Spatio-temporal Event Prediction via Deep Point Processes

Maya Okawa

Dissertation submitted to the Graduate School of Informatics
Kyoto University
Doctoral dissertation

Department of Intelligence Science and Technology
Graduate School of Informatics
Kyoto University
Supervisor: Hisashi Kashima

February 7, 2022

Submitted to the degree of Doctor of Informatics

Department of Intelligence Science and Technology
Graduate School of Informatics
Kyoto University

Abstract

With the recent developments in mobile and sensor technologies, a large amount of event data on social and natural phenomena are being continuously generated from a plurality of sources such as social media, medical records, and business transactions. The growing volumes of event data open up new opportunities for researchers and practitioners to better understand the underlying phenomena and mitigate social problems. Especially, predicting spatio-temporal events is a key component of applications in many fields including transportation, public safety, and health care. Point processes provide a principled framework for modeling event data and have found many applications in a diverse range of fields. However, these models are limited since the event occurrence is governed by various complex contextual factors; and thus it dynamically changes over time. Such factors can be either observable or unobservable. It remains unexplored how these factors cause the emergence of spatio-temporal dynamics in event data.

This thesis focuses on establishing point process models for modeling and predicting spatio-temporal event data. In this thesis, we introduce two approaches to take into account the influence of either observable or unobservable factors, by integrating deep neural networks and the point process frameworks. First, we explore how to fully exploit information on contextual factors to achieve an accurate prediction for spatio-temporal events. In Chapter 2, we present a Poisson process model combined with a convolutional neural network (CNN) that effectively utilizes rich contextual information. In Chapter 3, we extend this approach to triggering processes and develop a Hawkes process model that learns the time-decaying influence from the past events and the contribution of contextual observable factors. Next, we describe how to estimate the underlying effect of unobservable factors from event data. In Chapter 4, we propose a Hawkes process model that infers the influence of unobservable contextual factors via a neural network, where the impact of unobservable contextual factors are modeled by latent variables. In each chapter, we carry out extensive experiments on real-world datasets from several representative applications, including transportation, public safety, crime, public health, social media, and natural disasters, and

demonstrate the proposed models' capabilities for event prediction.

Acknowledgements

First, I would like to express my sincere gratitude to my advisor Dr. Hisashi Kashima, Professor of Graduate School of Informatics, Kyoto University. I would like to thank you for giving me the opportunity to pursue this degree in his lab and for supporting my research.

I would also like to thank Professors Akihiro Yamamoto and Masatoshi Yoshikawa, who have served on my thesis committee and have provided helpful comments and invaluable advice.

I have been privileged to have a marvelous group of coauthors and collaborators. I appreciate each of them: Tomoharu Iwata, Hiroyuki Toda, Hideaki Kim, Yusuke Tanaka, Takeshi Kurashima, Aki Hayashi, Hiroshi Sawada, and Naonori Ueda. I would especially like to thank Dr. Hiroyuki Toda for his unselfish guidance and support in developing my research career. I also owe a special thank you to Dr. Tomoharu Iwata for active discussions and collaboration, and also for being a great role model as a researcher.

I wish to extend my gratitude to Kyoto University and my employer for providing the appropriate research environment and resources to do this study. My special thanks also go to my friends and colleagues at the NTT Laboratories. My research achievements would not have been possible without their constant encouragement and inspiration over the many years.

I have been blessed with financial support during my studies. I gratefully acknowledge the Business Communication co., Ltd. for granting me the Ph.D. Research Scholarship.

Last but most importantly, I want to express my deep appreciation to my family. More than anyone else, I am grateful to my parents for their unfailing encouragement and loving support. I am also thankful to my grandparents, my sister, my brother, my brother-in-law and my niece for their great sense of humor, wisdom, and friendship.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Applications

In today's digital world, a large amount of data on spatio-temporal (or temporal) events have become available. For example, traffic sensors generate a variety of spatio-temporal event data in urban areas on human mobility and traffic events. Social media produces vast quantities of event data on social behaviors generated by the users. We refer to such data as spatio-temporal (temporal) event data. The growing volumes of event data open up new opportunities for expanding our knowledge about social and natural phenomena.

The main interest of our research lies in modeling temporal and spatio-temporal event data and investigating spatial and temporal evolutions of phenomena to predict future events. Modeling event data and predicting future events is crucial for many practical applications across domains such as transportation, public safety, health care, and environmental management. This thesis analyzes temporal and spatio-temporal event data from the following domains.

- *Transportation* – With advanced sensing techniques, it is possible to collect trip data of many kinds of transport vehicles including taxis, sharing bikes, and buses. Such trip data records the trip starting and ending geo-location, along with time. The dataset could enable us to better understand traffic phenomena, and help mitigate traffic problems. For example, if taxi dispatch service operators can estimate with high accuracy the future taxi pick-up times and locations, they can allocate taxis to the right places and the right times in advance.

- *Public safety* – Several data projects produce a vast amount of data on political conflict and violence, compiled from press reports, social media, and government archives. Such data contains the dates, locations, and types of all reported political

violence and protest events across the countries, including protests, conflicts, fights, and mass violence. Providing predictions and early warnings of violence benefits conflict management sectors in allocating resources to prevent violence.

- *Crime* – Modern police organizations collect and store detailed reports on crime incidents. The police's criminal reports typically involve the date and time of occurrence, crime types, geographic location. Understanding patterns in criminal activity and predicting crime hot spots enable law enforcement agencies to effectively allocate officers to prevent or respond to incidents.

- *Public health* – Disease surveillance systems collect and monitor data on disease outbreaks to trigger appropriate public health interventions. This dataset codes the locations, dates, and times of disease outbreaks. Policymakers would be able to design prompt interventions to curb the spread of disease given a better understanding of the mechanisms behind the transmission and more reliable predictions.

- *Social media* – The widespread adoption of communication tools, such as social media platforms, has generated a wealth of data on human activities such as information sharing in social networks. These activities are often represented as a sequence of timestamped events. For instance, we have the records of information spreading traces with the timestamps of spreading behaviors (e.g., creating and sharing content). Predicting cascade dynamics may help optimize business performance (e.g. designing social marketing campaigns).

- *Natural disasters* – Annually, hundred earthquakes are obtained from various seismic stations all over the world. Earthquake data provide important information on the time, magnitude, and epicenter location of an earthquake. Accurate and timely prediction of an earthquake and early warning can save lives and prevent damages caused by earthquakes.

Many social and natural phenomena exhibit self-exciting or triggering properties, where the previous events trigger future events. For instance, infectious diseases like COVID-19 are transmitted from one county to another, leading to a worldwide pandemic [91]. In the context of social media, ones' social behaviors (e.g., posting and spreading information) are likely to trigger other users' responses. These responses further trigger more responses, resulting in an information cascade. The occurrence of an earthquake increases the likelihood of a second earthquake nearby in space and time.

Point processes offer a powerful mathematical tool for modeling event data in continuous time and/or space. A spatio-temporal point process is a random process whose

realization consists of a list of discrete events in continuous time and space. The dynamics of the point process are determined by the so-called "intensity" function that describes the rate of events occurring at any location and at any time. The two most representative models of point processes are Hawkes processes and Inhomogeneous Poisson processes. Hawkes processes [31] explicitly model the influence of the past events and capture triggering patterns between events (i.e., diffusion processes). Hawkes processes have been proven effective for modeling diffusion processes, including earthquakes and aftershocks [59], near-repeat patterns of crimes [59], financial transactions [23, 5, 32], online purchases [108, 22, 18, 102], and information cascades [119, 81, 26]. Inhomogeneous Poisson processes provide a flexible way to learn the temporal and/or spatial variation in the event occurrences. This class of point processes does not directly model the triggering patterns; but it can learn the local spatio-temporal interactions between events empirically. Inhomogeneous Poisson processes have been successfully applied to a wide spectrum of events such as financial events [45], wildfires [85] and infrastructure failures [24].

However, predicting spatio-temporal events is still challenging, because event occurrence is determined by various contextual factors, which can change over time. Contextual factors can be classified into observable and unobservable ones.

- *Observable contextual factors.* Observable contextual factors are defined as a set of observable features. Such features might include transportation networks [28], land use [8], and social and traffic information [121, 100, 12] as well as time of day and weather [**?**, 12]. Most of them take the form of unstructured representations. For example, information about geographical characteristics can be obtained from map images. Traffic and social event information can be expressed in the form of natural language expressions. This demands a framework that can capture the complex dynamics of event occurrence given the rich contextual features present.

- *Unobservable contextual factors.* Some of the relevant contextual factors are unobservable or inaccessible. These factors implicitly affect temporal and/or spatial dynamics in the occurrence of events. For example, information diffusion heavily depends on ongoing peoples' interests; nevertheless, the time-evolution of peoples' interest in a given topic is generally unknown and not directly observable. This demands a framework that models the dynamics occurring due to underlying hidden context.

In this thesis, we develop two approaches for modeling temporal and spatio-temporal event data to capture the temporal and/or spatial variation of event occurrence that is governed by contextual factors. First, we present two methods for predicting spatio-temporal

events with the use of rich contextual information based on two popular point processes: Hawkes process (chapter 2) and inhomogeneous Poisson processes (chapter 3). Second, we propose a novel Hawkes process method that can capture the impact of underlying and unobservable factors behind the diffusion processes (chapter 4).

1.2 Overview and Summary of Contributions

The key contributions of this thesis are three new methods for modeling temporal and spatio-temporal event data. This thesis consists of three main parts corresponding to the three methods as follows. The first two parts propose novel point process models integrating rich observable features and learning their complex effects on the event occurrence. In chapter 2, we develop an inhomogeneous Poisson process model that can leverage the contextual features, such as map images and social/traffic event descriptions, that impact event occurrence. In chapter 3, we propose a novel Hawkes process model for modeling diffusion processes and predicting spatio-temporal events, which leverages the external features contained in georeferenced images (e.g., satellite images and map images), that impact triggering processes. The last part aims to learn the effects of underlying and unobservable factors on the event occurrence. In chapter 4, we present a novel Hawkes process model for modeling diffusion processes and predicting future events, which estimates latent features that influence the time-evolving dynamics behind the diffusion processes.

Table 1.1 summarizes the core contributions of this thesis and their organization with references to the chapters. The next few subsections provide a chapter by chapter outline of the thesis.

Table 1.1: Contributions and organization of the thesis.

	Contextual factors	
Conventional tools	Observable	Unobservable
Inhomogeneous Poisson process	Inhomogeneous Poisson process with use of rich contextual features. **Chapter 2**; [70]	Intensity-free approach for point process modeling. [106, 13]
Hawkes process	Hawkes process with use of rich contextual features. **Chapter 3**; [73]	Hawkes process for learning impact of underlying factors. **Chapter 4**; [72]

1.2.1 Spatio-temporal Event Prediction with Rich Contextual Information (Chapter 2)

Originally published at the 25th International Conference on Knowledge Discovery & Data Mining (KDD 2019) [70] and *Transactions of the Japanese Society for Artificial Intelligence* [71].

This chapter aims to predict when and where events will occur in cities, like taxi pick-ups, crimes, and vehicle collisions. Though many point processes have been proposed to model events in a continuous spatio-temporal space, none of them allow for the consideration of the rich contextual factors that affect event occurrences, such as weather, social activities, geographical characteristics, and traffic.

In this chapter, we propose DMPP (Deep Mixture Point Processes), a point process model for predicting spatio-temporal events with the use of rich contextual information; a key advance is its incorporation of the heterogeneous and high-dimensional context available in image and text data. Specifically, we design the intensity of our point process model as a mixture of kernels, where the mixture weights are modeled by a deep neural network. This formulation allows us to automatically learn the complex nonlinear effects of the contextual factors on event occurrence. At the same time, this formulation makes analytical integration over the intensity, which is required for point process estimation, tractable.

We conduct extensive experiments on real-world data sets from three urban domains. Concerning event occurrence, the proposed method achieves better predictive performance than all existing methods on all data sets.

1.2.2 Context-aware Spatio-temporal Event Prediction via Convolutional Hawkes Processes (Chapter 3)

Originally published at Machine Learning Journal (ECML-PKDD Journal Track) [73]

This chapter tackles the problem of predicting spatio-temporal events like disease outbreaks, armed conflicts, and crimes and revealing the underlying triggering patterns is a crucial task for many applications, ranging from disease control to global politics.

Traditional event prediction models based on Hawkes processes capture the spatio-temporal relationships between events, but cannot incorporate complex and heterogeneous external features, including population distribution, weather, and terrain. In this chapter, we propose an event prediction method that effectively utilizes the rich external informa-

tion present in sets of unstructured data (e.g., map images, satellite images, and weather maps). Specifically, we extend a convolutional neural network (CNN) by combining it with continuous kernel convolution; and design the conditional intensity of Hawkes process based on the extended neural network model that accepts images as its input. Our approach of using the continuous convolution kernel provides a flexible way to discover the complex effect of external factors on the triggering process, as well as yielding tractable optimization algorithms.

We use real-world event data from different domains (i.e., disease outbreaks, armed conflicts, and protests) to demonstrate that the proposed method has better prediction performance than existing methods.

1.2.3 Dynamic Hawkes Processes for Discovering Time-evolving Communities' States behind Diffusion Processes (Chapter 4)

Originally published at the 27th International Conference on Knowledge Discovery & Data Mining (KDD 2021) [72]

Sequences of events including infectious disease outbreaks, social network activities, and crimes are ubiquitous and the data on such events carry essential information about the underlying diffusion processes between communities (e.g., regions, online user groups). Modeling diffusion processes and predicting future events is crucial in many applications including epidemic control, viral marketing, and predictive policing.

Hawkes processes offer a central tool for modeling the diffusion processes, in which the influence from the past events is described by the triggering kernel. However, the triggering kernel parameters, which govern how each community is influenced by past events, are assumed to be static over time. In the real world, the diffusion processes depend not only on the influences from the past but also the current (time-evolving) states of the communities, e.g., people's awareness of the disease and people's current interests.

In this chapter, we propose a novel Hawkes process model that can capture the underlying dynamics of community states behind the diffusion processes and predict the occurrences of events based on the dynamics. Specifically, we model the latent dynamic function that encodes these hidden dynamics by a mixture of neural networks. Then we design the triggering kernel using the latent dynamic function and its integral. The proposed method, termed DHP (Dynamic Hawkes Processes), offers a flexible way to learn complex representations of the time-evolving communities' states, while at the same time it allows computing the exact likelihood, which makes parameter learning tractable.

We carry out extensive experiments using four real-world event datasets: Reddit, News, Protest, and Crime. The results show that DHP outperforms the existing works. Case studies demonstrate that DHP uncovers the hidden state dynamics of communities that underlie the diffusion processes by the latent dynamic function.

Chapter 2

Spatio-temporal Event Prediction with Rich Contextual Information

2.1 Introduction

With the fast development of the internet of things (IoT) technology, large volumes of event data are being generated from various sources including surveillance systems and sensors. Such event data includes information about time and geolocation, indicating where and when each event occurred. For instance, taxi pick-up records are represented as a list of events consisting of the pick-up locations and the departure times. Crimes are recorded together with the time and location at which the crime took place. Predicting events is a key component of applications in many fields such as urban planning, transportation optimization, and location-based marketing. If taxi dispatch service operators can estimate with high accuracy the future taxi pick-up times and locations, they can allocate taxis to the right places and the right times in advance. Criminal incident prediction will help law enforcement agencies to implement effective police activities that can suppress criminality.

Predicting spatio-temporal events, however, is extremely challenging, because event occurrence is determined by various contextual factors. Such contextual features also include geographical characteristics, e.g., transportation networks [28] and land use [8]; temporal attributes, e.g., day of week and weather conditions [114, 12]; and other features, e.g., social and traffic information [121, 100, 12]. Data on these contextual features can either be observable or unobservable. In this chapter, we explore how to integrate observable contextual features affecting the occurrence of spatio-temporal events.

In this chapter, we aim to develop a framework that can capture the complex dynamics of event occurrence given the contextual features present. The conventional approach to

this problem is based on regression models [121, 88, 35]. They are intended to model the aggregated number of events within a predefined spatial region and time interval, which is fundamentally different from our task. We focus more on the point process approach to model a sequence of events in continuous time and space, without aggregation, by using explicit information about location and/or time; and predicting the precise time and location at which each event will occur.

Point process is a sophisticated framework for modeling a sequence of events in continuous time and space; it directly estimates an *intensity* function that describes the rate of events occurring at any location and any time. The influence of the contextual features can be modeled by special point process models [17, 33, 29], where the intensity function is described as a function of covariates, i.e., the contextual features. However, this approach has a fundamental limitation. In many practical cases, their assumptions on the functional form of covariates may be too restrictive to capture complex and intricate effects of contextual features; they do not accommodate unstructured data such as images and texts. Most contextual features take the form of unstructured representations. For example, information about geographical characteristics can be obtained from map images. Traffic and social event information can be expressed in the form of natural language expressions.

In this chapter, we propose an event prediction method that effectively incorporates such unstructured data into the point process model. Motivated by the recent success of the deep learning approach, we use it to enhance the point process model. The naive approach is to directly model the intensity by a deep neural network. Unfortunately, this approach triggers the intractable optimization problem as integral computations are required to determine the likelihood needed for estimation.

We address this through a novel formulation of spatio-temporal point processes. Specifically, we design the intensity as a deep mixture of experts, whose mixture weights are modeled by a deep neural network. This method called DMPP (Deep Mixture Point Processes), enables us to incorporate unstructured contextual features (e.g., road networks and social/traffic event descriptions) into the predictive model, and to automatically learn their complex effects on event occurrence. Moreover, this formulation yields a tractable optimization problem. Our mixture model-based approach permits the likelihood to be determined from tractable integration. Learning can be done with simple back-propagation.

We conduct experiments on three real-world data sets from multiple urban domains and show that our DMPP consistently outperforms existing methods in event prediction tasks. The experiments also demonstrate that DMPP provides useful insights about why and under which circumstances events occur. By utilizing a recently developed self-attention

mechanism [54, 52], DMPP helps us better understand how the contextual features influence event occurrence. Such insights could further aid policy makers in creating more effective strategies.

The main contributions of this chapter can be summarized as follows:

- We propose DMPP, a novel method for spatio-temporal event prediction. It accurately and effectively predicts spatio-temporal events by leveraging the contextual features, such as map images and social/traffic event descriptions, that impact event occurrence.
- We integrate the deep learning approach into the point process framework. Specifically, we extract the intensity by using a deep mixture of experts, whose mixture weights are modeled by a deep neural network. This formulation allows us to utilize the information present in unstructured contextual features, and to automatically discover their complex effects on event occurrence, while at the same time yielding tractable optimization.
- We develop an efficient estimation procedure for training and evaluating DMPP.
- We conduct extensive experiments on real-world data sets from three urban domains. With regard to event occurrence, the proposed method achieves better predictive performance than all existing methods on all data sets.

2.2 Related Work

Point process is a general mathematical framework for modeling a sequence of events; it directly estimates the rate of event occurrence, by using explicit information about location and/or time. Early work mainly focused on the temporal aspect of events. The temporal Hawkes processes [31] are a class of temporal point process models that can capture burst phenomena; in these models, the probability of future events is assumed to be strengthened by past events, with the influence decaying exponentially over time. They have been used for analysing disease transmissions [15], financial transactions [5, 2], terrorist attacks [78], social activities [37, 25], search behaviors [51], and so on. Recent studies have expanded its application to human mobility modeling. Wang *et al.* [99] proposed Hawkes process variant to identify trip purpose. Du *et al.* [21] presented a recurrent marked temporal point process (RMTPP) and demonstrated its effectiveness in predicting the timing of taxi pick-ups. Log Gaussian Cox process (LGCP) has been used to effectively model temporal events, such as wildfires [85] and infrastructure failures [24], in which the logarithm of the intensity is

assumed to be drawn from a Gaussian process. The spatio-temporal point process is a more general framework, and considers both spatial and temporal domains. The spatio-temporal self-exciting point processes, an extension of temporal Hawkes processes, have been used for modeling seismicity [66], contagious diseases [83], and crime incidents [59], among other applications The spatio-temporal LGCP has been applied to model wildfires [85] and infrastructure failures [24].

All these methods, however, have one fundamental limitation: they ignore contextual features even though they are known to influence event occurrence. Human activities are largely influenced by environmental features, i.e., weather, geographical characteristics and traffic conditions. These features must be considered to accurately predict future events. Their influence has been modeled by a special point process model, called the proportional hazards model [17]; it treats the intensity rate as a function of covariates. One major limitation of this model is that it assumes that the contextual features create only linear effects. Most features have highly non-linear effects on real world event occurrence. The simplest solution is to fit non-linear functions, such as polynomials [33] and splines [29], to covariates. Unfortunately, their assumptions may be too restrictive to capture complex and intricate effects of contextual features. Also, this approach forces us to carefully choose or design the functional form of the covariates so that they accurately capture reality. However, in practice, how the contextual features influence event occurrence is largely unknown.

This paper constructs a novel point process method called DMPP; it extends the spatio-temporal point process with a deep learning model. The pioneering work by [21, 106, 107] is most related to our approach. However, they focus only on the temporal dynamics of event occurrence, and so ignore spatial dynamics. Also, those methods are optimized to predict the timing of the next event. Instead, we are interested in predicting longer event sequences. Moreover, none of these methods accept contextual features.

2.3 Preliminaries

In this section, before introducing our method, we first provide the necessary theoretical background to the point process.

Point process is a random sequence of event occurrences over a domain. We assume here a sequence of events with known times and locations. Let $\mathbf{x} = (t, s)$ be the event written as the pair of time $t \in \mathbb{T}$ and location $s \in \mathbb{S}$, where $\mathbb{T} \times \mathbb{S}$ is a subset of $\mathbb{R} \times \mathbb{R}^2$. In the following, we denote the number of events falling in subset A of $\mathbb{T} \times \mathbb{S}$ as $N(A)$. The general approach to identifying a point process is to estimate the "intensity" $\lambda(\mathbf{x})$. The

intensity $\lambda(\mathbf{x})$ represents the rate of event occurrence in a small region, and is defined as

$$\lambda(\mathbf{x}) = \lambda(t, s) \equiv \lim_{|dt|\to 0, |ds|\to 0} \frac{\mathbb{E}[N(dt \times ds)]}{|dt||ds|}, \tag{2.1}$$

where dt is a small interval around time t, $|dt|$ is its duration, ds is a small region containing location s, and $|ds|$ is its area. $\mathbb{E}$ indicates an expectation measure. The functional form of intensity is designed to appropriately capture the underlying dynamics of event occurrence.

Given a sequence of events $\mathcal{X} = \{\mathbf{x}_i = (t_i, \mathbf{s}_i)\}_{i=1}^N$, $t_i \in \mathbb{T}$ and $\mathbf{s}_i \in \mathbb{S}$, the likelihood is given by

$$p(\mathcal{X}|\lambda(\mathbf{x})) = \prod_{i=1}^{N} \lambda(\mathbf{x}_i) \cdot \exp\left(- \int_{\mathbb{T}\times\mathbb{S}} \lambda(\mathbf{x}) d\mathbf{x} \right). \tag{2.2}$$

2.4 Deep Mixture Point Processes

This section presents the proposed method referred to as DMPP (Deep Mixture Point Processes). We first introduce the notations and definitions used in this paper. We then provide the model formulation of DMPP followed by parameter learning and prediction. The neural network architecture used in DMPP is detailed in Section 2.4.2.

2.4.1 Problem Definition

We introduce the notations used in this paper and formally define the problem of event prediction.

Let $\mathcal{X} = \{\mathbf{x}_i = (t_i, \mathbf{s}_i)\}_{i=1}^N$ denote a sequence of events over space and time, where $(t_i,\mathbf{s}_i) \in \mathbb{T} \times \mathbb{S} \in \mathbb{R} \times \mathbb{R}^2$ and N is the total number of events known.

Further, we are also given contextual information associated with the spatio-temporal region $\mathbb{T} \times \mathbb{S}$. Let $\mathcal{D} = A_1, A_2, ..., A_K$ be a set of contextual features, where A_k is the k-th feature, and K is the number of contextual features. Examples of the contextual features include weather, social/traffic event information and geographical characteristics. The social/traffic event information may be a collection of social/traffic event descriptions that include locations and times. In this case, $A_\bullet$ is represented by a set of four-element tuples, each of which has the following format: `<time, latitude, longitude, description>`. Information about the geographical characteristics can be obtained from map images.

Given the contextual features $\mathcal{D}$ up to time $T + \Delta T$, and the event sequence $\mathcal{X}$ up to time T, we aim to learn a predictor that:

- predicts times and locations of events in the future time window $[T, T + \Delta T]$;

- predicts the number of events within any given spatial region and the time period in $[T, T + \Delta T]$,

by leveraging $\mathcal{D}$ and $\mathcal{X}$.

2.4.2 Model Formulation

In this work, we construct a novel point process method for spatio-temporal event prediction that can incorporate unstructured contextual features such as map images and social/traffic event descriptions. Our point process intensity must be designed so that it is flexible enough to capture the highly complex effects of contextual features, while at the same time being tractable. Deep learning models have proven to be an extremely useful, especially in automatically extracting the meaningful information contained in the unstructured data including images and text descriptions. Inspired by this, we propose a novel formulation of point process model by integrating it with deep learning approach. The proposed method is referred to as DMPP (Deep Mixture Point Processes). In particular, we model the intensity by a neural network function that accepts contextual features as its input.

Intensity function.

We develop a flexible and computationally effective way of using kernel convolution to specify the intensity function. Formally, we design the intensity as a function of contextual features:

$$\lambda(\mathbf{x}|\mathcal{D}) = \int f\big(\mathbf{u}, \mathbf{Z}(\mathbf{u}; \mathcal{D}); \theta\big) k(\mathbf{x}, \mathbf{u}) d\mathbf{u}, \tag{2.3}$$

where $\mathbf{u} = (\tau, \mathbf{r})$ for $\tau \in \mathbb{T}$ and $\mathbf{r} \in \mathbb{S}$, $k(\cdot, \mathbf{u})$ is a kernel function centered at $\mathbf{u}$. $f(\cdot)$ is any deep learning model that returns a nonnegative scalar, and θ denotes a set of the parameters of the deep neural network. $\mathbf{Z}(\mathbf{u}, \mathcal{D}) = \{Z_1(\mathbf{u}; A_1), ..., Z_K(\mathbf{u}; A_K)\}$ is a set of the feature values at the spatio-temporal point $\mathbf{u}$, where Z_k is defined as the operator to extract values of k-th feature at $\mathbf{u}$. As one example, social/traffic event descriptions can be represented by tuples of time, location and event descriptions. In this case, operator Z outputs a list of social/traffic event descriptions scheduled within $[\tau - \Delta\tau, \tau + \Delta\tau]$ and located within a predefined distance, $||\mathbf{r} - \mathbf{r}'|| < \Delta\mathbf{r}$, given $\mathbf{u} = (\tau, \mathbf{r})$. As another example, given a map image representing geographical characteristics, Z returns the feature vectors (e.g., RGB values) of the map image around $\mathbf{r}$. The formulation of (2.7) is built upon a process convolution approach [34, 49, 47]; but we extend it so that the point process intensity

accepts unstructured contextual features, by integrating it with a deep neural network. This extension enables us to integrate unstructured contextual data, and automatically learn their complex effects on event occurrence. Although being flexible and expressive, this intensity is intractable as it involves the integral of the neural network function $f(\cdot)$. Thus, by introducing J representative points $\mathcal{U} = \{\mathbf{u}_j\}_{j=1}^{J}$ in the spatio-temporal region, we obtain a discrete approximation to (2.7):

$$\lambda(\mathbf{x}|\mathcal{D}) = \sum_{j=1}^{J} f\big(\mathbf{u}_j, \mathbf{z}_j; \theta\big) k(\mathbf{x}, \mathbf{u}_j), \tag{2.4}$$

where each point $\mathbf{u}_j = (\tau_j, \mathbf{r}_j)$ consists of its time $\tau_j \in \mathbb{T}$ and location $\mathbf{r}_j \in \mathbb{S}$. Here we define $\mathbf{Z}(\mathbf{u}_j; \mathcal{D})$ as $\mathbf{z}_j$, which represents the contextual feature vector associated with the j-th point $\mathbf{u}_j$. Consequently, the intensity is described as a mixture of kernel experts, in which mixture weights are modeled by a deep neural network whose inputs are contextual features. The resulting model yields the automatic learning of their influences as well as making the learning problem tractable (discussed in Section 4.3).

Configuration of representative points.

The set of representative points is structured as follows. We first introduce M discrete points placed uniformly along time axis within $[0, T + \Delta T]$ to define time points $\mathcal{T}$: $0 = \tau'_1 < ... < \tau'_M = T + \Delta T$. Similarly, we set L discrete points within the spatial region to define space points $\mathcal{S}$: $\mathbf{r}'_1, ..., \mathbf{r}'_L$, where $\mathbf{r}'_l \in \mathbb{S}$. The set of representative points is defined by the Cartesian product of the space and time points:

$$\mathcal{U} = \{(\tau, \mathbf{r}) \mid \tau \in \mathcal{T} \wedge \mathbf{r} \in \mathcal{S}\}. \tag{2.5}$$

Therefore $J = ML$. There are some options in locating the representative points, either fixing them or optimizing them in terms of spatial coordinates. In this paper, we choose the former and fix them on a regular grid, as simplifies the computation. Note that the number of representative points, J, determines the trade-off between approximation accuracy and computation complexity. Larger J improves approximation, while reducing computational cost. A sensitivity analysis of the impact of J is given in the experimental section.

Kernel function.

We can make various assumptions as to the kernel function $k(\mathbf{x}, \mathbf{u}_j)$. For example, we can use a Gaussian kernel:

$$k(\mathbf{x}, \mathbf{u}_j) = \exp\big(-(\mathbf{x} - \mathbf{u}_j)^\top \Sigma^{-1} (\mathbf{x} - \mathbf{u}_j)\big), \tag{2.6}$$

where Σ is a 3×3 covariance matrix (bandwidth) of the kernel. Other kernel functions, such as Matern, sigmoid, periodic (trigonometric), and compactly supported kernels [104] are viable alternatives. In the experiment, we explored three kinds of kernel functions: uniform, Gaussian, and compactly supported Gaussian. We define each kernel below.

Uniform kernel.

$$k(\mathbf{x}, \mathbf{u}_j) = \mathbb{1}(||\mathbf{x} - \mathbf{u}_j|| < w),$$

where $\mathbb{1}(\cdot)$ is an indicator function.

Gaussian kernel.

$$k(\mathbf{x}, \mathbf{u}_j) = \exp\big(-(\mathbf{x} - \mathbf{u}_j)^\top \Sigma^{-1} (\mathbf{x} - \mathbf{u}_j)\big),$$

where Σ is a 3×3 covariance matrix.

Compactly supported Gaussian kernel.

$$k(\mathbf{x}, \mathbf{u}_j) = \exp\big(-(\mathbf{x} - \mathbf{u}_j)^\top \Sigma^{-1} (\mathbf{x} - \mathbf{u}_j)\big) \cdot \mathbb{1}(||\mathbf{x} - \mathbf{u}_j|| < w),$$

where Σ is a 3×3 covariance matrix, $\mathbb{1}(\cdot)$ is an indicator function, and w is a positive parameter that thresholds the kernels, $||\mathbf{x} - \mathbf{u}_j|| \geq w$, to zeros. This means that $k(\mathbf{x}, \mathbf{u}_j)$ will be zero when $\mathbf{x}$ and $\mathbf{u}_j$ are far enough away. The use of the compactly supported kernel allows for an effective learning algorithm, especially for large data size N and for large numbers of representative points, J. The objective (3.13) involves kernel evaluations for all pairs of $\mathbf{x}_i$ and $\mathbf{u}_j$, resulting in an $N \times J$ kernel matrix K with elements $K_{ij} = k(\mathbf{x}_i, \mathbf{u}_j)$. The back-propagation is carried out by taking the derivation of K, which requires $\mathcal{O}(NJ)$ operations at each iteration ($\mathcal{O}(|\mathcal{I}|J)$ for the mini-batch optimization). The computation burden can be impractically heavy when the data size N (the mini-batch size $|\mathcal{I}|$) or the number of representative points J is large. The use of the compactly supported kernel allows us to scale up the learning algorithm. Such a kernel ensures that the kernel element K_{ij} is zero whenever the distance between $\mathbf{x}_i$ and $\mathbf{u}_j$ is above a certain threshold. This leads to a sparse structure in the kernel matrix, K, thus allowing for fast sparse matrix

computations.

Neural network model.

The neural network model $f(\cdot)$ can be designed to suit the input data. We consider the general case wherein image features (e.g., map images) and text features (e.g., social/traffic event descriptions) are available. We propose an attention network architecture that fully exploits visual and textual information. The proposed architecture consists of three components: *an image attention network* to extract image features, *a text attention network* to encode text features, and *a multimodal fusion module*. We construct *the image attention network* by combining a CNN with the spatial attention model proposed by [54]. We design *the text attention network* on a CNN designed for sentences [39] and an attention mechanism [52]. The structured texts are split into words and then processed by the text attention network. Lastly, the extracted features from images and texts are fused into a single representation via *the multimodal fusion module*, and input to the intensity function.

In the following, we detail each component of the neural network. This section details the architecture of the neural network used in our experiment, see Figure 2.1. Our neural network consists of three components: (i) *the image attention network*, (ii) *the text attention network*, (iii) *the multimodal fusion module*. This section describes each component in detail. In this paper, we describe the proposal assuming the use of two types of features: map images and social/traffic event descriptions. Note that the proposed method can be easily extended to handle other types of features.

(i) Image attention network. We construct the *image network* by combining CNN with a self-attention mechanism, which extracts attention for regions of the image. Suppose we have a collection of map images $\{I_j\}_{j=1}^{J}$, $I_j \in \mathbb{R}^{N_\mathrm{w} \times N_\mathrm{h} \times N_\mathrm{c}}$, where N_w, N_h, N_c represent width, height, and the number of image features (e.g., three color channels), respectively. In the following discussion, we omit index j for the sake of simplicity. *The image attention network* accepts images I, and passes them through convolutional transformation followed by pooling and activation layers.

$$P = g_p(C_p * I),\ Q = g_q(C_q * I) \tag{2.7}$$

where $*$ denotes the convolution; C_p and C_q are the parameter matrices to be learnt; $g_p(\cdot)$ and $g_q(\cdot)$ are a set of activation and pooling operations. For our experiment, we use 3 $\times$ 3 same convolution so as to straightforwardly visualize the attention weights developed for the image features. Subsequently, we process P through a spatial attention model

consisting of a single self-attention layer followed by a softmax function:

$$A_m = \text{softmax}\big(M_2 \text{tanh}(M_1 P^\top)\big), \tag{2.8}$$

where $M_1 \in \mathbb{R}^{d \times d_a}$ and $M_1 \in \mathbb{R}^{r \times d_a}$ are parameter matrices. In the experiment, we set $d = N_c$, $d_a = 32, r = 1$. The attention weights $A_m \in \mathbb{R}^{N_\text{h} \times N_\text{w}}$ indicate which regions of the image were focused on during training. Then, we multiply the intermediate map P by the attention A_m. The output of the self-attention layer, B, is processed by a three layer CNN with a set of 3×3 convolutions. The output is then processed by two fully connected layers, with size of 512, and the rectified linear unit (ReLU) activation functions.

(ii) Text attention network. Social/traffic event descriptions are represented as a sequence of words $\{W_j\}_{j=1}^J$, $W_j \in \mathbb{R}^{N_\text{s} \times N_\text{v}}$, where N_s is the length of the sentence and N_v is the vocabulary size. We design the *text network* on the CNN designed for sentences [39] and an attention mechanism [52]. First, *the text attention network* reads the input sequence of 1-of-K word vectors $W = [\mathbf{w}_1, ..., \mathbf{w}_{N_\text{s}}]$, $\mathbf{w} \in \{0, 1\}^{N_\text{v}}$ and transforms it into a set of hidden vectors $H = [\mathbf{h}_1, ..., \mathbf{h}_{N_\text{s}}]$, where $\mathbf{h}_i$ is a r-dimensional vector. We then feed the vectors into the attention network. In particular, we transform the set of hidden vectors $H = [\mathbf{h}_1, ..., \mathbf{h}_{N_\text{s}}]$ into new vectors with dimension d_c such that

$$A_t = \text{softmax}\big(T_2 \text{tanh}(T_1 H^\top)\big), \tag{2.9}$$

where $T_1 \in \mathbb{R}^r$ and $T_1 \in \mathbb{R}^r$ are parameter matrices. We set $r = 1$ in our experiment. $A_t \in \mathbb{R}^{N_\text{s}}$, the attention weight for each word, reflects the importance of the word. We multiply the hidden vectors H with the attention weights A_t, and feed the results through a three layer CNN. The output of the CNN is transformed by two fully connected layers with size of 8, ReLU activation functions, and dropout of 0.1.

(iii) Multimodal fusion module. The positions of the representative points, $\mathbf{u}$, are processed by two fully connected layers with 32 units and relu activations. Their output and the outputs of the attention network (*the image attention network* or *text attention network*) are concatenated. This is followed by fully connected layers with relu activation functions. The hyper-parameters of the last fully connected layers for *the multimodal fusion module* are tuned on the validation set.

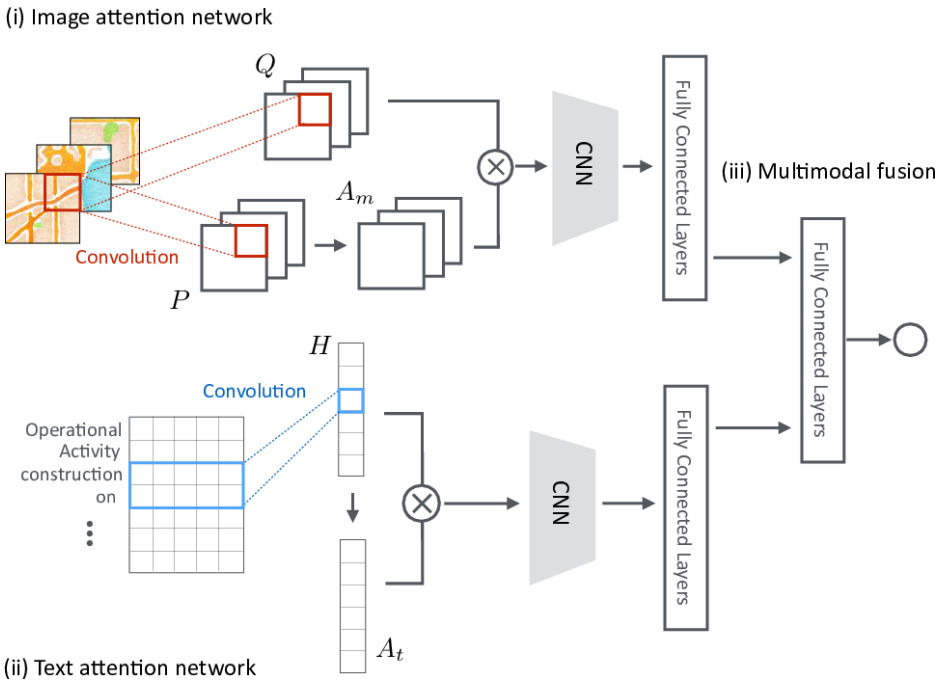


Figure 2.1: The architecture of the neural network used in the proposed method.

2.4.3 Parameter Learning

Given a list of observed events up to time T (total of N events) $\mathcal{X}$, the logarithm of the likelihood function is written as

$$\begin{aligned}\log p(\mathcal{X}|\lambda(\mathbf{x})) &= \sum_{i=1}^{N} \log \lambda(\mathbf{x}_i|\mathcal{D}) - \int_{\mathbb{T}\times\mathbb{S}} \lambda(\mathbf{x}|\mathcal{D})d\mathbf{x} \\ &= \sum_{i=1}^{N} \log \sum_{j=1}^{J} f(\mathbf{u}_j, \mathbf{z}_j; \theta) k(\mathbf{x}_i, \mathbf{u}_j) - \sum_{j=1}^{J} f(\mathbf{u}_j, \mathbf{z}_j; \theta) \int_{\mathbb{T}\times\mathbb{S}} k(\mathbf{x}, \mathbf{u}_j)d\mathbf{x}, \quad (2.10)\end{aligned}$$

where $\mathbb{T} \times \mathbb{S}$ is the domain of the observation. Notably, the above log-likelihood can be solved tractably with integratable kernel functions. Our mixture model-based approach with representative points allows the neural network model $f(\cdot)$, which cannot be integrated analytically in general, to be moved outside the integral. This permits us to use the simple back-propagation algorithm. For many well-known kernel functions, such as Gaussian, polynomial, the integral of the second term is written as closed-form solutions or approximations. In the case of the Gaussian kernel, it is described by an error function. During the training phase, we adopt mini-batch optimization. Over the set of indices selected in a mini-batch $\mathcal{I}$, by normalizing the first term in (3.13), the objective function can

be written as

$$\log p(\mathcal{X}|\lambda(\mathbf{x})) = \frac{N}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\log\sum_{j=1}^{J} f(\mathbf{u}_j,\mathbf{z}_j;\theta)k(\mathbf{x}_i,\mathbf{u}_j) - \sum_{j=1}^{J} f(\mathbf{u}_j,\mathbf{z}_j;\theta)\int_{\mathbb{T}\times\mathbb{S}} k(\mathbf{x},\mathbf{u}_j)d\mathbf{x}, \tag{2.11}$$

where $|\mathcal{I}|$ denotes the mini-batch size. We apply back-propagation to find all the model parameters, $\Theta = \{\Sigma, \theta\}$, that maximize the above log-likelihood, by taking the derivative of (2.11) w.r.t. kernel parameter Σ and neural network parameters θ.

2.4.4 Prediction

Here we present a procedure for future event prediction.

We denote representative points within the test period $\mathbb{T}^* = (T, T + \Delta T]$ as

$$\mathcal{U}^* = \{(\tau, \mathbf{r}) \mid T < \tau \leq T + \Delta T\} \subset \mathcal{U}. \tag{2.12}$$

Given the learned parameters of the neural network, $\hat{\theta}$, we first calculate $f(\mathbf{u}_j, \mathbf{z}_j; \hat{\theta})$ for each representative point. Using the set of estimated functions $\{f(\mathbf{u}_j, \mathbf{z}_j; \hat{\theta})\}_{\mathbf{u}_j\in\mathcal{U}^*}$ and the estimated kernel parameter $\hat{\Sigma}$, we derive intensity $\hat{\lambda}(\mathbf{x})$ for the test period based on (4.3).

Given the sequence of events observed in the test period $\mathbb{T}^*$, $\mathcal{D} = \{\mathbf{x}_{N+1}, ..., \mathbf{x}_{N+n}\}$, analogous to (3.13), the log-likelihood for the test data is calculated as

$$\begin{aligned}\mathcal{L}^* = \log p(\mathcal{D}|\hat{\lambda}(\mathbf{x})) = &\sum_{i=N+1}^{N+n}\log\sum_{\mathbf{u}_j\in\mathcal{U}^*} f\big(\mathbf{u}_j,\mathbf{z}_j;\hat{\theta}\big)k(\mathbf{x}_i,\mathbf{u}_j)\\ &- \sum_{\mathbf{u}_j\in\mathcal{U}^*} f\big(\mathbf{u}_j,\mathbf{z}_j;\hat{\theta}\big)\int_{\mathbb{T}^*\times\mathbb{S}} k(\mathbf{x},\mathbf{u}_j)d\mathbf{x}.\end{aligned} \tag{2.13}$$

The point process model can be used to predict the expected number of events. The number of events is derived by integrating the estimated intensity over specific time period $P \subset \mathbb{T}^*$ and region of interest $Q \subset \mathbb{S}$ such that

$$N(P\times Q) = \int_{P\times Q}\hat{\lambda}(\mathbf{x})d\mathbf{x} = \sum_{\mathbf{u}_j\in\mathcal{U}^*} f\big(\mathbf{u}_j,\mathbf{z}_j;\hat{\theta}\big)\int_{P\times Q} k(\mathbf{x},\mathbf{u}_j)d\mathbf{x}, \tag{2.14}$$

where $N(A)$ is the number of events that fall into subset A. As discussed in Section 4.3, the above integral has a tractable solution.

2.5 Experiments

In this section, we use real-world data sets from different domains to evaluate the predictive performance of our model.

2.5.1 Datasets

Event data

We used three event data sets from different domains collected in New York City and Chicago from Jan 1, 2016 to April 1, 2016 (the observation period is 13 weeks). The details are as follows.

NYC Collision Data. New York City vehicle collision (NYC Collision) data set contains $\sim$ 32 thousand motor vehicle collisions. Every collision is recorded in the form of time and location (latitude and longitude coordinates).

Chicago Crime Data. Chicago crime data set is a collection of reported incidents of crime that occurred in Chicago; it contains $\sim$ 13 thousand records, each of which shows time, and latitude and longitude of where the crime happened.

NYC Taxi Data. New York City taxi pick-up (NYC Taxi) data set consists of $\sim$ 30 million pick-up records in New York City collected by the NYC Taxi and Limousine Commission (TLC). Each record contains pick-up time, latitude and longitude coordinate. To reduce data size, we randomly selected 100 thousand events for our experiment.

For the 13-week observation period, we selected the last seven days as the test set, the last seven days before the test period as the validation set, and used the remaining data as the training set. Thus, $T = 120960$min and $\Delta T = 10080$min.

Urban contextual data

We used the following urban data as the contextual features.

Map Image. As the image features, we used *the map image* of the cities acquired from OpenStreetMap (OSM) database [1]. For each representative point $\mathbf{u}_j = (\tau_j, \mathbf{r}_j)$, we extracted the image around $\mathbf{r}_j$ (i.e., about 300m$\times$ 500m square grid space) and used its RGB vector as the input of the *image attention network*.

Social/Traffic Event Description. We collected *traffic events* (e.g., major street construction works and street events) and *social events* (e.g., sports events, musical concerts and festivals) in New York City as held by the 511NY website[2] during Jan 2016 through

[1] Map data copyrighted OpenStreetMap contributors and available from https://www.openstreetmap.org
[2] https://www.511ny.org

April 2016. The 13 week period contained a total of 8,968 descriptions. Each social/traffic event record contains a description of the event, as well as its start time t_s, end time t_e and location (latitude and longitude coordinates). For each representative point $\mathbf{u}_j = (\tau_j, \mathbf{r}_j)$, we extracted social/traffic event descriptions satisfying $t_s < \tau_j < t_e$ and located within a predefined distance, Δr from $\mathbf{r}_j$. If the point contains more than one sentence, we selected the spatially closest one. If the point contains no sentences, we used dummy variables. We used their descriptions, a sequence of 1-of-K coded word vectors, as the input of the text network. In this paper, we set Δr to the walking distance of 620m, following [12].

2.5.2 Experimental Setup

Hyper-parameters of each model are tuned by grid-search on the validation set. For DMPP, we used the Adam algorithm [42] as the optimizer, with $\beta_1 = 0.01$, $\beta_2 = 0.9$, and learning rate of 0.01. For *the multimodal fusion module* of DMPP, we tuned the hyper-parameters as follows: layer size n_l in {1,2,3,4}; number of units per layer n_u in {16, 32, 64}. The mini-batch size $|\mathcal{I}|$ is selected from the set {8, 16, 32}. Following the prior settings in [21], we also applied L2 regularization with λ=0.001 in both models. The number of representative points are tuned on the validation set in terms of the number of time points, M, and the number of space points, L. For each representative point, we extract a 20×20 image patch from the map image of the entire region-of-interest (i.e., Manhattan for NYC Collision data and NYC Taxi data, City of Chicago for Chicago Crime data), resize it to 10×10 pixels, and use its RGB vector as the input image. This corresponds to 290m$\times$ 500m square grid space for NYC Collision data and NYC Taxi data, and 290m$\times$ 600m square grid space for Chicago Crime data. Therefore, $N_w = 10$, $N_h = 10$, $N_c = 3$. In *the text network*, we only consider the first 200 most frequent words, and use the first 5 words of each sentence. For our input descriptions, we zero-pad to ensure a sentence length of 5 words. Thus, $N_v = 200$ and $N_s = 5$. The tested combinations were $M = \{24, 28, 168\}$ and $L = \{4, 8, 10, 12\}$. We used three kinds of kernel functions: uniform, Gaussian, compactly supported Gaussian (the definitions are provided in Section 2.4.2). Also, we explored various map styles: OSM default (the original map of Figure 2.8a), Watercolor (the original map of Figure 2.8b), Greyscale. The best settings for DMPP are given in the corresponding section.

2.5.3 Evaluation Metrics

We evaluated the predictive performance using two metrics: **LogLike** (predictive log-likelihood) and **MAPE** (Mean Absolute Percentage Error). For the first metric, given the learned model, we calculated log-likelihood on the test data (**LogLike**) for each event as

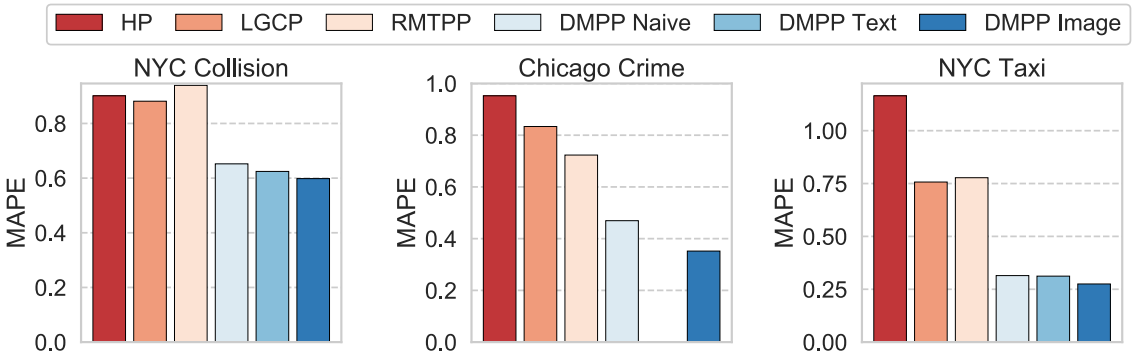


Figure 2.2: MAPE for event number prediction from six methods on three data sets: NYC Collision data (left); Chicago Crime data (middle); NYC Taxi data (right). Lower is better. DMPP is the proposed method. The error bars are omitted as the deviations are negligible.

$\mathcal{L}^*/N_t$, where $\mathcal{L}^*$ is the test log-likelihood defined by Equation (7) and N_{t} is the number of test events. **MAPE** is used to evaluate the performance of event number prediction; it is defined as the absolute difference between the predicted number of events and the actual number:

$$\text{MAPE} = \sum_{r=1}^{N_r} \sum_{t=1}^{N_b} |n_{r,t} - \hat{n}_{r,t}|/n_{r,t}, \tag{2.15}$$

where $n_{r,t}$ is the number of events observed in the r-th grid cell and t-th time interval, and $\hat{n}_{r,t}$ is the corresponding prediction. N_r is the number of grid cells and N_b is the number of time bins for which predictions are made. In our experiment, we partitioned the region of interest using a 10×10 uniform grid, and divided the test period (seven days) into 14 time bins with a fixed uniform interval of 12 hours. Therefore $N_r = 100$ and $N_b = 14$. For DMPP, we predicted the number of events for each pair of spatial grid cell and future time bin, using Equation (6).

2.5.4 Comparison Methods

We compared the proposed model and its variants with three existing methods.

- HP (Homogeneous Poisson process): The intensity is assumed to be constant over space and time: $\lambda(\mathbf{x}) = \lambda_0$. The optimization can be solved in closed form. Given the test period $[T, T + \Delta T]$ and the region of interest $\mathbb{S}$, the likelihood of HP is written as

$$\log p(\mathcal{X}|\lambda(\mathbf{x}) = \lambda_0) = n \log \lambda_0 - \lambda_0 \Delta T |\mathbb{S}|, \tag{2.16}$$

where n is the number of test samples, ΔT is the length of the test period, and $|\cdot|$ is the operator providing the area of a spatial region.

- LGCP (Log Gaussian Cox process) [20]: LGCP is a kind of Poisson process with varying intensity, where the log-intensity is assumed to be drawn from a Gaussian process (See Appendix C). For LGCP, we performed the comparison only on event number prediction, since the log-likelihood of this model is computationally intractable. The inference is based on the Markov chain Monte Carlo (MCMC) approach (see [92] for details). For event number prediction, we sampled events from LGCP using the thinning method [50], and compared the aggregated number of events within predefined spatial regions and time periods with the ground truth. We implement the general LGCP method described in [92], whose intensity is defined as

$$\lambda(\mathbf{x}) = \mu(t)\psi(s)\exp\big(y(\mathbf{x})\big), \tag{2.17}$$

where $\mu(t)$ and $\psi(s)$ are temporal and spatial background rates, respectively. $y(\cdot)$ is a Gaussian process with the following covariate function:

$$\mathrm{cov}(\mathbf{x}, \mathbf{x}') = \sigma_{\mathrm{GP}}^2 \exp\left(-(\mathbf{x}-\mathbf{x}')^\top \theta_{\mathrm{GP}}(\mathbf{x}-\mathbf{x}')\right), \tag{2.18}$$

where σ_{GP} is the scale parameter, θ_{GP} is the bandwidth.

- RMTPP (Recurrent Marked Temporal Point Process) [21]: RMTPP uses RNN to describe the intensity of the marked temporal point process; it assumes a partially parametric form for the intensity, and can capture temporal burst phenomena. This model is primarily intended to model event timing; to allow comparison, we mapped latitude and longitude values into location names and treating them as marks, using Neighborhood Names GIS data [3,4] (details are provided in Appendix C). Finally, we obtained 48 unique locations for NYC Collision and NYC Taxi data, 44 for Chicago Crime data. The following hyper-parameters are tuned on the validation set: Number of units per layer in {16, 32, 64} and mini-batch size in {8, 16, 32}. We set unit size as 16 and batch size as 32 for NYC Collision data, unit size as 16 and batch size as 8 for Chicago Crime data, unit size as 16 and batch size as 16 for NYC Taxi data. We use the log-likelihood (LogLike) as defined in [21]. Note that the likelihood of

[3]NYC Neighborhood Names GIS data, https://data.cityofnewyork.us/City-Government/Neighborhood-Names-GIS/99bc-9p23

[4]Chicago Neighborhood Names GIS data, https://data.cityofchicago.org/Facilities-Geographic-Boundaries/Boundaries-Neighborhoods/bbvz-uum9

Table 2.1: Comparison of the proposed method and its variants to the baselines. The number indicates the predictive log-likelihood per event (**LogLike**) for the test data. Higher is better.

	NYC Collision	Chicago Crime	NYC Taxi
HP	8.232	9.384	10.473
DMPP *Naive*	8.296	9.614	11.311
DMPP *Text*	8.297	NA	11.318
DMPP *Image*	**8.409**	**9.621**	**11.333**

RMTPP is for the location names, not for latitude and longitude values. For event number prediction, we generate the sequence of events by sequentially predicting the timing of the next event. In particular, given the event sequence $\{t_1, ..., t_N\}$, we compute the timing of next event $\hat{t}_{N+1}$, using Equation (13) in [21]. Using $\hat{t}_{N+1}$ as known data, we then predict the timing of next event $\hat{t}_{N+2}$ based on the new sequence $\{t_1, ..., t_N, \hat{t}_{N+1}\}$. This procedure is repeated until $\hat{t}_{N+i} > T + \Delta T$. The predicted location names are mapped to latitude and longitude coordinates by simply using their centroids; and then the generated events are aggregated into counts.

We introduce three variants of DMPP below.

- DMPP *Naive*: The simplest variant of DMPP, it does not incorporate any contextual features. The neural network of DMPP accepts the location and time of each representative point $\mathbf{u}_j$.
- DMPP *Image*/*Text*: The DMPP variants that incorporate either map images or the social/traffic event descriptions, as well as the locations and times of the representative points.

2.5.5 Environment

RMTPP and our DMPP are implemented using the Chainer deep network toolkit [93]. All the methods are run on a Linux server with an Intel Xeon CPU, and a GeForce GTX TITAN GPU. The GPU code is implemented using CUDA 9.

2.5.6 Quantitative Results

Figure 2.2 shows the overall MAPE of the six different methods on the three data sets for event number prediction. In this figure, the error bars are omitted as the deviations are

negligible. The results indicate the superiority of our approach. HP performs worse than the other methods across almost all data sets, as it does not consider the spatio-temporal variation of the rate of event occurrence. We can see this in Figure 2.3, which depicts events generated by four different methods from the Chicago Crime data. We simulated events with the thinning algorithm [50], using the learned intensity of each method. LGCP presents better performance for NYC Collision and NYC Taxi data, as it captures spatio-temporal variations. RMTPP achieves relatively better performance than LGCP only for the Chicago Crime data. The result suggests that the assumption of RMTPP, the temporal burst phenomena, holds for Chicago Crime data, but not for NYC Collision data and NYC Taxi data. Even our simple model, DMPP *Naive*, largely surpasses all existing methods. The result implies that the parametric assumptions of the existing methods are too restrictive, and do not capture real urban phenomena. Also, RMTPP intensity is influenced by all the past events, regardless of how spatially far away, so it is not suited for spatio-temporal events. The differences between DMPP *Naive* and the best among the existing methods are significant (two-sided t-test: p-value$<$ 0.01) for all data sets. DMPP *Naive* outperforms LGCP in terms of MAPE by 0.229 for the NYC Collision data, from 0.778 to 0.315 for the NYC Taxi data. DMPP *Naive* outperforms RMTPP in terms of MAPE by 0.254 for the Chicago crime data. DMPP *Text* further improves DMPP *Naive* to 0.624 for the NYC Collision data, to 0.312 for NYC Taxi data. We can clearly see that DMPP *Image* offers significantly improved prediction performance. From these results, we can conclude that considering urban contexts is very effective in improving event prediction performance. The results also suggest that our proposal, DMPP, effectively utilizes the information provided by urban contexts.

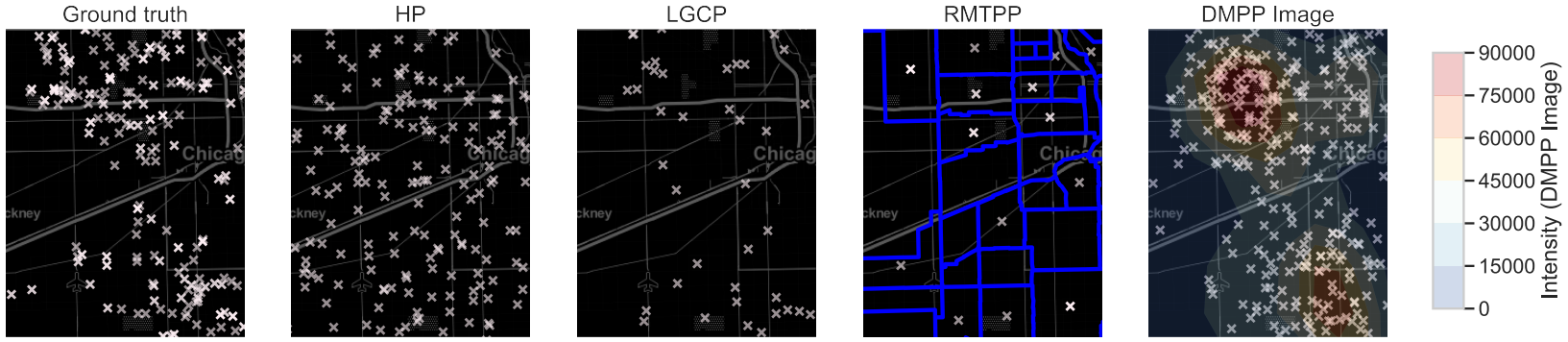
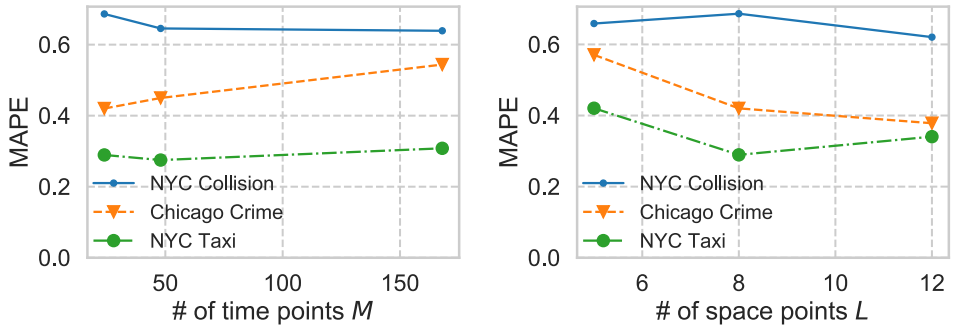


Figure 2.3: Events generated by four of the implemented methods for Chicago Crime data in Central and West Chicago between 0:00 am and 24:00 pm, on Mar 31th. The cross markers (x) denote the events generated by simulations. In the fourth plot, the blue lines denote the location boundaries used for RMTPP in the experiment. In the right-most plot, we overlaid the estimated intensity for DMPP *Image* at Mar 31th 12:00 am.



(a) Number of time points

(b) Number of space points

Figure 2.4: Impact of numbers of representative points on MAPE performance of DMPP *Image*.

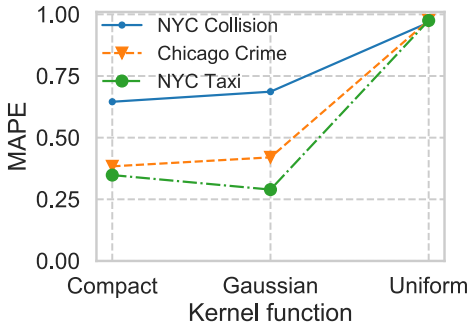
Table 2.1 lists the LogLike values (predictive log-likelihood) of the four different methods for the three data sets, i.e., NYC Collision (vehicle collision) Data, Chicago Crime Data and NYC Taxi (taxi pick-up) Data. Note that DMPP *Text* is not applicable to Chicago Crime data, as no text data is available for Chicago. The proposal, DMPP, outperforms HP. Even the simplest variant of DMPP, DMPP *Naive* explains the observed event sequences better than these existing methods, which demonstrates the expressiveness of DMPP. DMPP *Text* also outperforms HP. DMPP *Image* achieves the best performance among all methods. This again shows the effectiveness of incorporating the urban contexts and our point process formulation with the deep neural network.

Sensitivity study

Here we analyze the impact of parameters on DMPP, including (1) number of representative points; (2) kernel function (3) map style; and (4) neural network structure.

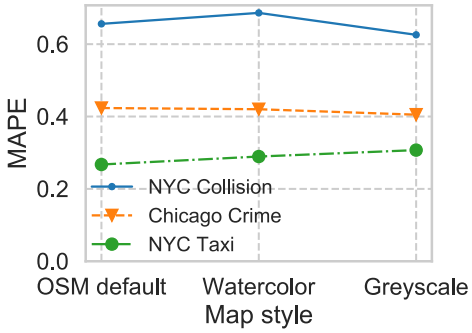
Number of Representative Points. Figure 2.4 shows the impact of the numbers of representative points on the performance of DMPP *Image*. Figure 2.4(a) shows that MAPE is slightly improved when $M = 168$ for NYC Collision data, $M = 48$ for Chicago Crime data, and $M = 24$ for NYC Taxi data. As shown in Figure 2.4(b), the prediction performance of DMPP generally tends to increase with number of time points M. Overall, DMPP is moderately robust to variations in the numbers of representative points. In this experiment, we fixed the network depth n_l to 4, the number of units per layer n_u to 32 and batch size $|\mathcal{I}|$ to 16.

Choice of Kernel Function. Figure 2.5 presents the prediction results with three kernel functions: Uniform, Gaussian, compactly supported Gaussian(definitions are pro-



(a) Kernel function

Figure 2.5: Impact of kernel functions on MAPE performance of DMPP *Image*.



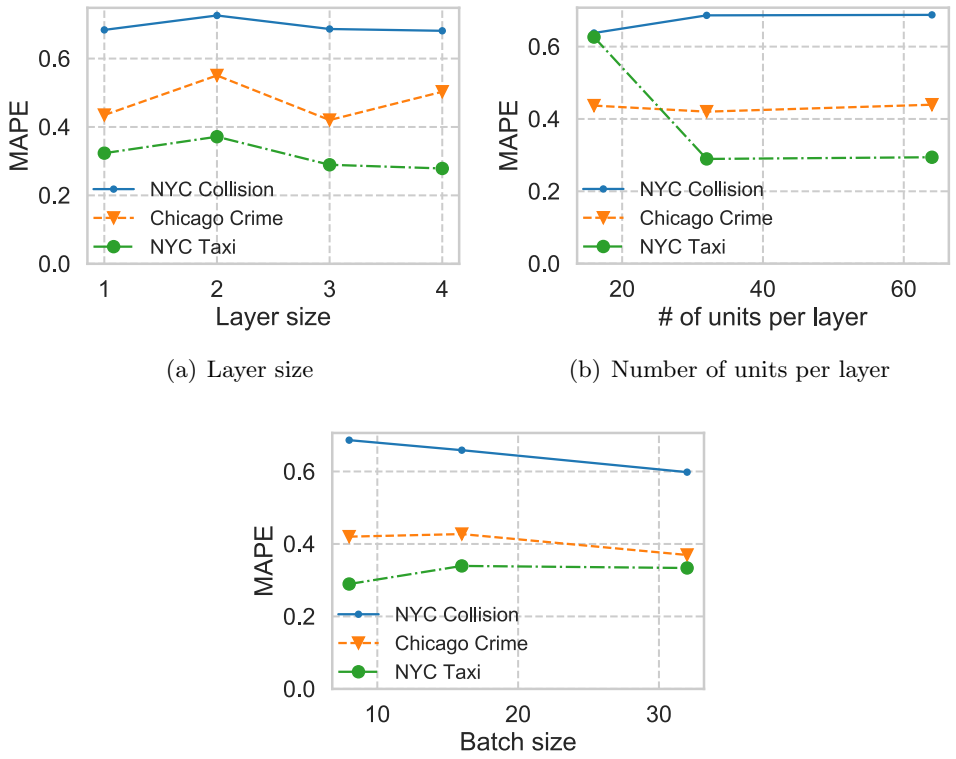
(a) Map style

Figure 2.6: Impact of map style on MAPE performance of DMPP *Image*.

vided in Appendix B). We can observe that the compactly supported kernel offers similar accuracy to the Gaussian kernel, while affording a computational advantage (See Appendix B). The uniform kernel performs worst. Throughout this paper, we use the compactly supported Gaussian kernel as the default setting. The hyper-parameters were set to $n_l = 4$, $n_u = 32$, $|\mathcal{I}| = 16$, $M = 24$ and $L = 20$ in this experiment.

Choice of Map Style. Figure 2.6 demonstrates the effect of map style. The predictive performance appears to be insensitive to the style of map images. For NYC Taxi data, MAPE is slightly improved when using the OSM default style. This may because the OSM default map distinguishes minor and major roads by color (as shown in the original map image of Figure 2.8b). As the default setting, we use OSM default style for NYC Collision and NYC Taxi data, and Watercolor for Chicago Crime data. In this experiment, we set $n_l = 4$, $n_u = 32$, $|\mathcal{I}| = 16$, $M = 24$ and $L = 20$.

Network Structure. We show the impact of network structures in Figure 2.7. The prediction performance slightly improves when layer size is 4, for NYC Collision data and



(a) Layer size

(b) Number of units per layer

(c) Batch size

Figure 2.7: Impact of network structures on MAPE performance of DMPP *Image*.

NYC Taxi data, layer size 3 for Chicago Crime data. DMPP performs robustly for large number of units $n_u >= 32$ across all the data sets. The prediction accuracy is likely to become better for larger batch size (Figure 2.7(c)). In this experiment, we fix $M = 24$ and $L = 12$, respectively. The optimal value of network hyper-parameters correspond to $n_l = 4$, $n_u = 16$, $|\mathcal{I}| = 32$ for NYC Collision data, $n_l = 3$, $n_u = 64$, $|\mathcal{I}| = 32$ for Chicago Crime data, $n_l = 4$, $n_u = 64$, $|\mathcal{I}| = 8$ for NYC Taxi data.

In conclusion, DMPP is moderately robust to variations in the hyper-parameters, and so can yield steady performance under different conditions.

2.5.7 Qualitative Results

To demonstrate that our model provides useful insights as to why and under which circumstances events occur, we analyze what was learned by our method.

Figure 2.8 visualizes the learned attention for the map images from NYC Taxi data (Figure 2.8a) and Chicago Crime data (Figure 2.8b). Here we fed the map images (left) into the learned *image attention network*, and plot the output attention weights (right).

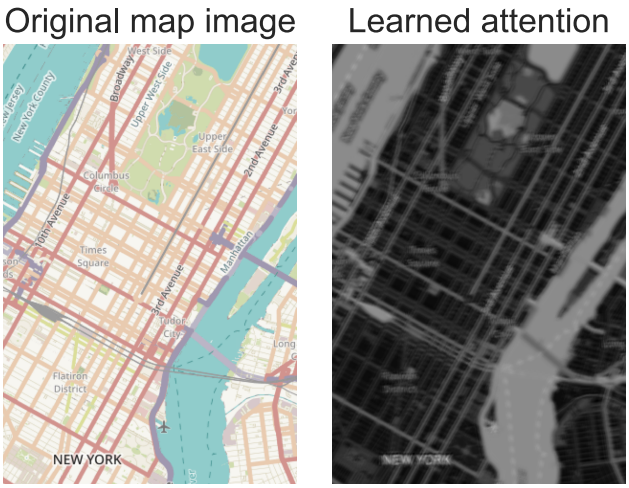
In the attention heatmaps (right), the light and dark regions correspond to high and low attention weights, respectively. We can see that DMPP assigns high attention weights to major roads (depicted by pink in the original map image) for the NYC Taxi data in Figure 2.8a. The minor roads (depicted by light orange) draw less attention. This indicates that taxi pick-ups take place mostly on roads, especially on major roads. Interestingly, for Chicago Crime data (Figure 2.8b), the attention mechanism weights both the roads and land cover. Apparently, the roads have the highest attention weights. This may because crime is found both on roads and in building. These results suggest that our method can elucidate the key spatial components related to event occurrence.

Figure 2.9 shows the attention weights for the event descriptions overlaid with the learned intensity around Midtown Manhattan from Mar 24th to 31th. For the sake of clarity, only the first word of each sentence is depicted. The word *Special* appears usually with *event*; *Operational* stands for *Operational (activity)*. The darker shade of red for the texts indicates higher attention values. In the heatmaps, red corresponds to high intensity value, while blue represents low value. For NYC Taxi data (Figure 2.9a), the learned intensity yields high values in Midtown and Murrey Hill of Manhattan. Seemingly, the attention mechanism also highlights the words associated with these regions. Mainly words related to the social events, e.g., *Concert* and *Special (event)*, are highlighted. In contrast, the words associated with traffic events, such as *Construction* and *Operational (activity)*, gain more attention, in the NYC Collision data (Figure 2.9b). The above results suggest that traffic events affect the collision rate, whereas social events drive the taxi pick-up demand. Figure 2.10 further supports this. It visualizes the top 15 words ranked by attention weight learned from each data set; larger size denotes higher attention. For NYC Taxi data, social events (e.g., *special (event)* and *concert*), as well as traffic event (e.g., *construction*), tend to receive attnetion. For NYC Collision data, traffic events, e.g., *construction* and *operational (activity)*, seem to draw more attention. These results demonstrate that DMPP identifies important words that affect event occurrence. The descriptions thus found help us explain why and in which contexts events occur.

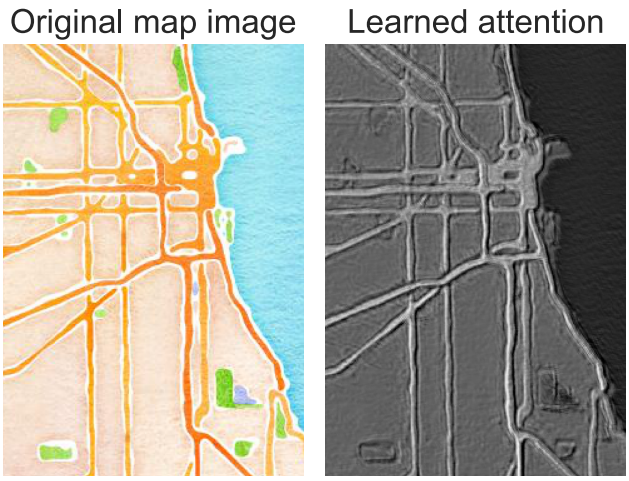
2.6 Conclusion and Future work

In this chapter, we studied the problem of event prediction with the use of rich contextual features. Our solution, DMPP (Deep Mixture Point Process), is a novel point process model based on a deep learning approach. DMPP models the point process intensity by a deep mixture of kernels. The key advantage of DMPP over existing methods is that it can utilize the highly-dimensional and multi-sourced data provided by rich urban contexts,

including images and sentences, and automatically discover their complex effects on event occurrence. Moreover, by taking advantage of the mixture model-based approach, we have developed an effective learning algorithm. Using real-world data sets from three different domains, we demonstrated that the proposed method outperforms existing methods in terms of prediction accuracy.

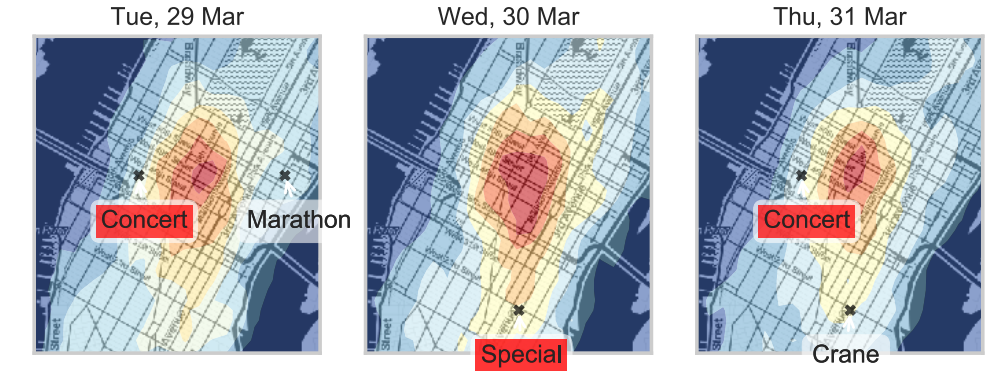


(a) NYC Taxi

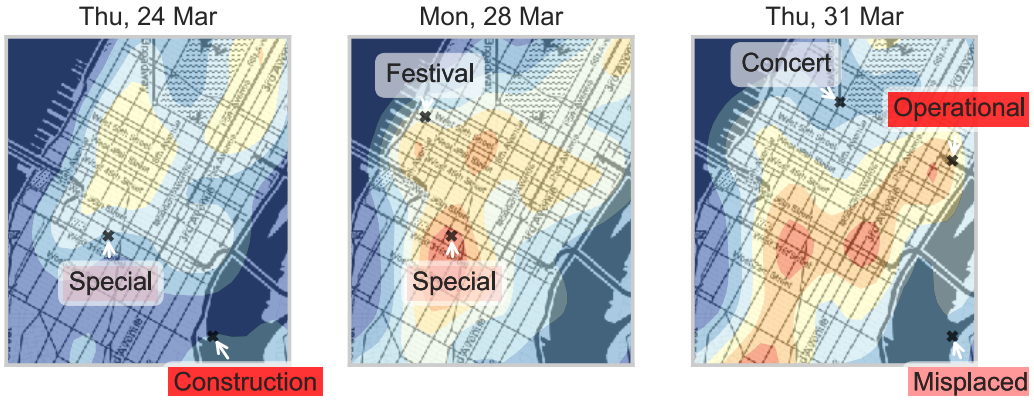


(b) Chicago Crime

Figure 2.8: Attention weights for the map images learned from NYC Taxi data and Chicago Crime. Left: original map image. Right: learned attention weights; lighter shade indicates stronger attention values.



(a) NYC Taxi



(b) NYC Collision

Figure 2.9: Learned attention weights for social/traffic descriptions with the learned intensity around Midtown Manhattan from Mar 28th to Apr 4th. Darker shade of red for the texts denotes higher attention weight.

(a) NYC Taxi (b) NYC Collision

Figure 2.10: Word cloud of top 15 words by attention weight; larger size denotes higher attention.

Chapter 3

Context-aware Spatio-temporal Event Prediction via Convolutional Hawkes Processes

3.1 Introduction

Spatio-temporal event data are being accumulated in many important fields such as health care and public safety. Such data contains time and location, indicating when and where events have happened. For example, electronic health records are represented as a sequence of events with locations and times of disease outbreaks. Armed conflicts are recorded with locations and times at which the conflicts took place.

A wide range of event sequences are demonstrations of spatio-temporal processes that have "self-exciting" or triggering patterns. In all the aforementioned examples, event occurrence is triggered by preceding events. For instance, disease outbreaks can ignite secondary outbreaks, often leading to epidemics. A conflict between rival ethnic groups may trigger a cycle of retaliation.

Modeling such triggering processes and predicting future events is crucial for realizing many applications such as disease control and harmonizing global politics. For instance, if local health authorities can predict when, where and which events will trigger disease outbreaks, they can make more effective intervention policies [97]. Better understanding and prediction of conflicts will help governments take more appropriate actions to reduce life and economic losses.

Hawkes process is a general mathematical framework for modeling triggering processes; it is characterized by a *conditional intensity* that describes the rate of events occurring

at any location and at any time. Hawkes process has been adopted for modeling a wide spectrum of events, including infectious disease [80], terrorist attacks [78], crimes [59] and earthquakes [68]. However, these models fail to adequately depict the real diffusion process, since its conditional intensity is modeled as a function of spatio-temporal distance, and the impact of contextual factors on triggering processes is ignored. Real-world triggering processes are determined not only by the spatio-temporal relationship between events but also by contextual factors such as population distribution, weather, road network, and terrain. These contextual features can be spatially heterogeneous and change over time. For example, infectious diseases spread among high population areas [60]. The transmission of diseases is also influenced by other contextual factors, including trading patterns [64], land use [76] and weather [75]. Conflicts tend to be more accentuated in densely populated areas [**?**].

In this thesis, we develop two promising approaches for capturing the effect of contextual factors on the triggering process based on two different assumptions: Contextual information is fully observable and unobservable. In this chapter, we consider the case where rich contextual information sets are becoming accessible. For example, with the development of remote sensing techniques, high-resolution satellite images are being collected and are available at various spectral, spatial, and temporal resolutions. Also, open-source GIS platforms have become commonplace; they provide geographic features including road network and land use, in the form of a colored map. These images contain meaningful information that can rarely be found in traditional information sources, and offer detailed spatial patterns of various contextual factors, ranging from human demography to weather and land use, as well as their temporal variations.

Several studies [57, 40, 86] have extended Hawkes process to incorporate contextual factors, e.g., regional populations [57], mobility flows between regions [40] and weather conditions [86]. But these methods are based on hand-crafted features engineered by domain experts and make a simplified assumption on the conditional intensity as a function of these features. Thus these methods cannot handle unstructured data like images, which contain rich, meaningful information.

In this chapter, we propose an event prediction method that effectively utilizes the rich contextual features present in georeferenced images. Inspired by the recent success of deep learning models in computer vision [95, 113], we use them to enhance the Hawkes process model. The most straightforward way is to directly replace the Hawkes process intensity with a neural network that accepts these images as its input. Although this approach enables the automatic discovery of meaningful information from the images and

thus improve event prediction performance, it suffers from the intractable optimization problem, as integral computations are required to determine the likelihood needed for estimation.

We solve this by introducing a novel architecture for Hawkes processes. In particular, we extend a convolutional neural network (CNN) by combining it with continuous kernel convolution; the conditional intensity of Hawkes process is designed on the extended model. Our approach of using the continuous convolution kernel provides a flexible way of learning the complex contextual features present in the images, allowing us to capture the spatial heterogeneity of the triggering process. Notably, our formulation permits the likelihood to be determined by tractable integration. In the proposed method, referred to as Convolutional Hawkes process (ConvHawkes), the parameters of the neural network and the convolutional kernel can be simultaneously optimized to maximize the likelihood by using gradient-based algorithms.

We conduct experiments on three real-world datasets from multiple domains and show that ConvHawkes consistently outperforms existing methods in event prediction tasks. The experiments also demonstrate that ConvHawkes provides a better understanding of the underlying mechanisms by which various contextual factors influence the triggering processes.

The main contributions of this chapter are as follows:

- We propose a novel Hawkes process model, ConvHawkes (Convolutional Hawkes process) for modeling diffusion processes and predicting spatio-temporal events. It accurately and effectively predicts spatio-temporal events by leveraging the contextual features contained in georeferenced images (e.g., satellite images and map images), that impact triggering processes.

- We present an extension of the neural network model and integrate it into the Hawkes process framework. This formulation allows us to utilize the contextual features present in the unstructured image data, and to automatically discover their complex effects on the triggering process, while at the same time yielding tractable optimization.

- We conduct extensive experiments on real-world datasets from different domains. With regard to event occurrence, the proposed method achieves better predictive performance than several existing methods on all datasets.

3.2 Related Work

Spatio-temporal prediction constitutes an important problem with various applications such as public safety, transportation, health care, and environment. The conventional approach to this problem is regression. Early works are based on traditional machine learning methods, including classical time-series models like vector autoregression (VAR) [10, 125] and autoregressive integrated moving average (ARIMA) [94], and support vector regression (SVR) [117]. Recently, deep learning models have been successfully applied to this problem. For example, Ma *et al.* [55] and Zhao *et al.* [120] employ long short-term memory (LSTM) networks for traffic prediction, which captures the long-term temporal dependencies. Several studies [115, 114, 38] use convolutional neural networks (CNNs) to capture the non-linear spatial dependencies. Yao *et al.* [109] combine LSTM and CNN to jointly model both spatial and temporal dependencies in traffic data. In recent literature, graph neural networks (GNNs) have been adopted for spatio-temporal traffic graphs [112, 30, 118] and epidemic forecasting [40] to handle the complex spatio-temporal correlations. However, all the aforementioned methods focus on predicting the aggregated number of events within a predefined spatial region and time interval. This task is fundamentally different from ours. In this paper, we aim to directly model a sequence of events in continuous time and space, without aggregation, by using explicit information about location and/or time.

Point process is a powerful mathematical framework for modeling a sequence of events that occur in a continuous space and/or time domain. Hawkes processes [31] have been proven effective in describing the phenomenon of mutual excitation between events (i.e., triggering process); examples include earthquakes and aftershocks [61, 66, 124], gang-on-gang violence [53], terrorist attacks [78], near repeat crimes [59, 123], disease transmission [15, 80], financial transactions [5], and social activities [7, 27]. Early work made fixed parametric assumptions regarding the functional form of the conditional intensity, which is often too restrictive to depict real triggering process. Recent studies employ neural networks to enhance the expressiveness of point processes. For example, Xiao *et al.* [106] present a generative adversarial network-based framework for estimating the intensity of an inhomogeneous Poisson process. Chen *et al.* [13] leverage neural ODEs to parameterize marked temporal point processes. These models are based on inhomogeneous Poisson processes; they do not directly consider the influence of past events. Some other works [21, 56] propose to parameterize the intensity of temporal Hawkes processes by a recurrent neural network (RNN) to learn the non-linear influence from past events. Omi *et al.* [74] generalize the RNN-based Hawkes process model to further improve its expressive power. Transformer Hawkes process [126] and self-attentive Hawkes process [116] employ

a self-attention mechanism to capture the non-linear temporal correlation between events. These models focus on learning the temporal dependencies between events, and cannot be easily extended to account for the spatial aspect. More recent work [122] extends this approach to spatio-temporal Hawkes processes to consider both spatial and temporal domains. Despite the advances, all the above methods ignore the effects of external factors on the triggering processes.

Some efforts have been made to incorporate external features into Hawkes processes. For instance, several studies have proposed temporal Hawkes process methods that take account of external features such as population density [57], transportation networks [105, 3], human mobility patterns [40], weather [86, 59], fault structure [61]. However, it is still challenging to effectively utilize complex unstructured data like images.

Another line of work [70] takes account of the external features represented in images and texts by combining Poisson process modeling and deep neural network. However, the method of [70] assumes that events occur independently of one another, and thus does not adequately describe the triggering phenomena in which there exists strong interaction between events. We focus on the triggering process, and aim at capturing history-dependent and self-exciting phenomena such as diseases, armed conflicts and earthquakes.

3.3 Preliminaries

This section starts by providing the theoretical background to spatio-temporal Hawkes processes.

Point process is a random sequence of event occurrences over a domain. We assume here a sequence of events with known times and locations. Let $(t, \mathbf{s})$ be the event written as the pair of time $t \in \mathbb{T}$ and location $s \in \mathbb{S}$, where $\mathbb{T} \times \mathbb{S}$ is a subset of $\mathbb{R} \times \mathbb{R}^2$. We denote the number of events falling in subset A of $\mathbb{T} \times \mathbb{S}$ as $N(A)$. The general approach to identifying a point process is to estimate "intensity" function $\lambda(t, \mathbf{s})$. Intensity $\lambda(t, \mathbf{s})$ represents the rate of event occurrence in a small region. Given the history $\mathcal{H}(t)$ up to t, intensity is defined as

$$\lambda(t, \mathbf{s}|\mathcal{H}(t)) \equiv \lim_{|dt|\to 0, |d\mathbf{s}|\to 0} \frac{\mathbb{E}[N(dt \times d\mathbf{s})|\mathcal{H}(t)]}{|dt||d\mathbf{s}|}, \tag{3.1}$$

where dt is a small interval around time t, $|dt|$ is its length and $d\mathbf{s}$ is a small region containing location s, $|d\mathbf{s}|$ is its area. $\mathbb{E}$ is an expectation term. The functional form of intensity is designed to appropriately capture the underlying dynamics of event occurrence.

The Hawkes process is an important class of point process models, and its intensity is

modeled as the cumulative effects from all the past events $\mathcal{H}(t)$, represented by

$$\lambda\big(t, s|\mathcal{H}(t)\big) = \mu + \sum_{i:t_i<t} \alpha_i g(t - t_i, \mathbf{s} - \mathbf{s}_i), \tag{3.2}$$

where μ is a base intensity independent of the preceding events. t_i and $\mathbf{s}_i$ is the time and location of the i-th event; α_i is a constant that represents the strength of the influence of the i-th event; $g(\cdot) \geq 0$ is a triggering kernel that specifies the decaying effect of the i-th event. For computational simplicity, the triggering kernel function is often factorized into temporal and spatial components as follows:

$$g(t - t_i, \mathbf{s} - \mathbf{s}_i) = g_1(t - t_i)g_2(\mathbf{s} - \mathbf{s}_i), \tag{3.3}$$

where $g_1(\cdot)$ and $g_2(\cdot)$ are temporal and spatial decay functions, respectively. Typical choices for the temporal decay function include power-law, exponential, and Rayleigh functions [58]. Gaussian kernel is commonly used as the spatial decay function.

Given a sequence of events, $\mathcal{D} = \{(t_n, \mathbf{s}_n)\}_{n=1}^N$, $t_n \in \mathbb{T}$ and $\mathbf{s}_n \in \mathbb{S}$, the likelihood is given by

$$p\big(\mathcal{D}|\lambda(t, \mathbf{s})\big) = \prod_{n=1}^{N} \lambda(t_n, \mathbf{s}_n) \cdot \exp\bigg(- \int_{\mathbb{T}\times\mathbb{S}} \lambda(t, \mathbf{s}) dt d\mathbf{s} \bigg). \tag{3.4}$$

3.4 Problem Definition

This subsection formally defines the problem of spatio-temporal event prediction.

Event Sequence. Each event is represented by the tuple $(t, \mathbf{s})$, where $t \in \mathbb{T} \subseteq \mathbb{R}$ denotes its time and $\mathbf{s} \in \mathbb{S} \subseteq \mathbb{R}^2$ is its location (i.e., latitude and longitude). We assume that we have a sequence of N events up to time T, denoted by $\mathcal{D} = \{(t_n, \mathbf{s}_n)\}_{n=1}^N$.

Image Sequence. Additionally, we have an image dataset (e.g., satellite image, night light image, weather map). The image dataset is represented as a sequence of images, e.g., a collection of satellite images acquired at different times covering the area of interest $\mathbb{S}$. An image dataset example is presented on the left in Figure 3.2. Formally, we denote $I \in \mathbb{R}^{C\times H\times W}$ as the image, where H and W are image height and width, respectively; C is the number of channels. Each image is annotated with time τ when the observation was made. Each pixel of image $I[h, w]$ is georeferenced and corresponds to a fixed geospatial area (e.g., 500 m by 500 m). The corresponding latitude/longitude coordinates of the geospatial area for the (h, w)-th pixel are represented by $\mathbf{x}_{h,w}$, where $\mathbf{x}_{h,w}$ is the coordinates of the pixel center. For specific kinds of images (e.g., weather map), besides historical

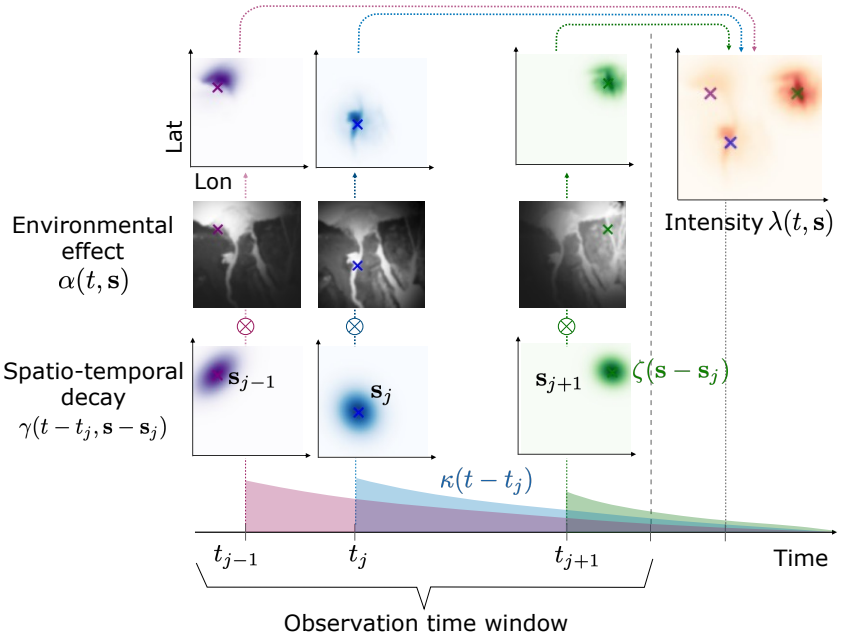


Figure 3.1: Illustration of the proposed method.

sequence, future sequence of the images (e.g., weather forecast maps) is available. Let $\mathcal{I} = \{(I_l, \tau_l)\}_{l=1}^{L}$ be the sequence of images over the time window $[0, T + \Delta T)$, where L is the number of observations.

Event Prediction Problem. Given the event sequence $\mathcal{D}$ in the observation time window $[0, T)$, and the image dataset $\mathcal{I}$ in the time period $[0, T + \Delta T]$, we aim to

- predict the number of events within any given spatial area and time period in $[T, T + \Delta T]$
- predict times and locations of events in the future time window $[T, T + \Delta T]$,

by leveraging $\mathcal{D}$ and $\mathcal{I}$.

3.5 Convolutional Hawkes processes

This section presents the proposed method for spatio-temporal event prediction, referred to as ConvHawkes (Convolutional Hawkes process). We provide the model formulation of ConvHawkes followed by parameter learning and prediction.

3.5.1 Model Overview

We propose a novel extension of Hawkes process for modeling triggering processes and predicting spatio-temporal events. The triggering processes are significantly influenced by

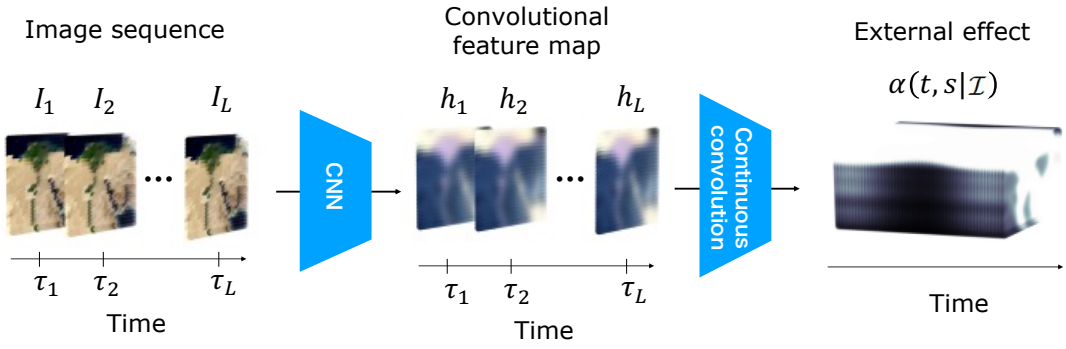


Figure 3.2: Overall architecture of the contextual effect module.

the contextual factors such as population, weather, road network and terrain.

The idea of this work is to leverage image data (e.g., satellite image and weather map) to capture such heterogeneity in the contextual factors and determine their effect on the triggering process. To this end, we incorporate the neural network model into the Hawkes process formulation. We illustrate our method in Figure 4.1. Specifically, we extend the neural network that learns the influence of the contextual factors by incorporating continuous kernel convolution, and parameterize the Hawkes process intensity based on the extended model. The proposed model learns latent contextual features from georeferenced images; and also learns contextual effects at each location, while at the same time providing tractable learning.

3.5.2 Model Formulation

We develop a flexible and tractable framework based on Hawkes process to learn the underlying contextual effects and spatio-temporal relationships between events from image data, e.g., satellite image, map image and weather map. Formally, ConvHawkes designs the conditional intensity as follows:

$$\lambda(t, \mathbf{s}|\mathcal{H}(t)) = \mu + \sum_{j:t_j<t} \underbrace{\alpha(t, \mathbf{s}|\mathcal{I})}_{\substack{\text{contextual} \\ \text{effect}}} \underbrace{\gamma(t - t_j, \mathbf{s} - \mathbf{s}_j)}_{\substack{\text{Spatio-temporal} \\ \text{decay}}}, \tag{3.5}$$

where μ is the background rate of event occurrence. As seen in Equation (4.3), our model consists of two components: *contextual effect* and *spatio-temporal decay*. The contextual effect $\alpha(\cdot)$ is specified by a neural network function, which captures the influence of the contextual factors. The spatio-temporal decay $\gamma(\cdot)$ is designed by a triggering kernel function over space and time that describes the decay in the influence of past events with

spatio-temporal distance. In the following, we describe the formulation of each component and the rationale behind them.

contextual effect. We model the contextual effect $\alpha(\cdot)$ based on a neural network model. The architecture of the contextual effect module is given in Figure 3.2.

For each image dataset, the image sequence is first processed by a convolutional neural network (CNN). The CNN is designed such that its output has the same size of the input image sequence, which makes it straightforward to utilize the time stamps, and location information of the images in the subsequent continuous convolution layer. We can use the encoder-decoder-based CNN [111, 110], CNN-RNN encoder-decoder [4], or other deep neural networks that are suitable for the given image data. In this paper, we choose a simple CNN with N_l layers. As shown in the experimental section (Section 3.6.6), our proposed method produces satisfactory prediction performance even with this simple neural architecture. Each image of the image sequence I_l is fed into the CNN architecture and transformed into the latent feature map $\mathbf{h}_l$, where $\mathbf{h}_l \in \mathbb{R}^{H \times W \times d}$. Here d is the dimension size of the latent feature map. For the sake of simplicity, we fix $d = 1$ in the experiments.

Next we apply continuous kernel convolution to these latent feature map to expand the learned latent feature map over discrete pixel space onto the continuous spatio-temporal space. Formally, given the latent feature map $\mathbf{h}_l$ and their associated time τ_l and latitude/-longitude coordinates for each pixel $\mathbf{x}^{h,w}$, the output of the convolutional layer at time t and location $\mathbf{s}$ is written by

$$\alpha\big((t, \mathbf{s})|\mathcal{I}\big) = \sum_{l} \sum_{h,w} \mathbf{h}_l[h, w] f(t - \tau_l, \mathbf{s} - \mathbf{x}^{h,w}), \tag{3.6}$$

where $f(\cdot)$ is a convolution kernel defined as continuous functions over the temporal and spatial plane. The definition for the continuous convolution kernel $f(\cdot)$ is provided later in this subsection. $\mathbf{h}_l[h, w] \in \mathbb{R}^d$ denotes the (h, w)-th pixel of latent feature map $\mathbf{h}_l$. $\alpha(\cdot)$ is a scalar function that quantifies the contextual effects at time t and location $\mathbf{s}$. Intuitively, the contextual feature map $\alpha\big((t, \mathbf{s})|\mathcal{I}\big)$ indicates how likely an event is to occur at time t and location $\mathbf{s}$ given preceding events that trigger it. This procedure is inspired by the work of [84, 101], which generalizes the discrete convolution used in standard CNNs to a continuous one. Our method is unique in that it does not require any discrete approximation. The above formulation enables the neural network model to be directly injected in the end-to-end framework of Hawkes process. At the same time, it yields tractable optimization (as discussed in Section 4.4.2).

Continuous convolution kernel. To ensure computation simplicity, we factorize the

continuous convolution kernel $f(\cdot)$ into temporal and spatial components such that:

$$f(t-\tau, \mathbf{s}-\mathbf{x}) = h(t-\tau)k(\mathbf{s}-\mathbf{x}), \tag{3.7}$$

where $h(\cdot)$ and $k(\cdot)$ are the kernel functions for temporal and spatial convolutions, respectively. In our case, we use the uniform kernel for the temporal convolution, which is defined by

$$h(t-\tau) = \mathbb{1}[\tau - \Delta < t < \tau + \Delta], \tag{3.8}$$

where $\mathbb{1}[\cdot]$ is an indicator function that indicates 1 when the condition holds, and 0 otherwise; Δ is the binwidth parameter. Without loss of generality, in our experiment, we fix Δ as the time interval between the observations. This is equivalent to piece-wise approximation. If we have no future observations or predictions of the images, the last image in the image sequence is used for prediction. For the spatial convolution, we can select a Gaussian kernel:

$$k(\mathbf{s}-\mathbf{x}) = \exp\big(-(\mathbf{s}-\mathbf{x})^\top \Sigma_k^{-1}(\mathbf{s}-\mathbf{x})\big), \tag{3.9}$$

where Σ_k is a 2×2 covariance matrix (bandwidth) of the kernel. We can use other convolution kernel functions, such as uniform and Rayleigh.

Spatio-temporal decay. Following previous work [19, 79], the spatio-temporal decay kernel functions are taken to be separable in space and time such that:

$$\gamma(t-t_j, \mathbf{s}-\mathbf{s}_j) = \kappa(t-t_j)\zeta(\mathbf{s}-\mathbf{s}_j). \tag{3.10}$$

Regarding the temporal decay function $\gamma(\cdot)$, the exponential decay function is the standard choice:

$$\kappa(t-t_j) = \exp\big(-\beta(t-t_j)\big), \tag{3.11}$$

where $\beta > 0$ is the decay factor. This implies that the occurrence of an event grows when events occur but their influence decreases exponentially at the rate of β over time.

A typical form of the spatial decay function is based on a Gaussian distribution as follows:

$$\zeta(\mathbf{s}-\mathbf{s}_j) = \exp\big(-(\mathbf{s}-\mathbf{s}_j)^\top \Sigma_\zeta^{-1}(\mathbf{s}-\mathbf{s}_j)\big), \tag{3.12}$$

where Σ_ζ is a 2×2 covariance matrix (bandwidth) of the kernel. Intuitively, when the j-th event occurs, the probability of the next event occurring is higher in the neighborhood of location $\mathbf{s}_j$. The bandwidth parameter Σ_ζ quantifies how strongly the influence from each past event decays over space. Other kernel functions, such as uniform and Rayleigh are viable alternatives.

3.5.3 Parameter Learning

Given a list of observed events up to time T (total of N events) $\mathcal{D}$ and the image dataset $\mathcal{I}$, the logarithm of the likelihood function is written as

$$\begin{aligned}\mathcal{L} = &\sum_{n=1}^{N} \log \left[\mu + \alpha(t_n, \mathbf{s}_n) \sum_{j:t_j<t_n} \gamma(t_n - t_j, \mathbf{s}_n - \mathbf{s}_j) \right] \\ &- \Bigg[\mu T|\mathbb{S}| + \sum_{n=1}^{N} \underbrace{\int_{t_n}^{T} \int_{\mathbb{S}} \alpha(t, \mathbf{s}) \gamma(t - t_n, \mathbf{s} - \mathbf{s}_n) dt d\mathbf{s}}_{\text{call this } \Lambda_n} \Bigg],\end{aligned} \tag{3.13}$$

where $|\mathbb{S}|$ denotes the area of spatial region $\mathbb{S}$. The computation difficulty comes from the integral of the neural network function (i.e., CNN) in the contextual effect $\alpha(\cdot)$ of term Λ_n. With our formulation, the neural network function $\mathbf{h}_l$ can be moved outside the integral, and Λ_n is rewritten as

$$\Lambda_n = \sum_l \sum_{h,w} \mathbf{h}_l[h, w] \int_{t_n}^{T} \kappa(t - t_n) h(t - \tau_l) dt \int_{\mathbb{S}} \zeta(\mathbf{s} - \mathbf{s}_n) k(\mathbf{s} - \mathbf{x}^{h,w}) d\mathbf{s}. \tag{3.14}$$

Consequently, we can obtain closed form solutions of the integral in term Λ_n for standard decay and convolution kernel functions. With the uniform kernel function (Equation 3.22), the integral over time can be performed analytically as follows:

$$\int_{t_n}^{T} \mathbb{1}[\tau - \Delta\tau < t < \tau] g(t - t_n) dt = \big[G(t - t_n) \big]_{\max(t_n, \tau - \Delta\tau)}^{\max(\tau, \max(t_n, \tau - \Delta\tau))}, \tag{3.15}$$

where $G(\cdot)$ is the derivative of the temporal decay kernel $g(\cdot)$. For the exponential decay defined in Equation (3.11), it can be written as

$$G(t - t_n) = -\exp\big(-\beta(t - t_n) \big). \tag{3.16}$$

For the pair of the Gaussian convolutional kernel (Equation 3.9) and Gaussian decay

function (Equation 3.12), the integral over space $\mathbb{S}$ is described as the sum of error functions:

$$\begin{aligned}
&\int_{\mathbb{S}} \zeta(\mathbf{s}-\mathbf{s}_j)k(\mathbf{s}-\mathbf{x}_{h,w})d\mathbf{s} \qquad (3.17)\\
&= \int_{\mathbb{S}} \exp\big(-(\mathbf{s}-\mathbf{x})^\top \Sigma_k^{-1}(\mathbf{s}-\mathbf{x})\big)\exp\big(-\frac{1}{2}(\mathbf{s}-\mathbf{s}_j)^\top \Sigma_\zeta^{-1}(\mathbf{s}-\mathbf{s}_j)\big)d\mathbf{s}\\
&= \frac{1}{\sqrt{\det(2\pi(\Sigma_k+\Sigma_\zeta))}}\exp\big[-\frac{1}{2}(\mathbf{s}_j-\mathbf{x})^\top(\Sigma_k+\Sigma_\zeta)^{-1}(\mathbf{s}_j-\mathbf{x})\big]\\
&\times \int_{\mathbb{S}} \exp\big(-\frac{1}{2}(\mathbf{s}-\mathbf{x}_c)^\top \Sigma_c^{-1}(\mathbf{s}-\mathbf{x}_c)\big)d\mathbf{s},
\end{aligned}$$

where

$$\mathbf{x}_c = \big(\Sigma_k^{-1}+\Sigma_\zeta^{-1}\big)^{-1}\big(\Sigma_k^{-1}\mathbf{s}_j+\Sigma_\zeta^{-1}\mathbf{x}\big) \qquad (3.18)$$

$$\Sigma_c = \big(\Sigma_k^{-1}+\Sigma_\zeta^{-1}\big)^{-1} \qquad (3.19)$$

The Gaussian integral in the above equation can be expressed in terms of the error function for the diagonal covariance matrices Σ_k and Σ_ζ. The integral in the likelihood (Equation 3.13) has analytic form for many other kernels including Rayleigh and power-law. In the case of the Gaussian kernel pair defined by Equation (3.9) and Equation (3.12), it is given by an error function. The resulting log-likelihood is fully tractable, permitting the use of gradient-based algorithms. We apply simple back-propagation for training ConvHawkes. During the training phase, we adopt mini-batch optimization.

3.5.4 Event Number Prediction

The point process model can be used to predict the expected number of events by integrating the estimated intensity over specific time period $W_T = [T_p, T_q]$ and the area of interest $W_S \subset \mathbb{S}$ such that

$$\begin{aligned}
N(W_T \times W_S) &= \int_{W_T}\int_{W_S} \lambda(t,\mathbf{s})dtd\mathbf{s} \qquad (3.20)\\
&= \sum_l \sum_{h,w} \mathbf{h}_l[h,w]\int_{T_p}^{T_q}\zeta(t-t_n)h(t-\tau_l)dt\int_{W_S}\zeta(\mathbf{s}-\mathbf{s}_n)k(\mathbf{s}-\mathbf{x}^{h,w})d\mathbf{s},
\end{aligned}$$

where $N(A)$ is the number of events that fall into subset A. As mentioned in Section 4.4.2, we can obtain closed form solutions of the above integral.

Moreover, the ConvHawkes model can simulate the occurrence time of the next event and its location by adopting the thinning algorithm [80].

Table 3.1: Statistics of Datasets used in this paper.

	Area	Time span	Source	# of events
Conflict	Africa	1 Mar, 2018 - 31 Mar, 2020	ACLED[3]	16,801
Protest	Middle East	1 Mar, 2018 - 31 Mar, 2020	ACLED[3]	34,243
Disease	Europe	1 Mar, 2020 - 31 Aug, 2020	EMPRES-i[2]	21,529

3.6 Experiments

We used real-world event datasets from different domains to evaluate the predictive performance of ConvHawkes.

3.6.1 Datasets

We used three real-world event datasets and five image datasets. All the datasets are publicly available. The statistics of these datasets are given in Table 4.3.

Event Data

We conducted experiments on three event datasets from different domains.

- **Conflict**: Conflict dataset, which is provided by ACLED project[1], consists of roughly 17,000 armed conflicts in Africa dated from April 1, 2018 to March 31, 2020. Every event is recorded in the form of time and location (latitude and longitude coordinates).

- **Protest**: Protest dataset, which was gathered by ACLED project[3], contains over 34,000 demonstration events in Middle East over a four year period from April 1, 2018 to March 31, 2020. Each record contains time and location of the protest.

- **Disease**: Disease dataset is a collection of reported incidents of animal disease outbreaks that occurred in Europe, provided by EMPRES-i[2], it contains 21,529 records, each of which shows time, latitude and longitude.

[1]Armed Conflict Location and Event Dataset (ACLED). `https://www.acleddata.com`. Accessed on December 10, 2021.

[2]EMPRES Global Animal Disease Information System (EMPRES-i). `http://empres-i.fao.org/eipws3g/`. Accessed on April 1, 2021.

Georeferenced Image

We incorporated five image datasets as the contextual features: **nightlight**, **landcover**, **weather**, **population** and **road**. These georeferenced images were all sourced from open GIS databases.

- The source of **nightlight** image is the Night time Lights of the World data processed and distributed by the NGDC[3], we used the $16,801 \times 43,201$ tiles that cover the entire world.

- For **landcover** image, the data source is the world map image file, at scale of $1 : 10\,\text{m}$, provided within the Natural Earth[4] package.

- The world map files for **weather** and **population** were taken from GeoNetwork website[5] with a spatial resolution of 5 arc minutes, namely,

  - weather: [clim.tif]
  - population: [popd.tif]
  - livestock: [lvstd.tif]
  - terrain: [slp.tif]

- For **road**, the shapefile of roads was downloaded from gROADS[6]. The shapefile was converted into a GeoTIFF file.

The input images were saved in GeoTIFF format. As preprocessing, we cropped GeoTIFF images for the three areas of interest (i.e., Africa, Middle East, Europe) and resized them to 120×114 pixels for Africa, 120×147 for Middle East, 120×127 for Europe. The examples of a **population** image is given in Figure 3.5(a) and Figure 3.6(a), and **landcover** image in Figure 3.7(a). In the experiment, we only used static images which not contain time information. Thus, the number of observations L is fixed to 1. Details of the data collection procedure are given below.

[3]Image and Data processing by the National Oceanic and Atmospheric Administration's (NOAA) National Geophysical Data Center (NGDC). `https://ngdc.noaa.gov/ngdc.html`. Accessed on April 1, 2021.

[4]Natural Earth. `https://www.naturalearthdata.com`. Accessed on April 1, 2021.

[5]Food and Agriculture Organization (FAO), GeoNetwork. `http://www.fao.org/geonetwork`. Accessed on April 1, 2021.

[6]NASA Socioeconomic Data and Applications Center (SEDAC), Global Roads Open Access Data Set, Version 1 (gROADSv1). `http://dx.doi.org/10.7927/H4VD6WCT`. Accessed on April 15, 2021.

3.6.2 Comparison Methods

We compared the proposed ConvHawkes against four widely used point process methods.

- HPP (Spatio-temporal homogeneous Poisson Process): The intensity is assumed to be constant over time and space:

$$\lambda(t, \mathbf{s}) = \lambda_0, \tag{3.21}$$

  where λ_0 denotes the constant intensity rate. This optimization can be solved in closed form.

- RMTPP (Recurrent Marked Temporal Point Process) [21]: RMTPP uses RNN to describe the intensity of the marked temporal point process. RMTPP is primarily intended to model event timing and categorical event feature (marker). To allow comparison, we partitioned the area of interest using a pre-defined rectangular grid; and mapped latitude and longitude values of event data into particular grids (hereafter referred to as *regions*). Then the latitude and longitude coordinates were replaced by a region index. The region indices are regarded as marks.

- Hawkes (Spatio-temporal Hawkes Process) [80]: Intensity is given by Eq. (3.3), which does not accept any additional features. We choose an exponential decay function, see Eq. (3.11), as the temporal decay function $h(\cdot)$, and Gaussian kernel shown as Eq. (3.12) for the spatial decay function $k(\cdot)$.

- DMPP (Deep Mixture Point Process) [70]: This method incorporates the contextual features represented in images and texts by combining Poisson process modeling and deep neural networks. We used the same image datasets used in ConvHawkes as the contextual features for DMPP.

3.6.3 Experimental Settings

For the experiments, we divided each dataset into training, validation and test sets in chronological order with the ratios of 80%, 10%, and 10%. The model parameters were trained using the ADAM optimizer [42] with a learning rate of 0.002. We tuned all the models using early stopping based on the log-likelihood performance on the validation set with a maximum of 200 epochs and a patience of 10 epochs. Batch size was set to 256 for all methods. The hyperparameters of each model were optimized via grid search. For the neural networks-based models (i.e., RMTPP, DMPP and ConvHawkes), we chose the number

of layers N_l from $\{1, 2, 3, 4, 5\}$, and the number of units per layer N_u from $\{1, 3, 5, 8\}$. For CNN-based methods (i.e., DMPP and ConvHawkes), we searched the filter size N_k in the CNN over $\{1, 3, 5\}$. The uniform kernel function was selected for the temporal and spatial convolution. We factorize the convolutional kernel function $f(\cdot)$ into temporal and spatial components, and model each component by the uniform kernel:

$$f(t - \tau, \mathbf{s} - \mathbf{x}) = \mathbb{1}[|t - \tau| < \Delta]\mathbb{1}[||\mathbf{s} - \mathbf{x}|| < w], \tag{3.22}$$

where $\mathbb{1}[\cdot]$ is an indicator function, and Δ and w are positive parameters that threshold the kernels to zeros. In our experiment, we fix Δ as the time interval between the observations; w is the pixel size of the image. This is equivalent to a piece-wise approximation. Here we consider the simplest case for the implementation simplicity; but note that our method can be easily generalized to other forms. The chosen hyperparameters are presented in Section 4.6.7. The pixel intensities of color channels were normalized to [0,1], and then used as input of our model.

3.6.4 Implementation Details

All code was implemented using Python 3.9 and Keras [16] with a TensorFlow backend [1]. We conducted all experiments on a machine with four 2.8GHz Intel Cores and 16GB memory.

3.6.5 Evaluation Metrics

Our experiments use the following two metrics in evaluating all models. For both metrics, lower values indicate better performance.

- **NLL** (Negative Log-Likelihood) is used to assess the likelihood of the occurrence of the events over the test period; it is calculated as

$$\sum_{n=N}^{N+N_t} \left[-\log \lambda(t_n, \mathbf{s}_n) + \int_{t_{i-1}}^{t_n} \int_{\mathbb{S}} \lambda(t, \mathbf{s}) dt d\mathbf{s} \right], \tag{3.23}$$

  where N_t is the number of events in the test period.

- **NMAE** (Normalized Mean Absolute Error) evaluates the discrepancies between the predicted number of events in small time intervals and pre-defined regions and the ground truth. We first split the test time period $[T, T + \Delta T]$ into S successive small time intervals. Also, we partitioned the area of interest $\mathbb{S}$ into R uniform grid

Table 3.2: Negative log-likelihood (NLL). Lower is better. The best performance is shown in bold. Our proposal, ConvHawkes, outperforms four existing methods.

	Conflict	Protest	Disease
HPP	-8.872	-9.130	-9.081
Hawkes	-10.156	-10.525	-10.806
DMPP	-9.531	-9.465	-9.902
Proposed	**-11.548**	**-11.583**	**-11.988**

Table 3.3: Normalized Mean Absolute Error (NMAE) with standard deviation (in the bracket). Lower is better. The best performance is shown in bold. Our proposal, ConvHawkes, outperforms four existing methods.

	Conflict	Protest	Disease
HPP	1.144 (0.055)	1.116 (0.096)	1.277 (0.230)
RMTPP	0.876 (0.094)	0.925 (0.159)	0.940 (0.300)
Hawkes	0.464 (0.023)	0.520 (0.064)	0.481 (0.087)
DMPP	0.685 (0.041)	0.867 (0.119)	0.865 (0.231)
Proposed	**0.344** (0.015)	**0.466** (0.043)	**0.423** (0.073)

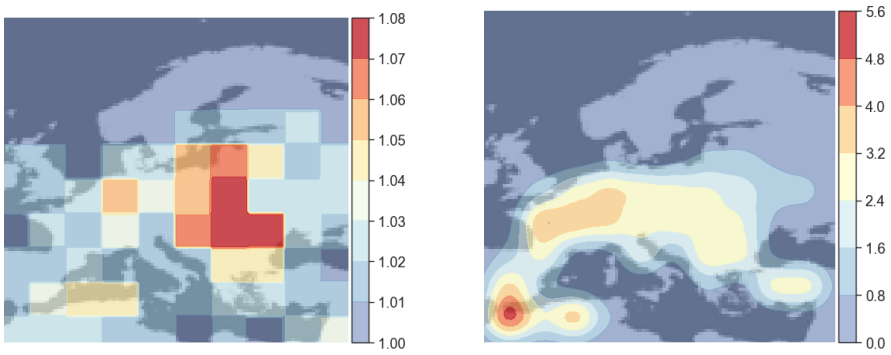
regions. For each time interval $(t_s, t_{s+1}]$ and each region $(\mathbf{s}_r, \mathbf{s}_{r+1})$, given the history of events up to t_s, we predicted the number of events in $(t_s, t_{s+1}]$ and $(\mathbf{s}_r, \mathbf{s}_{r+1}]$, $\hat{N}((t_s, t_{s+1}], (\mathbf{s}_r, \mathbf{s}_{r+1}])$, described in Eq. (3.20). Then, we measured the average normalized difference between the predicted and observed number of events over all the time intervals and the pre-defined regions as follows:

$$\text{NMAE} = \frac{\sum_{r=1}^{R}\sum_{s=1}^{S}\left|\hat{N}\big((t_s, t_{s+1}], (\mathbf{s}_r, \mathbf{s}_{r+1}]\big) - N\big((t_s, t_{s+1}], (\mathbf{s}_r, \mathbf{s}_{r+1}]\big)\right|}{\sum_{r=1}^{R}\sum_{s=1}^{S} N\big((t_s, t_{s+1}], (\mathbf{s}_r, \mathbf{s}_{r+1}]\big)}, \quad (3.24)$$

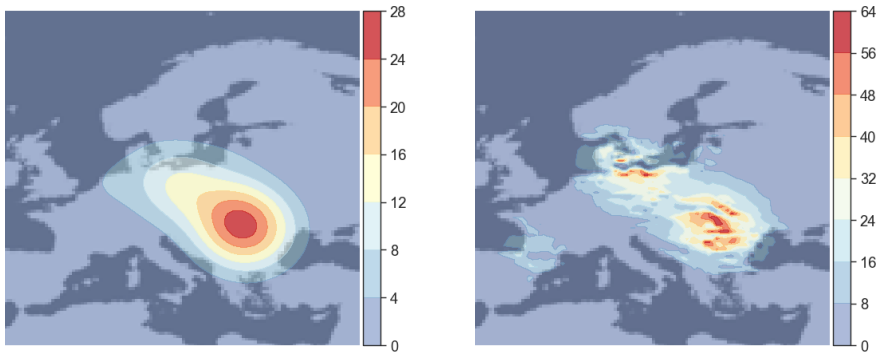
where $\hat{N}\left((t_{s+1}, t_s], (\mathbf{s}_r, \mathbf{s}_{r+1}]\right)$ is the predicted number of events in the small time interval $(t_{s+1}, t_s]$ and the grid region $(\mathbf{s}_r, \mathbf{s}_{r+1}]$ and $N(\cdot)$ is the ground truth at the s-th time interval and r-th region. In our experiment, we partitioned the spatial area of interest using a 5×5 uniform grid, and divided the test period into 20 time intervals. Therefore $S = 20$ and $R = 25$.

3.6.6 Performance Comparison

In this section, we compare ConvHawkes with existing point process methods for event prediction.



(a) RMTPP (b) DMPP



(c) Hawkes (d) ConvHawkes

Figure 3.3: Conditional intensity of diseases in Europe estimated by each method at March 1st 2020. The x-axis and y-axis represent longitude and latitude respectively.

Table 3.2 shows the negative log-likelihood (NLL) of the test data for the three event datasets. Note that since the temporal point processes (i.e., RMTPP) cannot calculate spatial likelihood, the NLL results of these methods are not reported on this table. We trained the proposed method with each of the five image datasets (i.e., nightlight, landcover, weather, population, road) and reported the best performance among the different image datasets in Table 3.2 and Table 3.3. The population image yields the best prediction performance for Conflict and Protest datasets; the landcover produces the best result for Disease dataset. We can see that the proposal, ConvHawkes, outperforms all existing methods examined across all the datasets. HPP delivers the worst prediction accuracy since it fails to account for temporal or spatial dependencies between events. DMPP performs worse than Hawkes on all the datasets. This is expected, because DMPP does not explicitly model the mutual excitation between events and thus cannot capture triggering patterns. For all the datasets, Hawkes outperformed the other existing methods. This is possibly because Hawkes models the mutual excitation between events with decay over spatio-temporal distances, while DMPP does not explicitly consider the spatial dependen-

Table 3.4: Performance comparison of the proposed method with different images on three event datasets. The number indicates NLL. Lower is better. The best performance is in boldface and second best is underlined.

	Conflict	Protest	Disease
nightlight	<u>-11.207</u>	**-11.379**	**-11.272**
landcover	-11.021	-10.814	-11.115
weather	-11.111	<u>-11.336</u>	<u>-11.149</u>
population	**-11.548**	-10.918	-10.959
road	-10.937	-11.050	-11.088

cies between events. ConvHawkes produces even better performance than Hawkes. The results suggest that our method can extract the meaningful features from the images, and effectively learn their impact on the triggering processes.

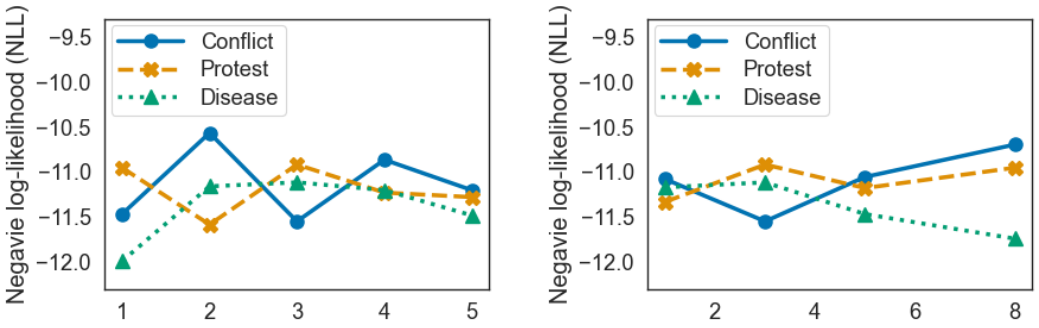
Table 3.3 reports the Normalized Mean Absolute Error (NMAE) of five different methods on the three event datasets. The result again demonstrates the effectiveness of our approach. Compared to the strongest baseline, ConvHawkes offers a NMAE improvement of 34.9% for the Conflict data ($p < 0.001$; paired t-test), 11.6% NMAE improvement for the Protest data ($p < 0.1$), 13.7% NMAE improvement for the Disease data ($p < 0.001$). This supports the above conclusion.

Our ConvHawkes demonstrated improvements in all evaluation metrics used. This is probably because ConvHawkes can capture the spatial heterogeneity of the triggering process as well as the spatio-temporal decay effects. We can see this in Figure 3.3, which depicts the conditional intensity of diseases learned by four different methods on March 1, 2020. In Figure 3.3(c), the spatial influences seem to be evenly distributed for Hawkes. ConvHawkes intensity (Figure 3.3(d)) is more unevenly distributed along the densely populated urban areas.

Sensitivity Analysis

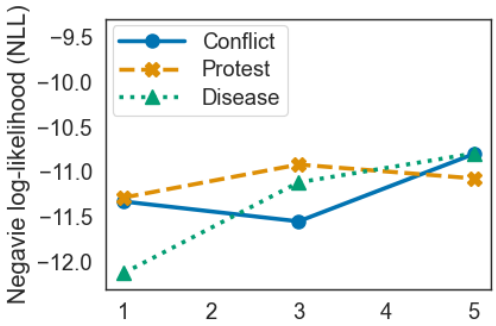
In this section, we analyze the impact of hyperparameters and experimental settings. We report the prediction performance of ConvHawkes under different settings for the three event datasets.

Impact of Different Images. Table 3.4 examines the importance of different images for event prediction by individually incorporating each of the image datasets into the proposed model. For Conflict data, NLL is improved when adding population image. This is consistent with the prior observation: unrest spreads among densely populated areas. We can see that incorporating nightlight images improves the prediction performance for



(a) Number of layers

(b) Number of units

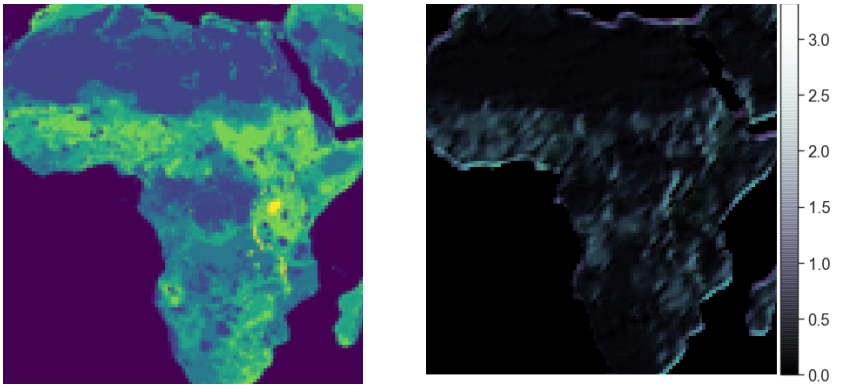


(c) Filter size

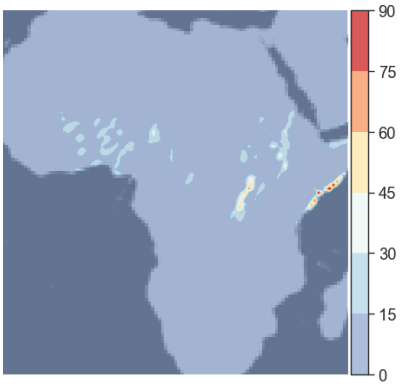
Figure 3.4: Impact of hyper-parameters on NLL performance.

Protest and Disease datasets. This is probably because nightlight is correlated to population density. We can observe that the weather image is important for Disease data. This finding matches the previous study: weather change affects on disease transmission [75]. In general, ConvHawkes can achieve stable performance across different image datasets. ConvHawkes with different image datasets is consistently better than all the comparison methods (Table 3.2), which ensures all the image datasets used in this paper are important for event prediction, and that ConvHawkes can effectively utilize these images.

Network Structure. We show the impact of network structures in Figure 3.4(a)-3.4(c). Except for the parameters being tested, all other parameters were held to default values. The NLL performance tends to be stable for all datasets. The prediction performance slightly improves when layer size N_l is 3 for Conflict data, 2 for Protest data, and 1 for Disease data. As shown in Figure 3.4(b), ConvHawkes performs robustly for different number of units, N_u, across all data sets. The prediction performance saturates as filter size N_k in the CNN increases. The proposed method yields similar results for the other metrics (i.e., NMAE). Throughout the experiment, we set $N_l = 3$, $N_u = 3$, $N_k = 3$ for



(a) Input population image (b) Learned feature map



(c) Intensity

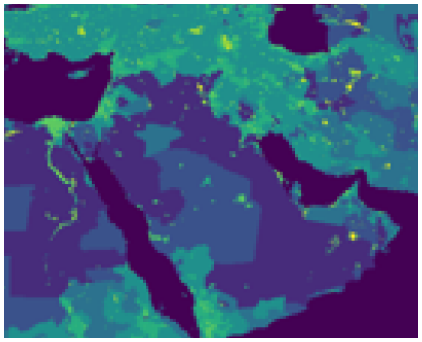
Figure 3.5: Learned feature map and intensity for Conflict dataset

Conflict dataset; $N_l = 2$, $N_u = 3$, $N_k = 3$ for Protest dataset; and $N_l = 1$, $N_u = 3$, $N_k = 3$ for Disease dataset.

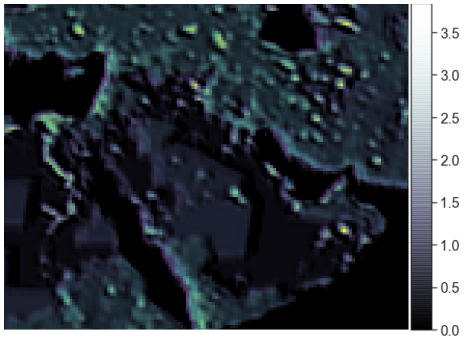
3.6.7 Analysis of Feature Learning

To further verify the above conclusion, we qualitatively explore the estimated intensity and the latent feature maps learned from the input image by our method.

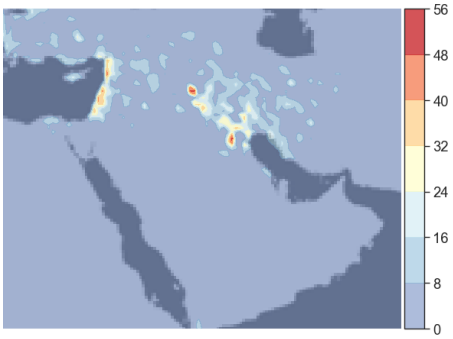
Figure 3.5-3.7 show the input image, the learned latent feature map and intensity for Conflict, Protest, Disease datasets. The x-axis and y-axis represent longitude and latitude respectively. Figure 3.5(a) and Figure 3.6(a) show the input population image for Africa and Middle East, respectively. Figure 3.7(a) is the input landcover image for Europe. In the learned feature maps (Figure 3.5(b), 3.6(b), 3.7(b)), the lighter shades are higher feature values and the darker shades indicate lower feature values. In Figure 3.5(b) and Figure 3.6(b), we can observe that ConvHawkes highlights coastal areas for Conflict and Protest datasets. This is expected, since the unrest events are strengthened



(a) Input population image



(b) Learned feature map

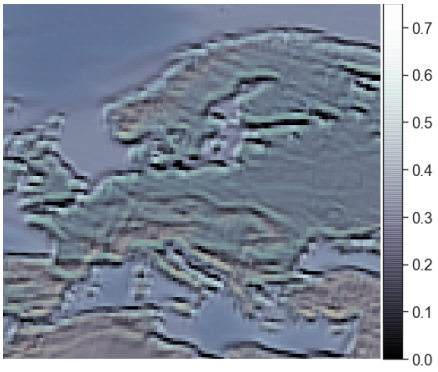


(c) Intensity

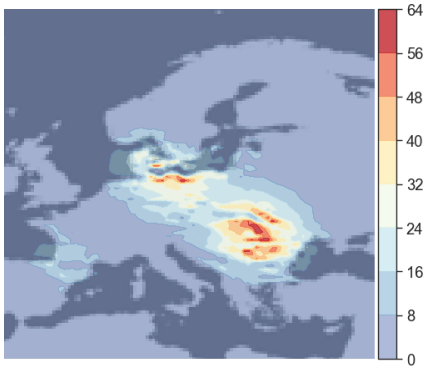
Figure 3.6: Learned feature map and intensity for Protest dataset

in densely populated coastal areas. ConvHawkes (Figure 3.5(c) and Figure 3.6(c)) exhibits heterogeneous intensity, in which the spatial influence is spread along the coastal areas. As shown in Figure 3.7, the landcover image serves as an important feature for Disease dataset. This may because landcover is associated with other characteristics including weather and population. The proposed method can automatically discover discriminative features from the images, providing insights about the effects the underlying contextual factors have on the triggering process.

(a) Input landcover image



(b) Learned feature map



(c) Intensity

Figure 3.7: Learned feature map and intensity for Disease dataset

3.7 Conclusion

In this thesis, we tackle the problem of spatio-temporal event prediction, with the aim of incorporating the spatio-temporal inhomogeneity of the triggering pattern driven by the contextual factors (e.g. population, weather, and road network). In this chapter, we consider the case where there are rich contextual information is available. In order to take into account rich contextual information, we develop a novel Hawkes process model based on a deep learning approach, referred to as ConvHawkes (Convolutional Hawkes Process). Specifically, we combine CNN with continuous kernel convolution and model the Hawkes process intensity parameter by using an extended neural network model. The key advantage of ConvHawkes over existing methods is that it can utilize the rich contexts present in image data, including satellite images, map images, and weather maps, and automatically discover their complex effects on the event triggering processes. At the same time, this formulation makes analytical integration over the intensity, which is required for Hawkes process estimation, tractable. Using three real-world datasets from different domains (i.e., armed conflicts, protests, diseases), we demonstrated that the proposed method is able to provide higher event prediction accuracy than existing methods. To the best of our knowledge, this work is the first attempt towards incorporating image data into self-exciting spatio-temporal point process models.

Chapter 4

Dynamic Hawkes Processes for Discovering Time-evolving Communities' States behind Diffusion Processes

4.1 Introduction

Throughout this thesis, we study the problem of modeling and predicting spatio-temporal events. A huge amount of data on spatio-temporal events is generated from diverse social and natural phenomena. In this chapter, we focus on social phenomena that exhibit self-exciting or triggering patterns, including user activities in social networks and information dissemination.

Many phenomena exhibit diffusion patterns among multiple communities. For example, infectious diseases like COVID-19 are transmitted from one county to another, leading to a worldwide pandemic [91]. Information such as opinions, news, and articles are shared and disseminated among online communities, e.g., user groups in social networks, news websites, and blogs. Such diffusion phenomena are recorded as multiple sequences of events, which indicate when and in which community the event occurred. Understanding the diffusion mechanism and predicting future events is crucial for many practical applications across domains. For example, policymakers would be able to design prompt and appropriate interventions to curb the spread of disease given a better understanding of the mechanisms behind the transmission and more reliable predictions.

Temporal point processes provide an elegant mathematical framework for modeling

event sequences. In these methods, the probability of event occurrences is determined by the *intensity* function. Hawkes process is an important class of point processes for modeling diffusion processes. These models use the *triggering kernel* to characterize diffusion processes and estimate their parameters via maximum likelihood. The triggering kernel encodes the magnitude and speed of influence from the past events, namely, how likely and quickly the past events in one community (i.e., "source" community) will affect the occurrence of a particular event in another community (i.e., "target" community). Hawkes process and its variants have been applied in diverse areas, from epidemic modeling [41] to social network analysis [119, 81, 26]. However, they have focused on learning the static influence of the past events on the current event, thereby largely overlooking the factor of time-evolution. In reality, the diffusion processes depend not only on the influences from the past but also on the current state of the target communities. For example, the outbreaks of infectious diseases in one community (e.g., country) can also be driven by people's awareness of the disease in each community (country) and their preventive behaviors which can constantly change over time, on top of the record of the disease occurrence. As another example, information diffusion heavily depends on ongoing peoples' interests in the target community (e.g., online user group). In particular, the spread of information to one target community (user group) is strengthened when a topic deemed to be important by the target community emerges in the online space; while it is weakened in accordance with a gradual loss in peoples' interest in the topic.

A few studies have considered the underlying dynamics of such "states" in communities. For instance, the SIR-Hawkes model [82] redesigned the triggering kernel of the Hawkes process by incorporating the recovered (immune) population dynamics over the course of the pandemic. Kobayashi *et al.* [43] proposed a time-dependent triggering kernel that varies periodically in time for modeling daily cycles of human activity. However, these approaches rely on hand-crafted functions for describing the latent dynamics of states and so demand expert domain knowledge. Moreover, they may not be flexible enough to accommodate the complexity and heterogeneity of the real world. In fact, in many practical applications, the complete set of factors is largely unknown and thus difficult to model through restricted parametric forms. Taking information diffusion as an example, the time-evolution of peoples' interest in a given topic is generally unknown and not directly observable.

A potential solution is to directly model the triggering kernel parameters using a flexible function of time (e.g., neural network). Alas, naively employing this approach makes parameter learning intractable since the log-likelihood of Hawkes processes involves the

integral of the triggering kernel. Computing the integral of the triggering kernel in combination with the neural network is generally infeasible.

In this chapter, we propose a novel Hawkes process model referred to as DHP (Dynamic Hawkes Process) which automatically learns the underlying dynamics of the communities' states behind the diffusion processes in a manner that allows tractable learning. We introduce the *latent dynamics function* for each community that represents its hidden dynamic states. Our core idea is to extend the triggering kernel by combining it with the latent dynamics function and its integral. Specifically, we model the magnitude of diffusion by the latent dynamics function and the speed of diffusion by the integral of the latent dynamics function. This design choice offers two benefits. First, the resulting triggering kernel can be expressed as a product of two components: composite function with the "inner" function being the integral of the latent dynamics function and the "outer" function being the basic triggering kernel; and the derivative of the inner function of that composite function (i.e., the latent dynamics function). Hence, by applying the substitution rule for definite integrals (i.e., the chain rule in reverse), we can obtain a closed-form solution for the integral of the triggering kernel involved in the log-likelihood. Second, it allows capture of the simultaneous changes of magnitude and speed of the diffusion as they are related through the latent dynamics function and its integral, which is desirable for many applications. For example, in the context of disease spread, active preventative measures can reduce both the magnitude and the speed of the infection. To model the integral of the latent dynamics function, we utilize and extend a monotonic neural network [89, 14]. This formulation enables DHP to learn flexible representations of the community state dynamics that underlie the diffusion processes. It should be noted that DHP can be easily extended to capture the time-evolving relationships between communities, by introducing the latent dynamics function for pairs of communities. In this work, we adopt DHP to demonstrate the hidden state dynamics of individual communities.

The main contributions of this chapter are:

- We propose a novel Hawkes process framework, DHP (Dynamic Hawkes Process) for modeling diffusion processes and predicting future events. The proposal, DHP, is able to learn the time-evolving dynamics of community states behind the diffusion processes.

- We introduce *latent dynamics function*; it reflects the hidden community dynamics and designs the triggering kernel of the Hawkes process intensity using the latent dynamics function and its integral. The resulting model is computationally tractable and flexible enough to approximate the true evolution of the community states un-

derlying the diffusion processes.

- We carry out extensive experiments using four real-world event datasets: Reddit, News, Protest, and Crime. The results show that DHP outperforms the existing works. Case studies demonstrate that DHP uncovers the hidden state dynamics of communities that underlie the diffusion processes by the latent dynamic function.

4.2 Related Work

With the evolution of data collection technology, extensive event sequences with precise timestamps are becoming available in an array of fields such as public health and safety [59, 103, 46], economics and finance [11, 5, 32], communications [26], reliability [6, 96, 90], and seismology [65, 69, 67]. Temporal point processes provide a principled theoretical framework for modeling such event sequences, in which the occurrences of events are determined by the intensity function.

Classical examples of temporal point processes include reinforced Poisson process [77], self-correcting point process [36], and Hawkes process [31]. The Reinforced Poisson process [77] considers the cumulative count of past events and a time-decreasing trend, and has been recently applied for predicting online popularity [87]. The intensity of the self-correcting point process [36] increases steadily and this trend is corrected by past observed events. Although these models have been widely used, they are not suitable for modeling diffusion processes between communities as they cannot explicitly model the influence of the past events underlying diffusion processes. Hawkes process [31] explicitly models the influence of the past events and captures triggering patterns between events (i.e., diffusion processes). Hawkes processes have been proven effective for modeling diffusion processes, including earthquakes and aftershocks [59], near-repeat patterns of crimes [59], financial transactions [23, 5, 32], online purchases [108, 22, 18, 102], and information cascades [119, 81, 26].

Recent studies employ neural network architectures to model point process intensity. In [21], the authors design the intensity using RNN. Omi *et al.* [74] extended our work by combining it with a monotonic neural network. Compared to classical point process methods, RNN-based models provide a more flexible way to handle the complex dependencies between events. However, the above methods focus on learning the triggering patterns of diffusion processes, i.e., influences from past events, and disregard the current (time-evolving) states of the communities. In the real world, event occurrences largely depend on the current community states (e.g., people's awareness of the disease, ongoing people's interest), which can evolve over time, as well as the past.

Several studies incorporate the time-variant dynamics of the community states behind diffusion processes into the Hawkes process formulation. For instance, the SIR-Hawkes model [82] considers recovered (immune) population dynamics to enhance the prediction of infectious disease events over the course of pandemic. Kobayashi *et al.* [43] proposed a time-dependent Hawkes process that accounts for the circadian and weekly cycles of human activity. Navaroli *et al.* [62] used nonparametric estimation to learn cyclic human activities underlying digital communications. All of the above methods, unfortunately, rely on a domain expert's knowledge to elucidate the dynamics of the communities states behind diffusion processes. Such dynamics are often quite complex and remain unexplored in many practical applications.

Different from the existing methods, our proposed method both incorporates the temporal dynamics of communities' states and the past influences.

4.3 Preliminaries

This section provides the general framework of point processes on which our work is built, and the formal definition of the event prediction problem studied in this paper.

4.3.1 Hawkes Processes

Point process is a random sequence of events occurring in continuous time $\{t_1, t_2, \cdots t_I\}$, with $t_i \in [0, T)$. Point processes are fully determined by "intensity" function $\lambda(t)$. Given the history of events $\mathcal{H}(t)$ up to time t, the intensity is defined as

$$\lambda(t) \equiv \lim_{\Delta t \to 0} \frac{\mathbb{E}[N(t + \Delta t) - N(t) | \mathcal{H}(t)]}{\Delta t}, \tag{4.1}$$

where $N(t)$ is a number of events falling in $[0, t)$, Δt is a small time interval, and $\mathbb{E}$ is an expectation. The intensity value $\lambda(t)$ at time t measures the probability that an event occurs in the infinitesimal time interval $[t, t + \Delta t)$ given past events $\mathcal{H}(t)$.

Hawkes process [31] is an important class of point processes, and can describe self-exciting phenomena. The intensity of Hawkes process is defined as

$$\lambda(t) = \mu + \sum_{j:t_j<t} g(t - t_j), \tag{4.2}$$

where $\mu \geq 0$ is a background rate and $g(\cdot) \geq 0$ is a *triggering kernel* encoding the augmenting or attenuating effect of past events on current events. Intuitively, each event at time t_j elevates the occurrence rate of events at time t by the amount $g(t - t_j)$ for $t > t_j$.

The univariate Hawkes process can be extended to multivariate Hawkes Process (MHP) to handle the mutual excitation of events (i.e., diffusion) among different communities (denoted by dimensions). Suppose we have I historical observations $\mathcal{D} = \{(t_i, m_i)\}_{i=1}^{I}$ with time $t_i \in [0, T)$ and community $m_i \in \{1, ..., M\}$. In our setting, the communities indicate countries, city districts, online user groups or news websites. For an M-dimensional multivariate Hawkes process, the intensity of the m-th dimension takes the following form:

$$\lambda_m(t) = \mu_m + \sum_{j:t_j<t} g_{m,m_j}(t - t_j), \tag{4.3}$$

where μ_m is the background rate of dimension m and $g_{m,m_j}(\cdot) \geq 0$ is the triggering kernel that captures the impact of an event in community m_j on the occurrence of an event in community m. The typical choice for the triggering kernel is the exponential memory kernel, which is defined by

$$g_{m,m_j}(\Delta_j) = \alpha_{m,m_j} \exp\left(-\beta_{m,m_j}\Delta_j\right), \tag{4.4}$$

where Δ_j represents the time interval $\Delta_j = t - t_j$, α_{m,m_j} quantifies the magnitude of the influence from community m_j on the event occurrence in community m, and β_{m,m_j} controls how quickly its effect decays in time (i.e., speed of the diffusion). Other candidates include power law kernel [67], Raleigh kernel [98], and log-normal distribution [62].

The negative log-likelihood function of a multivariate Hawkes process over time interval $[0, T]$ is given by:

$$\mathcal{L} = \sum_{i=1}^{I} \log \lambda_{m_i}(t_i) - \sum_{m=1}^{M} \int_0^T \lambda_m(t)dt. \tag{4.5}$$

4.3.2 Problem Definition

An event is represented by the pair (t, m), where t and m denote time and community (e.g., country, news website) where the event happened, respectively. An event sequence is defined as the set of events $\mathcal{D} = \{(t_i, m_i)\}_{i=1}^{I}$ with $t_i \in [0, T)$, where I denotes the number of events that have occurred up to time T.

Event Prediction Problem. Given the event sequence $\mathcal{D}$ in the observation time window $[0, T)$, we aim to leverage $\mathcal{D}$ to predict the number of events within any given time period; and the event times in the future time window $[T, T + \Delta T]$.

The key notations used in the paper are listed in Table 4.1.

Table 4.1: Table of symbols.

Symbol	Definition
$\mathcal{H}(t)$	event sequence up to t
$N(t)$	total number of events up to t
$N^m(t)$	number of events of dimension m up to t
$\lambda_m(t)$	intensity function for dimension m
μ_m	background rate for dimension m
$g_{m,m'}(t)$	triggering kernel between dimension m and dimension m'
$\alpha_{m,m'}$	interactions between dimension m and demension m'
$f_m(t)$	dynamic function for dimension m
$F_m(t)$	integral of dynamic function for dimension m
$\Phi_m^c(t)$	neural network function of dimension m for component c
M	number of dimensions
C	number of mixture components
L	number of layers of neural network
S	number of time intervals in test time period

4.4 Dynamic Hawkes Processes

In this section, we present DHP (Dynamic Hawkes Process), a novel multivariate Hawkes process framework for event prediction; it can learn the time-evolution of the communities underlying the diffusion processes. Figure 4.1 illustrates DHP. We design the triggering kernel of DHP intensity (panel A in Figure 4.1) as the product of two components: the triggering kernel with the input of time-rescaled events (panel B in Figure 4.1), which learns the decay influence from the past events; and the *latent dynamics function* (panel C) to adjust the magnitude of the influence from the past events. The latent dynamics function describes the time-evolving states of the communities (indicated by dimensions). In the context of disease spread, the latent dynamics function represents the dynamics of people's awareness of the disease in each country. For information diffusion, it characterizes the temporal evolution of readers' interests in news websites. We elaborate on the formulation of DHP in Section 4.4.1, followed by parameter learning (Section 4.4.2) and prediction procedure (Section 4.5).

4.4.1 Model Formulation

The proposed model specifies the intensity of Hawkes process for dimension m as

$$\lambda_m(t) = \mu_m + \sum_{j:t_j<t} g_{m,m_j}(\tilde{\Delta}_j) f_m(t), \tag{4.6}$$

where μ_m is the background rate for the m-th dimension (i.e., community), $g_{m,m_j}(\cdot)$ is any chosen triggering kernel between dimension m and dimension m_j such as exponential memory kernel or log-normal distribution, and $\tilde{\Delta}_j$ is the time-rescaled or transformed time interval between the current time t and the time of j-th event t_j. $f_m(t) \geq 0$ represents the dynamics of the m-th community underlying the diffusion processes at time t, which controls the magnitude of diffusion. The transformed time interval $\tilde{\Delta}_j$ is defined by the integral of the latent dynamics function between t_j and t as follows:

$$\tilde{\Delta}_j = \int_{t_j}^{t} f_m(\tau) d\tau = F_m(t) - F_m(t_j), \tag{4.7}$$

where $F_m(t)$ denotes the integral function of the continuous-time dynamics $f_m(t)$, that is

$$F_m(t) = \int_0^t f_m(\tau) d\tau. \tag{4.8}$$

The above formulation can be understood by considering an analogy drawn from the time-rescaling theorem [9]. Intuitively, this transformation adjusts the influence of each event by stretching or shrinking time based on the value of the latent dynamics function $f_m(t)$. When $f_m(t) < 1$, the interval times are lengthened so that event times are further separated. Likewise, when $f_m(t) > 1$, the interval times are compressed so that events are drawn closer together. If $f_m(t) = 1$ for all $t > 0$,

$$\tilde{\Delta}_j = \int_{t_j}^{t} 1 d\tau = t - t_j, \tag{4.9}$$

and Equations 4.6 and 4.7 reduce to a simple multivariate Hawkes process (Equation 4.3). This formulation assumes that the speed of diffusion varies according to the temporal dynamics of the target community m, which is captured by $f_m(t)$. This assumption is realistic; for instance, disease spread is controlled by people's awareness of the disease in each country and their preventive behaviors. Information diffusion is largely influenced by the reader's interest in each news website. It is worth mentioning that the latent dynamic function can be easily extended to consider the dynamics of pairwise interactions between dimensions, by redefining the latent dynamics function as $f_{m,m_j}(t)$. The following discussion holds even under this extension.

Our formulation allows considering the latent state of each dimension m at current time t as well as the influence from the past events. Also, it can capture the simultaneous changes in diffusion magnitude and speed, which is desirable for many applications (as discussed in the following paragraph). Most importantly, it enables us to compute the analytic integral

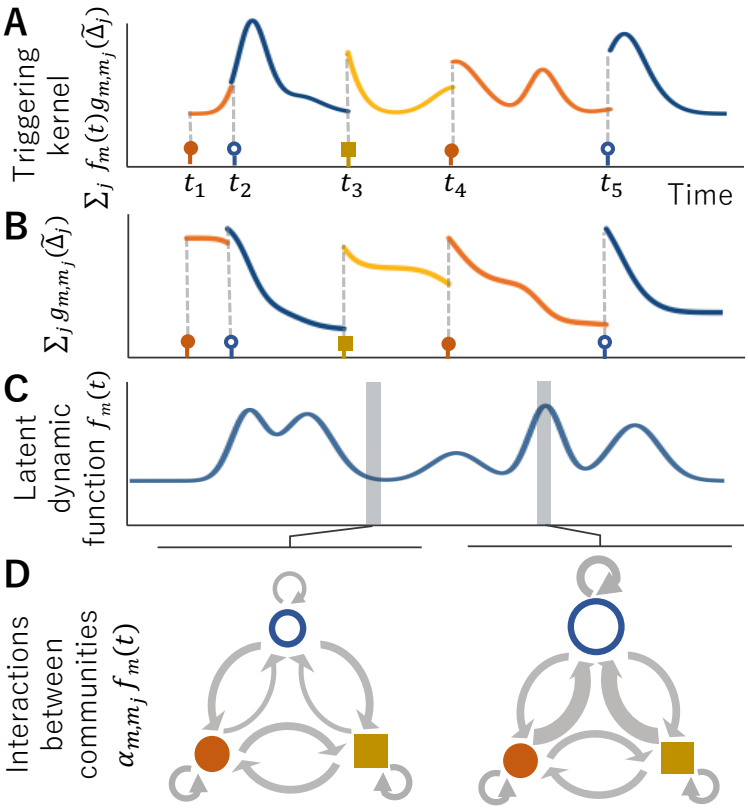


Figure 4.1: An illustration of DHP. Panel A represents individual events (i.e., input event sequence) and the dynamic interaction term for dimension ∘. The different dimensions (i.e., communities) are shown in different colors and markers. Panels B and C show the modified triggering kernel $\sum_j g_{m,m_j}(\tilde{\Delta}_j)$ and the latent dynamic function $f_m(t)$ for dimension ∘, respectively. Panel D depicts the interactions between communities at two different times. The size of each node represents the level of activity; the width of each arrow represents the level of interaction at each time.

of the intensity, which is required for evaluating the log-likelihood (further discussion can be found in Section 4.4.2) and predicting the number of future events (See Section 4.5).

Triggering Kernel. The triggering kernel can have many forms. For example, we can assume the exponential memory kernel for $g_{m,m_j}(\cdot)$, i.e.,

$$\lambda_m(t) = \mu_m + \sum_{j:t_j<t} f_m(t)\alpha_{m,m_j} \exp\left(-\beta_{m,m_j}\left(F_m(t) - F_m(t_j)\right)\right), \tag{4.10}$$

where α_{m,m_j} encompasses the magnitude of the static interaction between the m-th and m_j-th dimension; and β_{m,m_j} weights the decay of the influence over time. Notice that, the above formulation relies on the implicit assumption that the magnitude and speed of diffusion are related through the latent dynamics function $f_m(t)$, which controls the

magnitude of diffusion; its integral $F_m(t)$ governs the speed of diffusion. For example, when $f_m(t) = 2$ for every t, the second term of Equation 4.10 is $\underline{2\alpha}\exp(-\underline{2\beta}\Delta_j)$, where $\Delta_j = t - t_j$. When $f_m(t) = 0.5$ for every t, it is $\underline{0.5\alpha}\exp(-\underline{0.5\beta}\Delta_j)$. This assumption is reasonable since the magnitude and speed of diffusion vary simultaneously in many cases. Taking disease transmission as an example, active prevention measures can both reduce the magnitude and the speed of the infection. The influence on the magnitude of diffusion from the latent dynamics function is tuned by α, and the influence on the speed of diffusion from its integral is tuned by β.

Latent dynamics function. The design of $f_m(\cdot)$ is flexible to so any non-negative function can be used. Inspired by [74], we utilize and extend a monotonic neural network [89, 14] that learns a strictly monotonic function to design the latent dynamics function. Concretely, we model the integral function $F_m(t)$ using the monotonic neural network. This guarantees that its derivative (i.e., the latent dynamics function $f_m(t)$) is strictly non-negative, so intensity $\lambda_m(t)$ results in a non-negative function. In describing the integral function we propose to further enhance the expressiveness of the monotonic neural network by using a mixture of monotonic neural networks. Formally,

$$F_m(t) = \sum_{c=1}^{C} \pi_c \Phi_m^c(t) + b_0 t, \tag{4.11}$$

where C is the number of mixture components, $\Phi_m^c(\cdot)$ is the c-th monotonic neural network, π_c is the mixture weight of the c-th component, and b_0 is a bias parameter for the output layer. To preserve monotonicity of the integral of the latent dynamics function $F_m(t)$, we impose non-negative constraints on the mixture weights $\{\pi_1, ..., \pi_C\}$ and parameter b_0. For each dimension, we construct L fully connected neural layers with monotonic activation functions. Whenever the context is clear, we simplify notation $\Phi_m^c(\cdot)$ to $\Phi(\cdot)$. At each layer $l \in \{1, 2, ..., L\}$ of the monotonic neural network, the hidden-state vector $\mathbf{h}^{(l)}$ is given by

$$\mathbf{h}^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}), \tag{4.12}$$

where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are parameter matrix and vector to learn for l-th layer, respectively. $\sigma(\cdot)$ is a monotonic non-linear function. The input of the first layer is time t: $\mathbf{h}^{(0)} = t$. Following the previous work [89, 14], we use *tanh* activation for hidden layers and *softplus* for the last layer. The output of the monotonic neural network is

$$\Phi(t) = \mathbf{B}\mathbf{h}^{(L)}, \tag{4.13}$$

where $\mathbf{B}$ is a learnable weight matrix. The weight parameter matrices $\mathbf{W}^{(l)}$ and $\mathbf{B}$ are imposed to be non-negative. The latent dynamics function $f_m(t)$, which is the derivative of the monotonic neural network $F_m(t)$, takes the following form,

$$f_m(t) = \sum_c \pi_c \phi_m^c(t) + b_0, \tag{4.14}$$

where $\phi_m^c(t)$ is the gradient of the monotonic neural network $\Phi_m^c(t)$ with respect to time t, namely

$$\phi_m^c(t) = \frac{d\Phi_m^c(t)}{dt}. \tag{4.15}$$

The gradient $\phi_m^c(t)$ can be obtained by applying the automatic differentiation implemented in deep learning frameworks such as TensorFlow[1]. As we place no restriction on the parametric forms of the community dynamics underlying the diffusion processes, our model can fit various complex dynamics of each community's state.

This design choice enables us to automatically learn unknown complex dynamics of the communities' states behind the diffusion processes, while at the same time allowing us to compute the exact log-likelihood for training as described in Section 4.4.2.

4.4.2 Parameter Learning

Given the history of events up to but not including T, $\mathcal{D} = \{(t_i, m_i)\}_{i=1}^I$, we learn all the parameters of DHP by minimizing the negative log-likelihood of the observed event sequences. Specifically, we simultaneously estimate the neural network weights and the kernel parameters $\{\boldsymbol{\mu}, \mathbf{A}, \boldsymbol{\beta}\}$: the background rates $\boldsymbol{\mu} = \{\mu_1, .., \mu_M\}$, the matrix of interactions $\mathbf{A} = (\alpha_{i,j}) \in \mathbb{R}^{M \times M}$, and the decay rates $\boldsymbol{\beta} = \{\beta_1, .., \beta_M\}$. The negative log-likelihood function of DHP over time interval of $[0, T)$ is obtained by substituting Equation 4.6 into Equation 4.5:

$$\begin{aligned} \mathcal{L} = \sum_{i=1}^{I} \Bigg[& \log \Big(\mu_{m_i} + \sum_{j:t_j<t_i} g_{m_i,m_j} \big(F_{m_i}(t_i) - F_{m_i}(t_j) \big) f_{m_i}(t_i) \Big) \\ & - \sum_{m=1}^{M} \Big(\mu_m (t_i - t_{i-1}) + \underbrace{\int_{t_{i-1}}^{t_i} \sum_{j:t_j<t} g_{m,m_j} \big(F_m(t) - F_m(t_j) \big) f_m(t) dt}_{\Lambda_i} \Big) \Bigg]. \end{aligned} \tag{4.16}$$

Table 4.2: Integral for some common kernels.

Type	Equation $g(t)$	Integral $G(t)$
Exponential (EXP)	$\alpha \exp(-\beta t)$	$-\frac{\alpha}{\beta} \exp(-\beta t)$
Power-law (PWL)	$\frac{\alpha\beta}{(\alpha+\beta t)^{(p+1)}}$	$-\frac{\alpha}{p(\alpha+\beta t)^p}$
Raleigh (RAY)	$\alpha t \exp(-\beta t^2)$	$-\frac{\alpha}{2\beta} \exp(-\beta t^2)$

The problem here is to obtain integral Λ_i in the last term, which reduces to

$$\Lambda_i = \sum_{j:t_j<t_{i-1}} \int_{t_{i-1}}^{t_i} g_{m,m_j}\Big(\underbrace{F_m(t) - F_m(t_j)}_{\text{inner part}}\Big) f_m(t)dt. \tag{4.17}$$

Notice that the integrand of the above integral can be regarded as the product of composite function $g_{m,m_j}(F_m(\cdot))$ and the derivative $f_m(t)$ of the "inner" part of that composite function. Hence, we can solve the integral of Equation 4.17 in closed form by applying u substitution (also called "the reverse chain rule") technique. Making the substitution $x = F_m(t) - F_m(t_j)$ gives $dx = f_m(t)dt$. Changing variables from t to x, the integral of Equation 4.17 becomes,

$$\begin{aligned}
&\int_{t_{i-1}}^{t_i} g_{m,m_j}\Big(F_m(t) - F_m(t_j)\Big) f_m(t)dt \\
&= \int_{F(t_{i-1})-F(t_j)}^{F(t_i)-F(t_j)} g_{m,m_j}(x)dx = \big[G_{m,m_j}(x)\big]_{F(t_{i-1})-F(t_j)}^{F(t_i)-F(t_j)} \\
&= G_{m,m_j}\left(F(t_i) - F(t_j)\right) - G_{m,m_j}\left(F(t_{i-1}) - F(t_j)\right)
\end{aligned} \tag{4.18}$$

where $G_{m,m_j}(t)$ denotes the integral of $g_{m,m_j}(t)$. This can be computed analytically for many common kernels. In our experiment, we used three types of triggering kernel: Exponential, Power-law and Raleigh. Table 4.2 presents their equations and integrals, where α and β are parameters of the triggering kernel. p is the scaling exponent of the power-law ($p > 1$) and we fix $p = 2$ in the experiment.

Given the exact log-likelihood, we back-propagate the gradients of the loss function $\mathcal{L}$. In the experiment, we employ mini-batch optimization.

Table 4.3: Statistics of Datasets used in this paper.

	Source	Time span	# Events	Communities
Reddit	Reddit[1]	1 Mar - 31 Aug, 2020	23,059	25 subreddits
News	GDELT[2]	20 Jan - 24 Mar, 2020	19,541	40 news websites
Protest	ACLED[3]	1 Mar - 21 Nov, 2020	22,313	35 countries
Crime	Chicago Data Portal[4]	1 Mar - 19 Dec, 2020	29,318	13 community areas

4.5 Preidiction

For each time interval $(t_s, t_{s+1}]$ and each dimension m, given the history of events up to time t_s, we calculate the expected number of events in $(t_s, t_{s+1}]$ by

$$\int_{t_s}^{t_{s+1}} \lambda_m(\tau)\tau. \tag{4.19}$$

As discussed in Section 4.4.2, this integral takes analytic form for the proposed method. Similarly to Equation 4.18, we obtain

$$\begin{aligned} \hat{N}^m((t_s, t_{s+1}]) &= \int_{t_s}^{t_{s+1}} \lambda_m(\tau)\tau = \mu_m(t_{s+1} - t_s) \\ &+ \sum_{j:t_j<t_s} G_{m,m_j}\left(F(t_{s+1}) - F(t_j)\right) - G_{m,m_j}\left(F(t_s) - F(t_j)\right), \end{aligned} \tag{4.20}$$

where $\hat{N}^m\left((t_{s+1}, t_s]\right)$ is the predicted number of events in the given time interval $(t_{s+1}, t_s]$ for dimension m.

4.6 Experiments

We start by setting up the qualitative and quantitative experiments, and then report their results.

4.6.1 Datasets

We used four real-world event datasets from different domains.

- **Reddit**: We crawled the official Reddit API[1] to gather timestamped hyperlinks between Reddit communities (i.e., subreddits) over 6 months from March 1 to August 31, 2020. The data collection procedure followed the one used in [44]. During crawling

[1]Reddit API. http://www.reddit.com/dev/api. Accessed on December 10, 2020.

we selected the 25 most popular subreddits, and retrieved hyperlinks among those subreddits: we identified and recorded posts in one source subreddit that contain links to different target subreddits. This process finally yielded a total of roughly 23,000 posts, each of which had submission time, source subreddit, and target subreddit. We treated a list of hyperlinks to each target subreddit as a separate sequence and considered target subreddits as communities (i.e., dimensions). The source subreddit was not used for training but for qualitative evaluation (Figure 4.3). Following the work of [63], we use a list of hyperlinks to each target subreddit as a separate sequence and consider target subreddits as communities (i.e., dimensions).

- **News**: News dataset, which is provided by GDELT project [48] through its API[2], consists of roughly 20,000 news articles related to COVID-19 dated from January 20 to March 24, 2020. The original dataset contains over a million of news articles related to COVID-19. Each piece of news had a timestamp and a URL. We extracted the domain of news websites from a URL and obtained more than 1,000 unique domains. We filtered out 40 country-specific domains and used them as communities. The granularity of time is one second.

- **Protest**: Protest dataset, which was gathered by ACLED[3], contains over 20,000 demonstration events in 35 countries during 9 months from March 1 to November 21, 2020. We sampled 35 popular countries and retrieved events from those countries. Each event was associated with two attributes: timestamp and country. The dataset was recorded at minute level.

- **Crime**: Crime dataset is publicly available from the City of Chicago Data Portal[4]; it includes about 30,000 reported crimes from 13 community areas of Chicago from 1 March to 19 December 2020. Each event recorded the time and community area where a crime happened. The time granularity is one minute.

All the datasets are publicly available. The statistics of these datasets are given in Table 4.3.

[2]Global Dataset of Events, Location, and Tone (GDELT). http://gdeltproject.org. Accessed on December 23, 2020.

[3]Armed Conflict Location and Event Dataset (ACLED). https://www.acleddata.com. Accessed on December 10, 2020.

[4]Chicago Data Portal. https://data.cityofchicago.org/. Accessed on December 30, 2020.

4.6.2 Comparison Methods

We compare DHP against five widely used point process methods that incorporate the influence of the past events:

- HPP (Homogeneous Poisson Process): It is the simplest point process where the intensity is assumed to be constant over time. Its intensity is defined by $\lambda_m(t) = \lambda_m$, where λ_m denotes the constant intensity rate for m-th community.

- RPP (Reinforced Poisson Processes) [87, 77]: RPP accounts for the aging effect and the cumulative count of past events. For each dimension m, the intensity of RPP is characterized by

$$\lambda_m(t) = \gamma_m(t) N^m(t), \tag{4.21}$$

  where $\gamma_m(t)$ is the relaxation function that characterizes the aging effect, and $N^m(t)$ is the number of events of dimension m that have occurred up to t. Following the prior work [87], we define $\gamma_m(t)$ by the following relaxation log-normal function:

$$\gamma_m(t) = \frac{\exp\left(-(\log t - \alpha_m)^2 / 2\beta_m^2\right)}{\sqrt{2\pi}\beta_m t}, \tag{4.22}$$

  where α_m and β_m are parameters, which are local to the dimension.

- SelfCorrecting (Self-correcting Point Process) [36]: Its intensity is assumed to increase linearly over time and this tendency is corrected by the historical events. The intensity function of SelfCorrecting is assumed to increase steadily over time with the rate $\beta_m > 0$; this trend is corrected by constant $\rho_m > 0$ every time an event arrives. Its intensity function associated with dimension m is given by

$$\lambda_m(t) = \exp\left(\alpha_m + \beta_m (t - \rho_m N^m(t))\right), \tag{4.23}$$

  where α_m, β_m, and ρ_m are parameters, and $N^m(t)$ is the number of events of dimension m in $(0, t]$.

- Hawkes (Hawkes Process): Its intensity is parameterized by Equation 4.3, which explicitly models the influence of the past events by using the static triggering kernel.

- RMTPP (Recurrent Marked Temporal Point Process) [21]: It employs RNN to encode the non-linear effects of past events. The event sequences are first embedded by RNN and then used as the input of the intensity.

4.6.3 Experimental Settings

For the experiments, we divided each dataset into train, validation, and test sets by chronological order with the ratios of 70%, 10%, and 20%. The model parameters were trained using the ADAM optimizer [42]. We tuned all the models using early stopping based on the log-likelihood performance on the validation set with a maximum of 100 epochs for the Reddit and News datasets and 30 epochs for the Protest and Crime datasets. Batch size is set to 128. The hyperparameters of each model are optimized via grid search. For the neural networks-based models (i.e., RMTPP and DHP), we choose the number of layers from $\{1, 2, 3, 4, 5\}$. For Hawkes process methods (i.e., Hawkes and DHP), the kernel function is selected from three commonly used kernels: exponential memory, power-law, and Raleigh kernels. These are mathematically defined in Table 4.2. For DHP, we search on the number of mixtures C over $\{1, 2, 3, 4, 5\}$. The chosen hyperparameters are presented in Section 4.6.5.

4.6.4 Evaluation Metrics

Our experiments use the following two metrics in evaluating all models. For both metrics, lower values indicate better performance.

- **NLL** (Negative Log-Likelihood) is used to assess the likelihood of the occurrence of the events over the test period; it is calculated as

$$\sum_{i=I}^{I+n} \left[-\log \lambda_{m_i}(t_i) + \sum_{m=1}^{M} \int_{t_{i-1}}^{t_i} \lambda_m(t) dt \right], \tag{4.24}$$

  where n is the number of events in the test period.

- **MAPE** (Mean Absolute Percentage Error) evaluates the discrepancies between the predicted number of events in small time intervals and the ground truth. We first split the test time period $[T, T+\Delta T]$ into S successive small time intervals using 15-minute periods. For each time interval $(t_s, t_{s+1}]$ and each dimension m, given the history of events up to t_s, we predict the number of events in $(t_s, t_{s+1}]$, $\hat{N}^m((t_s, t_{s+1}])$, described in Equation 4.20 of Section 4.5. Then, we measure the average normalized difference between the predicted and observed number of events across all time intervals as follows:

$$\text{MAPE} = \frac{1}{M} \sum_{m=1}^{M} \frac{\left| \sum_{s=1}^{S} \hat{N}^m\big((t_s, t_{s+1}]\big) - \sum_{s=1}^{S} N^m\big((t_s, t_{s+1}]\big) \right|}{\sum_{s=1}^{S} N^m\big((t_s, t_{s+1}]\big)}, \tag{4.25}$$

where $\hat{N}^m((t_{s+1}, t_s])$ is the predicted number of events in the small time interval $(t_{s+1}, t_s]$ and $N^m(\cdot)$ is the ground truth at the s-th time interval and m-th dimension.

4.6.5 Implementation Details

All code was implemented using Python 3.9 and Keras [16] with a TensorFlow backend [1]. We conducted all experiments on a machine with four 2.8GHz Intel Cores and 16GB memory.

The model parameters were trained using the ADAM optimizer [42] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a learning rate of 0.002. For the neural networks-based models (i.e., RMTPP and DHP), the number of hidden units in each layer is fixed as 8. In our experiment, the number of mixtures is set to 3 for Reddit, News and Protest datasets, and to 5 for Crime dataset. In all experiments, we used the power-law kernel. The number of layers is set to 2 for Reddit and Protest datasets, 1 for News dataset, 3 for Crime dataset, respectively.

Table 4.4: Negative log-likelihood (NLL) and Mean Absolute Percentage Error (MAPE) with standard deviation (in the bracket). Lower is better. The best performance is in bold. Our proposal, DHP, outperforms five existing methods.

	Reddit		News		Protest		Crime	
	NLL	MAPE	NLL	MAPE	NLL	MAPE	NLL	MAPE
HPP	-5.637	0.553 (0.204)	-5.710	0.600 (0.044)	-5.753	0.345 (0.060)	-6.795	0.144 (0.014)
Hawkes	-5.696	0.458 (0.107)	-6.167	0.471 (0.085)	-6.260	0.415 (0.371)	-6.799	0.179 (0.016)
RPP	-5.568	0.595 (0.259)	-6.150	0.481 (0.522)	-5.643	0.581 (0.759)	-6.781	0.175 (0.021)
SelfCorrecting	-5.662	0.475 (0.158)	-5.973	0.452 (0.059)	-5.750	0.524 (0.674)	-6.803	0.123 (0.005)
RMTPP	-	0.311 (0.061)	-	0.446 (0.125)	-	0.639 (1.337)	-	0.302 (0.010)
Proposed	**-6.447**	**0.305** (0.045)	**-6.301**	**0.442** (0.039)	**-6.914**	**0.318** (0.049)	**-6.983**	**0.117** (0.008)

4.6.6 Performance Evaluation

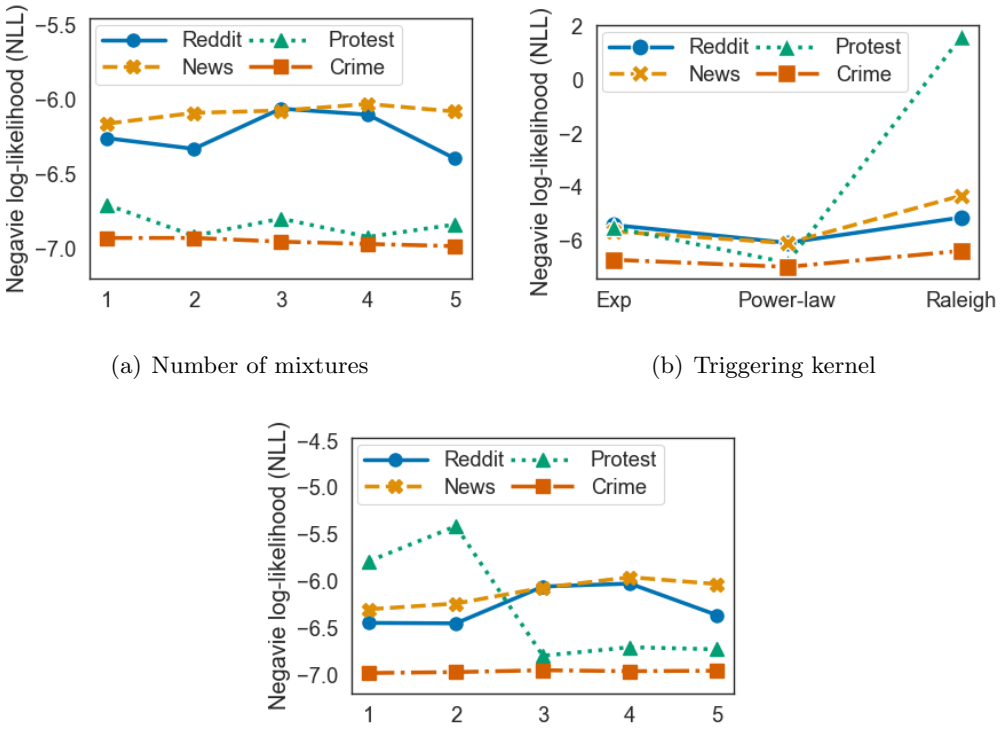
In this section, we first compare DHP with existing methods on event prediction. Table 4.4 presents the negative log-likelihood (NLL) of the test data and Mean Absolute Percentage Error (MAPE) for different methods on the real-world event datasets. In this table, we omit the result of RMTPP since its log-likelihood function differs from those used in the other methods, (it is defined for the whole event sequence from all communities, not for the separate sequences of the individual communities, which precludes fair comparison). As shown in the table, our proposal, DHP, outperforms the four existing methods across all the datasets in terms of NLL. HPP has the worst NLL in most cases since it does not explore the temporal variation of the event occurrences. RPP and SelfCorrecting cannot achieve good results as they encode strong assumptions on the functional forms of the intensity, which limits the expressivity of the model. Hawkes surpasses HPP, RPP, and SelfCorrecting, which explicitly models the dependencies between past and current events. However, it still falls short for modeling the dynamic changes of the community states in the diffusion process. Our DHP achieves even better NLL than Hawkes. This verifies that incorporating latent community dynamics is essential for event prediction and that DHP can learn effective representations of the time-evolving dynamics of community states.

DHP achieves the best MAPE for all datasets. RMTPP performs the second best in terms of MAPE for Reddit and News datasets, which is probably because RMTPP exploits the power of RNN for learning non-linear dependencies between events. But RMTPP performs poorly for Protest and Crime datasets since it cannot capture changes in the event occurrences due to the temporal evolution of communities' states, e.g., a large reduction in protest events due to the COVID-19. DHP outperforms all other methods across the datasets on the two metrics. The above result reveals the effectiveness of encoding the community state dynamics governing the diffusion process for event prediction. It also suggests that the assumption of DHP, i.e., the magnitude and speed of diffusion are related, holds for real diffusion processes.

4.6.7 Sensitivity Study

In this section, we analyze the impacts of hyperparameters or experimental settings. We report the prediction performance of DHP under different settings for the four datasets.

Number of mixtures. We examine how the number of mixture components, C, determines the prediction performance of DHP. Figure 4.2(a) shows the negative log-likelihood (NLL) on the test data with respect to different numbers of mixtures $\{1, 2, 3, 4, 5\}$. In this experiment, we fixed the number of layers as 3 and used the power-law kernel. The



(a) Number of mixtures

(b) Triggering kernel

(c) Number of layers

Figure 4.2: Sensitivity Study: NLL performance of DynamicHawkes on different settings for four datasets.

NLL performance tends to be stable for all the datasets. It slightly increases as the number of mixture components becomes larger for Protest and Crime dataset. The results indicate that increasing the number of mixtures can improve the expressiveness of the model.

Kernel functions. We investigate the effect of three kernel functions: exponential kernel, power-law kernel, and Raleigh kernel, where the number of mixtures and the number of layers are set to 3. As shown in Figure 4.2(b), the power-law kernel yields the best performance on all datasets.

Number of layers. Figure 4.2(c) evaluates the sensitivity of our neural network $\Phi_m^c(t)$ to the number of layers $L \in \{1, 2, 3, 4, 5\}$ by fixing the number of mixtures as 3 and using the power-law kernel. We observe that DHP yields better NLL results for the Protest dataset with larger numbers of layers. For the other three datasets, it has little effect on the performance.

In general, DHP shows stable and robust prediction performance across different settings.

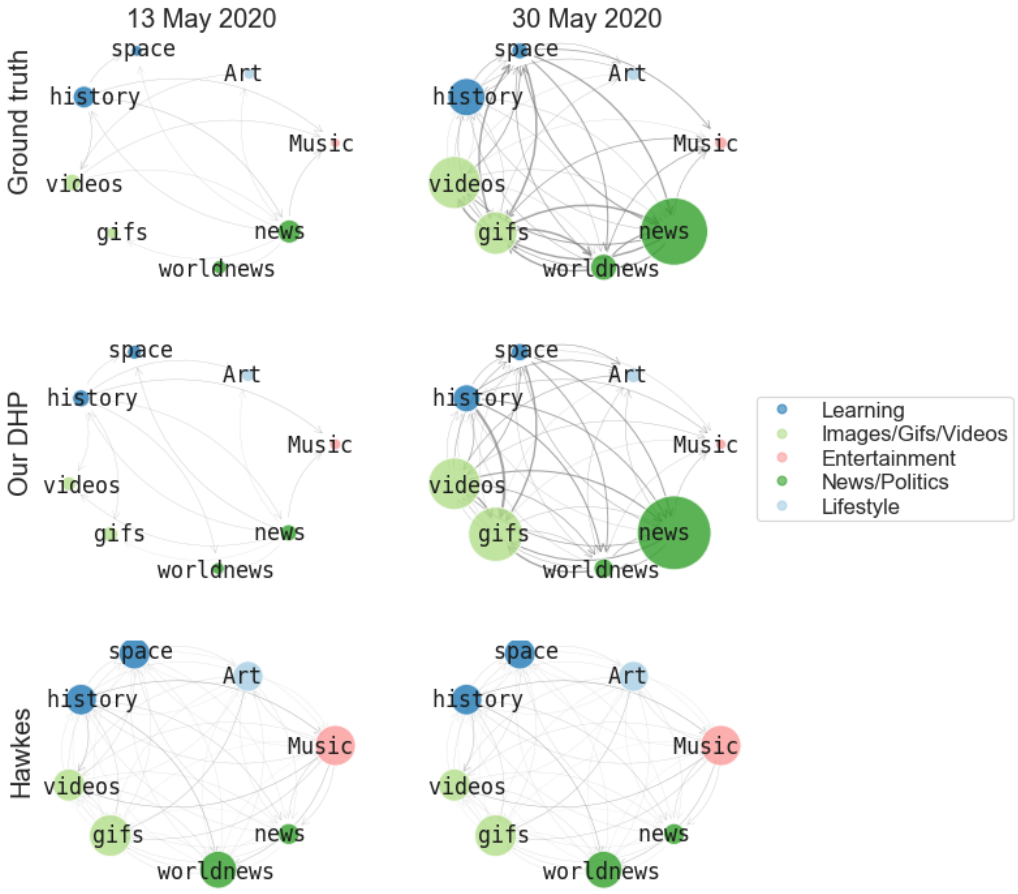


Figure 4.3: Visual comparison of predicted interactions among reddit communities (i.e., subreddits) from Reddit dataset. The bottom two rows are the predicted results of our DHP and Hawkes. The last row is the ground truth. Columns correspond to times indicated by the label on the top. Nodes represent subreddits. Their colors indicate their categories. For DHP (middle), the size of the m-th node is proportional to $\sum_{m'} \alpha_{m,m'} f_m(t)$ and the width of edge between nodes m and m' is proportional to $\alpha_{m,m'} f_m(t)$.

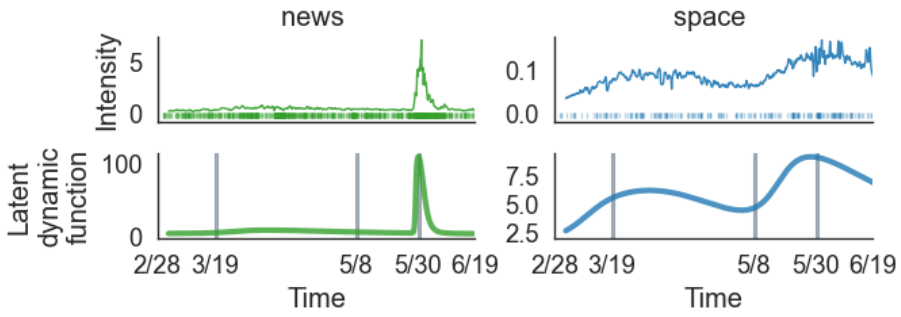


Figure 4.4: Intensity with observed event sequences and latent dynamics function for two Reddit communities (i.e., subreddits): `news` and `space`. The latent dynamics function increases rapidly for `news` and slowly for `space` following the onset of the COVID-19 lockdown.

Figure 4.5: Learned triggering kernel between 8 selected subreddits at 3 different time points. Nodes denote subreddits, color indicates category. Node size is proportional to latent dynamics function for each subreddit. Edge width is proportional to triggering kernel, which indicates strength of diffusion between pairs of subreddits. We can see the latent dynamics function increases over time for most subreddits from March to May, 2020.

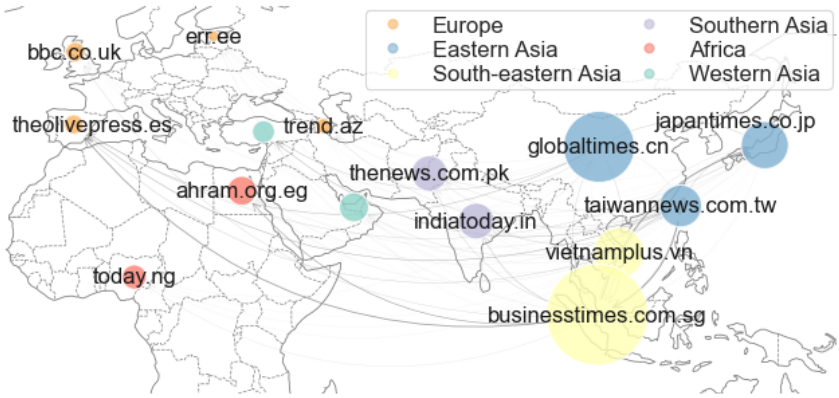


Figure 4.6: February 24, 2020.

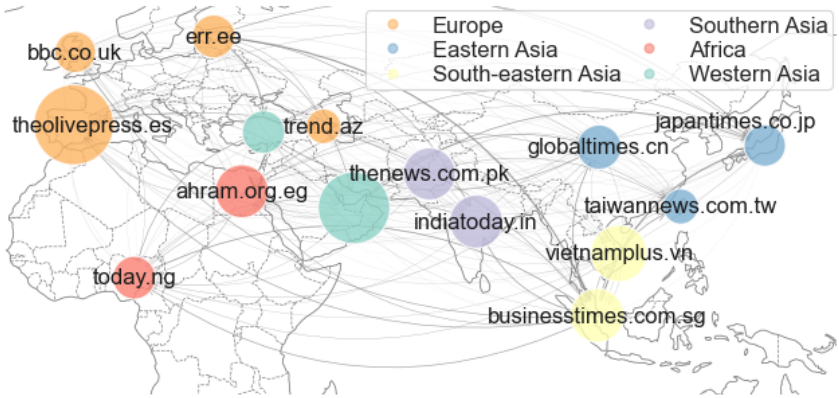


Figure 4.7: March 15, 2020.

Figure 4.8: Inferred interactions among 15 major news websites from different countries by DHP on News dataset at two different time points. Nodes refer to domain names of news websites. We used the top-level domain to specify the country in which each news website is based. Notes are colored by regions. The size of the m-th node is given by $\sum_{m'} \alpha_{m,m'} f_m(t)$ and the edge width between nodes m and m' by $\alpha_{m,m'} f_m(t)$.

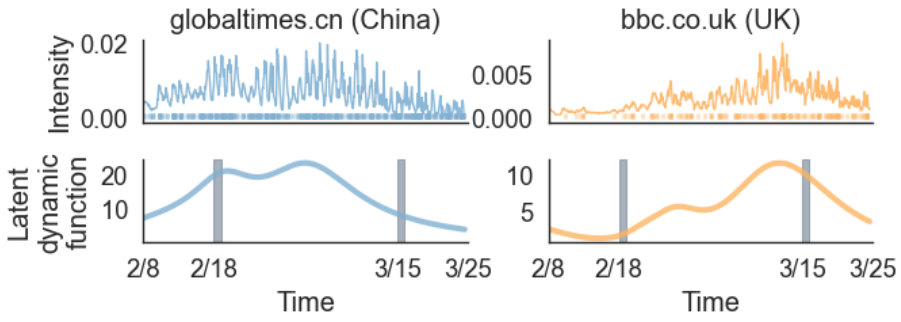


Figure 4.9: Learned intensity and latent dynamic function for two news websites in China and UK from February 14 to March 25, 2020.

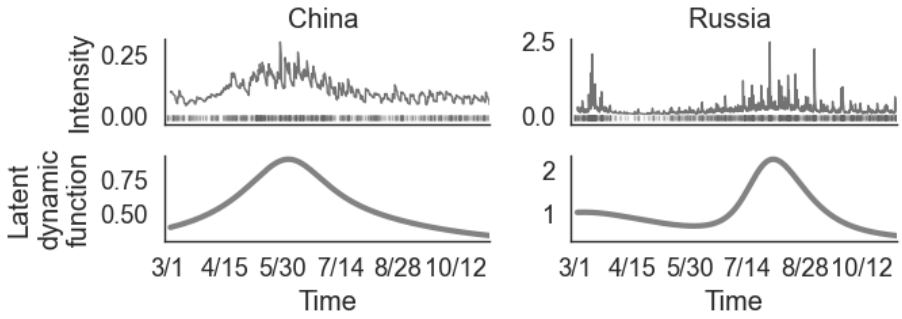


Figure 4.10: Intensity and latent dynamic function learned by DHP on Protest dataset for two countries.

4.6.8 Case Studies

In order to further verify the capability of DHP, we analyze the temporal dynamics of the community states behind the diffusion process learned by DHP from each dataset.

Figure 4.3 visualizes the interactions between selected 8 Reddit communities (i.e., subreddits) learned from Reddit dataset. In each row, we compare the estimated interactions between subreddits by DHP (middle) and Hawkes (bottom), and the ground truth (top). For the ground truth, node size corresponds to the aggregated number of hyperlinks for each "target" community in the default 5-day interval; the weight of each edge represents the number of hyperlinks between source community m' and target community m. For DHP, node size is proportional to $\sum_{m'} \alpha_{m,m'} f_m(t)$; edge width is $\alpha_{m,m'} f_m(t)$. For Hawkes, node size is $\sum_{m'} \alpha_{m,m'}$; edge width is $\alpha_{m,m'}$. Note that Hawkes produces the same results across times since it assumes the triggering kernel is static over time. We can see that the interactions learned by DHP are more consistent with the true evolution of the interactions between online user communities compared to Hawkes.

In Figure 4.4 and 4.5, we present the the dynamics of community states learned by our DHP for Reddit dataset. Figure 4.4 shows the intensity $\lambda_m(t)$ and estimated dynamics function $f_m(t)$ learned for Reddit dataset, along with the observed event sequences for the two subreddits. Figure 4.5 shows the learned triggering kernel between eight selected subreddits at three different time points. In this figure, the nodes denote subreddits colored by category; its size is proportional to latent dynamics function for each subreddit. The edge width is proportional to the triggering kernel, which indicates the strength of diffusion between pairs of subreddits. The latent dynamics function increases up to the end of May, rapidly for `news` and slowly for `space`. This is probably due to the COVID-19 lockdown. These results demonstrate that our DHP learns a reasonable representation of the latent temporal dynamics of the online communities.

Figure 4.8 shows inferred interactions among news websites from 15 countries learned for News dataset. In these figures, the node size denotes the value of the latent dynamics function $\sum_{m'} \alpha_{m,m'} f_m(t)$ for each news website; the edge width denotes the strength of interactions between them $\alpha_{m,m_j} f_m(t)$. East Asian and South-East Asian countries (denoted by blue and yellow) rise to their peaks around late February (See Figure 4.6) and then decrease until mid-March (Figure 4.7), while the other countries are peaked around or after March 15 (Figure 4.7), not in February (Figure 4.6). We can also see in Figure 4.9 that the dynamic function peaks around mid-February for China (left), followed by the United Kingdom with the peak of mid-March (right). These trends are synchronized to the growth of the pandemic in each country. East Asian and South-East Asian countries

experienced their first peak in COVID-19 cases ahead of the other countries, which would trigger the people's early interest on COVID-19 related topics and accelerate the spread of COVID-19 related news early on. This confirms that our proposal, DHP, well reproduces the complex evolution in news website activities.

Figure 4.10 shows the intensity and latent dynamic function learned from the Protest dataset. According to a previous study[5], in contrast to the online events, the pandemic initially leads to a reduction in protest events and the trend was corrected after several weeks. DHP well characterizes this trend. In China (left), the dynamic function decreased following the onset of the coronavirus around the beginning of March and returned to a moderate level by mid-June. For Russia, it declined gradually from March until the beginning of July, where the first peak of the pandemic occured around May 11. In conclusion, DHP uncovers the latent community dynamics underlying the diffusion processes, and so provides meaningful insights about the diffusion mechanism.

4.7 Conclusion and Future Work

Modeling and predicting diffusion processes are important tasks in many applications. In this chapter, we presented a novel Hawkes process framework, DHP (Dynamic Hawkes Process), that can learn the temporal dynamics of the community states underlying diffusion processes. The proposed DHP allows for the automatic discovery of the community state dynamics underlying the diffusion processes as well as offering tractable learning. By conducting extensive experiments on four real event datasets, we demonstrate that DHP provides better performance for modeling and predicting diffusion processes than several existing methods.

For future work, we plan to explore the following two directions. First, DHP can be extended to capture the pairwise dynamics of the interactions among communities by introducing the latent dynamics function for pairs of communities. We will extend DHP to this case and conduct experiments to evaluate the performance of the extended DHP in capturing the time-evolving dynamics of the pairwise interactions between communities. Secondly, DHP is built on the assumption that the magnitude and speed of the diffusion are related to each other, which may limit the flexibility of the model. We will explore how to modify DHP to ease this assumption.

[5] https://acleddata.com/2020/09/03/demonstrations-political-violence-in-america-new-data-for-summer-2020/

Chapter 5

Conclusion

5.1 Summary

In this thesis, we address the problem of predicting spatio-temporal events to capture the influence of contextual factors on the spatio-temporal event occurrences. We present novel point process models by integrating spatio-temporal point processes with deep neural networks. In chapters 2 and 3, we propose two new point process models integrating rich observable features and learning their complex effects on the event occurrence. In chapter 2, we study how to learn the effect of rich contextural factors on the event occurrence. To this end, we propose an inhomogeneous Poisson process model that can effectively exploit unstructured data that represents contextual information. In chapter 3, we investigate how to learn the influence of rich contextural factors on the triggering processes. To do so, we develop a Hawkes process model that effectively utilizes the rich external information present in unstructured data. In chapter 4, we aim to learn the effects of underlying and unobservable factors on the triggering processes. chapter 4 presents a Hawkes process model that learns the underlying dynamics of community states behind the diffusion processes and predicts the occurrences of events based on the dynamics. In this section, we briefly summarize the main contributions of this thesis.

5.1.1 Spatio-temporal Event Prediction with Rich Contextual Information (Chapter 2)

We propose an inhomogeneous point process model, referred to as Deep Mixture Point Processes (DMPP), for spatio-temporal event prediction. It accurately and effectively predicts spatio-temporal events by leveraging the contextual features, such as map images and social/traffic event descriptions, that impact event occurrence. We integrate the deep

learning approach into the point process framework. Specifically, we extract the intensity by using a deep mixture of experts, whose mixture weights are modeled by a deep neural network. This formulation allows us to utilize the information present in unstructured contextual features, and to automatically discover their complex effects on event occurrence, while at the same time yielding tractable optimization. We develop an efficient estimation procedure for training and evaluating DMPP. We conduct extensive experiments on real-world data sets from three urban domains. With regard to event occurrence, the proposed method achieves better predictive performance than all existing methods on all data sets.

5.1.2 Context-aware Spatio-temporal Event Prediction via Convolutional Hawkes Processes (Chapter 3)

We propose a novel Hawkes process model, Convolutional Hawkes process (ConvHawkes) for modeling diffusion processes and predicting spatio-temporal events. It accurately and effectively predicts spatio-temporal events by leveraging the contextual features contained in georeferenced images (e.g., satellite images and map images), that impact triggering processes. We present an extension of the neural network model and integrate it into the Hawkes process framework. This formulation allows us to utilize the contextual features present in the unstructured image data, and to automatically discover their complex effects on the triggering process, while at the same time yielding tractable optimization. We conduct extensive experiments on real-world datasets from different domains. With regard to event occurrence, the proposed method achieves better predictive performance than several existing methods on all datasets.

5.1.3 Dynamic Hawkes Processes for Discovering Time-evolving Communities' States behind Diffusion Processes (Chapter 4)

We propose a novel Hawkes process framework, Dynamic Hawkes Process (DHP) for modeling diffusion processes and predicting future events. The proposal, DHP, is able to learn the time-evolving dynamics of community states behind the diffusion processes. We introduce *latent dynamics function*; it reflects the hidden community dynamics and designs the triggering kernel of the Hawkes process intensity using the latent dynamics function and its integral. The resulting model is computationally tractable and flexible enough to approximate the true evolution of the community states underlying the diffusion processes. We carry out extensive experiments using four real-world event datasets: Reddit, News, Protest, and Crime. The results show that DHP outperforms the existing works. Case studies demonstrate that DHP uncovers the hidden state dynamics of communities that

underlie the diffusion processes by the latent dynamic function.

5.2 Future Research

In this section, we discuss possible future directions for extending the proposed models.

Although our experiments throughout the thesis have proven the effectiveness of our proposed models, their predictive power still has room for improvement. To further enhance the prediction performance, we can consider several approaches. First, additional features can also be incorporated into the proposed models. Throughout this thesis, we focus on incorporating particular contextual features. In chapters 2 and 3, we leverage rich contextual features (e.g., satellite images and map images). In addition to these features, we can also consider simple features like time of day, weather, and transportation networks. Second, the studies in this thesis only use either observable or unobservable contextual features. In chapters 2 and 3, we exploit observable contextual features, directly, whereas chapter 4 relies on inference to learn the contribution of unobservable contextual factors. In practice, however, contextual features are partially observable, where only some contextual features are explicitly known while others are missing. We want to integrate the two different approaches in chapters 2 and 3 and chapter 4 to exploit partially observable features. Last but not least, our approach can be extended to more recent point process models. The proposed models are based on basic point process models: Inhomogeneous Poisson processes and Hawkes processes. Recent works employ state-of-the-art neural network architectures to capture the non-linear temporal correlation between events, including Transformer Hawkes process [126] and self-attentive Hawkes process [116]. Different from these methods, this thesis focuses on modeling the time-evolving states of target locations at a future time, instead of the past influence. The proposed models can be integrated into these recent point processes. This allows for capturing both complex influences from historical events and the time evolution of current states driven by external factors.

In this thesis, we proposed three new point process models and applied the proposed models to several applications to evaluate their effectiveness: transportation, public safety, crime, public health, social media, and natural disasters. For future work, we try to apply our models to other applications such as financial transactions, online purchases, and infrastructure failures. In financial transactions, social mood plays an important role in modeling the change in the transaction price. We can employ ConvHawkes (Convolutional Hawkes process) to extract indicators of social mood from large-scale online data such as surveys, media content. We can also use DHP (Dynamic Hawkes Process) to automatically learn hidden social moods from transaction data. ConvHawkes (Convolutional Hawkes

process) can be used to leverage item features (e.g., reviews, item description, and user' feedback) for online purchasing prediction.

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A. Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B. Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467, 2016.

[2] Yacine Aït-Sahalia, Julio Cacho-Diaz, and Roger JA Laeven. Modeling financial contagion using mutually exciting jump processes. *Journal of Financial Economics*, 117(3):585–606, 2015.

[3] M Aldrin, RB Huseby, and PA Jansen. Space–time modelling of the spread of pancreas disease (pd) within and between Norwegian marine salmonid farms. *Preventive Veterinary Medicine*, 121(1-2):132–141, 2015.

[4] Mohamed Attia, Mohammed Hossny, Saeid Nahavandi, and Hamed Asadi. Surgical tool segmentation using a hybrid deep CNN-RNN auto encoder-decoder. In *Proceedings of the 14th International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3373–3378. IEEE, 2017.

[5] Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(01):1550005, 2015.

[6] Laurence A Baxter, Masaaki Kijima, and Michael Tortorella. A point process model for the reliability of a maintained system subject to general repair. *Stochastic models*, 12(1):12–1, 1996.

[7] Charles Blundell, Jeff Beck, and Katherine A Heller. Modelling reciprocating relationships with Hawkes processes. In *Advances in Neural Information Processing Systems 26 (NeurIPS)*, pages 2600–2608, 2012.

[8] Patricia L. Brantingham and Paul J. Brantingham. Mobility, notoriety, and crime: A study in the crime patterns of urban nodal points. *Journal of Environmental Systems*, 11(1):89–99, 1981.

[9] Emery N Brown, Riccardo Barbieri, Valérie Ventura, Robert E Kass, and Loren M Frank. The time-rescaling theorem and its application to neural spike train data analysis. *Neural computation*, 14(2):325–346, 2002.

[10] Srinivasa Ravi Chandra and Haitham Al-Deek. Predictions of freeway traffic speeds and volumes using vector autoregressive models. *Journal of Intelligent Transportation Systems*, 13(2):53–72, 2009.

[11] Valerie Chavez-Demoulin*, Anthony C Davison, and Alexander J McNeil. Estimating value-at-risk: A point process approach. *Quantitative Finance*, 5(2):227–234, 2005.

[12] Longbiao Chen, Daqing Zhang, Leye Wang, Dingqi Yang, Xiaojuan Ma, Shijian Li, Zhaohui Wu, Gang Pan, Thi-Mai-Trang Nguyen, and Jérémie Jakubowicz. Dynamic cluster-based over-demand prediction in bike sharing systems. In *Proceedings of the 18th ACM International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp)*, pages 841–852. ACM, 2016.

[13] Ricky T. Q. Chen, Brandon Amos, and Maximilian Nickel. Neural spatio-temporal point processes. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[14] Pawel Chilinski and Ricardo Silva. Neural likelihoods via cumulative distribution functions. In *Conference on Uncertainty in Artificial Intelligence*, pages 420–429. PMLR, 2020.

[15] Edward Choi, Nan Du, Robert Chen, Le Song, and Jimeng Sun. Constructing disease network and temporal progression model via context-sensitive Hawkes process. In *Proceedings of the 15th IEEE International Conference on Data Mining (ICDM)*, pages 721–726. IEEE, 2015.

[16] Francois Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[17] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220, 1972.

[18] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. Deep coevolutionary network: Embedding user and item features for recommendation. *arXiv preprint arXiv:1609.03675*, 2016.

[19] Stéphane De La Rocque, Thomas Balenghien, Lénaïg Halos, Klaas Dietze, Filip Claes, G Ferrari, Vittorio Guberti, and Jan Slingenbergh. A review of trends in the distribution of vector-borne diseases: Is international trade contributing to their spread? *Rev Sci Tech*, 2011.

[20] Peter J Diggle, Paula Moraga, Barry Rowlingson, Benjamin M Taylor, et al. Spatial and spatio-temporal log-Gaussian Cox processes: Extending the geostatistical paradigm. *Statistical Science*, 28(4):542–563, 2013.

[21] 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 Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1555–1564. ACM, 2016.

[22] Nan Du, Yichen Wang, Niao He, Jimeng Sun, and Le Song. Time-sensitive recommendation from recurrent user activities. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*, pages 3492–3500, 2015.

[23] Paul Embrechts, Thomas Liniger, and Lu Lin. Multivariate Hawkes processes: An application to financial data. *Journal of Applied Probability*, 48(A):367–378, 2011.

[24] Seyda Ertekin, Cynthia Rudin, and Tyler H. McCormick. Reactive point processes: A new approach to predicting power failures in underground electrical systems. *Annals of Applied Statistics*, 9(1):122–144, 2015.

[25] Mehrdad Farajtabar, Nan Du, Manuel Gomez-Rodriguez, Isabel Valera, Hongyuan Zha, and Le Song. Shaping social activity by incentivizing users. In *Advances in Neural Information Processing Systems 28 (NeurIPS)*, pages 2474–2482, 2014.

[26] Mehrdad Farajtabar, Yichen Wang, Manuel Gomez-Rodriguez, Shuang Li, Hongyuan Zha, and Le Song. Coevolve: A joint point process model for information diffusion and network evolution. *The Journal of Machine Learning Research*, 18(1):1305–1353, 2017.

[27] Mehrdad Farajtabar, Yichen Wang, Manuel Gomez Rodriguez, Shuang Li, Hongyuan Zha, and Le Song. Coevolve: A joint point process model for information diffusion and network co-evolution. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*, pages 1954–1962, 2015.

[28] Song Gao, Yaoli Wang, Yong Gao, and Yu Liu. Understanding urban traffic-flow characteristics: A rethinking of betweenness centrality. *Environment and Planning B: Planning and Design*, 40(1):135–153, 2013.

[29] Roch Giorgi et al. A relative survival regression model using B-spline functions to model non-proportional hazards. *Statistics in Medicine*, 22(17):2767–2784, 2003.

[30] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 922–929, 2019.

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

[32] Alan G Hawkes. Hawkes processes and their applications to finance: A review. *Quantitative Finance*, 18(2):193–198, 2018.

[33] Junichiro Hayano et al. Increased non-gaussianity of heart rate variability predicts cardiac mortality after an acute myocardial infarction. *Frontiers in Physiology*, 2:65, 2011.

[34] Dave Higdon. Space and space-time modeling using process convolutions. In *Quantitative methods for current environmental issues*, pages 37–56. Springer, 2002.

[35] Minh X Hoang, Yu Zheng, and Ambuj K Singh. FCCF: Forecasting citywide crowd flows based on big data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, page 6. ACM, 2016.

[36] Valerie Isham and Mark Westcott. A self-correcting point process. *Stochastic processes and their applications*, 8(3):335–347, 1979.

[37] Tomoharu Iwata, Amar Shah, and Zoubin Ghahramani. Discovering latent influence in online social activities via shared cascade Poisson processes. In *Proceedings of the*

19th ACM SIGKDD International Conference on Knowledge Discovery and Data mining, pages 266–274. ACM, 2013.

[38] Hyeon-Woo Kang and Hang-Bong Kang. Prediction of crime occurrence from multi-modal data using deep learning. *PLoS ONE*, 12(4):e0176244, 2017.

[39] Amol Kapoor, Xue Ben, Luyang Liu, Bryan Perozzi, Matt Barnes, Martin Blais, and Shawn O'Banion. Examining COVID-19 forecasting using spatio-temporal graph neural networks. *CoRR*, abs/2007.03113, 2020.

[40] Minkyoung Kim, Raja Jurdak, and Dean Paini. Modeling reflexivity of social systems in disease spread. *CoRR*, abs/1711.06359, 2017.

[41] Minkyoung Kim, Dean Paini, and Raja Jurdak. Modeling stochastic processes in disease spread across a heterogeneous social system. *Proceedings of the National Academy of Sciences*, 116(2):401–406, 2019.

[42] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[43] Ryota Kobayashi and Renaud Lambiotte. Tideh: Time-dependent Hawkes process for predicting retweet dynamics. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

[44] Srijan Kumar, William L Hamilton, Jure Leskovec, and Dan Jurafsky. Community interaction and conflict on the web. In *Proceedings of the 27th International World Wide Web Conference (WWW)*, pages 933–943, 2018.

[45] David Lando. On Cox processes and credit risky securities. *Review of Derivatives research*, 2(2):99–120, 1998.

[46] Thomas A Lasko. Efficient inference of gaussian-process-modulated renewal processes with application to medical event data. In *Proceedings of the 30 th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 2014, page 469. NIH Public Access, 2014.

[47] Herbert K H Lee, Bruno Sanso, Weining Zhou, and David M Higdon. Inference for a proton accelerator using convolution models. *Journal of the American Statistical Association*, 103(482):604–613, 2008.

[48] Kalev Leetaru and Philip A Schrodt. Gdelt: Global data on events, location, and tone, 1979–2012. In *Proceedings of the 7th International Conference on Information Security and Assurance (ISA)*, volume 2, pages 1–49. Citeseer, 2013.

[49] Ricardo T Lemos and Bruno Sansó. A spatio-temporal model for mean, anomaly, and trend fields of north atlantic sea surface temperature. *Journal of the American Statistical Association*, 104(485):5–18, 2009.

[50] PA W Lewis and Gerald S Shedler. Simulation of nonhomogeneous Poisson processes by thinning. *Naval Research Logistics Quarterly*, 26(3):403–413, 1979.

[51] Liangda Li, Hongbo Deng, Anlei Dong, Yi Chang, and Hongyuan Zha. Identifying and labeling search tasks via query-based Hawkes processes. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 731–740. ACM, 2014.

[52] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[53] Kym Louie, Marina Masaki, and Mark Allenby. A point process model for simulating gang-on-gang violence. *Project Report*, 2010.

[54] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 6, page 2, 2017.

[55] Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng Wang. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54:187–197, 2015.

[56] Hongyuan Mei and Jason M Eisner. The neural Hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 6754–6764, 2017.

[57] Sebastian Meyer et al. Self-exciting point processes: Infections and implementations. *Statistical Science*, 33(3):327–329, 2018.

[58] Swapnil Mishra, Marian-Andrei Rizoiu, and Lexing Xie. Feature driven and point process approaches for popularity prediction. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM)*, pages 1069–1078. ACM, 2016.

[59] George O Mohler, Martin B Short, P Jeffrey Brantingham, Frederic Paik Schoenberg, and George E Tita. Self-exciting point process modeling of crime. *Journal of the American Statistical Association*, 106(493):100–108, 2011.

[60] Stephen S Morse. Factors in the emergence of infectious diseases. *Plagues and politics*, pages 8–26, 2001.

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

[62] Nicholas Martin Navaroli and Padhraic Smyth. Modeling response time in digital human communication. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[63] Maximilian Nickel and Matthew Le. Learning multivariate Hawkes processes at scale. *CoRR*, abs/2002.12501, 2020.

[64] Gaëlle Nicolas, Benoît Durand, Raphaël Duboz, René Rakotondravao, and Véronique Chevalier. Description and analysis of the cattle trade network in the madagascar highlands: Potential role in the diffusion of rift valley fever virus. *Acta tropica*, 126(1):19–27, 2013.

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

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

[67] Yosihiko Ogata. Seismicity analysis through point-process modeling: A review. In *Seismicity Patterns, Their Statistical Significance and Physical Meaning*, pages 471–507. Springer, 1999.

[68] Yosihiko Ogata, Koichi Katsura, and Masaharu Tanemura. Modelling heterogeneous space–time occurrences of earthquakes and its residual analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 52(4):499–509, 2003.

[69] Yosihiko Ogata and David Vere-Jones. Inference for earthquake models: A self-correcting model. *Stochastic Processes and their Applications*, 17(2):337–347, 1984.

[70] Maya Okawa, Tomoharu Iwata, Takeshi Kurashima, Yusuke Tanaka, Hiroyuki Toda, and Naonori Ueda. Deep mixture point processes: Spatio-temporal event prediction with rich contextual information. In *Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 373–383. ACM, 2019.

[71] Maya Okawa, Tomoharu Iwata, Takeshi Kurashima, Yusuke Tanaka, Hiroyuki Toda, Naonori Ueda, and Hisashi Kashima. Deep mixture point processes. *Transactions of the Japanese Society for Artificial Intelligence*, 36(5):C–L37, 2021.

[72] 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 and Data Mining (KDD)*, pages 1276–1286. ACM, 2021.

[73] Maya Okawa, Tomoharu Iwata, Yusuke Tanaka, Hiroyuki Toda, Takeshi Kurashima, and Hisashi Kashima. Context-aware spatio-temporal event prediction via convolutional Hawkes processes. *Machine Learning Journal (Special Issue of ECML PKDD)*, 107(8-10):1283–1302, 2022.

[74] Takahiro Omi, Kazuyuki Aihara, et al. Fully neural network based model for general temporal point processes. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pages 2122–2132, 2019.

[75] Paul Edward Parham and Edwin Michael. Modeling the effects of weather and climate change on malaria transmission. *Environmental health perspectives*, 118(5):620–626, 2010.

[76] Jonathan A Patz, Peter Daszak, Gary M Tabor, A Alonso Aguirre, Mary Pearl, Jon Epstein, Nathan D Wolfe, A Marm Kilpatrick, Johannes Foufopoulos, David Molyneux, et al. Unhealthy landscapes: Policy recommendations on land use change and infectious disease emergence. *Environmental health perspectives*, 112(10):1092–1098, 2004.

[77] Robin Pemantle et al. A survey of random processes with reinforcement. *Probability surveys*, 4:1–79, 2007.

[78] Michael D Porter, Gentry White, et al. Self-exciting hurdle models for terrorist activity. *The Annals of Applied Statistics*, 6(1):106–124, 2012.

[79] H Pratiwi, I Slamet, DRS Saputro, et al. Self-exciting point process in modeling earthquake occurrences. *Journal of Physics: Conference Series*, 855(1):012033, 2017.

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

[81] Marian-Andrei Rizoiu, Young Lee, and Swapnil Mishra. Hawkes processes for events in social media. In *Frontiers of Multimedia Research*, pages 191–218. ACM / Morgan & Claypool, 2018.

[82] Marian-Andrei Rizoiu, Swapnil Mishra, Quyu Kong, Mark Carman, and Lexing Xie. SIR-Hawkes: Linking epidemic models and Hawkes processes to model diffusions in finite populations. In *Proceedings of the 27th International World Wide Web Conference (WWW)*, pages 419–428, 2018.

[83] Frederic Schoenberg, Marc Hoffmann, and Ryan Harrigan. A recursive point process model for infectious diseases. *arXiv preprint arXiv:1703.08202*, 2017.

[84] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 991–1001, 2017.

[85] Laura Serra, Marc Saez, Jorge Mateu, Diego Varga, Pablo Juan, Carlos Díaz-Ávalos, and Håvard Rue. Spatio-temporal log-Gaussian Cox processes for modelling wildfire occurrence: The case of catalonia, 1994–2008. *Environmental and Ecological Statistics*, 21(3):531–563, 2014.

[86] Joseph L Servadio, Samantha R Rosenthal, Lynn Carlson, and Cici Bauer. Climate patterns and mosquito-borne disease outbreaks in South and Southeast Asia. *Journal of infection and public health*, 11(4):566–571, 2018.

[87] Hua-Wei Shen, Dashun Wang, Chaoming Song, and Albert-László Barabási. Modeling and predicting popularity dynamics via reinforced Poisson processes. *CoRR*, abs/1401.0778, 2014.

[88] Masamichi Shimosaka, Keisuke Maeda, Takeshi Tsukiji, and Kota Tsubouchi. Forecasting urban dynamics with mobility logs by bilinear Poisson regression. In *Proceed-*

ings of the 17th ACM International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp), pages 535–546. ACM, 2015.

[89] Joseph Sill. Monotonic networks. In *Advances in Neural Information Processing Systems 12 (NeurIPS)*, pages 661–667, 1998.

[90] Monika Tanwar, Rajiv N Rai, and Nomesh Bolia. Imperfect repair modeling using Kijima type generalized renewal process. *Reliability Engineering and System Safety*, 124:24–31, 2014.

[91] Andrew J Tatem, David J Rogers, and Simon I Hay. Global transport networks and infectious disease spread. *Advances in Parasitology*, 62:293–343, 2006.

[92] Benjamin M Taylor, Tilman M Davies, Barry S Rowlingson, Peter J Diggle, et al. Lgcp: An r package for inference with spatial and spatio-temporal log-Gaussian Cox processes. *Journal of Statistical Software*, 52(4):1–40, 2013.

[93] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: A next-generation open source framework for deep learning. In *Proceedings of workshop on Machine Learning Systems (LearningSys) in the 22th Conference on Neural Information Processing Systems (NeurIPS)*, volume 5, pages 1–6, 2015.

[94] Mascha Van Der Voort, Mark Dougherty, and Susan Watson. Combining Kohonen maps with ARIMA time series models to forecast traffic flow. *Transportation Research Part C: Emerging Technologies*, 4(5):307–318, 1996.

[95] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 5998–6008, 2017.

[96] Boštjan Veber, Marko Nagode, and Matija Fajdiga. Generalized renewal process for repairable systems based on finite Weibull mixture. *Reliability Engineering and System Safety*, 93(10):1461–1472, 2008.

[97] Michael Wagner, Fuchiang Tsui, Gregory Cooper, Jeremy U Espino, Hendrik Harkema, John Levander, Ricardo Villamarin, Ronald Voorhees, Nicholas Millett, Christopher Keane, et al. Probabilistic, decision-theoretic disease surveillance and control. *Online Journal of Public Health Informatics*, 3(3), 2011.

[98] Jacco Wallinga and Peter Teunis. Different epidemic curves for severe acute respiratory syndrome reveal similar impacts of control measures. *American Journal of Epidemiology*, 160(6):509–516, 2004.

[99] Pengfei Wang, Yanjie Fu, Guannan Liu, Wenqing Hu, and Charu Aggarwal. Human mobility synchronization and trip purpose detection with mixture of Hawkes processes. In *Proceedings of the 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 495–503. ACM, 2017.

[100] Senzhang Wang, Lifang He, Leon Stenneth, Philip S Yu, and Zhoujun Li. Citywide traffic congestion estimation with social media. In *Proceedings of the 23rd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, page 34. ACM, 2015.

[101] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In *Proceedings of the 31st IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2589–2597, 2018.

[102] Yichen Wang, Nan Du, Rakshit Trivedi, and Le Song. Coevolutionary latent feature processes for continuous-time user-item interactions. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 4547–4555, 2016.

[103] Jeremy C Weiss and David Page. Forest-based point process for event prediction from electronic health records. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 547–562. Springer, 2013.

[104] Holger Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(1):389–396, 1995.

[105] Annelies Wilder-Smith and Duane J Gubler. Geographic expansion of dengue: The impact of international travel. *Medical Clinics of North America*, 92(6):1377–1390, 2008.

[106] Shuai Xiao, Mehrdad Farajtabar, Xiaojing Ye, Junchi Yan, Le Song, and Hongyuan Zha. Wasserstein learning of deep generative point process models. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 3247–3257, 2017.

[107] 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 31st AAAI Conference on Artificial Intelligence (AAAI)*, volume 17, pages 1597–1603, 2017.

[108] Lizhen Xu, Jason A Duan, and Andrew Whinston. Path to purchase: A mutually exciting point process model for online advertising and conversion. *Management Science*, 60(6):1392–1412, 2014.

[109] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, volume 32, 2018.

[110] Robail Yasrab. DCSeg: Decoupled CNN for classification and semantic segmentation. In *Proceedings of the International Conference on Knowledge and Smart Technologies*, 2017.

[111] Robail Yasrab, Naijie Gu, and Xiaoci Zhang. An encoder-decoder based convolution neural network for future advanced driver assistance system. *Applied Sciences*, 7(4):312, 2017.

[112] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3634–3640. ijcai.org, 2018.

[113] Han Zhang, Ian J. Goodfellow, Dimitris N. Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pages 7354–7363. PMLR, 2019.

[114] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[115] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. DNN-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, pages 1–4, 2016.

[116] Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive Hawkes process. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 11183–11193. PMLR, 2020.

[117] Yang Zhang and Yuncai Liu. Traffic forecasting using least squares support vector machines. *Transportmetrica*, 5(3):193–213, 2009.

[118] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2019.

[119] Qingyuan Zhao, Murat A Erdogdu, Hera Y He, Anand Rajaraman, and Jure Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1513–1522, 2015.

[120] Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. LSTM network: A deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2):68–75, 2017.

[121] Tian Zhou, Lixin Gao, and Daiheng Ni. Road traffic prediction by incorporating online information. In *Proceedings of the 23rd International World Wide Web Conference (WWW)*, pages 1235–1240. ACM, 2014.

[122] Shixiang Zhu, Shuang Li, and Yao Xie. Reinforcement learning of spatio-temporal point processes. *CoRR*, abs/1906.05467, 2019.

[123] Shixiang Zhu and Yao Xie. Crime linkage detection by spatio-temporal-textual point processes. *CoRR*, abs/1902.00440, 2019.

[124] Jiancang Zhuang, Yosihiko Ogata, and David Vere-Jones. Stochastic declustering of space-time earthquake occurrences. *Journal of the American Statistical Association*, 97(458):369–380, 2002.

[125] Eric Zivot and Jiahui Wang. Vector autoregressive models for multivariate time series. *Modeling Financial Time Series with S-Plus®*, pages 385–429, 2006.

[126] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer Hawkes process. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 11692–11702. PMLR, 2020.