# Decomposable Transformer Point Processes


**Aristeidis Panos**
University of Cambridge
ap2313@cam.ac.uk



## Abstract

The standard paradigm of modeling marked point processes is by parameterizing the intensity function using an attention-based (Transformer-style) architecture. Despite the flexibility of these methods, their inference is based on the computationally intensive thinning algorithm. In this work, we propose a framework where the advantages of the attention-based architecture are maintained and the limitation of the thinning algorithm is circumvented. The framework depends on modeling the conditional distribution of inter-event times with a mixture of log-normals satisfying a Markov property and the conditional probability mass function for the marks with a Transformer-based architecture. The proposed method attains state-of-the-art performance in predicting the next event of a sequence given its history. The experiments also reveal the efficacy of the methods that do not rely on the thinning algorithm during inference over the ones they do. Finally, we test our method on the challenging long-horizon prediction task and find that it outperforms a baseline developed specifically for tackling this task; importantly, inference requires just a fraction of time compared to the thinning-based baseline.


## 1 Introduction

Continuous-time event sequences are commonly found in real-world scenarios and applications such as financial transactions [3], communication in a social network [34], and purchases in e-Commerce systems [16]. This abundance of data for discrete events occuring at irregular intervals has lead to an increasing interest of the community in the last decade to marked temporal point processes which are the standard way of modeling this kind of data.

Historically, Hawkes processes [14] and Poisson processes [7] have been extensively applied to various domains such as finance [13], seismology [15], and astronomy [1]. Despite their elegant mathematical framework and interpretability, the strong assumptions of the models reduce their flexibility and fail to capture the complex dynamics of real-world generating processes.

Advances in deep learning have allowed the incorporation of neural models like LSTMs [17] or recurrent neural networks (RNN) into temporal point processes [5, 12, 24, 26, 29, 35, 38]. As a result, these models are able to learn more complex dependencies and attain superior performance than Hawkes/Poisson processes. Recently, the introduction of the (self-) attention mechanism [36] to modeling temporal point processes [41, 42, 44] has led to new state-of-the-art methods with extra flexibility.

Despite the advantages of these neural-based models, their dependence on modeling the conditional intensity function creates limitations for both training and inference [35]. Training usually requires a Monte Carlo approximation of an integral that appears in the log-likelihood. [29] proposed a method to circumvent this approximation; however, the main shortcomings remained as discussed in [35]. More importantly, inference is based on the thinning algorithm [21, 22] which is computationally intensive and sensitive to the choice of intensity function. To deal with these downsides, [35] parameterized the conditional distribution of the inter-event times by combining a log-normal mixture density network


38th Conference on Neural Information Processing Systems (NeurIPS 2024).


with an RNN. The model's performance is comparable to that of the other intensity-based methods which use RNN/LSTM architecture but still inferior to the Transformer-based methods.

A more recent work [30] has referred to the decomposition of the log-likelihood of a marked point process [6] to parameterize the distribution of marks given the time and history and the distribution of times given the history. This decomposition, as with [35], eliminates the need for the thinning algorithm and additional approximations, while offering a rigorous, yet flexible framework for defining different distributions for occurrence times and marks. [30] used two different parametric models for each distribution, and their results for the time prediction task, despite the simplicity of their framework, were competitive or superior to neural-based baselines.

Inspired by the state-of-the-art performance of the Transformer-based architectures and the computational efficiency/flexibility of the intensity-free models, we develop a model for marked point processes that combines the advantages of these two methodologies. Our contributions are summarized below:

- We propose a novel model that is defined by two distributions: a distribution for the marks based on a Transformer architecture and a simple log-normal mixture model for the inter-event times which satisfies a simple Markov property.
- Through an extensive experimental study, we show the efficiency of our model in the next-event prediction task and the suitability of the intensity-free models for correctly predicting the next occurrence time over the methods relied on the thinning algorithm.
- To the best of our knowledge, we are the first to experimentally show the limitations of the thinning algorithm on the predictive ability of the neural point processes.
- We test our model on the more challenging long-horizon prediction task and we provide strong evidence that we can achieve better results in a fraction of time compared to models that have been specifically designed to solve this task and, uncoincidentally, depend on the thinning algorithm.

## 2   Background

A marked temporal point process (MTPP), observed in the interval $(0, T)$, is a stochastic process whose realizations are sequences of discrete events occurring at times $0 < t_1 < \ldots < t_N < T$ with corresponding event types (or marks) $k_1, \ldots, k_N$, where $k_i \in \{1, \ldots, K\}$. The entire sequence is denoted by $\mathcal{H}_T = \{(t_1, k_1), \ldots, (t_N, k_N)\}$. The process is fully specified by the conditional intensity function (CIF) of the event of type $k$ at time $t$ conditioned on the event history $\mathcal{H}_{t_i} = \{(t_j, k_j) \mid t_j < t_i\}, \lambda_k^*(t) := \lambda_k(t \mid \mathcal{H}_{t_i}) \geq 0, t > t_i$; we use the asterisk $*$ to denote the dependence on $\mathcal{H}_{t_i}$. The CIF is used to compute the infinitesimal probability of event $k$ occurring at time $t$, i.e. $\lambda_k^*(t)dt = \mathbb{P}(t_{i+1} \in [t, t + dt], k_{i+1} = k \mid t_{i+1} \notin (t_i, t), \mathcal{H}_{t_i})$. The log-likelihood of such an autoregressive multivariate point process is given by [14, 22]

$$\mathcal{L}(\mathcal{H}_T) = \sum_{i=1}^{N} \lambda_{k_i}^*(t_i) - \sum_{k=1}^{K} \int_0^T \lambda_k^*(t)\, dt. \tag{1}$$

Modeling the intensity function by a flexible model and then learning its parameters by maximizing Eq. (1) has been the standard approach of many works [5, 12, 24, 26, 29, 35, 38, 41, 42, 44].

An equivalent way of deriving the log-likelihood in (1) without the use of $\lambda_k^*(t)$ is by following the decomposition of a multivariate distribution function in [6] (expression 2), expressed as

$$\mathcal{L}(\mathcal{H}_T) = \sum_{i=1}^{N} \left\{ \log p^*(k_i \mid t_i) + \log f^*(t_i) \right\} + \log\left(1 - F(T \mid \mathcal{H}_{t_N})\right), \tag{2}$$

where $p^*(k \mid t_i) := p(k \mid t_i, \mathcal{H}_{t_i})$[1] and $f^*(t) := f(t \mid \mathcal{H}_{t_i})$ are the conditional probability mass function (CPMF) of the event types and the conditional probability density function (CPDF) for the occurrence times, respectively. $F(t \mid \mathcal{H}_{t_i}) = \int_{t_i}^t f^*(t)dt$, $\forall t > t_i$ is the cumulative distribution function of $f^*(t)$. The last term in (2) is the logarithm of the survival function that expresses

---

[1]The notation of $^*$ is slightly different compared to $\lambda_k^*$ but the definition remains consistent.

2

the probability that no event occurs in the interval $(t_N, T)$. The relation between $\lambda_k^*(t)$ and the density/PMF is given by $\lambda_k^*(t) = \frac{f^*(t)p^*(k|t)}{1-F(t|\mathcal{H}_t)}$; see Section 2.4 in [33].

We can represent the temporal part of the process in terms of the inter-event times $\tau_i := t_i - t_{i-1} \in \mathbb{R}_+, t_0 = 0$; the two representations are isomorphic and the relation between the conditional PDF of the inter-event time $\tau_i$ until the next event and the conditional intensity function is given by

$$g^*(\tau_i) := g^*(\tau_i \mid \mathcal{H}_{t_i}) = \sum_{k=1}^{K} \lambda_k^*(t_{i-1} + \tau_i) \exp\left(-\sum_{k=1}^{K} \int_0^{\tau_i} \lambda_k^*(t_{i-1} + x) dx\right) = f^*(t_i).$$

## 3 Decomposable Transformer Point Processes

To develop our proposed framework *Decomposable Transformer Point Process (DTPP)*, we adopt the decomposition in (2) and model $g^*(\tau)$ and $p^*(k \mid t)$, separately. Despite the advantages of modeling the intensity function and the arguments in favor of this [9], we believe that modeling the probability density/mass function offers not only the same benefits as modeling the intensity function as discussed in [35], but, more importantly, it allows us not to depend on the thinning algorithm during inference. The technical details of each model are described in the next two sections.

### 3.1 Distribution of Marks

The conditional distribution of the event types is parameterized by a continuous-time Transformer architecture as the one described in [41]. More specifically, for any pair of events $(t, k)$, we evaluate an embedding $\mathbf{h}_k(t) \in \mathbb{R}^D$ based on the history $\mathcal{H}_t$. Assuming an $L$-layer architecture, $\mathbf{h}_k(t)$ is given by the concatenation of the embedding of each individual layer, i.e. $\mathbf{h}_k(t) = [\mathbf{h}_k^{(0)}(t); \mathbf{h}_k^{(1)}(t); \ldots; \mathbf{h}_k^{(L)}(t)]$. The embedding of the base layer $\mathbf{h}_k^{(0)}(t)$ is independent of time and it is learned by a simple weight vector for each mark, i.e. $\mathbf{h}_k^{(0)}(t) := \mathbf{h}_k^{(0)} \in \mathbb{R}^{D^{(0)}}$. The embedding of layer $\ell \in \{1, \ldots, L\}$ for $(t, k)$ is defined as

$$\mathbf{h}_k^{(\ell)}(t) := \mathbf{h}_k^{(\ell-1)}(t) + \tanh\left(\sum_{(t_i,k_i)\in\mathcal{H}_t} \frac{\mathbf{v}_{k_i}^{(\ell)}(t_i)\, \alpha_{k_i}^{(\ell)}(t_i; t, k)}{1 + C}\right) \in \mathbb{R}^{D^{(\ell)}}, \tag{3}$$

where $C > 0$ is the normalization constant given by $C = \sum_{(t_i,k_i)\in\mathcal{H}_t} \alpha_{k_i}^{(\ell)}(t_i, t, k)$ and the unnormalized attention weight is

$$\alpha_{k_i}^{(\ell)}(t_i; t, k) = \exp\left(\frac{1}{\sqrt{D}} \mathbf{k}_{k_i}^{(\ell)}(t_i)^\top \mathbf{q}_k^{(\ell)}(t)\right) > 0. \tag{4}$$

The operation of the non-linear activation function $\tanh$ is element-wise and $D = \sum_{\ell=0}^{L} D^{(\ell)}$. The query, key, and value vectors $\mathbf{q}_k^{(\ell)}(t), \mathbf{k}_k^{(\ell)}(t)$, and $\mathbf{v}_k^{(\ell)}(t)$, respectively, can be computed by using the embedding of the previous layer and the corresponding weight matrices $Q^{(\ell)}, K^{(\ell)} \in \mathbb{R}^{D\times(D+D^{(\ell-1)})}, V^{(\ell)} \in \mathbb{R}^{D^{(\ell)}\times(D+D^{(\ell-1)})}$ as follows,

$$\mathbf{q}_k^{(\ell)}(t) = Q^{(\ell)} X_t^\ell, \ \ \mathbf{k}_k^{(\ell)}(t) = K^{(\ell)} X_t^\ell, \ \ \mathbf{v}_k^{(\ell)}(t) = V^{(\ell)} X_t^\ell, \tag{5}$$

where $X_t^\ell = \left[\mathbf{z}(t); \mathbf{h}_k^{(\ell-1)}(t)\right] \in \mathbb{R}^{D+D^{(\ell-1)}}$. By $\mathbf{z}(t) \in \mathbb{R}^D$, we denote a temporal embedding of time defined as

$$[\mathbf{z}(t)]_d = \begin{cases} \cos\left(t/10^{\frac{4(d-1)}{D}}\right), & \text{if d is odd}, \\ \sin\left(t/10^{\frac{4d}{D}}\right), & \text{if d is odd}, \end{cases} \tag{6}$$

where $d = 0, \ldots, D - 1$. This encoding is the same as in [44] with small differences than the one used in [41] where we found empirically the former to work slightly better than the latter. For a more detailed discussion regarding the architecture of the model and how it compares to previous Transformer-based methods, see Appendix A in [41]. Finally, we note that for extra model flexibility, multi-head self-attention can be easily obtained by the three equations in (5).

3

Having computed the top-layer embeddings $\mathbf{h}_k(t)$ for all $k = 1, \ldots, K$, we model the conditional PMF $p^*(k \mid t)$ as

$$p^*(k \mid t) = \frac{\exp\left(\mathbf{w}_k^\top \mathbf{h}_k(t)\right)}{\sum_{l=1}^K \exp\left(\mathbf{w}_l^\top \mathbf{h}_l(t)\right)}, \tag{7}$$

where $\mathbf{w}_k$ are the learnable classifier weights. As is typical for these architectures, to avoid any data leakage from future events, we mask all future events $(t_i, k_i)$ where $t < t_i$ and only use previous events for computing these embeddings.

## 3.2 Distribution of Inter-Event Times

For the modeling of the inter-event times, since they always take positive values, we choose a mixture of log-normal distributions whose parameters depend on the value of the previously seen mark. Specifically, given that the previous occurred mark is $k$, the PDF[2] of the next inter-event time $\tau$ is defined as

$$g^*(\tau) = g(\tau \mid k) = \sum_{m=1}^M w_m^{(k)} \frac{1}{\tau s_m^{(k)} \sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{\log \tau - \mu_m^{(k)}}{s_m^{(k)}}\right)^2\right), \tag{8}$$

where $\{w_m^{(k)}\}_{m=1}^M \in \Delta^M$ are the mixture weights, $\{\mu_m^{(k)}\}_{m=1}^M \in \mathbb{R}^M$ are the mixture means, and $\{s_m^{(k)}\}_{m=1}^M \in \mathbb{R}_+^M$ are the standard deviations, for any $k = 1, \ldots, K$. The log-normal mixture has several desirable features that justifies our choice: (i) it efficiently approximates distributions in low dimensions such as 1-d distributions of inter-event times [23, 35] while satisfying a universal approximation property that provides theoretical guarantees regarding its approximation ability [8], (ii) closed-form moments are available and can be used for predicting the next time; for instance, the mean of the distribution is given as the weighted average of each of the log-normal means, i.e.

$$\mathbb{E}_g^{(k)}[\tau] = \sum_{m=1}^M w_m^{(k)} \exp\left(\mu_m^{(k)} + \frac{(s_m^{(k)})^2}{2}\right), \tag{9}$$

(iii) learning the small number of parameters $\{w_m^{(k)}, \mu_m^{(k)}, s_m^{(k)}\}_{m=1}^M$ can be done in a fraction of time using fast off-the-shelf implementations based on the EM algorithm [10]. Finally, note that the dependence of the model only on the most recent mark implies a Markov property since we do not need the entire history $\mathcal{H}_{<t}$ to define our distribution.

At first glance, this assumption might seem restrictive when it comes to capturing the complex dynamics of the process. Nevertheless, this assumption holds only for $g^*$ while $p^*$ is modeled by the flexible Transformer architecture that models the full history up to the current time $t$. Hence, we do not sacrifice any modeling power at all to achieve efficiency. On the contrary, we can maintain both modeling power and computational efficiency due to the decomposition in (2) and the chosen models in (8) and (7). Moreover, as our extensive experiments on the real-world data show, this assumption provides a robust predictive model which is less prone to overfitting compared to more flexible neural net architectures since the Markov property can act as a strong regularizer.

## 3.3 Training and Prediction

The parameters $\{w_m^{(k)}, \mu_m^{(k)}, s_m^{(k)}\}_{m,k}$, of $g^*(\tau)$ and the parameters $\{\mathbf{w}_k, \mathbf{h}_k^{(0)}, Q^{(\ell)}, K^{(\ell)}, V^{(\ell)}\}_{\ell,k}$ of $p^*(k \mid t)$ can be estimated by maximizing the log-likelihood in (2) using any stochastic gradient method. A crucial benefit of using the decomposition in (2) is that it permits us to learn the above parameters separately as follows:

$$\{w_m^{(k)^*}, \mu_m^{(k)^*}, s_m^{(k)^*}\}_{m,k} = \operatorname{argmax} \sum_{i=1}^N \log g^*(\tau_i) + \log\left(1 - G(T \mid \mathcal{H}_T)\right) \tag{10}$$

$$\{\mathbf{w}_k^*, \mathbf{h}_k^{(0)^*}, Q^{(\ell)^*}, K^{(\ell)^*}, V^{(\ell)^*}\}_{\ell,k} = \operatorname{argmax} \sum_{i=1}^N \log p^*(k_i \mid t_i), \tag{11}$$

---

[2]To avoid notation cluttering, we use $\tau$ and $k$ in lieu of $\tau_i$ and $k_{i-1}$, respectively, to define the PDF.

4

where $G(T \mid \mathcal{H}_T) = \int_0^{T-t_N} g^*(\tau)d\tau$ . This is a major difference from previous work where the parameters of the neural net parameterizing the conditional intensity function had to be learned all at once. By dividing the main objective function into two sub-objectives, since there is no parameter-sharing between the two models, we can maximize the two objectives independently, and thus, having an easier optimization task than maximizing a single set of parameters of a given objective, such as (1).

The trained models $g^*$ and $p^*$ can now be used to predict either the time/type of the next event (next-event prediction) or the next $P > 1$ events (long-horizon prediction). For the next-event-prediction, the predicted time $\hat{t}$ given the history $\mathcal{H}_{t_i}$ is computed by using the mean of the appropriate mixture of log-normals while the corresponding predicted type of this event $\hat{k}$ is evaluated based on $\mathcal{H}_{t_i}$ and the true time $t_{i+1}$, i.e.

$$\hat{t} = t_i + \mathbb{E}_g^{(k_i)}[\tau], \qquad \hat{k} = \underset{k}{\arg\max}\, p^*(k \mid t_{i+1}). \tag{12}$$

The above procedure is based on the minimum Bayes risk (MBR) principle [24] which aims to predict the time and type that minimizes the expected loss. This is an average $L_2$ loss in the case of time prediction (deriving a root mean squared error) and an average 0-1 loss for the type prediction (deriving an error rate).

For the long-horizon prediction task [11, 40], we need to predict a sequence of events where, unlike the next-event prediction task, we do not have access to the true time when we predict the next event type. This could potentially lead to a cascading error effect due to the autoregressive nature of the models designed for the less challenging task of next-event prediction. This is because after an error is made in the sequence of the predictions, it cannot be corrected, and thus the error accumulates and affects all subsequent predictions. We argue that this pathology can be alleviated by using a model of times that provides accurate and robust predictions given the history. This assumption is verified experimentally in Section 5.2. To generate a predicted sequence, we require the trained models $g()$ and $p()$ to sequentially predict events as in (12). Since we have no access to the true time $t_{i+1}$, we use as a proxy the prediction $\hat{t}$ to predict $\hat{k}$ in turn. After the prediction of the new event, we append it to the history and then we repeat the same step given the updated history until we generate a sequence of $P$ events. The exact procedure is described in Algorithm 1 in the Appendix.

The main advantage of Algorithm 1 over other methods [40] that are based on the thinning algorithm is its computational efficiency. The algorithm is fully parallelizable, and it can produce single steps in parallel for a batch of event sequences. This is not possible for thinning-based methods that require one to consider single sequences each time [40]. Consequently, our method is able to generate sequences orders of magnitude faster, which is verified by our experiments. Unlike other competitors [40] that are based on the thinning algorithm and therefore require random sampling, our algorithm is fully deterministic; for comparison the thinning algorithm is described in Algorithm 2 of the Appendix.

## 4   Related Work

The decomposition in (2) has been used in the past to provide both expressive and interpretable models [27, 30]. For instance, [30] model the mark distribution with a parametric model, inspired by the exponential intensity function of a Hawkes process and the time distribution with a single log-normal; however, they use the mode instead of the mean of the distribution for predicting the time of the next event. They learn the parameters of their models separately, as we describe in (10) and (11), but they use Variational Inference [4] to learn the parameters of $p^*$. They attain competitive results in terms of next-time prediction, but the model lacks the flexibility of a Transformer-based architecture, as our experiments show.

[35] is another work that takes advantage of the mixture of log-normal distributions to model the distribution of the inter-event times. The model is based on an RNN architecture that produces a fixed-dimensional embedding of the event history, which is used to generate the parameters of the mixture model, and the same embedding is employed to define the CPMF of the marks. In our case, we use a Transformer architecture to obtain this history embedding, which is utilized by the CPMF, exclusively. Finally, the proposed model in [35] assumes that the marks are conditionally independent of the time given the history, which is not the case for our framework, as is evident in (2).

5
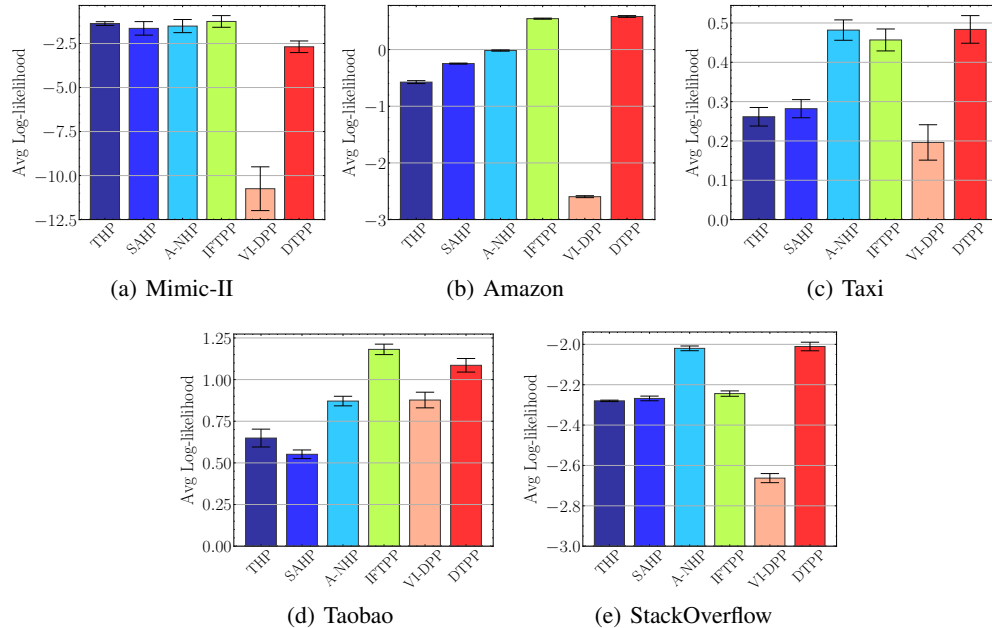
Figure 1: Goodness-of-fit evaluation over the five real-world datasets. We compare our DTPP model against five strong baselines. Results (larger is better) are accompanied by 95% bootstrap confidence intervals.

Finally, the CPMF of the marks for our DTPP model shares the same architecture as the Attentive Neural Hawkes Process (A-NHP) [41]. Nevertheless, they use it to model the CIF while in our case we model $p^*$.

## 5   Experiments

We considered two different tasks to assess the predictive performance of our proposed method: Goodness-of-fit/next-event prediction and long-horizon prediction. We compared our method DTPP to several strong baselines over five real-world datasets and three synthetic ones. Description and summary statistics for all datasets used in this section are given in Appendix A.1. For the competing methods, we used their published implementations; more details are given in A.3. Experimental details not available in this section can be found in Appendix A. Our framework was implemented with PyTorch [31] and scikit-learn [32]; the code is available at https://github.com/aresPanos/dtpp.

### 5.1   Goodness-of-Fit / Next-Event Prediction

We evaluated our **DTPP** model to determine how well it generalizes and predicts the next event given the history on the held-out dataset. For comparison, we used five state-of-the-art baselines where the three of them model the CIF using Transformers, while the other two model the CPDF of inter-event times and the CPMF of marks (see Section 4). The CIF-based baselines are the **Transformer Hawkes Process (THP)** [44], the **Self-Attentive Hawkes Process (SAHP)** [42], and the **Attentive Neural Hawkes Process (A-NHP)** [41]. The CPDF-based ones are the **Intensity-Free Temporal Point Process (IFTPP)** [35], and the **VI-Decoupled Point Process (VI-DPP)** [30].

We fit the above six models on a diverse collection of five popular real-world datasets, each with varied characteristics: **MIMIC-II** [19], **Amazon** [28], **Taxi** [37], **Taobao** [43], and **StackOverlfow-V1** [20, 41]. Training details are given in Appendix A.2.

**Goodness-of-Fit.**   Figure 1 shows the average log-likelihood for each model on the held-out data of the five real-world datasets. Our DTPP model consistently outperforms the simple parametric

6

Table 1: Performance comparison between our model DTPP and various baselines in terms of next-event prediction. The root mean squared error (RMSE) measures the error of the predicted time of the next event, while the error rate (ERROR-%) evaluates the error of the predicted mark given the true time. The results (lower is better) are accompanied by 95% bootstrap confidence intervals. $^{\dagger}$,$^{\triangleleft}$,$^{\triangleright}$ denote the CIF-based methods, the CPDF-based methods that use a single model, and the ones using a seperate model, respectively.

| | AMAZON | | TAXI | | TAOBAO | | STACKOVERFLOW-V1 | |
|---|---|---|---|---|---|---|---|---|
| METHODS | RMSE | ERROR | RMSE | ERROR | RMSE | ERROR | RMSE | ERROR |
| THP$^{\dagger}$ | 0.62 $\pm$0.03 | 65.06 $\pm$1.04 | 0.37 $\pm$0.02 | 8.55 $\pm$0.65 | 0.13 $\pm$0.02 | 41.43 $\pm$0.78 | 1.32 $\pm$0.03 | 53.86 $\pm$0.76 |
| SAHP$^{\dagger}$ | 0.58 $\pm$0.01 | 62.58 $\pm$0.32 | 0.28 $\pm$0.04 | 8.37 $\pm$0.43 | 2.26 $\pm$0.59 | 45.63 $\pm$0.58 | 1.93 $\pm$0.04 | 53.00 $\pm$0.32 |
| A-NHP$^{\dagger}$ | 0.42 $\pm$0.01 | 65.94 $\pm$0.31 | 0.29 $\pm$0.02 | 7.67 $\pm$0.44 | 1.44 $\pm$0.53 | 43.96 $\pm$0.55 | 1.18 $\pm$0.01 | 52.17 $\pm$0.32 |
| IFTPP$^{\triangleleft}$ | 0.41 $\pm$0.05 | 64.08 $\pm$0.34 | 0.39 $\pm$0.09 | 8.17 $\pm$0.46 | 0.33 $\pm$0.00 | 41.92 $\pm$0.54 | 1.91 $\pm$0.10 | 53.68 $\pm$0.30 |
| VI-DPP$^{\triangleright}$ | 0.38 $\pm$0.00 | 65.49 $\pm$0.34 | 0.11 $\pm$0.02 | 9.49 $\pm$0.42 | 0.07 $\pm$0.00 | 41.89 $\pm$0.57 | 1.68 $\pm$0.04 | 55.06 $\pm$0.32 |
| DTPP$^{\triangleright}$ | **0.12** $\pm$0.00 | **59.06** $\pm$0.35 | **0.08** $\pm$0.01 | **7.12** $\pm$0.40 | **0.05** $\pm$0.00 | **40.12** $\pm$0.59 | **1.07** $\pm$0.03 | **50.41** $\pm$0.31 |

VI-DPP, indicating the flexibility of using the self-attention mechanism to model the CPDF. Except Mimic-II, DTPP achieves the highest or the second highest log-likelihood across the remaining datasets. Therefore, the separate parameterization of $p^*$ and $g^*$ does not hurt performance compared to the models with a common set of learnable parameters. Finally, notice that DTPP outperforms all the CPDF-based methods on average, while the two CPDF-based methods that employ deep learning architectures, i.e., DTPP and IFTPP, exhibit better performance than the CIF-based baselines. A plausible explanation is that the log-likelihood computation for the CIF-based baselines requires Monte Carlo integration, which could cause approximation errors; for the CPDF-based methods, this computation is exact. A-NHP is the clear winner among the CIF-based methods, as also shown in [41].

**Next-Event Prediction.** We evaluate the predictive capacity of all models by predicting each event $(t_i, k_i)$ given its history $\mathcal{H}_{t_i}, i = 2, \ldots, N$ on held-out data. Event time prediction is measured by root mean squared Error (RMSE) and event type prediction by error rate; Table 1 summarizes the results. DTPP outperforms all the baselines in both tasks. The wider performance gaps in RMSE between our model and the other baselines justify our choice of a inter-event distribution satisfying a Markov property; this result also implies that we do not need long event histories to capture the dynamics of these datasets. We also compare the average performance between CIF-based and CPDF-based (excluding VI-DPP) methods. We see that for the CIF-based baselines the average RMSE is 0.58 and the average error rate is 35.39% while for the CPDF-based ones we have 0.95 and 37.0%, respectively. These results support our argument that the thinning algorithm tends to harm the time prediction accuracy; they also highlight the efficiency of using a separate model for the inter-event times. Additional results on Mimic-II can be found in Appendix B.1.

**Synthetic datasets.** To extra investigate the capabilities of our model in a more controlled manner, we created a dataset by generating sequences from a randomly initialized SAHP model. Although, each event has strong dependence on its history, Figure 2 shows that our model approximates the true log-likelihood as well as A-NHP. Moreover, DTPP's mixture model is more accurate than the thinning-based A-NHP in time prediction. Moreover, we found that the only case that DTPP was significantly outperformed by A-NHP was on a synthetic dataset generated by a 1-d Hawkes process. Since no event types are present we only use a single mixture of log-normals which apparently is the wrong model for this data. The results are illustrated in Figure 4 of the Appendix.

## 5.2 Long-Horizon Prediction

To test the performance of our model for this task, we followed the experimental setup of [40]. From the same work, we used the proposed **HYPRO**, which is the state-of-the-art method for the long-horizon prediction task, to compare with DTPP. HYPRO is a globally normalized model that aims to address cascading errors that occur in auto-regressive and locally normalized models, such as the models in Section 5.1. HYPRO and DTPP are based on the same Transformer architecture of

7
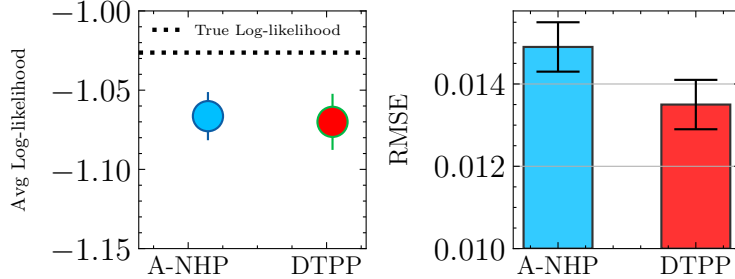
Figure 2: Performance comparison between DTPP and A-NHP over the SAHP-Synthetic dataset.

A-NHP so the main difference is that HYPRO is a CIF-based method, and thus, it requires the thinning algorithm to sample sequences. For our experiments, we use the distance-based regularization variant of HYPRO with a Multi-NCE loss as this method attains the best results in [40]. As DTPP and HYPRO share the same Transformer architecture, so we used the exact same hyperparameters for fair comparison. Note that even in this case, HYPRO has more than double number of parameters compared to DTPP since HYPRO requires an extra Transformer to model the energy function used for global normalization. More details on HYPRO training and hyperparameters can be found in the Appendix A.

(a) Taxi  (b) Taobao  (c) StackOverflow-V2
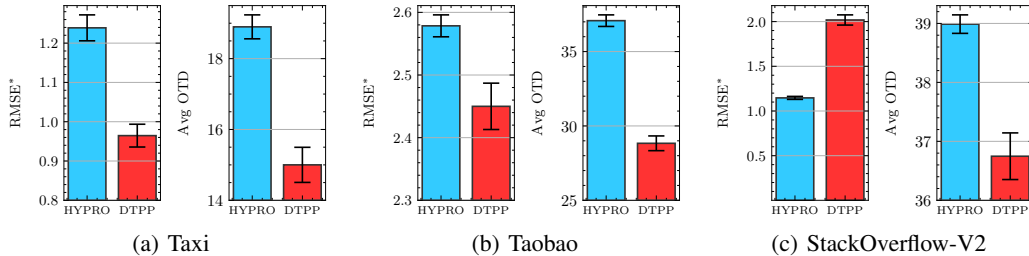
Figure 3: Performance comparison over the three real-world datasets measured by RMSE⋆ and average OTD (lower is better). The reported results for HYPRO are based on 16 weighted samples, i.e. $M = 16$ for Algorithm 2 in [40].

We used three of the previous real-world datasets for evaluation because of their long sequences. These are **Taxi**, **Taobao**, and **StackOverflow-V2** [20, 40]. For each dataset, our goal is to predict the last 20 events in a sequence, denoted by $\mathcal{H}_P$, given the history; that is, $P = 20$ in Algorithm 1. As is typical for the long-horizon prediction, the standard scores used for evaluating the model's performance are the The optimal transport distance (**OTD**) [25] and the long-horizon RMSE (**RMSE**⋆) [40].

In Figure 3, we see that our DTPP method outperforms HYPRO across all datasets in terms of average OTD and RMSE. HYPRO achieves a lower RMSE score only in StackOverflow. These results provide corroborating evidence on our argument that the thinning algorithm might negatively affect the performance of a neural point process even in the case of globally normalized models as HYPRO. It is also evident that a locally normalized CPDF-based model such as DTPP is much more robust against the cascading error which CIF-based methods are vulnerable [41]. We believe that this robustness stems from the accurate predictions of the log-normal mixture model.

Apart from the predictive performance, we investigated the time required for the two methods to generate all the predicted sequences of the held-out dataset. Since HYPRO's inference time is heavily relied on the thinning algorithm and a hyperparameter that indicates the number of proposal sequences (denoted as $M$ in [40]), we conducted an ablation study for a varied number of proposals to investigate the inference time and performance of HYPRO against DTPP. For HYPRO's inference time, apart from the prediction time, we included the time required to generate the noise sequences so the energy function can be trained on. The inclusion of this time is justified by the importance the energy function has as a component of the framework, and it can be seen as a necessary pre-inference

8

Table 2: Performance comparison between our model DTPP and HYPRO for the long-horizon prediction task. For HYPRO, we use $\{2, 4, 8, 16, 32\}$ weighted proposals (Algorithm 2 in [40]). We report the average optimal transport distance (avg OTD) and the time (in minutes) required for predicting all the long-horizon sequences of the held-out dataset (lower is better). "Params" denotes the number ($\times 10^3$) of trainable parameters of each method. We include error bars based on five runs.

| METHODS | PARAMS | TAXI | | TAOBAO | | STACKOVERFLOW-V2 | |
|---|---|---|---|---|---|---|---|
| | | AVG OTD | TIME | AVG OTD | TIME | AVG OTD | TIME |
| HYPRO-2 | | 20.35 $\pm0.24$ | 44.81 $\pm0.01$ | 39.73 $\pm0.37$ | 43.53 $\pm0.03$ | 39.84 $\pm0.12$ | 46.32 $\pm0.01$ |
| HYPRO-4 | | 19.86 $\pm0.18$ | 47.31 $\pm0.04$ | 38.93 $\pm0.22$ | 46.57 $\pm0.08$ | 39.57 $\pm0.17$ | 48.84 $\pm0.06$ |
| HYPRO-8 | 850 | 19.25 $\pm0.30$ | 52.61 $\pm0.20$ | 37.30 $\pm0.48$ | 53.10 $\pm0.17$ | 39.37 $\pm0.30$ | 54.14 $\pm0.22$ |
| HYPRO-16 | | 18.90 $\pm0.34$ | 62.36 $\pm0.27$ | 37.08 $\pm0.39$ | 65.63 $\pm0.44$ | 38.99 $\pm0.16$ | 64.82 $\pm0.34$ |
| HYPRO-32 | | 18.81 $\pm0.16$ | 81.30 $\pm0.23$ | 36.96 $\pm0.19$ | 89.11 $\pm0.71$ | 38.84 $\pm0.20$ | 83.05 $\pm0.45$ |
| DTPP | 400 | **15.00** $\pm0.30$ | **0.01** $\pm0.00$ | **28.83** $\pm0.26$ | **0.17** $\pm0.01$ | **36.75** $\pm0.40$ | **0.03** $\pm0.00$ |
| SPEEDUP | | $8,130\times$ | | $524.2\times$ | | $2768.3\times$ | |

step. However, for completeness, we compute only the prediction time of HYPRO and report it in Table 6 of the Appendix.

The results are presented in Table 2 where we measure the performance using the average OTD; a similar table for RMSE* is in Appendix B.3. We see that our parallelizable framework takes advantage of modern GPU hardware and performs inference in a few seconds. Instead, the thinning algorithm constitutes HYPRO extremely slow and impractical for inference on large datasets. In some cases like the Taxi dataset, HYPRO needs $8,130\times$ more time than DTPP to perform inference. Moreover, DTPP attains better performance across all datasets even for a larger number of proposals in HYPRO. These results verify our assumption about the robustness of the mixture model to predict accurately the next time; they also highlight the inaccurate predictions and computational burden of the thinning algorithm.

# 6    Discussion

We have presented DTPP, a Transformer-based probabilistic model for continuous-time event sequences. The model has been derived using the decomposability of the likelihood of a MTPP in terms of its CPDF and CPMF. We have used a mixture of log-normals and a Transformer architecture to model CPDF and CPMF, respectively. Our model satisfies some desirable properties compared to previous works that tried to model the CIF such as closed-form computation of the log-likelihood and inference without resorting to the thinning algorithm. Extensive experiments on the standard task of next-event prediction showed that our method outperformed all state-of-the-art autoregressive models, The results also reveal a more robust performance of the methods that do not require the thinning algorithm to generate event sequences over those they do. Finally, we have tested our model on the challenging task of long-horizon prediction of event sequences. Although our model has not been designed for this task, it outperformed the state-of-the-art baseline HYPRO which is also based on the thinning algorithm. This performance for DTPP was achieved in orders of magnitude faster than HYPRO.

**Limitations and future work.**    The main limitation of the model stems from the modeling of $p^*$ using a deep learning architecture which is usually data-hungry and thus requires large amount of data to learn the model's parameters. For this reason, the model might be unsuitable for data-scarce regimes since it could be prone to overfit. Regarding future work, the limitations of the thinning algorithm revealed by the experiments raise many interesting questions on how can we improve this pathology for the CIF-based methods so they match the performance of the CPDF-based ones since their representations are equivalent. Another interesting research direction would be the development of a globally normalized model similar to HYPRO for CPDF-based models.

9

## Acknowledgements

We would like to thank Petros Dellaportas and Lina Gerontogianni for helpful discussions. This work was funded by Toyota Motor Europe.

## References


[1] Gutti Jogesh Babu and Eric D Feigelson. Spatial point processes in astronomy. *Journal of Statistical Planning and Inference*, 50(3):311–326, 1996.

[2] E. Bacry, M. Bompaire, S. Gaïffas, and S. Poulsen. tick: a Python library for statistical learning, with a particular emphasis on time-dependent modeling. *ArXiv e-prints*, July 2017.

[3] Emmanuel Bacry, Adrian Iuga, Matthieu Lasnier, and Charles-Albert Lehalle. Market impacts and the life cycle of investors orders. *Market Microstructure and Liquidity*, 1(02):1550009, 2015.

[4] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[5] Alex Boyd, Robert Bamler, Stephan Mandt, and Padhraic Smyth. User-dependent neural sequence models for continuous-time event data. *Advances in Neural Information Processing Systems*, 33:21488–21499, 2020.

[6] David R Cox. Partial likelihood. *Biometrika*, 62(2):269–276, 1975.

[7] Daryl J Daley and David Vere-Jones. Basic properties of the Poisson process. *An introduction to the theory of point processes: Volume I: Elementary theory and methods*, pages 19–40, 2003.

[8] Anirban DasGupta. *Asymptotic theory of statistics and probability*, volume 180. Springer, 2008.

[9] Abir De, Utkarsh Upadhyay, and Manuel Gomez-Rodriguez. Temporal point processes. *Technical report, Technical report, Saarland University*, 2019.

[10] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (methodological)*, 39(1):1–22, 1977.

[11] Prathamesh Deshpande, Kamlesh Marathe, Abir De, and Sunita Sarawagi. Long horizon forecasting with temporal point processes. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 571–579, 2021.

[12] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1555–1564, 2016.

[13] Joel Hasbrouck. Measuring the information content of stock trades. *The Journal of Finance*, 46(1):179–207, 1991.

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

[15] Alan G Hawkes and David Oakes. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, pages 493–503, 1974.

[16] Sergio Hernandez, Pedro Alvarez, Javier Fabra, and Joaquin Ezpeleta. Analysis of users' behavior in structured e-commerce websites. *IEEE Access*, 5:11941–11958, 2017.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.


10


[19] Joon Lee, Daniel J Scott, Mauricio Villarroel, Gari D Clifford, Mohammed Saeed, and Roger G Mark. Open-access mimic-ii database for intensive care research. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 8315–8318. IEEE, 2011.

[20] Jure Leskovec and Andrej Krevl. Snap datasets: Stanford large network dataset collection, 2014.

[21] PA W Lewis and Gerald S Shedler. Simulation of nonhomogeneous Poisson processes by thinning. *Naval research logistics quarterly*, 26(3):403–413, 1979.

[22] Thomas Josef Liniger. *Multivariate hawkes processes*. PhD thesis, ETH Zurich, 2009.

[23] Geoffrey J McLachlan, Sharon X Lee, and Suren I Rathnayake. Finite mixture models. *Annual review of statistics and its application*, 6:355–378, 2019.

[24] Hongyuan Mei and Jason M Eisner. The neural Hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*, pages 6754–6764, 2017.

[25] Hongyuan Mei, Guanghui Qin, and Jason Eisner. Imputing missing events in continuous-time event streams. In *International Conference on Machine Learning*, pages 4475–4485. PMLR, 2019.

[26] Hongyuan Mei, Guanghui Qin, Minjie Xu, and Jason Eisner. Neural datalog through time: Informed temporal modeling via logical specification. In *International Conference on Machine Learning*, pages 6808–6819. PMLR, 2020.

[27] Santhosh Narayanan, Ioannis Kosmidis, and Petros Dellaportas. Flexible marked spatio-temporal point processes with applications to event sequences from association football. *Journal of the Royal Statistical Society Series C: Applied Statistics*, page qlad085, 2023.

[28] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197, 2019.

[29] Takahiro Omi, Naonori Ueda, and Kazuyuki Aihara. Fully neural network based model for general temporal point processes. *arXiv preprint arXiv:1905.09690*, 2019.

[30] Aristeidis Panos, Ioannis Kosmidis, and Petros Dellaportas. Scalable marked point processes for exchangeable and non-exchangeable event sequences. In *International Conference on Artificial Intelligence and Statistics*, pages 236–252. PMLR, 2023.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32:8026–8037, 2019.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[33] Jakob Gulddahl Rasmussen. Lecture notes: Temporal point processes and the conditional intensity function. *arXiv preprint arXiv:1806.00221*, 2018.

[34] Diego Rybski, Sergey V Buldyrev, Shlomo Havlin, Fredrik Liljeros, and Hernán A Makse. Communication activity in a social network: relation between long-term correlations and inter-event clustering. *Scientific reports*, 2(1):560, 2012.

[35] Oleksandr Shchur, Marin Biloš, and Stephan Günnemann. Intensity-free learning of temporal point processes. In *International Conference on Learning Representations*, 2019.



11



[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[37] Chris Whong. Foiling nyc's taxi trip data. `https://chriswhong.com/open-data/foil_nyc_taxi/`, 2014.

[38] Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen Chu. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, 2017.

[39] Siqiao Xue, Xiaoming Shi, Zhixuan Chu, Yan Wang, Fan Zhou, Hongyan Hao, Caigao Jiang, Chen Pan, Yi Xu, James Y Zhang, et al. Easytpp: Towards open benchmarking the temporal point processes. *arXiv preprint arXiv:2307.08097*, 2023.

[40] Siqiao Xue, Xiaoming Shi, James Zhang, and Hongyuan Mei. Hypro: A hybridly normalized probabilistic model for long-horizon prediction of event sequences. *Advances in Neural Information Processing Systems*, 35:34641–34650, 2022.

[41] Chenghao Yang, Hongyuan Mei, and Jason Eisner. Transformer embeddings of irregularly spaced events and their participants. In *International Conference on Learning Representations*, 2022.

[42] Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive Hawkes process. In *International Conference on Machine Learning*, pages 11183–11193. PMLR, 2020.

[43] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. Learning tree-based deep model for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1079–1088, 2018.

[44] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International Conference on Machine Learning*, pages 11692–11702. PMLR, 2020.



12


# A  Experimental details

## A.1  Dataset Details

Summary statistics and characteristics of the datasets used are given in Table 3. A more detailed description is given below:

- **Hawkes1-Synthetic.** This dataset contains synthetic event sequences from a univariate Hawkes process sampled using Tick [2] whose conditional intensity function is defined by $\lambda^*(t) = \mu + \sum_{t_i < t} \sum_{j=1}^{J} \alpha_j \beta_j \exp(-\beta_j(t - t_i))$. We use $J = 1, \mu = 1, \alpha_1 = 0.8, \beta_1 = 1.0$. This dataset has been used in [29].

- **Hawkes2-Synthetic.** Same as **Hawkes1-Synthetic** where the parameters here are set as $J = 2, \mu = 0.2, \alpha_1 = 0.4, \beta_1 = 1.0, \alpha_2 = 0.4, \beta_2 = 20$.

- **SAHP-Synthetic.** We sample sequences from a randomly initialized SAHP model using the thinning algorithm. The number of layers and the dimension of hidden states are 4 and 32, respectively. The same dataset has been used in [41].

- **MIMIC-II.** The Multiparameter Intelligent Monitoring in Intensive Care (MIMIC-II) is a medical dataset of de-identified clinical visit records of intensive care unit patients for seven years. There are records of 650 patients/sequences where each one contains the time of the visit and the diagnosis of this visit; there are K=75 unique diseases. We try to predict the time and the diagnosis of a patient.

- **Amazon.** This dataset includes time-stamped user product reviews behavior from January, 2008 to October, 2018. Each user has a sequence of review events with each event containing the timestamp and category of the reviewed product, with each category corresponding to an event type. As in [39], we use a subset of 5200 most active users with an average sequence length of 70 which is comprised of K = 16 event types.

- **Taxi.** This dataset records the times of taxi pick-up and drop-off events across the five boroughs of the New York city (Manhattan, Brooklyn, Queens, The Bronx, Staten Island). For each borough, we can have pick-up or drop-off event and thus, there are K = 10 event types in total. As in [40], we pick a randomly sampled subset of 2000 drivers where each driver has a sequence.

- **Taobao.** The dataset comes from the 2018 Tianchi Big Data Competition. It consists of time-stamped behavior records (e.g., browsing, purchasing) of anonymous users on the online shopping platform Taobao from the 25th of November through the 3rd of December in 2017. The K=17 event types represent a category group (e.g. men's clothing). The browsing sequences of the most active 2000 users are picked as event sequences. The time unit is 3 hours and the average inter-arrival time is 0.06.

- **StackOverflow-V1.** The data comes from the well-known question-answering website StackOverflow where users are encouraged to answer questions so they can earn badges. There are 22 different types of badges. Each sequence corresponds to a user and each event gives the time and the type of budge a user has been awarded. This dataset is the one processed and used in [12].

- **StackOverflow-V2.** A truncated version of the original **StackOverflow-V1.**. The time unit is 11 days and the average inter-arrival time is 0.95. The dataset has been used in [40]

## A.2  Training Details

We use the Adam optimizer [18] with its default settings to train all the models in Section 5. We use 200 epochs in total, a batch size of 8 sequences, and we apply early-stopping based on the log-likelihood of the held-out dev set.

Following [41], we use a common hyperparameter $D$ to define all dimensionalities of the query, key, and value vectors for the models with attention mechanisms, i.e. THP, SAHP, A-NHP, and DTPP-CPMF. For these methods, we also need to specify the number of layers $L$. We also denote by $D$ the state space of the IFTPP model.

13

Table 3: Characteristics of the synthetic and real-world datasets.

| DATASET | $K$ | # OF EVENT TOKENS | | | SEQUENCE LENGTH | | |
|---------|-----|-------|------|------|------|------|------|
| | | TRAIN | VAL | DEV | MIN | MEAN | MAX |
| HAWKES1-SYNTHETIC | 1 | 65536 | 16384 | 32768 | 64 | 64 | 64 |
| HAWKES2-SYNTHETIC | 1 | 65536 | 16384 | 32768 | 64 | 64 | 64 |
| SAHP-SYNTHETIC | 10 | 71023 | 10242 | 20237 | 81 | 101 | 121 |
| MIMIC-II | 75 | 1930 | 252 | 237 | 2 | 4 | 33 |
| AMAZON | 16 | 288377 | 40995 | 84048 | 14 | 44 | 94 |
| TAXI | 10 | 51854 | 7404 | 14820 | 36 | 37 | 38 |
| TAOBAO | 17 | 75205 | 11737 | 28455 | 32 | 57 | 64 |
| STACKOVERFLOW-V1 | 22 | 345116 | 38065 | 97233 | 41 | 72 | 736 |
| STACKOVERFLOW-V2 | 22 | 90497 | 25762 | 26518 | 41 | 65 | 101 |

Table 4: Performance comparison between our model DTPP and various baselines in terms of next-event prediction on Mimic-II dataset. The root mean squared error (RMSE) measures the error of the predicted time of the next event, while the error rate (ERROR-%) evaluates the error of the predicted mark given the true time. The results (lower is better) are accompanied by 95% bootstrap confidence intervals. $^{\dagger},^{\triangleleft},^{\triangleright}$ denote the CIF-based methods, the CPDF-based methods that use a single model, and the ones using a seperate model, respectively.

| | MIMIC-II | |
|---------|------|-------|
| METHODS | RMSE | ERROR |
| THP$^{\dagger}$ | 1.00 $\pm 0.13$ | 15.12 $\pm 0.99$ |
| SAHP$^{\dagger}$ | 1.49 $\pm 0.15$ | 16.39 $\pm 5.72$ |
| A-NHP$^{\dagger}$ | 1.00 $\pm 0.19$ | 15.19 $\pm 5.16$ |
| IFTPP$^{\triangleleft}$ | 0.74 $\pm 0.39$ | 14.87 $\pm 5.33$ |
| VI-DPP$^{\triangleright}$ | 0.95 $\pm 0.40$ | 16.96 $\pm 6.30$ |
| DTPP$^{\triangleright}$ | **0.72** $\pm 0.39$ | **14.49** $\pm 5.86$ |

The hyperparameters $D$ and $L$ were fine-tuned for each combination of dataset and model. We grid-search the two parameters using the search spaces $D \in \{4, 8, 16, 32, 64, 128\}$ and $L \in \{1, 2, 3, 4, 5\}$. We pick the set of values that achieve the highest log-likelihood on the dev set.

For IFTPP we use the same search for $D$ as before. Since IFTPP and DTPP (CPDF part) are based on a mixture of log-normals, we need to define the number of components $M$. We fine tune $M$ over the space $M \in \{1, 2, 4, 8, 16\}$.

VI-DPP has only one hyperparameter that requires tuning. This is the number of cutt-off points $Q$ [30]. As the original paper, we found that $Q = 1$ works the best in all of our datasets.

For long-horizon prediction experiments, we chose a common size architecture for the HYPRO / DTPP transformer by setting $D = 128$ and $L = 2$ since this combination worked well on both A-NHP and DTPP in all datasets in Section 5.1.

All experiments were carried out on the same Linux machine with a dedicated reserved GPU used for acceleration.

## A.3 Implementation Details

For A-NHP, we use the public Github repository at https://github.com/yangalan123/anhp-andtt. For THP and SAHP, we also use the same repository where the corrected implementations of THP and SAHP are provided. For IFTPP, we use the public Github repository at https://github.com/shchur/ifl-tpp while for VI-DPP we use the code at https://github.com/aresPanos/Interpretable-Point-Processes. Our code will be made publicly available on a Github repository after the review period.

14

# B  Extra Experimental Results

## B.1  Results on Mimic-II

Table 4 presents the results for next-event prediction task on Mimic-II dataset. The results follow similar patterns as in Table 1.

## B.2  Results on Synthetic datasets

Figure 4 shows the results for the two synthetic datasets generated by two different one-dimensional Hawkes processes. The DTPP's mixture of log-normals fails to model correctly the two processes since it lacks the flexibility of the neural network that A-NHP and IFTPP are based on. However, when it comes to next-time prediction, DTPP performs on par with A-NHP, showing once more that the thinning algorithm decreases prediction accuracy. IFTPP is the clear winner for these two datasets.

## B.3  Results on Long-Horizon Prediction

We present additional results for the long-horizon prediction in Tables 5 and 6. Table 5 reports the RMSE$^*$ where the results show that DTPP outperforms HYPRO, regardless of its number of proposals. The only exception is StackOverflow-v2, where HYPRO achieved a lower score but required at least 1,544 more running time than DTPP.

Finally, we report HYPRO's time without considering the time required to generate noise sequences (Algorithm 1 in [40]) from the trained auto-regressive model $p_{\text{auto}}$ for training the energy function $E_\theta$; see discussion in Section 5.2. Even in this case, our DTPP model is orders of magnitude faster than HYPRO.

Table 5: Performance comparison between our model DTPP and HYPRO for the long-horizon prediction task. For HYPRO, we use $\{2, 4, 8, 16, 32\}$ weighted proposals (Algorithm 2 in [40]). We report the root mean squared error of the number of tokens for each type of event (RMSE$^\star$) and the time (in minutes) required to predict all the long-horizon sequences of the held-out dataset (lower is better). We include error bars for five runs.

| METHODS | # PARAMS (K) | TAXI | | TAOBAO | | STACKOVERFLOW-V2 | |
|---|---|---|---|---|---|---|---|
| | | RMSE$^\star$ | TIME | RMSE$^\star$ | TIME | RMSE$^\star$ | TIME |
| HYPRO-2 | | 1.39 ±0.03 | 44.81 ±0.01 | 2.87 ±0.03 | 43.53 ±0.03 | 1.16 ±0.01 | 46.32 ±0.01 |
| HYPRO-4 | | 1.34 ±0.03 | 47.31 ±0.04 | 2.74 ±0.02 | 46.57 ±0.08 | 1.15 ±0.01 | 48.84 ±0.06 |
| HYPRO-8 | 850 | 1.27 ±0.02 | 52.61 ±0.20 | 2.65 ±0.03 | 53.10 ±0.17 | 1.15 ±0.02 | 54.14 ±0.22 |
| HYPRO-16 | | 1.24 ±0.03 | 62.36 ±0.27 | 2.58 ±0.02 | 65.63 ±0.44 | 1.15 ±0.02 | 64.82 ±0.34 |
| HYPRO-32 | | 1.20 ±0.03 | 81.30 ±0.23 | 2.55 ±0.02 | 89.11 ±0.71 | **1.14** ±0.01 | 83.05 ±0.45 |
| DTPP | 400 | **0.96** ±0.03 | **0.01** ±0.00 | **2.45** ±0.04 | **0.17** ±0.01 | 2.02 ±0.06 | **0.03** ±0.00 |
| SPEEDUP | | 8,130× | | 524.2× | | 2768.3× | |

15

(a) Hawkes-1  (b) Hawkes-2
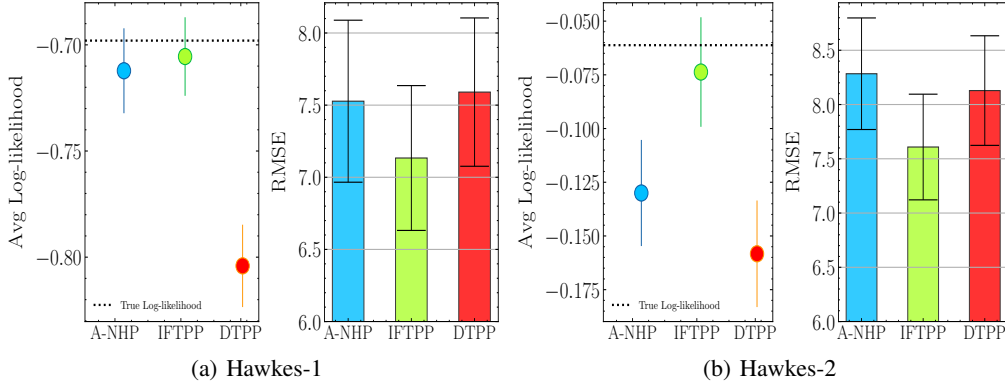
Figure 4: Goodness-of-fit and next-time prediction comparison over the two 1-d synthetic examples generated from a Hawkes process. The reported results are based on the test dataset. The black dotted line represents the true log-likelihood of the data (in nats).

Table 6: Time comparison (in minutes) between our model DTPP and HYPRO for the long-horizon prediction task. Unlike Tables 2 and 5, here we only report the prediction time of HYPRO without including the time required to generate noise sequences (Algorithm 1 in [40]) from the trained auto-regressive model $p_{\text{auto}}$ for training the energy function $E_\theta$. We include error bars based on five runs.

| METHODS | # PARAMS (K) | TAXI | TAOBAO | STACKOVERFLOW-V2 |
|---|---|---|---|---|
| HYPRO-2 | | 2.41 ±0.01 | 3.13 ±0.03 | 2.48 ±0.01 |
| HYPRO-4 | | 4.91 ±0.04 | 6.17 ±0.08 | 5.01 ±0.06 |
| HYPRO-8 | 850 | 10.21 ±0.20 | 12.70 ±0.17 | 10.30 ±0.22 |
| HYPRO-16 | | 19.96 ±0.27 | 25.23 ±0.44 | 20.98 ±0.34 |
| HYPRO-32 | | 38.90 ±0.23 | 48.70 ±0.71 | 39.21 ±0.44 |
| DTPP | 400 | **0.01** ±0.00 | **0.17** ±0.01 | **0.03** ±0.00 |
| SPEEDUP | | 3,890× | 286.5× | 1,307× |

---

**Algorithm 1** Long-Horizon Prediction for Decomposed Transformer Point Processes

---

**Input:** an observed sequence $\mathcal{H}_T$ of $N$ events over the interval $(0, T)$ and the number of prediction steps $P > 1$; trained inter-event model $g$ and event types model $p$
**Output:** predicted sequence $\hat{\mathcal{H}}_P$ of $P$ events
1: **function** LHP($\mathcal{H}_T, P, g^*, p^*$)
2: &nbsp;&nbsp;&nbsp;&nbsp;$\hat{\mathcal{H}}_P \leftarrow \emptyset, \ \hat{\mathcal{H}} \leftarrow \mathcal{H}_T$
3: &nbsp;&nbsp;&nbsp;&nbsp;**for** $p = 1$ **to** $P$:
4: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;▷ *Predict next-event time and event type*
5: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;$t_{N+p} = t_{N+p-1} + \mathbb{E}_g^{(k_{N+p-1})}[\tau \mid \hat{\mathcal{H}}]$
6: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;$k_{N+p} = \underset{k}{\arg\max}\, p(k \mid t_{N+p}, \hat{\mathcal{H}})$
7: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;▷ *Update predicted sequence and event history*
8: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;$\hat{\mathcal{H}}_P \leftarrow \hat{\mathcal{H}}_P \cup (t_{N+p}, k_{N+p})$
9: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;$\hat{\mathcal{H}} \leftarrow \hat{\mathcal{H}} \cup (t_{N+p}, k_{N+p})$
10: &nbsp;&nbsp;&nbsp;**end for**
11: &nbsp;&nbsp;&nbsp;**return** $\hat{\mathcal{H}}_P$
12: **end function**

16

---

**Algorithm 2** Thinning Algorithm

---

**Input:** an observed sequence $\mathcal{H}_T$ of $N$ events over the interval $(0, T)$ and the number of prediction steps $P > 1$; intensity function $\lambda_k$

**Output:** predicted sequence $\hat{\mathcal{H}}_P$ of $P$ events

1: **function** Thinning($\mathcal{H}_T$, $P$, $\lambda_k$)
2: &nbsp;&nbsp;&nbsp;&nbsp;$\hat{\mathcal{H}}_P \leftarrow \emptyset$, $\hat{\mathcal{H}} \leftarrow \mathcal{H}_T$, $p \leftarrow 0$, $\hat{t} \leftarrow T$
3: &nbsp;&nbsp;&nbsp;&nbsp;**for** $p = 1$ **to** $P$:
4: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;$\lambda_{\max} = \max_{t \in (\hat{t}, \infty)} \sum_{k=1}^{K} \lambda_k(t \mid \hat{\mathcal{H}})$ &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;▷ *Compute upper bound $\lambda_{max}$*
5: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**repeat**:
6: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;$\tau \sim \text{Exp}(\lambda_{\max})$
7: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;$\hat{t} \leftarrow \hat{t} + \tau$
8: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**until** $u \le \sum_{k=1}^{K} \lambda_k(\hat{t} \mid \hat{\mathcal{H}})/\lambda_{\max}$ &nbsp;&nbsp;&nbsp;&nbsp;▷ *Accept proposed occurrence time with probability* $\sum_{k=1}^{K} \lambda_k(\hat{t} \mid \hat{\mathcal{H}})/\lambda_{max}$
9: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;$\hat{k} \sim \text{Cat}(p_1, \ldots, p_K)$ where $p_k = \lambda_k(\hat{t} \mid \hat{\mathcal{H}})/\sum_{k=1}^{K} \lambda_k(\hat{t} \mid \hat{\mathcal{H}})$ &nbsp;&nbsp;&nbsp;&nbsp;▷ *Sample event type $\hat{k} \in \{1, \ldots, K\}$*
10: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;$t_{N+p} \leftarrow \hat{t}$, $k_{N+p} \leftarrow \hat{k}$
11: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;$\hat{\mathcal{H}}_P \leftarrow \hat{\mathcal{H}}_P \cup (t_{N+p}, k_{N+p})$, &nbsp;&nbsp;$\hat{\mathcal{H}} \leftarrow \hat{\mathcal{H}} \cup (t_{N+p}, k_{N+p})$ &nbsp;&nbsp;&nbsp;&nbsp;▷ *Update predicted sequence and event history*
12: &nbsp;&nbsp;&nbsp;&nbsp;**end for**
13: &nbsp;&nbsp;&nbsp;&nbsp;**return** $\hat{\mathcal{H}}_P$
14: **end function**

---

17

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist",**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See Section 1

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See section 6

18

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: -

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Sections 5 and A.3

   Guidelines:

   - The answer NA means that the paper does not include experiments.

19

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: See Sections 5 and A.3

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

20

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See Sections 5 and A.3

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines: See Section 5

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Section A

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

21

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification:

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Guidelines: This work is concerned with modeling point processes and this

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification:

22

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: See Sections 5 and A.1

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

23

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

24