frontiers | Frontiers in Computational Neuroscience

TYPE Original Research
PUBLISHED 12 December 2023
DOI 10.3389/fncom.2023.1292842

OPEN ACCESS

EDITED BY
Guillermo Huerta Cuellar,
University of Guadalajara, Mexico

REVIEWED BY
Rider Jaimes Reategui,
University of Guadalajara, Mexico
Liang Tian,
Hong Kong Baptist University, Hong Kong SAR, China

*CORRESPONDENCE
Byounghwa Lee
byounghwa.lee@etri.re.kr

RECEIVED 12 September 2023
ACCEPTED 24 November 2023
PUBLISHED 12 December 2023

CITATION
Lee B, Lee J-H, Lee S and Kim CH (2023) Burst and Memory-aware Transformer: capturing temporal heterogeneity.
Front. Comput. Neurosci. 17:1292842.
doi: 10.3389/fncom.2023.1292842

Burst and Memory-aware Transformer: capturing temporal heterogeneity

Byounghwa Lee*, Jung-Hoon Lee, Sungyup Lee and Cheol Ho Kim

CybreBrain Research Section, Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea

Burst patterns, characterized by their temporal heterogeneity, have been observed across a wide range of domains, encompassing event sequences from neuronal firing to various facets of human activities. Recent research on predicting event sequences leveraged a Transformer based on the Hawkes process, incorporating a self-attention mechanism to capture long-term temporal dependencies. To effectively handle bursty temporal patterns, we propose a Burst and Memory-aware Transformer (BMT) model, designed to explicitly address temporal heterogeneity. The BMT model embeds the burstiness and memory coefficient into the self-attention module, enhancing the learning process with insights derived from the bursty patterns. Furthermore, we employed a novel loss function designed to optimize the burstiness and memory coefficient values, as well as their corresponding discretized one-hot vectors, both individually and jointly. Numerical experiments conducted on diverse synthetic and real-world datasets demonstrated the outstanding performance of the BMT model in terms of accurately predicting event times and intensity functions compared to existing models and control groups. In particular, the BMT model exhibits remarkable performance for temporally heterogeneous data, such as those with power-law inter-event time distributions. Our findings suggest that the incorporation of burst-related parameters assists the Transformer in comprehending heterogeneous event sequences, leading to an enhanced predictive performance.

KEYWORDS

burst, temporal heterogeneity, event sequence, timestamp, inter-event time, temporal point process, self-attention, Transformer

1 Introduction

Temporal heterogeneity is frequently referred to as *burst* within the context of complex systems. Numerous natural and social phenomena exhibit bursty temporal patterns such as single-neuron firing (Kemuriyama et al., 2010; Chan et al., 2016; Metzen et al., 2016; Zeldenrust et al., 2018), earthquakes (Corral, 2004; de Arcangelis et al., 2006), solar flares (Wheatland et al., 1998), and human activity (Barabasi, 2005; Karsai et al., 2018). The term temporal heterogeneity rigorously implies that the distribution of inter-event times, which is the time intervals between two consecutive events, exhibits a heavy-tailed distribution such as a power-law distribution. Moreover, when the system is generally temporally heterogeneous, it implies the presence of temporal correlations among inter-event times. For example, the inter-spike interval distribution display temporally heterogeneous patterns, which cannot be simply interpreted as a random or regular process. Numerous studies have addressed temporal correlations between bursty spikes using approaches such as the non-renewal process (Shahi et al., 2016), intensity functions with voltage-dependent terms

(Yamauchi et al., 2011), and transitions between burst and non-burst states (Dashevskiy and Cymbalyuk, 2018). To quantify temporal heterogeneity, two commonly employed single-value metrics are burstiness and memory coefficient.

Figure 1 illustrates the distinction between temporally heterogeneous inter-event times and those that tend toward homogeneity. The event sequences in Figures 1A, D, F serve as examples of temporal heterogeneity with a power-law inter-event time distribution. The event sequences in Figures 1B, C, E, G present instances that exhibit a more homogeneous random characteristic with an exponential inter-event time distribution. Evidently, the bursty event sequence exhibits clustered events within burst trains, in contrast to the non-burst sequence. Such uneven event occurrences can affect the prediction of event sequences. Without properly accounting for the complicated correlation structure and heterogeneity therein, naive models may struggle to effectively discern hidden patterns.

Event sequence data encompass the temporal occurrences of events spanning various domains, ranging from natural phenomena to social activities. Unlike time series data, event sequence data are defined by sequentially ordered timestamps that signify the timing of individual event occurrences. Numerous studies have focused on predicting the timing of subsequent events have been conducted using temporal point processes (TPPs) (Daley and Vere-Jones, 2008). One of the most widely employed TPP is the Hawkes process (Hawkes, 1971). This process embodies a self-exciting mechanism, wherein preceding events stimulate the occurrence of subsequent events. In contrast to the Hawkes process, the self-correcting process provides a feasible method for establishing regular point patterns (Isham and Westcott, 1979).

The Poisson point process can be employed to generate entirely random and memory-less events (Kingman, 1992). In the Poisson process, the inter-event time (IET) follows an exponential distribution. The Cox process is a generalized Poisson process in which the intensity function varies with the stochastic process (Cox, 1955); thus, it is also referred to as a doubly stochastic Poisson process. Cox processes are frequently employed to model and predict the arrival of insurance claims, enabling insurers to assess risk and manage resources effectively (Rolski et al., 2009). If the intensity function is not entirely random, as in the Cox process, but given as a deterministic time-varying function, it is referred to as an inhomogeneous Poisson process.

Leveraging advancements in deep neural networks, recent studies have introduced Hawkes process models based on neural network frameworks. Specifically, the models of Marked Temporal Point Processes (RMTPP) (Du et al., 2016) and Continuous Time LSTM (CTLSTM) (Mei and Eisner, 2017), utilizing Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), exhibited better performance than Hawkes processes. More recently, the Transformer Hawkes Process (THP) (Zuo et al., 2020) and the Self-Attentive Hawkes Process (SAHP) (Zhang et al., 2020), both grounded in self-attention mechanisms, have demonstrated improved performance.

Our research was primarily motivated by the idea that incorporating temporal heterogeneous characteristics into event sequence predictions yields a superior performance in forecasting events. We propose a *Burst and Memory-aware Transformer* (BMT) model, signifying its capability to train the Transformer by leveraging insights derived from burstiness and memory coefficient, both of which are associated with temporal heterogeneity. Notably, these two metrics were incorporated as embedding inputs for the Transformer architecture. Moreover, a loss function related to these metrics was formulated and employed, thereby enabling the model to naturally capture

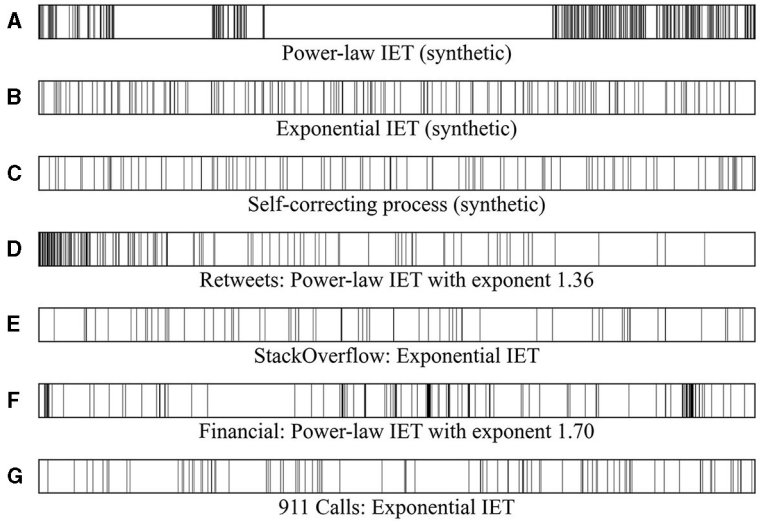

FIGURE 1
(A, D, F) Heterogeneous event sequences with a power-law inter-event time distribution. These event sequences exhibit a high burstiness parameter with significant temporal heterogeneity. **(B, C, E, G)** Event sequences with an exponential inter-event time distribution. These event sequences have burstiness parameters close to 0 and memory coefficients clustered around 0.

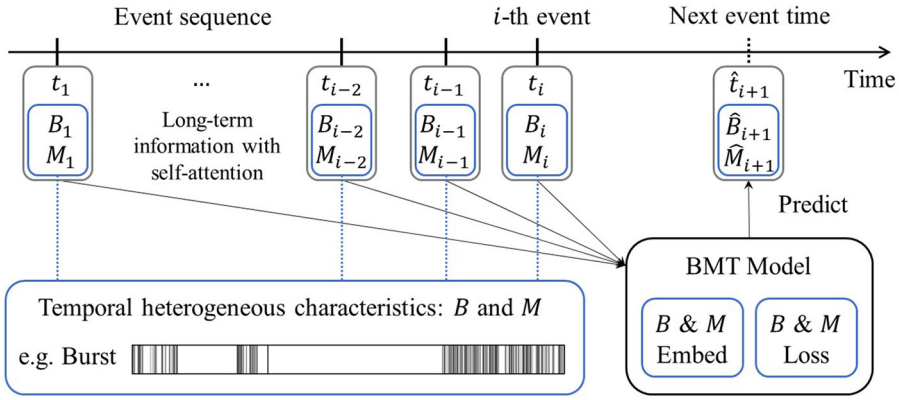

FIGURE 2
Schematic diagram of the Burst and Memory-aware Transformer model. Leveraging information from the preceding events, including burstiness *B* and memory coefficient *M*, the model predicts the timing of the next event through *B* & *M* embedding and the corresponding *B* & *M* loss.

temporal heterogeneity. The overall schematic diagram of the BMT model is depicted in Figure 2.

The main contributions of this paper is summarized as follows:

- The BMT model was developed to integrate insights from the complex systems theory into the Transformer-based temporal point process model, enhancing the capability to incorporate temporal heterogeneity. This study offers a preliminary approach to connect these two distinct disciplines.
- The BMT model surpasses state-of-the-art models by effectively integrating burstiness and memory coefficient into both the embedding procedure and associated loss functions. This is confirmed through extensive numerical experiments across a range of scenarios, including those with and without burstiness and memory coefficient embedding and related loss functions, using real-world datasets and synthetic datasets generated via a copula-based algorithm.
- Our investigation revealed that the BMT model offers particular advantages when dealing with temporally heterogeneous data, such as datasets characterized by a power-law inter-event time distribution, commonly observed in bursty event sequences.
- Our research indicates that excluding either burstiness and memory coefficient embedding or their corresponding loss functions leads to a noticeable reduction in performance. This emphasizes the imperative nature of integrating both elements to achieve optimal performance.

In cases where the inter-event time distribution of the target event sequence exhibits a heavy-tailed distribution, such as a power-law distribution, or where the values of burstiness and memory coefficient significantly deviate from zero, the BMT model ensures superior performance compared to basic Transformer-based models.

The structure of the paper is outlined as follows: Section 2 introduces the background pertaining to the temporal point process, temporal heterogeneity, and generating method for synthetic datasets; Section 3 introduces our Burst and Memory-aware Transformer model; Section 4 presents numerical experiments on synthetic and real-world datasets; Section 5 presents the performance evaluation results; and Section 6 presents the conclusion.

2 Background

2.1 Temporal point process

A temporal point process (TPP) is a stochastic process involving the occurrence of multiple events as time progresses. The foundational data employed to construct the TPP model consists of event sequence data, encompassing event times $\{t_i\}_{i=1}^n$ along with optional marks $\{\kappa_i\}_{i=1}^n$. For example, spike train sequences of neurons are composed of timings of occurrences, along with action potential as associated marks.

In this study, we examine the unmarked case to specifically investigate the effects of burst and memory phenomena, while excluding the influence of correlations with marks that do not align with the research direction. For the prediction of the marked TPP model, one approach involves the independent modeling of the target's marks by thresholding. Alternatively, based on contextual analysis (Jo et al., 2013), interactions with multiple neighbors within an egocentric network can be considered as marks and subsequently modeled.

TPP encompasses the modeling of the conditional intensity function $\lambda(t|\mathcal{H}_t)$ given the history of event times $\mathcal{H}_t \equiv (t_1, ...t_n)$. The notation for the history of event times, $\mathcal{H}_t$ will be omitted for convenience. The intensity function characterizes the instantaneous event rate at any given time by considering past event occurrences. The probability density function $P(t)$ and cumulative distribution function $F(t)$ can be derived based on the intensity function, as follows (Rasmussen, 2018):

$$P(t) = \lambda(t) \exp\left(-\int_{t_{i-1}}^{t} \lambda(t')dt'\right), \quad (1)$$

$$F(t) = 1 - \exp\left(-\int_{t_{i-1}}^{t} \lambda(t')dt'\right). \quad (2)$$

2.1.1 Hawkes process

The Hawkes process, also known as the self-exciting point process, is for a situation where a preceding event excites the occurrence of a subsequent event (Hawkes, 1971). The intensity function $\lambda(t)$ of the Hawkes process is defined as

$$\lambda(t) = \zeta + \eta \sum_{t_i<t} \exp\big(-(t - t_i)\big), \quad (3)$$

where the base intensity ζ and η are positive parameters. When a new event occurs during this process, the intensity increases with η and immediately decays exponentially. The probability of the next event occurring is highest immediately following the incidence of the previous event, and it gradually decreases as time elapses. As a result, this process causes events to cluster together. This includes events that happen quickly in a short time and then long times when nothing happens. The generalized Hawkes process is defined as follows:

$$\lambda(t) = \zeta + \eta \sum_{t_i<t} \gamma(t - t_i), \quad (4)$$

where $\zeta \geq 0$, $\eta > 0$, and $\gamma(t)$ is a density on $(0, \infty)$.

2.1.2 Self-correcting process

In contrast to the Hawkes process, the self-correcting process generates regular inter-event time sequences with randomness (Isham and Westcott, 1979). The intensity function $\lambda(t)$ for the self-correcting process is defined as follows:

$$\lambda(t) = \exp\Big(\zeta t - \sum_{t_i<t} \eta\Big), \quad (5)$$

where ζ and η are positive parameters.

2.1.3 Neural Hawkes process

A limitation of the Hawkes process is that the preceding event cannot inhibit the occurrence of a subsequent event. To overcome these limitations, the neural Hawkes process, which considers the nonlinear relationship with past events using recurrent neural networks, was introduced (Mei and Eisner, 2017). The intensity function $\lambda(t)$ for the neural Hawkes process is defined as follows:

$$\lambda(t) = f\big(\mathbf{w}^\top \mathbf{h}(t)\big), \quad (6)$$

where $f(x) = \beta \log\big(1 + \exp(x/\beta)\big)$ is the softplus function with parameter β which guarantees a positive intensity, and $\mathbf{h}(t)$s are hidden representations of the event sequence from a continuous-time LSTM model. Here, the intensity we refer to is not the marked intensity λ_k; Instead, our focus is on the inherent temporal heterogeneity structure, excluding any interference from correlations between event types and times.

2.1.4 Transformer-based Hawkes process

The Transformer is a deep learning architecture for sequence processing such as natural language processing, with a multi-head self-attention module that captures long-range dependencies within sequences (Vaswani et al., 2017). The Transformer is used not only in language models but also in computer vision, audio processing, and time series forecasting (Lim et al., 2021; Wen et al., 2022; Ma et al., 2023). Recently, the Transformer architecture has also been applied to modeling temporal point processes. The Transformer Hawkes Process (Zuo et al., 2020) and the Self-Attentive Hawkes Process (Zhang et al., 2020) were introduced to model the Hawkes process with a self-attention mechanism to capture the long-range correlations underlying both event times and types.

THP and SAHP differ in two aspects: their use of positional encoding and the form of the intensity function. SAHP employs time-shifted positional encoding to address the limitations of conventional methods, which solely account for the sequence order and neglect inter-event times. The intensity function of the THP model is the softplus function of the weighted sum of three terms: ratio of elapsed time from the previous event, hidden representation vector from the encoder, and base. Conversely, the intensity function of the SAHP model is formulated as a softplus of the Hawkes process terms, each of which is derived from the scalar transformation and nonlinear activation function applied to the hidden representation vector from the encoder.

For both the THP and SAHP models, across synthetic and real-world datasets, their performances in event type prediction and event time prediction surpassed that of the baseline model: Hawkes Process as described in Equation (3), Fully Neural Network model (Omi et al., 2019), Log-normal Mixture model (Shchur et al., 2019), Time Series Event Sequence (TSES) (Xiao et al., 2017), Recurrent Marked Temporal Point Processes (Du et al., 2016), and Continuous Time LSTM (Mei and Eisner, 2017). Given the superior performance of THP over the remaining baseline models, this study refrains from direct performance comparison with the SAHP and baseline models (Zuo et al., 2020), opting to concentrate exclusively on performance comparison with the THP model.

2.2 Temporal heterogeneity

Temporal heterogeneity or burst is characterized by various metrics. The most fundamental quantity is the probability density function of the inter-event times. The inter-event time is defined as the time interval between two consecutive events, that is, $\tau_i \equiv t_{i+1} - t_i$, where t_i is i-th event time of the event sequence.

When the inter-event time distribution is heavy-tailed, the corresponding event sequence exhibits temporal heterogeneity. Specifically, the power-law inter-event distribution found in diverse natural and social phenomena is as follows:

$$P(\tau) \sim \tau^{-\alpha}, \quad (7)$$

where a is a constant and α is a power-law exponent.

2.2.1 Burstiness parameter

Several metrics characterize the properties of temporal heterogeneity. Burstiness B measures the burst phenomenon (Goh and Barabási, 2008), and is defined as follows:

$$B \equiv \frac{r-1}{r+1} = \frac{\sigma - \langle\tau\rangle}{\sigma + \langle\tau\rangle}, \tag{8}$$

where $r \equiv \sigma/\langle\tau\rangle$ is the coefficient of variation (CV) of the inter-event time and σ and $\langle\tau\rangle$ is the standard deviation and average of τs, respectively. Here, $B = -1$ for regular event sequences, $B = 0$ for Poissonian random cases, and $B = 1$ for extremely bursty cases.

When the number of events is sufficiently small, the burstiness parameter causes errors. The fixed burstiness parameter considering the finite-size effect is as follows (Kim and Jo, 2016):

$$B_n \equiv \frac{\sqrt{n+1}r - \sqrt{n-1}}{(\sqrt{n+1}-2)r + \sqrt{n-1}}. \tag{9}$$

We employed the fixed burstiness parameter (9) to handle short-length event sequences throughout this study.

2.2.2 Memory coefficient

The memory coefficient M quantifies the correlations between consecutive inter-event times within a sequence consisting of n inter-event times, that is, $\{\tau_i\}_{i=1,\ldots,n}$, as follows (Goh and Barabási, 2008):

$$M \equiv \frac{1}{n-1}\sum_{i=1}^{n-1}\frac{(\tau_i - \langle\tau\rangle_1)(\tau_{i+1} - \langle\tau\rangle_2)}{\sigma_1\sigma_2}, \tag{10}$$

where $\langle\tau\rangle_1$ ($\langle\tau\rangle_2$) and σ_1 (σ_2) are the average and standard deviation of the inter-event times $\tau_1, \tau_2, \ldots, \tau_{n-1}$ ($\tau_2, \tau_3, \ldots, \tau_n$), respectively. This is the Pearson correlation coefficient between consecutive inter-event times. Here, $M = 0$ indicates no correlation, and $M > 0$ indicates a positive correlation, which means that a large inter-event time follows after a large inter-event time and vice versa for small inter-event time. $M < 0$ indicates a negative correlation, which means small inter-event time follows after the large inter-event time and vice versa for a large inter-event time.

2.2.3 Applications of B and M to BMT model

When plotting M on the x-axis and B on the y-axis for datasets with various inter-event time distributions, it can be observed that event sequences with similar inter-event time distributions tend to cluster at similar positions (Goh and Barabási, 2008). Essentially, if the ranges of B and M values are known, a rough estimate of the inter-event time distribution can be anticipated. Building on this insight, we devised a BMT model to facilitate learning by designing a method in which the values of B and M were combined and fed into the encoder as inputs. Specifically, when the values of B and M exhibit temporal heterogeneity in their ranges, the encoder of the Transformer can produce inter-event time prediction values with a heavy-tailed inter-event time distribution.

Moreover, B and M are not independent: they are intertwined and move in conjunction. For instance, even when attempting to alter only M by shuffling the inter-event times, B can also change. This serves as evidence that embedding both B and M concurrently yields superior performance compared with embedding either one of them individually.

2.3 Copula-based algorithm for generating sequence of inter-event times

To comprehend the impact of burstiness and memory coefficient on the model, we generated synthetic datasets using a copula-based algorithm (Jo et al., 2019). The content of the copula-based algorithm in this study was obtained from Jo et al. (2019). For convenience, we provide a brief overview of the relevant content. The copula-based algorithm models the joint probability distribution of two consecutive inter-event times, that is, $P(\tau_i, \tau_{i+1})$, by adopting the Farlie-Gumbel-Morgenstern (FGM) copula (Nelsen, 2006). The joint probability distribution according to the FGM copula is formulated as follows:

$$P(\tau_i, \tau_{i+1}) = P(\tau_i)P(\tau_{i+1})[1 + rf(\tau_i)f(\tau_{i+1})], \tag{11}$$

where

$$f(\tau) \equiv 2F(\tau) - 1, \quad F(\tau) \equiv \int_0^\tau d\tau' P(\tau'). \tag{12}$$

Parameter r is used to control the correlation between τ_i and τ_{i+1} and is in the range of $-1 \leq r \leq 1$. $F(\tau)$ is the cumulative distribution function (CDF) of $P(\tau)$. After applying the transformation method (Clauset and Shalizi, 2009), the next inter-event time τ_{i+1} can be obtained as Jo et al. (2019)

$$\tau_{i+1} = F^{-1}\left[\frac{c_i - 1 + \sqrt{(c_i+1)^2 - 4c_i x}}{2c_i}\right], \tag{13}$$

where F^{-1} is the inverse of $F(\tau)$, $c_i \equiv rf(\tau_i)$, and x is a random number sampled from a uniform distribution within interval $0 \leq x < 1$. The copula-based algorithm has the advantage of generating event sequences with independent control of the inter-event time distribution and memory coefficient.

3 Burst and Memory-aware Transformer

3.1 Discretization of B and M

Given that the burstiness parameter and memory coefficient are real numbers within the range of $[-1, 1]$, it is necessary to discretize them for embedding within the Transformer. We adopt the uniform discretization transform; the range $[-1, 1]$ is divided into segments of fixed length by the number of bins b, respectively, and subsequently mapped to a single natural number. The continuous values of the burstiness parameter B and memory coefficient M are discretized into natural numbers d_B and d_M, respectively. For example, when the number of bins is $b = 4$, then $d_B = 3$ if $M = 0.2$, and it $d_B = 1$ if $M = -0.7$. Then, one can obtain the discretized pairs of B and M as (d_B, d_M), where d_B and

d_M are ranging from 1 to b. To map the pair into a unique natural number, the Cantor pairing function was employed. The Cantor pairing function maps discretized d_B and d_M into a unique natural number $d_{B,M}$ as

$$d_{B,M} \equiv \frac{1}{2}(d_B + d_M)(d_B + d_M + 1) + d_M. \tag{14}$$

When the number of discretization bins is b, the number of $d_{B,M}$ is b^2, corresponding to the vocabulary size of the Transformer. Then, we can obtain the one-hot vector of the discretized B & M as $\mathbf{d}_{B,M} \in \mathbb{R}^{b^2}$.

3.2 Embedding event times, and *B* and *M*

The event sequence $\mathcal{S} = \{t_i\}_{i=1}^n$ of n events and discretized and one-hot B & M, $\mathbf{d}_{B,M}$ are fed into the self-attention module after proper encoding. First, the event times are transformed using the positional encoding method (Vaswani et al., 2017) to embed the temporal order information into an event sequence. The j-th element of sinusoidal positional encoding for the i-th event time t_i is calculated as:

$$[\mathbf{z}_t(t_i)]_j = \begin{cases} \sin(\omega_k t_i), & \text{if} \quad j = 2k \\ \cos(\omega_k t_i), & \text{if} \quad j = 2k+1, \end{cases} \tag{15}$$

where $\omega_k = 1/10,000^{2k/d}$, the embedding index k is the quotient when dividing j by 2, and $\mathbf{z}_t(t_i) \in \mathbb{R}^d$, where d is the encoding dimension. By multiplying ω_k with the event time t_i, it is converted into an angle, which is then mapped to sine and cosine functions, providing different positional information for each event time.

For the given event times $\{t_i\}_{i=1}^n$, the inter-event times are $\tau_i \equiv t_{i+1} - t_i$ for $i = 1, ..., n-1$. The burstiness parameter (9) and memory coefficient (10) were calculated for all partial sequences. This essentially implies that the input to the encoder is fed sequentially from t_1, ..., t_i,..., t_n, and for each of these instances, the B & M embedding incorporates the calculated B and M values up to t_1 (i.e., B_1 and M_1), ..., up to t_i (i.e., B_i and M_i), ..., and up to t_n ($B_n = B$ and $M_n = M$ for the entire sequence). Note that, during the actual operation of the Transformer, computations are performed in parallel; thus, the sliding B & M embedding vectors form a lower triangular matrix.

The B & M embedding vector $\mathbf{z}_e(B_i, M_i)$ for the one-hot vector of the discretized B_i and M_i, $\mathbf{d}_{B_i,M_i}$, is calculated using a linear embedding layer as follows:

$$\mathbf{z}_e(B_i, M_i) = \mathbf{W}_E \mathbf{d}_{B_i,M_i}, \tag{16}$$

where $\mathbf{W}_E \in \mathbb{R}^{d \times b^2}$ denotes an embedding matrix. Then for the i-th event, the event time embedding vector $\mathbf{z}_t(t_i) \in \mathbb{R}^d$ and the B & M embedding vector $\mathbf{z}_e(B_i, M_i) \in \mathbb{R}^d$ are summed together to acquire the hidden representation of the i-th event $\mathbf{z}_i \in \mathbb{R}^d$ as:

$$\mathbf{z}_i = \mathbf{z}_t(t_i) + \mathbf{z}_e(B_i, M_i). \tag{17}$$

Then the embedding matrix for a whole single event sequence is given by:

$$\mathbf{Z} = [\mathbf{z}_i]_{i=1,...,n}, \tag{18}$$

where $\mathbf{Z} \in \mathbb{R}^{n \times d}$ and n is the length of the event sequence, that is, the number of events in a single sequence.

3.3 Self-attention module

After acquiring the embedding matrix $\mathbf{Z}$ for each event sequence according to Equation (18), we propagated $\mathbf{Z}$ into the input of the self-attention module. The resulting attention output $\mathbf{S}$ is defined as follows:

$$\mathbf{S} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_K}}\right)\mathbf{V}, \tag{19}$$

where $\mathbf{Q} = \mathbf{Z}\mathbf{W}^{\mathbf{Q}}$, $\mathbf{K} = \mathbf{Z}\mathbf{W}^{\mathbf{K}}$, $\mathbf{V} = \mathbf{Z}\mathbf{W}^{\mathbf{V}}$, and $\mathbf{S} \in \mathbb{R}^{n \times d_V}$. Here, $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ represent the query, key, and value matrices, respectively, obtained by applying distinct transformations to $\mathbf{Z}$. The transformation parameters are $\mathbf{W}^Q \in \mathbb{R}^{d \times d_K}$, $\mathbf{W}^K \in \mathbb{R}^{d \times d_K}$, and $\mathbf{W}^V \in \mathbb{R}^{d \times d_V}$, respectively. In contrast to conventional RNN models, the self-attention mechanism enables an equitable comparison of not only recent values but also the significance of distant past values of the sequence. Consequently, this facilitates the learning of long-term dependencies.

The BMT model employs multi-head attention, similar to other Transformers. Multi-head attention enables the model to manage diverse patterns and contexts of the input sequence. The multi-head attention output $\mathbf{S}$ is given by $\mathbf{S} = [\mathbf{S}_1, ..., \mathbf{S}_i, ..., \mathbf{S}_m]\mathbf{W}_O$, where $\mathbf{S}_i \in \mathbb{R}^{n \times d_V/m}$ is the attention output for the i-th multi-head and $\mathbf{W}_O \in \mathbb{R}^{m \cdot d_V \times d}$ is aggregation parameters.

After the multi-head attention, the resulting attention output $\mathbf{S}$ is subsequently passed into a position-wise feed-forward network, yielding hidden representations $\mathbf{h}(t)$ for the event sequence as:

$$\mathbf{H} = \text{ReLU}(\mathbf{S}\mathbf{W}_{\text{FC1}} + \mathbf{b}_1)\mathbf{W}_{\text{FC2}} + \mathbf{b}_2, \tag{20}$$

where $\mathbf{W}_{\text{FC1}} \in \mathbb{R}^{d \times d_H}$, $\mathbf{W}_{\text{FC2}} \in \mathbb{R}^{d_H \times d}$, $\mathbf{b}_1 \in \mathbb{R}^{d_H}$, and $\mathbf{b}_2 \in \mathbb{R}^d$ are the parameters of each neural network. The i-th event of the event sequence corresponds to the i-th row of the hidden representation matrix $\mathbf{H}$, that is, $\mathbf{h}(t_i) = \mathbf{H}(i, :)$. Furthermore, masks are employed to prevent the model from learning about the future in advance. The hidden representation $\mathbf{H} \in \mathbb{R}^{n \times d}$ encapsulates insights regarding burstiness and memory coefficient for each event within the sequence, acquired through the self-attention mechanism. We further enhanced the incorporation of sequential information by applying LSTM to the hidden representation.

3.4 Training and loss function

The BMT model employs five types of loss functions: (1) squared error of the event time, (2) event log-likelihood loss as described in Equation (22), (3) cross entropy of discretized B & M, (4) squared error of B, and (5) squared error of M.

3.4.1 Event time loss

The most crucial loss function within the model is how accurately it predicts the next event times. The next event time

prediction is $\hat{t}_{i+1} = \mathbf{W}_t \mathbf{h}(t_i)$, where $\mathbf{W}_t \in \mathbb{R}^{1\times d}$ is the parameter of the event time predictor. To address this, the squared error loss function of the event times for the event sequence is defined as:

$$\mathcal{L}_t = \sum_{i=2}^{n} (t_i - \hat{t}_i)^2, \tag{21}$$

where $\hat{t}_i$ is the predicted event time.

3.4.2 Event log-likelihood

The typical approach for optimizing the parameters of the Hawkes process involves utilizing Maximum Likelihood Estimation (MLE). There are two constraints: (1) no events before time 0, and (2) unobserved event time must appear after the observed time interval. When the observed event sequences are $t_1, ..., t_n \in [0, T)$, then likelihood of an event sequence is given by $\mathcal{L}' = P(t_1) \cdots P(t_{n-1})(1-F(T))$, where $F(\cdot)$ is the cumulative distribution function, and the last term is for the second constraint. Using (1) and (2), and applying the logarithm function, we obtain the following log-likelihood:

$$\mathcal{L}_\lambda = \sum_{i=1}^{n} \log \lambda(t_i) - \int_0^T \lambda(t')dt'. \tag{22}$$

The first term denotes the sum of the log-intensity functions for the past n events, and the second term represents the non-event log-likelihood.

Here, the intensity function $\lambda(t)$ is defined in the interval $t \in [t_i, t_{i+1}]$ according to the following expression:

$$\lambda(t) = \beta \log\big(1 + \exp(\mathbf{w}_\lambda^\top \mathbf{h}(t_i)\beta^{-1})\big), \tag{23}$$

where β is the softness parameter, $\mathbf{w}_\lambda \in \mathbb{R}^{d\times 1}$ is a parameter that converts the term inside the exponential function into a scalar, and $\mathbf{h}$ is the hidden representation derived from the encoder. The essence of this intensity function aligns with that of the Neural Hawkes Process, as shown in Equation (6). The softplus function formulation was employed to guarantee non-negativity of the intensity.

3.4.3 Discretized *B* and *M* loss

The model predicts the discretized $\hat{B}_i$ & $\hat{M}_i$, $\hat{d}_{B_i,M_i}$, based on the hidden representations $\mathbf{h}(t_{i-1})$ as:

$$\hat{\mathbf{p}}_i = \text{Softmax}(\mathbf{W}_{B,M}\mathbf{h}(t_{i-1})), \tag{24}$$

$$\hat{d}_{B_i,M_i} = \arg\max_{d'} \hat{\mathbf{p}}_i(d'), \tag{25}$$

where $\mathbf{W}_{B,M} \in \mathbb{R}^{b^2\times d}$ is the parameter of the discretized B_i & M_i predictor, and $\hat{\mathbf{p}}_i(d')$ is the d'-th element of $\hat{\mathbf{p}}_i$. To measure the accuracy of B_i & M_i embedding, the following cross-entropy between the ground truth discretized B_i & M_i, d_{B_i,M_i}, and the predicted discretized $\hat{B}_i$ & $\hat{M}_i$, $\hat{d}_{B_i,M_i}$, is calculated:

$$\mathcal{L}_{B,M} = -\sum_{i=2}^{n} \mathbf{d}_{B_i,M_i}^\top \log(\hat{\mathbf{p}}_i), \tag{26}$$

where $\mathbf{d}_{B_i,M_i} \in \mathbb{R}^{b^2}$ is the ground truth one-hot encoding vector.

3.4.4 *B* loss and *M* loss

Additionally, the model utilizes the squared errors of the burstiness parameter directly as:

$$\mathcal{L}_B = \sum_{i=2}^{n} (B_i - \hat{B}_i)^2, \tag{27}$$

where B_i and $\hat{B}_i$ are the ground truth and predicted burstiness parameters, respectively. The squared errors of the memory coefficient value can be defined in a similar manner.

$$\mathcal{L}_M = \sum_{i=2}^{n} (M_i - \hat{M}_i)^2, \tag{28}$$

where M_i and $\hat{M}_i$ is ground truth and predicted memory coefficient, respectively.

3.4.5 Overall loss

By aggregating the aforementioned loss functions (21), (22), and (26)–(28), the overall loss function of the model is defined as follows:

$$\mathcal{L} = \mathcal{L}_t + \alpha_1\mathcal{L}_\lambda + \alpha_2\mathcal{L}_{B,M} + \alpha_3\mathcal{L}_B + \alpha_4\mathcal{L}_M, \tag{29}$$

where α_1 to α_4 are the hyperparameters that balance each loss function determined using the validation datasets. The overall framework of the BMT model is illustrated in Figure 3.

4 Experiments

4.1 Synthetic datasets

We generated synthetic data using the copula-based algorithm for two different inter-event time distributions. The model was tested for the exponential and power-law inter-event time distribution, which also have a different range of memory coefficient and burstiness, to directly understand the impact of temporal heterogeneity on the BMT model and other models. Along with the two synthetic datasets below, we tested the regular event sequences generated by the self-correcting process, as in Equation (5). The statistics of the datasets are displayed in Table 1.

4.1.1 Power-law inter-event time distribution

The power-law inter-event time distribution with a power-law exponent α is defined as $P(\tau) = (\alpha - 1)\tau^{-\alpha}\theta(\tau - 1)$ and the corresponding cumulative distribution function is $F(\tau) = (1 - \tau^{1-\alpha})\theta(\tau - 1)$, where $\theta(\cdot)$ represents the Heaviside step function with a lower bound of 1. After substituting the inter-event time distribution into Equation (13), we obtain the next inter-event time τ_{i+1} from a given previous inter-event time τ_i and random number x in $0 \le x < 1$ as

$$\tau_{i+1} = \left[\frac{2c_i}{c_i + 1 - \sqrt{(c_i+1)^2 - 4c_i x}}\right]^{1/(\alpha-1)}, \tag{30}$$

where $c_i = \frac{(2\alpha-3)^2}{(\alpha-1)(\alpha-3)} M(1 - 2\tau_i^{1-\alpha})$ (Jo et al., 2019).

A total of 1,000 sequences with a power-law inter-event time distribution were generated with different parameters according to Equation (30). The power-law exponent α, memory coefficient M, and the number of events for each event sequence are randomly and independently drawn from $2.1 \leq \alpha \leq 2.9$, $-1/3 \leq M \leq 1/3$, and $50 \leq n \leq 500$, respectively. The initial inter-event time was randomly drawn from 1 to 2. Depending on the power-law exponent and memory coefficient, the burstiness ranged from 0.297 to 0.962.

As depicted in Figure 4, the power-law inter-event time datasets exhibit pronounced dispersion toward the region of larger burstiness and memory coefficients (B and M scatter plots). Moreover, these datasets show a power-law inter-event time distribution with exponent values $\alpha = 2.4$ close to the average within the range of exponents $2.1 < \alpha < 2.9$.

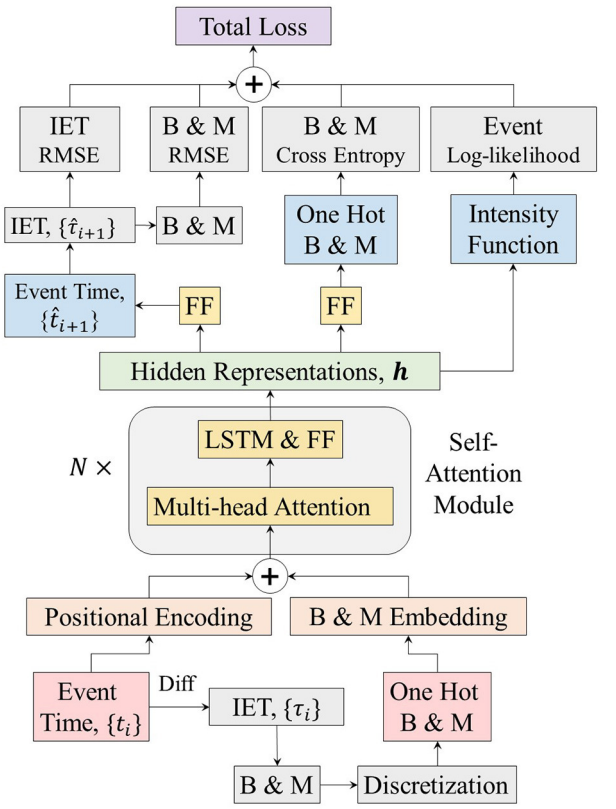

FIGURE 3
Architecture of the Burst and Memory-aware Transformer model. IET; inter-event time; FF, feed-forward neural network; *B*, burstiness; *M*, memory coefficient.

4.1.2 Exponential inter-event time distribution

The exponential inter-event time distribution with mean μ is defined as $P(\tau) = \mu^{-1}e^{-\tau/\mu}$ and the corresponding cumulative distribution function is $F(\tau) = 1 - e^{-\tau/\mu}$ and the relationship between the parameter and memory coefficient is $r = 4M$. After substituting the inter-event time distribution into Equation (13), we obtain the next inter-event time τ_{i+1} from a given previous inter-event time τ_i and random number x in $0 \leq x < 1$ as follows:

$$\tau_{i+1} = \mu \ln\left[\frac{2c_i}{c_i + 1 - \sqrt{(c_i+1)^2 - 4c_i x}}\right], \tag{31}$$

where $c_i = 4M(1 - 2e^{-\tau_i/\mu})$ (Jo et al., 2019).

A total of 1,000 sequences with an exponential inter-event time distribution were generated using different parameters, according to Equation (31). The mean inter-event time μ, memory coefficient M, and the number of events n for each event sequence were randomly and independently drawn from $1 \leq \mu \leq 100$, $-1/3 \leq M \leq 1/3$, and $50 \leq n \leq 500$, respectively. The initial inter-event time was set to μ for each event sequence.

As illustrated in Figure 4, the B and M scatter plots of the exponential inter-event time datasets show that B values are concentrated in the lower range, whereas M values exhibit a broader distribution spread both above and below. This contrasts with the self-correcting process datasets, where the B and M scatter plots show that both B and M clustered at $\sim$0. Although both datasets have an exponential inter-event time distribution, their heterogeneity differs owing to variations in the relationship between B and M. Even with an exponential inter-event time distribution, appropriately shuffling inter-event times can generate event sequences with temporal heterogeneity (i.e., burst) characteristics. We examine this difference further later, as it plays a role in generating variations in performance.

4.2 Real-world datasets

We adopted four real-world datasets to evaluate the models: the *Retweets*, *StackOverflow*, *Financial Transaction*, and *911 Calls*

TABLE 1 Datasets statistics.

Datasets		Power-law	Exponential	Self-correcting	Retweets	Stack Overflow	Financial	911 Calls
IET	Mean	3.4645	57.495	0.20015	2,840.8	0.58677	1.5853	338.14
	S.D.	2.9372	32.938	0.00193	2,157.8	0.16667	5.6152	249.20
B	Mean	0.176	−0.008	−0.048	0.754	0.052	0.522	0.100
	S.D.	0.335	0.068	0.028	0.137	0.083	0.069	0.100
M	Mean	−0.021	−0.059	−0.084	0.442	0.031	0.155	0.009
	S.D.	0.195	0.200	0.058	0.274	0.119	0.078	0.158

IET, inter-event time; B, burstiness; M, memory coefficient; S.D., standard deviation.

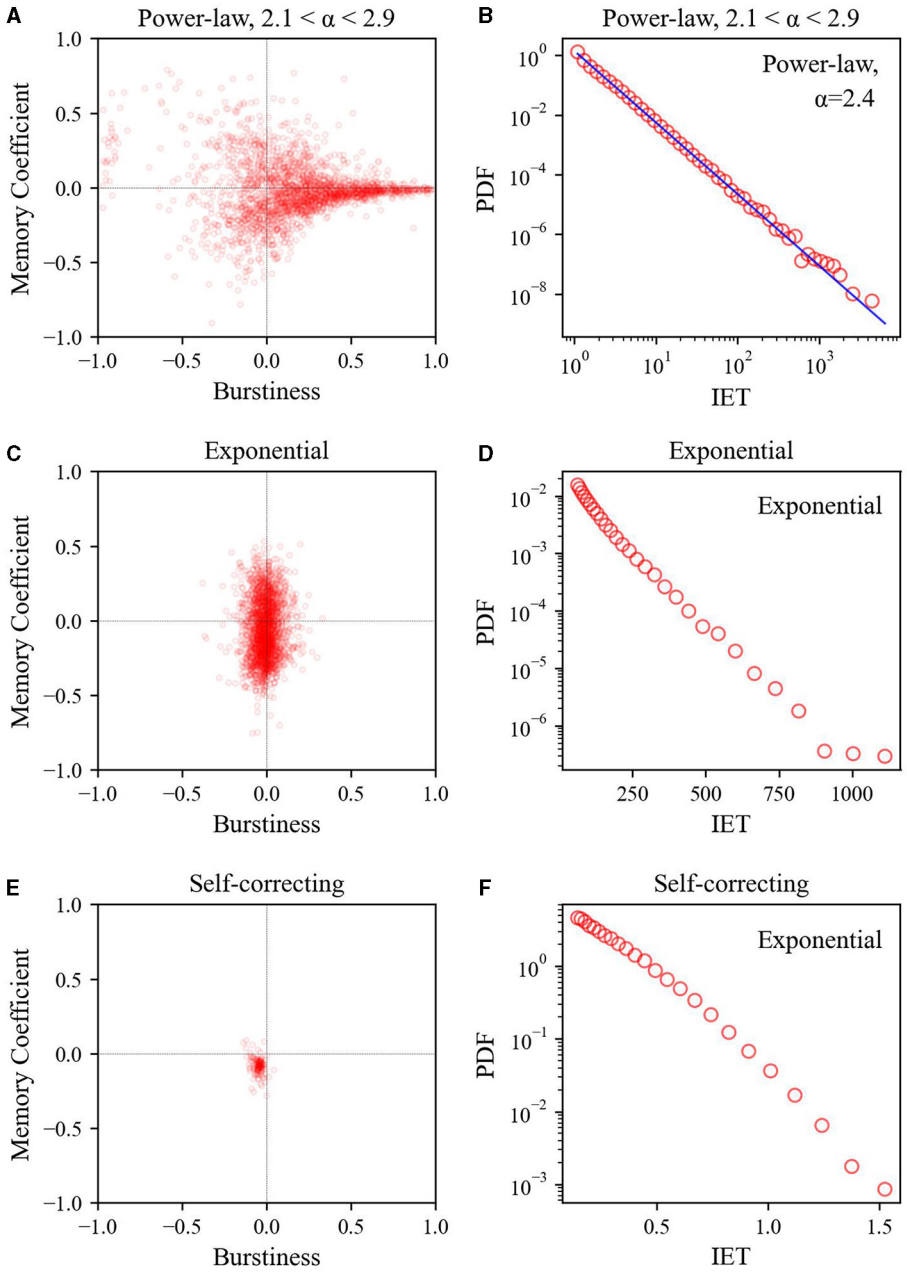

FIGURE 4
Relationship between burstiness and memory coefficient (left) and inter-event time distribution (right) across three synthetic datasets: **(A, B)** power-law inter-event time, **(C, D)** exponential inter-event time, and **(E, F)** self-correcting process. For calculating the inter-event time distribution, logarithmic binning was employed.

datasets. The Retweets dataset (Zhao et al., 2015) contains sequences of tweets and follow-up tweets. The original datasets contained three categories, based on the number of followers. The StackOverflow dataset (Leskovec and Krevl, 2014) contains each user's reward history, that is, the timestamp of users receiving the badge and the type of the badge. The Financial Transaction dataset (Du et al., 2016) includes raw order book records from the New York Stock Exchange (NYSE) for a stock in one day, with a millisecond-level time granularity. The events correspond to two types of actions: buy and sell orders. The 911 Calls datasets[1] contains emergency phone call records for Montgomery County, PA. This dataset contains information such as calling times and location, and we conducted aggregation based on location, utilizing zip codes as identifiers. The dataset covers a five-year period, which is a relatively extensive time frame for prediction purposes. Therefore, we partitioned the data into monthly intervals. Additionally, to ensure statistical significance, we included only

1 The dataset is available on https://www.kaggle.com/datasets/mchirico/montcoalert/.

those locations where the number of events exceeded 50 in the data.

Although there are other commonly used datasets, the burst and memory-aware characteristics assumed by the BMT model are applicable when the sequence length is sufficiently long. Furthermore, we sampled event sequences in quantities comparable to synthetic data while concurrently excluding sequences with short lengths. The time units for each dataset are as follows: Retweet and StackOverflow datasets are in days, Financial Transaction datasets are in milliseconds, and 911 Calls datasets are in minutes. The statistics of the datasets are displayed in Table 1.

As shown in Figure 5, when comparing the Retweets datasets (or Financial Transaction datasets) to the StackOverflow datasets (or 911 Calls datasets), it is evident that the Retweets datasets and Financial Transaction datasets are more temporally heterogeneous. In the B and M scatter plots, the Retweets datasets (or Financial Transaction datasets) are concentrated in regions with larger values for both B and M, whereas the StackOverflow datasets (or 911 Calls datasets) are centered around values near 0 for both B and M. However, when compared to the self-correcting process datasets, the StackOverflow (or 911 Calls datasets) datasets exhibit greater dispersion. Additionally, the inter-event time distribution reveals that the Retweets datasets and Financial Transaction datasets follow a power-law distribution (exponent of 1.36 and 1.70, respectively), whereas the StackOverflow datasets and 911 Calls datasets follow an exponential distribution.

4.3 Impact of *B* and *M* embedding and losses

While altering the combination of loss functions during the experimental process, there were five control groups.

1. **BMT-NoE&NoL** (BMT without B & M embedding and without corresponding losses). The simplest scenario occurs when $\alpha_2 = \alpha_3 = \alpha_4 = 0$, utilizing only time and event losses. In this case, only event time and intensity were considered.
2. **BMT-NoE&L** (BMT without embedding for B & M, but with losses for either B or M). To incorporate the effects of the B & M losses, we also consider the case of $\alpha_2 = 0, \alpha_3 > 0$, and $\alpha_4 > 0$ with time and event losses. Note that the case for $\alpha_2 > 0$ relates to predicting the discretized on-hot B & M, and hence it is not applicable in this scenario.
3. **BMT-E&NoL** (BMT without losses related to B & M, but with embedding for B & M). The control group examines the impact of loss for B & M; the representation vector remains consistent with the BMT model, as shown in Equation (18), but without $\mathcal{L}_{B,M}$, $\mathcal{L}_B$, and $\mathcal{L}_M$, that is, $\alpha_2 = \alpha_3 = \alpha_4 = 0$.
4. **BMT-B** (BMT with B embedding only and the corresponding loss). In the case where only B is embedded and the model is trained, the loss is also computed exclusively based on B as $\alpha_2 = 0, \alpha_3 > 0$, and $\alpha_4 = 0$ with time and event losses.
5. **BMT-M** (BMT with M embedding only and corresponding loss). In the case where only M is embedded and the model is trained, the loss is also computed exclusively based on M as $\alpha_2 = \alpha_3 = 0$, and $\alpha_4 > 0$ with time and event losses.

5 Results and discussion

We tested several hyperparameters for both the BMT and THP models and chose the configuration that yielded the best validation performance. The hyperparameters are as follows: the number of bins for discretization (b) is set to 40, mini-batch size is 16, dropout rate is 0.1, embedding dimensions (d and d_H) are both 128, self-attention dimensions (d_K and d_V) are 32, with eight layers in the encoder and 8 heads. For the loss function, hyperparameters were fine-tuned, mainly as follows: $\alpha_1 = 1e3, \alpha_2 = 4e3, \alpha_3 = \alpha_4 = 1e4$. We employed the ADAM (adaptive moment estimation) optimizer with hyperparameters β set to (0.9, 0.999). Regarding the learning rate, we utilized PyTorch StepLR, initializing it at 1e-4 and reducing the learning rate by a factor of 0.9 every 15 steps.

The performance evaluation results for different models across diverse datasets are presented in Table 2. The results indicate that BMT achieves superior performance compared to THP and other control models in terms of the root mean squared error (RMSE) of the event times and log-likelihood. The main metric, RMSE, is a unit-adjusted value obtained by taking the square root of Equation (21). It measures how much predicted event times of the model differ from the actual event times. However, RMSE has a drawback, especially in the case of heterogeneous data, where it can perform well by accurately predicting large values while potentially struggling with smaller ones. To address this limitation, we introduce the event log-likelihood, defined in Equation (22), as a second metric. This metric arises when probabilistically modeling event sequences using the intensity function λ derived from Equation (1). A higher likelihood of the intensity function calculated with predicted event times of the model indicates that the model better probabilistically mimics the actual event sequence. Consequently, larger values of this metric correspond to better performance. Additionally, when considering the B and M losses in Equation (26), they represent how well the model captures discretized burstiness and discretized memory coefficients. Smaller values of these losses indicate better performance in replicating these aspects.

In particular, as the data became more heterogeneous, performance improvement became more pronounced. In synthetic datasets, the performance enhancement of the BMT model was greater for power-law inter-event time data than for self-correcting data, which is a less heterogeneous exponential inter-event time distribution (see Figure 4). Similarly, in real-world datasets, the overall performance of the BMT model was superior in the Retweets dataset, which exhibited a more power-law inter-event time distribution, compared to the StackOverflow datasets with a less heterogeneous exponential inter-event time distribution (see Figure 5).

When compared to the BMT-NoE&L model with respect to the RMSE of the event times, the BMT model shows that superior performance across all datasets except StackOverflow. This suggests that the inclusion of B & M embedding processes aids in augmenting the performance of the model by enabling the encoder to grasp the burst structure of event sequences. Compared with the BMT-E&NoL model, the BMT model demonstrates enhanced performance across all datasets, indicating that the integration of B & M losses into the overall loss function contributes to the

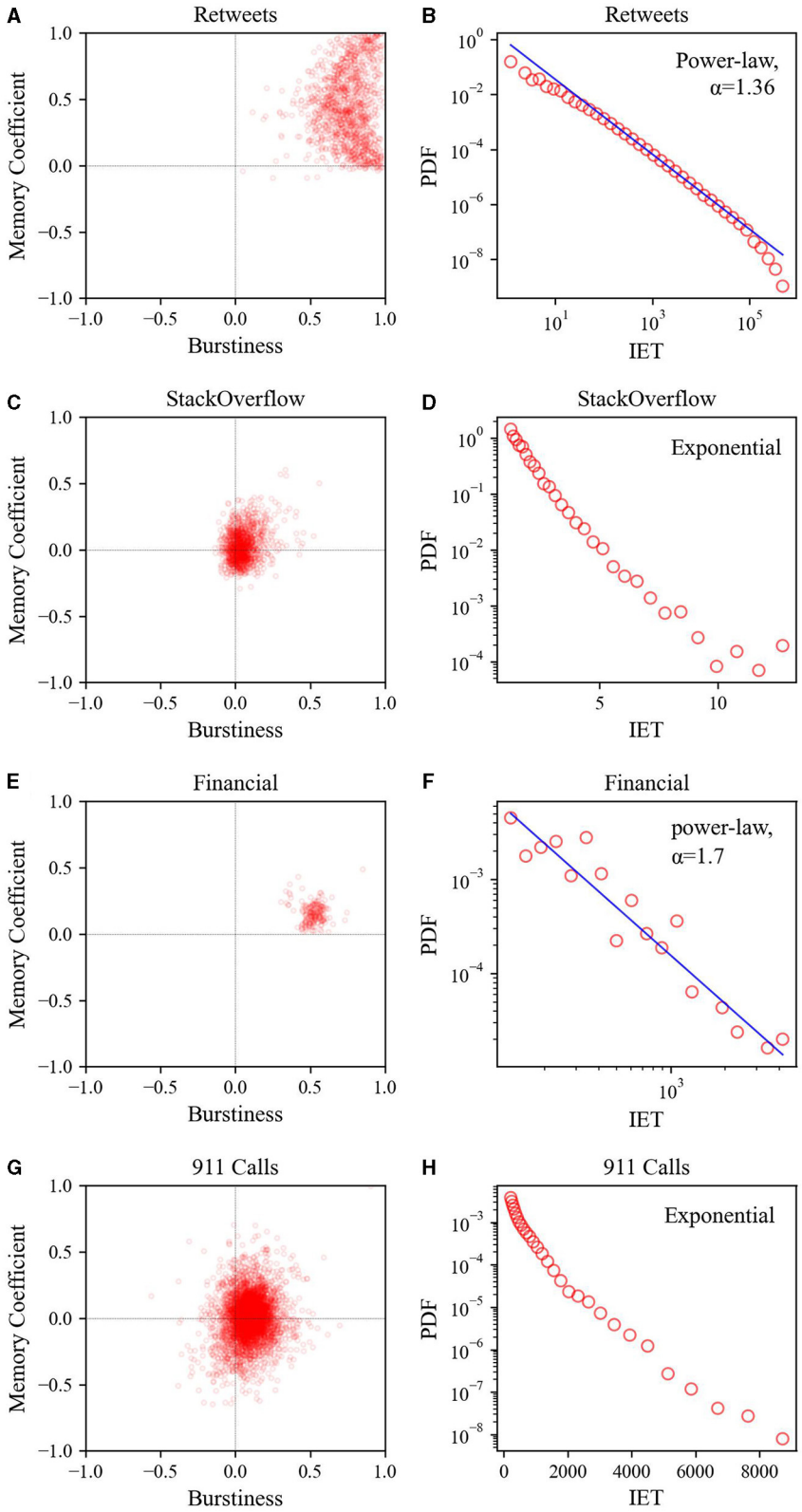

FIGURE 5
Relationship between burstiness and memory coefficient (left) and inter-event time distribution (right) for four real-world datasets: **(A, B)** Retweets, **(C, D)** StackOverflow, **(E, F)** Financial Transaction, and **(G, H)** 911 Calls. For calculating the inter-event time distribution, logarithmic binning was employed.

TABLE 2 Performance evaluation results across diverse datasets for different models.

Dataset	Model	RMSE of time	LL	CE of BM
Power- law	THP	238.0	−2.303	N/A
	BMT-NoE&NoL	66.6	−2.451	N/A
	BMT-NoE&L	68.2	−2.493	N/A
	BMT-E&NoL	82.0	−2.348	12.46
	BMT-B	107.6	−2.712	N/A
	BMT-M	52.3	−2.730	N/A
	BMT	**40.5**	**−2.302**	**8.05**
Exponential	THP	1,973.7	−19.910	N/A
	BMT-NoE&NoL	158.7	−6.486	N/A
	BMT-NoE&L	208.8	−15.640	N/A
	BMT-E&NoL	102.2	−5.633	12.70
	BMT-B	106.1	−7.201	N/A
	BMT-M	140.8	−9.912	N/A
	BMT	**80.1**	**−5.171**	**5.77**
Self-correcting	THP	0.184	0.200	N/A
	BMT-NoE&NoL	0.192	−0.281	N/A
	BMT-NoE&L	0.185	0.329	N/A
	BMT-E&NoL	0.183	0.592	7.30
	BMT-B	0.198	−0.425	N/A
	BMT-M	0.209	−0.456	N/A
	BMT	**0.181**	**0.605**	**5.43**
Retweets	THP	36,080.8	−9.01	N/A
	BMT-NoE&NoL	16,360.8	−111.13	N/A
	BMT-NoE&L	16,362.9	−110.95	N/A
	BMT-E&NoL	16,257.7	**−8.14**	28.92
	BMT-B	16,090.8	−16.21	N/A
	BMT-M	16,266.5	−11.19	N/A
	BMT	**15,825.8**	−11.28	**2.70**
Stack Overflow	THP	127.0	−0.373	N/A
	BMT-NoE&NoL	0.658	**−0.266**	N/A
	BMT-NoE&L	**0.643**	−0.277	N/A
	BMT-E&NoL	0.726	−0.339	17.81
	BMT-B	0.858	−0.718	N/A
	BMT-M	3.969	−0.505	N/A
	BMT	0.663	−0.358	**6.38**
Financial	THP	38.13	−1.826	N/A
	BMT-NoE&NoL	44.26	−11.843	N/A
	BMT-NoE&L	62.72	−11.759	N/A
	BMT-E&NoL	38.39	−2.104	7.39
	BMT-B	37.93	−1.848	N/A
	BMT-M	77.58	−1.796	N/A
	BMT	**37.92**	**−1.775**	**4.41**
911 Calls	THP	6,183.4	−7.190	N/A
	BMT-NoE&NoL	358.3	−17.662	N/A
	BMT-NoE&L	469.3	−41.818	N/A
	BMT-E&NoL	342.4	**−6.608**	28.49
	BMT-B	353.4	−6.832	N/A
	BMT-M	364.8	−6.835	N/A
	BMT	**339.6**	−6.883	**8.56**

RMSE, root mean squared error; LL, log-likelihood; CE, cross entropy; B, burstiness; M, memory coefficient. The bold value indicates the metric of the model with the best performance for each individual dataset.

improved performance of the model. Even in the prediction of one-hot discretized *B* & *M*, it can be observed that including *B* & *M* losses contributes to a reduction in cross entropy. No significant differences in performance were observed between the BMT-NoE&NoL and BMT-NoE&L models. This suggests that the incorporation of *B* & *M* losses is less significant in the absence of *B* & *M* embedding.

Summarizing the aforementioned findings, it is evident that both *B* & *M* embedding and *B* & *M* losses contribute to performance enhancement. Excluding either of these components would likely impede the attainment of a substantial performance improvement, comparable to that observed with the BMT model. If either of the *B* embedding or *M* embedding is omitted, a significant performance improvement comparable to that of the BMT model cannot be expected. This was substantiated by comparing the BMT model with the BMT-B and BMT-M models, which revealed the superior performance of the BMT model across all datasets. These results can also be observed in the training curves shown in Figure 6.

We also conducted experiments on mixed synthetic datasets, the results of which are presented in Table 3. The mixed synthetic datasets comprised a combination of three individual datasets: power-law, exponential, and self-correcting datasets. However, when separately examining the RMSE of event time and log-likelihood, the performance of the original BMT model appeared slightly inferior compared to some of the control BMT models, demonstrating an overall superior performance when considering both metrics together.

In summary, the BMT model demonstrates improved performance on heterogeneous data owing to its capability to capture heterogeneous characteristics through the embedding of *B* & *M*, combined with the inclusion of corresponding loss functions.

The BMT model has two limitations. First, in cases where the event sequence length is short, the incorporation of *B* and *M* into the BMT model may result in reduced effectiveness. This aspect originates from the statistical characteristics of *B* and *M*, because their meaningful representation is hindered by fluctuations and noise, particularly when the number of events is small. In the BMT model, during the calculation of sliding

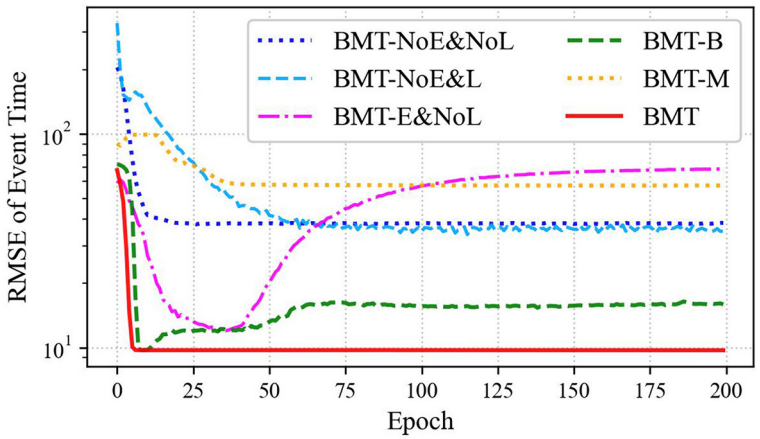

FIGURE 6
Training curves of RMSE for event times fitted on Financial Transaction datasets are presented for various BMT model scenarios: BMT-NoE&NoL, BMT-NoE&L, BMT-E&NoL, BMT-B, BMT-M, and the standard BMT model.

TABLE 3 **Performance evaluation results for the mixed synthetic datasets: power-law, exponential, and self-correcting datasets.**

Model	RMSE of time	LL	CE of BM
THP	1,338.65	−14.190	N/A
BMT-NoE&NoL	261.20	−9.015	N/A
BMT-NoE&L	**64.19**	−8.228	N/A
BMT-E&NoL	77.14	**−4.414**	12.593
BMT-B	64.74	−7.439	N/A
BMT-M	71.31	−10.476	N/A
BMT	66.98	−4.748	**6.269**

The bold value indicates the metric of the model with the best performance for each individual dataset.

B and *M* values, masking was applied to exclude the first three events. However, considering that temporal heterogeneity becomes a meaningful characteristic only when the length of the event sequence is sufficiently long, this limitation can be viewed as unavoidable.

The second limitation is the inability to consider event types, which will be addressed in future studies. To account for event types, it is necessary to reflect the correlation structure between inter-event times and event types to generate synthetic data and subsequently test the model using these data. In the context of performance enhancement, the improvement of the BMT model over the THP model can also be attributed to the fact that the BMT model does not embed event types. This allows the model to focus more on predicting the event times. Because the BMT-NoE&NoL model is analogous to a version of the THP model that does not consider event types, comparing the performance of the BMT-NoE&NoL model with the BMT model would provide a more equitable assessment. However, upon comparing the BMT-NoE&NoL model with the BMT model, it becomes evident that the BMT model exhibits superior performance across all datasets, except for StackOverflow.

6 Conclusion

Our study addresses the challenges presented by bursty temporal patterns in event sequences across various domains. By leveraging recent advancements in predicting event sequences using Transformer models based on the Hawkes process with self-attention mechanisms, we introduced a Burst and Memory-aware Transformer (BMT) model. This model effectively captures the nuances of burst patterns by embedding burstiness and memory coefficient within its self-attention module. The incorporation of a specialized loss function tailored for burstiness and memory coefficient further refines the model's predictive capabilities.

Through comprehensive numerical experiments conducted on a diverse array of synthetic and real-world datasets encompassing various scenarios, we validated the outstanding performance of the BMT model by comparing it with the existing models and control groups. This is particularly evident in scenarios involving heterogeneous data, such as power-law inter-event time distributions. Hence, the explicit consideration of burst-related parameters within the Transformer contributes to a deeper comprehension of complex event sequences, ultimately leading to an enhanced predictive performance. In future work, we will focus on integrating a multitude of insights from complex systems into the development of deep neural network models for temporal data.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found in the article/supplementary material. The synthetic data can be found here: https://github.com/bh0903lee/BMT-synthetic-datasets.

Author contributions

BL: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation,

Visualization, Writing—original draft, Writing—review & editing. J-HL: Formal analysis, Resources, Software, Writing—review & editing. SL: Formal analysis, Resources, Software, Writing—review & editing. CK: Formal analysis, Resources, Software, Writing—review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government (23ZS1100, Core Technology Research for Self-Improving Integrated Artificial Intelligence System).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Barabasi, A.-L. (2005). The origin of bursts and heavy tails in human dynamics. *Nature* 435, 207–211. doi: 10.1038/nature03459

Chan, H. K., Yang, D.-P., Zhou, C., and Nowotny, T. (2016). Burst firing enhances neural output correlation. *Front. Comput. Neurosci.* 10, 42. doi: 10.3389/fncom.2016.00042

Clauset, A., and Shalizi, C. R. (2009). Power-law distributions in empirical data. *SIAM Rev.* 51, 661–703. doi: 10.1137/070710111

Corral, Á. (2004). Long-term clustering, scaling, and universality in the temporal occurrence of earthquakes. *Phys. Rev. Lett.* 92, 108501. doi: 10.1103/PhysRevLett.92.108501

Cox, D. R. (1955). Some statistical methods connected with series of events. *J. R. Stat. Soc. Ser.* 17, 129–157. doi: 10.1111/j.2517-6161.1955.tb00188.x

Daley, D. J., and Vere-Jones, D. (2008). *An Introduction to the Theory of Point Processes: Volume II: General Theory and Structure*. New York, NY: Springer.

Dashevskiy, T., and Cymbalyuk, G. (2018). Propensity for bistability of bursting and silence in the leech heart interneuron. *Front. Comput. Neurosci.* 12, 5. doi: 10.3389/fncom.2018.00005

de Arcangelis, L., Godano, C., Lippiello, E., and Nicodemi, M. (2006). Universality in solar flare and earthquake occurrence. *Phys. Rev. Lett.* 96, 051102. doi: 10.1103/PhysRevLett.96.051102

Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., and Song, L. (2016). "Recurrent marked temporal point processes: Embedding event history to vector," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY: ACM), 1555–1564.

Goh, K.-I., and Barabáasi, A.-L. (2008). Burstiness and memory in complex systems. *Europhys. Lett.* 81, 48002. doi: 10.1209/0295-5075/81/48002

Hawkes, A. G. (1971). Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 83–90. doi: 10.1093/biomet/58.1.83

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Isham, V., and Westcott, M. (1979). A self-correcting point process. *Stochast. Process. Appl.* 8, 335–347. doi: 10.1016/0304-4149(79)90008-5

Jo, H.-H., Lee, B.-H., Hiraoka, T., and Jung, W.-S. (2019). Copula-based algorithm for generating bursty time series. *Phys. Rev. E* 100, 022307. doi: 10.1103/PhysRevE.100.022307

Jo, H.-H., Pan, R. K., Perotti, J. I., and Kaski, K. (2013). Contextual analysis framework for bursty dynamics. *Phys. Rev. E* 87, 062131. doi: 10.1103/PhysRevE.87.062131

Karsai, M., Jo, H.-H., and Kaski, K. (2018). *Bursty Human Dynamics*. Cham: Springer.

Kemuriyama, T., Ohta, H., Sato, Y., Maruyama, S., Tandai-Hiruma, M., Kato, K., et al. (2010). A power-law distribution of inter-spike intervals in renal sympathetic nerve activity in salt-sensitive hypertension-induced chronic heart failure. *BioSystems* 101, 144–147. doi: 10.1016/j.biosystems.2010.06.002

Kim, E.-K., and Jo, H.-H. (2016). Measuring burstiness for finite event sequences. *Phys. Rev. E* 94, 032311. doi: 10.1103/PhysRevE.94.032311

Kingman, J. F. C. (1992). *Poisson Processes, Vol. 3*. New York, NY: Oxford University Press Inc.

Leskovec, J., and Krevl, A. (2014). *Snap Datasets: Stanford Large Network Dataset Collection*. Ann Arbor, MI. Available online at: http://snap.stanford.edu/data

Lim, B., Arık, S. Ö., Loeff, N., and Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *Int. J. Forecast.* 37, 1748–1764. doi: 10.1016/j.ijforecast.2021.03.012

Ma, Y., Tang, Y., Zeng, Y., Ding, T., and Liu, Y. (2023). An n400 identification method based on the combination of soft-dtw and transformer. *Front. Comput. Neurosci.* 17, 1120566. doi: 10.3389/fncom.2023.1120566

Mei, H., and Eisner, J. M. (2017). The neural hawkes process: a neurally self-modulating multivariate point process. *Adv. Neural Inf. Process. Syst.* 30, 6754–6764.

Metzen, M. G., Krahe, R., and Chacron, M. J. (2016). Burst firing in the electrosensory system of gymnotiform weakly electric fish: mechanisms and functional roles. *Front. Comput. Neurosci.* 10, 81. doi: 10.3389/fncom.2016.00081

Nelsen, R. B. (2006). *An Introduction to Copulas*. New York, NY: Springer.

Omi, T., Ueda, N., and Aihara, K. (2019). Fully neural network based model for general temporal point processes. *Adv. Neural Inf. Process. Syst.* 32, 2120–2129.

Rasmussen, J. G. (2018). Lecture notes: temporal point processes and the conditional intensity function. *arXiv* [preprint]. doi: 10.48550/arXiv.1806.00221

Rolski, T., Schmidli, H., Schmidt, V., and Teugels, J. L. (2009). *Stochastic Processes for Insurance and Finance*. West Sussex: John Wiley & Sons.

Shahi, M., Van Vreeswijk, C., and Pipa, G. (2016). Serial spike time correlations affect probability distribution of joint spike events. *Front. Comput. Neurosci.* 10, 139. doi: 10.3389/fncom.2016.00139

Shchur, O., Biloš, M., and Günnemann, S. (2019). "Intensity-free learning of temporal point processes," in *International Conference on Learning Representations* (Addis Ababa).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Adv. Neural Inf. Process. Syst.* 30, 5998–6008.

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., et al. (2022). Transformers in time series: a survey. *arXiv* [preprint]. doi: 10.24963/ijcai.2023/759

Wheatland, M., Sturrock, P., and McTiernan, J. (1998). The waiting-time distribution of solar flare hard x-ray bursts. *Astrophys. J.* 509, 448. doi: 10.1086/306492

Xiao, S., Yan, J., Yang, X., Zha, H., and Chu, S. (2017). "Modeling the intensity function of point process via recurrent neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence* (Palo Alto, CA: AAAI Press), Vol. 31.

Yamauchi, S., Kim, H., and Shinomoto, S. (2011). Elemental spiking neuron model for reproducing diverse firing patterns and predicting precise firing times. *Front. Comput. Neurosci.* 5, 42. doi: 10.3389/fncom.2011.00042

Zeldenrust, F., Wadman, W. J., and Englitz, B. (2018). Neural coding with bursts—current state and future perspectives. *Front. Comput. Neurosci.* 12, 48. doi: 10.3389/fncom.2018.00048

Zhang, Q., Lipani, A., Kirnap, O., and Yilmaz, E. (2020). "Self-attentive hawkes process," in *International Conference on Machine Learning* (PMLR), 11183–11193.

Zhao, Q., Erdogdu, M. A., He, H. Y., Rajaraman, A., and Leskovec, J. (2015). "Seismic: a self-exciting point process model for predicting tweet popularity," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY: ACM), 1513–1522.

Zuo, S., Jiang, H., Li, Z., Zhao, T., and Zha, H. (2020). "Transformer hawkes process," in *International Conference on Machine Learning* (PMLR), 11692–11702.