A Multi-Channel Neural Graphical Event Model with Negative Evidence

Tian Gao,[†][*] **Dharmashankar Subramanian,**[†][*] **Karthikeyan Shanmugam,**[†]
Debarun Bhattacharjya,[†] **Nicholas Mattei**[§]
[†]Research AI, IBM T. J. Watson Research Center, Yorktown Heights, NY, USA
[§]Department of Computer Science, Tulane University, New Orleans, LA, USA
{tgao, dharmash, debarunb}@us.ibm.com
Karthikeyan.Shanmugam2@ibm.com, nsmattei@tulane.edu

Abstract

Event datasets are sequences of events of various types occurring irregularly over the time-line, and they are increasingly prevalent in numerous domains. Existing work for modeling events using conditional intensities rely on either using some underlying parametric form to capture historical dependencies, or on non-parametric models that focus primarily on tasks such as prediction. We propose a non-parametric deep neural network approach in order to estimate the underlying intensity functions. We use a novel multi-channel RNN that optimally reinforces the negative evidence of no observable events with the introduction of fake event epochs within each consecutive inter-event interval. We evaluate our method against state-of-the-art baselines on model fitting tasks as gauged by log-likelihood. Through experiments on both synthetic and real-world datasets, we find that our proposed approach outperforms existing baselines on most of the datasets studied.

Introduction

Event stream data is collected to explore the dynamics and behavior of a wide variety of systems including social networks, biochemical networks, electronic health records, and computer logs in data centers. In a multivariate event stream, events of multiple types (labels) occur at irregularly spaced time stamps on a common timeline. Event models seek to capture the joint stochastic dynamics of such event streams. Multivariate point processes and *conditional intensity functions* provide the mathematical framework for capturing event dynamics (Aalen, Borgan, and Gjessing 2008). In this framework, the instantaneous arrival rate of an event type at any point in time typically depends on the history of all historical event arrivals before that point in time.

The model fitting task for event stream data has a long history in machine learning, including prior work in temporal point process modeling (Gunawardana, Meek, and Xu 2011; Parikh, Gunawardana, and Meek 2012; Weiss and Page 2013; Bhattacharjya, Subramanian, and Gao 2018). It is fundamentally different from the prediction task as it is unsupervised since no ground truth (graph) model is given. Various models have been proposed in the literature to capture history-dependent arrival rates including graphical event models like proximal graphical event models (Bhattacharjya, Subramanian, and Gao 2018), forest-based point processes (Weiss and Page 2013), piecewise-constant conditional intensity models (Gunawardana, Meek, and Xu 2011), and Poisson networks (Rajaram, Graepel, and Herbrich 2005), and others like non-homogeneous Poisson processes (Goulding, Preston, and Smith 2016), continuous time noisy-or (CT-NOR) models (Simma et al. 2008), and Poisson cascades (Simma and Jordan 2010). One drawback is that all these approaches make assumptions about the parametric form of the corresponding model, and this is challenging in practice without first-hand knowledge of the underlying data generating process.

On the other hand, researchers have also proposed sequential deep learning techniques for event data sets (Xiao et al. 2017b) such as recurrent marked temporal point processes (RMTPP) (Du et al. 2016) and neural Hawkes processes (Mei and Eisner 2017). These models use a recurrent neural network to capture the history dependency of the conditional intensity function. Note that these methods are semi-parametric; beyond just the network weights, they make functional assumptions for how to translate the hidden states of the network to a corresponding event arrival rate. They also show a remarkable similarity to the Hawkes process where each historical event exerts an additive influence on conditional intensity, and which decays over time. In neural Hawkes (Mei and Eisner 2017), the cell state of the network decays exponentially. In RMTPP (Du et al. 2016), there is a single event arrival rate whose epochs carry a label to capture multiple event types, and this single rate is an exponential function of the sum of a baseline intensity along with additive influences from the past. While such representations can be powerful and expressive, they may not adequately learn general-purpose forms of history-dependence. For example, consider history-dependence that doesn't involve exponential decay; examples include piecewise constant intensities (Gunawardana, Meek, and Xu 2011; Bhattacharjya, Subramanian, and Gao 2018), or processes involving time lags with delayed excitation or inhibition, or if different event types follow varying time scales, etc. Non-simple-decay and delayed, nonlinear intensity functions are not satisfactorily addressed by existing works.

In this paper, we propose a non-parametric deep learning approach to model multivariate event data sets in continuous-time. We seek to learn history-dependent conditional intensity

[*]indicates equal contribution

functions in a fully data-driven, non-parametric manner, i.e. using only network weights and activation functions, and via learning a suitable representation of all available (strict) histories. Firstly, we note that the inter-event interval is an important source of information for modeling event data, i.e. not only is the presence of an event at a point in continuous time an important signal for learning, but also the absence of an event at a point in time. We present a deep learning model that reinforces this negative evidence. We also note that real-life events exhibit multiple time scales. Historical sequences of fixed length that are used in classical recurrent neural networks (RNNs) do not suffice to address this issue. We present a multi-scale, multi-channel sequential representation that is sensitive to the base rate of various event types. Lastly, real-life data sets exhibit complex history dependencies like delayed excitation due to events of either the same or another type. We use different types of spatial (across types) and temporal (across time) attention models. Our work is motivated by the framework of graphical event models (GEMs) (Gunawardana and Meek 2016) where the historical event epochs of a set of parents affect the instantaneous rate of a child point process.

Contribution. We make the following contributions: (1) We propose a simple, yet effective, non-parametric way to approximately capture the continuous-time variation of historical influence on conditional intensity by exploiting the negative evidence from each successive inter-event duration. (2) We develop an efficient multi-scale, multi-channel internal state representation, which lets us align our architecture with graphical event models. (3) We also propose a spatio-temporal attention model to capture whose histories (among event labels), and which points in time are most influential toward determining the instantaneous arrival rate for any chosen label. The resulting proposed **multi-channel neural graphical event model (MCN-GEM)** is thereby similar to structure learning in the graphical modeling literature. We demonstrate non-trivial gains in evaluating point process log likelihood estimates on test data over state-of-the-art models.

Multi-Channel Neural Graphical Event Model

Following the notation in graphical event models (Gunawardana and Meek 2016; Bhattacharjya, Subramanian, and Gao 2018), we denote an event dataset as $D = \{(l_i, t_i)\}_{i=1}^{N}$, where t_i is the occurrence time of the i^{th} event, $t_i \in \mathbb{R}^+$, and l_i is an event label corresponding a finite set $\mathcal{L}$ of possible labels (types), where $|\mathcal{L}| = M$. We assume a time ordered dataset where at most one event may occur at any point in continuous time. So we have $t_i < t_j$ for $i < j$, with initial time $t_0 = 0 \leq t_1$ and end time $t_{N+1} = T \geq t_N$, where T is the total time period. We denote the strict history relative to any point t in continuous time as $\mathcal{H}_t$, and this is defined as the sequence of event epochs before time t, i.e. $\{(t_i, l_i)|t_i < t\}$. Lastly, $\lambda_t^k|\mathcal{H}_t$ denotes the instantaneous conditional intensity of event type k in label set $\mathcal{L}$ at time t. It governs the history-dependent instantaneous rate of occurrence of event type k at time t. Note that we use the term epoch to denote an event epoch, i.e. an event arrival on the timeline. Figure 1 shows the overall system of our proposed MCN-GEM, which consists of three main components: a recurrent neural network to represent the continuous time evolution, the introduction of negative evidence between two consecutive arrival epochs, and the multi-channel view with spatial and temporal attention. Below we discuss each component in detail.

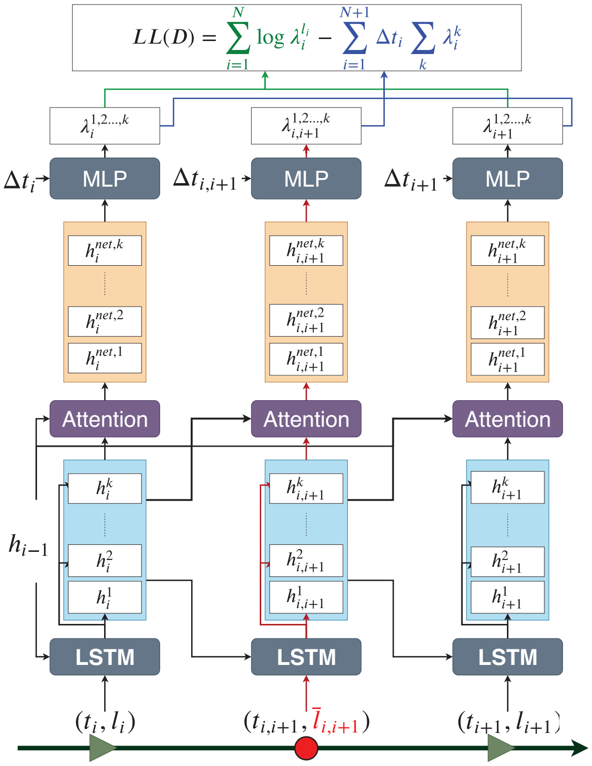

Figure 1: System overview of the proposed MCN-GEM model with negative evidence $\bar{l}$.

Continuous Time in Deep Event Models

Given a dataset D, a graphical event model (GEM) $\mathcal{G}$ is a directed graph with nodes $\mathcal{L}$ and edges $\mathcal{E}$. It defines a family of marked point processes whose likelihood of data D is (Gunawardana and Meek 2016):

$$p(D|\theta) = \sum_{l \in \mathcal{L}} \sum_{i=1}^{n} \lambda_{t_i}^{l_i \cdot \mathbf{1}_{l_i}} \cdot e^{\int_{t_{i-1}}^{t_i} \sum_{k=1}^{M} \lambda_\tau^k | \mathcal{H}_\tau d\tau}$$

where the indicator function $\mathbf{1}_{l_i} = 1$ if $l_i = l$ and zero otherwise.

One may write the log-likelihood of the event data over interval $[0, T]$ using the conditional intensity functions as:

$$\text{logL}(D) = \sum_{i=1}^{N} \log \lambda_{t_i}^{l_i} | \mathcal{H}_{t_i} - \sum_{i=1}^{N+1} \int_{t_{i-1}}^{t_i} \sum_{k=1}^{M} \lambda_\tau^k | \mathcal{H}_\tau d\tau \quad (1)$$

Our objective is to train a deep neural network to produce $\lambda_t^k|\mathcal{H}_t$ for each type k. We use a sequence modeling approach with recurrent neural networks and long short term memory

(LSTM) cells. The sequence of tokens that we feed into the LSTM network is the temporally ordered event sequence D, where each token corresponds to an event arrival, i.e., a label and a time stamp. To model continuous time, each token is represented in the raw input as a concatenation of its one-hot encoded event label (1-out-of-M) and its continuous-valued time stamp, similar to previous work (Du et al. 2016; Mei and Eisner 2017). The internal states of the recurrent LSTM cell evolve in response to each current raw input. We note that there are alternative choices for modeling continuous time, such as contextualizing the event label embedding with a time mask (Li, Du, and Bengio 2017).

For an event sequence D as per our notation, l_0 denotes a token that marks the beginning of the sequence and l_{N+1} denotes a token that marks the end of the sequence. Then for each event epoch i in sequence D, we have $0 \leq i < N+1$, and we compute the hidden state h_i^k for each event label k in the augmented set $\mathcal{L}$ as:

$$h_i^k = \text{LSTM}_k([\text{Emb}(l_i), t_i]; h_{i-1}^k), \quad \forall k \in \mathcal{L}$$

where Emb denotes the embedding matrix for label l_i. Embedding consists of one embedding layer on top of one-hot encoding of labels. h_{-1}^k is initialized to be all-zero vectors. For practical computation reasons, we find that sharing the LSTM parameters among different event labels k is sufficient, i.e., $\text{LSTM}_k = \text{LSTM}$.

Basic Model. Figure 2 (a) shows a basic model for history dependent conditional intensity. The black lines show the common timeline on which two consecutive epochs (t_i, l_i) are shown. The upper half shows the recurrent cells unrolled in time,and the bottom half shows a neural network layer, namely the λ-network, to produce the instantaneous conditional intensity vector λ. The recurrent cell receives two inputs at each sequential token (event arrival): the raw input x_i which is the concatenation of the one-hot encoded event label l_i and the time stamp t_i, and the penultimate value of the internal states (h_{i-1}, c_{i-1}). The sub-network that produces the conditional intensity $(\lambda(t)|\mathcal{H}_t)$ takes two inputs: the most recent internal states that are strictly before t, i.e. in correspondence with the most recent event (say at $t_i < t$), and the duration $(t - t_i)$.

Modeling Continuous Time History with Negative Evidence

Motivation on New Interval State Representation. We note that the internal states (h_i, c_i) in the basic model evolve discretely across tokens while staying fixed in between tokens. In event modeling, the inter-event duration between any two consecutive events is just as important as the event arrivals themselves, as evident from the integral terms in Equation 1. As such, it is important that the internal states of the sequential model also reflect the continuous passage of time in between successive events. The neural Hawkes process (Mei and Eisner 2017) proposes capturing this continuous variation using an exponential decay function, motivated by the classical Hawkes process (Hawkes 1971). However, this raises the question of whether an exponential form of decay is always the best choice, and whether there is a more non-parametric alternative that adapts to data. For example, consider proximal graphical event models (Bhattacharjya, Subramanian, and Gao 2018) and the more general family of piece-wise constant intensity models (Gunawardana, Meek, and Xu 2011) that don't exhibit exponential decay. We propose the use of ”fake epochs” to reinforce the negative evidence of no observable events within each inter-event dead-space as shown in Figure 2(b).

Fake Epochs as Negative Evidence. We introduce an auxiliary $(M+1)^{\text{th}}$ label into our label set $\mathcal{L}$, and refer to it as the ”fake label”. We then interject within each inter-event interval a certain number K of fake epochs (i.e. with label $M+1$) that are spread uniformly in time over the interval. Note that K is a hyperparameter, and Figure 2(b) shows one fake epoch $(K = 1)$ introduced at $\tilde{t}_{i,i+1}$, which is the center of $[t_i, t_{i+1}]$. The fake event epoch then participates in the recurrent computations like any other real event epoch. It allows using the LSTM dynamics to further evolve the internal states within each inter-event interval, albeit in a discrete manner i.e. at each such fake event epoch on the timeline. The resulting finer sequence of internal states is then a summary of all the event trace history as well as the passage of time in the intervening dead-space intervals. Note how the internal state has evolved to $(\tilde{h}_i, \tilde{c}_i)$ in the interior of the interval. The fake event epochs also allow us to compute the integral terms in Equation 1. We take a numerical quadrature procedure using the fake epoch time stamps as the sampling time points for the quadrature. We note that while the entire inter-event interval corresponds to negative evidence in theory, the proposed finite approximation suffices in practice as shown later in the experimental section. Lastly, while the conditional intensity corresponding to the fake label doesn't enter Equation 1 in any way, we use it to regularize the learning objective via a target-label reconstruction term described later.

A similar intent of using fake epochs was pursued in a different graphical model (Gopalratnam, Kautz, and Weld 2005), which proposed to model the negative evidence of no observable events. It also has some similarity with generative adversarial training (Ganin et al. 2016; Xiao et al. 2017a). The fake epochs with a fake event label are akin to adversarial sample points relative to the real set of labels. Using the fake epochs to regularize the learning via target label reconstruction, or via binary classification of real-versus-fake makes it similar to adversarial training.

A Multi-Channel View for Spatial and Temporal Attention

In multivariate event data, the different labels may have different arrival rates. Further, they may mutually influence the arrival rates of each other in label-specific ways in the sense of graphical event models. It is therefore desirable to model a label-specific hidden state. We achieve this by associating each label with a corresponding partition of a single hidden state vector. We achieve this by choosing the hidden state dimension to be an integer multiple of the number of labels, i.e. $m(M+1)$ for some positive integer m, and effectively realize an m-dimensional hidden state for each label in $\mathcal{L}$. These label-specific sub-vectors are selectively channeled for

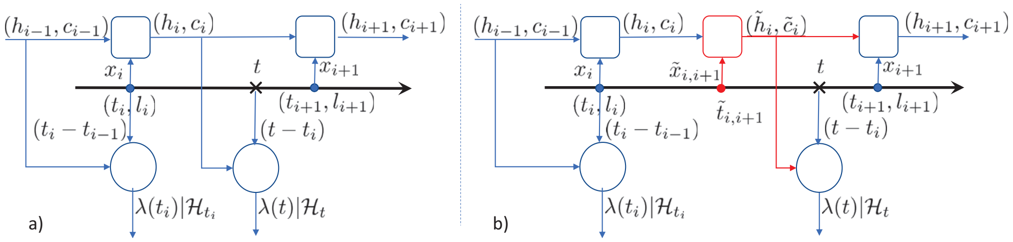

Figure 2: Deep sequential model for conditional intensity: a) A basic model, b) A model with fake event epochs to reinforce negative evidence.

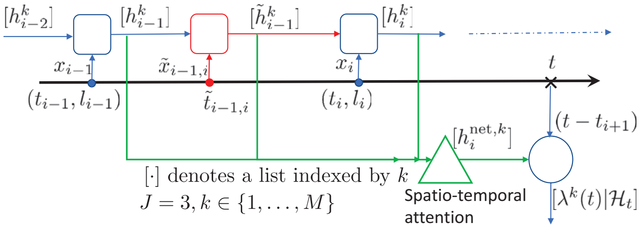

Figure 3: Spatio-temporal attention for deep graphical event models.

computing the label-specific rates through the network layer in Figure 2. Such a multi-channel view also enables modeling temporal and spatial attention within the λ-network.

Spatial attention is a way to model inter-label dependence, as in graphical event models, and temporal attention is a way to model the lagged dependence on parental event history. For example, in a piece-wise constant intensity model (Gunawardana, Meek, and Xu 2011), the active basis functions consist of lagged intervals relative to any point in time. We achieve these two types of dependence by maintaining a memory bank $\mathcal{M}$ of historical label-specific hidden states that span the most recent J event arrivals. Note that the above multi-channel view gives us a total of $M * J$ hidden state sub-vectors in $\mathcal{M}$, i.e. one hidden-state sub-vector for each real label and for each time-stamp corresponding to the most recent J event arrivals. These raw hidden states in the memory bank are combined using an attention mechanism into a net hidden state that becomes input for the lambda-network. Figure 3 shows this for $J = 3$, where $[h^k_i]$ in the figure denotes a list of hidden states indexed by label k, and at time t_i. The attention block produces a corresponding list of net hidden states $[h^{\text{net},k}_i]$ which enter the λ-network to produce conditional intensities for each label at time t.

By adding an attention layer after h^k_i, intensity rates can be shown to credit the relevant event occurrences in history. Our memory bank $\mathcal{M}_i$ (for each epoch i) has size $J \times M$ to stores the raw hidden states of each of M channels from the most recent J epochs relative to i, i.e. from $(i - J)$ till $(i-1)$. Note that J is a constant hyperparameter in the model. The net attentive hidden state is given as,

$$h^{\text{net},k}_i = \tanh(W_c[c^k_i, h^k_i]), \quad \forall k \in \mathcal{L}$$

where c_i is the context vector at epoch i, and is computed as a weighted average of the raw hidden states in $\mathcal{M}_i$ The weighting is done by an alignment vector α^k_i, i.e., $c^k_i = \sum_m \alpha^k_{im} h_m$, where h_m is each raw label-specific hidden state in the memory bank $\mathcal{M}_i$ Note that index m runs over each of the JM hidden states in $\mathcal{M}_i$ thereby giving both temporal and spatial attention. W_C represents weights to compute new hidden representation given attention. The attention weight α^k_{im} is derived by comparing the current raw hidden state h^k_i to each raw hidden state h_m in the memory bank at time i:

$$\alpha^k_{im} = \text{align}(h^k_i, h_m) = \frac{exp(score(h^k_i, h_m)}{\sum_m exp(score(h^k_i, h_m))}$$

where $score(h^k_i, h_m) = (h^k_i)^T h_m$. We note that other types of attentions can be used as well in the above spatio-temporal attention model.

Training

We then use two feed-forward layers to learn intensity rate $\lambda^k_{t_{i+1}}$ given the net hidden state $h^{\text{net},k}_i$ and time interval $\Delta t_{i+1} = t_{i+1} - t_i$:

$$\lambda^k_{t_{i+1}}|\mathcal{H}_{t_i} = \sigma_2(f_2(\sigma_1(f_1([h^{\text{net},k}_i, \Delta t_{i+1}])))), \quad \forall k \in \mathcal{L}$$

where f_1 and f_2 are feed-forward neural layers. We use a ReLU and softplus activation function for $\sigma_1(\cdot)$ and $\sigma_2(\cdot)$, respectively. The Softplus activation ensures a positive conditional intensity, which is a standard component and has been used in previous deep models (Mei and Eisner 2017).

To train MCN-GEM, we use the same LL function as in Equation 1 with the assumption of constant intensity in between two consecutive events, real or fake, and this leads to:

$$LL(D) = \sum_{i=1}^{N} \log \lambda^{l_i}_{t_i}|\mathcal{H}_{t_i} - \sum_{i=1}^{N+1} \Delta t_i \sum_{k=1}^{M} \lambda^k_{t_i}|\mathcal{H}_{t_i} \qquad (2)$$

where $\Delta t_i = t_i - t_{i-1}$ is the time interval since the last event.

In addition, since our model is the most non-parametric compared to other related work, we find that adding two regularization terms helps with generalization. First, we consider the target prediction loss $\mathcal{L}_p$ of the next event label l_{i+1}

given $\lambda^k_{t_{i+1}}$, for which we use a cross entropy loss between ground truth l_{i+1} and softmax$(\lambda_{t_{i+1}})$. This classification loss also emphasizes the adversarial nature between real and fake epochs. Second, we add another term $\mathcal{L}_w$ which penalizes the $\mathcal{L}_2$-norm of the weights on f_1 and f_2. Hence, the overall regularized objective for training is as follows:

$$\mathcal{L}_{\text{train}} = LL(D) + \lambda_p \mathcal{L}_p + \lambda_w \mathcal{L}_w \tag{3}$$

For the model fitting computation task, the overall likelihood objective for training is computed as Equation 3. The testing loss is simply computed as $\text{LL}(D)$, since our aim is to gauge how well the model fits the data.

Empirical Evaluation

We compare our methods with state-of-the-art algorithms for model fitting of event streams. We compare our proposed neural method with a recent model in the GEM class, PGEM (Bhattacharjya, Subramanian, and Gao 2018), and the most relevant neural baseline Neural Hawkes Process (NHP) (Mei and Eisner 2017). Some other works (Xiao et al. 2017b; Du et al. 2016) also use neural networks but do not perform the model fitting task. We implement MCN-GEM in Pytorch, PGEM in Python, and use publicly available code for NHP[1]. Please refer to the appendix for further implementation details of MCN-GEM, especially on choices of hyper parameters. Our proposed MCN-noF refers to the version of our model where no fake epochs are introduced; we show this to highlight the effect of fake epochs. If not otherwise specified, MCN-GEM uses one fake epoch. We use log likelihood as the metric to evaluate model fitting and bold the best performing approach.

For our experiments, we divide an event dataset into $70\%-30\%$ train-test splits. For data involving single event streams, we perform the split based on time, i.e. select events in the first 70% of the entire duration in the train set. For data involving multiple streams, we split by randomly selecting a subset of 70% of the streams in the train set. All shown results are those evaluated on the test set. All methods are compared in the same experimental setting.

Synthetic Datasets. We first conduct experiments using event streams based on proximal graphical event models (PGEMs). The data is generated following the same sampling procedure described in Bhattacharjya, Subramanian, and Gao (2018). Specifically, for each node, we first sample the number of its parents K, the set of parents, windows, and then intensity rates in that order. Please refer to the supplement material for exact details and generating parameters.

We generate 5 PGEMs, each with 5 nodes. We use default values to generate the models as provided in the supplementary material of Bhattacharjya, Subramanian, and Gao (2018). For each model, 10 event streams are produced with a synthetic PGEM data generator, up to $T = 1000$. Table 1 shows the performance of the algorithms on the test sets for the 5 PGEM models. We observe that when PGEM is the data-generating model, all methods including neural-based models can recover a log likelihood (LL) very close to the PGEM learner. What is particularly striking is that with the introduction of a fake epoch, MCN-GEM performs much better than the other models, even the PGEM learner, usually by 20% to 35%. We note that the state-of-the-art PGEM learner (Bhattacharjya, Subramanian, and Gao 2018) involves a greedy coordinate ascent approach for learning the optimal windows and graphical structure, which allows for improvement using our non-parametric neural model with a limited number of data. The quadrature approximation that we use (see Equation 3) across the augmented set of consecutive inter-event intervals is effectively a fine piecewise-constant conditional intensity model, thereby giving MCN-GEM enough expressive power to effectively approximate the true generating PGEM.

Dataset	PGEM	NHP	MCN-noF	MCN-GEM
PGEM1	-6207.3	-6248.8	-6268.1	**-4764.2**
PGEM2	-8285.4	-8289.4	-8292.0	**-5739.2**
PGEM3	-7933.1	-8023.0	-8177.7	**-5714.8**
PGEM4	-8853.2	-8895.7	-8904.3	**-5608.5**
PGEM5	-7811.0	-7842.5	-7912.5	**-5601.6**

Table 1: Average LL score for Synthetic Datasets.

Dataset	PGEM	NHP	MCN-noF	MCN-GEM
Argentina	-3150.6	-3181.9	-3559.5	**-2677.9**
Brazil	-3865.7	-3822.3	-4242.3	**-3329.0**
Venezuela	-1663.2	**-1567.4**	-1915.9	-1655.1
Colombia	-1095.1	-1128.5	-1244.5	**-1029.3**
Mexico	-1927.6	-1936.5	-2375.7	**-1756.7**

Table 2: Average LL score for ICEWS Datasets.

Political News Datasets. A real world example of numerous, asynchronous events on a timeline are socio-political world events. We use the Integrated Crisis and Early Warning System (ICEWS) political event dataset (O'Brien 2010) to test our model. This dataset is a set of dyadic events, i.e., *X does Y to Z*, encoded as a set of over 100 actors and 20 high level actions from the Conflict and Mediation Event Observations (CAMEO) ontology (Gerner et al. 2002). The events in ICEWS take the form of "The Police (Brazil) has Verbal Conflict with Protesters (Brazil)." Following the same preprocessing as Bhattacharjya, Subramanian, and Gao (2018), we restrict our attention to a 4 year time period from Jan 1, 2012 to Dec 31, 2015 and use 5 countries, 5 actor types, and 5 types of actions.

Table 2 compares model fit on these datasets across methods. Performances of PGEM and NHP are similar on these datasets, and MCN-GEM with no fake epochs performs worse than NHP. It is possibly harder to learn without some parametric grounding in these datasets. However, with the introduction of fake epochs, MCN-GEM again performs better than the other models by 5% to 15%, with the exception of Venezuela, on which NHP performs the best. For Venezuela, we suspect that MCN-GEM's proficiency in learning from history is slightly detrimental as there are discrepancies be-

[1] https://github.com/HMEIatJHU/neurawkes

tween train and test sets; the data exhibits a significant increase in event occurrences in recent years (test set) compared to previous years (train set).

Healthcare Dataset. Following Mei and Eisner (2017), we also test algorithms on the MIMIC-II dataset, which includes electronic health records from various patients with clinical visit records in Intensive Care Unit for 7 years. Each patient has a sequence of hospital visit events, and each event records its time stamp and diagnosis, which serves as the event label.

Since the PGEM learner cannot handle datasets with unknown labels during testing time, we also create a clean version of MIMIC, MIMIC-short (MIMIC-s), ensuring that MIMIC-s train and test have a common label set. We compare our method with PGEM on MIMIC-s, and with NHP on the original MIMIC. Table 3 shows again that MCN-GEM with the fake epochs outperforms both NHP and PGEM by $15 \sim 20\%$ for the log likelihood computation.

	NHP	MCN-noF	MCN-GEM
MIMIC	-1165.0	-1380.4	**-837.4**
	PGEM	MCN-noF	MCN-GEM
MIMIC-s	-677.8	-704.4	**-586.8**

Table 3: Log likelihood for MIMIC Dataset.

Ablation Study

To gain more insights of the contribution of different components, we also study the effectiveness of each component of the proposed model via an ablation study, namely the effectiveness of multi-channel vs. single channel, the usage of attention vs. no attention, and the impact of attention memory bank sizes for negative evidence. As shown in Table 4 below (showing LL with standard deviation), we conducted an ablation study on different model components on ICEWS Argentina. We tested single channel (SC) vs. multi-channel (MC), one fake epoch (1F) vs. no fake (NoF), and different memory bank sizes (M=1,2,3,5,10). We use neural Hawkes (NHP) as the baseline. We repeat the experiments 5 times with 5 different random seeds, reporting their average LL along with their standard deviation.

Impact of Multi-Channel Modeling. As shown in the top row of Table 4, the single-channel (SC noF and SC 1F) versions always perform worse than that of multi-channel versions (MC noF and MC 1F). It shows the importance of multi-channel modeling, by letting the hidden states evolved separately for each channel.

Impact of the Memory Bank Size. As shown in the top row of Table 4, without the attention (equivalent to the memory size of 0), the performance of the proposed model (MC-1F) is worse than that of NHP. The second row of Table 4 shows the performance of MCN with memory size of 1 (1M), 2 (2M), 3(3M), 5(5M), and 10 (10M), and all outperform NHP. In addition, memory size of 3 is the best performing in Argentina, while the larger memory sizes could degrade the performance.

Impact of the Number of Fake Epochs

We study the impact of the number of fake epochs on the performance of MCN-GEM through a study on the ICEWS Argentina dataset. We vary the number of fake epochs, from 0 to 5, to see whether adding more fake epochs would further improve the LL. We use 1F to indicate the usage of 1 fake epoch, 2F for 2 fake epochs, etc. As shown in Table 5, introducing one fake epoch has a large gain in LL. Introducing many more leads to minimal improvements, and can even hurt the performance. We note that without fake epochs, the computation of integral in Equation 2 is assumed to be piece-wise constant in each inter-event interval. This would perform badly in general, such as when there is decay, growth, or more complex temporal variations in conditional intensity. Even with 1 fake epoch, MCN-GEM appears effective in approximating such variation. Introducing more fake epochs leads to a subtle trade-off. While they allow a finer approximation in the quadrature-sum for the second term, they also lead to an additional imbalance in the relative proportions of real and fake epochs. This may result in reduced intensity rates of real events at real epochs due to fitting their rates better at the fake epochs. Fake epochs are also related to the windows and basis functions in GEMs (Gunawardana and Meek 2016). Instead of manually providing a set of basis functions, MCN-GEM uses the data to adaptively infer the start and expiration of new basis functions. It is very simple yet effective to do so in a non-parametric way.

To visualize the impact of fake epochs on the learned intensity rates, we plot them from MCN-GEM $-$noF and MCN-GEM with one fake epoch in the ICEWS Argentina dataset. We show only Event Type 1's intensity rates in Figure 4, but the trends hold for all other event types. In Figure 4, the top figure shows the rates, in blue lines, of Event Type 1 from noF over time and the bottom figure shows rates from MCN-GEM. A red triangle in both figures indicates the rate at the occurrence of Event Type 1, and a yellow cross indicates the rate at the occurrence of all other event types. As one can see, with the introduction of fake epochs, MCN-GEM produces higher rates at event occurrences and lower rates at the dead space between occurrences. This results in a sharper intensity rate landscape. Also, it is interesting to observe that some event occurrences lead to the increase of Event Type 1's rate while many others do not.

Graph Visualization of the Attention

One other advantage of using attention is that it can enable visualization of the relationships among the variables as a graph. We use the average attention of all event channels across time to compute the graph connection. Specifically, let A be a graph adjacency matrix with element A_{kq}, the kth row and qth column of A, indicating event k's influence on the occurrence of event q. For clarity of notation, let $\alpha^k_{ijq} = \alpha^k_{im}$ be the computed attention for kth channel at time t_i. $q \in M$ and $j \in J$ represent the (parental) event type q at time t_{i-j}.

	NHP	SC noF	SC 1F	MC noF	MC 1F
Argentina	-3154.8 ± 25.4	-4323.8 ± 35.4	$-3839.4.9 \pm 25.4$	-4282.2 ± 58.1	-3302.6 ± 28.1
	MC 1F 1M	MC 1F 2M	MC 1F 3M	MC 1F 5M	MC 1F 10M
Argentina	-2841.0 ± 23.5	-2693.9 ± 27.0	-2682.6 ± 29.3	-2871.8 ± 126.9	-2947.3 ± 190.0

Table 4: Ablation study of MCN-GEM model components: LL performance on ICEWS Argentina.

noF	1F	2F	3F	4F	5F
-3559.5	-2677.9	**-2652.3**	-2772.1	-2803.1	-2881.0

Table 5: Study of the number of fake epochs vs. LL performance on ICEWS Argentina.

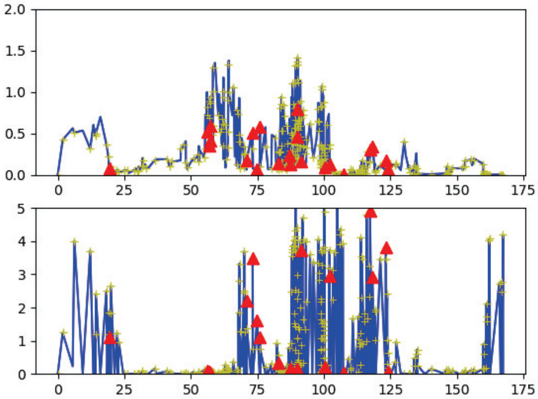

Figure 4: Visualization of lambdas (conditional intensity rates) for Event Type 1 in ICEWS Argentina Dataset. The top shows rates from MCN-GEM −noF, and the figure shows rates from MCN-GEM. Red triangles indicate the occurrence of Event Type 1, yellow crosses indicate the occurrence of all other real events.

$$A_{kq} = \frac{1}{T \times J} \sum_{i=1}^{T} \sum_{j=1}^{J} \alpha_{ijq}^{k}$$

We then threshold A_{kq} to remove small numerical values and obtain $\tilde{A}$. We plot $\tilde{A}$ in Figure 5 for ICEWS Argentina, where each node is a different event type and edges represent $\tilde{A}$. We use 0.01 as the threshold. We include the node label in the appendix. Some example edges learned include: chain 36 -¿ 40 -¿ 23, which indicates that when the head of the Argentina government cooperates with the Brazil government, it often leads to internal conflicts between him/her and other branches of the government, and this in turn leads to citizens' unhappiness with the head of the Argentina government. Chain 55 -¿ 62 indicates that if citizens are in conflict with protestors, citizens will likely cooperate with the police.

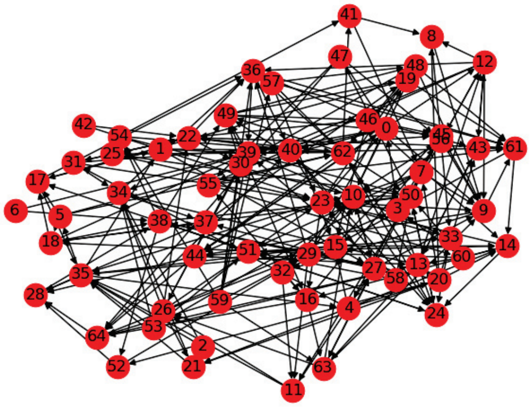

Figure 5: Visualization of attention graph in ICEWS Argentina Dataset.

Conclusion

We have introduced a new multi-scale multi-channel neural GEM with two-dimensional attentions for modeling event sequences. Our model exploits the negative evidence of no observable events in each successive inter-event duration by introducing fake epochs, which eliminates the need to assume specific functional forms. This makes our approach practically appealing with respect to approximately capturing the variation of hidden states in continuous time in a nonparametric manner. Our model combines the framework of GEMs and the modeling power of deep neural networks. On synthetic and benchmark model fitting datasets, our method outperforms other state-of-the-art models by a significant amount, demonstrating that this is a promising approach for modeling event stream data. Alongside the lambda-network in Figure 2, a parallel integral-lambda-network could be used to learn the integral terms in Equation 1 with a constraint that connects these two networks and their parameters. This would provide an alternative to using a quadrature sum for the integral terms.

The focus of the proposed model with negative evidence is to fit a model for learning dependencies in event stream data. It requires full visibility into the event sequence for the purposes of placement or sampling of the fake epochs. This may not be true in prediction tasks, where the exact use of negative evidence presents an interesting future direction.

Acknowledgement

We thank Hongyuan Mei from JHU for providing baseline code and results, and anonymous reviewers for their constructive and helpful feedback.

References

Aalen, O. O.; Borgan, O.; and Gjessing, H. K. 2008. *Survival and Event History Analysis: A Process Point of View*. New York, NY, USA: Springer Science & Business Media.

Bhattacharjya, D.; Subramanian, D.; and Gao, T. 2018. Proximal graphical event models. In *Advances in Neural Information Processing Systems (NIPS)*.

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*, 1555–1564. ACM.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* 17(1):2096–2030.

Gerner, D. J.; Schrodt, P. A.; Yilmaz, O.; and Abu-Jabr, R. 2002. Conflict and mediation event observations (CAMEO): A new event data framework for the analysis of foreign policy interactions. *International Studies Association (ISA) Annual Convention*.

Gopalratnam, K.; Kautz, H.; and Weld, D. S. 2005. Extending continuous time Bayesian networks. In *Proceedings of the Twentieth National Conference on Artificial Intelligence - Volume 2*, 981–986. AAAI Press.

Goulding, J.; Preston, S.; and Smith, G. 2016. Event series prediction via non-homogeneous Poisson process modelling. In *Proceedings of the Sixteenth IEEE Conference on Data Mining (ICDM)*, 161–170.

Gunawardana, A., and Meek, C. 2016. Universal models of multivariate temporal point processes. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 556–563.

Gunawardana, A.; Meek, C.; and Xu, P. 2011. A model for temporal dependencies in event streams. In *Advances in Neural Information Processing Systems (NIPS)*, 1962–1970.

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

Li, Y.; Du, N.; and Bengio, S. 2017. Time-dependent representation for neural event sequence prediction. *arXiv preprint arXiv:1708.00065*.

Mei, H., and Eisner, J. M. 2017. The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*, 6754–6764.

O'Brien, S. P. 2010. Crisis early warning and decision support: Contemporary approaches and thoughts on future research. *International Studies Review* 12:87–104.

Parikh, A. P.; Gunawardana, A.; and Meek, C. 2012. Conjoint modeling of temporal dependencies in event streams. In *Proceedings of Uncertainty in Artificial Intelligence Workshop on Bayesian Modeling Applications*.

Rajaram, S.; Graepel, T.; and Herbrich, R. 2005. Poisson-networks: A model for structured point processes. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 277–284.

Simma, A., and Jordan, M. I. 2010. Modeling events with cascades of Poisson processes. In *Proceedings of the Twenty-Sixth International Conference on Uncertainty in Artificial Intelligence (UAI)*, 546–555.

Simma, A.; Goldszmidt, M.; MacCormick, J.; Barham, P.; Black, R.; Isaacs, R.; and Mortie, R. 2008. CT-NOR: Representing and reasoning about events in continuous time. In *Proceedings of the Twenty-Fourth International Conference on Uncertainty in Artificial Intelligence (UAI)*, 484–493.

Weiss, J. C., and Page, D. 2013. Forest-based point process for event prediction from electronic health records. In *Machine Learning and Knowledge Discovery in Databases*, 547–562.

Xiao, S.; Farajtabar, M.; Ye, X.; Yan, J.; Song, L.; and Zha, H. 2017a. Wasserstein learning of deep generative point process models. In *Advances in Neural Information Processing Systems*, 3247–3257.

Xiao, S.; Yan, J.; Yang, X.; Zha, H.; and Chu, S. M. 2017b. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, volume 17, 1597–1603.