SPATIO-TEMPORAL POINT PROCESSES WITH DEEP NON-STATIONARY KERNELS

Anonymous authors
Paper under double-blind review

ABSTRACT

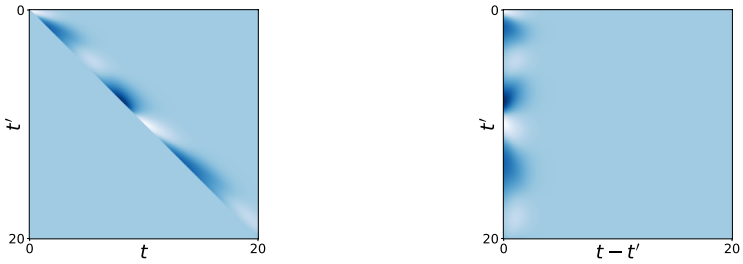
Point process data are becoming ubiquitous in modern applications, such as social networks, health care, and finance. Despite the powerful expressiveness of the popular recurrent neural network (RNN) models for point process data, they may not successfully capture sophisticated non-stationary dependencies in the data due to their recurrent structures. Another popular type of deep model for point process data is based on representing the influence kernel (rather than the intensity function) by neural networks. We take the latter approach and develop a new deep non-stationary influence kernel that can model non-stationary spatio-temporal point processes. The main idea is to approximate the influence kernel with a novel and general low-rank decomposition, enabling efficient representation through deep neural networks and computational efficiency and better performance. We also take a new approach to maintain the non-negativity constraint of the conditional intensity by introducing a log-barrier penalty. We demonstrate our proposed method's good performance and computational efficiency compared with the state-of-the-art on simulated and real data.

1 INTRODUCTION

Point process data, consisting of sequential events with timestamps and associated information such as location or category, are ubiquitous in modern scientific fields and real-world applications. The distribution of events is of great scientific and practical interest, both for predicting new events and understanding the events' generative dynamics (Reinhart, 2018). 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 infectious diseases (Schoenberg et al., 2019; Dong et al., 2021), and wildfire propagation (Hering et al., 2009).

A modeling challenge is to accurately capture the underlying generative model of event occurrence in general spatio-temporal point processes (STPP) while maintaining the model efficiency. Specific parametric forms of conditional intensity are proposed in seminal works of Hawkes process (Hawkes, 1971; Ogata, 1988) to tackle the issue of computational complexity in STPPs, which requires evaluating the complex multivariate integral in the likelihood function. They use an exponentially decaying *influence kernel* to measure the influence of a past event over time and assume the influence of all past events is positive and linearly additive. Despite computational simplicity (since the integral of the likelihood function is avoided), such a parametric form limits the model's practicality in modern applications.

Recent models use neural networks in modeling point processes to capture complicated event occurrences. RNN (Du et al., 2016) and LSTM (Mei and Eisner, 2017) have been used by taking advantage of their representation power and capability in capturing event temporal dependencies. However, the recurrent structures of RNN-based models cannot capture long-range dependency (Bengio et al., 1994) and attention-based structure (Zhang et al., 2020; Zuo et al., 2020) is introduced to address such limitations of RNN. Despite much development, existing models still cannot sufficiently capture spatio-temporal non-stationarity, which are common in real-world data (Graham et al., 2013; Dong et al., 2021). Moreover, while RNN-type models may produce strong prediction performance, the models consist of general forms of network layers and the modeling power relies on the hidden states, thus often not easily interpretable.



(a) Kernel matrix of $k(t', t)$ with rank 298. (b) Kernel matrix of $k(t', t - t')$ with rank 7.

Figure 1: An example: equivalent representation of kernel by "displacement", from $k(t', t)$ to $k(t', t - t')$ can significantly decrease the rank of the kernel function: from 298 to 7, where t' and t represent the historical event time and the current time, respectively. On fitting the one-dimensional synthetic data generated by k, the model parameterized with $(t', t - t')$ outperforms the model parameterized with (t', t). See Section 5 and Appendix C for experiment and formulation details.

A promising approach to overcome the above model restrictions is point process models that combine statistical models with neural network representation, such as Zhu et al. (2022) and Chen et al. (2020), to enjoy both the interpretability and expressive power of neural networks. In particular, the idea is to represent the (possibly non-stationary) influence kernel based on a spectral decomposition and represent the basis functions using neural networks. However, the prior work (Zhu et al., 2022) is not specifically designed for non-stationary kernel and the low-rank representation can be made significantly more efficient, which is the main focus of this paper.

Contribution. In this paper, we develop a non-stationary kernel (referred to as DNSK) for (possibly non-stationary) spatio-temporal processes that enjoy efficient low-rank representation, which leads to much improved computational efficiency and predictive performance. The construction is based on an interesting observation that by reparameterize the influence kernel from the original form of $k(t', t)$, (where t' is the historical even time, and t is the current time) to an equivalent form $k(t', t - t')$ (which thus is parameterized by the displacement $t - t'$ instead), the rank can be reduced significantly, as shown in Figure 1. This observation inspired us to design a much more efficient representation of the non-stationary point processes with much fewer basis functions to represent the same kernel.

In summary, the contributions of our paper include

- We introduce an efficient low-rank representation of the influence kernel based on a novel "displacement" re-parameterization. Our representation can well-approximate a large class of general non-stationary influence kernels and is generalizable to spatio-temporal kernels (also potentially to data with high-dimensional marks). Efficient representation leads to lower computational cost and better prediction power, as demonstrated in our experiments.
- In model fitting, we introduce a log-barrier penalty term in the objective function to ensure the non-negative conditional intensity function so the model is statistically meaningful, and the problem is numerically stable. This approach also enables the model to learn general influence functions (that can have negative values), which is a drastic improvement from existing influence kernel-based methods that require the kernel functions to be non-negative.
- Using extensive synthetic and real data experiments, we show the competitive performance of our proposed methods in both model recovery and event prediction compared with the state-of-the-art, such as the RNN-based and transformer-based models.

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 and 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 random events that happen in continuous time and space. Let $\mathcal{H} = \{(t_i, s_i)\}_{i=1}^n$ denote the event stream with time $t_i \in [0, T] \subset \mathbb{R}$ and location $s_i \in \mathcal{S} \subset \mathbb{R}^{d_S}$ of ith event. The event number n is also random. Given the observed history $\mathcal{H}_t = \{(t_i, s_i) \in \mathcal{H} | t_i < t\}$ before time t, an STPP is then fully characterized by the conditional intensity function

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

where $B(s, \Delta s)$ is a ball centered at $s \in \mathbb{R}^{d_S}$ with radius Δs, and the counting measure $\mathbb{N}$ is defined as the number of events occurring in $[t, t+\Delta t] \times B(s, \Delta s) \subset \mathbb{R}^{d_S+1}$. Naturally $\lambda(t, s|\mathcal{H}_t) \geq 0$ for any arbitrary t and s. In the following, we omit the dependency on history $\mathcal{H}_t$ and use common shorthand $\lambda(t, s)$. The log-likelihood of observing $\mathcal{H}$ on $[0, T] \times \mathcal{S}$ is given by (Daley et al., 2003)

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

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 $\lambda(t) = \delta(t, \boldsymbol{h}_i)$, where δ 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 δ. 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 generalized point process model. Assuming the influences from 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 intricate dependencies between events, it is 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

For the influence kernel function $k(t', t, s', s)$, by using the displacements in t and s as variables, we first re-parameterize the kernel as $k(t', t-t', s', s-s')$, where the minus in $s-s'$ refers to element-wise difference between s and s' when $d_{\mathcal{S}} > 1$. Then we achieve a finite-rank decomposed representation based on (truncated) singular value decomposition (SVD) for kernel functions (Mollenhauer et al., 2020) (which can be understood as the kernel version of matrix SVD, where the eigendecomposition is based on Mercer's Theorem (Mercer, 1909)), and that the decomposed spatial (and temporal) kernel functions can be approximated under shared basis functions (cf. Assumption A.2). The resulting approximate finite-rank representation is written as (details are in Appendix A.1)

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

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 kernel parameterization with displacement significantly reduces the rank needed when representing the complicated kernel encountered in practice as shown in Figure 1. (ii) The non-stationarity of original influence of historical events over spatio-temporal space can be conveniently captured by in-homogeneous $\{\psi_l\}_{l=1}^{L}$, $\{u_r\}_{r=1}^{R}$, making the model applicable in modeling general STPPs. (iii) The propagating patterns of influence are characterized by $\{\varphi_l\}_{l=1}^{L}$, $\{v_r\}_{r=1}^{R}$ which go beyond simple parametric forms. In particular, when the events' influence has finite range, *i.e.* there exist τ_{max} and a_{max} such that the influence decays to zero if $|t - t'| > \tau_{\text{max}}$ or $||s - s'|| > a_{\text{max}}$, we can restrict the parameterization of $\{\varphi_l\}_{l=1}^{L}$ and $\{v_r\}_{r=1}^{R}$ on a local domain $[0, \tau_{\text{max}}] \times B(0, a_{\text{max}})$ instead of $[0, T] \times \mathcal{S}$, which further reduce the model complexity. Details of choosing kernel and neural network architectures are described in Appendix C.

Remark 1 (the class of influence kernel expressed). The proposed deep kernel representation covers a large class of non-stationary kernels generally used in STPPs. In particular, the proposed form of the kernel does not need to be positive semi-definite or even symmetric (Reinhart, 2018). The low-rank decomposed formulation equation 4 is of SVD-type (cf. Appendix A.1). While each φ_l (and v_r) can be viewed as stationary (i.e., shift-invariant), the combination with left modes in the summation enables to model spatio-temporal non-stationarity. The technical assumptions A.1 and A.2 do not require more than the existence of a low-rank decomposition motivated by kernel SVD. As long as the $2(R + L)$ many functions ψ_l, φ_l, and u_r, v_r are sufficiently regular, they can be approximated and learned by a neural network. The universal approximation power of neural networks enables our framework to express a broad range of general kernel functions, and the low-rank decomposed form reduces the modeling of a spatio-temporal kernel to finite many functions on time and space domains (the right modes are on truncated domains), respectively.

4 EFFICIENT COMPUTATION OF MODEL

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. The extension of the approach to point process data with marks is given in Appendix B.

4.1 Model optimization with log-barrier

We re-denote $\ell(\mathcal{H})$ in equation 2 by $\ell(\theta)$ in terms of model parameter θ. 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

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

where w is a weight to control the trade-off between log-likelihood and log-barrier. Both w and b can be set accordingly during the learning procedure. Details can be found in Appendix A.2.

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 computation of STPPs 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 $[0, \tau_{\text{max}}]$ 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 $B(0, a_{\max}) \subset \mathbb{R}^{d_S}$ since it decays outside the ball. For each s_i, $\int_{\mathcal{S}} v_r(s - s_i)ds = \int_{B(0,a_{\max}) \cap \{\mathcal{S} - s_i\}} v_r(s)ds$, 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. More details about grid-based computation can be found in Appendix A.3.

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 baseline equipped with a softplus activation function (`NSMPP`) and our model with log-barrier method (`DNSK+Barrier`) on a 1D synthetic data set and a 3D synthetic data set. The quantitative results in Table 1 demonstrates the efficiency improvement of our model by using log-barrier technique. More details about the computation complexity analysis can be found in Appendix A.4.

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 Experiment

We use large-scale synthetic and real data sets to demonstrate the superior performance of our model and present the results in this section. Experimental details and results can be found in Appendix C. Codes will be released upon publication.

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+exp`) 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 others learn the influence kernel in the framework of equation 3. In particular, `NSMPP` also

Table 2: Synthetic data results. Testing log-likelihood per event (on the left side of slash, higher the better) and MRE (on the right side of slash, lower the better) are reported in each entry.

Model	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	$-0.467_{(0.009)}/0.086$	$-2.591_{(0.010)}/0.259$	$-1.353_{(0.002)}/0.212$	$-5.268_{(0.053)}/0.195$	$-2.561_{(0.015)}/0.117$	$-2.289_{(0.002)}/0.316$
NH	$-0.459_{(0.002)}/0.068$	$-2.543_{(0.007)}/0.092$	$-1.315_{(0.032)}/0.204$	$-5.223_{(0.051)}/0.174$	$-2.524_{(0.002)}/0.098$	$-2.291_{(0.003)}/0.319$
THP	$-0.537_{(0.008)}/0.843$	$-2.554_{(0.005)}/0.106$	$-1.319_{(0.003)}/0.115$	$-5.292_{(0.029)}/0.182$	$-2.527_{(0.001)}/0.041$	$-2.497_{(0.018)}/0.350$
PHP+exp	$-0.451_{(0.001)}/0.093$	$-2.725_{(0.002)}/0.181$	$-1.524_{(0.015)}/0.223$	$-2.737_{(0.003)}/0.306$	$-2.683_{(0.002)}/0.291$	$-2.424_{(0.003)}/0.282$
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	$\mathbf{-0.451}_{(0.002)}/\mathbf{0.039}$	$\mathbf{-2.536}_{(0.003)}/\mathbf{0.016}$	$\mathbf{-1.298}_{(0.002)}/\mathbf{0.031}$	$\mathbf{-2.585}_{(0.001)}/\mathbf{0.028}$	$\mathbf{-2.498}_{(0.003)}/\mathbf{0.021}$	$\mathbf{-2.251}_{(0.001)}/\mathbf{0.082}$

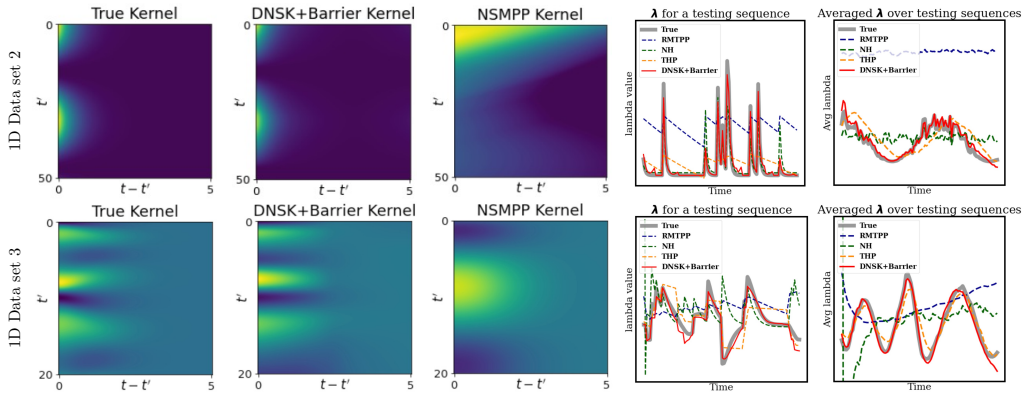


Figure 2: Kernel recovery results of 1D Data set 2 and 3. The first three columns show true kernels, kernels learned by DNSK+Barrier, and kernels learned by NSMPP, respectively. The last two columns show the predicted conditional intensity for a testing sequence and the averaged conditional intensity for all testing sequences in each data set, which can be regarded as the event intensity by taking the expectation of the history.

designs the kernel based on singular value decomposition but parameterizes it without displacement. The model parameters are estimated using the training data via Adam optimization method (Kingma and Ba, 2014). Details of training can be found in Appendix A.2 and C.

5.1 Synthetic Data Experiments

Synthetic data sets. To show the effectiveness of DNSK+Barrier, we conduct all the models on three temporal data sets and three spatio-temporal data sets generated by 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. Data sets are generated using thinning algorithm in Daley and Vere-Jones (2008). Each data set is composed of 2000 sequences. Details of kernel formulas and data generation can be found in Appendix C.

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 $\int_{\mathcal{X}} |\lambda^*(x) - \hat{\lambda}(x)|/\lambda^*(x)dx$ 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 non-stationary kernel recovery for DNSK+Barrier and NSMPP on 1D Data set 2 and 3 (The true kernel used in 1D Data set 3 is the one in Figure 1). DNSK+Barrier recovers the true kernel more accurately than NSMPP, indicating the strong representation power of the low-rank kernel parameterization with displacements. Line charts in Figure 2 present the recovered intensities with the true ones (dark grey curves). It demonstrates that our method can accurately capture the temporal dynamics of events. In particular, the average conditional intensity λ over multiple testing sequences shows the model's ability to recover data non-stationarity over time. While DNSK+Barrier successfully captures the non-stationarity among data, both RMTPP and NH fails to do so by showing a flat curve of the averaged intensity. Note that THP with

Table 3: Results of real data sets with time and categorical marks. We compare the testing log-likelihood (Testing ℓ, higher the better), event time prediction error (lower the better), and event type prediction accuracy (higher the better) of DNSK+Barrier and other baselines on each data set.

	Financial			**StackOverflow**		
Model	Testing $-\ell$	Time RMSE	Type Accuracy	Testing $-\ell$	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	-0.938	1.019	0.596	**–1.231**	11.804	0.436
NSMPP	-3.058	1.276	0.608	-3.182	8.735	0.447
DNSK+Softplus	-0.889	0.327	0.621	-2.173	6.416	0.497
DNSK+Barrier	**–0.709**	**0.153**	**0.630**	-2.089	**4.833**	**0.508**

positional encoding recovers the data non-stationarity (as shown in two figures in the last column). However, our method still outperforms THP which suffers from limited model expressiveness when complicated propagation of event influence is involved (see two figures in the penultimate column).

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 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. THP outperforms RMTPP in non-stationary cases but is still limited due to its pre-assumed parametric form of influence propagation. More results about kernel and intensity recovery can be found in Appendix C.

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 STPPs (refer to Appendix B for detailed definition and kernel modeling): (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.

Next, 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-world 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

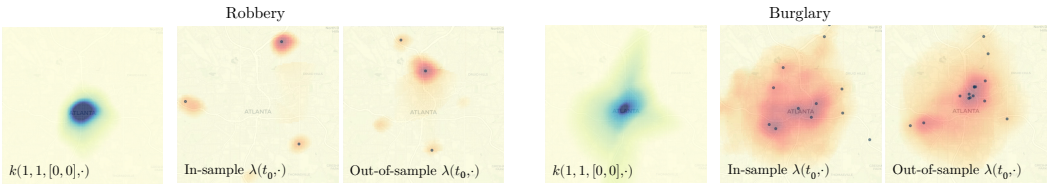


Figure 3: Kernel fitting and intensity prediction for ATL robbery & burglary data. First panel in each category shows the learned influence kernel of each crime at fixed geolocation in downtown ATL. Other panels show the predicted conditional intensity for two crimes over space at an in-sample and out-of-sample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

Table 4: Real data results with spatial and mark information. Testing log-likelihood (higher the better) and prediction mean absolute error of event time and location (lower the better) are reported.

Model	South California Earthquake			Atlanta Robbery	Atlanta Burglary	Atlanta Textual Crime
	Testing ℓ	Time MAE	Location MAE	Testing ℓ	Testing ℓ	Testing ℓ
RMTPP	$-1.825_{(0.053)}$	6.963	0.602	$-2.349_{(0.018)}$	$-0.246_{(0.008)}$	/
NH	$-1.818_{(0.037)}$	6.880	0.458	$-2.445_{(0.016)}$	$-0.308_{(0.012)}$	/
THP	$-1.784_{(0.007)}$	6.113	0.633	$\mathbf{-2.221}_{(0.005)}$	$-0.251_{(0.001)}$	/
PHP+exp	$-2.048_{(0.093)}$	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+Softplus	$-1.807_{(0.096)}$	1.685	0.492	$-2.435_{(0.009)}$	$0.493_{(0.008)}$	$-5.876_{(0.057)}$
DNSK+Barrier	$\mathbf{-1.751}_{(0.080)}$	**1.474**	**0.431**	$-2.255_{(0.008)}$	$\mathbf{0.519}_{(0.012)}$	$\mathbf{-5.279}_{(0.044)}$

[1] RMTPP, NH, and THP are not applicable when dealing with high-dimensional data.

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 $f(t,s) = \lambda(t,s)\exp\left(-\int_{\mathcal{S}}\int_{t_n}^{t}\lambda(\tau,\nu)d\tau d\nu\right)$. The predicted time and location of the next event can be computed as $\mathbb{E}\left[t_{n+1}|\mathcal{H}_t\right] = \int_{t_n}^{\infty} t\int_{\mathcal{S}} f(t,s)dsdt, \quad \mathbb{E}\left[s_{n+1}|\mathcal{H}_t\right] = \int_{\mathcal{S}} s\int_{t_n}^{\infty} f(t,s)dtds$. We predict the the time and location of the last event in each sequence. The mean absolute error (MAE) of the predictions is computed. The 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 ability of the proposed method to recover data non-stationarity with different sequence lengths, and improve the limited model interpretability of other neural network-based methods (RMTPP, NH, and THP) 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 dimension $d = 50$ using Restricted Boltzmann Machine (RBM) (Fischer and Igel, 2012). The average testing log-likelihoods per event for each model are reported in Table 4. The results show that DNSK+Barrier outperforms PHP+exp 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 A.4 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

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

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.

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.

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.

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.

Michael Graham, Jarno Kiviaho, and Jussi Nikkinen. Short-term and long-term dependencies of the s&p 500 index and commodity prices. *Quantitative Finance*, 13(4):583–592, 2013.

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.

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.

Mattes Mollenhauer, Ingmar Schuster, Stefan Klus, and Christof Schütte. Singular value decomposition of operators on reproducing kernel hilbert spaces. In *Advances in Dynamics, Optimization and Computation: A volume dedicated to Michael Dellnitz on the occasion of his 60th birthday*, pages 109–131. Springer, 2020.

J Mercer. Functions ofpositive and negativetypeand theircommection with the theory ofintegral equations. *Philos. Trinsdictions Rogyal Soc*, 209:4–415, 1909.

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.