Journal of Big Data

RESEARCH Open Access

EventFormer: a hierarchical neural point process framework for spatio-temporal clustering events prediction

Shanshan Yu[1], Danhuai Guo[1*], Yanjie Fu[2] and Wei Jin[3,4]

*Correspondence:
gdh@buct.edu.cn

[1] Spatio-Temporal Data Intelligence Lab, School of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China
[2] School of Computing and Augmented Intelligence, Arizona State University, Tempe 85281, USA
[3] Beijing Academy of Science and Technology, Beijing 100094, China
[4] Beijing Institute of New Technology Applications, Beijing 100094, China

Abstract

In real-world scenarios, event data often exhibits inherent randomness, complex historical dependencies, and hierarchical spatio-temporal clustering. However, existing neural point process models typically overlook the hierarchical nature of spatial information, or treat temporal and spatial relevance as separate factors. This oversight results in suboptimal performance when handling data with prominent spatio-temporal clustering features. Additionally, current models have not explicitly considered the interdependencies between event types. To remedy these limitations, we introduce a novel neural point process framework named EventFormer. Leveraging the time-oriented and type-oriented multi-head attention modules, along with a Ladder Attention mechanism that progressively refines spatial embeddings across hierarchical levels, EventFormer adeptly captures the nuanced dynamics of event occurrences. Furthermore, EventFormer incorporates a type-aware conditional intensity function to explicitly model interactions between event types, enhancing both predictive accuracy and interpretability. Extensive experiments on real-world datasets demonstrate the outstanding performance of EventFormer in event likelihoods modeling and prediction tasks.

Keywords: Event sequence prediction, Neural point process, Hierarchical spatio-temporal clustering, Attention mechanism

Introduction

In the real world, diverse activities generate substantial event sequence data, which consists of a series of events with specific information, such as timestamps and classification. Examples include user interactions on social media (e.g., posting and reposting) [1], financial transactions (e.g., buying and selling) [2], patient visit records in healthcare systems [3], and records of hotline calls. Accurately modeling the distribution of event sequences can reveal underlying patterns in real-world activities, enabling precise predictions of future events [4]. Such predictive capabilities are crucial for proactive decision-making, particularly in mitigating potential risks before they materialize [5].

Event sequences often demonstrate spatio-temporal clustering characteristics, where events occur frequently within certain time periods and are concentrated in

Springer Open

© The Author(s) 2025. **Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

specific geographic regions. Moreover, unlike traditional time series data, which typically involves sensor-based measurements such as traffic flow [6], event sequence data presents additional challenges. First, events occur irregularly within a continuous time frame, leading to dynamic and variable intervals between occurrences. Second, the presence of multiple event types exacerbates the complexity of historical dependencies, where certain event types may stimulate or inhibit others. The above features make it difficult to model and predict future events accurately. Therefore, achieving this requires addressing the inherent complexities of event sequences, including their spatio-temporal clustering properties, irregular time intervals, and multi-type dependencies.

Temporal point process (TPP) provides a powerful framework for analyzing discrete and asynchronous event sequences in continuous time. As its essence, TPP treats events as stochastic points distributing along the time axis, enabling the construction of conditional intensity functions to model the distribution of these sequences [7]. In order to further enhance the fitting ability and flexibility of TPP models, many scholars have introduced deep neural networks into them, forming the so-called neural point process (NPP) models. These models retain the advantages of TPP in processing event sequences, while leveraging the representation power of deep learning to significantly improve the performance in capturing complex event patterns and predicting future events.

However, despite their advancements, current NPP models often overlook the clustering properties of events in both temporal and spatial dimensions, resulting in inadequate performance when confronted with event sequences exhibiting pronounced spatio-temporal patterns. For instance, Fig. 1 delineates the spatio-temporal clustering of event sequences sourced from an urban hotline (*12345*) platform in Northwest China. Among them, (a) displays the spatial distribution of call locations of the hotline within six months, where each point represents a grid area (the smallest management unit in

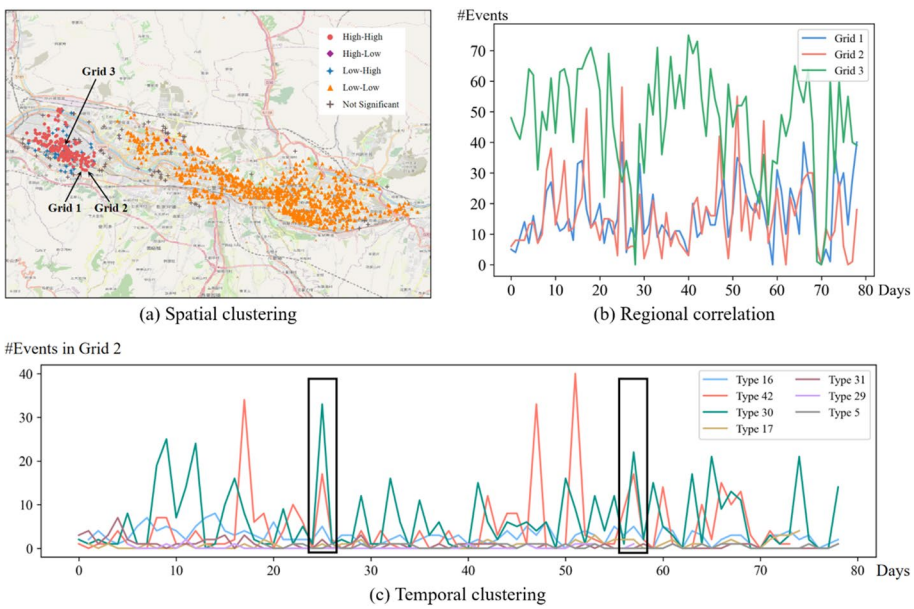


Fig. 1 Spatio-temporal clustering analysis in event sequences

the city). Moran's I spatial autocorrelation analysis of call volumes reveals dominant high-value (High–High) and low-value (Low–Low) clustering patterns, confirming significant spatial aggregation. (b) compares call volume curves over 80 days in three grid areas (Grid 1, Grid 2, and Grid 3) at different distances. Geographically adjacent Grids 1 and 2 exhibit significantly higher pattern similarity than distant Grid 3, demonstrating temporal persistence of spatial clustering. (c) visualizes call volume fluctuations of different types within Grid 2, revealing that diverse types of events show concentrated occurrences over time, as indicated by the black box. This reflects the clustering of call events in the temporal dimension. Notably, the above clustering considerations primarily revolve around a broader geographical regions of event incidence, rather than precise latitude and longitude coordinates, as practical applications often require regional-level insights.

Concurrently, owing to the inherent organizational structure and administrative divisions in urban settings-such as states, counties and cities-geographic regions exhibit a natural hierarchical structure ranging from coarse to fine granularity. This structure introduces multi-level correlations among events. Existing methods, including graph-based approaches, struggle to convey this hierarchical association effectively, limiting their ability to model the complex spatio-temporal dependencies present in real-world data. Additionally, the diversity of event types and their potential mutual influences, such as one event type triggering or inhibiting another, remain underexplored in current models. These limitations highlight a critical research gap: while hierarchical spatio-temporal clustering and diverse event interactions are fundamental characteristics of many event sequence datasets, they are inadequately addressed by existing modeling frameworks.

To address the challenges of modeling event sequences with spatio-temporal clustering characteristics while capturing the mutual influence among diverse event types, we propose EventFormer, a novel neural point process framework built on the Transformer [8] encoder. Transformers are well-suited for sequence modeling due to their multi-head self-attention mechanism, which dynamically focus on crucial segments of the sequence, thereby facilitating the comprehensive modeling of short-term and long-term dependencies. Building on this strength, we incorporate two specialized attention modules: a Time-oriented Multi-head Attention (TM-MA) module and a Type-oriented Multi-head Attention (TP-MA) module. These modules meticulously capture the dependencies of event occurrence time and types on historical data through calculating attention scores, ensuring a more precise reflection of the dynamic characteristics of event sequence. To thoroughly grasp the hierarchical spatial relationships among events, we design a Ladder Attention mechanism, systematically integrating spatial information layer by layer. Moreover, we introduce a trainable event type influence matrix to explicitly model the excitation or inhibition influences between different event types, thereby enhancing interpretability and enabling more accurate predictions of future events.

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

- We propose a novel framework designed for modeling event sequences, which adeptly captures the intricate dynamics of events while enhancing the representation of spatio-temporal clustering effects. Notably, this framework is the first to address

point process modeling from a regional perspective, making it particularly suitable for real-world applications with hierarchical spatial properties.

- We introduce a hierarchical attention mechanism, termed Ladder Attention, which assimilates region information across varying levels of granularity. This mechanism enables the model to foster connections between related events at different spatial scales, addressing the limitations of existing methods in capturing hierarchical spatial correlations.
- We design a type-aware conditional intensity function that explicitly quantifies the positive or negative impacts of a particular type of event on both its homogeneous events and heterogeneous events. Through modeling these dependencies, our framework delicately captures the complex interactions between different event types, improving both interpretability and predictive performance.
- We evaluate our model on both empirical and public datasets. The experimental results on the empirical dataset unequivocally demonstrate the superior capability of our model in terms of event distribution modeling and event prediction accuracy, with at least 1.5% (RMSE) and 1.4% (ACC) improvement over existing baselines. Additional experiments on public datasets further validate the model's ability to handle real-world event sequences effectively.

Related work

Temporal Point Process (TPP) plays a pivotal role as a mathematical instrument in the realm of statistics, capable of modeling asynchronous event data generated in real-world scenarios. Its crux lies in the formulation of the conditional intensity function, where the likelihood of events occurrence is encapsulated. Common TPP models encompass the Poisson process [9], which includes both homogeneous and non-homogeneous variants, as well as the Hawkes process [10], also dubbed the self-exciting process. The homogeneous Poisson process assumes that the intensity function is constant and independent of historical events and time, while the non-homogeneous Poisson process considers the intensity function value to vary over time. In particular, the Hawkes process has garnered considered scholarly attention owing to its inclusion of historical events [11–14]. It posits that past events exert an excitatory influence on future occurrences, with this excitation diminishing exponentially over time while accumulating in an additive fashion. The spatio-temporal clustering characteristics observed in the aforementioned hotline event data imply self-exciting properties in event occurrences, which aligns with the core assumptions of the Hawkes process. However, the predefined intensity function forms in TPP models imposes limitations on the model's expressiveness, hindering its ability to fully comprehend the intricate dynamics of event sequences.

Neural point process (NPP) represents an integration of temporal point process, typically Hawkes process, with deep neural network architectures. Leveraging the exceptional nonlinear approximation capability of deep neural networks, NPP models exhibit greater flexibility and adaptability, enabling them to effectively handle complex and variable event sequence data in real-world scenarios. Recently, NPP models have demonstrated significant application potential in multiple fields, including but not limited to human behavior prediction [15–17], causal inference [18, 19], and document topic

modeling [20, 21]. In the realm of event prediction under investigation in this study, numerous research efforts have been made that provide rich and diverse tools and unique perspectives for precise modeling and prediction of event sequences. Specifically, many studies have focused on integrating neural networks with TPP to enhance the modeling capabilities of event sequence data. For instance, some works combined Recurrent Neural Networks (RNNs) with the Hawkes process, leveraging RNNs to derive dynamic representations of event sequence data, thereby augmenting the model's expressive capabilities [22–24]. Other works incorporated the self-attention mechanism of Transformers into the Hawkes process, empowering the model with a heightened capacity to precisely capture intricate historical dependencies within event sequences [15, 25, 26]. Moreover, some researchers adopted a generalized feedforward neural network to simulate the intensity function of TPP, further enhancing the model's flexiblity and expressive prowess [27]. In terms of event representation, Yang et al. [28] enhanced the model's temporal contextual awareness by introducing external knowledge (factual databases) to furnish time-varying embeddings for events. Ding et al. [29] implemented a sequential variational autoencoder to capture the sparsity pattern of events, and Shang et al. [30] incorporated a graph structure to facilitate the learning of relational information among various events. These models demonstrate remarkable fitting capabilities, significantly surpassing traditional TPP models in terms of modeling event sequence likelihood and predicting future events. Despite this, the aforementioned methods generally overlook the inherent spatio-temporal clustering characteristics of real-world events. This oversight may seriously hampers their predictive ability.

Meanwhile, some studies attempted to model the spatio-temporal distribution of event occurrences within NPP framework. For example, Shchur et al. [31] proposed a hybrid model that integrates TPP with Continuous Normalizing Flows (CNFs), while Chen et al. [32] adopted a similar approach to model the spatio-temporal patterns of event sequences. Zhou et al. [33] utilized amortized variational inference to construct spatio-temporal intensity functions, and Yuan et al. [34] employed diffusion models to learn the spatio-temporal joint distribution of event sequences. Nevertheless, these efforts primarily focus on accurately modeling the time and location of event occurrences, disregarding the crucial information regarding event types. Event types not only directly reflect the essential attributes of events, but also imply complex interactions and dependencies between events, which is crucial for understanding the dynamic evolution of event sequences.

In recent years, Transformer-based models have made significant progress in sequence modeling, offering new perspectives for capturing complex temporal and spatial dependencies. For instance, some studies enhanced the modeling capabilities of Transformers by combining them with Convolutional Neural Networks (ConvNets), leveraging the local feature extraction strengths of ConvNets to complement the global context modeling of Transformers [35–37]. Additionally, some studies explored the integration of spectral methods with Transformers, utilizing frequency domain analysis to enhance the model's ability to capture temporal patterns [38, 39]. Despite these advancements, existing Transformer-based models still struggle to address the challenges posed by data with multi-level correlations, such as the hierarchical spatial structure commonly found in real-world event sequences. In the context of temporal modeling, existing approaches can be broadly

categorized into continuous-time models and discrete-time models. Continuous-time models, including those based on spatio-temporal diffusion processes, typically employ stochastic differential equations to capture the dynamic nature of event sequences [40, 41]. These models excel at modeling the continuous evolution of events, but often require substantial computational resources and may face difficulties in incorporating categorical information into their formulations. This limitation hinders their applicability in scenarios characterized by diverse event types and complex interactions. Discrete-time models typically rely on time windows or segments to capture patterns within event sequences [42, 43]. While these models are computationally efficient and easy to implement, they often fail to fully represent the continuity and irregularity of real-world events. In summary, although the above models have advanced the field of sequence modeling, there are still significant challenges in dealing with multi-level associations and inherent features of event sequences.

In contrast to previous research, our work delves into the hierarchical spatio-temporal clustering characteristics of event occurrences while thoroughly accounting for the mutual influence among diverse event types, constructing a more comprehensive and detailed event modeling framework.

Methodology

Preliminary

Problem formulation

Let $\mathcal{S} = \{(t_j, k_j, r_j)\}_{j=1}^{L}$ denote a sequence of discrete events, where each event (t_j, k_j, r_j) is characterized by its occurrence time t_j, event type $k_j \in \{1, 2, \cdots, N_K\}$, and region r_j. Here, L represents the total number of events, and N_K denotes the number of distinct event types. Notably, each event is associated with region-specific information, reflecting the spatial context of its occurrence. Within this framework, our primary objective is to predict the time t_{j+1} and type k_{j+1} of the $(j+1)$-th event, given the preceding j events in the sequence $\mathcal{S}$. Additionally, we undertake the responsibility of assessing the model's capability in accurately capturing the probabilistic nature underlying event sequencing. Key notations used in this paper are summarized in Table 1.

Hawkes process and neural point process

Temporal point process embodies a stochastic process that generates a sequence of discrete events at times $\{t_j\}_{j=1}^{L}$ within a defined observation period. The conditional intensity function represents the probability of an event occurring within an infinitesimal time window $[t, t + dt)$, conditioned on the historical event sequence $H_t = \{t_j \mid t_j < t\}$:

$$\lambda(t \mid H_t)dt = P(\text{one event occurs in } [t, t + dt) \mid H_t), \tag{1}$$

where $\lambda(t \mid H_t)$ is usually written as $\lambda^*(t)$. Notably, various TPP models diverge primarily in the approach to defining conditional intensity functions.

Hawkes process [10] serves as a mathematical framework specifically tailored for modeling self-excitation phenomena. Its intensity function is defined as:

$$\lambda^*(t) = \mu + \sum_{t_j < t} g(t - t_j), \tag{2}$$

Table 1 The descriptions of notations in EventFormer

Notations	Description
$\mathcal{S}$	Event sequence to be analyzed
L	The length of the event sequence
N_K	The number of event types
t_j	The occurrence time of the j-th event
k_j	The type of the j-th event
r_j	The region where the j-th event occurred
τ_j	The inter-event time between t_{j-1} and t_j
B	The number of time bins for quantile-based discretization
R	The number of regional levels
L_A	The number of attention layers
d_m	The feature dimension of the model
d_k	The dimension of the *Query* and *Key* matrices
d_v	The dimension of the *Value* matrix
d_{in}	The dimension of the position-wise feedforward network
N_H	The number of attention heads

where $\mu > 0$ represents the basic intensity (also known as background intensity) of the event, and $g(\cdot)$ is called excitation function, which is usually set as an exponential decay function to describe the phenomenon of the impact of historical events decaying with the increase of time span. The general form of the excitation function is $g(t) = \alpha e^{-\beta t}$ $(\alpha, \beta > 0)$.

Neural point process models skillfully utilize neural networks to extract historical information within event sequences, and subsequently parameterize the intensity function of Hawkes process or its variants based on the historical information. For example, Du et al. [22] defined the conditional intensity function in their NPP model as:

$$\lambda^*(t) = \exp(b^t + \boldsymbol{\nu}^{t\top} \cdot \mathbf{h}_j + \omega^t(t - t_j)), \tag{3}$$

where the exponential function is used to ensure that the intensity of event occurrence is positive, and the vector $\mathbf{h}$ represents the hidden representation of historical events in the sequence. In the formula, the first term is the basic intensity, the second term is the influence of past events, and the third term is the influence of current event j.

The parameter values of Hawkes process are generally estimated by maximizing the log-likelihood function. For a sequence $\mathcal{S}$ over the observation interval $[t_1, t_L]$, its log-likelihood is given by the following equation:

$$ll(\mathcal{S}) = \sum_{j=1}^{L} \log \lambda^*(t_j) - \int_{t_1}^{t_L} \lambda^*(t)dt. \tag{4}$$

The first term is the sum of the log-likelihoods of past events. And the second term is the log-likelihoods of an infinite number of non-events, which is integral, usually approximated using Monte Carlo integration [44].

Attention mechanism

The integration of attention mechanism into neural networks has proven to be a pivotal advancement in information processing [45]. By leveraging this mechanism, neural networks can automatically prioritize and focus on pertinent details within the input data, thereby enhancing model performance and bolstering ability to generalize. Presently, attention mechanism is ubiquitously employed across a spectrum of deep learning tasks [46–49], underscoring its indispensable role in modern computational frameworks.

In the Seq2Seq model with attention mechanism, elements of the input sequence can be conceptualized as a sequence of $< Key, Value >$ pairs. When dealing with a specific element *Query* within the target sequence, the attention mechanism assesses the correlation between *Query* and various *Keys* in the input sequence to determine attention weight coefficients. Then, these weights are applied to the corresponding *Values*, which are then aggregated to yield the final attention value. In the Transformer [8] architecture, the attention calculation follows a defined formula represented as:

$$\mathbf{O}_{\text{Att}} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}}\right)\mathbf{V}, \tag{5}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ stand emblematic as the *Query*, *Key*, and *Value* matrices, constituting fundamental components pivotal in the computation of attention value. Central to this process is the scaled dot-product method, which is employed to gauge correlation scores between $\mathbf{Q}$ and $\mathbf{K}$. Notably, the dimensionality of $\mathbf{K}$, denoted as d_k, holds significance in this context, shaping the intricacies of the attention computation.

Model structure

The model structure outlined in Fig. 2 delineates the architecture we have devised. Beginning with the sequence $\mathcal{S}$, our process initiates by encoding event information via embedding layers, thereby enabling the transformation of historical data into a feature

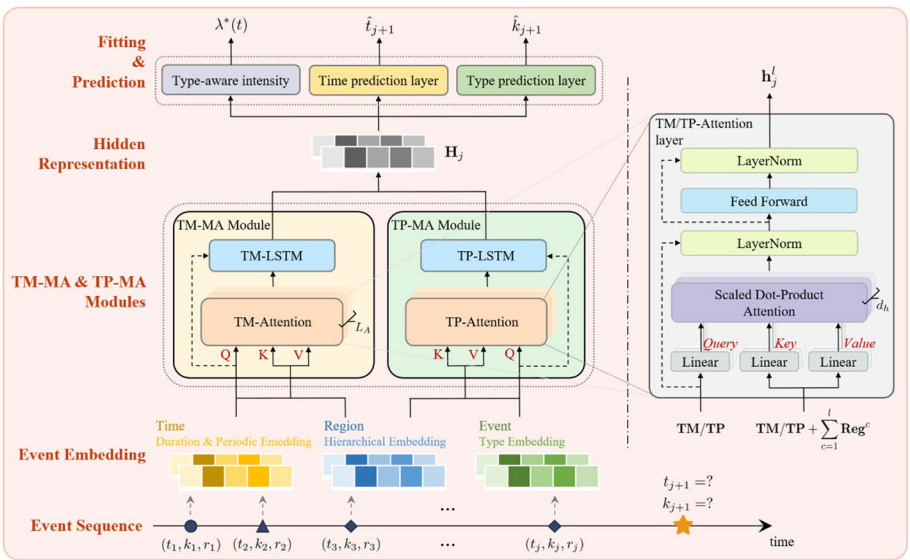


Fig. 2 The architecture of EventFormer

space. These transformed features are fed into the TM-MA and TP-MA modules, each containing L_A attention network layers (TM-Attention and TP-Attention) and an LSTM layer (TM-LSTM and TP-LSTM) to learn the dynamic representations of events. Spatial information is refined layer by layer in the two modules. The modeling and prediction of event sequences rely on the final hidden representations generated by the TM-MA and TP-MA modules. Our training regimen involves joint optimizing the predictive loss and the log-likelihood calculated by the conditional intensity function.

In the subsequent section, we delve into the specifics of our event information embedding methodology, elaborate on the functionality of the TM-MA and TP-MA modules, and expound upon the conditional intensity function employed within our model.

Event embedding

Motivation Event data encompasses rich semantic information, including the timing, type, and location of occurrences, which are essential for understanding the intrinsic relationships between events and predicting future dynamics. However, the asynchronous and discrete nature of such data poses challenges for direct application in machine learning models. To address this, we employ embedding techniques to convert these attributes into continuous vector representations, enabling efficient computation and model processing.

Method The temporal, type, and regional attributes of events represent information from distinct dimensions, necessitating separate embedding processes. Through training the embedding layers, the model is able to precisely capture the similarities and differences among events in the embedding space, thereby gaining deeper insights into the complex patterns within event sequences.

Time embedding: Accurately grasping the temporal patterns of event sequences is crucial. To this end, we adopt a joint representation embedding strategy. On the one hand, given the asynchronous nature of events, in order to accommodate varying time intervals (i.e., inter-event times) while collaboratively handling frequent and infrequent event sequences, we design a quantile-based time binning method. Specifically, we first compute all inter-event times, where the j-th inter-event time is defined as $\tau_j = t_j - t_{j-1}$, representing the interval between the j-th event and its preceding event. From these intervals, we derive B quantile values and construct corresponding time bins. Each inter-event time is then assigned to a bin, and its vector representation is obtained through a trainable embedding matrix $W_b \in \mathbb{R}^{B \times d_m}$. The embedding formula is as follows:

$$\mathbf{TM}_{dur}(\tau_j) = \mathbf{b}_j W_b, \tag{6}$$

where $\mathbf{b}_j$ denotes the one-hot encoding of the bin index for the j-th inter-event time, and d_m is the model dimension. This approach enables precise characterization and efficient processing of event times with unequal intervals, and naturally separate frequent and infrequent sequences in the feature space. Shorter intervals indicate higher temporal clustering of events, which allows the model to easily learn temporal aggregation patterns.

On the other hand, to capture the sequential characteristics of event occurrence times within specific periods while accounting for periodic patterns, we introduce Hour-of-Day (*HoD*) and Day-of-Week (*DoW*) as supplementary features. Mathematically expressed as:

$$\mathbf{TM}_{per}(t_j) = [HoD(t_j), DoW(t_j)]. \tag{7}$$

Here, the $HoD(\cdot)$ function maps the event occurrence time to a specific hour of the day, and the $DoW(\cdot)$ function maps it to a particular day of the week. These features enhance our understanding of temporal dynamics and provide additional temporal information about event clustering, thereby addressing the challenge posed by the frequently changing times of event aggregation. Moreover, the flexibility of our approach allows for the inclusion of other periodic features tailored to the specific characteristics of the data under examination. By combining inter-event time embedding with sequential and periodic features, we generate a comprehensive representation of event occurrence time in the feature space, as depicted by:

$$\mathbf{TM}_j = \text{concat}(\mathbf{TM}_{dur}(\tau_j), \mathbf{TM}_{per}(t_j)). \tag{8}$$

Among them, both $\mathbf{TM}_{dur}(\tau_j)$ and $\mathbf{TM}_{per}(t_j)$ are vectors of dimension d_m. The concatenated vector is then remapped back to the original dimension to obtain the new vector $\mathbf{TM}_j$. This integrated framework facilitates a nuanced analysis of temporal trends and enhances the effectiveness of our modeling approach.

Type embedding: To effectively capture and manipulate type-related features, we utilize an embedding layer to generate a compact yet informative representation of the type information within the feature space. This embedding is achieved through the equation:

$$\mathbf{TP}(k_j) = \mathbf{k}_j W_k, \tag{9}$$

where $\mathbf{k}_j$ denotes the one-hot vector of the type k to which the event j belongs, and $W_k \in \mathbb{R}^{N_K \times d_m}$ represents a learnable type embedding matrix. To streamline our notation, we refer to the type embedding representation $\mathbf{TP}(k_j)$ of event j as $\mathbf{TP}_j$.

Region embedding: In real-world scenarios, a specific geographical location usually belongs to multiple hierarchical regional structures. Figure 3 provides an intuitive example, demonstrating how the location of event A belongs to a four-level hierarchy of "District-Street-Community-Grid" in descending order. To accurately model the spatial clustering effects of event occurrences while improving data processing efficiency, we introduce a novel embedding-centric approach aimed at transforming discrete region data into continuous vector representations, thus facilitating a more thorough capture of inter-region relationships. Specifically, our method entails the conversion of all regions within each hierarchical level into numerical labels based on proximity metrics, where labels are sequentially assigned, commencing from 0. For an event j in an R-level hierarchy, the regional labels are represented as $r_j = \{r_j^1, r_j^2, ..., r_j^R\}$, where the superscript 1 denotes the highest-level region with the broadest coverage. These labels are then individually fed into spatial embedding layers to generate fixed-length vector representations, ensuring effective information conversion and retention. The embedding process at the c-th level is expressed mathematically as:

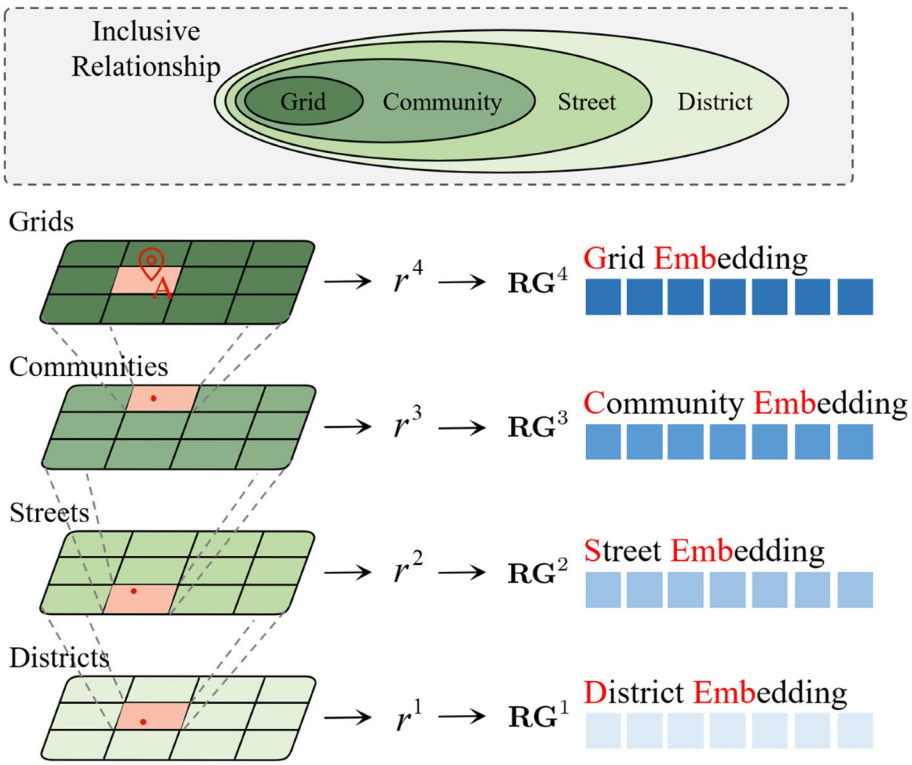


Fig. 3 Region embedding

$$\mathbf{RG}_j^c = \mathbf{r}_j^c W_r^c, \tag{10}$$

where $\mathbf{r}_j^c$ represents the one-hot encoding of r_j^c, and $W_r^c \in \mathbb{R}^{R \times d_m}$ is the learnable region embedding matrix at the c-th level. In this way, event j integrates a set of hierarchical spatial embeddings, denoted as $\{\mathbf{RG}_j^1, \mathbf{RG}_j^2, \cdots, \mathbf{RG}_j^R\}$, which collectively provide detailed spatial features of the event.

Example 1 *Region Embedding*: Figure 3 depicts a four-tier hierarchical regional structure, where the location of event A is labeled as r^1, r^2, r^3, and r^4 across the districts, streets, communities, and grids levels, respectively. These labels are processed through embedding layers to produce region embeddings at each level: $\mathbf{RG}^1$ (District Embedding), $\mathbf{RG}^2$ (Street Embedding), $\mathbf{RG}^3$ (Community Embedding), and $\mathbf{RG}^4$ (Grid Embedding). These embeddings effectively capture the spatial contextual information of event A.

This embedding approach not only simplifies data processing but also enhances the model's expressive capability and flexibility. It empowers the model to discern intricate relationships among regions, moving beyond treating regions as isolated entities. For instance, if two events from different regions share the same parent region, their spatial embeddings may exhibit proximity in vector space, signaling geographical closeness and potentially shared properties or behaviors.

Time-oriented and type-oriented multi-head attention

Motivation The occurrence of events often follows specific temporal patterns and type regularities, which not only affect the generation of individual event, but also determine the evolutionary trend of event sequences. In order to accurately capture these dynamic information, we introduce the time-oriented multi-head attention (TM-MA) and type-oriented multi-head attention (TP-MA) core modules, while considering regional information that provides the spatial context for event occurrences. By intergrating regional information with temporal and type dynamics, we aim to construct a more comprehensive and accurate event sequence model, providing robust support for subsequent prediction and analysis.

Method Within the TM-Attention component of the TM-MA module, the *Query* matrix is derived from time embeddings, enabling the model to attend to and prioritize temporal cues. Concurrently, we fuse the time embeddings with region embeddings to construct the *Key* and *Value* matrices, thereby ensuring that the model incorporates spatial region information during temporal processing. This is represented as follows:

$$\begin{cases} \mathbf{Q}_{TM} = \mathbf{TM} \cdot W_{TM}^{Q} \\ \mathbf{K}_{TM} = (\mathbf{TM} + \mathbf{RG}) \cdot W_{TM}^{K} \\ \mathbf{V}_{TM} = (\mathbf{TM} + \mathbf{RG}) \cdot W_{TM}^{V} \end{cases}, \tag{11}$$

where $\mathbf{TM}$ represents time embeddings, and $W_{TM}^{Q}, W_{TM}^{K} \in \mathbb{R}^{d_m \times d_k}$, $W_{TM}^{V} \in \mathbb{R}^{d_m \times d_v}$ denote the weights of the linear transformation. This architectural design facilitates the model's ability to effectively capture temporal dynamics of events while concomitantly incorporating spatial region information, thereby accurately grasping the spatio-temporal clustering patterns of event occurrences.

Similarly, within the TP-Attention component, we utilize type embeddings to generate the *Query* matrix, and merge type embeddings with region embeddings to construct the *Key* and *Value* matrices, ensuring that the model incorporates region information when capturing type dynamics:

$$\begin{cases} \mathbf{Q}_{TP} = \mathbf{TP} \cdot W_{TP}^{Q} \\ \mathbf{K}_{TP} = (\mathbf{TP} + \mathbf{RG}) \cdot W_{TP}^{K} \\ \mathbf{V}_{TP} = (\mathbf{TP} + \mathbf{RG}) \cdot W_{TP}^{V} \end{cases}, \tag{12}$$

where $\mathbf{TP}$ represents type embeddings, and $W_{TP}^{Q}, W_{TP}^{K} \in \mathbb{R}^{d_m \times d_k}$, $W_{TP}^{V} \in \mathbb{R}^{d_m \times d_v}$.

The computation of attention adheres to the scaled dot-product method of the Transformer architecture, as shown in Eq. (5). By employing the TM-Attention and TP-Attention components, the model is capable of directing its focus towards different aspects of event sequence, thereby facilitating a more comprehensive understanding of events occurrence patterns.

Ladder attention mechanism: To address the hierarchical nature of regional information, we introduce a Ladder Attention mechanism within both the TM-MA and TP-MA modules. This mechanism progressively integrates region embeddings at different hierarchical levels, enabling the model to capture and emphasize relevant spatial details at varying granularities. Specifically, the *Key* and *Value* matrices in the attention layers are dynamically constructed by combining time (or type) embeddings

with region embeddings from coarser to finer levels. For instance, in the l-th attention layer of TM-MA, the *Key* matrix is computed as:

$$\mathbf{K}_{TM}^{l} = (\mathbf{TM} + \sum_{c=1}^{l} \mathbf{RG}^{c}) \cdot W_{TM}^{K}, \tag{13}$$

where $\sum_{c=1}^{l} \mathbf{RG}^{c}$ denotes the sum of region embeddings from the first to the l-th hierarchical level. The same formulation applies to $\mathbf{V}_{TM}^{l}$, $\mathbf{K}_{TP}^{l}$ and $\mathbf{V}_{TP}^{l}$.

Example 2 *Ladder Attention Mechanism*: As illustrated in Fig. 4, Event A's region embeddings span four hierarchical levels (district, street, community, and grid). In the first layer of TM-Attention component, the *Key* and *Value* matrices are derived from the fusion of time embeddings and district-level region embeddings of historical events. In subsequent layers, these matrices progressively incorporate finer-grained region embeddings (e.g., street, community, and grid) while retaining coarser-level information. The TP-Attention component is analogous to the TM-Attention component, except that time embeddings are replaced with type embeddings.

Under this mechanism, the number of attention layers L_A matches the number of region levels R, so that spatial information at all granularities can be effectively learned. This progressive integration strategy enhances the model's ability to understand the hierarchical spatial context of events, which is pivotal for predicting events

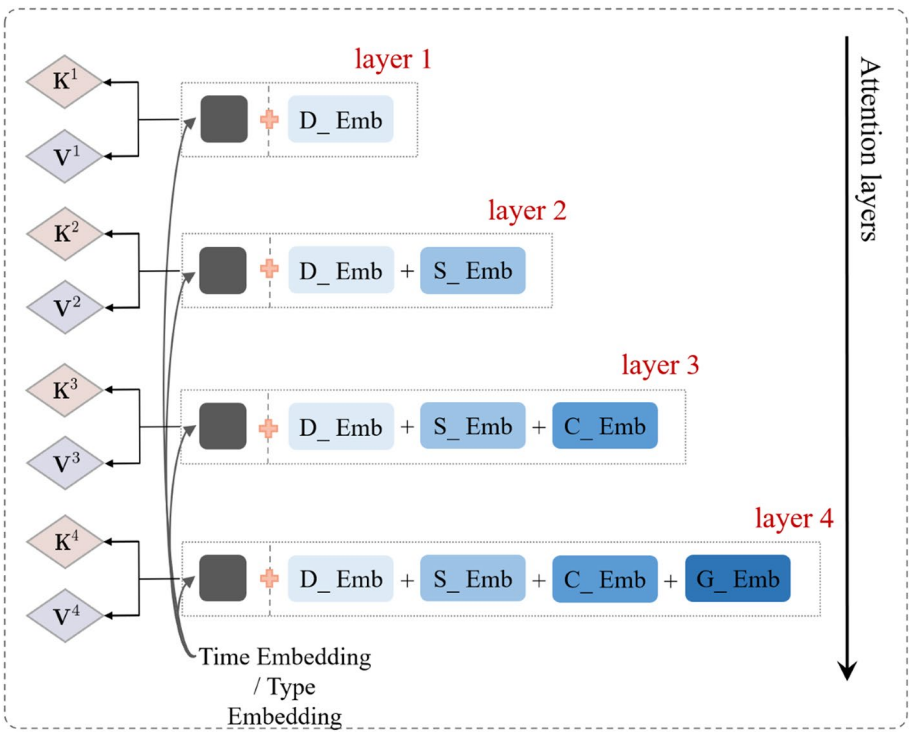


Fig. 4 Ladder attention mechanism

with multi-level spatial correlations. Furthermore, it improves interpretability by explicity modeling the influence of regions at different scales.

In our framework, we employ a multi-head attention mechanism to enrich the model's adaptability, with the number of heads denoted by N_H. After computing the attention output $\mathbf{O}_{\text{Att}}$, we pass it through a position-wise feedforward network (FFN) to further extract local features from the event sequence. The hidden representations of the TM-Attention and TP-Attention components are then obtained as follows:

$$\mathbf{h}_j = \text{Relu}(\mathbf{O}_{\text{Att}} \cdot W^{F1} + b^{F1}) \cdot W^{F2} + b^{F2}, \tag{14}$$

where $W^{F1} \in \mathbb{R}^{d_m \times d_{in}}$, $W^{F2} \in \mathbb{R}^{d_{in} \times d_m}$, $b^{F1} \in \mathbb{R}^{d_{in}}$ and $b^{F2} \in \mathbb{R}^{d_m}$ are parameters of the neural network, and d_{in} stands as the dimensionality of FFN.

These hidden representations are aggregated across layers within their respective components and combined with the original input via residual connections. The integrated outputs are then fed into the corresponding TM-LSTM and TP-LSTM components. Through concatenating the outputs of the two LSTM layers and reshaping the dimensions, we derive the ultimate hidden representation $\mathbf{H}_j \in \mathbb{R}^{1 \times d_m}$ as defined by the equation set:

$$\begin{cases} \mathbf{H}_{TM,j} = \text{LSTM}(\sum_{i=1}^{L_A} \mathbf{h}_{TM,j}^i + \mathbf{TM}_j) \\ \mathbf{H}_{TP,j} = \text{LSTM}(\sum_{i=1}^{L_A} \mathbf{h}_{TP,j}^i + \mathbf{TP}_j) \\ \mathbf{H}_j = \text{concat}(\mathbf{H}_{TM,j}, \mathbf{H}_{TP,j}) \end{cases} . \tag{15}$$

The utilization of LSTM has been demonstrated to facilitate the capturing of sequential dependencies with sequential data [50]. Our motivation for using two separate LSTMs is the distinct characteristics of temporal and event type information. By decoupling these two dimensions, TM-LSTM can focus exclusively on capturing temporal dynamics, such as irregular intervals and long-term dependencies, while TP-LSTM specializes in learning the complex interactions and dependencies among event types. Additionally, we incorporate masking in the attention modules to ensure predictions are based solely on historical data.

Type-aware conditional intensity function

Motivation In constructing models for event sequence modeling and prediction, we face the challenge of precisely capturing and simulating the complex dependencies among events. Especially when dealing with a diverse set of event types, these types not only possess their unique occurrence patterns but also exhibit intricate interdependencies among each other. To explicitly model the basic intensities of various types of events, consider the cumulative effects of historical events and the immediate effects of current events, while also incorporating the mutual constraints and promotional relationships between types, we design a novel type-aware conditional intensity function.

Method In our model, we define the formula for the conditional intensity function as follows:

$$\lambda^*(t) = \sum_{k=1}^{N_K} \lambda_k^*(t) = \sum_{k=1}^{N_K} f_k(b_k + \mathbf{H}_j \mathbf{w}_k^\top \mathbf{M}_{EI} + \gamma(t - t_j)), \tag{16}$$

$$f_k(x) = s_k \log(1 + \exp(x/s_k)). \tag{17}$$

We establish separate conditional intensities for each event type and employ the softplus function $f_k(\cdot)$ to ensure positive intensity, where s_k denotes the scale parameter for event type k. In the formula, the first term b_k represents the inherent basic intensity of each type, the H_j in the second term represents the integrated hidden representation from all preceding events (1 to j), the $\mathbf{w}_k \in \mathbb{R}^{N_K \times d_m}$ serves as the weight matrix for linear transformation, the t_j in the third term denotes the occurrence time of the j-th event, and the γ is a scaling factor. These three components in the formula work together to achieve more precise intensity calculations. Notably, we introduce a trainable **E**vent type **I**nfluence matrix $\mathbf{M}_{EI} \in \mathbb{R}^{N_K \times N_K}$ to explicitly capture the interplay between different event types, whether they exhibit excitatory or inhibitory effects. The value at position (i, j) in $\mathbf{M}_{EI}$ indicates the influence strength of type i events on type j events. The diagonal elements of $\mathbf{M}_{EI}$ represent the event type's influence on its own recurrence, while off-diagonal elements signify mutual influences between distinct event types. Positive values denote excitatory effects, while negative values indicate disincentive effects.

Prediction and optimization

Following the acquisition of the final hidden representation $\mathbf{H}_j$, EventFormer employs two linear layers to predict both the occurrence time and type of subsequent events independently. The prediction formula is:

$$\begin{cases} \hat{t}_{j+1} = \mathbf{H}_j W_{time} \\ \hat{k}_{j+1} = \underset{k}{\text{argmax}}(\hat{\mathbf{k}}_{j+1}(k)), \hat{\mathbf{k}}_{j+1} = \text{Softmax}(\mathbf{H}_j W_{type}) \end{cases}, \tag{18}$$

where $W_{time} \in \mathbb{R}^{d_m \times 1}$, $W_{type} \in \mathbb{R}^{d_m \times N_K}$, and $\hat{\mathbf{k}}_{j+1}(k)$ represents the k-th element within the vector $\hat{\mathbf{k}}_{j+1}$. We initiate our predictions from the second event within the sequence, facilitating a more meticulous scrutiny of the model's predictive accuracy.

To minimize prediction errors and enhance the model's ability to effectively capture the distribution of event sequences, we adopt a joint training strategy. The loss function is defined as the negative log-likelihood $ll(\mathcal{S})$ loss plus the losses associated with time and type predictions:

$$Loss(\mathcal{S}) = -ll(\mathcal{S}) + \alpha \cdot loss_{time}(\mathcal{S}) + loss_{type}(\mathcal{S}). \tag{19}$$

Specifically, square error gauges the loss of time prediction, while cross-entropy loss quantifies the loss in type prediction. The weight α balances the relative scales between the time prediction loss and other terms, ensuring stable optimization. For sequence $\mathcal{S}$:

$$loss_{time}(\mathcal{S}) = \sum_{j=2}^{L} (t_j - \hat{t}_j)^2, \; loss_{type}(\mathcal{S}) = -\sum_{j=2}^{L} \mathbf{k}_j^\top \log(\hat{\mathbf{k}}_j). \tag{20}$$

These components are represented mathematically, where $\mathbf{k}_j$ denotes the one-hot encoding for ground-truth type of event j, and $\hat{\mathbf{k}}_j$ signifies the output from the type prediction layer. The model optimization is facilitated by the Adam optimizer.

Experiment

Datasets

In this study, we utilize two types of datasets: an empirical dataset and three public datasets. The empirical dataset comprises call records from the urban hotline, including information regarding the time, type, and location of the calls. The publicly available datasets include Financial Transactions [22], COVID-19 [32], and Citibike [32], which offer extensive external references for us to conduct fair and open comparative analysis and model validation. Table 2 shows some statistical information of these datasets.

Empirical dataset

Government Service Convenience Hotline (GSCH): Utilizing data from the public service platform in Lanzhou, Gansu Province, China, we undertook an analysis to evaluate the efficacy of our model. The platform functions as a channel for addressing non-emergency appeals from both enterprises and citizens, encompassing a wide range of inquiries, assistance requests, complaints and reports across various domains. We consider each hotline call as an event, and the call records contain specific timestamp (event time), classification of appeals (event type), and the caller's geographical area (event region), providing us with crucial clues to gain deeper insights into the characteristics and geographical distribution of the events. Our analysis covers a six-month period in 2016, resulting in a carefully cleaned dataset comprising approximately 510,000 events.

Within the GSCH dataset, a comprehensive array of 48 event types is cataloged, with an unequal distribution in their quantities, encompassing a broad spectrum of societal concerns. These events span various domains, ranging from pressing environmental issues such as air pollution, solid waste pollution, and water environment pollution to matters concerning urban infrastructure like landscaping, municipal public facilities, and road traffic facilities. Additionally, there are a series of challenges pertaining to urban management affairs, such as advertising promotion, street order maintenance, and construction oversight. Notably, these events are contextualized within different geographical regions, with granularity ranging from districts to streets, communities, and ultimately, the smallest management unit—the grids. This meticulous classification

Table 2 Dataset statistics

Dataset	#Sequences	Avg.length	#Types	Spatial information
GSCH	7156	71	48	Region information
Financial	200	2074	2	–
COVID-19	1650	98	–	Latitude & longitude
Citibike	3060	135	–	Latitude & longitude

#Sequences represents the number of event sequences, Avg.length represents the average length of event sequences, #Types represents the number of event types, and Spatial information represents the form of spatial information."–" indicates that the dataset does not have this information

enables a nuanced understanding of localized challenges and facilitates targeted interventions for effective management and resolution.

In light of the pronounced spatio-temporal clustering exhibited by the hotline data, as depicted in Fig. 1, our initial step involved implementing spatial pre-clustering procedures before compiling the dataset. This involved randomly generating event sequences on a grid basis, ensuring their chronological order, with each sequence limited to a maximum length of 500 events transpiring within the same grid. Adhering to established methodologies from prior studies, we partitioned the dataset into training, testing, and validation subset, maintaining in an 8:1:1 ratio, thereby facilitating robust model training and evaluation.

Public datasets

Financial transactions serves as a pivotal resource frequently employed for the evaluation of temporal point process models. It encompasses the high-frequency trading activities of a particular stock throughout a single day. Each record contains the time and action (buy or sell) taken during the transaction. The dataset is a single long sequence, while the event sequence is further divided through timestamps.

COVID-19 is commonly used to evaluate neural point process models for modeling spatio-temporal distribution. It comes from the daily case data of COVID-19 in New Jersey published by the New York Times (2020). Each data in the sequence records the time and location information of the occurrence of the case. The data is summarized at the county level.

Citibike is collected by a bike-sharing service company in New York City. Each data in the sequence represents the start of a trip and includes the start time and location information of the trip. It is also often utilized in academic research as a significant resource for evaluating neural point process models that model spatio-temporal distribution.

Evaluation metrics

To evaluate the performance of our event sequence modeling framework, we adopt the Negative Log-Likelihood (NLL) as the primary metric. NLL measures the alignment between the model's predicted probability distribution and the true distribution of event sequences, making it a natural choice for assessing the fitting capability of neural point process models. A lower NLL value indicates a closer match between the model's predictions and the ground truth, reflecting better modeling performance.

For event prediction tasks, we utilize three complementary metrics: Root Mean Square Error (RMSE), Accuracy (ACC), and Area Under the Curve (AUC). RMSE quantifies the accuracy of time predictions by measuring the deviation between predicted and actual event timestamps. This metric is particularly suitable for evaluating temporal precision in event sequences, as it penalizes large error more heavily, aligning with the need for precise event timing in real-world applications.

To assess event type prediction, we employ both ACC and AUC. ACC provides a global measure of the model's performance by calculating the proportion of correctly predicted events types. While intuitive and widely used, ACC can be influenced by imbalanced event type distributions in the dataset. To address this limitation, we also use AUC, which evaluates the model's performance across different classification

thresholds by calculating the area under the ROC curve. Specifically, we treat each event type as a binary classification problem, calculate the AUC for each type, and then average the results to obtain an overall AUC score. This approach provides a more robust and nuanced understanding of the model's ability to distinguish between different event types.

By combining these metrics, we comprehensively evaluate the model's performance in both event sequence modeling and event prediction. This multi-faceted evaluation ensures that our model's strength and limitations are thoroughly analyzed in the context of the specific challenges posed by event sequence data.

Baselines

To comprehensively evaluate the performance of EventFormer, we compare it against a diverse set of baseline models, including both classical and state-of-the-art approaches. Specifically, we select six representative NPP models: RMTPP [22], NHP [24], SAHP [25], THP [26], ANPP [15], and A-NHP [28]. In addition to these NPP models, we include two specialized models designed for spatio-temporal joint distribution modeling: DeepSTPP [33] and DSTPP [34]. These two models leverage fine-grained latitude and longitude information alongside timestamps to capture complex spatio-temporal dependencies of event occurrence, providing a robust benchmark for evaluating EventFormer's ability to handle spatial and temporal clustering. The neural network architecture used by all baseline models is shown in Table 3.

Experimental details

The experiments conducted herein were performed using a robust computational setup featuring the NVIDIA GeForce RTX 3080 GPU. The implementation framework utilized CUDA 11.3, Python 3.7, and PyTorch 1.10, ensuring compatibility and efficiency throughout the process.

Noteworthy parameters were established for model training, including a learning rate is initialized to 1×10^{-4}, the batch size set to 4, $B = 20$, $L_A = R = N_H = 4$, $d_m = 64$, $\alpha = 100$, and the number of training epochs set to 50. These selections were carefully made to afford the model ample opportunities for learning and convergence. Furthermore, to safeguard against overfitting, a dropout rate of 0.1 was employed, contributing to the model's generalization capabilities. The parameters of the baseline models are set as suggested in their papers. To avoid random errors, the experimental results show the average of five independent runs.

Table 3 The neural network architecture used by baselines

	RMTPP	NHP	SAHP	THP	ANPP	A-NHP	DeepSTPP	DSTPP
RNN-based	✓	✓						
Transformer-based			✓	✓	✓	✓	✓	✓
Diffusion-based								✓

Table 4 Comparison of prediction results for event occurrence time and type

Model	Predictive metrics		
	RMSE↓	ACC↑	AUC↑
RMTPP	12.50	77.20	0.60
NHP	12.69	79.29	–
SAHP	15.78	77.79	0.40
THP	12.41	79.09	0.83
ANPP	12.63	79.60	0.89
A-NHP	15.85	78.62	0.53
DSTPP	12.66	–	–
EventFormer	12.23	80.71	0.86

Results and discussion

In this section, we conduct a detailed comparison and analysis of the performance of EventFormer and baseline models on the GSCH dataset. Special attention is paid to the contributions of the internal components within EventFormer, and how these components work together to optimize the overall model performance. Furthermore, to verify the universality and effectiveness of EventFormer, we also conduct validation experiments on public datasets. It is noteworthy that, in comparing EventFormer with baseline models, we only compare the experimental results based on the same methodology and computational metrics to ensure the fairness and rigor of the experiments.

Comparison of event prediction on the GSCH dataset

Our assessment focuses on the efficacy of EventFormer in predicting forthcoming events utilizing historical data from the GSCH dataset. Table 4 details the performance comparison between EventFormer and a series of baselines with the ability to predict event occurrence time and type across key metrics.[1] The results indicate that EventFormer achieves competitive performance across all metrics, demonstrating its effectiveness in handling complex event sequence prediction tasks. In terms of RMSE and ACC, EventFormer outperforms all baseline models, with 1.5% and 1.4% improvement over the best-performing baseline respectively. This highlights its superior capability in accurately predicting event times and correctly classifying event types. In the AUC metric, EventFormer ranks second, slightly behind ANPP, but still demonstrates robust performance in distinguishing between positive and negative samples. Notably, compared to the state-of-the-art spatio-temporal model DSTPP, EventFormer achieves a 3.4% improvement in RMSE, underscoring its effectiveness in temporal prediction tasks.

EventFormer demonstrates robust performance across all predictive metrics, indicating its high suitability for event sequence prediction tasks characterized by spatio-temporal clustering and multi-type interactions. The performance gains can be attributed to the novel architecture of EventFormer, particularly the TM-MA and TP-MA core modules, which effectively capture temporal and type-specific dependencies. These advantages mean that EventFormer can provide more accurate prediction and classification

[1] The DeepSTPP model do not predict event sequences, while the DSTPP model predicts time but not event types. For the NHP model, the AUC metric could not be obtained due to technical limitations in the outdated codebase.

results in practical applications, which is of great significance for scenarios requiring high accuracy and reliability, such as financial risk control and medical diagnosis.

Comparison of likelihood on the GSCH dataset

In the research on the ability to model the probability distribution of event sequences within the context of the GSCH dataset, we delve into a comparative analysis between EventFormer and several baseline models that also calculate the NLL metric. Figure 5 depicts the trajectory of NLL values for THP, A-NHP, DeepSTPP, and EventFormer throughout the entire training process. Observing these curves, we can discern that the curve of EventFormer exhibits a more stable and pronounced downward trend, ultimately achieving the lowest value among all the models. This significant result clearly demonstrates that compared to baseline models, EventFormer exhibits higher efficiency and accuracy in modeling the intricate dynamics of event sequence distributions.

Mutual influence between different types

In EventFormer, we introduce a type-aware conditional intensity function that incorporates a trainable matrix $\mathbf{M}_{EI}$ to capture the excitatory or inhibitory effects among diverse event types. To visualize these interactions, we present $\mathbf{M}_{EI}$ as a heatmap, as in Fig. 6.

The heatmap reveals several key patterns: values along the diagonal are close to 1, indicating strong self-excitation for each event type, while off-diagonal values range from −0.15 to 0.27, reflecting both positive (excitation) and negative (inhibition) influences among event types. These results suggest that the model successfully captures the complex interdependencies among event types. For instance, the row indicated by the black triangle corresponds to the category of urban environment. Events in this category significantly inhibit the occurrence of events in other categories, which aligns with real-world scenarios where effective urban environment management can reduce issues such as street disorder, solid waste pollution, and construction mismanagement. Conversely, the column indicated by the red triangle corresponds to the category of legal and litigation-related. The occurrence of events in this category is positively influenced by other

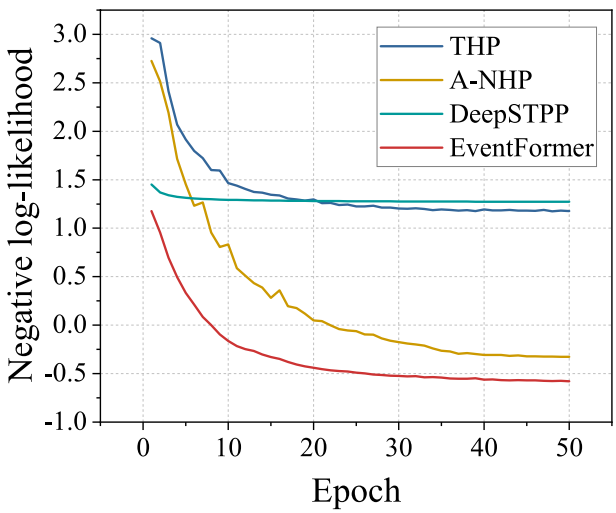


Fig. 5 Training curves

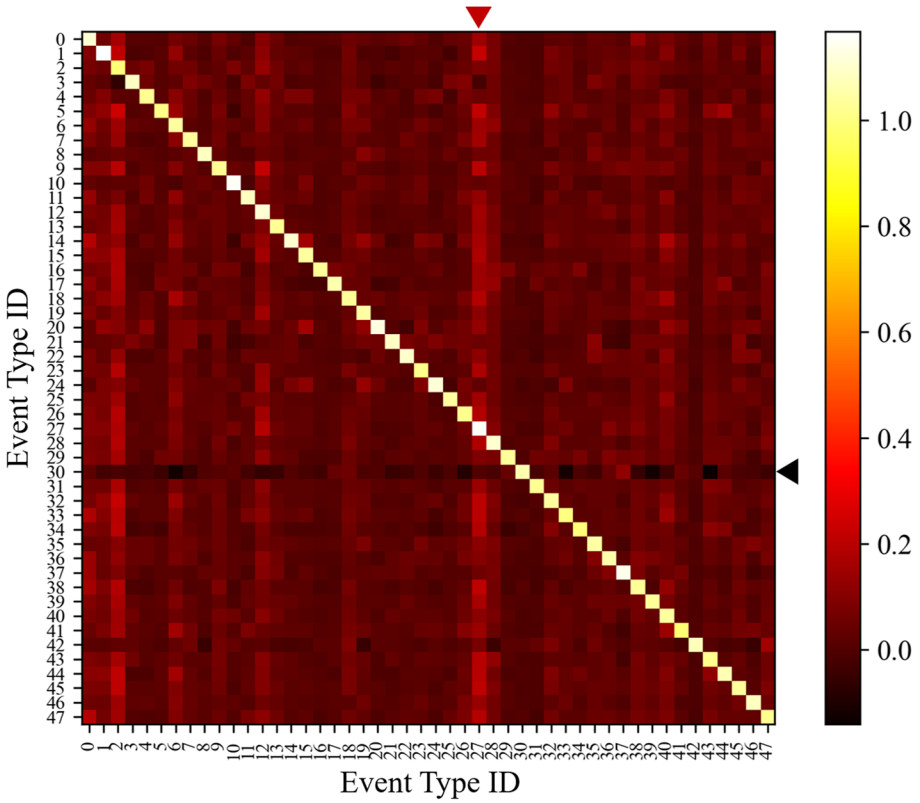


Fig. 6 Heatmap of event type influence matrix $\mathbf{M}_{EI}$. Both axes represent event type IDs, with tick marks ranging from 0 to 47 corresponding to the 48 event types in the GSCH dataset

event types, reflecting the tendency for unresolved issues in other categories to escalate into legal disputes as citizens seek redress. These findings highlight the complexity of urban operations and management, as captured by the government service hotline data.

Ablation study

In order to assess the efficacy of the Ladder Attention mechanism as well as the type-aware conditional intensity function within our model, we undertake experiments examining various variants of EventFormer on the GSCH dataset. Notably, the variant **w/o R-info** signifies the exclusion of **R**egional **info**rmation, in which case the Ladder Attention mechanism degrades to a general self-attention computation, with the *Key* and *Value* matrices being identical to the *Query* matrix; **w/ Di-emb** represents **Di**rectly **emb**edding all regional information into the model through an additive manner in the first layer of the attention network, rather than hierarchical embedding; while **w/ G-cif** represents the use of a **G**eneral **c**onditional **i**ntensity **f**unction as shown in Equation (3), rather than the type-aware conditional intensity function. These experiments are crucial in elucidating the impact of different configurations on the overall performance and functionality of the model.

Table 5 presents the experimental results pertaining to the variants of EventFormer. Notably, even without incorporating regional information, our model outperforms other NPP baseline models. This achievement underscores the efficacy of our

Table 5 Comparison of variants of EventFormer

Model	Metrics			
	NLL↓	RMSE↓	ACC↑	AUC↑
w/o R-info	−0.50	12.29	80.45	0.83
w/ Di-emb	−0.51	12.28	80.55	0.83
w/ G-cif	−1.75	12.27	80.62	0.85
EventFormer	−0.58	12.23	80.71	0.86

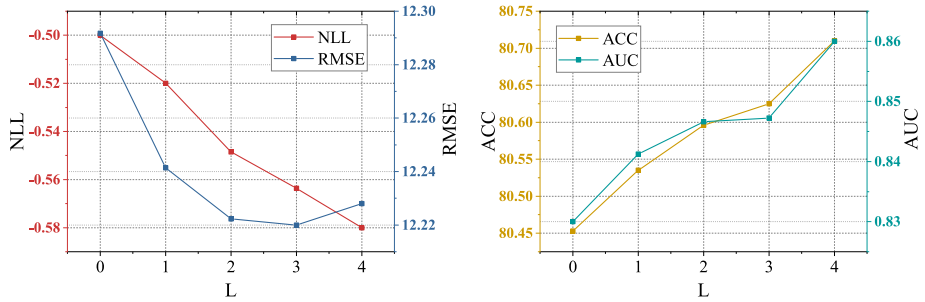


Fig. 7 Results of incorporating regional information at different granularities

approach in leveraging temporal and categorical information of events to improve prediction accuracy and reliability. Meanwhile, our hierarchical embedding approach proves superior to directly embedding regional information. By systematically integrating spatial information across multiple hierarchical levels, this approach enhances the model's ability to capture complex spatial dependencies, leading to improved experimental outcomes. Moreover, compared to the variant that uses a general intensity function, our model achieves higher performance in predictive metrics. This suggests that our designed intensity function is more effective at modeling the interactions and dependencies among diverse event types.

Furthermore, we also investigate the impact of incorporating regional information of different granularities on the model's performance. In our model, as the number of attention network layers increases, the granularity of the regional information embedded in the model gradually refines. The GSCH dataset contains four levels of regional information. When $L = 1$, the model integrates only the first-level (coarsest) regional information using a single attention layer. When $L = 4$, the model incorporates all four levels of regional information, enabling a comprehensive understanding of hierarchical spatial structures. It is important to clarify that $L = 0$ does not represent zero attention layers but rather the complete absence of regional information, corresponding to the w/o R-info variant discussed earlier. Figure 7 presents the experimental results of the model under different metrics with L varies from 0 to 4. As the number of layers increases, the model gradually incorporates finer-grained regional information, leading to a gradual improvement in various metrics. This adequately demonstrates that EventFormer can effectively utilize hierarchical region information, revealing its capability to handle data with multi-level spatial features.

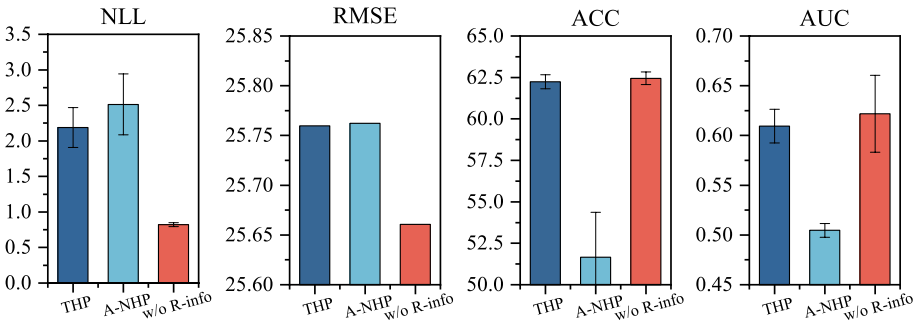


Fig. 8 Results in financial transactions

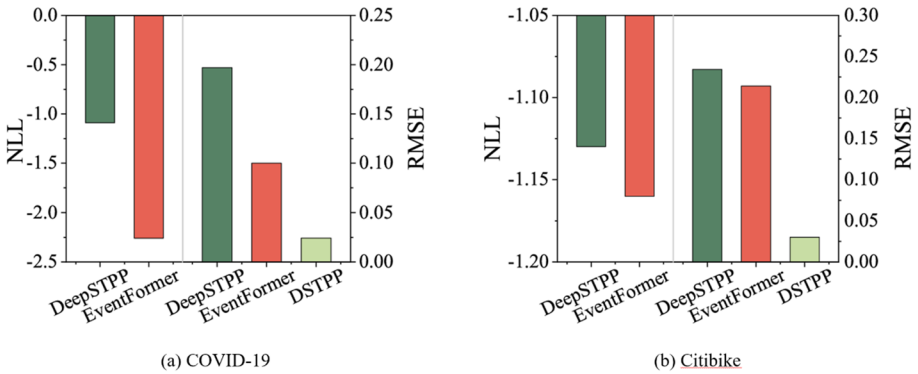


Fig. 9 Results in COVID-19 and Citibike

Validation experiments on public datasets

In our pursuit to rigorously validate the efficacy of EventFormer, we undertake experiments utilizing publicly available datasets. For the Financial Transactions dataset, which contains only time and type information, we opt to employ the variant of EventFormer without regional information (w/o R-info) for experimental validation. The findings of these experiments are presented in Fig. 8, where the dataset is sampled five times to produce different training, validation, and testing sets. We perform five independent experiments for THP, A-NHP, and EventFormer (w/o R-info), with error bars generated to reflect variability. These results show that EventFormer (w/o R-info) achieves competitive performance, surpassing THP and A-NHP in both the fitting metric (NLL) and predictive metrics (RMSE, ACC, AUC). This outcome serves as a compelling validation of the robustness and superiority of EventFormer, demonstrating its adaptability to diverse datasets.

For the COVID-19 and Citibike datasets, which consist time and location information (longitude and latitude), preprocessing is essential to adapt them for use with EventFormer. Specifically, we partition the location data into four hierarchical levels utilizing a series of 2x2 grid divisions, encoding the regions to which events belong. All events are assigned a uniform event type (indicated by 1). In Fig. 9, we compare the performance[2]

[2] Given the distinct NLL calculation in DSTPP model versus our approach and other baselines, we limit comparison to RMSE for consistency. The results of DeepSTPP on COVID-19 and Citibike were referenced from the DSTPP article.

of EventFormer with two baseline models (DeepSTPP and DSTPP) that specialize in spatio-temporal distribution modeling. The experimental data shows that EventFormer significantly outperforms DeepSTPP in terms of the NLL metric, while its performance in the RMSE metric falls between DeepSTPP and DSTPP. Despite utilizing only coarse regional information, EventFormer still achieves comparable or even superior results compared to the two baseline models, which rely on precise latitude and longitude data. This highlights EventFormer's ability to effectively capture spatio-temporal dependencies of event occurrence, even with less granular spatial information.

Conclusion

This paper introduces EventFormer, a robust framework tailored specifically for the analysis of event sequences characterized by spatio-temporal clustering and multiple types. Notably, it stands as the first work to predict point processes from a regional perspective. Our model incorporates two attention modules designed to adeptly capture the intricate dynamics inherent in event occurrence. Among them, we employ a layer-by-layer refinement embedding technique, enhancing the model's capacity to glean spatial information from a regional standpoint with heightened efficacy. Furthermore, a type-aware conditional intensity function is designed to accurately model the dependencies of event occurrence. Extensive experiments conducted on real-world datasets underscore the prowess of EventFormer, highlighting its capability to achieve state-of-the-art performance in both event likelihood modeling and prediction accuracy.

Although EventFormer can effectively utilize hierarchical region information, its performance depends on the granularity of region partitioning. Coarse partitions may fail to capture fine-grained spatial dependencies, while overly fine partitions can increase computational complexity. EventFormer demonstrates moderate execution time performance. Subsequent work will explore more flexible region partitioning methods as well as event-type-specific sparse computation patterns to improve efficiency. In addition, integrating more external factors (e.g., weather, socioeconomic data) may help better predict future events, which is a promising direction for future research. Looking ahead, we anticipate leveraging EventFormer in Earth natural system studies, where it will encounter more complex spatio-temporal dynamics and regional information, potentially providing novel insights for fields such as climate science and disaster prediction.

Author contributions

S.Y.: Data curation, methodology, writing—original draft, writing—review \& editing. D.G.: supervision, methodology, writing—review \& editing. Y.F.: supervision, methodology, writing—review \& editing. W.J.: formal analysis, methodology, writing—review \& editing. All authors reviewed the manuscript.

Funding

This work was supported by the National Natural Science Foundation of China [grant numbers 42371476, 41971366]; and the Fundamental Research Funds for the Central Universities [grant numbers buctrc202132].

Data availability

The datasets used and analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 27 September 2024 Accepted: 30 June 2025
Published online: 09 July 2025

References

1. Yang SH, Long B, Smola A, Sadagopan N, Zheng Z, Zha H. Like like alike: joint friendship and interest propagation in social networks. In: Proceedings of the 20th international conference on World wide web; 2011. p. 537–46.
2. Bacry E, Mastromatteo I, Muzy JF. Hawkes processes in finance. Market Microstruct Liq. 2015;1(01):1550005.
3. Wang L, Zhang W, He X, Zha H. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining; 2018. p. 2447–56.
4. Shi X, Liu Q, Bai Y, Shang M. RTiSR: a review-driven time interval-aware sequential recommendation method. J Big Data. 2023;10(1):32.
5. Wang Z, Ding D, Ren M, Conti M. TANGO: a temporal spatial dynamic graph model for event prediction. Neurocomputing. 2023;542: 126249.
6. Huang X, Ye Y, Yang X, Xiong L. Multi-view dynamic graph convolution neural network for traffic flow prediction. Expert Syst Appl. 2023;222: 119779.
7. Stockman S, Lawson DJ, Werner MJ. Forecasting the 2016–2017 Central Apennines earthquake sequence with a neural point process. Earth's Future. 2023;11(9):e2023EF003777.
8. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Advances in neural information processing systems; 2017. p. 30.
9. Palm C. Intensitatsschwankungen im fernsprechverker. Ericsson Technics. 1943;44:1–189.
10. Hawkes AG. Spectra of some self-exciting and mutually exciting point processes. Biometrika. 1971;58(1):83–90.
11. Chiang WH, Liu X, Mohler G. Hawkes process modeling of COVID-19 with mobility leading indicators and spatial covariates. Int J Forecast. 2022;38(2):505–20.
12. Dutta HS, Dutta VR, Adhikary A, Chakraborty T. HawkesEye: detecting fake retweeters using Hawkes process and topic modeling. IEEE Trans Inf Forens Secur. 2020;15:2667–78.
13. Hawkes AG. Hawkes processes and their applications to finance: a review. Quant Financ. 2018;18(2):193–8.
14. Rizoiu MA, Lee Y, Mishra S, Xie L. Hawkes processes for events in social media. In: Frontiers of multimedia research; 2017. p. 191–218.
15. Gu Y. Attentive neural point processes for event forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35; 2021. p. 7592–600.
16. Jiang L, Wang P, Cheng K, Liu K, Yin M, Jin B, et al. Eduhawkes: a neural hawkes process approach for online study behavior modeling. In: Proceedings of the 2021 SIAM international conference on data mining (SDM). SIAM; 2021. p. 567–75.
17. Ru Y, Qiu X, Tan X, Chen B, Gao Y, Jin Y. Sparse-attentive meta temporal point process for clinical decision support. Neurocomputing. 2022;485:114–23.
18. Zhang W, Panum T, Jha S, Chalasani P, Page D. Cause: learning granger causality from event sequences using attribution methods. In: International conference on machine learning. PMLR; 2020. p. 11235–45.
19. Gao T, Subramanian D, Bhattacharjya D, Shou X, Mattei N, Bennett KP. Causal inference for event pairs in multivariate point processes. Adv Neural Inf Process Syst. 2021;34:17311–24.
20. Zhang DC, Lauw H. Dynamic topic models for temporal document networks. In: International conference on machine learning. PMLR; 2022. p. 26281–92.
21. He X, Rekatsinas T, Foulds J, Getoor L, Liu Y. Hawkestopic: A joint model for network inference and topic modeling from text-based cascades. In: International conference on machine learning. PMLR; 2015. p. 871–80.
22. Du N, Dai H, Trivedi R, Upadhyay U, Gomez-Rodriguez M, Song L. 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; 2016. p. 1555–64.
23. Xiao S, Yan J, Yang X, Zha H, Chu S. Modeling the intensity function of point process via recurrent neural networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 31; 2017. https://api.semanticscholar.org/CorpusID:7487003.
24. Mei H, Eisner JM. The neural hawkes process: A neurally self-modulating multivariate point process. Advances in neural information processing systems; 2017. p. 30.
25. Zhang Q, Lipani A, Kirnap O, Yilmaz E. Self-attentive Hawkes process. In: International conference on machine learning. PMLR; 2020. p. 11183–93.
26. Zuo S, Jiang H, Li Z, Zhao T, Zha H. Transformer hawkes process. In: International conference on machine learning. PMLR; 2020. p. 11692–702.
27. Omi T, Aihara K, et al. Fully neural network based model for general temporal point processes. Advances in neural information processing systems; 2019. p. 32.
28. Yang C, Mei H, Eisner J. Transformer embeddings of irregularly spaced events and their participants; 2021. arXiv preprint arXiv:2201.00044. https://doi.org/10.48550/arXiv.2201.00044.
29. Ding F, Yan J, Wang H. c-NTPP: learning cluster-aware neural temporal point process. In: Proceedings of the AAAI conference on artificial intelligence. vol. 37; 2023. p. 7369–77.

30. Shang J, Sun M. Geometric hawkes processes with graph convolutional recurrent neural networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33; 2019. p. 4878–85.
31. Shchur O, Biloš M, Günnemann S. Intensity-free learning of temporal point processes; 2019. arXiv preprint arXiv:1909.12127. https://doi.org/10.48550/arXiv.1909.12127.
32. Chen RT, Amos B, Nickel M. Neural spatio-temporal point processes; 2020. arXiv preprint arXiv:2011.04583. https://doi.org/10.48550/arXiv.2011.04583.
33. Zhou Z, Yang X, Rossi R, Zhao H, Yu R. Neural point process for learning spatiotemporal event dynamics. In: Learning for dynamics and control conference. PMLR; 2022. p. 777–89.
34. Yuan Y, Ding J, Shao C, Jin D, Li Y. Spatio-temporal diffusion point processes. In: Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining; 2023. p. 3173–84.
35. Tang Y, Zhang L, Wu H, He J, Song A. Dual-branch interactive networks on multichannel time series for human activity recognition. IEEE J Biomed Health Inform. 2022;26(10):5223–34.
36. Shi J, Gao Y, Gu D, Li Y, Chen K. A novel approach to detect electricity theft based on conv-attentional transformer neural network. Int J Electr Power Energy Syst. 2023;145: 108642.
37. Mourchid Y, Slama R. D-stgcnt: a dense spatio-temporal graph conv-gru network based on transformer for assessment of patient physical rehabilitation. Comput Biol Med. 2023;165: 107420.
38. Yu G, Li A, Zheng C, Guo Y, Wang Y, Wang H. Dual-branch attention-in-attention transformer for single-channel speech enhancement. In: ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE; 2022. p. 7847–51.
39. Shao Z, Han J, Zhao W, Zhou K, Yang S. Hybrid model for short-term wind power forecasting based on singular spectrum analysis and a temporal convolutional attention network with an adaptive receptive field. Energy Convers Manag. 2022;269: 116138.
40. Fang Y, Liu R, Huang H, Zhao P, Wu Q. A Spatio-Temporal Diffusion Model for Missing and Real-Time Financial Data Inference. In: Proceedings of the 33rd ACM international conference on information and knowledge management; 2024. p. 602–11.
41. Chunxiang A, Shen Y, Zeng Y. Dynamic asset-liability management problem in a continuous-time model with delay. Int J Control. 2022;95(5):1315–36.
42. Suresh K, Severn C, Ghosh D. Survival prediction models: an introduction to discrete-time modeling. BMC Med Res Methodol. 2022;22(1):207.
43. Ma T, Jiang C, Xiang J, Wang X, Chau K, Long T. Modeling and analysis of wireless power transfer system via unified full-load discrete-time model. IEEE Trans Ind Electr. 2022;70(6):5626–36.
44. Robert CP, Casella G, Casella G. Monte Carlo statistical methods, vol. 2. Springer; 1999.
45. Chaudhari S, Mithal V, Polatkan G, Ramanath R. An attentive survey of attention models. ACM Trans Intell Syst Technol (TIST). 2021;12(5):1–32.
46. Laghrissi F, Douzi S, Douzi K, Hssina B. IDS-attention: an efficient algorithm for intrusion detection systems using attention mechanism. J Big Data. 2021;8(1):149.
47. Hou Q, Zhou D, Feng J. Coordinate attention for efficient mobile network design. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2021. p. 13713–22.
48. Kaddari Z, Bouchentouf T. A novel self-attention enriching mechanism for biomedical question answering. Expert Syst Appl. 2023;225: 120210.
49. Ma J, Hu S, Fu J, Chen G. A hierarchical attention detector for bearing surface defect detection. Expert Syst Appl. 2024;239: 122365.
50. Wang C, Li M, Smola AJ. Language models with transformers; 2019. arXiv preprint arXiv:1904.09408.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.