Learning Multivariate Hawkes Process via Graph Recurrent Neural Network

Kanghoon Yoon*
KAIST ISysE
Daejeon, Republic of Korea
ykhoon08@kaist.ac.kr

Youngjun Im*
KAIST AI
Daejeon, Republic of Korea
youngjun@kaist.ac.kr

Jingyu Choi
Shinhan Card
Seoul, Republic of Korea
jingyu.choi@shinhan.com

Taehwan Jeong
Shinhan Card
Seoul, Republic of Korea
xikizima@shinhan.com

Jinkyoo Park†
KAIST AI & ISysE
Daejeon, Republic of Korea
jinkyoo.park@kaist.ac.kr

ABSTRACT

This paper presents a novel approach for modeling and predicting patterns of events in time-series learning, named graph recurrent temporal point process (GRTPP). Prior research has focused on using deep learning techniques, such as recurrent neural networks (RNNs) or attention-based sequential data embedding, on modeling the time-varying intensity of events. However, these models were typically limited to modeling a single intensity function capturing the event occurrence of all event types simultaneously. GRTPP addresses this issue by encoding multivariate event sequences into a sequence of graphs, where each node contains information about the event occurrence and time. The sequence of graphs is then embedded into node embeddings for each event type, taking into account the relationships between the event types. By integrating the estimated intensity functions, GRTPP predicts the event type and the timing of the next event. The proposed GRTPP model offers improved effectiveness and explainability compared to previous models, as demonstrated through empirical evaluations on five real-world datasets and the actual credit card transaction dataset. The code is available at https://github.com/im0j/GRTPP.

CCS CONCEPTS

• **Computing methodologies** → **Probabilistic reasoning**; **Temporal reasoning**.

KEYWORDS

Graph neural network, temporal point process, spatiotemporal learning, graphical event modeling

*Both authors contributed equally to this research.
†Corresponding Author

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599857

ACM Reference Format:
Kanghoon Yoon, Youngjun Im, Jingyu Choi, Taehwan Jeong, and Jinkyoo Park. 2023. Learning Multivariate Hawkes Process via Graph Recurrent Neural Network. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, Long Beach, CA, USA, 12 pages. https://doi.org/10.1145/3580305.3599857

1 INTRODUCTION

The ability to model and predict the patterns of events is of paramount importance across a variety of fields, including mechanical engineering, biology, geology, and social science [4, 5, 14]. Understanding the temporal and spatial patterns of events is crucial for addressing real-world challenges, such as tracking the spread of diseases like COVID-19, ensuring the quality of manufacturing processes, and gaining insight into the functional connections within the brain.

However, due to the complex interactions among events and the inherent stochasticity of event occurrences, it can be challenging to model and predict these patterns effectively. As a result, it is essential to develop a robust spatial-temporal event modeling scheme that considers the inter-dependencies between events.

A temporal point process (TPP) is a probabilistic model used to describe the random and dynamic patterns of event occurrences. This model is characterized by an intensity function that captures the time-varying rate of event occurrence. There have been various statistical and probabilistic models proposed for modeling TPP, including Poisson processes [8], Hawkes processes [6], and self-correcting processes [7]. These models typically use simple parametric functions to represent the intensity function, which can limit their expressiveness in fitting the event sequence. Some models have been extended to handle events in multiple regions or multiple types of events. For example, the multivariate Hawkes process models multiple events by marking events and considering the mutual influence between different event types [2, 5]. However, as the dimensionality of the data increases, this traditional model becomes increasingly difficult to apply due to its limitations in expressive power.

Recently, neural networks have been utilized in modeling TPPs in order to estimate the intensity of event occurrence using complex event sequence data. The core elements of such modeling involve learning the representation of complex event sequences to uncover

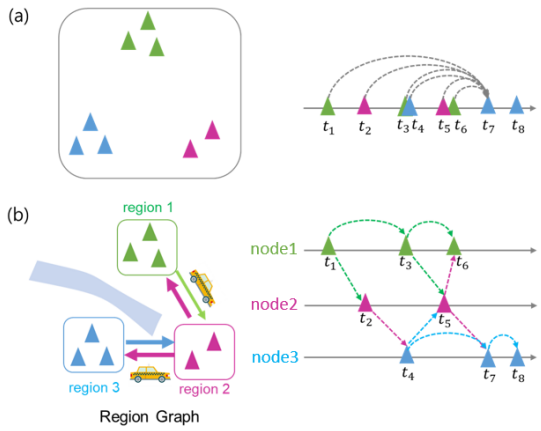


Figure 1: The comparison between (a) conventional multivariate event process and (b) Graph Recurrent Temporal Point Process (GRTPP). In the conventional model (a), the occurrence of a particular event type can be influenced by all the events, although they are irrelevant. On the other hand, in GRTPP (b), the occurrence of a particular event type is affected only by the relevant events (i.e., events that occurred from geographically connected regions).

the underlying intensity of the TPP, as well as modeling a flexible intensity function structure. Recurrent neural networks (RNNs) have been utilized to capture the non-linear and mutual effects between multiple event types over a continuous time period [3, 10, 11]. A self-attention layer has been implemented for history encoding, which enables the identification of important events among the event streams [17]. Additionally, the success of the Transformer model in natural language processing has inspired the adaptation of positional embedding and attention mechanisms for modeling Hawkes processes [19].

While neural network-based models have shown promise in expanding the capabilities of TPPs, they have limitations in capturing the relational information present within event sequences. These models often fail to effectively account for irrelevant events with regard to the target event being modeled. For instance, events in one region may not be directly influenced by events in another geographically distinct region, or a user's behavior on a website may shift abruptly to an entirely different topic. While the relationship between events, such as geographic proximity and topical similarity, contains valuable information, models such as RNN- and attention-based TPPs tend to compute the impact of all event pairs without considering these relationships. Additionally, many existing models only consider a single event point process with a marked type and thus cannot effectively capture changes in intensity functions resulting from interactions between event types.

In this work, we propose a graph recurrent temporal point process (GRTPP), a novel approach for modeling multivariate event streams that incorporates the relational information between events in the prediction of future event occurrences. To model co-influencing multivariate event streams, our approach merges the temporal point process framework with graphical event modeling (GEM), which utilizes graphical information to model the interactions between different event types [1, 16]. The event types and their relationships are represented as nodes and edges in the event graph, respectively. This graph representation allows for the modeling of event locality and the ability to model individual intensity functions per event type while considering the mutual influences among multiple event streams. Figure 1 illustrates the main differences between our model's history update procedure and the previous methods. By employing the graph with event sequence, GRTPP facilitates leveraging the prior knowledge like the proximity and the similarity in the dataset, which improves the generalization ability of learning TPP models.

The procedure of GRTPP consists of the three following steps: (1) constructing a graph and encoding the multivariate event sequences, along with their occurrence times, into a sequence of graphs, (2) as dynamic node embeddings for each event type while considering the interrelations among the multivariate event sequences with a graph, and (3) estimating the intensity of each event type based on the historical representations from the event nodes.

The main contributions and the novelties of the current study are summarized as:

- To the best of our knowledge, our study is the first to extend the existing TPP models to multivariate event modeling using an event-based graph recurrent neural network.
- With mutually updated intensity functions on a graph, the proposed model is shown to be applicable for real-world event streams with relational information such as the proximity of geographic region data and the similarity of web data.
- Our empirical results indicate that GRTPP outperforms existing state-of-the-art neural point process models in terms of learning a more flexible representation and making more accurate predictions for future event occurrence times.

2 PROBLEM DEFINITION

We define an event stream as $\mathcal{D}_n = \{(t_i, e_i)\}_{i=1}^n$, where t_i is the timestamp with sequence length n for which $t_1 < ... < t_n$, $e_i \in \{1, ..., E\}$ is the event type, and E is the total number of event types. We assume that E is known, and each event is expressed as a pair of the timestamp t_i and the event type e_i. We further assume a correlation between the event types, i.e., the occurrence of certain event types increases or decreases the occurrence probability of the other event types.

We model such interactions using a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, ..., v, ...E\}$ is the set of node indexes, and $\mathcal{E} = \{e^{uv}\}$ for $u \in \mathcal{V}$ and $v \in \mathcal{V}$ is the set of directed edges. Each node v corresponds to an event type, and the edge e^{uv} exists if the occurrence of event type u affects the occurrence of event type v.

We formulate the multivariate temporal point process on the graph by assigning each event type to each graph node and seek to estimate the conditional intensity function λ^v for all nodes $v \in \mathcal{V}$ (i.e., event types):

$$\lambda^v(t|\mathcal{H}_t) = \lim_{\Delta \to 0} \frac{E[N^v(t+\Delta) - N^v(t)|\mathcal{H}_t]}{\Delta} \tag{1}$$

where N^v is the counting process that counts the number of event type v until t, and $\mathcal{H}_t = \{(t_i, e_i)|t_i < t\}$ represents the event history until t. $\lambda^v(t|\mathcal{H}_t)$ estimates the event-occurring rate of v event type at time t, given $\mathcal{H}_t$.

With $\lambda^v(t|\mathcal{H}_t)$ defined, we can easily formulate the event stream modeling problem by maximizing the log-likelihood of the event data with respect to the model parameters θ. Once $\lambda^v(t|\mathcal{H}_t)$ is trained, we can use it to predict the time of the next event for all event types.

3 RELATED WORK

3.1 Neural Point Process

In the field of temporal point processes (TPP), most previous studies that utilized recurrent neural networks (RNN) aimed at learning a comprehensive historical representation, h_i, to embed the event history $\mathcal{H}_t = \{(t_i, e_i)|t_i < t\}$. The primary focus of these studies was to derive an expressive embedding for the sequence of time intervals and event types, $\{(d_1, e_1), ..., (d_i, e_i)\}$, where $d_i = t_i - t_{i-1}$, using RNN[3, 10]. Nevertheless, these methods are incapable of capturing the long-term dependencies between events. To address this issue, the transformer Hawkes process adopted the transformer architecture from the NLP domain and its ability to capture patterns in sequences [15]. This approach introduced the positional encoding and self-attention layer to TPP frameworks [19].

The ultimate objective of these efforts is to derive the hidden representation h_t that summarizes the event history $\mathcal{H}_t$ and to estimate the intensity function. Formally, neural TPP models estimate the conditional intensity function $\lambda(t|\mathcal{H}t)$ as follows:

$$\lambda(t|h_t) = \text{Softplus}(w \cdot h_t + \delta(t - t_{\text{last}}) + b), \tag{2}$$

where $w \in \mathbb{R}^M$ is a learnable parameter, $\delta \in \mathbb{R}$ is an event modulation parameter, and $b \in \mathbb{R}$ is the base intensity function for event type v. The *Softplus* function is utilized to guarantee that the intensity is positive.

In this study, we aim to learn a historical embedding that reduces the effect of irrelevant event information. To achieve this, we present the graph recurrent neural networok (GRNN), an encoder for historical events that combines graph neural networks and recurrent neural networks.

3.2 Graphical Event model

The marked point process task can be transformed into a Graphical Event Modeling (GEM) problem, in which events are represented by nodes, and the relationships between event types are indicated by edges. Unlike traditional TPP problems, GEM assumes that events are dependent only on their parent nodes in the graph. Consequently, the intensity function $\lambda^v(t|\mathcal{H}_t)$ depends solely on the features of the parent nodes in $\mathcal{G}$. That is, $\lambda^v(t|\mathcal{H}_t) = \lambda^v(t|\mathcal{H}_t^v)$ where $\mathcal{H}_t^v = \{(t_i, e_i)|t_i < t \text{ and } e_i \in \{\mathcal{N}(v) \cup v\}\}$ where $\mathcal{N}(v)$ are the set of neighbor for v. The relational graph structure enables the extraction of meaningful features from complex event data, thereby enhancing the representation power of hidden features. This study aims to develop neural TPP models that are based on the principles of GEM.

4 GRAPH RECURRENT TEMPORAL POINT PROCESS

We propose the graph recurrent temporal point process (GRTPP), a model for probabilistically modeling time-varying multi-event occurrences while considering the interactions between events. The architecture of GRTPP, depicted in Figure 2, consists of three steps: (1) graph construction and feature encoding, (2) dynamic node embedding, and (3) intensity function estimation and next-event time prediction. In the first step, connections between event nodes are defined through region proximity or topic similarity, and node features are encoded. Since the graph construction is data-specific, please refer to Appendix A for more details on constructing graphs. The second step involves learning a dynamic node embedding matrix $H_n = \{h_n^1, ..., h_n^v, ..., h_n^E\}$ to represent event history $\mathcal{H}_t$ at time t_n as sufficient statistic. Here, h_n^v denotes v^{th} rows of H_n. The use of GRNN allows the embedding matrix to incorporate both event occurrence dynamics and graphical information between events. GRTPP differs from other TPP methods in its ability to propagate messages to relevant nodes and exclude messages from irrelevant nodes. With the history embedding, GRTPP is able to estimate multiple intensity functions for every node.

4.1 Step 1: Graph construction and feature encoding

GRTPP first transforms event sequence $\mathcal{D}_n = \{(t_i, e_i)\}_{i=1}^n$ into the sequence of graph and node features $\{(\mathcal{G}_i, X_i)\}_{i=1}^n$, where $\mathcal{G}_i = \mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the graph and X_i is the node feature matrix, for all i. The node feature x_i^v, which is the v^{th} row of X_i, is defined as:

$$x_i^v = [\text{TE}(t_i), \mathbb{1}(v = e_i), c^v] \tag{3}$$

where:

- $\text{TE}(t_i) \in \mathbb{R}^d$ is the d dimensional temporal encoding with trigonometric function which has been proposed by [19]. The j^{th} component of $\text{TE}(t_i)$ is defined as:

$$[\text{TE}(t_i)]_j = \begin{cases} \cos\left(t_i / 10000^{\frac{j-1}{d}}\right), & \text{if } j \text{ is odd} \\ \sin\left(t_i / 10000^{\frac{j}{d}}\right), & \text{if } j \text{ is even.} \end{cases} \tag{4}$$

- $\mathbb{1}(v = e_i)$ is the binary indicator representing whether the occurred event coincides with the current node v. Note that if the other event type occurs, this feature becomes zero to signal the event type assigned to the current node.
- $c^v \in \mathbb{R}^d$ is the learnable vector for characterizing the event type v. These feature vectors assigned to each event type are introduced to differentiate the interaction between the event types effectively. For example, in Figure 2, events 2 and 4 can affect event type 3 differently. Without the event indicator vector, such a difference would be omitted from the model.

For more details on graph construction, please refer to Appendix A.

4.2 Step 2: Dynamic Node Embedding

Next, GRTPP computes the node embedding matrix H_n by employing the GRNN, and temporal attention [9, 15, 18] to the constructed graph sequence $\{(\mathcal{G}_i, X_i)\}_{i=1}^n$. The node embedding procedure is composed of two steps: (1) Spatial propagation and (2) Temporal propagation. Once H_n is obtained, which summarizes the past multivariate event sequences $\mathcal{H}_t$, we use it to predict the future event times and types.

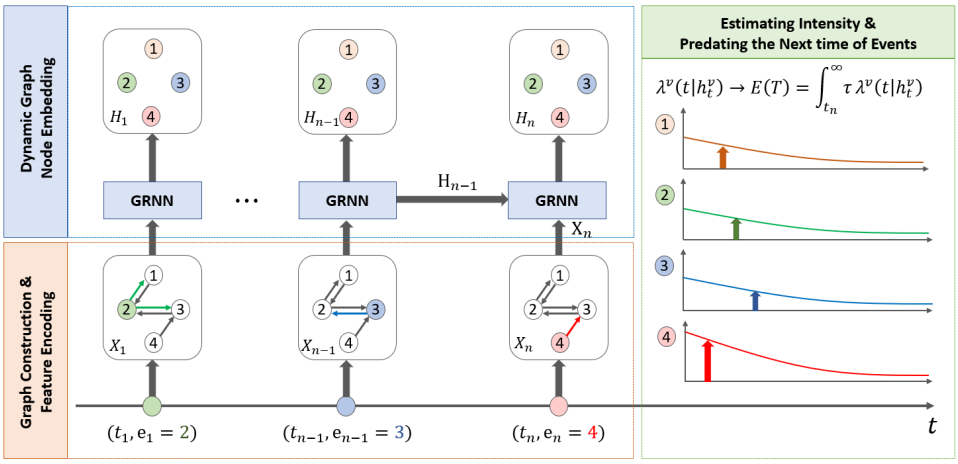


Figure 2: Overview of GRTPP architecture. Given $\mathcal{G}$ and (t_i, e_i), (1) GRTPP constructs a graph and node features by encoding the event feature, (2) embeds $\{\mathcal{G}_i, \mathcal{X}_i\}_{i=1}^n$ to learn the node representation H_i, and performs the intensity estimation and predicts the next event.

4.2.1 Spatial Module. The spatial module aggregates the influence of event occurrence and non-occurrence of the relevant events, which are represented by neighboring nodes, to update the representation of the target node. For example, at t_1, the event occurrence of node2 influences nodes 1 and 3, while non-event occurrences of node1 and node3 influence node 2.

We apply the graph neural network (GNN) to update the node embedding with a predefined event graph. Though various GNN schemes can be applied to our work, we select the message passing neural network (MPNN) to build the event propagation framework. MPNN updates input node features through the message producing, aggregating, and updating steps. The MPNN layer first produces messages m_i^{vu} to model the hidden relational information between the source node u and the destination node v at t_i as:

$$m_i^{vu} = f_m(x_i^v, x_i^u; \theta_m) \tag{5}$$

where f_m is the feed-forward neural network that generates the message. The generated messages are then aggregated as follows:

$$\bar{m}_i^v = \sum_{u \in \mathcal{N}(v)} \alpha_i^{vu} m_i^{vu} \tag{6}$$

where α_i^{vu} is the attention coefficient quantifying the importance of message m_i^{vu} to node v. α_i^{vu} is computed as:

$$z_i^{vu} = f_\alpha(x_i^v, x_i^u; \theta_\alpha) \tag{7}$$

$$\alpha_i^{vu} = \frac{\exp(z_i^{vu})}{\sum_{u \in \mathcal{N}(v)} \exp(z_i^{vu})} \tag{8}$$

where f_α is a feed forward neural network.

The aggregated messages $\bar{m}_i^v$ summarize the influence of events of other types on the target node v. $\bar{m}_i^v$ can be considered as the localized global state information for the target node v with respect to its neighboring event types. In addition, the attention mechanisms allow the learning to differentiate the importance of the interactions between different event types.

4.2.2 Temporal Module. The temporal module propagates the impact of the previous event history into the future while accounting for the aggregated node embedding computed from the spatial module. This is a type of transition model that projects past events into the future, thus allowing the model to estimate the event occurrence probability at any time in the future.

By using the previous node embedding h_{i-1}^v, the temporal module computes the updated node embedding h_i^v of target event type v as follows:

$$h_i^v = \text{RNN}(h_{i-1}^v, r_i^v) \tag{9}$$

where $r_i^v \in \mathbb{R}^d$ is the input feature for the temporal module. Depending on how $r_i^v \in \mathbb{R}^d$ is computed, the two variants of GRTPP are defined as:

- GRTPP: This is a basic model using $r_i^v = \bar{m}_i^v$ to use the spatially aggregated message $\bar{m}_i^v$ as input for updating the hidden temporal feature h_i^v.
- GRTPP-TA: This is an extended version of GRTPP that employs temporal self-attention (TA) [19] to compute the sequence of input $R = (r_1^v, ..., r_i^v, ..., r_n^v)$ as:

$$R = \text{Softmax}\left(\frac{QK^\top}{\sqrt{s_k}}\right) V \tag{10}$$

where $Q = \bar{\text{M}}^v W_Q, K = \bar{\text{M}}^v W_Q$, and $V = \bar{\text{M}}^v W_V$, are, respectively, the query, key, and value matrix. In addition, s_k is the scaling factor. The matrix $\bar{\text{M}}^v = [\bar{m}_1^v, ..., \bar{m}_n^v]^\top$ is the collection of the aggregated message in terms of node v using the spatial module. The weight parameters $W_Q \in \mathbb{R}^{d_Q \times n}$, $W_K \in \mathbb{R}^{d_K \times n}$, and $W_V \in \mathbb{R}^{d_V \times n}$ the ones need to be trained for temporal representation learning. By applying the scale-dot product, the temporal module extracts the temporal dependency, paying attention to the significant event among the event stream.

The updated node embedding matrix H_n then summarizes and localizes the history event sequence $\mathcal{H}_t$; it can be perceived as sufficient statistics. This temporal updating procedure simultaneously occurs for all nodes.

4.3 Step 3: Estimating Intensity and Next Event Time

Finally, GRTPP estimates $\lambda^v(t|h_n^v)$ using h_n^v for all $v \in \mathcal{V}$. GEM assumes that $\lambda^v(t|\mathcal{H}_n)$ depends on the event features of only the parent nodes of $\mathcal{G}$, i.e., $\lambda^v(t|\mathcal{H}_n) = \lambda^v(t|\{x_n^u|u \in \mathcal{N}(v)\})$. Since h_n^v is updated with the features of neighboring nodes $\{x_i^u|u \in \mathcal{N}(v)\}$ using MPNN in the spatial propagation module, it can replace the $\mathcal{H}_t^v$ with h_n^v. Thus, the conditional intensity function is modeled as follows:

$$\lambda^v(t|h_n^v) = \text{Softplus}(w^v \cdot h_n^v + \delta^v(t - t_n) + b^v) \tag{11}$$

where $w^v \in \mathbb{R}^M$ is a learnable parameter for all v, $\delta^v \in \mathbb{R}$ is an event modulating parameter, and $b^v \in \mathbb{R}$ is the base intensity function for event type v.

In Equation (11), the first term includes the hidden states of $\mathcal{N}(v)$ that encode past events before t_n. The second term models the influence of the last event. Its influence can either magnify or attenuate the intensity of the next event depending on the sign of δ^v. Using the likelihood $f(t) = \lambda^v(t|h_t^v)\exp\{-\sum_v \int \lambda(t|h_t^v)\}$, we compute the average of the next event time as $E[T] = \int_{t_n}^{\infty} \tau f(\tau) d\tau$.

4.4 Model Inference

In the training phase, given λ_v for $v \in \mathcal{V}$, we optimize the model parameters, i.e., the parameters for GRNN and the intensity functions, while maximizing the log-likelihood of $\mathcal{D} = \{(t_i, e_i)\}_{i=1}^n$. The log-likelihood can be calculated as follows:

$$\mathcal{L}(\mathcal{D}|\theta) = \sum_i \sum_{v \in \mathcal{V}_i'} \mathbb{1}(v = e_i) \log \lambda^v(t_i|h_i^v) - \int_{t_{i-1}}^{t_i} \lambda^v(\tau|h_i^v) d\tau \tag{12}$$

In Equation (12), the first term is related to the probability that $e_i = v$ occurs at t_i, while the second term is related to the probability that none of the event types occurs between the last timestamp of an event occurrence and the current timestamp $[t_{i-1}, t_i]$. Here, $\mathcal{V}_i'$ is the union of randomly sampled nodes from $\mathcal{V}$ and the node where e_i occurs. When E is large, only a small fraction of events can be used to fit the first term in Equation (12), while most non-event occurring data is used to suppress the intensities. The overfitted intensity to the non-event occurring data makes the trained intensity functions similar to each other. Thus, we substitute the non-event occurring likelihood using sampled events $\mathcal{V}_i'$ instead of considering all event types, which is analogous to training a supervised model with imbalanced labeled data. After tuning the hyperparameter through several experiments, we select the final number of samples as 8.

To estimate $\int_{t_{i-1}}^{t_i} \lambda^v(\tau|h_i^v) d\tau$, the Monte Carlo method is utilized, which is a well-known technique for approximating integrals. As per the method in previous studies such as [10, 19], the time interval is divided by taking 10 samples from a uniform distribution. However, when the timestamps are densely arranged, this approach incurs substantial computation time and memory usage. Therefore, we adopt the trapezoid rule to reduce the computational cost while maintaining accuracy, as suggested in [12].

5 EXPERIMENTS

This section presents the details of the experimental setup and reports the results of the experiments conducted using the various datasets. Additionally, a thorough analysis of the results of various ablation studies and qualitative evaluations is presented to assess the effectiveness of the model components.

5.1 Experimental Setup

5.1.1 Datasets. We employ the proposed GRTPP to model the multivariate event sequence data from various fields. Specifically, we use the **NYC TAXI**, **Reddit**, **Earthquake**, **Stack Overflow**, and **911 Calls** datasets, which have been used in the previous works[10, 13, 19]. These datasets were pre-processed in the same manner described in [10, 19].

The datasets consist of a sequence of timestamps and event types, $\mathcal{D}_n = \{(t_i, e_i)\}_{i=1}^n$, with varying numbers of event types. To incorporate the relational inductive bias into the probabilistic modeling of the event sequence, the graph is constructed using prior knowledge according to the domain of data. For example, the graph for the NYC taxi and earthquake datasets is generated based on geographic proximity, provided by the geographic tools such as NYC city information[1] and map data[2]. The graph for Reddit is based on user-post connectivity. In the absence of domain knowledge (namely for Stack Overflow), a complete graph is used.

- **NYC TAXI**: The NYC taxi[3] dataset contains millions of individual taxi pickup records for 2013-2019. The location information is available in the form of latitude/longitude coordinates. With this spatial information, we mapped the NYC Neighborhood Names dataset to each 299 event type. We generate the graph with 299 nodes assigned to each city and 7474 edges bridging the neighboring zone.
- **Reddit**: The Reddit[4] dataset includes the time of posting and its posting type. Though we had 2 million postings in January 2014, we randomly selected 1,000 users and 100 categories of subreddit. In other words, we have 100 event types and 1000 sequences having a different number of postings. We construct the graph by defining nodes as categories of posting. Then, we define the edges according to the posting history of users. If a user uploads posts on "video" and "movie," these two nodes are connected by edges, considering these two types of posting have an association. The Reddit graph contains 100 nodes and 9694 edges.
- **Earthquake**: The Earthquakes[5] includes earthquakes and aftershocks data in Japan from 1990 to 2020. We define 162 observatories where earthquakes occur as event nodes. Similarly, the latitude and longitude information of the observatory is used to generate edges of the graph. Finally, we use the graph with 162 nodes and 8100 edges.

[1] https://data.cityofnewyork.us/City-Government/Neighborhood-Names-GIS/99bc-9p23
[2] https://www.latlong.net/place
[3] http://www.andresmh.com/nyctaxitrips/
[4] https://github.com/dewarim/data-tools-for-reddit
[5] https://earthquake.usgs.gov/earthquakes/

- **Stack Overflow**: The Stack Overflow[6] is a question-answering website. We used sequences of answering time and reward badges as event types. This data have 480,413 events with 22 types of badges, and we divide them by four folds to repeatedly verify the performances. Since we can not intuitively take the domain knowledge for badges, we use the fully connected graph for the stack overflow dataset.
- **911 Calls**: 911 Calls[7] data records emergency phone calls. The calling time and location of the caller are logged during the 2019 year. We gather the region information of the caller to assign the event nodes to the city where the caller makes a call. By connecting nodes with the adjacency of the region, we make the geographic graph with 69 nodes and 272 edges.

5.1.2 Baselines. To validate the performances of the proposed GRTPP, we use three types of baseline models: traditional statistical models (*Stat*) and neural temporal point process models (*NPP*).

- *Stat* contains stochastic models with parametric intensity functions such as Poisson Process[8], Hawkes process[6], and self-correcting process[7].
- *NPP* contains RMTPP[3] and NHP[10], which use a simple RNN to embed the history of the event sequences, and THP[19] uses self-attention for sequential event embedding.
- *NGEM* contains GRTPP, which is a neural graphical event model.

We use these baselines to investigate the effectiveness of (1) the graph-based temporal and spatial encoding of multivariate event sequences and (2) the flexible intensity function modeled by a neural network that does not have a specific form.

5.1.3 Model Training. The hyperparameters for model training are explained in Table 5. Our models are trained from scratch without any pre-training steps. The models were trained on a single machine with NVIDIA Geforce RTX 3090 GPU.

5.2 Performance Evaluation

We first compare the prediction accuracy of the next event time, measured by the root mean squared error (RMSE). The comparison results between the proposed GRTPP model and the baseline models are summarized in Table 1, which presents the mean and standard deviation of the RMSEs. The results indicate that the GRTPP model achieved the lowest RMSE for the Reddit dataset, and the GRTPP with attention (GRTPP-TA) outperformed other baselines in three out of five datasets. The following are the primary observations:

- Deep learning-based approaches that incorporate the hidden embedding of the past event sequence outperformed traditional statistical models in terms of prediction accuracy due to the effective representation of the past events.
- The GEM models, which model each event type individually, outperformed the NPP models, which use a single RNN channel to model the multivariate event sequence in most datasets. This implies that the relative interaction among different event types improved the prediction accuracy. In the Stack Overflow dataset where the fully connected graph was used, the GRTPP model did not outperform THP, indicating that constructing meaningful graphs is crucial for effective embedding and predictability.

In addition, we also compare the complexity of GRTPP with other baseline models in terms of the number of learnable parameters. The model complexity and prediction accuracy are shown in Figure 3. The results indicate that the GEM models generally require more parameters than NPP models, as GEM models each event type individually using graph-node-based updating. Among GEM models, GRTPP outperforms THP in terms of both model complexity and accuracy. When an attention module is added to GRTPP, it further improves the accuracy but requires more parameters than THP and GRTPP. Thus, one can choose between GRTPP and GRTPP-TA based on the available computational resources.

5.3 Ablation Study

In this section, we aim to assess the impact of the various components of the proposed model on its performance.

5.3.1 Effects of GRNN. We first investigate whether the GRNN layer in GRTPP captures the relationships among multiple event types and improves predictability. To this end, we compare the performances of GRTPP with the RNN-based TPP (RTPP) without using graph information. Figure 4(a) and (b) compare the RMSE and the negative log-likelihood (NLL) for the two cases. For all datasets, a model that uses GRTPP as an encoder has a lower RMSE and NLL than that of RTPP. It implies that GNN and graph information are critical to both predictive performance and model flexibility. For the StackOverflow dataset, GRTPP shows a similar NLL value to RTPP. As we discussed, the graph information is not clearly defined for the StackOverflow dataset; hence, the GNN-based representation does not improve the accuracy.

5.3.2 Using Non-occurring Event Feature. The node feature of GRTPP contains the binary indicator $\mathbb{1}(v = e_i)$ that represents the event-occurring and non-event-occurring nodes. According to this, if the other event type occurs, the corresponding node feature is set to zero to signal that the event has not occurred. Thus, both the occurrence and non-occurrence of an event are treated equally as features to update GRTPP, and all nodes with or without event occurrence are updated simultaneously. To investigate the effectiveness of using non-event-occurrence information, we train the GRTPP by updating only the event-occurring nodes (GRTPP-E). Figure 5(a) and (b) verify that updating non-event-occurring nodes (GRTPP) decreases the RMSE and NLL compared to considering only event-occurring nodes (GRTPP-E). This result indicates that both the occurrence and non-occurrence of an event can be used as important features for modeling the multivariate event sequence.

5.4 Analysis

5.4.1 Spatial Attention Values for NYC TAXI. We investigate the node embedding and the attention values of GRTPP with the estimated intensities. Figure 6 visualizes the attention values of all edges in the graph computed by the spatial module for **NYC TAXI** dataset. We can see that GRTPP creates several communities based on the sequence of taxi ride events. Notably, the major city makes a relationship with neighboring regions referring to the geographic

[6]https://drive.google.com/drive/folders/0BwqmV0EcoUc8UklIR1BKV25YR1U?resourcekey=0-OrlU87jyc1m-dVMmY5aC4w
[7]https://www.kaggle.com/mchirico/montcoalert

Table 1: The performance evaluation of baselines and GRTPP for five datasets. The RMSE values and their standard deviations are presented. The best values are colored blue and the second best values are colored cyan.

		NYC Taxi	Reddit	Stack Overflow	Earthquake	911 Calls
Stat	Poisson	1.051 ± 0.000	307.4 ± 0.000	11.91 ± 0.000	103.7 ± 0.000	5.488 ± 0.000
	Hawkes	1.021 ± 0.000	48.54 ± 0.000	12.31 ± 0.000	13.24 ± 0.000	5.822 ± 0.000
	Self-correcting	1.042 ± 0.000	62.42 ± 0.000	11.35 ± 0.000	11.24 ± 0.000	5.543 ± 0.000
NPP	RMTPP	0.947 ± 0.003	16.18 ± 0.224	9.782 ± 0.052	9.105 ± 0.190	6.231 ± 0.052
	NHP	0.932 ± 0.003	15.88 ± 0.182	9.832 ± 0.082	9.081 ± 0.033	5.449 ± 0.012
	THP	0.842 ± 0.005	16.56 ± 0.309	**4.990 ± 0.019**	8.988 ± 0.107	5.329 ± 0.015
GEM	GRTPP	0.605 ± 0.031	**11.49 ± 0.121**	5.566 ± 0.084	7.509 ± 0.036	5.392 ± 0.022
	GRTPP-TA	**0.599 ± 0.021**	11.87 ± 0.152	5.522 ± 0.032	**7.415 ± 0.077**	**5.159 ± 0.033**

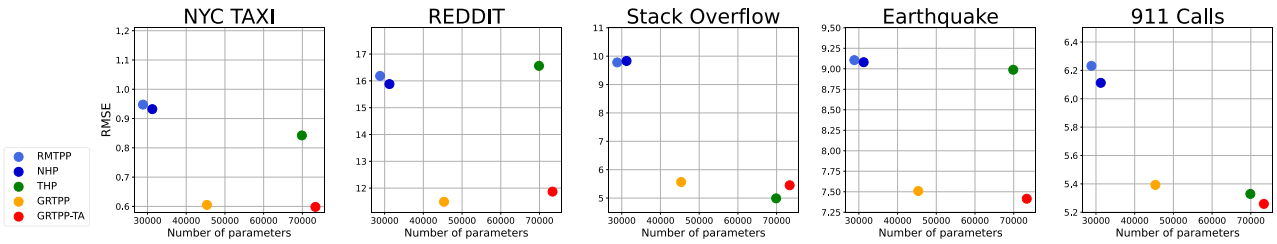
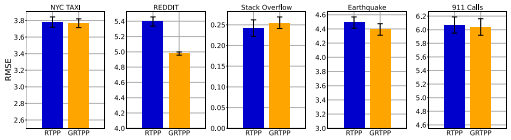
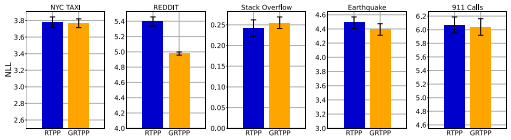


Figure 3: The number of parameters and RMSE for RMTPP, NHP, THP, and GRTPP (Ours). The x-axis represents the model complexity (number of model parameters), and the y-axis represents the mean RMSE. The model with both low complexity and error is preferred.

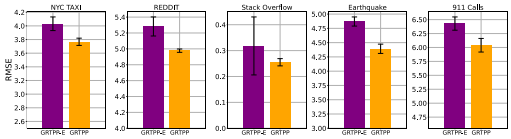


(a) RMSE

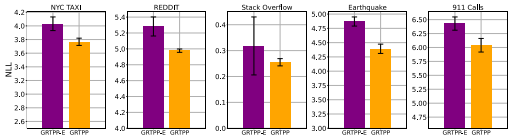


(b) NLL

Figure 4: Ablation study on five datasets when the GRTPP uses RNN (blue) and GRNN (yellow) for history representation; (a) RMSE for predictive performance and (b) NLL for representativity of RNN and GRNN layer, respectively.



(a) RMSE



(b) NLL

Figure 5: Ablation study on five datasets when the GRTPP uses only event-occurring nodes (purple) and when uses non-occurring event feature (yellow) for GRTPP update; (a) RMSE for predictive performance and (b) NLL for the representativity of each model.

path, even though the graph we provided for GRTPP does not contain information about the region's shape. It implies that our attention module identifies pivotal zones that serve as a transportation hub among the connected region. We further analyze the attention matrix for the other dataset and compare the sequential prediction of intensity function between GRTPP and RMTPP.

5.4.2 Relational Attention Values for Reddit. We analyze the attention values of GRTPP for the network dataset. Figure 7 illustrates the attention value between two subreddits (i.e., event category). The high attention values between source and destination can be considered as an influential category to each other. For *video*, the high attention values appear in *worldnews* and *movies*. Also, reliable patterns such as *pokemon* and *dogecoin* have emerged with high attention values. Though we only use the posting event information of the subreddit without any text information for the subreddit, GRTPP learns the semantics of the category from the patterns of the subreddit visit sequence.

5.4.3 Region Prediction Comparison. We further compare the intensity prediction of RMTPP and GRTPP. Figure 8 describes the

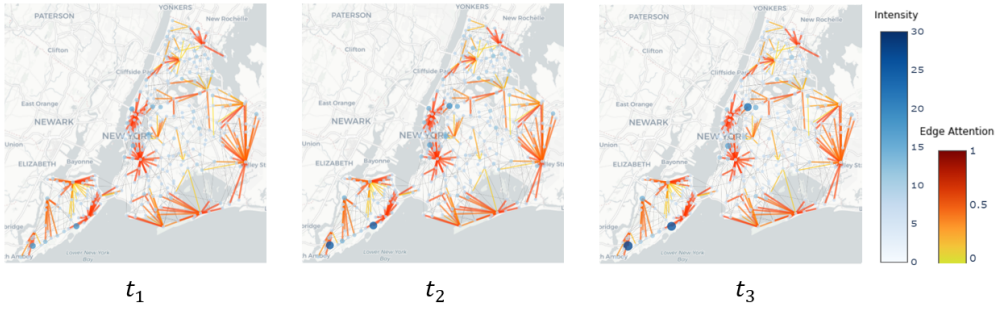


Figure 6: Visualization of graph attention values for NYC TAXI data. The thicker red lines represent high attention values, and the thin grey lines represent low attention values between the two regions.

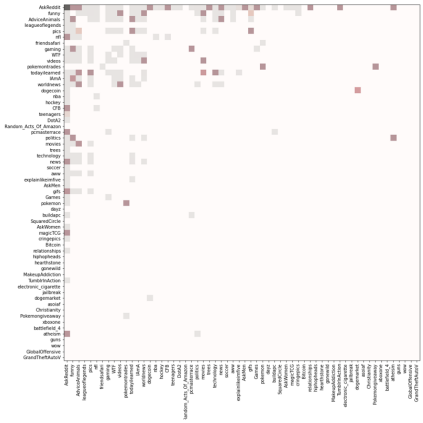


Figure 7: The matrix of attention values for Reddit data. The red represents the high attention value, and the grey represents the low attention value between the two subreddits. The white represents no edges connecting two subreddit.

intensity prediction on the NYC TAXI region. We can see that RMTPP predicts high intensity neglecting the region's closeness. On the other hand, GRTPP predicts the intensity based on the recent event occurrences of the nearby region. We can interpret that the GRTPP filters the unnecessary event occurrences (i.e., events on far regions) by propagating the event features through edges.

6 EXPERIMENTS ON REAL-WORLD DATASET

In this section, we evaluate the GRTPP and empirically validate its performance on real-world financial data, namely, the credit card transaction dataset. We describe the dataset and the experimental setup, followed by the presentation of the performance of GRTPP and the result analysis.

6.1 Dataset

We used the credit card transaction dataset from Shinhan Card, the major credit card company in Korea. The dataset comprises the anonymized transaction records of 4,050,335 sampled individuals for three years (2019 to 2021), where each transaction data records the type and region of the store with its corresponding timestamp. There are 20,702,115 events across 9 store types and 71 regions, resulting in 639 event types. This dataset poses a more demanding task compared to previous datasets, given that users' purchasing behaviors are only partially observable, as they may pay using alternative payment methods such as cash or other credit cards. Moreover, the large number of events requires a more compact model that is capable of handling multivariate event sequences efficiently. We constructed an event graph by connecting nodes with edges if the two stores share either the same type or region.

6.2 Performance Evaluation

The model's predictive performance was evaluated in terms of event type prediction, with a focus on two metrics: (1) Top-k accuracy (Top-k Acc.) measures the proportion of times the target event type is in the top-k predicted event types, and (2) Average Rank (Avg. Rank) measures the average rank of the target event type among the predicted event types. With these metrics, we measure the performance of GRTPP with three NPP models and the two primitive models that make predictions as the most frequent event type (Frequent) or the latest event type (Latest) are also included for comparison.

The generalization performance of the model was assessed in two scenarios: (1) training the model on past transactions (Jan. 2019 to Sep. 2021) and testing it on future transactions (Oct. 2021 to Dec. 2021), and (2) training the model on transactions of 11/12 of all users and testing it on transactions of the remaining users.

The results, presented in Table 2, show that GRTPP outperforms the baseline models and demonstrates competitive to superior event prediction performance in both scenarios. This suggests that GRTPP can effectively learn both temporal and spatial patterns from event sequences, resulting in improved predictions even for different time periods or users.

6.3 Analysis

To gain insights into the improved predictive performance of GRTPP, Table 3 presents three instances of credit card transaction data, along with the top-10 predictions made by GRTPP. Each data consists of a pair of store region and store type. Baseline models treat each event as separate, so they have to learn the relationship between all event pairs from scratch. On the other hand, GRTPP uses the inductive bias provided in the form of a graph. By providing a

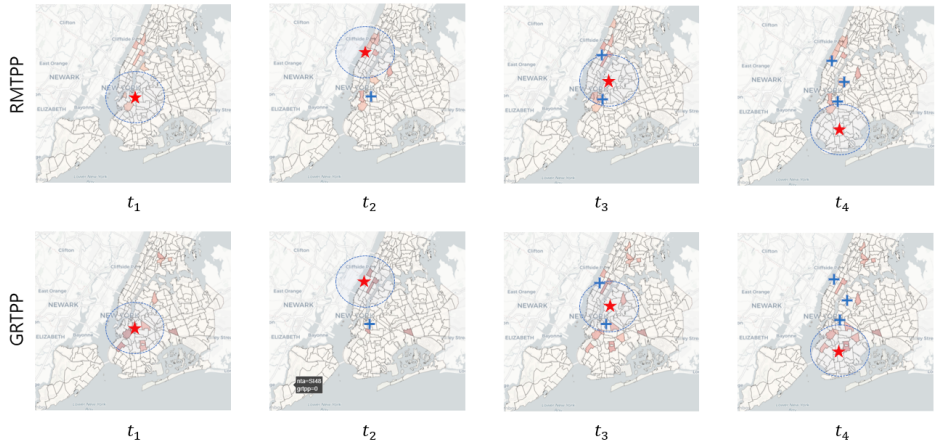


Figure 8: Visualization of the intensity function for NYC TAXI data. '+' and the star represent the previous and last event records, respectively. (above) RMTPP predicts the high intensities in sporadic regions because it captures the frequent occurrence in the event sequence. (below) GRTPP predicts the high intensities based on the distance of the regions using relational information (geometric graph).

Table 2: The generalization performance of GRTPP in comparison to the baseline models on the Shinhan Card transaction data. The evaluation results are shown in two aspects: (a) over time and (b) across users. The performance is presented in terms of Top-k accuracy ($k = 1, 3, 5$) and Average rank. The best values are colored blue and the second best values are colored cyan.

		(a) Time				(b) Users			
		Top-1 Acc.	Top-3 Acc.	Top-5 Acc.	Avg. Rank	Top-1 Acc.	Top-3 Acc.	Top-5 Acc.	Avg. Rank
Baselines	Frequent	24.00%	45.21%	55.70%	2.711	23.93%	43.20%	54.46%	2.621
	Latest	33.24%	54.08%	63.85%	2.015	32.37%	51.63%	61.93%	1.994
	RMTPP	34.02%	57.58%	65.60%	2.032	32.57%	56.47%	64.82%	1.853
	NHP	33.51%	56.32%	67.63%	1.920	33.21%	56.30%	65.56%	1.797
	THP	34.21%	59.79%	**71.77%**	1.823	33.05%	57.33%	**69.53%**	1.618
Ours	GRTPP	**34.42%**	**62.92%**	71.29%	**1.568**	**33.70%**	**60.78%**	69.16%	**1.561**

Table 3: The three sampled event streams $\{t_i, e_i\}_{i=1}^{10}$ along with the top-10 predictions made by GRTPP. Each e_i is a pair of store region and store type. The target events are boldened.

	Event Sequence										Top-10 Event Predictions									
e_i	(38,2)	(35,3)	(35,3)	(44,2)	(35,2)	(38,3)	(44,2)	(35,2)	(35,2)	(44,2)	(44,2)	(35,2)	**(44,6)**	(46,8)	(47,0)	(47,1)	(47,3)	(47,4)	(47,5)	(47,6)
t_i	0.000	6.252	9.048	39.348	40.452	43.842	48.084	49.356	51.906	52.392										
e_i	(20,2)	(20,2)	(5,0)	(22,6)	(22,6)	(22,6)	(22,6)	(0,2)	(0,7)	(46,3)	(46,3)	(22,6)	(0, 7)	(0,2)	(0,0)	**(47,0)**	(47,2)	(47,4)	(47,5)	(47,6)
t_i	0.000	0.630	7.950	34.200	34.206	34.206	34.206	35.310	35.784	61.638										
e_i	(40,2)	(40,2)	(44,3)	(50,3)	(40,2)	(44,2)	(44,2)	(44,2)	(44,3)	(50,3)	(50,3)	(44,2)	(40,2)	(44,3)	**(44,8)**	(44,6)	(47,1)	(47,2)	(47,3)	(47,4)
t_i	0.000	2.460	3.336	5.124	7.824	11.418	11.850	11.874	12.102	13.758										

graph, GRTPP can concentrate on finding the relationship between the events connected by edges in the graph.

7 CONCLUSION

We presented the graph recurrent temporal point process (GRTPP), a deep learning model capable of estimating the rate of occurrence for multiple events in continuous time. Our model outperforms most existing deep learning-based neural point process models that use a single RNN or attention model. Our work serves as a starting point for modeling TPPs using various graph neural network paradigms that extract relational information. However, the current work requires the graph representation that consists of event types and their relations from the dataset. We consider improving GRTPP by exploring the potential of using data-driven approaches to automatically generate the latent graph from the dataset as future work.

ACKNOWLEDGMENTS

This work was supported by Shinhan Card.

REFERENCES

[1] Debarun Bhattacharjya, Dharmashankar Subramanian, and Tian Gao. 2018. Proximal graphical event models. *Advances in Neural Information Processing Systems* 31 (2018).

[2] Shizhe Chen, Ali Shojaie, Eric Shea-Brown, and Daniela Witten. 2017. The multivariate Hawkes process in high dimensions: Beyond mutual excitation. *arXiv preprint arXiv:1707.04928* (2017).

[3] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1555–1564.

[4] Ambedkar Dukkipati, Tony Gracious, and Shubham Gupta. 2021. CoviHawkes: Temporal Point Process and Deep Learning based Covid-19 forecasting for India. *arXiv preprint arXiv:2109.06056* (2021).

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

[6] Alan G Hawkes and David Oakes. 1974. A cluster process representation of a self-exciting process. *Journal of Applied Probability* 11, 3 (1974), 493–503.

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

[8] John Frank Charles Kingman. 1992. *Poisson processes*. Vol. 3. Clarendon Press.

[9] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926* (2017).

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

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

[12] WH Press, SA Teukolsky, WT Vetterling, and BP Flannery. 1994. Numerical Recipes in C (2d ed. repr. with corr.; Cambridge.

[13] Jin Shang and Mingxuan Sun. 2019. Geometric hawkes processes with graph convolutional recurrent neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 4878–4885.

[14] Maria Nicolette Margaretha van Lieshout and A Stein. 2012. Earthquake modelling at the country level using aggregated spatio-temporal point processes. *Mathematical geosciences* 44, 3 (2012), 309–326.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[16] Xiufan Yu, Karthikeyan Shanmugam, Debarun Bhattacharjya, Tian Gao, Dharmashankar Subramanian, and Lingzhou Xue. 2020. Hawkesian Graphical Event Models. In *International Conference on Probabilistic Graphical Models*. PMLR, 569–580.

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

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

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

A GRAPH CONSTRUCTION

We classify our real-world datasets into two categories: geometric and web data. The geometric datasets have event streams that can be influenced by nearby regions. For example, the next region that a taxi picks up depends on the last event and the accessibility of regions. *NYC TAXI*, *Earthquake*, *911 Calls*, and *Shinhan Card* dataset belongs to geometric data. The web dataset has an event stream in the form of posting with posting time and the posting topics. It includes *Reddit* and *StackOverflow*.

A.1 Geometric Data

- **NYC TAXI**: Taxi demand patterns vary depending on the region of the city. We construct a graph to represent such spatial relationships between cities effectively. We define each city as a node in the graph. For all cities, we add edges between cities for the twenty-five closest cities. We used the longitude and latitude information to compute the distance between cities.
- **Earthquake**: Earthquakes can occur sporadically. When it first occurs, the aftershock will be derived based on the distance from the origin. We assign earthquake occurrence to the closest observatory (event nodes). We add edges between the city that is the ten closest observatories. We used the longitude and latitude information to compute the distance between observatories.
- **911 Calls**: For 911 calls, we assume that an accident in a certain region causes several calls from the place. We define all calls from certain regions as events and assign them based on the city to which the zip code belongs. We add edges between the city and its eight closest cities for the NYC TAXI construct graph.
- **Shinhan Card**: A credit card user's next purchase is likely to be influenced by the time, location, and category of the previous purchase. We define the product of the category and the region of the store as an event node. We used the graph where the edges are added between the event nodes that share the same category or the same region.

A.2 Web Data

- **Reddit**: The posting on Reddit shares several user interests. We assume that a user who likes soccer is likely to be interested in baseball. Thus, we construct the relational information graph based on the topic similarity of subreddits. We define all subreddits as event nodes. Then, we compute node similarity based on raw features and use it to construct a k-nearest neighbors graph. Thus, the relation graph of Reddit includes the event occurrence in similar subreddits.
- **StackOverflow**: For StackOverflow, the definition of an event is the type of obtained badge when a user answers some questions. Since it is difficult to define a graph in such a situation, we use a fully-connected graph.

B HYPERPARAMETERS

To guarantee the reproducibility of our work, we run our implementation several times to fill in the main performance table. For each dataset, we did a grid search for hyperparameter tuning, and hyperparameters for each dataset are described in Table 5. We select the number of stacking GNN layers to be 2. It shows that the optimal estimation for event likelihood is achieved by the 2-hop neighbor event. For the number of negatives, we set it to be 8 for all datasets.

Table 4: The data statistics of GRTPP for six datasets.

Dataset	# Event instances	# Event types	# Edges	Mean length
NYC TAXI	100 000	299	7474	38
Reddit	192 864	100	9694	104
Stack Overflow	480 413	22	231	72
Earthquake	256 932	69	8 100	500
911 Calls	290 293	82	272	403
Shinhan Card	20 702 115	639	24 921	10

Table 5: The hyperparameter settings for each dataset.

Dataset	NYC TAXI	Reddit	Stack Overflow	Earthquake	911 Calls	Shinhan Card
Hidden dimension	32	64	16	64	32	16
Hidden layers	2	2	2	2	2	2
Batch size	16	32	8	16	16	4