TransFeat-TPP: An Interpretable Deep Covariate Temporal Point Processes

Zizhuo Meng[a]**, Boyu Li**[a]**, Xuhui Fan**[b]**, Zhidong Li**[a]**, Yang Wang**[a]**, Fang Chen**[a] **and Feng Zhou**[c,d,*]

[a]Data Science Institute, University of Technology Sydney, Australia
[b]School of Computing, Macquarie University, Australia
[c]Center for Applied Statistics and School of Statistics, Renmin University of China, China
[d]Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, China

Abstract. The classical temporal point process (TPP) constructs an intensity function by taking the occurrence times into account. Nevertheless, occurrence time may not be the only relevant factor, other contextual data, termed covariates, may also impact the event evolution. Incorporating such covariates into the model is beneficial, while distinguishing their relevance to the event dynamics is of great practical significance. In this work, we propose a Transformer-based covariate temporal point process (TransFeat-TPP) model to improve the interpretability of deep covariate-TPPs while maintaining powerful expressiveness. TransFeat-TPP can effectively model complex relationships between events and covariates, and provide enhanced interpretability by discerning the importance of various covariates. Experimental results on synthetic and real datasets demonstrate improved prediction accuracy and consistently interpretable feature importance when compared to existing deep covariate-TPPs. Our code is available at https://github.com/waystogetthere/TransFeat.git.

1 Introduction

A temporal point process (TPP) is a mathematical framework used to describe the occurrence of events in time. It is a statistical model that is widely used in fields such as ecology [15, 35], epidemiology [7, 22], neuroscience [28, 41] and telecommunication [4, 10]. In a TPP model, events are represented as points in time, and the goal is to understand the underlying process that generates these events. The model can be used to predict the occurrence of future events, identify patterns or trends in the data, and estimate the probability of different outcomes. Classical TPPs include the Poisson process [13], Cox process [23] and Hawkes process [8], etc. These models are typically formulated in a parametric manner and often suffer from limited expressiveness. To overcome these limitations, deep TPPs leverage deep learning architectures and tools such as RNN [5, 20, 30], auto-grad in DNN [27], self-attention mechanism [38, 43] to more effectively model event generation processes.

However, an inherent flaw of both parametric and deep TPPs is that they cannot include additional information about relevant variables that may influence the occurrence of the events. On the contrary, a *covariate*-TPP [29, 34] is a type of TPP that incorporates additional information, called covariates, that may be related to the event occurrence. Covariates can be any relevant information, such as weather conditions, demographic characteristics or physical attributes of the location. It is expected that incorporating such covariates into events modelling can be more beneficial than considering events only, because by including covariates the model can account for the relationship between the events and other factors that may affect them. As far as we know, the research on covariate-TPPs is currently limited compared to the extensive results on TPPs.

Due to the rapid development of deep learning, the *deep* covariate-TPPs has emerged as a burgeoning topic, which refers to a type of covariate-TPPs that incorporates deep learning techniques to model complex relationships between events and covariates. For example, [37] integrates diverse time-series signals and events using two parallel RNNs, allowing for the joint modelling of event dynamics; [25] presents a deep mixture point process model that leverages rich, multi-source contextual data to define the intensity function using a combination of kernels. While aforementioned works can be flexible and powerful in modelling such relationships, the resulting model may be difficult to interpret and explain because neural networks are inherently black box in nature. This lack of interpretability can make it challenging for researchers to gain insights from the model and draw meaningful conclusions from the results. Moreover, existing covariate-TPPs works potentially make a strong assumption that the covariates to be incorporated always hold a strong relationship with the events' evolution. For instance, [37] incorporates the ATM's models, age, and location (covariates) to assist in the modelling of ATM's errors or failures (events). However, the selection of covariates always requires extra expertise to determine whether the covariates hold a strong relationship with the events of interest. Offering further insights about the covariates' effect on events can significantly improve the interpretability of covariate-TPPs but also remains an open challenge. In summary, while deep covariate-TPPs offer better expressiveness, their interpretability remains limited and researchers must balance these tradeoffs when using this approach.

To enhance the interpretability of deep covariate-TPPs while preserving their expressive power, we propose a **Trans**former-based covariate-**TPP** model equipped with **feat**ure importance (TransFeat-TPP). The Transformer is employed to encode sequences and model the relationships among events. Owing to its highly parallelizable architecture, the Transformer enables simultaneous computations, resulting in substantial improvements in training and inference speed compared to conventional RNN models [5, 20, 27, 37]. Furthermore, the Transformer's self-attention enables expressive input representations, addressing vanishing gradient issues in sequential models and

---

* Corresponding Author. Email: feng.zhou@ruc.edu.cn

enhancing long sequence handling. Most importantly, our model includes a module called the feature importance self-attention neural network (Fi-SAN). Fi-SAN, which is originally introduced in [32], acts as a classifier to predict the next event type while also unsupervisedly learning the importance of different features, allowing it to identify and prioritize the most relevant ones.

Specifically, our contributions are as follows: **(1)** We introduce TransFeat-TPP, a model designed to integrate supplementary covariates, which can provide valuable insights into the evolution of events. The Transformer architecture allows it to effectively model complex relationships between events and covariates. **(2)** In TransFeat-TPP, we incorporate the Fi-SAN module to discern the correlation between covariates and the occurrence of events. Unlike existing deep covariate-TPPs, our model can provide the importance of various covariates, thus offering enhanced interpretability. As far as we know, this work should be the first deep covariate-TPPs with covariate ranking. **(3)** We validate our model on both synthetic and real datasets. Experimental results demonstrate that our model achieves a higher prediction accuracy compared to traditional TPPs. More importantly, in contrast to existing deep covariate-TPPs, our model can consistently provide interpretable feature importance.

2 Related Work

In this section, we primarily review recent developments in the areas of TPPs, covariate-TPPs and feature importance.

2.1 Temporal Point Processes

Various models for TPPs have been proposed. Early statistical models include the Poisson process [13] and the Hawkes process [8], which laid the foundation for understanding event occurrences. The Poisson process assumes independent events, while the Hawkes process allows for event dependencies through self-excitation. In recent years, deep learning models have become popular for TPPs, offering improved prediction performance and modelling complex relationships. Examples include the recurrent marked temporal point process [5], neural Hawkes process [20], fully neural network point process [27], and Transformer Hawkes process [43], among others. These models leverage deep learning's expressive capabilities to capture temporal dependencies and enhance prediction accuracy in event sequences. However, the introduction of deep neural network may lead to some interpretability challenges due to its black-box nature.

2.2 Interpretability in Temporal Point Processes

Interpretability in temporal point processes remains an open challenge and has consistently attracted research interest in recent years. Explaining why events happen and revealing the causalities behind them is one aspect of interpretability. For example, temporal logic point processes [16, 17] utilize time logic rules to incorporate domain knowledge, aiming to elucidate the causality behind event dynamics. Additionally, some researchers focus on revealing the patterns of influence among event types, which are often explicitly learned through existing popular deep TPPs [5, 20, 39, 43]. Studies such as [31, 39] use graphical neural networks or self-attention mechanisms to explore these influencing patterns among event types. However, the aforementioned studies primarily focus on traditional TPPs, which consider only the event information. When it comes to covariate-TPPs, the scope of interpretability significantly broadens, presenting a rich area for further exploration and development.

In this work, our focus on interpretability primarily revolves around revealing the relevance of covariates to the event dynamics.

2.3 Covariate Temporal Point Processes

Covariate-TPPs have garnered significant attention due to their ability to model the impact of external factors on event sequences. To the best of our knowledge, current research either incorporates covariates as a factor in the intensity function [1, 26] or combines them with event sequences through deep neural networks [33, 37]. These models enabled researchers to analyze and predict event sequences more effectively, taking into account the various factors that influence event occurrences. However, treating covariates as a fixed factor in the intensity function explicitly results in reduced flexibility and expressiveness. Moreover, existing deep covariate-TPPs lack interpretability concerning the involvement of covariates. All these methods are based on the assumption that all covariates have a significant relationship with the events. This assumption requires domain expertise and may not always be valid in real-world scenarios.

2.4 Feature Importance

Feature importance plays a crucial role in interpreting machine learning models, particularly in applications where identifying relevant features is vital for decision-making and gaining insights. Various techniques have been developed to measure feature importance, ranging from traditional statistical methods like correlation coefficients [36] and information gain [12] to more advanced methods like permutation importance [2] and LASSO regularization [42]. In recent years, attention mechanisms have emerged as powerful tools for estimating feature importance. Fi-SAN [32] leverages self-attention to learn the significance of different features in an unsupervised manner. By assigning attention weights to each feature, the model can effectively identify the most relevant features and prioritize them during the learning process.

3 TransFeat-TPP

In this work, we propose a Transformer-based covariate-TPP model with feature importance. This model is capable of incorporating covariate information to predict the occurrence time and type of future events, while simultaneously assessing the relative importance of various covariates. The model structure is illustrated in Fig. 1 where our model comprises three components: the dependence module, the feature importance module and the decoder module. In the following, we elaborate on the function and implementation of each module individually.

3.1 Problem Definition

We present a concise definition of the problem we aim to address, along with an outline of the notations for the various variables employed throughout this work. Given a sequence of events $\mathcal{S} = \{(t_i, y_i, \mathbf{x}_i)\}_{i=1}^{L}$ where L is the number of events and each event is represented by a triplet: $t_i \in \mathcal{R}^+$ is the timestamp that the i-th event occurs; y_i represents the i-th event type which is a categorical variable; $\mathbf{x}_i \in \mathcal{R}^F$ is the covariate measured on the i-th event and F is the covariate dimension. Given the historical information $\mathcal{H}_{t_n} = \{(t_i, y_i, \mathbf{x}_i)\}_{i=1}^{n}$, our objective is to predict the timestamp and type of the next event $(\hat{t}_{n+1}, \hat{y}_{n+1})$.
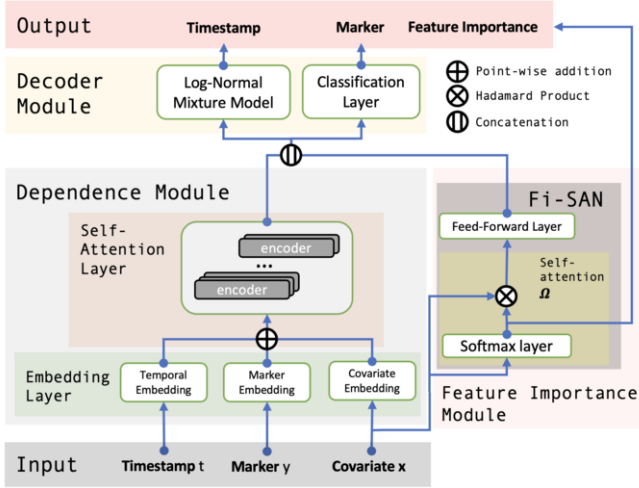
Figure 1: The model structure of TransFeat-TPP. TransFeat-TPP has three modules: the dependence module, the feature importance module and the decoder module. The dependence module extracts the dependencies among events to obtain the representation of the sequence; the feature importance module encodes the covariates and outputs their feature importance; the decoder module uses the learned representations from two previous modules to predict the next event's timestamp and type.

3.2 Dependence Module

The dependence module of TransFeat-TPP is designed to encode the sequence and learn the dependencies among events. Drawing inspiration from [43], we employ the Transformer architecture as the encoder. Specifically, we use the embedding layers to encode the triplets $(t_i, y_i, \mathbf{x}_i)$, and then adopt the self-attention layers to extract the dependencies among different events.

3.2.1 Embedding Layers

Given a sequence $\mathcal{S} = \{(t_i, y_i, \mathbf{x}_i)\}_{i=1}^L$, we have three types of inputs: timestamp, event type and covariate. We need to encode three kinds of information to obtain an appropriate representation. We utilize temporal encoding for event timestamps, representing a given timestamp t_i using an embedding vector:

$$[\mathbf{z}(t_i)]_j = \begin{cases} \cos\left(t_i/10000^{\frac{j-1}{M}}\right) & \text{if } j \text{ is odd} \\ \sin\left(t_i/10000^{\frac{j}{M}}\right) & \text{if } j \text{ is even} \end{cases} \in \mathcal{R}^M,$$

where M is the dimension of temporal embedding, and $[\mathbf{z}(t_i)]_j$ denotes the j-th entry of $\mathbf{z}(t_i)$. Therefore, the collection of time embedding is:

$$\mathbf{Z} = [\mathbf{z}(t_1), \mathbf{z}(t_2), \cdots, \mathbf{z}(t_L)]^\top \in \mathcal{R}^{L \times M}.$$

In addition to temporal embedding, we employ a learnable matrix $\mathbf{U} \in \mathcal{R}^{M \times K}$ to encode event type, where the k-th column of $\mathbf{U}$ represents the M-dimensional embedding vector for event type k. For event type y_i, its embedding $\mathbf{e}(y_i)$ can be expressed as:

$$\mathbf{e}(y_i) = \mathbf{U}\mathbf{k}_i \in \mathcal{R}^M,$$

where $\mathbf{k}_i \in \mathcal{R}^K$ is the one-hot encoding of y_i. Therefore, the collection of type embedding is:

$$\mathbf{E} = [\mathbf{e}(y_1), \mathbf{e}(y_2), \cdots, \mathbf{e}(y_L)]^\top \in \mathcal{R}^{L \times M}.$$

Analogous to event type embedding, we utilize a learnable matrix $\mathbf{W} \in \mathcal{R}^{M \times F}$ to encode covariates. For event covariate $\mathbf{x}_i$, its embedding $\mathbf{f}(\mathbf{x}_i)$ is expressed as:

$$\mathbf{f}(\mathbf{x}_i) = \mathbf{W}\mathbf{x}_i \in \mathcal{R}^M.$$

Therefore, the collection of covariate embedding is:

$$\mathbf{F} = [\mathbf{f}(\mathbf{x}_1), \mathbf{f}(\mathbf{x}_2), \cdots, \mathbf{f}(\mathbf{x}_L)]^\top \in \mathcal{R}^{L \times M}.$$

Combining the previously mentioned three types of embeddings, we can obtain the final embedding of the sequence $\mathcal{S} = \{(t_i, y_i, \mathbf{x}_i)\}_{i=1}^L$:

$$\mathbf{X} = \mathbf{Z} + \mathbf{E} + \mathbf{F} \in \mathcal{R}^{L \times M}, \tag{1}$$

where each row of $\mathbf{X}$ corresponds to the complete embedding of a single event in the sequence $\mathcal{S}$.

3.2.2 Self-attention Layers

After the embedding layers, our focus shifts to learning the dependence among events. This is achieved by passing the representation $\mathbf{X}$ to the multi-head self-attention layers. The attention output $\mathbf{S}_h$ of the h-th head is formulated as:

$$\mathbf{S}_h = \text{softmax}\left(\frac{\mathbf{Q}_h \mathbf{K}_h^\top}{\sqrt{M_K}}\right)\mathbf{V}_h \in \mathcal{R}^{L \times M_V},$$

$$\mathbf{Q}_h = \mathbf{X}\mathbf{W}_h^Q \in \mathcal{R}^{L \times M_K},$$

$$\mathbf{K}_h = \mathbf{X}\mathbf{W}_h^K \in \mathcal{R}^{L \times M_K},$$

$$\mathbf{V}_h = \mathbf{X}\mathbf{W}_h^V \in \mathcal{R}^{L \times M_V},$$

where $\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h$ are the query, key and value matrices of the h-th head obtained by different transformations from $\mathbf{X}$. M_K is the dimension of $\mathbf{Q}_h$ and $\mathbf{K}_h$. Matrices $\mathbf{W}_h^Q \in \mathcal{R}^{M \times M_K}$, $\mathbf{W}_h^K \in \mathcal{R}^{M \times M_K}$ and $\mathbf{W}_h^V \in \mathcal{R}^{M \times M_V}$ are the corresponding linear transformations of the h-th head. The final attention output is obtained by aggregating the output from H heads:

$$\mathbf{S} = [\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_H]\mathbf{W}^O \in \mathcal{R}^{L \times M},$$

where $\mathbf{W}^O \in \mathcal{R}^{HM_V \times M}$ is a learnable matrix that is used to combine the outputs of each head. Finally, the attention output $\mathbf{S}$ is passed through a position-wise feedforward network that consists of two dense layers to produce the hidden state $\mathbf{H}_1$:

$$\mathbf{H}_1 = \text{ReLU}(\mathbf{S}\mathbf{W}_1^{\text{FC}} + \mathbf{b}_1)\mathbf{W}_2^{\text{FC}} + \mathbf{b}_2 \in \mathcal{R}^{L \times M}, \tag{2}$$

where $\mathbf{W}_1^{\text{FC}}, \mathbf{W}_2^{\text{FC}}, \mathbf{b}_1, \mathbf{b}_2$ are the parameters of position-wise feedforward network. The hidden state $\mathbf{H}_1$ contains hidden representations of all events in the sequence, with each row corresponding to a specific event.

3.3 Feature Importance Module

To determine the importance of covariates at the feature level, we utilize an additional Fi-SAN module. This module, designed for a specific classification task, learns to fit data and extract each feature's relevance to the outcome using self-attention. Given a covariate $\mathbf{x}_i$ as input, the Fi-SAN module produces two outputs: the auxiliary representation of the covariates and their feature importance.

3.3.1 Auxiliary Representation

One of Fi-SAN's outputs is the auxiliary representation. To achieve this, $\mathbf{x}_i$ is firstly passed through a $\tilde{H}$-head self-attention layer $\boldsymbol{\Omega}$[1]:

$$\boldsymbol{\Omega}(\mathbf{x}_i) = \frac{1}{\tilde{H}} \sum_{h=1}^{\tilde{H}} \left[\mathbf{x}_i \otimes \mathrm{softmax}(\mathbf{W}_h^{\mathrm{FI}} \mathbf{x}_i + \mathbf{b}_h^{\mathrm{FI}}) \right] \in \mathcal{R}^F, \quad (3)$$

where $\mathbf{W}_h^{\mathrm{FI}}$ and $\mathbf{b}_h^{\mathrm{FI}}$ are the attention weights and bias of a softmax-activated layer of the h-th head. The softmax output can be interpreted as the importance score for each feature, which is then elementwise multiplied with $\mathbf{x}_i$ using the Hadamard product. The output of $\boldsymbol{\Omega}$ is the average over $\tilde{H}$ heads. The auxiliary representation is obtained from an additional dense layer on the top of $\boldsymbol{\Omega}$:

$$\mathbf{H}_2 = \boldsymbol{\Omega}_{\mathcal{S}} \mathbf{W}_3^{\mathrm{FC}} + \mathbf{b}_3^{\mathrm{FC}} \in \mathcal{R}^{L \times M},$$

$$\boldsymbol{\Omega}_{\mathcal{S}} = [\boldsymbol{\Omega}(\mathbf{x}_1), \boldsymbol{\Omega}(\mathbf{x}_2), \cdots \boldsymbol{\Omega}(\mathbf{x}_L)]^{\top} \in \mathcal{R}^{L \times F},$$

where $\mathbf{W}_3^{\mathrm{FC}}$ and $\mathbf{b}_3^{\mathrm{FC}}$ are the parameters of the dense layer, $\boldsymbol{\Omega}_{\mathcal{S}}$ is the result of passing all covariates $\{\mathbf{x}_i\}_{i=1}^L$ through $\boldsymbol{\Omega}$. $\mathbf{H}_2$ is the auxiliary representation that is used for modelling event type.

3.3.2 Feature Importance

Fi-SAN outputs the auxiliary representation while providing feature importance. Given the sequence $\mathcal{S}$, the final feature importance is the average of the feature importance across all points:

$$\mathrm{FI}(\mathbf{x}_i) = \frac{1}{\tilde{H}} \sum_{h=1}^{\tilde{H}} \mathrm{softmax}(\mathbf{W}_h^{\mathrm{FI}} \mathbf{x}_i + \mathbf{b}_h^{\mathrm{FI}}),$$

$$\mathrm{FI}(\mathcal{S}) = \frac{1}{L} \sum_{i=1}^{L} \mathrm{FI}(\mathbf{x}_i).$$

3.4 Decoder Module

Our objective is to predict the time and type of the next event. To accomplish this, we design the decoder module, which is responsible for the next event's time and type prediction. To predict the timestamp of the next event, we utilize an intensity-free method proposed by [30], which models the distribution of the next event's timestamp conditioned on the representation generated by the dependence module. To predict the type of the next event, we combine the representations from both dependence module and feature importance module, and use this combined representation to predict the next event type.

3.4.1 Timestamp Prediction

To model the interevent time, denoted by τ, which is the duration until the next event since the current event, we use a mixture of log-normal densities to represent the distribution of τ, since τ must be positive (details provided in Appendix D in [21]). The probability density function of a log-normal mixture distribution with C components is:

$$p(\tau | \mathbf{w}, \boldsymbol{\mu}, \mathbf{s}) = \sum_{c=1}^{C} w_c \frac{1}{\tau s_c \sqrt{2\pi}} \exp\left(-\frac{(\log \tau - \mu_c)^2}{2 s_c^2}\right),$$

where $\mathbf{w} = [w_1, \ldots, w_C]$ is the mixture weight, $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_C]$ and $\mathbf{s} = [s_1, \ldots, s_C]$ are the mixture mean and standard deviation. Let $\mathbf{h}_i$ be the representation for i-th event obtained by the dependence module (the i-th row of $\mathbf{H}_1$). We use $\mathbf{h}_i$ to define the parameters of the log-normal mixture distribution:

$$\mathbf{w}_i = \mathrm{softmax}(\mathbf{V}_w \mathbf{h}_i + \mathbf{b}_w), \quad \boldsymbol{\mu}_i = \mathbf{V}_\mu \mathbf{h}_i + \mathbf{b}_\mu,$$

$$\mathbf{s}_i = \exp(\mathbf{V}_s \mathbf{h}_i + \mathbf{b}_s),$$

where $\mathbf{V}_w, \mathbf{V}_\mu, \mathbf{V}_s, \mathbf{b}_w, \mathbf{b}_\mu, \mathbf{b}_s$ are the learnable parameters. Consequently, the time loss is the negative log-likelihood given by:

$$\mathcal{L}_1 = -\sum_{i=1}^{L} \log p(\tau_i | \mathbf{w}_i, \boldsymbol{\mu}_i, \mathbf{s}_i).$$

3.4.2 Type Prediction

We concatenate the representation $\mathbf{H}_1$ obtained from the dependence module with the representation $\mathbf{H}_2$ obtained from the feature importance module, and feed it into a softmax-activated dense layer that maps from $\mathcal{R}^{2M}$ to $\mathcal{R}^K$, where K is the number of event types:

$$\mathbf{P} = \mathrm{softmax}(\mathrm{FC}([\mathbf{H}_1, \mathbf{H}_2])) \in \mathcal{R}^{L \times K}.$$

The output of this layer provides the probability for each event type, so the cross entropy is used as the type loss:

$$\mathcal{L}_2 = -\sum_{i}^{L} \log p_i(y_i),$$

where $p_i(y_i)$ represents the probability of assigning i-th event to the correct label y_i. The total loss is the weighted sum of $\mathcal{L}_1$ and $\mathcal{L}_2$:

$$\mathcal{L} = \alpha_1 \mathcal{L}_1 + \alpha_2 \mathcal{L}_2,$$

where α_1 and α_2 are the parameters to balance the loss. We utilize the trick introduced in [11] to automatically adjust the weights based on the current magnitude of each loss component.

4 Experiments

We evaluate TransFeat-TPP on both synthetic and real datasets. We compare TransFeat-TPP to several deep TPP models and find that it is more flexible and interpretable. This is because TransFeat-TPP enables more expressive incorporation of covariates. At the same time, TransFeat-TPP is capable of effectively learning consistent feature importance of covariates for event type prediction, thus demonstrating its potential as an interpretable deep covariate-TPP.

4.1 Baseline

We compare our model against existing deep TPP models: recurrent marked temporal point process (**RMTPP**) [5], recurrent neural network point process (**RNNPP**) [37], Transformer Hawkes process (**THP**) [43].

RMTPP uses a recurrent neural network to learn the conditional intensity function by encoding event time and type, but it does not include covariates. To enable fair comparison, we propose **RMTPP+**, which incorporates covariate information. RNNPP models the temporal point process by incorporating both event and covariate information but offers limited interpretability regarding covariate relevance. THP employs the Transformer for enhanced computational efficiency, encoding only event time and type. We propose **THP+**, a modified version that includes covariate information. Details about RMTPP+ and THP+ are provided in Appendix A in [21].
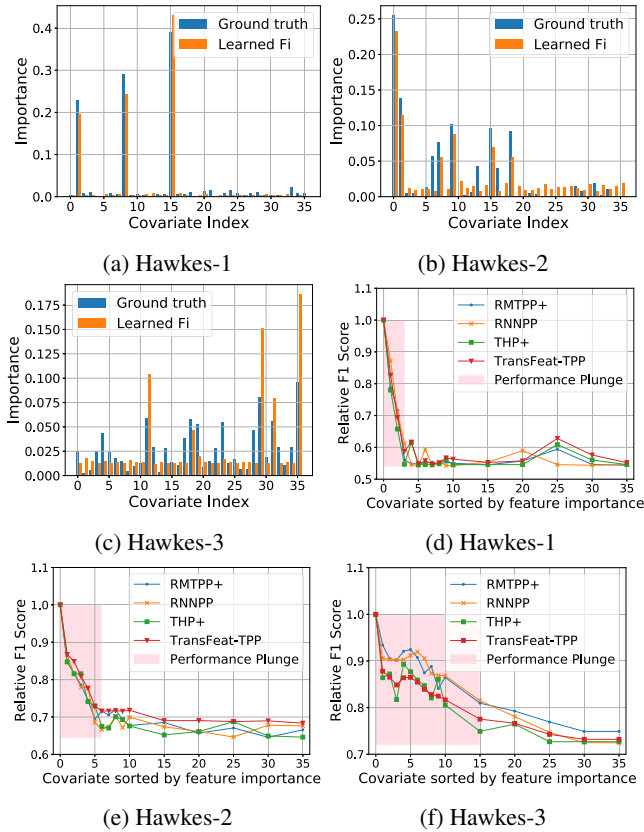
---

[1] The self-attention layer in Fi-SAN has a different structure with that in the dependence module.

(a) Hawkes-1

(b) Hawkes-2

(c) Hawkes-3

(d) Hawkes-1

(e) Hawkes-2

(f) Hawkes-3

Figure 2: The feature importance comparison and ablation study. (a)(b)(c): We compare the learned feature importance with the ground truth, verifying that TransFeat-TPP can differentiate between important and irrelevant covaraites. (d)(e)(f): We show the feature ablation study to indicate the consistency of the learned feature ranking. The x-axis represents the covariates sorted from most to least important. When more important covariates are removed, the models experience a significant performance drop. As less impactful covariates are removed, the performance decline becomes smoother.

4.2 Evaluation Metric

We evaluate the performance using a variety of widely-accepted metrics. (1) **log-likelihood**: A typical metric to evaluate the TPP model is the log-likelihood on test sets. However, RNNPP does not model the conditional intensity function and therefore it does not produce log-likelihood. As a result, we only compare to RMTPP and THP in terms of log-likelihood. (2) **Time Prediction**: For event time prediction, we employ the root mean square error (RMSE) as a metric to assess the discrepancy between the predicted and actual event times. (3) **Type Prediction**: For event type prediction, we use accuracy and F1 score as evaluation metrics. It is worth noting that in a multi-class classification problem, the F1 score is calculated as a weighted average across all classes.

4.3 Synthetic Data

In this section, we perform experiments on the synthetic covariate-TPP data, designed to resemble TPPs influenced by covariates. We verify that on one hand, by incorporating covariate information, our model can achieve better predictive performance; on the other hand, our model can uncover the latent feature importance of covariates.

4.3.1 Data Simulation

For the synthetic data, we assume covariates can impact both the arrival rate and type of future events. This is implemented by incorporating covariates into the conditional intensity function and considering covariates when assigning future events' types. Given historical events: $\{(t_i, y_i, \mathbf{x}_i)\}_{i=1}^n$, we assume the conditional intensity function can be written as: $\lambda(t) = f(\mathbf{x}_n, \mathcal{H}_t)$ where $\mathcal{H}_t$ is the historical information before t and $\mathbf{x}_n$ is the latest covariate before t. We use the Ogata thinning algorithm [24] to simulate the arrival time t_{n+1}. After that, we assume the event type y_{n+1} is binary and simulate it according to a Bernoulli distribution whose parameter p depends on the latest covariate $\mathbf{x}_n$ and the historical information $\mathcal{H}_t$: $p = g(\mathbf{x}_n, \mathcal{H}_t)$. It is worth noting that different point process models correspond to different functional forms of f and g.

4.3.2 Prediction Task

This section aims to demonstrate the improved predictive performance of our model through the incorporation of covariate information in an expressive way. To this end, we simulate two different types of covariate-TPP datasets, covariate inhomogeneous Poisson process and covariate Hawkes process, to validate the advantage of our model. The simulation details and statistics of the synthetic datasets are provided in Appendix B in [21]. We can observe that RMTPP+ and THP+ outperform their respective original models, indicating that the incorporation of covariates is beneficial for event prediction. Moreover, our proposed model TransFeat-TPP surpasses other baselines in all cases, which can be attributed to the incorporation of Fi-SAN. Fi-SAN concentrates on the covariates and delivers an auxiliary representation, enhancing the model's expressiveness and flexibility in handling covariates.

4.3.3 Feature Importance Task

Another advantage of our model is its ability to reveal the latent feature importance of covariates, thereby enhancing the model's interpretability. To achieve this, we validate our model on three Hawkes processes, labeled as Hawkes-1, 2 and 3, each with a unique feature importance configuration, corresponding to three levels of difficulty. Note that the Hawkes-1 is the "Hawkes" synthetic dataset presented in Table 1, while Hawkes-2 and 3 have more complex ground truth feature importance, making it more challenging for the model to distinguish between them. We present two categories of results: a comparison of feature importance and an ablation study. For the feature importance comparison, Figs. 2a to 2c demonstrate our learned feature importance in comparison to the ground truth. For Hawkes-1 where most covariates are irrelevant, TransFeat-TPP can readily capture the most important ones. As more covariates influence the event dynamics (Hawkes-2 and 3), our model remains capable of distinguishing the most important ones. The experimental results show that TransFeat-TPP can effectively identify the most important covariates while assigning lower importance to irrelevant ones. In the feature ablation study, we examine the impact of removing covariates according to the learned importance ranking. The results in Figs. 2d to 2f indicate that the model's performance experiences a significant decline when the top important covariates are removed, followed by a more gradual decrease when irrelevant ones are removed. This confirms the consistency of the learned ranking of feature importance.

Table 1: Experimental results from six different models on two synthetic datasets. The enhanced performance of the modified models, RMTPP+ and THP+, highlights the benefits of incorporating covariates. The superiority of TransFeat-TPP demonstrates the improvement in model expressiveness and flexibility. Standard deviation in brackets.

MODEL	INHOMOGENEOUS				HAWKES			
	LOG-LL	RMSE	ACC	F1	LOG-LL	RMSE	ACC	F1
RMTPP	-3.07 (0.092)	2.77 (0.075)	58.11% (0.010)	0.3844 (0.009)	-4.82 (0.091)	1.90 (0.113)	64.58% (0.042)	0.5165 (0.032)
RMTPP+	-2.89 (0.079)	2.80 (0.160)	88.30% (0.003)	0.8950 (0.010)	-3.98 (0.073)	1.75 (0.094)	84.04% (0.033)	0.8109 (0.015)
RNNPP	— —	3.26 (0.093)	85.63% (0.033)	0.8105 (0.013)	— —	1.89 (0.077)	85.33% (0.041)	0.8311 (0.033)
THP	-4.77 (0.043)	3.53 (0.054)	63.48% (0.098)	0.4188 (0.077)	-3.15 (0.092)	1.85 (0.071)	71.39% (0.043)	0.5415 (0.069)
THP+	-4.65 (0.019)	2.87 (0.054)	89.01% (0.033)	0.8901 (0.078)	-3.01 (0.037)	1.84 (0.043)	86.51% (0.013)	0.8663 (0.025)
TRANSFEAT	**-1.10** (0.053)	**2.41** (0.039)	**91.81%** (0.023)	**0.9132** (0.018)	**-2.37** (0.037)	**1.58** (0.050)	**89.06%** (0.015)	**0.8725** (0.025)

Table 2: Experimental results on two real datasets demonstrate the superior performance of modified models, RMTPP+ and THP+, compared to the original models. These findings provide compelling evidence that incorporating covariates enhances the predictive capabilities of the models. Our proposed TransFeat-TPP achieves the best performance in the majority of cases, highlighting the effectiveness of the Fi-SAN module in introducing greater flexibility and expressiveness. Note that the "NO Fi-SAN" indicates the TransFeat-TPP model without the Fi-SAN module, which is used for ablation study in Section 4.5. Standard deviation in brackets. For the PM2.5 dataset, we present results for Beijing and Shenyang here, while additional experimental results for other cities can be found in Table 8 in [21]

MODEL	PM2.5-BEJING				PM2.5-SHENYANG				CAR-ACCIDENT-LONDON			
	LOG-LL	RMSE	ACC	F1	LOG-LL	RMSE	ACC	F1	LOG-LL	RMSE	ACC	F1
RMTPP	-2.03 (0.012)	0.2876 (0.033)	45.11% (0.017)	0.4210 (0.011)	-1.84 (0.020)	0.2564 (0.056)	44.34% (0.011)	0.4187 (0.009)	-2.89 (0.095)	1.690 (0.101)	59.40% (0.054)	0.5447 (0.031)
RMTPP+	-1.64 (0.035)	0.2213 (0.029)	47.56% (0.035)	0.4653 (0.031)	-1.66 (0.020)	0.2255 (0.025)	49.34% (0.016)	0.4887 (0.016)	-2.78 (0.036)	1.271 (0.089)	65.03% (0.082)	0.5689 (0.096)
RNNPP	— —	0.2319 (0.010)	49.56% (0.017)	0.4811 (0.036)	— —	0.34 (0.021)	45.34% (0.025)	0.4377 (0.019)	— —	0.9181 (0.014)	63.06% (0.025)	0.5333 (0.013)
THP	-1.97 (0.077)	0.5094 (0.028)	44.28% (0.033)	0.3572 (0.036)	-2.24 (0.023)	0.3125 (0.021)	41.28% (0.025)	0.3272 (0.019)	-2.01 (0.049)	0.8994 (0.030)	58.12% (0.061)	0.5289 (0.033)
THP+	-1.86 (0.006)	0.4930 (0.091)	50.89% (0.005)	0.4883 (0.013)	-2.16 (0.009)	0.3881 (0.083)	49.53% (0.002)	0.4830 (0.022)	-1.98 (0.031)	0.8710 (0.082)	64.13% (0.003)	0.5579 (0.029)
TRANSFEAT	**-1.30** (0.051)	**0.1619** (0.070)	**51.70%** (0.089)	**0.4891** (0.066)	**-1.31** (0.016)	**0.1993** (0.051)	**51.24%** (0.071)	**0.5338** (0.035)	**-1.60** (0.041)	**0.6420** (0.009)	**66.33%** (0.046)	**0.6153** (0.015)
NO FI-SAN	-1.32 (0.010)	0.1703 (0.084)	49.07% (0.068)	0.4779 (0.054)	-1.34 (0.027)	0.2384 (0.083)	49.72% (0.067)	0.4818 (0.044)	**-1.60** (0.021)	0.6473 (0.011)	63.52% (0.013)	0.6014 (0.028)

4.4 Real Data

In this section, we verify our model on two real datasets: PM2.5 alert dataset and London car accident dataset.

4.4.1 PM2.5 Alert Dataset

PM2.5, a significant contributor to air pollution, triggers air pollutant alerts based on specific concentration thresholds. These alerts, characterized by timestamps and severity types, are considered noteworthy events. Besides, we also consider the hourly meteorological records as relevant covariates, which contain extra information such as temperature, humidity, precipitation, etc. More details can be found in appendix C.1 in [21]. The dataset[2] was collected from five Chinese cities (Beijing, Shenyang, Shanghai, Chengdu, and Guangzhou) between 2010 and 2015. For each city, separate models are trained using data from 2010 to 2013 for training, 2014 for validation, and 2015 for testing.

4.4.2 London Car Accident Dataset

Car accidents are a prevalent phenomenon in the real world and have attracted considerable research attention [18, 3]. We analyze a car accident dataset[3] containing all car accidents in London from 2005 to 2015. Additionally, relevant meteorological data[4] is considered as covariates for our analysis. Further information on both real datasets can be found in Appendix C in [21].

4.4.3 Prediction Task

We evaluate our model's predictive performance on the two mentioned real-world datasets. We reach a consistent conclusion on the real dataset that aligns with our findings on the synthetic dataset. Results shown in Table 2 indicate that the modified versions RMTPP+ and THP+ outperform their original counterparts in most cases. This indicates that incorporating meteorological data as covariates potentially improves the model's predictive capabilities for air pollution alerts and car accident predictions. Furthermore, our proposed TransFeat-TPP outperforms other baselines or achieves competitive results, which validates that the incorporation of Fi-SAN enables our model to be more expressive and flexible.
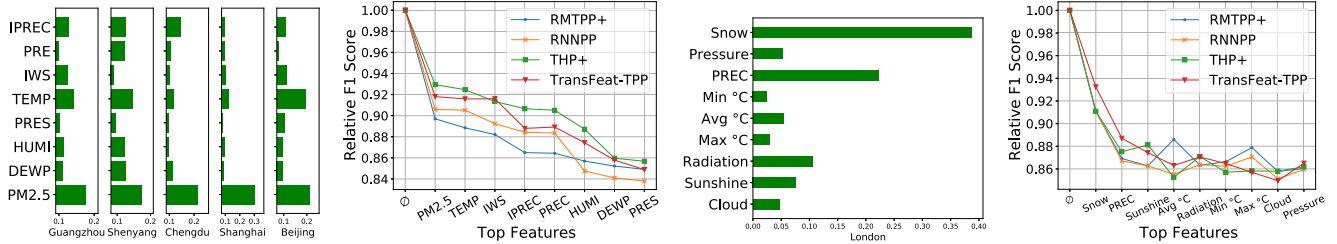
4.4.4 Feature Importance Task

We analyze the feature importance of covariates in relation to the future PM2.5 alert type. The results in Figs. 3a and 3b provide practical insights into which covariate has a significant impact on the future PM2.5 alert type. The comprehensive analysis leads us to a general conclusion that, consistently across various cities, the current concentration of PM2.5 plays a pivotal role in determining the type of future PM2.5 air quality alerts. This finding is both logical and reasonable. This is because the occurrence of an air pollution alert implies that the concentration of PM2.5 has transitioned from the current stage to a subsequent stage. Meanwhile, the temperature

---

[2] https://archive.ics.uci.edu/ml/datasets/PM2.5+Data+of+Five+Chinese+Cities

[3] https://www.kaggle.com/datasets/silicon99/dft-accident-data

[4] https://www.kaggle.com/datasets/emmanuelfwerr/london-weather-data

(a) Learned feature importance for PM2.5 Alert Dataset
(b) Feature ablation study for PM2.5 Alert Dataset
(c) Learned feature importance for London Car Accident Dataset
(d) Feature ablation study for London Car Accident Dataset

Figure 3: (a)(c): The learned feature importance for the PM2.5 alert dataset and the London car accident dataset, respectively. For the PM2.5 dataset, the two most important features are the current PM2.5 concentration and temperature, while for the car accident dataset, the two most critical environmental factors impacting accident severity are snow depth and precipitation. (b)(d): The feature ablation study for the PM2.5 alert dataset and the London car accident dataset. The x-axis represents the covariates sorted from most to least important. When the most important covariate (current PM2.5 concentration for the PM2.5 alert dataset and snow depth for the London car accident dataset) is removed, the models experience a significant performance drop. As less impactful covariates are removed, the performance decline becomes smoother.

is the second most important factor in most cities, which implies the seasonal variation pattern of PM2.5. This pattern has been studied and verified by numerous works [9, 14, 40]. Specifically, [14] proposed that significant seasonal variations are observed in Beijing, where the highest PM2.5 concentrations are typically observed in the winter, and the lowest are generally found in the summer. Similarly, we investigate the potential correlation between weather conditions and the severity levels of car accidents occurring during winter in London. In Figs. 3c and 3d, we can observe two significant covariates: snow depth and precipitation, which have a notable impact on accident severity. This finding is intuitive and aligns with common sense since both snowfall and rainfall can result in slippery road conditions and reduced visibility, potentially leading to more severe car accidents. In fact, this phenomenon has been studied and confirmed by numerous previous works [6, 19].

4.5 Ablation Study on Fi-SAN Module

We conduct an extra ablation study on the Fi-SAN module. Specifically, we remove the Fi-SAN module to create an ablation model named as "NO Fi-SAN". We compare its performance with other baseline models in Table 2. The results indicate that NO Fi-SAN demonstrates similar log-likelihood and RMSE performance but inferior classification performance (ACC, F1) compared to the original TransFeat-TPP. This suggests that Fi-SAN contributes to the classification performance more by introducing additional parameters into the model. In contrast, NO Fi-SAN surpasses all baselines except TransFeat-TPP in terms of log-likelihood and RMSE, which is attributed to the use of mixture log-normal modeling, a method previously validated by [30]. Therefore, we conclude that our TransFeat-TPP outperforms other models in predictive performance due to three key factors: (1) the incorporation of covariates (more relevant information); (2) the leveraging of the Fi-SAN module (improved event type prediction); (3) the utilization of mixture log-normal modeling (improved event time prediction).

4.6 Hyperparameter Analysis

Our model's configuration primarily encompasses several dimensions: the encoding dimension of temporal and type embeddings, denoted as M; the dimension of query and key embeddings, denoted as M_Q and M_K where $M_Q = M_K$; the dimension of the value

Table 3: We experiment with different hyperparameter configurations on one toy dataset (Hawkes-1) and one public dataset: PM2.5-Beijing. Note that we set $M = M_Q = M_K$ and $H = \tilde{H}$. The results indicate that our model is robust to these hyperparameters.

CONFIG	PM2.5-BEIJING		HAWKES-1	
	LOG-LL	ACC	LOG-LL	ACC
$M = 32, M_V = 16, H = 1, F = 32$	−1.58	47.19%	−2.48	88.00%
$M = 64, M_V = 32, H = 2, F = 64$	−1.30	51.70%	−2.37	89.06%
$M = 128, M_V = 64, H = 2, F = 128$	−1.29	52.03%	−2.39	88.12%
$M = 128, M_V = 64, H = 4, F = 128$	−1.30	51.11%	−2.40	87.84%

embeddings, denoted as M_V; the dimension of auxiliary representation, denoted as F; the number of attention heads in the dependence module and Fi-SAN, denoted as H and $\tilde{H}$. We test the sensitivity of our model to these hyperparameters by using various configurations. Specifically, we set $M = M_Q = M_K$, $H = \tilde{H}$, and conduct the hyperparameter analysis on one synthetic dataset and one public dataset: the Hawkes-1 dataset and the PM2.5 Beijing dataset. The results are shown in Table 3 which indicates that our model is not significantly affected by the hyperparameter variation. Besides, it can achieve reasonably good performance even with fewer parameters.

5 Conclusion and Discussion

We explore interpretability in covariate-TPPs and propose a novel TransFeat-TPP, a model that incorporates covariates in event modeling and extracts insights into feature importance. Our experiments show that incorporating covariates enhances event modelling, with our model outperforming other baselines, and the modified versions (RMTPP+, THP+) outperforming their original versions (RMTPP, THP). The ablation study supports the consistency of our learned feature importance ranking, highlighting the significant contribution of top features and the minimal impact of less important ones. Our proposed TransFeat-TPP effectively manages covariates and offers deeper insights than other baselines. This work is a significant advancement in exploring interpretability in deep TPPs. For future investigations, we aim to delve further into the interpretability of deep temporal point process models. Specifically, we plan to explore additional aspects, such as identifying which covariates influence specific types of events and understanding the nature of their effects, for insance, stimulation or inhibition. By addressing these aspects, we aim to gain a more comprehensive understanding of the role of covariates in temporal point process modelling.

References

[1] G. Adelfio and M. Chiodi. Including covariates in a space-time point process with application to seismicity. *Statistical Methods & Applications*, 30:947–971, 2021.

[2] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10): 1340–1347, 2010.

[3] R. Bergel-Hayat, M. Debbarh, C. Antoniou, and G. Yannis. Explaining the road accident risk: Weather effects. *Accident Analysis & Prevention*, 60:456–465, 2013.

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

[5] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1555–1564, 2016.

[6] G. Fountas, A. Fonzone, N. Gharavi, and T. Rye. The joint effect of weather and lighting conditions on injury severities of single-vehicle accidents. *Analytic methods in accident research*, 27:100124, 2020.

[7] A. C. Gatrell, T. C. Bailey, P. J. Diggle, and B. S. Rowlingson. Spatial point pattern analysis and its application in geographical epidemiology. *Transactions of the Institute of British geographers*, pages 256–274, 1996.

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

[9] T. Huang, Y. Yu, Y. Wei, H. Wang, W. Huang, and X. Chen. Spatial–seasonal characteristics and critical impact factors of pm2. 5 concentration in the beijing–tianjin–hebei urban agglomeration. *PLoS one*, 13(9): e0201364, 2018.

[10] B. Jahnel and W. König. *Probabilistic Methods in Telecommunications*. Springer, 2020.

[11] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[12] J. T. Kent. Information gain and a general measure of correlation. *Biometrika*, 70(1):163–173, 1983.

[13] J. F. C. Kingman. *Poisson processes*, volume 3. Clarendon Press, 1992.

[14] L. Kong, Q. Tan, M. Feng, Y. Qu, J. An, X. Liu, N. Cheng, Y. Deng, R. Zhai, and Z. Wang. Investigating the characteristics and source analyses of pm2. 5 seasonal variations in chengdu, southwest china. *Chemosphere*, 243:125267, 2020.

[15] R. Law, J. Illian, D. F. Burslem, G. Gratzer, C. Gunatilleke, and I. Gunatilleke. Ecological information from spatial patterns of plants: insights from point process theory. *Journal of Ecology*, 97(4):616–628, 2009.

[16] S. Li, L. Wang, R. Zhang, X. Chang, X. Liu, Y. Xie, Y. Qi, and L. Song. Temporal logic point processes. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5990–6000. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/li20p.html.

[17] S. Li, M. Feng, L. Wang, A. Essofi, Y. Cao, J. Yan, and L. Song. Explaining point processes by learning interpretable temporal logic rules. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=P07dq7iSAGr.

[18] Z. Li, L. Cui, and J. Chen. Traffic accident modelling via self-exciting point processes. *Reliability Engineering & System Safety*, 180:312–320, 2018.

[19] Z. Liu, L. Shen, C. Yan, J. Du, Y. Li, and H. Zhao. Analysis of the influence of precipitation and wind on pm2. 5 and pm10 in the atmosphere. *Advances in Meteorology*, 2020:1–13, 2020.

[20] H. Mei and J. M. Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems*, 30, 2017.

[21] Z. Meng, B. Li, X. Fan, Z. Li, Y. Wang, F. Chen, and F. Zhou. Transfeat-tpp: An interpretable deep covariate temporal point processes, 2024. URL https://arxiv.org/abs/2407.16161.

[22] S. Meyer. *Spatio-temporal infectious disease epidemiology based on point processes*. PhD thesis, Institut für Statistik, 2010.

[23] J. Møller, A. R. Syversveen, and R. P. Waagepetersen. Log gaussian cox processes. *Scandinavian journal of statistics*, 25(3):451–482, 1998.

[24] Y. Ogata. On lewis' simulation method for point processes. *IEEE transactions on information theory*, 27(1):23–31, 1981.

[25] M. Okawa, T. Iwata, T. Kurashima, Y. Tanaka, H. Toda, and N. Ueda. Deep mixture point processes: Spatio-temporal event prediction with rich contextual information. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 373–383, 2019.

[26] M. Okawa, T. Iwata, Y. Tanaka, H. Toda, T. Kurashima, and H. Kashima. Dynamic hawkes processes for discovering time-evolving communities' states behind diffusion processes. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1276–1286, 2021.

[27] T. Omi, K. Aihara, et al. Fully neural network based model for general temporal point processes. *Advances in neural information processing systems*, 32, 2019.

[28] L. Paninski. Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems*, 15 (4):243–262, 2004.

[29] A. Reinhart and J. Greenhouse. Self-exciting point processes with spatial covariates. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 67(5):1305–1329, 2018.

[30] O. Shchur, M. Biloš, and S. Günnemann. Intensity-free learning of temporal point processes. *arXiv preprint arXiv:1909.12127*, 2019.

[31] X. Shou, T. Gao, D. Subramanian, D. Bhattacharjya, and K. Bennett. Influence-aware attention for multivariate temporal point processes. In *Conference on Causal Learning and Reasoning*, pages 499–517. PMLR, 2023.

[32] B. Škrlj, S. Džeroski, N. Lavrač, and M. Petkovič. Feature importance estimation with self-attention networks. *arXiv preprint arXiv:2002.04464*, 2020.

[33] A. Solomon, M. Kertis, B. Shapira, and L. Rokach. A deep learning framework for predicting burglaries based on multiple contextual factors. *Expert Systems with Applications*, 199:117042, 2022.

[34] W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of neurophysiology*, 93(2):1074–1089, 2005.

[35] D. I. Warton and L. C. Shepherd. Poisson point process models solve the" pseudo-absence problem" for presence-only data in ecology. *The Annals of Applied Statistics*, pages 1383–1402, 2010.

[36] E. W. Weisstein. Correlation coefficient. *https://mathworld. wolfram. com/*, 2006.

[37] S. Xiao, J. Yan, X. Yang, H. Zha, and S. M. Chu. Modeling the intensity function of point process via recurrent neural networks. In S. Singh and S. Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 1597–1603. AAAI Press, 2017.

[38] Q. Zhang, A. Lipani, O. Kirnap, and E. Yilmaz. Self-attentive hawkes process. In *International conference on machine learning*, pages 11183–11193. PMLR, 2020.

[39] Q. Zhang, A. Lipani, and E. Yilmaz. Learning neural point processes with latent graphs. In *Proceedings of the Web Conference 2021*, pages 1495–1505, 2021.

[40] M. Zheng, L. G. Salmon, J. J. Schauer, L. Zeng, C. Kiang, Y. Zhang, and G. R. Cass. Seasonal trends in pm2. 5 source contributions in beijing, china. *Atmospheric Environment*, 39(22):3967–3976, 2005.

[41] F. Zhou, Q. Kong, Z. Deng, J. Kan, Y. Zhang, C. Feng, and J. Zhu. Efficient inference for dynamic flexible interactions of neural populations. *Journal of Machine Learning Research*, 23(211):1–49, 2022.

[42] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

[43] S. Zuo, H. Jiang, Z. Li, T. Zhao, and H. Zha. Transformer hawkes process. In *International conference on machine learning*, pages 11692–11702. PMLR, 2020.