPP
SYNTHETIC DATA 1 (SPACE-TIME)	.2781	.0433	**.0075**	.0134	N/A
SYNTHETIC DATA 2 (SPACE-TIME)	.3327	.0512	**.0124**	.0321	N/A
SYNTHETIC DATA 1 (TIME-ONLY)	.1734	.0135	**.0021**	.0048	.0146
SYNTHETIC DATA 2 (TIME-ONLY)	.2147	.0323	**.0036**	.0055	.0341

We report the average MSE of the one-step-ahead prediction in Figure 1 for each method on both two synthetic datasets. As we can see from the table, our model (based on different training methods)(`NEST+ML` and `NEXT+IL`) outperform other methods on two synthetic data sets in terms of prediction error. Since the ground truth of the model is known, we also visualize the parameters in the Gaussian component in comparison to the true parameters. Figure 8a and Figure 8a show the true parameters that are used to construct the generators and generate two synthetic data sets. We fit our `NEST` model using these two synthetic data sets separately and obtain the corresponding parameters. Figure 8c and Figure 8d show the recovered parameters learned from two synthetic data sets. It shows that the spatial distribution of the recovered parameters is similar to the true parameters. It also confirms that our model can capture the underlying spatial pattern in both linear and non-linear parameter distributions.