Unmixing Noise from Hawkes Process to Model Learned Physiological Events

Anonymous Authors[1]

Abstract

Physiological signal analysis often involves identifying events crucial to understanding biological dynamics. Traditional methods rely on handcrafted procedures or supervised learning, presenting challenges such as expert dependence, lack of robustness, and the need for extensive labeled data. Data-driven methods like Convolutional Dictionary Learning (CDL) offer an alternative but tend to produce spurious detections. This work introduces UNHaP (Unmix Noise from Hawkes Processes), a novel approach addressing the joint learning of temporal structures in events and the removal of spurious detections. Leveraging marked Hawkes processes, UNHaP distinguishes between events of interest and spurious ones. By treating the event detection output as a mixture of structured and unstructured events, UNHaP efficiently unmixes these processes and estimates their parameters. This approach significantly enhances the understanding of event distributions while minimizing false detection rates.

1. Introduction

The analysis of physiological signals often boils down to identifying events of interest. Typical examples are with electrocardiography (ECG), where the detection of the QRS complex *–a.k.a.* the heartbeat– is a fundamental step to characterize the status of the cardiovascular system, with biomarkers like the heart rate (HR; (Berkaya et al., 2018)) and heart rate variability (HRV; (Luz et al., 2016)). Another example is the identification of steps in inertial measurement unit recordings, which is a crucial feature in classifying pathological gait anomalies (Cimolin & Galli, 2014).

To automatize the event detection step, several approaches have been proposed. In most physiological signal processing applications, events are detected with handcrafted procedures based on signal processing techniques. For instance, the QRS complexes or the steps are identified using peak detection algorithms (Pan & Tompkins, 1985) or wavelet-based approaches (Martinez et al., 2004). While these algorithms perform well, they require large domain expertise, and their parameters tend to be sensible to the acquisition protocol. Data-driven approaches have also been proposed, using supervised deep learning (Xiang et al., 2018; Craik et al., 2019). These approaches demonstrate excellent performance on particular tasks. Yet, they require large labeled datasets. Another data-driven approach is unsupervised learning to extract repeating patterns, such as the convolutional dictionary learning (CDL) algorithm (Grosse et al., 2007). These methods aim to represent events through their prototypical patterns, which are directly learned from the data. While these solutions can be applied independently of the signal, they tend to detect more spurious events.

To reach satisfactory results, all these methods require post-processing steps to filter out spurious events. Developing and characterizing these extra steps is a tedious task, requiring domain expertise and time. In this paper, we propose a novel automatized framework to filter out spurious events based on their temporal distribution and the event detection confidence. A key observation for all event detection methods is that each event is detected independently, with an estimated confidence in the event detection. However, in most cases, the events are distributed with an informative temporal structure: the inter-heartbeat interval is around one second for a normal ECG. We propose to classify detected events between spurious and structured ones, by jointly learning the temporal structure of the events and filtering out spurious event detection based on the distribution of confidence levels.

To model the events' temporal distribution, we rely on Hawkes processes (HP; (Hawkes, 1971)), a classical type of point process (PP) to model past events' influence on future events. Recent works have proposed novel inference techniques adapted to physiological physiological events' distribution (Allain et al., 2022; Staerman et al., 2023). Yet, these models can't account for the confidence associated

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

with the event detection and need to be extended to deal with marked PP (Daley et al., 2003), to account for the effect of the marks in the intensity function.

In addition, inference with these models only works when all events come from the same process. In our context, a mixture of spurious events from a noise process and structured events is observed, and direct inference gives uninformative biased results. Mixtures of Hawkes processes have been considered in the literature either to cluster events (Liu et al., 2019; Yang & Zha, 2013) or sequences of events (Xu & Zha, 2017). They rely on feature-based mixture models (Li & Zha, 2013; Yang & Zha, 2013; Du et al., 2015) or associate a Dirichlet process to classical Hawkes models (Blei & Jordan, 2006). While these approaches are tailored to find different auto-excitation patterns, they are not designed to unmix noise and uninformative events from structured ones.

Contributions. To jointly model the temporal distribution of events and remove spurious events, we propose a novel method named UNHaP to Unmix Noise from Hawkes Processes. In our model, the output of the event detection algorithm is treated as a mixture of events of interest with a Hawkes process structure and spurious events that are not of interest, distributed as a Poisson process. UNHaP aims to learn to distinguish between these two distinct processes to select properly structured events and discard the spurious ones. Based on the FaDIn framework (Staerman et al., 2023), we propose an efficient algorithm to jointly unmix these events and estimate the parameters of the Hawkes process. We illustrate the benefits of using our unmixing models rather than the traditional Hawkes process models with real-world ECG and gait data.

2. Background on Marked Hawkes Processes

A multivariate marked Hawkes process (MMHP) is a self-exciting point process that models the occurrence of events in time, where each event is associated with supplementary information, referred to as the "*mark*" of the event. The mark may or may not integrate the event type in the literature. Throughout this paper, we separate the event type from the mark and consider continuous marks belonging to $\mathbb{R}$. We here give our notation and basic information about MMHP and refer the reader to (Daley et al., 2003) for a detailed account of these processes.

Counting processes. Let $\mathscr{F}_T$ be a set of observed marked events including D types such that for each $i \in [\![1\,, D]\!]$ we have $\mathscr{F}_T^i = \left\{(t_n^i, \kappa_n^i) : \ \kappa_n^i \in \mathcal{K},\ t_n^i \in [0,T]\right\}$ with t_n^i the time where the n-th event of type i occurs and κ_n^i its associated mark. We denote by $\mathbf{N}_i$ the random counting measure defined on $[0,T]\times\mathbb{R}_+$, such that $\mathbf{N}_i(\mathrm{d}t,\mathrm{d}\kappa) = \sum_{n=1}^{\infty} \delta_{(t_n^i,\kappa_n^i)}(\mathrm{d}t,\mathrm{d}\kappa)$, where t and κ represent respectively the time and the mark, and $T \in \mathbb{R}_+$ is the stopping time. Without limitations, the set of marks is assumed to be any compact set $\mathcal{K} \subset \mathbb{R}_+$. From this measure, we can define the marginal time arrival process, also called ground process, as $N_i(T) = \int_{[0,T]\times\mathbb{R}_+} \mathbf{N}_i(\mathrm{d}t,\mathrm{d}\kappa) = \sum_{n\geq 1} \mathbf{1}_{t_n^i \leq T}$.

Intensity function. The behavior of a MMHP can be described by its intensity function. Conditionally to observed events, it describes the instantaneous event rate at any given point in time. Given a MMHP and a set of observation $\mathscr{F}_T = \{\mathscr{F}_T^i\}_{i=1}^D$, each ground process N_i is described by the following conditional ground intensity function

$$\lambda_{g_i}(t|\mathscr{F}_t) = \mu_i + \sum_{j=1}^{D} \int_{[0,t)\times\mathcal{K}} h_{ij}(t-u,\kappa)\,\mathbf{N}_j(\mathrm{d}u,\mathrm{d}\kappa),$$

where μ_i is the baseline rate and $h_{ij} : \mathbb{R}_+ \times \mathcal{K} \to \mathbb{R}_+$ is the triggering or kernel function, quantifying the influence of the j-th process' past events onto the i-th process' future events. The ground intensity quantifies the time probability of future events, taking into account the marks of previous events. In the following, we consider independent probability for the marks (Daley et al., 2003), assuming a factorized form for the kernel $h_{ij}(t,\kappa) = \phi_{ij}(t)\omega_{ij}(\kappa)$. This leads to

$$\begin{aligned}\lambda_{g_i}(t|\mathscr{F}_t) &= \mu_i + \sum_{j=1}^{D} \int_{[0,t)\times\mathcal{K}} \omega_{ij}(\kappa)\ \phi_{ij}(t-u)\,\mathbf{N}_j(\mathrm{d}u\times\mathrm{d}\kappa)\\ &= \mu_i + \sum_{j=1}^{D} \sum_{n, t_n^j < t} \omega_{ij}(\kappa_n^j)\ \phi_{ij}(t-t_n^j),\end{aligned}$$

with $\omega_{ij} : \mathcal{K} \to \mathbb{R}_+$, $\phi_{ij} : \mathbb{R}_+ \to \mathbb{R}_+$ such that $\int_0^\infty \phi_{ij}(t)\mathrm{d}t < 1$ and $\int_{\mathcal{K}} \omega_{ij}(\kappa)\mathrm{d}\kappa < 1$. These conditions ensure the stability of such processes. The function $\omega_{ij}(\cdot)$ weights the probability that a future event occurs depending on the past events' marks. Assuming a collection $\{f_i : \mathcal{K} \to \mathbb{R}_+\}_{i=1}^D$ of density functions, we define the joint intensity function as $\lambda_i(t,\kappa) = \lambda_{g_i}(t|\mathscr{F}_t)\ f_i(\kappa)$, where the ground process depends on the mark distribution reflected by f_i and the distribution of the influence of the mark described by ω_{ij}.

ERM-based inference. Inference for MMHP is usually performed using the log-likelihood to align the model with the observed data (Daley et al., 2003; Bacry et al., 2015). While this can be efficient for Markovian kernels, it becomes computationally expensive for more general ones (Staerman et al., 2023). In this paper, we instead resort to the ERM-inspired least squares loss (refer to Eq. (II.4) in Bompaire, 2019, Chapter 2). The goal is to minimize

$$\mathcal{L}\left(\boldsymbol{\theta}, \mathscr{F}_T\right) = \sum_{i=1}^{D} \int_0^T \int_{\mathcal{K}} \lambda_i(s, \kappa; \boldsymbol{\theta})^2 \,\mathrm{d}\kappa\mathrm{d}s - 2\sum_{i=1}^{D} \sum_{(t_n^i, \kappa_n^i) \in \mathscr{F}_T^i} \lambda_i\left(t_n^i, \kappa_n^i; \boldsymbol{\theta}\right),$$

where $\boldsymbol{\theta} = \{\mu_i, \phi_{ij}, \omega_{ij}\}_{i=1}^D$. This loss function corresponds to the empirical approximation of the expected risk incurred by the model measured by $\|\lambda(\boldsymbol{\theta}) - \lambda^*\|_2$, with λ^* the true intensity function. It is more efficient to compute than the log-likelihood, especially for general parametric kernels (Staerman et al., 2023).

3. Unmixing Noise from Hawkes Process

Problem statement. We consider a set of observed events $\mathscr{F}_T = \left\{e_n^i = (t_n^i, \kappa_n^i),\ 1 \leq n \leq N_i(T)\right\}_{i=1}^D$ with events originating from two independent processes. We denote $\mathscr{F}_{T,k} = \left\{e_n^{i,k} = (t_n^{i,k}, \kappa_n^{i,k}); 1 \leq n \leq N_i^k(T)\right\}_{i=1}^D$ these two processes such that $\mathscr{F}_T = \mathscr{F}_{T,0} \cup \mathscr{F}_{T,1}$. We consider the case where $\mathscr{F}_{T,0}$ is a homogeneous marked Poisson process –representing spurious event detections– and $\mathscr{F}_{T,1}$ is a MMHP –for structured events. This problem is a denoising problem, where spurious events are considered as noise that should be discarded for the application.

Our goal is to unmix these two processes, *i.e.,* to associate each event $e_n^i \in \mathscr{F}_T$ with a label $Y_n^i \in \{0, 1\}$ such that $Y_n^i = 1$, if e_n^i originates from $\mathscr{F}_{T,1}$. This task amounts to binary classification for the events. However, the main difficulty lies in that the labels are unknown, and the events are not independent. To cope with the lack of labels, we propose to leverage the temporal MMHP structure of $\mathscr{F}_{T,1}$ to characterize structured events, assigning events with this process if they are plausible according to the MMHP model. This is an arduous assignment problem, which we address using a variational inference approach and a mean-field relaxation. This procedure allows us to jointly estimate the parameters of the processes while unmixing the events, see Figure 1.

Latent variables and risk function. Unmixing noise from MMHP events amounts to a binary classification task, where the underlying structure of the events allows to discriminate between the two classes and has to be inferred. Our goal is thus to infer the value of latent variables Y_n^i for each event such that $Y_n^i = 1$ if the n-th event of the i-th type is generated by $\mathscr{F}_{T,1}$ while $Y_n^i = 0$ if it is generated by $\mathscr{F}_{T,0}$.

When these latent variables are known, it is possible to write the intensity functions of both processes from the observed events $\mathscr{F}_T$. Spurious events from $\mathscr{F}_{T,0}$ are distributed following a marked Poisson process with intensity $\lambda_i^0(t, \kappa; \boldsymbol{\theta}_0) = \tilde{\mu}_i f_i^0(\kappa)$ such that $\tilde{\mu}_i \in \mathbb{R}_+$, $f_i^0 : \mathcal{K} \to \mathbb{R}_+$, $\int_{\mathcal{K}} f_i^0(\kappa)\,\mathrm{d}\kappa = 1$ and $\boldsymbol{\theta}_0 = \{\tilde{\mu}_i\}_{i=1}^D$. Non-spurious events follow a MMHP whose intensity, denoted $\lambda_i^1(t, \kappa; \boldsymbol{\theta}_1)$, can be derived from the observed events only. We have, for $t \in [0, T]$,

$$\lambda_i^1(t, \kappa; \boldsymbol{\theta}_1) = \Big(\mu_i + \sum_{j=1}^{D} \sum_{t_n^j < t} Y_n^j \phi_{ij}(t - t_n^j; \eta_{ij})\, \omega_{ij}(\kappa_n^j)\Big) f_i^1(\kappa),$$

where ϕ_{ij} is a parametric kernel parametrized by η_{ij} and $\boldsymbol{\theta}_1 = \{\mu_i, \eta_{ij}\}_{i,j=1}^D$. An important remark is that the intensity function depends only on past events from $\mathscr{F}_{T,1}$. This is where our model differs from classical MHHP models, as it is necessary to select the right events to be able to compute the intensity function.

Conditioned on the latent variables $\{Y_n^i\}$, both processes are independent. The risk for the parameters $\boldsymbol{\theta}$ is thus the sum of the least square loss, defined in (1), for each process, *i.e.*, $\mathcal{L}(\boldsymbol{\theta}; \mathscr{F}_T) = \mathcal{L}(\boldsymbol{\theta}_0; \mathscr{F}_{T,0}) + \mathcal{L}(\boldsymbol{\theta}_1; \mathscr{F}_{T,1})$. The complete loss, assuming $\mathcal{Y}_T = \{Y_n^i\}_{i,n}$ are observed, can thus be written as $\mathcal{L}\left(\boldsymbol{\theta}; \mathcal{Y}_T, \mathscr{F}_T\right) = \sum_{i=1}^D \mathcal{L}^i\left(\boldsymbol{\theta}; \mathcal{Y}_T, \mathscr{F}_T\right)$, where

$$\begin{aligned}\mathcal{L}^i\left(\boldsymbol{\theta}; \mathcal{Y}_T, \mathscr{F}_T\right) = &\int_0^T\int_{\mathcal{K}} \lambda_i^0(t, \kappa; \boldsymbol{\theta}_0)^2 \,\mathrm{d}\kappa\mathrm{d}t \\ &+ \int_0^T\int_{\mathcal{K}} \lambda_i^1(t, \kappa; \boldsymbol{\theta}_1)^2 \,\mathrm{d}\kappa\mathrm{d}t \\ &- 2\sum_{e_n^i \in \mathscr{F}_T^i} (1 - Y_n^i)\lambda_i^0(t_n^i, \kappa_n^i; \boldsymbol{\theta}_0) \\ &- 2\sum_{e_n^i \in \mathscr{F}_T^i} Y_n^i \lambda_i^1(t_n^i, \kappa_n^i; \boldsymbol{\theta}_1).\end{aligned} \tag{1}$$

If $\boldsymbol{\lambda}_i^0$ and $\boldsymbol{\lambda}_i^1$ are the true intensity functions of the underlying processes, then we have $\mathbb{E}_{\mathscr{F}_T}[\mathcal{L}^i\left(\boldsymbol{\theta}; \mathcal{Y}_T, \mathscr{F}_T\right)] = \|\lambda_i^0(\boldsymbol{\theta_0}) - \boldsymbol{\lambda}_i^0\|_2^2 + \|\lambda_i^1(\boldsymbol{\theta_1}) - \boldsymbol{\lambda}_i^1\|_2^2 - C$ where C is a constant in $\boldsymbol{\theta}$. This loss $\mathcal{L}\left(\boldsymbol{\theta}; \mathcal{Y}_T, \mathscr{F}_T\right)$ is thus the empirical risk of the model for a given set of observed events and an assignment $\{Y_n^i\}$, and the model's parameters can be inferred by minimizing it.

Mean-field-based Variational Inference. The goal of our procedure is also to infer the collection of $\{Y_n^i\}$. The classical procedure to solve such latent factor estimation with probabilistic models is to resort to the Expectation-Maximization (EM) algorithm. This algorithm allows the iterative refinement of the $\boldsymbol{\theta}$'s estimate by maximizing the likelihood marginalized over the latent factors Y_n^i. This requires being able to compute the marginalized likelihood or at least estimate it with Monte Carlo sampling. But this step is not possible with the assignment variable Y_n^i due to the complex dependency structure between the various Y_n^i imposed by the Hawkes process structure.

To alleviate this challenge, we propose to resort to a mean-field approximation with independent variables for each

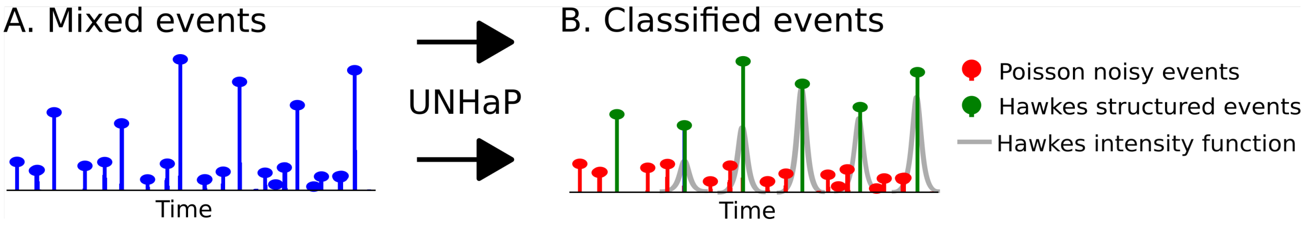

Figure 1. **Illustration of the UNHaP framework.** The goal of UNHaP is to distinguish between structured events (**green**) and spurious ones (**red**) by identifying the structure of the MMHP (**grey**) from the observed events (**blue**).

event. Concretely, we perform the following approximation

$$p(\mathbf{Y};\mathscr{F}_T) = \prod_{i=1}^{D} p(Y^i;\mathscr{F}_T^i) \approx \prod_{i=1}^{D}\prod_{n=1}^{N_T^i} q(Y_n^i;\rho_n^i), \qquad (2)$$

where $q(Y;\rho)$ is a univariate Bernoulli distribution with parameter ρ. The parameter ρ_n^i is the probability that $Y_n^i = 1$. It corresponds to a relaxation of the assignment variable $Y_n^i \in \{0,1\}$ to the interval $[0,1]$. This relaxation allows us to compute the expected risk of the model with respect to the latent variables. Therefore, we have $\bar{\mathcal{L}}\left(\boldsymbol{\rho},\boldsymbol{\theta};\mathscr{F}_T\right) = \mathbb{E}_{\mathbf{Y}}\left[\mathcal{L}\left(\boldsymbol{\theta};\mathcal{Y}_T,\mathscr{F}_T\right)\right] = \sum_{i=1}^{D} \bar{\mathcal{L}}^i\left(\boldsymbol{\rho},\boldsymbol{\theta};\mathscr{F}_T\right)$ with

$$\begin{aligned}\bar{\mathcal{L}}^i\left(\boldsymbol{\theta},\boldsymbol{\rho};\mathscr{F}_T\right) = &\int_0^T\!\!\int_{\mathcal{K}} \lambda_i^0(t,\kappa)^2 \,\mathrm{d}\kappa\mathrm{d}t \\ &+ \int_0^T\!\!\int_{\mathcal{K}} \bar{\lambda}_i^1(t,\kappa)^2 \,\mathrm{d}\kappa\mathrm{d}t + \boldsymbol{C}(\boldsymbol{\rho}) \\ &- 2\sum_{n,t_n^i\in\mathscr{F}_T^i} (1-\rho_n^i)\lambda_i^0\left(t_n^i,\kappa_n^i\right) \\ &- 2\sum_{n,t_n^i\in\mathscr{F}_T^i} \rho_n^i\bar{\lambda}^1\left(t_n^i,\kappa_n^i\right),\end{aligned} \qquad (3)$$

where $\boldsymbol{\rho} = \{\rho_n^i\}$,

$$\boldsymbol{C}(\boldsymbol{\rho}) = \sum_{j=1}^{D}\int_0^T \sum_{n,t_n^j<t} \rho_n^j(1-\rho_n^j)\,\omega_{ij}(\kappa_n^j)^2\phi_{ij}(t-t_n^j)^2\mathrm{d}t$$

and

$$\bar{\lambda}_i^1(t,\kappa;\boldsymbol{\theta}_1) = \Big(\mu_i + \sum_{j=1}^{D}\sum_{t_n^j<t} \rho_n^j\phi_{ij}(t-t_n^j;\eta_{ij})\omega_{ij}(\kappa_n^j)\Big) f_i^1(\kappa)$$

corresponds to λ_i^1 where $\boldsymbol{Y}$ has been replaced by $\boldsymbol{\rho}$. Here, we can replace Y_n^i by its expectation ρ_n^i in the integral of the squared intensity as $\mathbb{E}[Y_n^iY_l^i] = \rho_n^i\rho_l^i$ for the distribution q. However, this is not true for $\mathbb{E}[(Y_n^i)^2]$ which is equal to ρ_n^i and not $(\rho_n^i)^2$. $\boldsymbol{C}(\boldsymbol{\rho})$ corrects this discrepancy. Note that $\bar{\mathcal{L}}$ can also be seen as a relaxation of the assignment problem with continuous variables ρ_n^i.

Algorithm 1 UNHaP solver.

```
input  Set of events $\mathscr{F}_T$.
initialization $\boldsymbol{\rho}^{(0)} \overset{i.i.d.}{\sim} q(1/2)$, $\boldsymbol{\theta}^{(0)}$ initialized with Moments Matching.
  for $\ell$=1, ... $n_{\text{iter}}$ do
    (E-step) $\boldsymbol{\rho}^{(\ell)} = \underset{\boldsymbol{\rho}}{\text{argmin}} \sum_{i=1}^{D} \bar{\mathcal{L}}_{\mathcal{G}}^i(\boldsymbol{\rho};\boldsymbol{\theta}^{(\ell-1)},\mathscr{F}_T)$
    (C-step) Assign the events by computing
        $\mathcal{Y}_T^{(\ell)} = \left\{Y_n^{i,(\ell)} = \mathbb{I}\{\rho_n^{i,(\ell)} > 1/2\}\right\}_{i,n}$ .
    (M-step) $\boldsymbol{\theta}^{(\ell)} = \underset{\boldsymbol{\theta}}{\text{argmin}}\, \mathcal{L}_{\mathcal{G}}(\boldsymbol{\theta};\mathcal{Y}_T^{(\ell)},\mathscr{F}_T)$ initialized
    $\boldsymbol{\theta}$ at $\boldsymbol{\theta}^{(\ell-1)}$.
  end for
output  $\boldsymbol{\theta}^{(n_{\text{iter}})}, \boldsymbol{\rho}^{(n_{\text{iter}})}$.
```

Based on this mean-field approximation, we propose a variant of the classification EM algorithm (CEM; (Celeux & Govaert, 1992)) summarized in (3). The **E**-step consists in minimizing $\bar{\mathcal{L}}\left(\boldsymbol{\rho},\boldsymbol{\theta}^{\ell-1};\mathscr{F}_T\right)$ w.r.t. the latent parameters $\boldsymbol{\rho}$. The **C**-step assigns each event to the corresponding class $\{0,1\}$ by setting $Y_n^{i,(\ell)} = \mathbb{I}\{\rho_n^{i,(\ell)} > 1/2\}$. The **M**-step amounts to minimizing $\mathcal{L}(\boldsymbol{\theta};\mathcal{Y}_T,\mathscr{F}_T)$ w.r.t. $\boldsymbol{\theta}$. Repeating these steps yields an estimation of the parameter $\boldsymbol{\theta}$, encoding the structure of the events, as well as the assignment Y_n^i of each event e_n^i to one of the two processes. This procedure constitutes the core of the UNHaP unmixing procedure. In addition to this variational procedure, fast and efficient inference in UNHaP relies on several key points described below.

Efficient parameter inference. To allow UNHaP to scale to large physiological event detection applications, the estimation of the parameters $\boldsymbol{\theta}^{(\ell)}$ in the **M**-step relies on the FaDIn framework (Staerman et al., 2023). This framework is adapted to capture delays between large events with general parametric kernels and efficient inference. It relies on three key ingredients: (1) the discretization of the timeline with a stepsize Δ, (2) the use of finite support kernels ϕ_{ij} with length W such that $\phi_{ij}(t) = 0, \forall t \notin [0,W]$, and (3) precomputations terms for the ℓ_2 loss, allowing to make the computational complexity of the optimization steps independent of the number of events. Based on these ingredients, we add an index $\mathcal{G}$ to the losses, referring to the discretization grid on the previously introduced losses (1) and (3). For details on adapting this framework to our unmixing problem, we refer the reader to Section A.1.

Minimization steps. The **E** and **M** steps of (3) are performed using gradient-based optimization on the losses $\mathcal{L}_{\mathcal{G}}(\boldsymbol{\theta}; \mathcal{Y}_T, \mathscr{F}_T)$ and $\bar{\mathcal{L}}_{\mathcal{G}}(\boldsymbol{\rho}, \boldsymbol{\theta}; \mathscr{F}_T)$. To improve the flexibility of the CEM procedure, we define a parameter b that sets the number of optimization steps conducted on $\boldsymbol{\theta}$ before updating $\boldsymbol{\rho}$. This parameter controls a trade-off between recovering the parameters of the two mixed processes and recovering the correct latent mixture structure. The gradients w.r.t. each parameter are exhibited in the Section A.3. The gradient of $\boldsymbol{\rho}$ requires the gradient of the precomputation terms w.r.t. $\boldsymbol{\rho}$. Therefore, these terms must be computed at each update of $\boldsymbol{\rho}$, *i.e.,* every b optimization steps. The bottleneck of the computation cost of UNHaP is then the updates of precomputation terms. Given a number of iterations of our solver, say n_{iter}, the total cost of the precomputation is dominated by $O\big(\lfloor n_{\text{iter}}/b \rfloor\, D^2 L^2 G\big)$, where G is the number of elements of $\mathcal{G}$ and $L = \lfloor W/\Delta \rfloor$ is the number of elements of the grid used for the kernel discretization.

Initialization with Moments Matching. As it is generally the case when inferring Hawkes processes (Lemonnier & Vayatis, 2014), the loss $\mathcal{L}_{\mathcal{G}}$ is non-convex w.r.t. its parameters and may converge to a local minimum, thus yield suboptimal parameters. The quality of these minima strongly depends on the initialization scheme used for the initial value of the baselines and the kernel parameters. A natural approach is to select them randomly. However, this option can make the algorithm unstable and yield sub-optimal parameters as the solver can fall into irrelevant local minima. Another option is to take advantage of the observed event distribution and perform moment matching to initialize the parameters. We refer to this option as “Moments Matching initialization”. Moment matching ensures that the moment of the observed distribution matches the moment of the parametric model with the initial parameter. All the mathematical details and numerical experiments demonstrating the advantages of using Moments Matching are deferred in Section A.2 and Section B.3, respectively.

4. Numerical Validation

In this section, we evaluate the benefits of UNHaP in recovering the structure of the mixture of latent variables and the parameters of the structured events on simulated data. We also compare the performance of UNHaP with other PP solvers and show that it is more robust to noise while keeping a reasonable computational cost.

4.1. Joint inference and unmixing with UNHaP

Based on simulated processes, we show that UNHaP jointly recovers the parameters of the Hawkes events and the mixture’s latent variables in various noise settings.

Simulation. With the Immigration-Birth algorithm (Møller & Rasmussen, 2005; 2006), we generate one-dimensional marked events in $[0, T] \times \mathcal{K}$ with $T = \{100, 1000, 10000\}$ and $\mathcal{K} = [0, 1]$ from the mixture process with the following intensity function

$$\lambda(t, \kappa; \boldsymbol{\theta}) = \Big(\mu + \alpha \sum_{t_n < t} Y_n \omega(\kappa_n)\phi(t - t_n; \eta)\Big) f^1(\kappa) + \tilde{\mu}\, f^0(\kappa), \tag{4}$$

where $\omega(\kappa) = \kappa$ and $Y_n = 1$ if t_n is generated by the Hawkes process. The intensity $\tilde{\mu}$ of the Poisson process is amenable to the noise level of the mixture process and α characterizes how strong the excitation structure is[1]. We denote $\boldsymbol{\alpha} = \alpha \mathbb{E}_{f^1}[\omega(\kappa)]$ the excitation level such that $\boldsymbol{\alpha} \to 1$ indicates a high excitation structure, with most events in the MMHP stemming from previous ones, while $\boldsymbol{\alpha} \to 0$ indicates no structure, as the process is almost a Poisson process. f^0 and f^1 are the marks’ distributions and we set $f^1(\kappa) = 2\kappa$ to account for a linear mark distribution for structured events. For the noisy marks, we consider two settings: one linear with $f^0(\kappa) = 2(1 - x)$ and one uniform with $f^0(\kappa) = 1$. These two cases correspond to different information levels present in the marks on the probability of being a true event. The excitation kernel $\phi(\cdot; \eta)$ is chosen as a truncated Gaussian kernel, to model delays between the events. With $\eta = (m, \sigma)$, it reads

$$\phi(\cdot; \eta) = \frac{1}{\sigma} \frac{\gamma\left(\frac{\cdot - m}{\sigma}\right)}{F\left(\frac{W - m}{\sigma}\right) - F\left(\frac{-m}{\sigma}\right)} \mathbf{1}_{0 \leq \cdot \leq W},$$

where W is the kernel length and γ (resp. F) is the probability density function (resp. cumulative distribution function) of the standard normal distribution. In our experiments, we set $\eta = (0.5, 0.1)$.

Robust parameter inference with UNHaP. To highlight the robustness of the Hawkes excitation structure recovery with UNHaP and the necessity to infer the mixture parameters, we compare the parameter recovery for different noise levels $\tilde{\mu} \in [0.1, 1.5]$. We set $\mu = 0.8$, $\alpha = 1.45$, which correspond to a process with clear structure.

We infer the MMHP’s parameters $\boldsymbol{\theta} = \{\mu, \alpha, m, \sigma\}$ with UNHaP and compare our results with a marked version of FaDIn, which we called “JointFaDIn”. We set $\Delta = 0.01$ and $W = 1$ with 10000 optimization steps for UNHaP and JointFaDIn. The number of iterations chosen between two

[1] the maximum authorized α parameter to have a stable process is such that $\alpha \mathbb{E}_{f^1}[\omega(\kappa)] = \frac{2\alpha}{3} < 1$.

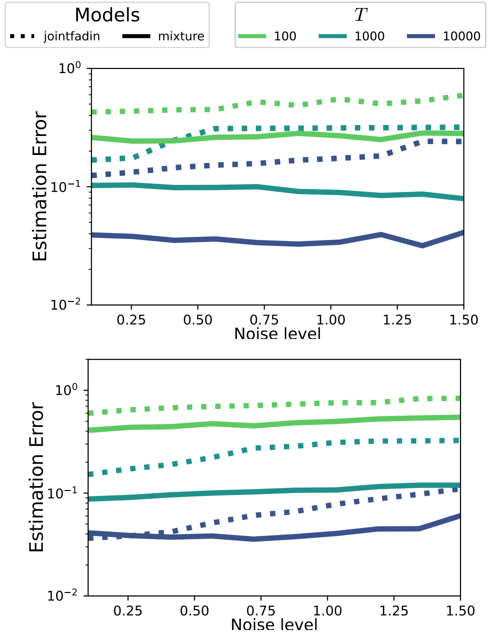

Figure 2. Parameters estimation errors for UNHaP and "jointfadin" for varying T w.r.t. different values of $\tilde{\mu}$ with linear (left) and uniform (right) distributions on noisy marks.

updates of $\hat{\rho}$ is set to $b = 200$ according to the sensitivity study depicted in Section B.1. Figure 2 reports the median value over 100 repetitions of $||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}||_2$, reflecting the error between the estimates and their actual values, for the linear marks (left) and uniform marks (right) settings. UNHaP outperforms JointFaDIn in all settings while being more robust w.r.t. the noise level $\tilde{\mu}$, as the performances remain constant. This experiment shows that accounting for the mixture's latent variables is crucial to recovering the parameters of the structured events. We also see that the linear mark distribution allows for better parameter recovery, as it is more informative to infer the mixture's latent variables.

UNHaP recovers the mixture structure. To show the performance of UNHaP to classify the observed events between the spurious and structured ones, we use the simulated processes defined above, varying $\alpha \in [0, 1]$. Here, we set $\mu = 0.4$ and $\tilde{\mu} = 0.1$. Experiments varying the noise levels ($\tilde{\mu} = 0.5$ and $\tilde{\mu} = 1$) are presented in Figure 6.

We consider the mixture parameter $\boldsymbol{\rho}$ inferred with UNHaP in the case of structured (linear setting) and unstructured (uniform) noise marks. We set $\Delta = 0.01$, $W = 1$ and $b = 200$ with 10000 optimization steps for UNHaP. In Figure 3, we report the Precision and Recall scores of the estimated mixture parameter $\hat{\rho}$ w.r.t. the ground truth. We can see the convergence $\hat{\rho}$ towards the true ρ^* when the excitation structure grows in both cases. When $\boldsymbol{\alpha}$ is small and the excitation structure absent, only the mark distribution may distinguish between the events stemming from μ and $\tilde{\mu}$. Figure 3 shows that the accuracy of $\hat{\boldsymbol{\rho}}$ stays high for a small $\boldsymbol{\alpha}$ when mark densities are different (right), but it is challenging when they overlap (left).

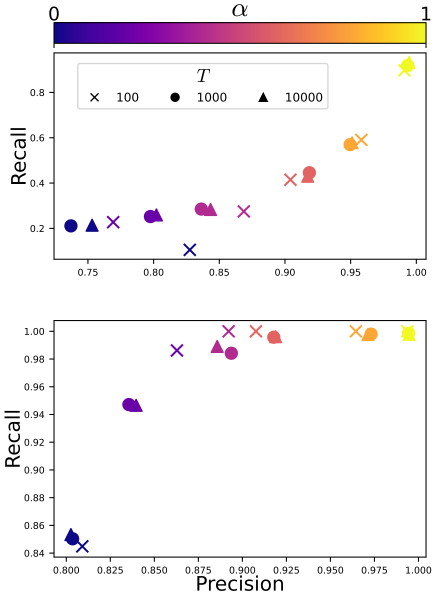

Figure 3. Precision/Recall values for the estimation of ρ for different values of T w.r.t. α with linear (left) and uniform (right) distributions on noisy marks.

4.2. Benchmarking inference and computation time

We compare UNHaP with various Hawkes process solvers by assessing approaches' statistical and computational efficiency in the case of simulated noisy and non-noisy data. Considering the few models and open-sourced code available in the marked Hawkes process area, we compare UNHaP with popular unmarked Hawkes solvers. We compare with parametric approaches 1) FaDIn (Staerman et al., 2023); 2) Neural Hawkes Process (Neural Hawkes; Mei & Eisner 2017) where authors model the intensity function with an LSTM module; and 3) Tripp (Shchur et al., 2020), where a triangular map is used to approximate the compensator function, i.e., the integral of the intensity function.

We simulate a marked Hawkes process in a high noise setting, where noisy events have a small mark compared to the Hawkes events. Its intensity function is defined as in (4), with a linear mark distribution in [0, 1]. We also simulate a Poisson Process for the noisy events, with a uniform mark distribution in [0, 0.2]. Therefore, $\omega(\kappa) = \kappa$, $f^1(\kappa) = 2\kappa$ and $f^0(\kappa) = \mathbf{1}_{0 \leq \kappa \leq 0.2}$. We set $\mu = 0.1$, $\alpha = 1$, imposing a high excitation phenomenon, and $\tilde{\mu} = 1$, corresponding to a high-noise setting.

We then conduct inference on the intensity function of the underlying Hawkes processes using UNHaP and the three

	NLL			Computation time (s)		
T	100	500	1000	100	500	1000
UNHaP	**0.624 ± 0.31**	**0.447 ± 0.12**	**0.346 ± 0.03**	96.2 ± 4.5	109.6 ± 5.9	117.4 ± 5.8
FaDIn	2.445 ± 0.19	2.442 ± 0.1	2.441 ± 0.14	41.3 ± 19.4	32.5 ± 12.8	30.9 ± 5.9
Tripp	4.27 ± 0.62	2.137 ± 0.18	1.555 ± 0.07	44.6 ± 6.7	50.9 ± 3.7	55.3 ± 3.5
Neural Hawkes	2.006 ± 0.7	1.574 ± 0.45	1.141 ± 0.2	43.4 ± 16.8	171.8 ± 38.1	183.3 ± 30.7

Table 1. Mean ± standard deviation (over ten runs) of the Negative Log-Likelihood (NLL) **on marked events in noisy settings** for various models and various sizes of events sequence.

aforementioned methods, $\Delta = 0.01$ applied consistently across all discrete approaches and $W = 1$ for FaDIn and UNHaP. This experimental procedure is replicated for various values of $T \in \{10, 500, 1000\}$. The NLL is computed on a test set simulated with parameters identical to the training data. The median NLL over ten runs and the computation time are displayed in Table 1.

In this marked and noisy setting, UNHaP demonstrates a statistical superiority over all methods. This outcome aligns with expectations in a parametric approach when the utilized kernel belongs to the same family as the one used for event simulation. It is essential to highlight that these results stem from analyzing a single (long) data sequence, contributing to the subpar statistical performance of Neural Hawkes, which excels in scenarios involving numerous repetitions of short sequences due to the considerable number of parameters requiring inference. From a computational time standpoint, UNHaP takes much longer than the other methods but is the only one converging to an accurate result. It is slower compared to FaDIn, which is expected due to the alternate minimization scheme, which performs repeated parameter inference using a procedure similar to FaDIn. UNHaP is the only successful solver in the noisy data context at a reasonable computation cost. In an unmarked setting, UNHaP performs on par with the other methods, with a slight advantage in noisy settings; see Table 3.

5. Application to Physiological Data

To demonstrate the usefulness of UNHaP in a real-world application, we use it to characterize the inter-event interval distribution in ECG and gait data. Additional experimental details are provided in Section 5. Statistics derived from ECG inter-beat intervals, such as the heart rate (HR) and the heart rate variability (HRV), are central in diagnosing heart-related health issues, like arrhythmia or atrial fibrillation (Shaffer & Ginsberg, 2017). Similarly, the study of a person's gait with inertial measurement units (IMU) is essential in diagnosing pathologies like Parkinson's disease or strokes (Truong et al., 2019), in particular by analyzing the inter-step time intervals. Computing these statistics requires a robust detection of heartbeats (Berkaya et al., 2018) or steps (Oudre et al., 2018). Classical domain-specific methods are typically used (Pan & Tompkins, 1985; Elgendi, 2013; Hamilton, 2002), in combination with heavily tailored post-processing steps (Merdjanovska & Rashkovska, 2022; Oudre et al., 2018) to cope with spurious event detection resulting from noisy signals. The design of such methods is cumbersome, requires domain expertise, and does not generalize well.

A more automatized approach to detect events is to use Convolutional Dictionary Learning (CDL; (Tour et al., 2018)). While it is more domain-agnostic than classical methods, this method is even more prone to spurious event detection. UNHaP circumvents this issue by post-processing the detected events to separate structured events from spurious ones. In the following, we use UNHaP to post-process ECG and gait events detected using CDL. We show on ECG data from the *vitaldb* (Lee et al., 2022) and gait data from (Truong et al., 2019) that our generic methodology reaches performance on par with state-of-the-art, heavily tailored methods. Our results showcase that UNHaP filters out noisy events and that the inferred parameters are coherent with the physiological data.

Experimental Pipeline for CDL+UNHaP method. We used the same method for ECG and gait recordings. In what follows, it is illustrated on ECG recordings. The proposed method relies on the CDL algorithm from the Python library `alphacsc` (Tour et al., 2018) to detect events. Denote by X an ECG slot, CDL decomposes it as a convolution between a dictionary of temporal atoms D and a sparse temporal activation vector Z: $X = Z * D + \varepsilon$. Starting from the ECG signal depicted in Figure 4 (A), panel (B1) shows the learned temporal atom on ECG, and panel (B2) shows the activation vector Z from ECG window obtained with CDL. There are non-zero activations for each beat, but they are mixed with noisy activations in Z. Handcrafted thresholding methods would typically be used here to remove noisy activations from Z, but would need to be adapted for each recording. Instead, we process the raw activation vector Z with the proposed UNHaP method. The solver separates the heartbeat Hawkes Process from the noisy activations (Figure 4 (C2)) and estimates the inter-burst interval (Figure 4

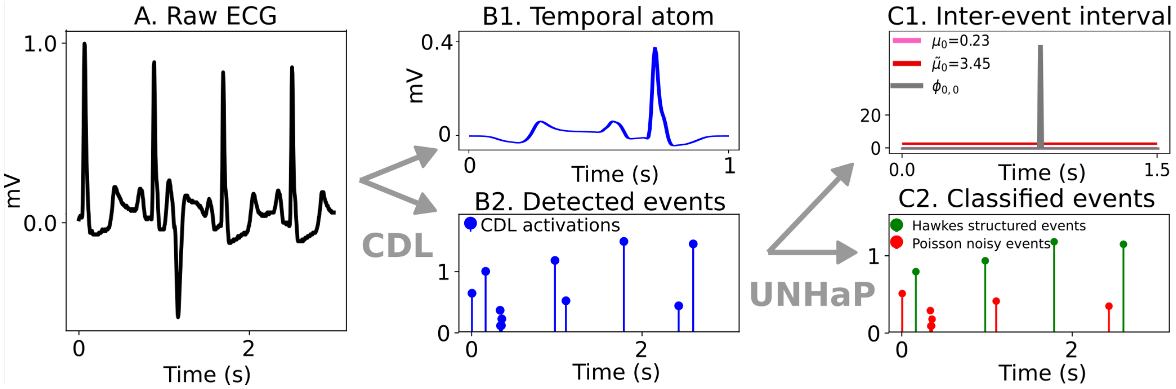

Figure 4. Experimental pipeline on ECG Data. **(A)** Sub-sample of raw ECG plot. **(B)** Output of Convolutional Dictionary Learning algorithm: **(B1)** learned temporal atom representing one heartbeat, **(B2)** detected events on the time interval. **(C)** Output of UNHaP: **(C1)** Estimated Hawkes parameters: noise baseline (red), baseline (pink), and kernel (dark green). The kernel is very close to the ground truth (orange dashed). **(C2)** Unmixing output ρ: events were classified either belonging to the Hawkes Process (**green**) or as spurious noisy events (**red**).

(C1)). The mean (respectively the standard deviation) of the parameterized truncated Gaussian ϕ estimates the mean inter-beat interval (respectively the heart rate variability) on the ECG slot X. With this example, we see that UNHaP successfully detects the structured events from the noisy ones, providing a good estimate of the inter-beat distribution.

Results. We compare several estimators of the inter-beat and inter-step interval distributions, including the developed pipeline UNHaP and with FaDIn (Staerman et al., 2023) to post-process the detected events. We also compare several domain specific libraries: `pyHRV` (Gomes, 2024), `Neurokit` (Makowski et al., 2021), which are tailored to ECG, and Template Matching (Oudre et al., 2018), a method specifically tailored for gait detection.

For ECG, the mean inter-beat interval obtained with these estimators is compared to the ground truth, which is given in the dataset. We average each estimator's absolute and relative absolute errors over the 19 ECG recordings, and report the results in Table 2. UNHaP, pyHRV, and Neurokit have equivalent performance, all providing good heart rate estimates, even though our method has a increased variance. Another interesting finding is that while working with the same detected events, UNHaP vastly outperforms FaDIn. This highlights again the benefit of our mixture model to separate structured events –here quasi-periodic– from spurious ones.

We applied the same pipelines to gait recordings. Results in Table 2 show that UNHaP performs as well as state-of-the-art on gait data, while methods designed for ECG fail to provide accurate estimates of the inter-step interval. This illustrates UNHaP's universality and robustness compared to existing methods. The pipeline developed here is agnostic and meant to be robust on a wide range of data modalities while being unsupervised and requiring no pre-processing or data adjustment.

	ECG		Gait
	AE	RAE	MAE
CDL + UNHaP	0.61±0.93	0.009±0.012	0.04
CDL + FaDIn	15.44±19.39	0.192±0.227	1.2
pyHRV	0.57±0.14	0.008±0.002	0.67
Neurokit	0.45±0.07	0.006±0.001	0.57
T. Matching	N/A	N/A	0.07

Table 2. (*left and center column*) Absolute Error (AE, in beats per minute) and Relative Absolute Error (RAE) of the estimated average inter-beat interval (mean ± std) on the *vitaldb* dataset. (*right column*) Median Absolute Error (MAE, in seconds) of the estimated average inter-step interval on the gait dataset.

6. Discussion

Having defined the challenge of distinguishing unstructured Poisson processes from structured events, this work introduces UNHaP – a model built upon a mixture marked Hawkes process designed to disentangle noise from structured events. UNHaP utilizes latent variables to represent the mixture of the two marked processes to eliminate spurious events. This is achieved by minimizing an ERM-inspired least squares loss, incorporating finite-support kernels and discretization, and ensuring reasonable computational costs. Additionally, UNHaP accommodates using any parametric form of triggering kernels, making it particularly pertinent for monitoring ECG heart rate. We demonstrate the benefits of using our unmixing models rather than the traditional Hawkes process models with simulated and real-world ECG and gait data.

References

Allain, C., Gramfort, A., and Moreau, T. DriPP: Driven Point Process to Model Stimuli Induced Patterns in M/EEG Signals. In *International Conference on Learning Representations (ICLR)*, April 2022.

Bacry, E., Mastromatteo, I., and Muzy, J.-F. Hawkes processes in finance. *Market Microstructure and Liquidity*, 1 (01):1550005, 2015.

Berkaya, S. K., Uysal, A. K., Gunal, E. S., Ergin, S., Gunal, S., and Gulmezoglu, M. B. A survey on ecg analysis. *Biomedical Signal Processing and Control*, 43:216–235, 2018.

Blei, D. M. and Jordan, M. I. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, March 2006.

Bompaire, M. *Machine learning based on Hawkes processes and stochastic optimization*. Theses, Université Paris Saclay (COmUE), July 2019. URL https://tel.archives-ouvertes.fr/tel-02316143.

Celeux, G. and Govaert, G. A classification em algorithm for clustering and two stochastic versions. *Computational statistics & Data analysis*, 14(3):315–332, 1992.

Cimolin, V. and Galli, M. Summary measures for clinical gait analysis: A literature review. *Gait & posture*, 39(4): 1005–1010, 2014.

Craik, A., He, Y., and Contreras-Vidal, J. L. Deep learning for electroencephalogram (EEG) classification tasks: A review. *Journal of Neural Engineering*, 16(3):031001, April 2019.

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

Du, N., Farajtabar, M., Ahmed, A., Smola, A. J., and Song, L. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 219–228, 2015.

Elgendi, M. Fast qrs detection with an optimized knowledge-based method: Evaluation on 11 standard ecg databases. *PloS one*, 8(9):e73557, 2013.

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23): e215–e220, 2000.

Gomes, P. PGomes92/pyhrv, January 2024. URL https://github.com/PGomes92/pyhrv. original-date: 2018-10-20T00:14:50Z.

Grosse, R., Raina, R., Kwong, H., and Ng, A. Y. Shift-Invariant Sparse Coding for Audio Classification. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 23, pp. 149–158, 2007.

Hamilton, P. Open source ecg analysis. In *Computers in cardiology*, pp. 101–104. IEEE, 2002.

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

Kirchner, M. Hawkes and INAR(∞) processes. *Stochastic Processes and their Applications*, 126(8):2494–2525, August 2016.

Kirchner, M. and Bercher, A. A nonparametric estimation procedure for the hawkes process: comparison with maximum likelihood estimation. *Journal of Statistical Computation and Simulation*, 88(6):1106–1116, 2018.

Lee, H.-C., Park, Y., Yoon, S. B., Yang, S. M., Park, D., and Jung, C.-W. Vitaldb, a high-fidelity multi-parameter vital signs database in surgical patients. *Scientific Data*, 9(1): 279, 2022.

Lemonnier, R. and Vayatis, N. Nonparametric markovian learning of triggering kernels for mutually exciting and mutually inhibiting multivariate hawkes processes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 161–176. Springer, 2014.

Li, L. and Zha, H. Dyadic event attribution in social networks with mixtures of hawkes processes. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pp. 1667–1672, 2013.

Liu, S., Yao, S., Liu, D., Shao, H., Zhao, Y., Fu, X., and Abdelzaher, T. A latent hawkes process model for event clustering and temporal dynamics learning with applications in github. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1275–1285. IEEE, 2019.

Luz, Eduardo José da S, L., Schwartz, W. R., Cámara-Chávez, G., and Menotti, D. ECG-based heartbeat classification for arrhythmia detection: A survey. *Computer Methods and Programs in Biomedicine*, 127:144–164, 2016.

Makowski, D., Pham, T., Lau, Z. J., Brammer, J. C., Lespinasse, F., Pham, H., Schölzel, C., and Chen, S. H. A. NeuroKit2: A Python toolbox for neurophysiological signal processing, February 2021. URL https:

//github.com/neuropsychology/NeuroKit. original-date: 2019-10-29T05:39:37Z.

Martinez, J. P., Almeida, R., Olmos, S., Rocha, A. P., and Laguna, P. A wavelet-based ECG delineator: evaluation on standard databases. *IEEE Transactions on Biomedical Engineering*, 51(4):570–581, 2004.

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

Merdjanovska, E. and Rashkovska, A. Comprehensive survey of computational ecg analysis: Databases, methods and applications. *Expert Systems with Applications*, 203: 117206, 2022.

Møller, J. and Rasmussen, J. G. Perfect simulation of hawkes processes. *Advances in applied probability*, 37 (3):629–646, 2005.

Møller, J. and Rasmussen, J. G. Approximate simulation of hawkes processes. *Methodology and Computing in Applied Probability*, 8:53–64, 2006.

Oudre, L., Barrois-Müller, R., Moreau, T., Truong, C., Vienne-Jumeau, A., Ricard, D., Vayatis, N., and Vidal, P.-P. Template-based step detection with inertial measurement units. *Sensors*, 18(11):4033, 2018.

Pan, J. and Tompkins, W. J. A real-time {QRS} detection algorithm. *IEEE Transactions on Biomedical Engineering*, 32(3):230–236, 1985.

Shaffer, F. and Ginsberg, J. P. An overview of heart rate variability metrics and norms. *Frontiers in public health*, pp. 258, 2017.

Shchur, O., Gao, N., Biloš, M., and Günnemann, S. Fast and flexible temporal point processes with triangular maps. In *Advances in Neural Information Processing Systems*, volume 33, pp. 73–84. Curran Associates, Inc., 2020.

Staerman, G., Allain, C., Gramfort, A., and Moreau, T. Fadin: Fast discretized inference for hawkes processes with general parametric kernels. In *International Conference on Machine Learning*, pp. 32575–32597. PMLR, 2023.

Tour, T. D. L., Moreau, T., Jas, M., and Gramfort, A. Multivariate convolutional sparse coding for electromagnetic brain signals, 2018.

Truong, C., Barrois-Müller, R., Moreau, T., Provost, C., Vienne-Jumeau, A., Moreau, A., Vidal, P.-P., Vayatis, N., Buffat, S., Yelnik, A., et al. A data set for the study of human locomotion with inertial measurements units. *Image Processing On Line*, 9:381–390, 2019.

Xiang, Y., Lin, Z., and Meng, J. Automatic qrs complex detection using two-level convolutional neural network. *Biomedical engineering online*, 17(1):1–17, 2018.

Xu, H. and Zha, H. A dirichlet mixture model of hawkes processes for event sequence clustering. *Advances in neural information processing systems*, 30, 2017.

Yang, S.-H. and Zha, H. Mixture of mutually exciting processes for viral diffusion. In *International Conference on Machine Learning*, pp. 1–9. PMLR, 2013.

A. Technical Details

A.1. Detailing UNHaP loss with Joseph E Mietus, George B Moody, Chung-Kang Peng,discretization

In the following, we assume that the functions $\omega_{ij}(\cdot)$ are identical for $1 \leq i, j \leq D$ and denote it by $\omega(\cdot)$.

Discretization and finite support kernels. Motivated by computational efficiency and the use of general parametric kernels, we adopt a setting similar to the one recently proposed by (Staerman et al., 2023). First, we discretize the time by projecting each event time t_n^i on a regular grid $\mathcal{G} = \{0, \Delta, 2\Delta, \dots, G\Delta\}$, where $G = \left\lfloor \frac{T}{\Delta} \right\rfloor$. We refer to Δ as the stepsize of the discretization and denote by $\widetilde{\mathscr{F}}_T^i$ the set of projected events of $\mathscr{F}_T^i$ on the grid $\mathcal{G}$. Second, we suppose the length of the kernels ϕ_{ij} to be finite. This assumption is consistent with scenarios in which an event's impact is limited to a relatively short time frame in the future. Examples of such applications include neuroscience (Allain et al., 2022) or high-frequency trading (Bacry et al., 2015). We denote by W the length of the kernel's support kernel, such that $\forall i, j,\ \forall t \notin [0, W], \phi_{ij}(t) = 0$. The size of the kernel of the discrete grid is then equal to $L = \left\lfloor \frac{W}{\Delta} \right\rfloor$. With these two key features, the intensity boils down to

$$\bar{\lambda}_i^1([s], \kappa; \boldsymbol{\theta}_1) = \left(\mu_i + \sum_{j=1}^{D} \sum_{\tau=1}^{L} \phi_{ij}^{\Delta}[\tau] \tilde{z}_j[s-\tau]\right) f_i^1(\kappa),$$

where $s \in [\![0, G]\!]$ and $\phi_{ij}^{\Delta}[\cdot], \tilde{z}_j[\cdot]$ are vector notations. Precisely, $\phi_{ij}^{\Delta}[s] = \phi_{ij}(s\Delta)$ and $\tilde{z}_j[s] = \sum_{t_n^j} \rho_n^j \, \omega(\kappa_n^j) \, \mathbf{1}_{\{|t_n^j - s\Delta| \leq \frac{\Delta}{2}\}}$. For notation convenience, we introduce the vectors $\rho^j[\cdot], z_j[\cdot]$ such that $\rho^j[s] = z_j[s] = 0$ when there is no events at location s and to $\rho^j[s] = \rho_n^j, z_j[s] = \omega(\kappa_n^j)$ if there is an event t_n^j at position s. Therefore, $\tilde{z}_j$ can be written as $\tilde{z}_j = \rho^j \odot z_j \in \mathbb{R}_+^{G+1}$ where $\odot$ is the Hadamard product. The computation of the intensity function is more efficient in the discrete approach, leveraging discrete convolutions with a worst-case complexity that scales as $O(N_g(T)L)$, where $N_g(T) = \sum_{i=1}^{D} N_{g_i}(T)$ is the total number of events, contrasting with the quadratic complexity w.r.t. $N_g(T)$ in general parametric kernels. The bias introduced by the discretization setting is negligible in most cases (Kirchner, 2016; Kirchner & Bercher, 2018; Staerman et al., 2023).

Efficient Inference. Our approach aims at minimizing the discretized version of $\bar{\mathcal{L}}(\boldsymbol{\rho}; \boldsymbol{\theta}, \mathscr{F}_T)$ and $\mathcal{L}(\boldsymbol{\theta}; \mathcal{Y}_T, \mathscr{F}_T)$ according to the latent mixture' parameters $\boldsymbol{\rho}$ and the process's parameters $\boldsymbol{\theta}$. Given the previous notations, we get

$$\begin{aligned}
\bar{\mathcal{L}}_{\mathcal{G}}^i(\boldsymbol{\rho}, \boldsymbol{\theta}, \widetilde{\mathscr{F}}_T) = {} & T(H_i^1 \mu_i^2 + H_i^0 \tilde{\mu}_i^2) + 2\Delta H_i^1 \mu_i \sum_{j=1}^{D} \sum_{\tau=1}^{L} \phi_{ij}^{\Delta}[\tau] \widetilde{\Phi}_j(\tau; G) \\
& + \Delta H_i^1 \sum_{j,k} \sum_{\tau=1}^{L} \sum_{\tau'=1}^{L} \phi_{ij}^{\Delta}[\tau] \phi_{ik}^{\Delta}[\tau'] \widetilde{\Psi}_{j,k}(\tau, \tau'; G) + \Delta \sum_{j=1}^{D} \sum_{\tau=1}^{L} \phi_{ij}^{\Delta}[\tau]^2 \, \widetilde{\Xi}_j(\tau; G) \\
& - 2\Bigg(\tilde{\mu}_i \sum_{(\tilde{t}_n^i, \kappa_n^i) \in \widetilde{\mathscr{F}}_T^i} f_i^0(\kappa_n^i) \left(1 - \rho^i \left[\frac{\tilde{t}_n^i}{\Delta}\right]\right) \\
& + \mu_i \sum_{(\tilde{t}_n^i, \kappa_n^i) \in \widetilde{\mathscr{F}}_T^i} f_i^1(\kappa_n^i) \rho^i \left[\frac{\tilde{t}_n^i}{\Delta}\right] + \sum_{j=1}^{D} \sum_{\tau=1}^{L} \phi_{ij}^{\Delta}[\tau] \widetilde{\Phi}_j(\tau; \widetilde{\mathscr{F}}_T^i)\Bigg),
\end{aligned}$$

where $H_i^\ell = \int_{\mathcal{K}} (f_i^\ell(\kappa))^2 \, \mathrm{d}\kappa$ for $\ell \in \{0, 1\}$ and $\widetilde{\Phi}_j(\tau; G) = \sum_{s=1}^{G} \tilde{z}_j[s-\tau]$, $\widetilde{\Psi}_{jk}(\tau, \tau'; G) = \sum_{s=1}^{G} \tilde{z}_j[s-\tau] \tilde{z}_k[s-\tau']$, $\widetilde{\Xi}_j(\tau; G) = \sum_{s=1}^{G} \left(z_j^2[s-\tau] \rho^j[s-\tau] - \tilde{z}_j^2[s-\tau]\right)$ and $\widetilde{\Phi}_j(\tau; \widetilde{\mathscr{F}}_T^i) = \sum_{(\tilde{t}_n^i, \kappa_n^i) \in \widetilde{\mathscr{F}}_T^i} f_i^1(\kappa_n^i) \rho^i \left[\frac{\tilde{t}_n^i}{\Delta}\right] \tilde{z}_j \left[\frac{\tilde{t}_n^i}{\Delta} - \tau\right]$. Conditionally to the knowledge of $\boldsymbol{\rho}$, these last four terms can be precomputed, removing the computational complexity's dependency on the number of events (here represented by the grid) during the optimization on parameters $\boldsymbol{\theta}$. The cost of computing $\widetilde{\Psi}_{j,k}(\cdot, \cdot; G)$ is dominating and requires $O(G)$ operations for each (τ, τ') and (j, k) leading to $O(D^2 L^2 G)$ as in the FaDIn framework. Note that the loss $\mathcal{L}_{\mathcal{G}}(\boldsymbol{\theta}; \mathscr{F}_T, \mathcal{Y}_T)$ can be derived identically, one may just replace the ρ_n^i by $Y_n^i = \mathbb{I}\{\rho_n^i > 1/2\}$ and removing the fourth term.

A.2. Initialization with Moments Matching

Moment matching ensures that the moment of the observed distribution matches the moment of the parametric model with the initial parameter. Let us consider a multivariate marked Hawkes Process of ground intensity functions $\{\lambda_{g_i}\}$ and ground counting processes $N_{g_1}, \dots, N_{g_D}$ being equal to the number of observed events on time interval $[0, T]$. The proposed initialization method relies on choosing initial parameters such that the empirical process expectation is equal to the expectation of the model, *i.e.*

$$N_{g_i}(T) = \mathbb{E}[N_{g_i}(T)] = \int_0^T \lambda_{g_i}(t)\, \mathrm{d}t. \tag{5}$$

This system is not fully determined as we only have one equation for multiple unknown variables. To compute a simple solution for this system, we make some extra assumptions. First, we consider that all ρ_n^i are equal to $\frac{1}{2}$. With this, we get $N_{g_i}^0(T) = \frac{N_{g_i}(T)}{2}$ and thus we can compute a moment matching value $\tilde{\mu}_i^{\mathrm{m}}$ since

$$\frac{N_{g_i}(T)}{2} = \int_0^T \lambda_{g_i}^0(s)\mathrm{d}s = T\tilde{\mu}_i \Rightarrow \tilde{\mu}_i^{\mathrm{m}} = \frac{N_{g_i}(T)}{2T}.$$

Similarly, we get $N_i^1(T) = \frac{N_{g_i}(T)}{2}$ and thus, as $N_i^1(T) = \int_0^T \lambda_{g_i}^0(s)\mathrm{d}s$, we get

$$\frac{N_{g_i}(T)}{2} = \mu_i T + \sum_{j=1}^{D} \alpha_{i,j}^{\mathrm{m}} \sum_{(\tilde{t}_n^j, \kappa_n^j) \in \widetilde{\mathscr{F}}_T^j} \omega(\kappa_n^j).$$

Once again, we have only one equation with $D + 1$ unknown parameters. We choose to assume that each parameter will generate the same amount of events, leading to

$$\mu_i^{\mathrm{m}} = \frac{N_{g_i}(T)}{2T(D+1)},$$

and

$$\alpha_{i,j}^{\mathrm{m}} = \frac{N_{g_i}(T)}{2T(D+1)\sum_{(\tilde{t}_n^j, \kappa_n^j) \in \widetilde{\mathscr{F}}_T^j} \omega(\kappa_n^j)}.$$

Replacing these values for $\tilde{\mu}_i^m$, μ_i^m, and $\alpha_{i,j}^m$ into (5) ensures that the number of events' expectation for the parametric model matches the one from the observed process. The other kernel parameters are initialized using the method of moments on the delay between events. Denoting by $\delta t_n^{i,j}$ the delay between t_n^i and the time of occurrence of the last event in channel j before t_n^i

$$\delta t_n^{i,j} = t_n^i - \max\{t | t \in \mathscr{F}_T^j, W < t < t_n^i\}. \tag{6}$$

For the truncated Gaussian kernel, defined in Section 4.1, the initial mean $m_{i,j}^{\mathrm{m}}$ and standard deviation $\sigma_{i,j}^{\mathrm{m}}$ are

$$m_{i,j}^{\mathrm{m}} = \frac{1}{N_{g_i}(T)} \sum_{t_n^i \in \mathscr{F}_T^i} \delta t_n^{i,j},$$

$$\sigma_{i,j}^{\mathrm{m}} = \sqrt{\frac{\sum_{t_n^i \in \mathscr{F}_T^i} (\delta t_n^{i,j} - m_{i,j}^{\mathrm{m}})^2}{N_{g_i}(T) - 1}}.$$

For the raised cosine kernel, detailed in the Section B.3, initial parameters $u_{i,j}^{\mathrm{m}}$ and $s_{i,j}^{\mathrm{m}}$ are computed similarly

$$u_{i,j}^{\mathrm{m}} = \max\left(0, m_{i,j}^{\mathrm{m}} - \sigma_{i,j}^{\mathrm{m}}\right),$$
$$s_{i,j}^{\mathrm{m}} = \sigma_{i,j}^{\mathrm{m}}.$$

The benefits of this approach is supported by the numerical studies in Section B.3. The moment matching initialization significantly improves convergences and lowers the risk of converging to irrelevant parameter values in the case of the raised cosine, while it behaves comparably in the case of the truncated Gaussian, see Figure 7.

For very noisy settings, where noisy events are very close to Hawkes process events in time, using the $\delta t_n^{i,j}$ defined in (6) leads to poor performance of UNHaP. This is because $\delta t_n^{i,j}$ is then tiny, leading to a very small initial mean, from which the solver has trouble converging to correct values. We circumvented this issue by computing $\delta t_n^{i,j}$ with a mean instead of a maximum.

$$\delta t_n^{i,j} = t_n^i - \frac{1}{\#\{t \in \mathscr{F}_T^j, W < t < t_n^i\}} \sum_{t \in \mathscr{F}_T^j, W<t<t_n^i} t. \tag{7}$$

A.3. Gradients of the UNHaP loss

This part present the derivation of the gradients of the loss function minimized by UNHaP for each parameter.

Gradient of the baseline. For any $m \in \{1, \dots, D\}$, we get

$$\frac{\partial \bar{\mathcal{L}}_{\mathcal{G}}}{\partial \mu_m} = 2TH_m^1 \mu_m + 2\Delta H_m^1 \sum_{j=1}^{D} \sum_{\tau=1}^{L} \phi_{mj}^{\Delta}[\tau] \widetilde{\Phi}_j(\tau; G) - 2 \sum_{(\tilde{t}_n^m, \kappa_n^m) \in \mathscr{F}_T^m} f_m^1(\kappa_n^m) \rho^m \left[\frac{\tilde{t}_n^m}{\Delta}\right]$$

Gradient of the noise baseline. For any $m \in \{1, \dots, D\}$, we get

$$\frac{\partial \bar{\mathcal{L}}_{\mathcal{G}}}{\partial \tilde{\mu}_m} = 2TH_m^0 \tilde{\mu}_m - 2 \sum_{(\tilde{t}_n^m, \kappa_n^m) \in \mathscr{F}_T^m} f_m^0(\kappa_n^m) \left(1 - \rho^m \left[\frac{\tilde{t}_n^m}{\Delta}\right]\right).$$

Gradient of the excitation kernel parameters. For any tuple $(m, l) \in \{1, \dots, D\}^2$, the gradient of η_{ml} is

$$\begin{aligned}\frac{\partial \bar{\mathcal{L}}_{\mathcal{G}}}{\partial \eta_{ml}} =& 2\Delta H_m^1 \mu_m \sum_{\tau=1}^{L} \frac{\partial \phi_{ml}^{\Delta}[\tau]}{\partial \eta_{ml}} \, \widetilde{\Phi}_l(\tau; G) + 2\Delta H_m^1 \sum_{k=1}^{D} \sum_{\tau=1}^{L} \sum_{\tau'=1}^{L} \phi_{mk}^{\Delta}[\tau'] \frac{\partial \phi_{ml}^{\Delta}[\tau]}{\partial \eta_{ml}} \, \widetilde{\Psi}_{l,k}(\tau, \tau'; G) \\ &+ 2\Delta \sum_{\tau=1}^{L} \frac{\partial \phi_{ml}^{\Delta}[\tau]}{\partial \eta_{ml}} \phi_{ml}^{\Delta}[\tau] \, \widetilde{\Xi}_l(\tau; G) - 2 \sum_{\tau=1}^{L} \frac{\partial \phi_{ml}^{\Delta}[\tau]}{\partial \eta_{ml}} \, \widetilde{\Phi}_l(\tau; \widetilde{\mathscr{F}}_T^m).\end{aligned}$$

Gradient of the mixture parameter. For any $m \in \{1, \dots, D\}$ and for any $u \in [\![1\,, N_{g_m}(T)]\!]$, we have

$$\begin{aligned}\frac{\partial \bar{\mathcal{L}}_{\mathcal{G}}}{\partial \rho_m[u]} &= 2\Delta \sum_{i=1}^{D} H_i^1 \mu_i \sum_{\tau=1}^{L} \phi_{im}^{\Delta}[\tau] \left(\sum_{s=1}^{G} z_m[u] \, \mathbb{I}\{u = s - \tau\} \right) \\ &+ 2\Delta \sum_{i,k} H_i^1 \sum_{\tau=1}^{L} \sum_{\tau'=1}^{L} \phi_{im}^{\Delta}[\tau] \phi_{ik}^{\Delta}[\tau'] \left(\sum_{s=1}^{G} \tilde{z}_k[s - \tau'] z_m[u] \, \mathbb{I}\Big\{u = s - \tau\Big\} \right) \\ &+ \Delta \sum_{i=1}^{D} \sum_{\tau=1}^{L} \phi_{im}^{\Delta}[\tau]^2 \left(\sum_{s=1}^{G} z_m[u] (z_m[u] - 2\tilde{z}_m[u]) \, \mathbb{I}\Big\{u = s - \tau\Big\} \right) \\ &- 2 \Bigg(- \tilde{\mu}_m \sum_{(\tilde{t}_n^m, \kappa_n^m) \in \widetilde{\mathscr{F}}_T^m)} f_i^0(\kappa_n^m) \, \mathbb{I}\Big\{u = \frac{\tilde{t}_n^m}{\Delta}\Big\} + \mu_m \sum_{(t_n^m, \kappa_n^m) \in \widetilde{\mathscr{F}}_T^m)} f_i^1(\kappa_n^m) \, \mathbb{I}\Big\{u = \frac{\tilde{t}_n^m}{\Delta}\Big\} \\ &+ \sum_{j=1}^{D} \sum_{\tau=1}^{L} \phi_{mj}^{\Delta}[\tau] \sum_{(\tilde{t}_n^m, \kappa_n^m) \in \mathscr{F}_T^m} f_m(\kappa_n^m) \tilde{z}_j[u - \tau] \mathbb{I}\Big\{u = \frac{\tilde{t}_n^m}{\Delta}\Big\} \\ &+ \sum_{i=1}^{D} \sum_{\tau=1}^{L} \phi_{im}^{\Delta}[\tau] \sum_{(\tilde{t}_n^i, \kappa_n^i) \in \mathscr{F}_T^i} f_i(\kappa_n^i) \rho^i \, [u + \tau] \, z_m \, [u] \, \mathbb{I}\Big\{u = \frac{\tilde{t}_n^i}{\Delta} - \tau\Big\} \Bigg).\end{aligned}$$

B. Additional Experiments

B.1. Sensitivity analysis of the alternate minimization parameter

The alternate minimization performed in UNHaP depends on a parameter b, the number of optimization steps done on the Hawkes parameters between each update of $\boldsymbol{\rho}$. It controls the trade-off between the number of gradients of the point process parameters and the latent variable $\boldsymbol{\rho}$. This part presents a sensitivity analysis of this parameter across several optimization iterations.

We conduct the experiment as follows. We simulate two univariate marked Hawkes processes with intensity functions defined as in (4), the first one corresponding to the non-noisy setting with $\tilde{\mu} = 0.1$ and the second one to the noisy setting with $\tilde{\mu} = 1$. We set $T = 1000$ for both settings. We set $\omega(\kappa) = \kappa$ and $f(\kappa) = 2\kappa\, \mathbf{1}_{0 \leq \kappa \leq 1}$ and the $g(\kappa) = \mathbf{1}_{0 \leq \kappa \leq 1}$. We set $\mu = 0.8$, $\alpha = 1.4$, imposing a high excitation phenomenon, and select ϕ^{η} to be a truncated Gaussian kernel with $W = 1$ and $\eta = (m, \sigma) = (0.5, 0.1)$.

We conduct inference on the intensity function of the underlying Hawkes processes using UNHaP with $\Delta = 0.01$, $W = 1$ and varying the value of b in $\{10, 25, 50, 75, 100, 200\}$. The median and the 25%-75% quantiles (over ten runs) of the estimation parameter are depicted in Figure 5 (left) according to the number of iterations and the size of b. The median precision score (over ten runs) of the estimated $\hat{\boldsymbol{\rho}}$ recovery of the mixture structure parameter $\boldsymbol{\rho}$ is reported in Figure 5 (middle). In both cases, and those for the two noisy and non-noisy settings, the size of b reversely orders the accuracy at a computational cost; see Figure 5 (right). However, the precision for each size b is close to each other after 10000 iterations. Regarding the computational cost, we advise to select $b = 200$ for UNHaP.

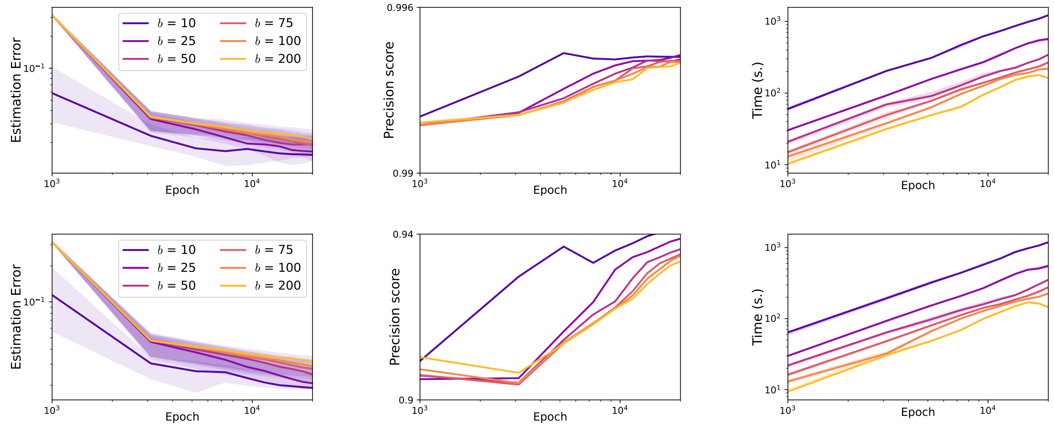

Figure 5. Inference comparison regarding the batch size of Hawkes parameters gradients between each ρ update. The error estimation on Hawkes parameters (left), the Precision score on the ρ recovering (middle) and the associated computational time (right) are displayed for non-noisy (top) and noisy settings (bottom).

B.2. Further experiments on the recovery of the mixture structure

Figure 6 displays the same experiment as in Section 4.1 but with two different noise level $\tilde{\mu} = 0.5$ and $\tilde{\mu} = 1$. These additional experiments confirm and reinforce the claims made in the core paper regarding the recovery of the mixture structure of Hawkes processes polluted by Poisson processes.

B.3. Moment Matching initialization

This section investigates the advantages of using the Moment Matching initialization introduced in Section A.2 over the classical random ones. The simulation study is conducted as follows. Relying on an Immigration-Birth algorithm (Møller & Rasmussen, 2005; 2006), we simulate one-dimensional marked events in $[0, T] \times \mathcal{K}$ with $T = \{100, 1000, 10000\}$ from the mixture process with the following intensity function

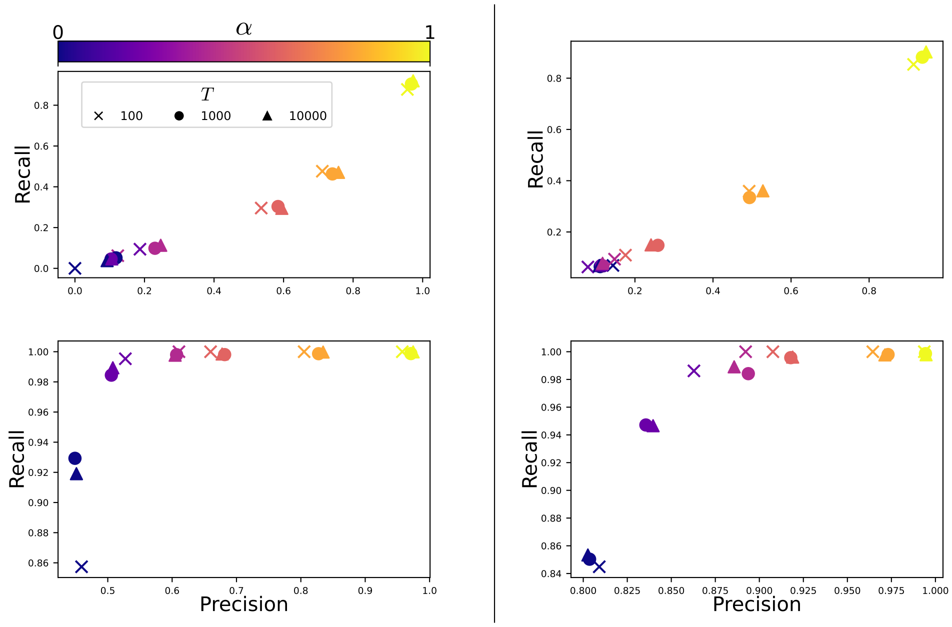

Figure 6. Precision/Recall values for the estimation of $\boldsymbol{\rho}$ for different values of T w.r.t. the auto-excitation parameter α with uniform (top) and linear (bottom) distributions on noisy marks for $\tilde{\mu} = 0.5$ (left) and $\tilde{\mu} = 1.$ (right).

$$\lambda(t, \kappa; \boldsymbol{\theta}) = \left(\mu + \alpha \sum_{t_n < t} \omega(\kappa_n)\phi(t - t_n; \eta)\right) f^1(\kappa) + \tilde{\mu}\, f^0(\kappa),$$

where $\omega(\kappa) = \kappa$ and $f^1(\kappa) = 2\kappa\, \mathbf{1}_{0\leq\kappa\leq 1}$. We define two settings of mark noise distribution: the "linear" where $f^0(\kappa) = 2(1-\kappa)\, \mathbf{1}_{0\leq\kappa\leq 1}$ and the "uniform" $f^0(\kappa) = \mathbf{1}_{0\leq\kappa\leq 1}$. We set $\mu = 0.8$ and $\alpha = 1.4$ and $\tilde{\mu} = 0.5$. Two excitation kernels $\phi(\cdot;\eta)$ are chosen. the first one is a truncated Gaussian, with $\eta = (m, \sigma)$, corresponding to

$$\phi(\cdot;\eta) = \frac{1}{\sigma} \frac{\gamma\left(\frac{\cdot - m}{\sigma}\right)}{F\left(\frac{W-m}{\sigma}\right) - F\left(\frac{-m}{\sigma}\right)} \mathbf{1}_{0\leq\cdot\leq W},$$

where W is the kernel length and γ (resp. F) is the probability density function (resp. cumulative distribution function) of the standard normal distribution.The second one is a raised cosine density defined as

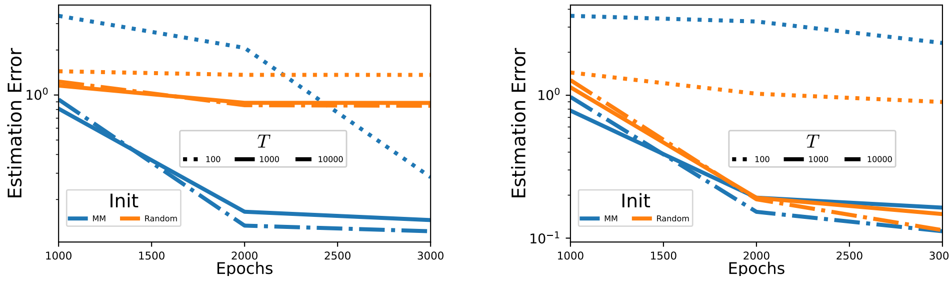

Figure 7. Hawkes parameters estimation error using UNHaP with a raised cosine (left) and a truncated Gaussian (right) kernels, Moments Matching (blue), and Random (orange) initializations for varying size sequences.

$$\phi(\cdot;\eta) = \alpha\left[1 + \cos\left(\frac{\cdot - u}{\sigma}\pi - \pi\right)\right] \mathbf{1}_{u\leq\cdot\leq u+2\sigma},$$

with $\eta = (u, \sigma)$. In contrast to the truncated Gaussian, the support of this kernel directly depends on its parameters and may induce some instability in the optimization. For the truncated Gaussian kernel, we set $\eta = (m, \sigma) = (0.5, 0.1)$ while we set $\eta = (u, \sigma) = (0.4, 0.1)$ for the raised cosine.

We compute UNHaP with both Moments Matching and Random initialization (with $b = 200$, $\Delta = 0.01$) and report the error estimation (median over 10 runs) between the true parameters $\boldsymbol{\theta} = \{\mu, \alpha, \eta\}$ and the estimated ones $\hat{\boldsymbol{\theta}} = \{\hat{\mu}, \hat{\alpha}, \hat{\eta}\}$, i.e., $||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}||_2$.

While the moment matching improves the convergence results of the parameter over the random initialization in the case of a raised cosine kernel, it behaves comparably in the case of a truncated Gaussian. This supports the average superiority of the moment matching over the random initialization and should be used consistently.

B.3.1. Benchmarking inference and computation time in an unmarked setting

The benchmark presented in Section 4.2 is done on simulated events with marks. However, the benchmarked methods do not account for marks, except for UNHaP, due to the scarcity of literature on marked point processes. To be exhaustive in our comparison, we present additional benchmarks of UNHaP, FaDIn, Tripp, and Neural Hawkes on unmarked events here.

	NLL								
	Non-noisy			Noisy			Computation time (s)		
T	100	500	1000	100	500	1000	100	500	1000
UNHaP	-0.18	**-1.7**	**-1.62**	**1.18**	**-1.23**	**-1.20**	29	31	35
FaDIn	**-0.19**	**-1.7**	**-1.62**	1.2	-1.18	-1.17	3	3	3
Tripp	2.9	-0.26	-0.98	5.4	2	1.71	19	27	31
Neural Hawkes	0.57	-1.27	-1.46	2.9	1.87	1.66	20	149	281

Table 3. Median (over ten runs) Negative Log-Likelihood (NLL) **on unmarked events** in noisy and non-noisy settings for various models and various sizes of events sequence. Bold numbers correspond to the best results. Computation time associated with the non-noisy setting is also reported.

The events are simulated similarly to in Section 4.2. We simulate a Marked Hawkes process. Its intensity function is defined as in (4). We also simulate a Poisson Process for the noisy events. The marks are not taken into account here. Therefore, $\omega(\cdot)$, $f^1(\cdot)$ and $f^0(\cdot)$ are a Dirac function in one, $\delta_1(\cdot)$. Similarly to the benchmark in Section 4.2, we set $\mu = 0.1$, $\alpha = 1$, imposing a high excitation phenomenon, and select $\phi(\cdot; \eta)$ to be a truncated Gaussian kernel with width $W = 1$ and parameters $\eta = (m, \sigma) = (0.5, 0.1)$. We benchmarked the methods on two noise settings: the non-noisy setting ($\tilde{\mu} = 0.1$) and the noisy setting ($\tilde{\mu} = 1$). Once the data is simulated, the inference and testing of the methods are done as developed in Section 4.2.

The median Negative Log-Likelihood and computational time are shown in Table 3. UNHaP demonstrates statistical superiority over all methods in a noisy environment while exhibiting comparable performance to FaDIn in a non-noisy context. This outcome aligns with expectations in a parametric approach when the utilized kernel belongs to the same family as the one used for event simulation. It is essential to highlight that these results stem from analyzing a single (long) data sequence, contributing to the subpar statistical performance of Neural Hawkes. It excels in scenarios involving numerous repetitions of short sequences due to the considerable number of parameters requiring inference. From a computational time standpoint, UNHaP performs similarly to Tripp, and significantly faster than Neural Hawkes. It is also slower than FaDIn, which is expected due to the alternate minimization scheme, which performs repeated parameter inference using a procedure similar to that of FaDIn. UNHaP offers an interesting alternative to existing methods in the context of unmarked noisy data at a reasonable computation cost.

B.4. Application to physiological data

B.4.1. ECG

Electrocardiograms (ECG) measure the electrical activity of the heart. They are the gold standard for observing heartbeats. Statistics derived from ECG, such as the heart rate (HR, average number of beats per minute) and the heart rate variability, are central in diagnosing heart-related health issues, like arrhythmia or atrial fibrillation (Shaffer & Ginsberg, 2017). These statistics require a robust estimate of the inter-beat interval duration. To automatically measure the inter-beat interval, the first step is to accurately detect heartbeats (Berkaya et al., 2018). This is usually done using knowledge-based methods based on analyses of slope, amplitude, and width of ECG waves (Pan & Tompkins, 1985; Elgendi, 2013; Hamilton, 2002). However, raw ECG signals usually contain noise, which can lead to spurious event detection unrelated to the biological source of interest. These noisy events cause classical solvers to fail to recover the heart rate variability correctly. The usual route to circumvent this problem is handmade. It applies a post-processing step to the detected events, for instance, by thresholding them by amplitude or time-filtering them (Merdjanovska & Rashkovska, 2022). The design of such a step is cumbersome, requires domain expertise, and does not generalize well. Indeed, ECG recordings often have considerable inter-individual variability, so it has no "one-fits-all" value.

The procedure we developed circumvents this problem by using the structure of the detected event location to remove spurious events. The underlying mixture model separates the data into events caused by the underlying Hawkes Process and events caused by noise. In the following, we use UNHaP to post-process ECG events detected using CDL. Our results showcase that UNHaP filters out noisy events, and the obtained Hawkes process parameters are consistent with the biological ground.

Experimental Pipeline. Experiments are run on ECG data from the *vitaldb* dataset (Lee et al., 2022; Goldberger et al., 2000). Nineteen 5-minute long ECG slots were isolated among 7 patients and downsampled from 500 Hz to 200 Hz to reduce the computational cost. Figure 4 (A) shows a 3-second extract of an ECG slot. Each upward peak is a heartbeat. This succession of events is very regular and almost periodic. Hence, it is appropriate to model it with an MMHP and parameterize it with UNHaP. The downward peak at 1.5s is an example of an artifact. Below, we describe the event detection and UNHaP parameterization, done on each ECG slot separately.

We run a CDL algorithm to detect events using the Python library `alphacsc` (Tour et al., 2018). Denote by X an ECG slot, CDL decomposes it as a convolution between a dictionary of temporal atoms D and a temporal activation vector Z: $X = Z * D + \varepsilon$. Figure 4 (B1) shows the learned temporal atom on ECG slot 1, and Figure 4 (B2) shows the learned activation vector Z from ECG window in Figure 4 (A). There is at least one non-zero activation for each beat. Z could, therefore, be used as a proxy for event detection. In addition, noisy events are visible in Z: some are very close to beat activations, and some are caused by the ECG artifact at 1.5s. Handcrafted thresholding methods would typically be used here to remove noisy activations from Z. Instead, we process the raw activation vector Z, which is composed of sparse events, with our UNHaP solver with a truncated Gaussian kernel. The solver separates the heartbeat Hawkes Process from the noisy activations (Figure 4 (C2)) and estimates the inter-burst interval (Figure 4 (C1)). The mean (respectively the standard deviation) of the parameterized truncated Gaussian ϕ estimates the mean inter-beat interval (respectively the heart rate variability) on the ECG slot X. With this example, we see that UNHaP successfully detects the structured events from the noisy ones, providing a good estimate of the inter-beat distribution.

Results. We compare the error made by several estimators of the inter-beat interval, including the developed pipeline with FaDIn (Staerman et al., 2023) and UNHaP initialized with moment matching. We benchmark domain specific Python libraries `pyHRV` (Gomes, 2024) and `Neurokit` (Makowski et al., 2021), which are tailored to ECG.

The mean inter-beat interval given by these estimators is compared to the ground truth, which is given in the dataset. We average each estimator's absolute and relative absolute errors over the nineteen ECG slots, see Table 4. UNHaP, pyHRV, and Neurokit have equivalent performance. They all provide good heart rate estimates. Another interesting finding is that while working with the same detected events, UNHaP vastly outperforms FaDIn. This proves the benefit of our mixture model to separate structured events –here quasi-periodic– from spurious ones.

	CDL + UNHaP	CDL + FaDIn	pyHRV	Neurokit
AE (beats/min)	0.61±0.93	15.44±19.39	0.57±0.14	0.45±0.07
RAE	0.009±0.012	0.192±0.227	0.008±0.002	0.006±0.001

Table 4. Absolute Error (AE) and Relative Absolute Error (RAE) of the estimated average inter-beat interval (mean ± std) on the *vitaldb* dataset.

B.4.2. GAIT

The study of a person's manner of walking, or gait, is an important medical research field. Widespread pathologies, such as Parkinson's disease, arthritis, and strokes, are associated with an alteration of gait. Gait analysis is usually done by setting an inertial measurement unit to a patient's ankle and recording its vertical acceleration. These recordings can detect and infer essential features, such as steps, inter-step time intervals, and gait anomalies. We applied CDL + UNHaP to gait inertial measurement unit recordings. Our pipeline detects steps and infers the inter-step time interval from raw gait inertial measurement unit data (Truong et al., 2019). We found that CDL+UNHaP performs at least as well as domain-specific methods.

Experimental pipeline The experimental pipeline is the same as described in Section 5. We run a CDL algorithm to detect steps using the Python library `alphacsc` (Tour et al., 2018). The dictionary contains 1 atom of 1.5 seconds, and its loss is minimized with a regularization factor of 0.5. Detected events are then fed to the UNHaP solver. The Hawkes parameters are initialized with mean moment matching. The UNHaP gradient descent is done over 20,000 iterations, and the mixture parameter ρ is updated every 1000 iterations.

Results We benchmarked our method to other estimators, similarly to Section 5 . We compare the error made by several estimators of the inter-step interval, namely CDL + UNHaP developed in Section B.4.2 and CDL+FaDIn (Staerman et al., 2023). We compare these estimators with Template Matching (Oudre et al., 2018), a method specifically tailored for gait detection. Finally, the benchmark also includes the ECG Python libraries `pyHRV` (Gomes, 2024) and `Neurokit` (Makowski et al., 2021), which were benchmarked in Section 5. The results are shown in Section B.4.2. They highlight that as for ECG, UNHaP performs on par with Template Matching (Oudre et al., 2018), which is state-of-the-art for gait detection. Additionally, contrary to previous Template Matching, UNHaP does not require pre-processing or application-specific tailoring. UNHaP performs much better on gait than the ECG methods pyHRV and Neurokit, illustrating its universality. We also stress that the proposed unmixing problem is critical to achieving good performance, as highlighted by the failure of CDL+FaDIn, which has no unmixing.

	CDL + UNHaP	CDL + FaDIn	pyHRV	Neurokit	Template Matching
MAE (seconds)	**0.04**	1.2	0.67	0.57	0.07

Table 5. Median Absolute Error (MAE) of the estimated average inter-step interval on gait.