SPATIO-TEMPORAL POINT PROCESSES WITH DEEP NON-STATIONARY KERNELS

Anonymous authors
Paper under double-blind review

ABSTRACT

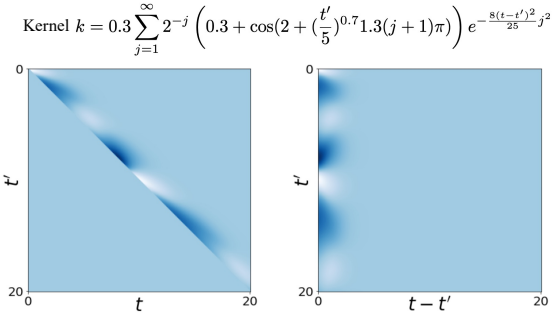
Deep neural networks, especially recurrent neural network (RNN) models, have become a popular tool for analyzing point process data. Despite the powerful expressiveness and memorizing ability of RNN models, they may not successfully model sophisticated non-stationary dependencies among data due to the recurrent structure. Meanwhile, another type of deep model for point process data was recently proposed, which represents the influence kernel rather than the intensity function by neural networks. This paper develops a deep non-stationary influence kernel for spatio-temporal point processes with a novel parameterization that enables us to well approximate complicated kernels in a low-rank form. A log-barrier penalty is introduced during network optimization to maintain the non-negativity of conditional intensity. Our new method can also be extended to model high-dimensional marks, and we demonstrate outstanding performance gain on real police text data. The new approach significantly reduces the model and computational complexities, and the benefits of kernel recovery and event prediction are demonstrated using synthetic and real point process data.

1 INTRODUCTION

Sequential events with timestamps and associated information such as location or category are ubiquitous in modern scientific fields and real-world applications. To model such discrete events in continuous time and space, spatio-temporal point processes (STPPs) are widely used in a diverse range of domains, including modeling earthquakes (Ogata, 1988; 1998), the spread of the epidemic and infectious diseases (Schoenberg et al., 2019; Dong et al., 2021), ambulance demand (Zhou et al., 2015), and wildfire propagation (Hering et al., 2009). The distribution of the event occurrence in STPPs is fully specified by a so-called *conditional intensity function*.

It is tricky for models to accurately capture the underlying mechanism of event occurrence in general STPPs and meanwhile maintain the model efficiency. Specific parametric forms of intensity are proposed in seminal works (Hawkes, 1971; Ogata, 1988) to tackle the issue of computational complexity, which requires solving multivariate integral in likelihood. They use an exponentially decaying *influence kernel* that measures the influence of a past event on a current event and assume the influences of past events are linearly additive. Such a parametric form of influence kernel limits the model practicality in modern applications with spatio-temporal *non-stationarity*: For instance, the disease is likely to spread more slowly in sparsely-populated rural than in densely-populated urban areas, which makes the event influence to be shift-variant. Recent works incorporate fast-developing neural networks into point processes to alleviate the oversimplifications and capture complicated underlying mechanisms of event occurrence. Vanilla RNN (Du et al., 2016) and LSTM (Mei and Eisner, 2017) have been used by taking advantage of their memorizing ability of event dependencies. Self-attentive structure (Zhang et al., 2020; Zuo et al., 2020) is introduced to improve the limited ability of RNN to capture sophisticated long-term dependencies (Bengio et al., 1994) among data. While these models are flexible, a specific functional form of influence decaying over time is still assumed which lacks model expressiveness and interpretability in modeling influence propagation. More importantly, only model-restricted discretized spatial distributions can be employed, making them not applicable in general STPPs.

To overcome the above model restrictions, STPPs in continuous spatio-temporal domain without any specific parametric assumption on kernel (Zhu et al., 2022) or conditional intensity function



(a) Kernel matrix of $k(t^{'}, t)$ with rank = 298

(b) Kernel matrix of $k(t^{'}, t-t^{'})$ with rank = 7

Figure 1: Example of low-rank kernel approximation. (a) shows the kernel matrix of k with rank 298 parameterized with t', t. (b) shows the kernel matrix of k with rank 7 parameterized with $t', t-t'$. Each kernel matrix is evaluated on a 300×300 uniform grid.

(Chen et al., 2020) have been studied. They use neural networks to achieve a powerful and flexible representation of STPPs. In particular, Zhu et al. (2022) employs a finite-rank spectral kernel decomposition, modeling the influence of the history and the current event using different neural networks, thus being able to model non-stationary dependencies. At the same time, they still maintain the model interpretability by taking the self-exciting form of standard Hawkes processes. One major drawback of this neural kernel-based approach lies in the great computational burden of solving the intractable integral with neural networks over multi-dimensional spatio-temporal space.

Contribution. In this paper, we propose a deep non-stationary kernel (referred to as the `DNSK`) for STPPs that can capture complicated event dependencies while significantly improving computational efficiency. A low-rank kernel representation using neural networks plays a pivotal role in the success of our model. Specifically, our approach addresses the above concerns of current models by making the following contributions:

- We replace the current event in influence kernel with the "displacement" between the past and current ones. Then the kernel is decomposed into neural network-based basis functions of historical events and the "displacement". Independent evaluation of history and displacement influences enables us to go beyond the parametric kernel in the standard Hawkes process and parametric decaying function in RNN-based models, capturing the most general spatio-temporal non-stationarity in the generative process.
- The proposed kernel parameterization enables significant low-rank kernel representation. As shown in Figure 1, the rank of the temporal kernel function k is reduced from 298 to 7 by changing the parameterization from (t', t) to $(t', t-t')$. Note that our efficient low-rank form can always be employed when dealing with practical tasks with complicated influence kernels.
- A log-barrier method is introduced in model optimization to guarantee a non-negative conditional intensity function. Compared with adding positive activation to intensity, our barrier method maintains the linearity of the conditional intensity and thus keeps the model interpretability of the kernel-based method. Based on that, we propose a computationally efficient strategy for model estimation.
- We use synthetic and real data experiments to show the superior performance of our proposed methods in both model recovery and event prediction. Our model enjoys particular performance gain and is highly interpretable when involving non-stationary STPPs, and can be conveniently adopted for tasks with high-dimensional marks.

1.1 Related Works

The original work of A. Hawkes (Hawkes, 1971) provides classic self-exciting point processes for temporal events, which express the conditional intensity function with an influence kernel and a base rate. Ogata (1988; 1998) propose a parametric form of spatio-temporal influence kernel which enjoys strong model interpretability and efficiency. However, such simple parametric forms own limited expressiveness in characterizing the complex events' dynamic in modern applications.

Neural networks have been widely adopted in point processes(Xiao et al., 2017; Chen et al., 2020). Du et al. (2016) incorporates recurrent neural networks in point processes modeling through an encoder of historical times and categorical event marks. Mei and Eisner (2017) use a continuous-time invariant of LSTM to model event influence with exponential decay over time. These RNN-based models may be unable to capture complicated event dependencies due to the recurrent structure. Zhang et al. (2020); Zuo et al. (2020) introduce self-attentive structures into point processes for their capability to memorize long-term influence by dealing with an event sequence as a whole. The main limitation is that they assume a dot-product-based score function and assume linearly decaying of event influence. Omi et al. (2019) propose a fully-connected neural network to model the cumulative intensity function to go beyond parametric decaying influence. However, the event embeddings are still generated by RNN, and fitting cumulative intensity function by neural networks lacks model interpretability. Note that all the above models tackle temporal events with categorical marks, which are inapplicable in continuous time and location space.

Recent works adopt neural networks in learning the influence kernel function. The kernel introduced in Okawa et al. (2021) uses neural networks to model the latent dynamic of time interval but still assumes an exponentially decaying influence over time. Zhu et al. (2022) proposes a kernel representation using spectral decomposition and represents feature functions using deep neural networks to harvest powerful model expressiveness when dealing with marked event data. Our method considers an alternative novel kernel representation that allows the general kernel to be expressed further low-rankly.

2 Background

Spatio-temporal point processes (STPPs) (Reinhart, 2018; Moller and Waagepetersen, 2003) have been widely used to model sequences of events that happen in continuous time and space. A realization of an STPP is an event stream denoted as $\mathcal{H} = \{(t_i, s_i)\}_{i=1}^n$ of event times $t_i \in [0, T] \subset \mathbb{R}$ and locations $s_i \in \mathcal{S} \subset \mathbb{R}^{d_S}$ with random number of event n. The conditional intensity function of an STPP given the observed history $\mathcal{H}_t = \{(t_i, s_i) | t_i < t, t_i \in \mathcal{H}\}$ is defined as

$$\lambda\left(t, s | \mathcal{H}_t\right) = \lim_{\Delta t \downarrow 0, \Delta s \downarrow 0} \frac{\mathbb{E}\left[\mathbb{N}([t, t+\Delta t], B(s, \Delta s)) | \mathcal{H}_t\right]}{|B(s, \Delta s)| \Delta t}, \tag{1}$$

with the counting measure $\mathbb{N}$ defined as the number of events occurring in $[t, t+\Delta t] \times B(s, \Delta s)$. In the following, we omit the dependency on history $\mathcal{H}_t$ and use common shorthand $\lambda(t, s)$. The joint log-likelihood of observing $\mathcal{H}$ on $[0, T] \times \mathcal{S}$ is given by (Daley et al., 2003)

$$\log \ell(\mathcal{H}) = \sum_{i=1}^{n} \log \lambda\left(t_i, s_i\right) - \int_0^T \int_{\mathcal{S}} \lambda(t, s) ds dt \tag{2}$$

Neural point processes parameterize the conditional intensity function by taking advantage of recurrent neural networks (RNNs). In Du et al. (2016), an input vector $\boldsymbol{x}_i$ which extracts the information of event t_i and the associated event attributes m_i (can be event mark or location) is fed into the RNN. A hidden state vector $\boldsymbol{h}_i$ is updated by $\boldsymbol{h}_i = \rho(\boldsymbol{h}_{i-1}, \boldsymbol{x}_i)$, where $\rho(\cdot)$ is a mapping fulfilled by recurrent neural network operations. The conditional intensity function on $(t_i, t_{i+1}]$ is then defined as

$$\lambda(t) = f(t, \boldsymbol{h}_i)$$

where f is an exponential transformation that guarantees a positive intensity. In Mei and Eisner (2017) the RNN is replaced by a continuous-time LSTM module with hidden states $\boldsymbol{h}(t)$ defined on $[0, T]$ and a Softplus function f. Attention-based models are introduced in Zuo et al. (2020); Zhang et al. (2020) to overcome the inability of RNNs to capture sophisticated event dependencies due to their recurrent structures.

Hawkes process (Hawkes, 1971) is a well-known generalization model of event streams. Assuming the influences from the past events are linearly additive, the conditional intensity function takes the form of

$$\lambda(t, s) = \mu + \sum_{(t', s') \in \mathcal{H}_t} k(t', t, s', s), \tag{3}$$

where k is an influence kernel function that captures event interactions. Commonly the kernel function is assumed to be *stationary*, that is, k only depends on $t - t'$ and $s - s'$, which limits the model expressivity. In this work, we aim to capture complicated non-stationarity in spatio-temporal event dependencies by leveraging the strong approximation power of neural networks in kernel fitting.

3 LOW-RANK DEEP NON-STATIONARY KERNEL

Due to the existence of intricate dependencies between events, it will be challenging to choose the form of kernel function that achieves great model expressiveness while enjoying high model efficiency. In this section, we introduce a unified point process model with a low-rank deep non-stationary kernel capable of modeling complex heterogeneity in events' influence over spatio-temporal space efficiently.

3.1 KERNEL WITH HISTORY AND SPATIO-TEMPORAL DISPLACEMENT

We parameterize the influence kernel k using the past event (t', s') and the "displacement" $(t - t', s - s')$ to current event (t, s) and then decompose it into different basis functions:

$$k(t', t - t', s', s - s') = \sum_{r=1}^{R} \sum_{l=1}^{L} \alpha_{lr} \psi_l(t') \varphi_l(t - t') u_r(s') v_r(s - s'). \tag{4}$$

Here $\{\psi_l, \varphi_l : [0, T] \to \mathbb{R}, l = 1, \ldots, L\}$ are two sets of temporal basis functions that characterize the temporal influence of event at t' and the decaying effect brought by elapsed time $t - t'$. Similarly, spatial basis functions $\{u_r, v_r : \mathcal{S} \to \mathbb{R}, r = 1, \ldots, R\}$ capture the spatial influence of event at s' and the decayed influence after spreading over the displacement of $s - s'$. The corresponding weight α_{lr} at different spatio-temporal ranks combines each set of basis functions into a weighted summation, leading to the final expression of influence kernel k.

To further enhance the model expressiveness, we use a fully-connected neural network to represent each basis function. The history or displacement is taken as the input and fed through multiple hidden layers equipped with Softplus non-linear activation function. To allow for inhibiting influence from past events (negative value of influence kernel k), we use a linear output layer for each neural network. For an influence kernel with temporal rank L and spatial rank R, we need $2(L + R)$ independent neural networks for modeling.

The benefits of our proposed kernel framework lies in the following: (i) The representation parameterize the event dependency from two perspectives of original influence patterns of history (modeled by $\{\psi_l\}_{l=1}^{L}$, $\{u_r\}_{r=1}^{R}$) and decaying patterns of that influence (modeled by $\{\varphi_l\}_{l=1}^{L}$, $\{v_r\}_{r=1}^{R}$). As shown in Figure 1, such a parameterization can well approximate a majority of complicated influence kernel even with infinite rank in practice, thus making our model be further low-rank when compared to Zhu et al. (2022). (ii) The non-stationarity of events' influence over spatio-temporal space can be conveniently captured by non-constant $\{\psi_l\}_{l=1}^{L}$, $\{u_r\}_{r=1}^{R}$, making the model applicable in modeling general STPPs. (iii) The influence decaying patterns are characterized by $\{\varphi_l\}_{l=1}^{L}$, $\{v_r\}_{r=1}^{R}$ which go beyond simple parametric forms. When the events' influence has finite range, *i.e.* there exist $\tau_{\max}$ and $a_{\max}$ such that the influence decays to zero if $|t - t'| > \tau_{\max}$ or $||s - s'|| > a_{\max}$, we can restrict the parameterization of $\{\varphi_l\}_{l=1}^{L}$ and $\{v_r\}_{r=1}^{R}$ on a local domain $[0, \tau_{\max}] \times B(0, a_{\max})$ instead of $[0, T] \times \mathcal{S}$, which further reduce the model complexity. Details of choosing kernel and neural network architectures are described in Section 5.

3.2 KERNEL INCORPORATING MARKS

One of the salient features of our spatio-temporal kernel framework is that it can be conveniently adopted in modeling marked point processes with additional sets of mark basis function $\{g_q, h_q\}_{q=1}^{Q}$. We need to modify the influence kernel function k accordingly as following:

$$k(t', t - t', s', s - s', m', m) = \sum_{q=1}^{Q} \sum_{r=1}^{R} \sum_{l=1}^{L} \alpha_{lrq} \psi_l(t') \varphi_l(t - t') u_r(s') v_r(s - s') g_q(m') h_q(m).$$

Here $m', m \in \mathcal{M} \subset \mathbb{R}^{d_{\mathcal{M}}}$ and $\{g_q, h_q : \mathcal{M} \to \mathbb{R}, q = 1, \ldots, Q\}$ represented by independent neural networks model the influence of historical mark m' and current mark m, respectively. Since the mark space $\mathcal{M}$ is always categorical and the difference between m' and m is of little practical meaning, we use g_q and h_q to model m' and m separately instead of modeling $m - m'$.

4 EFFICIENT COMPUTATION OF MODEL ESTIMATION

We consider model optimization through Maximum likelihood estimation (MLE) (Reinhart, 2018). The resulting conditional intensity function could now be negative by allowing inhibiting historical

influence. A common approach to guarantee the non-negativity is to adopt a nonlinear positive activation function in the conditional intensity (Du et al., 2016; Zhu et al., 2022). However, the integral of such a nonlinear intensity over spatio-temporal space is intractable and computationally expensive. To tackle that, we first introduce a log-barrier to the MLE optimization problem to guarantee the non-negativity of conditional intensity function λ and maintain its linearity. Then we provide a computationally efficient strategy that benefits from the linearity of the conditional intensity. Details about model optimization and computation with marks are in Appendix C.

4.1 Model optimization with log-barrier

The constrained MLE optimization problem for model parameter estimation can be formulated as:

$$\min_{\theta} -\ell(\theta), s.t. -\lambda(t,s) \leq 0, \ \forall t \in [0,T], \forall s \in \mathcal{S}.$$

We introduce a log-barrier method (Boyd et al., 2004) to ensure the non-negativity of λ. We penalize the values of λ on a dense enough grid $\mathcal{U}_{\text{bar},t} \times \mathcal{U}_{\text{bar},s} \subset [0,T] \times \mathcal{S}$. The log-barrier is defined as

$$p(\theta, b) := -\frac{1}{|\mathcal{U}_{\text{bar},t} \times \mathcal{U}_{\text{bar},s}|} \sum_{c_t=1}^{|\mathcal{U}_{\text{bar},t}|} \sum_{c_s=1}^{|\mathcal{U}_{\text{bar},s}|} \log(\lambda(t_{c_t}, s_{c_s}) - b), \tag{5}$$

where c_t, c_s indicate the index of the gird, and b is a lower bound of conditional intensity function on the grid to guarantee the feasibility of logarithm operation. The MLE optimization problem can be finally formulated as

$$\begin{aligned} \min_{\theta} \mathcal{L}(\theta) :&= -\ell(\theta) + \frac{1}{s} p(\theta, b) \\ &= -\left(\sum_{i=1}^{n} \log \lambda(t_i, s_i) - \int_0^T \int_{\mathcal{S}} \lambda(t,s) ds dt \right) \\ &\quad - \frac{1}{s|\mathcal{U}_{\text{bar},t} \times \mathcal{U}_{\text{bar},s}|} \sum_{c_t=1}^{|\mathcal{U}_{\text{bar},t}|} \sum_{c_s=1}^{|\mathcal{U}_{\text{bar},s}|} \log(\lambda(t_{c_t}, s_{c_s}) - b), \end{aligned} \tag{6}$$

where s is a weight to control the trade-off between log-likelihood and log-barrier. Both s and b can be set accordingly during the learning procedure. Details can be found in Appendix A.

Note that previous works (Du et al., 2016; Mei and Eisner, 2017; Pan et al., 2021; Zuo et al., 2020; Zhu et al., 2022) use a scaled positive transformation to guarantee non-negativity conditional intensity function. Compared with them, the log-barrier method preserves the linearity of the conditional intensity function. As shown in Table 1, such a log-barrier method enables efficient model computation (See more details in Section 4.2) and enhance the model recovery power.

4.2 Model computation

The log-likelihood calculation of the STPPs defined in equation 2 is often intractable and time-consuming due to the existence of neural networks and the double integral term. Given a sequence of events $\{x_i = (t_i, s_i)\}_{i=1}^n$ of number n, the complexity of neural network evaluation is of $\mathcal{O}(n^2)$ for the term of log-summation and of $\mathcal{O}(Kn)$ $(K \gg n)$ when using numerical integration for the double integral term with K sampled points in a multi-dimensional space. In the following, we circumvent the calculation difficulty by proposing an efficient computation for $\mathcal{L}(\theta)$ with complexity $\mathcal{O}(n)$ of neural network evaluation through a domain discretization strategy.

Computation of log-summation. The first log-summation term in equation 2 can be written as:

$$\sum_{i=1}^{n} \log \lambda(t_i, s_i) = \sum_{i=1}^{n} \log \left(\mu + \sum_{t_j < t_i} \sum_{r=1}^{R} \sum_{l=1}^{L} \alpha_{lr} \psi_l(t_j) \varphi_l(t_i - t_j) u_r(s_j) v_r(s_i - s_j) \right). \tag{7}$$

Note that each ψ_l only needs to be evaluated at event time $\{t_i\}_{i=1}^n$ and each u_r is evaluated at all the event location $\{s_i\}_{i=1}^n$. To avoid the redundant evaluations of φ_l over every pair of (t_i, t_j), we set up a uniform grid $\mathcal{U}_t$ over time horizon $[0, \tau_{\max}]$ and evaluate φ_l on the grid. The value of $\varphi_l(t_j - t_i)$

can be obtained by linear interpolation with values on two adjacent grid points of $t_j - t_i$. By doing so, we only need to evaluate φ_l for $|\mathcal{U}_t|$ times on the grids. Note that φ_l can be simply feed with 0 when $t_j - t_i > \tau_{\max}$ without any neural network evaluation.

We directly evaluate $v_r(s_i - s_j)$ since numerical interpolation is less accurate in location space. We do not need to evaluate every pair of index (i, j). Instead, we have $I := \{(i,j) \mid v_r(s_i - s_j) \text{ will be computed}\} = \{(i,j) \mid t_j < t_i \leq t_j + \tau_{\max}\} \cap \{(i,j) \mid \|s_i - s_j\| \leq a_{\max}\}$. We feed 0 to other pairs of (i, j).

Computation of integral. We note that it is unnecessary to perform numerical integration for the whole conditional intensity function since the design of the kernel allows us to decompose the desired integral to integrating basis functions. Specifically, we have

$$\begin{aligned}\int_0^T \int_{\mathcal{S}} \lambda(t,s)dsdt &= \mu|\mathcal{S}|T + \sum_{i=1}^n \int_0^T \int_{\mathcal{S}} I(t_i < t)k(t_i, t, s_i, s)dsdt \\ &= \mu|\mathcal{S}|T + \sum_{i=1}^n \sum_{r=1}^R u_r(s_i) \int_{\mathcal{S}} v_r(s - s_i)ds \sum_{l=1}^L \alpha_{rl}\psi_l(t_i) \int_0^{T-t_i} \varphi_l(t)dt.\end{aligned} \tag{8}$$

To compute the integral of φ_l, we take the advantage of the previously evaluated value of φ_l on the grid $\mathcal{U}_t$. Let $F_l(t) := \int_0^t \varphi_l(\tau)d\tau$. Then $F_l(T - t_i)$ can be computed by the linear interpolation of values of F_l at two adjacent grid points (in $\mathcal{U}_t$) of $T - t_i$. In particular, F_l evaluated on $\mathcal{U}_t$ equals to the cumulative sum of φ_l divided by the grid width.

The integral of v_r can be estimated based on a grid $\mathcal{U}_s$ in $B^{d_{\mathcal{S}}}(0, a_{\max})$ since it decays outside the ball. For each s_i, $\int_{\mathcal{S}} v_r(s - s_i)ds = \int_{B^{d_{\mathcal{S}}}(0,a_{\max}) \cap \{\mathcal{S} - s_i\}} v_r(s)ds$, where $\mathcal{S} - s_i := \{s' \mid s' = s - s_i, s \in \mathcal{S}\}$. Thus the integral is well estimated with the evaluations of v_r on grid set $\mathcal{U}_s \cap \mathcal{S} - s_i$. Note that in practice we only evaluate v_r on $\mathcal{U}_s$ once and use subsets of the evaluations for different s_i.

Computation of log-barrier. The barrier term $p(\theta, b)$ is calculated in a similar way as equation 7 by replacing (t_i, s_i, μ) with $(t_{c_t}, s_{c_s}, \mu - b)$, *i.e.* we use interpolation to calculate $\varphi_l(t_{c_t} - t_j)$ and evaluate v_r on a subset of $\{(s_{c_s}, s_j)\}_{c_s=1,\ldots,|\mathcal{U}_{\text{bar},s}|, j=1,\ldots,n}$.

4.3 Computational complexity

The evaluation of $\{u_r\}_{r=1}^R$ and $\{\psi_l\}_{l=1}^L$ over n events costs $\mathcal{O}((R + L)n)$ complexity. The evaluation of $\{\varphi_l\}_{l=1}^L$ is of $\mathcal{O}(L|\mathcal{U}_t|)$ complexity since it relies on the grid $\mathcal{U}_t$. The evaluation of $\{v_r\}_{r=1}^R$ costs no more than $\mathcal{O}(RC\tau_{\max}n) + \mathcal{O}(R|\mathcal{U}_s|)$ complexity. We note that $L, R, \tau_{\max}, |\mathcal{U}_t|, |\mathcal{U}_s|$ are all constant that much less than event number n, thus the overall computation complexity will be $\mathcal{O}(n)$. We compare the model training time per epoch for a 1D synthetic data set and a 3D synthetic data set in Table 1 to demonstrate the efficiency improvement of our model. Note that the log-barrier method also plays a crucial role in the feasibility of efficient computation by maintaining the linearity of the conditional intensity function. More details about the computation complexity analysis are in Appendix B.

Table 1: Comparison of model training time per epoch on a 1D and a 3D synthetic data. Time is measured in second.

	1D Data set 1		3D Data set 1	
Model	#Parameters	Training time	#Parameters	Training time
NSMPP	2576	60.040	9415	170.004
DNSK+Barrier	2307	1.299	9228	3.529

5 Experiments

We use large-scale synthetic and real data sets to demonstrate the superior performance of our model and present the results in this section. More experiment details can be found in Appendix.

Baselines. We compare our method (DNSK+Barrier) with: (i)RMTPP (RMTPP) (Du et al., 2016); (ii) Neural Hawkes (NH) (Mei and Eisner, 2017); (iii) Transformer Hawkes process (THP) (Zuo et al., 2020); (iv) Parametric Hawkes process (PHP) with exponentially decaying spatio-temporal kernel; (v) Neual spectral marked point processes (NSMPP) (Zhu et al., 2022); (vi) DNSK without log-barrier but with a non-negative Softplus activation function (DNSK+Softplus). We note that RMTPP, NH and THP directly model conditional intensity function using neural networks while

Table 2: Synthetic data results. Testing log-likelihood per event (higher the better) and MRE (lower the better) are reported in each entry.

Testing log likelihood/Mean Relative Error	1D Data set 1	1D Data set 2	1D Data set 3	2D Data set 1	3D Data set 1	3D Data set 2
RMTPP	$-0.467_{(0.009)}/0.086$	$-2.591_{(0.010)}/0.259$	$-1.353_{(0.002)}/0.212$	/	/	/
NH	$-0.459_{(0.002)}/0.068$	$-2.543_{(0.007)}/0.092$	$-1.315_{(0.032)}/0.204$	/	/	/
THP	$-0.491_{(0.049)}/0.633$	$\mathbf{-1.929}_{(0.080)}/0.910$	$\mathbf{-0.841}_{(0.011)}/0.598$	/	/	/
PHP	$-0.451_{(0.001)}/0.093$	$-2.725_{(0.002)}/0.181$	$-1.524_{(0.015)}/0.223$	$-2.737_{(0.003)}/0.306$	$-2.683_{(0.002)}/0.291$	$-2.424_{(0.003)}/0.282$
NSMPP	$-0.462_{(0.010)}/0.078$	$-2.638_{(0.008)}/0.162$	$-1.473_{(0.033)}/0.164$	$-2.807_{(0.016)}/0.156$	$-2.637_{(0.012)}/0.193$	$-2.381_{(0.012)}/0.280$
DNSK+Softplus	$-0.455_{(0.003)}/0.045$	$-2.539_{(0.002)}/0.037$	$-1.300_{(0.004)}/0.104$	$-2.592_{(0.002)}/0.042$	$-2.515_{(0.002)}/0.088$	$-2.279_{(0.003)}/0.119$
DNSK+Barrier	$\mathbf{-0.451}_{(0.002)}/\mathbf{0.039}$	$-2.536_{(0.003)}/\mathbf{0.016}$	$-1.298_{(0.002)}/\mathbf{0.031}$	$\mathbf{-2.585}_{(0.001)}/\mathbf{0.028}$	$\mathbf{-2.498}_{(0.003)}/\mathbf{0.021}$	$\mathbf{-2.251}_{(0.001)}/\mathbf{0.082}$

[1] RMTPP, NH, and THP can only deal with temporal data with categorical marks, thus not applicable in 2D or 3D case.

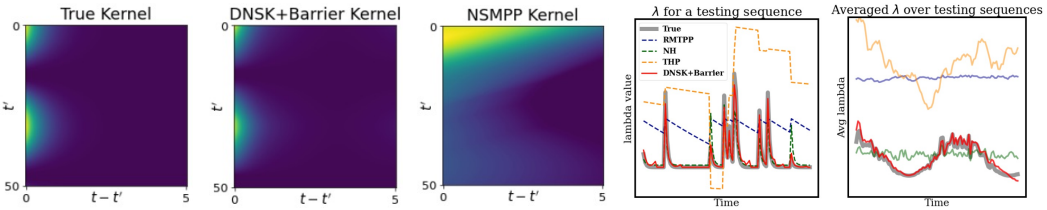


Figure 2: Kernel recovery results of 1d non-stationary data set. The first three panels show the true kernel, the kernel learned by DNSK+Barrier, and the kernel learned by NSMPP, respectively. The last two line charts show the predicted conditional intensity for a testing sequence and the averaged conditional intensity for all testing sequences, which can be regarded as the event intensity by taking the expectation of the history.

NSMPP, DNSK+Softplus, and DNSK+Barrier learn the influence kernel in the framework of equation 3. The model parameters are estimated using the training data via Adam optimization method (Kingma and Ba, 2014). Details of training procedure can be found in Appendix.

5.1 Synthetic Experiments

Synthetic data sets. To show the effectiveness of DNSK+Barrier, we generate three temporal data sets and three spatio-temporal data sets, which are generated with the following true kernels: (i) 1D exponential kernel (ii) 1D non-stationary kernel; (iii) 1D infinite rank kernel; (iv) 2D exponential kernel; (v) 3D non-stationary inhibition kernel; (vi) 3D non-stationary mixture kernel. All the data sets are generated using thinning algorithm in Daley and Vere-Jones (2008). Each data set is composed of 2000 sequences. Details of these kernels and data generation can be found in Appendix.

We consider two performance metrics for testing data evaluation: Mean relative error (MRE) of the predicted intensity and log-likelihood. The true and predicted $\lambda^*(x), \hat{\lambda}(x)$ can be calculated using equation 4 with true and learned kernel. The MRE for one test trajectory is defined as $\int_{\mathcal{X}} \frac{|\lambda^*(x)-\hat{\lambda}(x)|}{\lambda^*(x)} dx$ and the averaged MRE over all test trajectories is reported. The log-likelihood for observing each testing sequence can be computed according to equation 2, and the average predictive log-likelihood per event is reported. The log-likelihood shows the model's goodness-of-fit, and the intensity evaluation further reflects the model's ability to recover the underlying mechanism of event occurrence and predict the future.

The heat maps in Figure 2 visualize the results of kernel recovery for DNSK+Barrier and NSMPP on 1D Data set 2. We found that DNSK+Barrier recovers the ground truth significantly better than NSMPP, benefiting from the low-rank kernel approximation. Two line charts in Figure 2 present the recovered intensities with the true ones (black curves). It demonstrates that our method can accurately capture the unknown temporal dynamics of event series. In particular, the average conditional intensity λ over multiple testing sequences shows the model's ability to recover the unconditional event intensity over time. While DNSK+Barrier successfully captures the temporal non-stationarity in event intensity, both RMTPP and NH fails to characterize such non-stationarity by showing a flat curve of the averaged intensity. The problem can be alleviated by introducing Transformer structure instead of RNN. However, THP overestimates the event intensity by a large margin.

Tabel 2 summarized the quantitative results of testing log-likelihood and MRE. It shows that DNSK+Barrier has superior predictive performance against baselines in characterizing the dynamics of data generation in spatio-temporal space. Specifically, with evidently over-parameterization for

Table 3: Results of real data sets with time and categorical marks. We compare the testing log-likelihood per event, event time prediction error, and event type prediction accuracy of RMTPP, NH, THP, and DNSK on each data set.

	Financial			**StackOverflow**		
Model	Testing ℓ	Time RMSE	Type Accuracy	Testing ℓ	Time RMSE	Type Accuracy
RMTPP	-3.890	1.560	0.620	-2.600	9.780	0.459
NH	-3.600	1.560	0.622	-2.550	9.830	0.463
THP	-1.110	0.930	0.626	**0.042**	4.990	0.470
DNSK+Barrier	**–0.709**	**0.153**	**0.630**	-2.089	**4.833**	**0.508**

1D Data set 1 generated by a stationary exponentially decaying kernel, our model can still approximate the kernel and recover the true conditional intensity without overfitting, which shows the adaptiveness of our model. Moreover, DNSK+Barrier enjoys outstanding performance gain when learning a diverse variety of complicated non-stationary kernels. The comparison between DNSK+Softplus and DNSK+Barrier proves that the model with log-barrier achieves a better recovery performance by maintaining the linearity of the conditional intensity. Note that THP provides a higher testing log-likelihood in two non-stationary 1D data sets, but with significant intensity deviation from the MRE and the figures. More results about kernel and intensity recovery can be found in Appendix D.

5.2 Real data results

Real data sets. We provide a comprehensive evaluation of our approach on several real-world data sets: we first use two popular data sets containing time-stamped events with categorical marks to demonstrate the robustness of DNSK+Barrier in marked point process: (i) *Financial Transactions.* (Du et al., 2016). This data set contains transaction records of a stock in one day with time unit milliseconds and the action (mark) of each transaction. We partition the events into different sequences by time stamps. (ii) *StackOverflow* (Leskovec and Krevl, 2014): The data is collected from the website StackOverflow over two years, containing reward records for users who promote engagement in the community. Each user's reward history is treated as a sequence.

Then we demonstrate the practical versatility of the model using the following spatio-temporal data sets: (i) *Southern California earthquake data* provided by Southern California Earthquake Data Center (SCEDC) contains time and location information of earthquakes in Southern California. We collect 19,414 records from 1999 to 2019 with magnitude larger than 2.5 and partition the data into multiple sequences by month with average length of 40.2. (ii) *Atlanta robbery & burglary data*. Atlanta Police Department (APD) provides a proprietary data source for city crime. We extract 3420 reported robberies and 14958 burglaries with time and location from 2013 to 2019. Two crime types are preprocessed as separate data sets on a 10-day basis with average lengths of 13.7 and 58.7.

Finally, the model's ability to tackle high-dimensional marks is evaluated with *Atlanta textual crime data.* The proprietary data set provided by APD records 4644 crime incidents from 2016 to 2017 with time, location, and comprehensive text descriptions. The text information is preprocessed by TF-IDF technique, leading to a 5012-dimensional mark for each event.

Table 3 summarizes the results of models dealing with categorical marks. Event time and type prediction are evaluated by Root Mean Square Error (RMSE) and accuracy, respectively. We can see that DNSK+Barrier outperforms the baselines in all prediction tasks by providing less time RMSE and higher type accuracy.

For real spatio-temporal data, we report average predictive log-likelihood per event for the testing set since MRE is not applicable. Besides, we perform *online prediction* for earthquake data to demonstrate the model predicting ability. The probability density function $f(t, s)$ which represents the conditional probability that the next event will occur at (t, s) given history $\mathcal{H}_t$ can be written as $f(t, s) = \lambda(t, s) \exp\left(-\int_{\mathcal{S}} \int_{t_n}^{t} \lambda(\tau, w) d\tau dw\right)$. The predicted time and location of the next event can be computed as $\mathbb{E}\left[t_{n+1} | \mathcal{H}_t\right] = \int_{t_n}^{\infty} t \int_{\mathcal{S}} f(t, s) ds dt, \quad \mathbb{E}\left[s_{n+1} | \mathcal{H}_t\right] = \int_{\mathcal{S}} s \int_{t_n}^{\infty} f(t, s) dt ds$. We feed the past events except the last one into the estimated model, then predict the time and location of the next event. The mean absolute error (MAE) of the predictions is computed. The

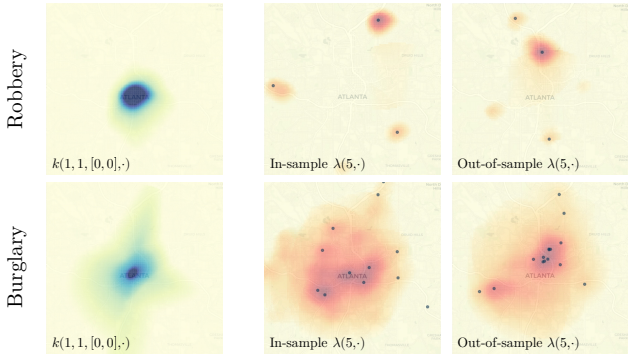


Figure 3: Kernel fitting and intensity prediction for ATL robbery & burglary data. The first column shows the learned influence kernels for two crime categories at fixed geolocation in downtown ATL. Four panels on the right show the predicted conditional intensity for two crimes over space at an in-sample and out-of-sample time, respectively. The dots represent reported events on that day.

Table 4: Real data results with spatial and mark information.

Model	South California Earthquake			Atlanta Robbery	Atlanta Burglary	Atlanta Textual Crime
	Testing ℓ	Time MAE	Location MAE	Testing ℓ	Testing ℓ	Testing ℓ
`PHP`	$-2.048_{(0.093)}$	8.132	0.487	$-2.641_{(0.006)}$	$0.382_{(0.005)}$	$-10.542_{(0.016)}$
`NSMPP`	$-4.152_{(0.187)}$	6.780	0.455	$-2.753_{(0.113)}$	$0.304_{(0.098)}$	$-8.694_{(0.336)}$
`DNSK+Barrier`	$\mathbf{-1.751}_{(0.080)}$	**1.474**	**0.431**	$\mathbf{-2.255}_{(0.008)}$	$\mathbf{0.519}_{(0.012)}$	$\mathbf{-5.279}_{(0.044)}$

quantitative results in Table 4 show that `DNSK+Barrier` provides more accurate predictions than other alternatives with higher event log-likelihood.

To demonstrate our model's interpretability and power to capture heterogeneous data characteristics, we visualize the learned influence kernels and predicted conditional intensity for two crime categories in Figure 3. The first column shows kernel evaluations at fixed geolocation in downtown Atlanta which intuitively reflect the spatial influence of crimes in that neighborhood. The influence of a robbery in the downtown area is more intensive but regional, while a burglary which is hard to be responded to by police in time would impact a larger neighborhood along major highways of Atlanta. We also provide the predicted conditional intensity over space for two crimes. As we can observe, `DNSK+Barrier` captures the occurrence of events in regions with a higher crime rate, and crimes of the same category happening in different regions would influence their neighborhoods differently. We note that this example emphasizes the significant ability of the proposed method to recover data non-stationarity with different sequence lengths, and improve the limited ability of other neural network-based methods (`RMTPP`, `NH`, and `THP`) to interpret their models in practice.

For Atlanta textual crime data, we borrow the idea in Zhu and Xie (2022) by encoding the highly sparse TF-IDF representation into a binary mark vector with much lower dimension $d = 50$ using Restricted Boltzmann Machine (RBM) (Fischer and Igel, 2012). The text embedding technique helps denoise the unstructured text information and transform the intractable integration to a summation over binary mark space $\mathcal{M}$. The average testing log-likelihoods per event for each model are reported in Table 4. The results show that `DNSK+Barrier` outperforms (`PHP`) in Zhu and Xie (2022) and `NSMPP` by achieving a higher testing log-likelihood. We visualize the basis functions of learned influence kernel by `DNSK+Barrier` in Figure 6 in Appendix.

6 Conclusion

We propose a deep non-stationary kernel for spatio-temporal point processes using a different parameterization scheme, which enables the model to be further low-rank when learning complicated influence kernel, thus significantly reducing the model complexity. The non-negativity of the solution is guaranteed by a log-barrier method which maintains the linearity of the conditional intensity function. Based on that, we propose a computationally efficient strategy for model estimation. The superior performance of our model is demonstrated using synthetic and real data sets.

REFERENCES

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

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

Frederic Paik Schoenberg, Marc Hoffmann, and Ryan J Harrigan. A recursive point process model for infectious diseases. *Annals of the Institute of Statistical Mathematics*, 71(5):1271–1287, 2019.

Zheng Dong, Shixiang Zhu, Yao Xie, Jorge Mateu, and Francisco J Rodríguez-Cortés. Non-stationary spatio-temporal point process modeling for high-resolution covid-19 data. *arXiv preprint arXiv:2109.09029*, 2021.

Zhengyi Zhou, David S Matteson, Dawn B Woodard, Shane G Henderson, and Athanasios C Micheas. A spatio-temporal point process model for ambulance demand. *Journal of the American Statistical Association*, 110(509):6–15, 2015.

Amanda S Hering, Cynthia L Bell, and Marc G Genton. Modeling spatio-temporal wildfire ignition point patterns. *Environmental and Ecological Statistics*, 16(2):225–250, 2009.

Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58 (1):83–90, 1971.

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*, pages 1555–1564, 2016.

Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems*, 30, 2017.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes process. In *International conference on machine learning*, pages 11183–11193. PMLR, 2020.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International conference on machine learning*, pages 11692–11702. PMLR, 2020.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

Shixiang Zhu, Haoyun Wang, Zheng Dong, Xiuyuan Cheng, and Yao Xie. Neural spectral marked point processes. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=0rcbOaoBXbq.

Ricky TQ Chen, Brandon Amos, and Maximilian Nickel. Neural spatio-temporal point processes. *arXiv preprint arXiv:2011.04583*, 2020.

Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen Chu. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Takahiro Omi, Kazuyuki Aihara, et al. Fully neural network based model for general temporal point processes. *Advances in neural information processing systems*, 32, 2019.

Maya Okawa, Tomoharu Iwata, Yusuke Tanaka, Hiroyuki Toda, Takeshi Kurashima, and Hisashi Kashima. Dynamic hawkes processes for discovering time-evolving communities' states behind diffusion processes. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1276–1286, 2021.

Alex Reinhart. A review of self-exciting spatio-temporal point processes and their applications. *Statistical Science*, 33(3):299–318, 2018.

Jesper Moller and Rasmus Plenge Waagepetersen. *Statistical inference and simulation for spatial point processes*. CRC press, 2003.

Daryl J Daley, David Vere-Jones, et al. *An introduction to the theory of point processes: volume I: elementary theory and methods*. Springer, 2003.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Zhimeng Pan, Zheng Wang, Jeff M Phillips, and Shandian Zhe. Self-adaptable point processes with nonparametric time decays. *Advances in Neural Information Processing Systems*, 34, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

Daryl J Daley and David Vere-Jones. *An Introduction to the Theory of Point Processes. Volume II: General Theory and Structure*. Springer, 2008.

Jure Leskovec and Andrej Krevl. Snap datasets: Stanford large network dataset collection, 2014.

Shixiang Zhu and Yao Xie. Spatiotemporal-textual point processes for crime linkage detection. *The Annals of Applied Statistics*, 16(2):1151–1170, 2022.

Asja Fischer and Christian Igel. An introduction to restricted boltzmann machines. In *Iberoamerican congress on pattern recognition*, pages 14–36. Springer, 2012.

PA W Lewis and Gerald S Shedler. Simulation of nonhomogeneous poisson processes by thinning. *Naval research logistics quarterly*, 26(3):403–413, 1979.