UNHaP: Unmixing Noise from Hawkes Processes

Virginie Loison
Université Paris-Saclay,
Inria, CEA,
Palaiseau, France.
Université Paris Cité,
Inserm, APHP,
Paris, France.

Guillaume Staerman
Université Paris-Saclay,
Inria, CEA,
Palaiseau, France.

Thomas Moreau
Université Paris-Saclay,
Inria, CEA,
Palaiseau, France.

Abstract

Physiological signal analysis often involves identifying events crucial to understanding biological dynamics. Many methods have been proposed to detect them, from handcrafted and supervised approaches to unsupervised techniques. All these methods tend to produce spurious events, particularly as they detect each event independently. 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. By treating the event detection output as a mixture of structured Hawkes and unstructured Poisson events, UNHaP efficiently unmixes these processes and estimates their parameters. This approach significantly enhances event distribution characterization while minimizing false detection rates on simulated and real data.

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 step identification in inertial measurement unit recordings of the gait, which is a crucial feature in classifying several pathologies (Cimolin and Galli, 2014).

Several approaches have been proposed to automatize event detection. 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 and Tompkins, 1985) or wavelet-based approaches (Martinez et al., 2004). While these algorithms perform well, they require extensive domain expertise, and their parameters tend to be sensitive 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. Other data-driven approaches leverage unsupervised learning to extract repeating patterns, such as the wavelet-based element analysis (Lilly, 2017) and 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. With each detected event, they provide a positive scalar quantifying the event amplitude, or the confidence associated with the event detection. While these solutions can be applied without domain expertise, 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. This paper proposes a novel automatized framework to filter out spurious events based on their temporal distribution and event detection confidence. A key observation

Corresponding author: Virginie Loison, `virginie.loison@inria.fr`.

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

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. For instance, the inter-heartbeat interval is around one second for a normal ECG.

In this paper, we propose to automatically classify detected events between spurious and structured ones by jointly learning the temporal structure of the events and filtering out spurious ones based on their low probability in the temporal structure.

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 events' distribution (Allain et al., 2022; Staerman et al., 2023). Yet, these models cannot account for the confidence associated with event detection. This confidence could be quantified by marks. Therefore, these models must be extended to deal with marked PP (Daley et al., 2003) and to integrate marks in the intensity function.

In addition, inference with these models only works when all events come from the same process. In the context of unsupervised event detection, 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 and Zha, 2013), sequences of events (Xu and Zha, 2017), or to identify exogenous events from induced ones with declustering (Zhuang et al., 2002; Linderman and Adams, 2014; Zhang et al., 2021). They rely on feature-based mixture models (Li and Zha, 2013; Yang and Zha, 2013; Du et al., 2015) or associate a Dirichlet process to classical Hawkes models (Blei and Jordan, 2006). While these approaches are tailored to find different auto-excitation patterns in multiple event streams, they are not designed to unmix uninformative events from structured ones in a single event history. To the best of our knowledge, only Bonnet et al. (2024) consider a similar unmixing problem using spectral analysis. However, their work can only be applied to exponential kernels, which are not adapted to physiological signals.

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 framework, the output of the event detection algorithm is treated as a mixture, or superposition, of events of interest and spurious events. The events of interest are temporally structured and, therefore, modeled as a Hawkes process structure. Consistently with physiological signals, the spurious events are unstructured and not time-dependent, which we model as a Poisson process. UNHaP aims to distinguish between these two processes, select structured events, and discard the spurious ones. We propose an efficient algorithm to jointly unmix these events and estimate the parameters of the Hawkes process building up from the FaDIn framework (Staerman et al., 2023). The associated code was added to the FaDIn Python package . We illustrate the benefits of using our unmixing models in simulations and with real-world ECG and gait data.

2 BACKGROUND ON MARKED HAWKES PROCESSES

A marked Hawkes process (MHP) 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 integrate the event type in the mark. The main text is written in the univariate setting, and we derive an extension of our framework to multivariate settings in Section A.1. We here give our notation and basic information about MHP and refer the reader to (Daley et al., 2003, Sec. 6) for a detailed account of these processes.

Counting processes. Let $\mathscr{F}_T = \{(t_n, \kappa_n) : \kappa_n \in \mathcal{K},\ t_n \in [0,T]\}$ be a set of observed marked events with $t_n \in [0,T]$ the time where the n-th event occurs and $\kappa_n \in \mathcal{K}$ its associated mark. We denote by $\mathbf{N}$ the random counting measure defined on $[0,T] \times \mathcal{K}$, such that $\mathbf{N}(\mathrm{d}t, \mathrm{d}\kappa) = \sum_{n=1}^{\infty} \delta_{(t_n,\kappa_n)}(\mathrm{d}t, \mathrm{d}\kappa)$, 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 $\mathcal{K}$, typically a subset of $\mathbb{R}_+$. From this measure, we can define the marginal time arrival process, also called ground process, as $N(T) = \int_{[0,T]\times\mathcal{K}} \mathbf{N}(\mathrm{d}t, \mathrm{d}\kappa) = \sum_{n \geq 1} \mathbf{1}_{t_n \leq T}$.

Intensity function. The behavior of an MHP can be described by its intensity function. Conditionally to observed events, it describes the instantaneous event rate at any given time. Given an MHP and a set of observation $\mathscr{F}_T$, each ground process N is described by the following conditional ground intensity function

$$\lambda_g(t|\mathscr{F}_t) = \mu + \int_0^t \int_{\mathcal{K}} h(t-u, \kappa)\ \mathbf{N}(\mathrm{d}u, \mathrm{d}\kappa),$$

where μ is the baseline rate and $h : \mathbb{R}_+ \times \mathcal{K} \to \mathbb{R}_+$ is

https://github.com/mind-inria/FaDIn

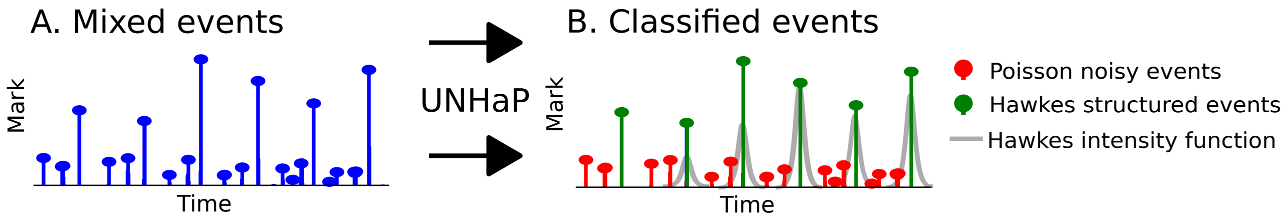

Figure 1: **Illustration of the UNHaP framework.** The goal of UNHaP is to distinguish between structured events (**green**) and spurious ones (**red**) by identifying the structure of the MHP (**grey**) from the observed events (**blue**).

the triggering or kernel function, quantifying the influence of the past events onto the 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 $h(t, \kappa) = \phi(t)\omega(\kappa)$. This leads to

$$\begin{aligned}\lambda_g(t|\mathscr{F}_t) &= \mu + \int_0^t \int_{\mathcal{K}} \omega(\kappa)\ \phi(t-u)\ \mathbf{N}(\mathrm{d}u \times \mathrm{d}\kappa) \\ &= \mu + \sum_{n, t_n < t} \omega(\kappa_n)\ \phi(t - t_n),\end{aligned}$$

with $\omega : \mathcal{K} \to \mathbb{R}_+$, $\phi : \mathbb{R}_+ \to \mathbb{R}_+$ such that $\int_0^\infty \phi(t)\mathrm{d}t < 1$ and $\int_{\mathcal{K}} \omega(\kappa)\mathrm{d}\kappa < 1$. These conditions ensure the existence and stability of such processes. The function $\omega(\cdot)$ weights the probability that a future event occurs depending on the past events' marks. Assuming a density function $f : \mathcal{K} \to \mathbb{R}_+$, we define the joint intensity function as $\lambda(t, \kappa) = \lambda_g(t|\mathscr{F}_t)\ f(\kappa)$, where the process depends on the mark distribution reflected by f and the distribution of the influence of the mark described by ω.

ERM-based inference. Inference for MHP 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 general kernels (Staerman et al., 2023). In this paper, we instead resort to the Empirical Risk Minimization (ERM)-inspired least squares loss (refer to Eq. (II.4) in Bompaire 2019, Chapter 2). Given some observed events $\mathscr{F}_T$, the goal is to minimize

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, \mathscr{F}_T) = &\int_0^T\!\!\int_{\mathcal{K}} \lambda(s, \kappa; \boldsymbol{\theta})^2\ \mathrm{d}\kappa\mathrm{d}s \\ &- 2 \sum_{(t_n, \kappa_n) \in \mathscr{F}_T} \lambda(t_n, \kappa_n; \boldsymbol{\theta}),\end{aligned} \tag{1}$$

where $\boldsymbol{\theta} = \{\mu, \phi, \omega\}$ are the parameters of the PP model. 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 parametric kernels with finite support (Staerman et al., 2023).

3 UNMIXING NOISE FROM HAWKES PROCESSES

Problem statement. We consider a set of observed events $\mathscr{F}_T = \{e_n = (t_n, \kappa_n),\ 1 \leq n \leq N(T)\}$. This set is a superposition of events originating from two independent processes, denoted $\mathscr{F}_{T,k} = \{e_n^k = (t_n^k, \kappa_n^k); 1 \leq n \leq N^k(T)\}$, such that $\mathscr{F}_T = \mathscr{F}_{T,0} \cup \mathscr{F}_{T,1}$. We are interested in the case where $\mathscr{F}_{T,0}$ is a homogeneous marked Poisson process –representing spurious event detections– and $\mathscr{F}_{T,1}$ is an MHP –for structured events. This problem is a denoising problem, where spurious events represent noisy events that should be discarded.

Our goal is to infer the Hawkes parameters of process $\mathscr{F}_{T,1}$. This requires first unmixing the two processes, *i.e.,* associating each event $e_n \in \mathscr{F}_T$ with a label $Y_n \in \{0, 1\}$ such that $Y_n = 1$ if e_n originates from $\mathscr{F}_{T,1}$. The Y_n are thus latent variables that should be inferred, amounting 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 MHP structure of $\mathscr{F}_{T,1}$ to characterize structured events, assigning events with this process if they are plausible according to the MHP model.

This is an arduous assignment problem, which we address using a variational inference approach and a mean-field relaxation.

UNHaP solves a classification problem conjoined with an inference task, as illustrated in Figure 1.

Latent variables and risk function. If 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 $\lambda^0(t, \kappa; \boldsymbol{\theta}_0) = \tilde{\mu} f^0(\kappa)$ such that $\tilde{\mu} \in \mathbb{R}_+$, $f^0 : \mathcal{K} \to \mathbb{R}_+$, $\int_{\mathcal{K}} f^0(\kappa)\ \mathrm{d}\kappa = 1$ and $\boldsymbol{\theta}_0 = \{\tilde{\mu}\}$. Non-spurious events follow a MHP whose intensity, denoted $\lambda_i^1(t, \kappa; \boldsymbol{\theta}_1)$, can only be derived from the observed events. We have, for $t \in [0, T]$,

$$\lambda^1(t, \kappa; \boldsymbol{\theta}_1) = \Big(\mu + \sum_{t_n < t} Y_n \phi(t - t_n; \eta)\ \omega(\kappa_n)\Big) f^1(\kappa),$$

where ϕ is a parametric kernel parameterized by η, and $\boldsymbol{\theta}_1 = \{\mu, \eta\}$. Interestingly, the intensity function depends only on past events from $\mathscr{F}_{T,1}$. To compute the intensity function, one must select the correct events.

Both processes are independent conditioned on the latent variables $\{Y_n^i\}$. Assuming $\mathcal{Y}_T = \{Y_n\}_n$ are observed, the risk for the parameters $\boldsymbol{\theta}$ is thus the sum of the least square loss, defined in (1), for each process, *i.e.*,

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

Our framework differs here from stochastic declustering (Zhuang et al., 2002), in which events from $\mathscr{F}_{T,1}$ *and* $\mathscr{F}_{T,0}$ contribute to the Hawkes process $\mathscr{F}_{T,1}$'s excitation. If $\boldsymbol{\lambda}^0$ and $\boldsymbol{\lambda}^1$ denote the true intensity functions of each underlying process, then we have $\mathbb{E}_{\mathscr{F}_T}[\mathcal{L}\left(\boldsymbol{\theta}; \mathcal{Y}_T, \mathscr{F}_T\right)] = \|\lambda^0(\boldsymbol{\theta_0}) - \boldsymbol{\lambda}^0\|_2^2 + \|\lambda^1(\boldsymbol{\theta_1}) - \boldsymbol{\lambda}^1\|_2^2 - C$ where C is a constant in $\boldsymbol{\theta}$. This loss $\mathcal{L}\left(\boldsymbol{\theta}; \mathcal{Y}_T, \mathscr{F}_T\right)$ is an empirical risk of the model for a given set of observed events and an assignment $\{Y_n\}_n$. 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 $\{Y_n\}_n$. 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. This requires computing the marginalized likelihood or at least estimating it with Monte Carlo sampling. But this step is not possible with the assignment variable Y_n due to the complex dependency structure between them imposed by the Hawkes process structure. Direct sampling of all the Y at once would not account for the temporal structure of the Y, while sampling $Y_n = 1$ for an event increases the probability of $Y_{n+k} = 1$ for all events in the kernel's support, making independent sampling of Y_n impossible. To alleviate this challenge, we resort to a mean-field approximation with independent variables for each event. Specifically, we perform the following approximation

$$p(\mathbf{Y}; \mathscr{F}_T) \approx \prod_{n=1}^{N_T} q(Y_n; \rho_n), \qquad (3)$$

where $q(Y; \rho)$ is a univariate Bernoulli distribution with parameter ρ. The parameter ρ_n is the probability that $Y_n = 1$. It corresponds to a relaxation of the assignment variable $Y_n \in \{0, 1\}$ to the interval $[0, 1]$. This relaxation allows us to compute the expected risk of the model with respect to the latent variables Y_n. Therefore, with $\boldsymbol{\rho} = \{\rho_n\}_n$, we have

$$\begin{aligned}\bar{\mathcal{L}}\left(\boldsymbol{\theta}, \boldsymbol{\rho}; \mathscr{F}_T\right) = \mathbb{E}_{\mathbf{Y}}\left[\mathcal{L}\left(\boldsymbol{\theta}; \mathcal{Y}_T, \mathscr{F}_T\right)\right] = \\ \boldsymbol{C}(\boldsymbol{\rho}) + \int_0^T\!\!\int_{\mathcal{K}} \lambda^0(t, \kappa)^2 + \bar{\lambda}^1(t, \kappa)^2 \ \mathrm{d}\kappa \mathrm{d}t \qquad (4)\\ - 2 \sum_{n, t_n \in \mathscr{F}_T} (1 - \rho_n)\lambda^0\left(t_n, \kappa_n\right) + \rho_n \bar{\lambda}^1\left(t_n, \kappa_n\right),\end{aligned}$$

where $\bar{\lambda}^1(t, \kappa; \boldsymbol{\theta}_1) = \Big(\mu + \sum_{t_n < t} \rho_n \phi(t - t_n; \eta) \omega(\kappa_n)\Big) f^1(\kappa)$ corresponds to λ_i^1 where $\boldsymbol{Y}$ has been replaced by $\boldsymbol{\rho}$ and $\boldsymbol{C}(\boldsymbol{\rho}) = \int_0^T \sum_{t_n < t} \rho_n (1 - \rho_n)\ \omega(\kappa_n)^2 \phi(t - t_n)^2 \mathrm{d}t$. Here, we can replace Y_n by its expectation ρ_n in the integral of the squared intensity as $\mathbb{E}[Y_n Y_l] = \rho_n \rho_l$ for the distribution q. However, this is not true for $\mathbb{E}[(Y_n)^2]$ which is equal to ρ_n and not $(\rho_n)^2$. $\boldsymbol{C}(\boldsymbol{\rho})$ corrects this discrepancy. Note that $\bar{\mathcal{L}}$ can also be seen as a relaxation of the assignment problem with continuous variables ρ_n.

Classification EM. Based on this mean-field approximation, we propose a variant of the classification EM algorithm (CEM; Celeux and Govaert 1992) summarized in Algorithm 1. 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 $Y_n^{(\ell)} = \mathbb{I}\{\rho_n^{(\ell)} > 1/2\}$. 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 of each event $e_n \in \mathscr{F}_T$ to one of the two processes. Minimizing Equation (4) with respect to ρ and θ is non-convex and, therefore, does not have an extensive convergence guarantee. {Moreover, as denoted in Zhang et al. (2021), the assignment problem is NP-hard in the more straightforward case of declustering. Thus, no guarantees can be obtained regarding the global convergence of the problem. However, if the CEM converges, it is guaranteed to converge to a critical point (Celeux and Govaert, 1992). In addition to this variational procedure, fast and efficient inference in UNHaP relies on several critical points described below.

Efficient parameter inference. The estimation of the parameters $\boldsymbol{\theta}^{(\ell)}$ in the **M**-step relies on the

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) $\boldsymbol{\rho}^{(\ell)} = \underset{\boldsymbol{\rho}}{\text{argmin}}\ \bar{\mathcal{L}}_{\mathcal{G}}(\boldsymbol{\rho}; \boldsymbol{\theta}^{(\ell-1)}, \mathscr{F}_T)$
    (C-step) Assign the events by computing
        $\mathcal{Y}_T^{(\ell)} = \{Y_n^{(\ell)} = \mathbb{I}\{\rho_n^{(\ell)} > 1/2\}\}_n$.
    (M-step) $\boldsymbol{\theta}^{(\ell)} = \underset{\boldsymbol{\theta}}{\text{argmin}}\ \mathcal{L}_{\mathcal{G}}(\boldsymbol{\theta}; \mathcal{Y}_T^{(\ell)}, \mathscr{F}_T)$
        with initial estimate $\boldsymbol{\theta}$ at $\boldsymbol{\theta}^{(\ell-1)}$.
  end for
output $\boldsymbol{\theta}^{(n_{\text{iter}})}, \boldsymbol{\rho}^{(n_{\text{iter}})}$.
```

FaDIn framework (Staerman et al., 2023), chosen for its computational efficiency. This framework relies on discretizing the timeline with a step size Δ. We, therefore, add an index $\mathcal{G}$ to the losses, referring to the discretization grid on the previously introduced losses (2) and (4). We refer the reader to Section A.2 for details on the discretization.

Minimization steps. The **E** and **M** steps of Algorithm 1 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 computed in the Section A.5.

Initialization with Moments Matching. As it is generally the case when inferring Hawkes processes (Lemonnier and 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 sub-optimal parameters. The quality of these minima strongly depends on the initialization scheme used the baseline and kernel parameters. Random selection can make the algorithm unstable and yield sub-optimal parameters. We propose to take advantage of the observed event distribution and initialize the parameters with moment matching, which ensures that the observed distribution's moments match the ones of the parametric model at initialization. We refer to this option as "Moments Matching initialization". Mathematical details and numerical experiments demonstrating the advantages of using Moments Matching are deferred in Section A.4 and Section B.3, respectively.

Complexity and scalability. Given a number of iterations of our solver, say n_{iter}, UNHaP's complexity is dominated by $O\big(\lfloor n_{\text{iter}}/b \rfloor\, L^2 T\big)$, where $L = \lfloor W/\Delta \rfloor$ is the number of elements of the time grid $\mathcal{G}$ used for the kernel discretization. On the contrary, the usual methods' complexity is dominated by $\mathcal{O}\big(T^2\big)$. As a consequence, UNHaP improves scalability for long time series. See Supplementary Section A.3 for more details.

4 NUMERICAL VALIDATION

In this section, we evaluate how UNHaP recovers the mixture's latent variables and parameters on simulated data, and compare its robustness and computational cost with existing PP solvers.

4.1 JOINT INFERENCE AND UNMIXING WITH UNHAP

Simulation. We generate a MHP observation 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

$$\lambda(t, \kappa; \boldsymbol{\theta}) = \Bigg(\mu + \alpha \sum_{t_n < t} Y_n \omega(\kappa_n) \phi(t - t_n; \eta)\Bigg) f^1(\kappa) + \tilde{\mu}\ f^0(\kappa), \tag{5}$$

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. 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 MHP stemming from previous ones. In contrast, $\boldsymbol{\alpha} \to 0$ 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 $f^0(\kappa) = 2(1 - \kappa)$ 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 an actual event. The excitation kernel $\phi(\cdot; \eta)$ is chosen as a truncated Gaussian kernel (Staerman et al., 2023; Allain et al., 2022; Hodara et al., 2018) to model delays between the events. With $\eta = (m, \sigma)$, it reads

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

where W is the kernel length and γ (resp. F) is the standard normal distribution's probability density

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

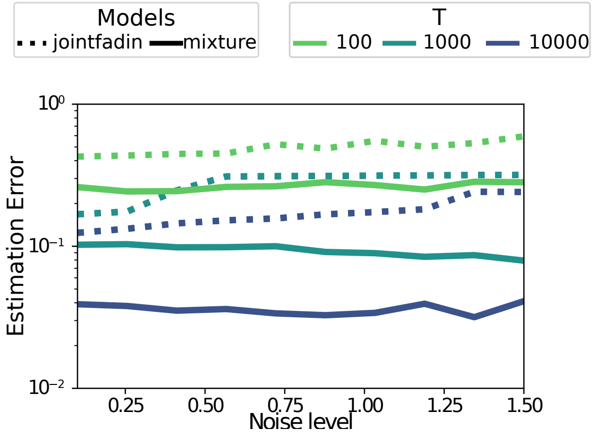

Figure 2: Parameters estimation errors for UNHaP and "jointfadin" for varying T w.r.t. different values of $\tilde{\mu}$ with linear distributions on noisy marks.

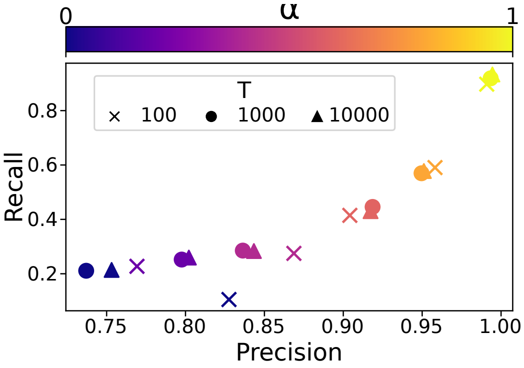

Figure 3: Precision/Recall values for the estimation of $\boldsymbol{\rho}$ for different values of T w.r.t. α with uniform distributions on noisy marks and $\tilde{\mu} = 1$.

function (resp. cumulative distribution function). We set $\eta = (0.5, 0.1)$ in our experiments.

Robust parameter inference with UNHaP. First, we compare UNHaP's 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 MHP's parameters $\boldsymbol{\theta} = \{\mu, \alpha, m, \sigma\}$ with UNHaP and compare our results with an extension of FaDIn which can handle marks, 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 $\hat{\rho}$ updates 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 setting. UNHaP outperforms JointFaDIn for all noise levels $\tilde{\mu}$, as it discards the part of the noisy events. We can also see it is robust as the performances remain constant with varying noise. Similar results for uniform marks are shown in Figure B.2, and Section B.3.2 discuss UNHaP's performance with non-Poisson noise.

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 $\tilde{\mu} = 1$. Experiments varying the noise levels ($\tilde{\mu} = 0.5$ and $\tilde{\mu} = 1$) are presented in Figure B.3. We consider the mixture parameter $\boldsymbol{\rho}$ inferred with UNHaP with uninformative (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 $\hat{\boldsymbol{\rho}}$ w.r.t. the ground truth. We can see the convergence $\hat{\boldsymbol{\rho}}$ towards the true $\boldsymbol{\rho}^*$ when the excitation structure grows stronger, reaching good accuracy levels. Figure B.3 shows results for varying mark distribution and noise level $\tilde{\mu}$, showing that when mark densities are informative, the accuracy stays high even with small α. When α is small and the excitation structure absent, only the mark distribution may distinguish between the events stemming from μ and $\tilde{\mu}$.

4.2 INFERENCE QUALITY AND COMPUTATION TIME

We compare UNHaP with various Hawkes process solvers. We assess approaches' statistical and computational efficiency in the case of simulated marked and unmarked data, see Table 4.1 and Table B.3. FaDIn is an unmarked Hawkes process solver without classification (Staerman et al., 2023), and MarkedFaDIn is an extension of FaDIn that accounts for the marks. StocUNHaP is a variant of UNHaP where instead of using CEM, each latent variable Y_n is assigned using a Bernoulli variable of parameter ρ_n, in the same fashion as Zhang et al. (2021). We also compare with other solvers that do not account for the noise in the observed events. In TPPSelect (Zhang et al., 2021), like in stochastic declustering (Zhuang et al., 2002), exogenous events are triggered by the Hawkes process self-excitation instead of a separated Poisson process and impact the downstream intensity. VB (Linderman and Adams, 2015) is a stochastic variational inference algorithm that leverages a Bayesian formulation of the problem. In Tripp (Shchur et al., 2020), a triangular map is used to approximate the integral of the unmarked intensity function. In NeuralHawkes (Mei and Eisner, 2017), the authors model the unmarked intensity function with an LSTM module.

The experiment goes as follows. We simulate a marked Hawkes process in a high noise setting. Its intensity function is defined as in (5), with a linear mark distribution in [0, 1]. We also simulate a Poisson Process

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.

	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
StocUNHaP	0.726 ± 0.407	0.504 ± 0.207	**0.344 ± 0.121**	95.5 ± 5.1	112 ± 4.7	120 ± 4.2
MarkedFaDIn	1.36 ± 0.22	1.18 ± 0.06	1.16 ± 0.04	17 ± 8.5	16.5 ± 8.2	15.5 ± 8.2
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
TPPSelect	1.196 ± 0.149	1.114 ± 0.120	1.104 ± 0.092	240 ± 240	2079 ± 549	4743 ± 704
VB	0.920 ± 0.183	0.980 ± 0.098	0.943 ± 0.031	6.5 ± 2.3	7.8 ± 2.5	9.3 ± 2.4
NeuralHawkes	2.006 ± 0.7	1.574 ± 0.45	1.141 ± 0.2	43.4 ± 16.8	171.8 ± 38.1	183.3 ± 30.7
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

for the noisy events, with a uniform mark distribution in [0, 0.2]. We recall that the mark associated with an event quantifies the confidence in the event belonging to the Hawkes process. The simulated processes' mark distributions are coherent with this assumption: the Hawkes process events have larger marks than the Poisson process events. 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 infer the parameters of the underlying Hawkes process's intensity function. This experimental procedure is replicated for various values of $T \in \{10, 500, 1000\}$. For all discretized approaches, we set $\Delta = 0.01$ and $W = 1$. The Negative Log-Likelihood (NLL) is computed on a set of left-out events simulated with the Hawkes process and parameters identical to the training data.

The median NLL over ten runs and the computation time are displayed in Table 4.1. In this marked and noisy setting, UNHaP outperforms other methods for parameter inference. This aligns with the expectations as UNHaP is the only method designed for noisy observation. The comparison with FaDIn and MarkedFaDIn demonstrates that accounting for the marks and the noise leads to big improvements in the intensity function inference. StocUNHaP obtains slightly worse results in short-time experiments, showing the benefit of using CEM over soft assignments. Interestingly, VB obtains reasonable results much faster, while TPPSelect does not improve inference quality over MarkedFadIn while being much slower. The deep-learning-based model NeuralHawkes is designed to process many short event sequences, explaining its subpar performance for a single short sequence and its long runtime for longer sequences. The increased runtime of UNHaP compared to FaDIn and MarkedFaDIn comes from the alternate minimization structure that requires running multiple FaDIn inferences when running UnHaP. In an unmarked setting, UNHaP performs on par with the unmarked methods, with a slight advantage in noisy settings; see Table B.3. We also compared the methods' error on the classification and Hawkes parameters inference. Table B.1 shows that UNHaP outperforms other methods on both tasks.

5 APPLICATION TO PHYSIOLOGICAL DATA

To demonstrate the usefulness of UNHaP in real-world physiological event detection cases, we use it to characterize the inter-event interval distribution in ECG and gait data. Figure 4 and Figure B.5 provide examples of these data modalities. 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 and Ginsberg, 2017). Similarly, studying 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 and Tompkins, 1985; Elgendi, 2013; Hamilton, 2002), in combination with heavily tailored post-processing steps (Merdjanovska and 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.

Convolutional Dictionary Learning (CDL; Grosse et al. 2007) is a general and unsupervised approach to detecting events. 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* dataset (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.

Experimental pipeline for CDL+UNHaP method. We used the same method for ECG and gait recordings. In what follows, it is detailed on ECG recordings. The proposed method relies on the CDL implementation from the Python library `alphacsc` (Dupré la 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. Each beat has non-zero activations, 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 (Allain et al., 2022; Staerman et al., 2023). 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; C1). The mean (respectively the standard deviation) of the parameterized truncated Gaussian ϕ estimates the mean inter-beat interval (respectively the inter-beat 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. Additional experimental details are provided in Section B.4.

Results. We compare UNHaP with several inter-beat and inter-step interval estimators to post-process the detected events. We included FaDIn (Staerman et al., 2023), an unmarked Hawkes process solver without classification, and StocUNHaP, a variant of UNHaP where the classification step of the EM algorithm is done stochastically instead of using a hard threshold. We also compare UNHaP to several domain specific libraries: `pyHRV` (Gomes, 2024), `Neurokit` (Makowski et al., 2021), for ECG, and Template Matching (Oudre et al., 2018), for gait detection.

For ECG, the mean inter-beat interval obtained with these estimators is compared to the ground truth given in the dataset. We average each estimator's absolute errors over the 19 ECG recordings and report the results in Table 2. UNHaP, StocUNHaP, pyHRV, and Neurokit have equivalent performance on the ECG data, all providing good heart rate estimates. Note that while working with the same detected events, the noise filtering performed by UNHaP largely 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 comparably to the state-of-the-art Template Matching on gait data. On the contrary, methods designed for ECG (pyHRV and Neurokit) fail to accurately estimate the inter-step interval. Interestingly, UNHaP is the only method that performs on par with state-of-the-art on both data modalities, illustrating its universality with no domain-specific tuning necessary.

Table 2: Error of Various Models on Physiological Data. Median (Q1−Q3) of the Absolute Error (AE) on the heart rate in beats per minute (ECG) and inter-step interval in seconds (Gait).

Model	**ECG**	**Gait**
CDL + UNHaP	0.27 (0.14–0.84)	0.04 (0.02–0.07)
CDL + StocUNHaP	0.25 (0.06–0.90)	NA
CDL + FaDIn	2.57 (0.26–40.4)	1.2 (1.1–1.3)
pyHRV	0.81 (0.16–2.08)	0.7 (0.3–1.3)
Neurokit	0.54 (0.51–0.61)	0.6 (0.5–0.7)
Template Matching	NA	0.07

6 DISCUSSION

Real-world applications for Hawkes processes seldom have access to clean event streams. They are usually contaminated with noisy events. With this paper, we highlight and solve this challenge. We derive a mathematical framework that includes the observation mixture structure and show that it improves the parameter estimation on both simulated and real-world data. The UNHaP method we propose in this work efficiently tackles the classification and inference tasks using a combination of latent variables, ERM-inspired least squares loss, and finite-support kernels. Unlike existing work, UNHaP accommodates using any parametric form of triggering kernels, enabling a wide range of practical applications. The pipeline developed here is unsupervised and requires no pre-processing or data adjustment. It is agnostic and meant to be robust on a wide range of data modalities.

UNHaP is specifically designed to work on marked data with noisy observations. While it can work in

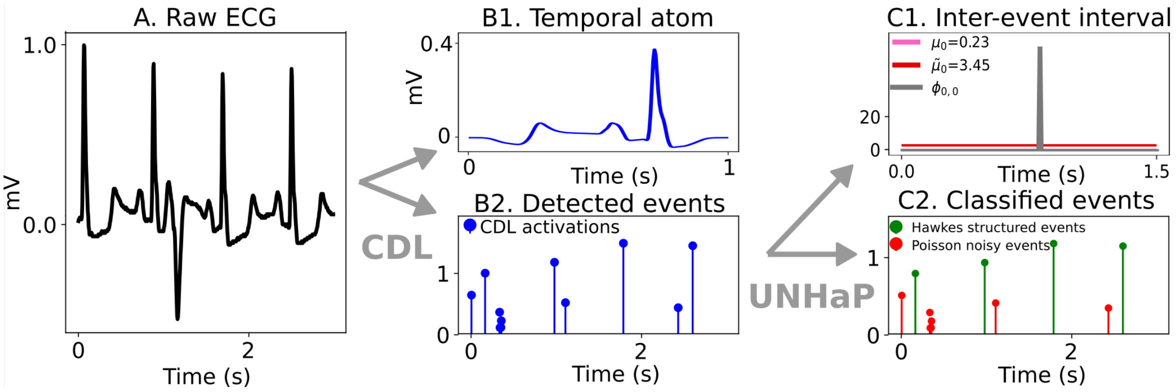

Figure 4: Experimental pipeline on ECG Data. **(A)** Sub-sample of raw ECG plot. **(B)** Output of Convolutional Dictionary Learning algorithm: **(B1)** learned temporal atom representing one heartbeat, **(B2)** detected events on the time interval. **(C)** Output of UNHaP: **(C1)** Estimated Hawkes parameters: noise baseline (red), baseline (pink), and kernel (grey). 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**).

the non-noisy, non-marked typical TPP experimental framework, exemplified by the EasyTPP benchmark (Xue et al., 2023), it is not expected to outperform standard TPP methods in this setting. By parameterizing a Hawkes kernel, UNHaP allows for estimating statistics on the inter-event time interval. This makes real-world applications detailed in Section 5 possible. However, in cases where only an intensity function is needed on large, unnoisy datasets, classical TPP methods could be preferred.

ACKNOWLEDGEMENTS

This work was supported by the ANR EBUL (ANR-23-CE23-0001). We thank the reviewers for their helpful remarks that have improved our work. We also thank our colleagues Jade Perdereau, Jérôme Cartailler and Miguel Martinez Herrera for their remarks on our work.

References

Selcan Kaplan Berkaya, Alper Kursat Uysal, Efnan Sora Gunal, Semih Ergin, Serkan Gunal, and M Bilginer Gulmezoglu. A survey on ecg analysis. *Biomedical Signal Processing and Control*, 43:216–235, 2018.

Luz Luz, Eduardo José da S, William Robson Schwartz, Guillermo Cámara-Chávez, and David Menotti. ECG-based heartbeat classification for arrhythmia detection: A survey. *Computer Methods and Programs in Biomedicine*, 127:144–164, 2016.

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

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

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

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

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

Jonathan M Lilly. Element analysis: A wavelet-based method for analysing time-localized events in noisy time series. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2200):20160776, 2017.

Roger Grosse, Rajat Raina, Helen Kwong, and Andrew Y. Ng. Shift-Invariant Sparse Coding for Audio Classification. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 23, pages 149–158, 2007.

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

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

Guillaume Staerman, Cédric Allain, Alexandre Gramfort, and Thomas Moreau. Fadin: Fast discretized inference for hawkes processes with general parametric kernels. In *International Conference on Machine Learning*, pages 32575–32597. PMLR, 2023.

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

Shengzhong Liu, Shuochao Yao, Dongxin Liu, Huajie Shao, Yiran Zhao, Xinzhe Fu, and Tarek Abdelzaher. 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)*, pages 1275–1285. IEEE, 2019.

Shuang-Hong Yang and Hongyuan Zha. Mixture of mutually exciting processes for viral diffusion. In *International Conference on Machine Learning*, pages 1–9. PMLR, 2013.

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

Jiancang Zhuang, Yosihiko Ogata, and David Vere-Jones. Stochastic declustering of space-time earthquake occurrences. *Journal of the American Statistical Association*, 97(458):369–380, 2002.

Scott Linderman and Ryan Adams. Discovering latent network structure in point process data. In *International conference on machine learning*, pages 1413–1421. PMLR, 2014.

Ping Zhang, Rishabh Iyer, Ashish Tendulkar, Gaurav Aggarwal, and Abir De. Learning to select exogenous events for marked temporal point process. *Advances in Neural Information Processing Systems*, 34:347–361, 2021.

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

Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J Smola, and Le Song. 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*, pages 219–228, 2015.

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

Anna Bonnet, Felix Cheysson, Miguel Martinez Herrera, and Maxime Sangnier. Spectral analysis for noisy hawkes processes inference. *arXiv preprint arXiv:2405.12581*, 2024.

Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(01):1550005, 2015.

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

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

Remi Lemonnier and Nicolas Vayatis. 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*, pages 161–176. Springer, 2014.

Pierre Hodara, Nathalie Krell, and Eva Löcherbach. Non-parametric estimation of the spiking rate in systems of interacting neurons. *Statistical Inference for Stochastic Processes*, 21:81–111, 2018.

Scott W Linderman and Ryan P Adams. Scalable bayesian inference for excitatory point process networks. *arXiv preprint arXiv:1507.03228*, 2015.

Oleksandr Shchur, Nicholas Gao, Marin Biloš, and Stephan Günnemann. Fast and flexible temporal point processes with triangular maps. In *Advances in Neural Information Processing Systems*, volume 33, pages 73–84. Curran Associates, Inc., 2020.

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

Fred Shaffer and Jay P Ginsberg. An overview of heart rate variability metrics and norms. *Frontiers in public health*, page 258, 2017.

Charles Truong, Rémi Barrois-Müller, Thomas Moreau, Clément Provost, Aliénor Vienne-Jumeau, Albane Moreau, Pierre-Paul Vidal, Nicolas Vayatis, Stéphane Buffat, Alain Yelnik, et al. A data set for the study of human locomotion with inertial measurements units. *Image Processing On Line*, 9:381–390, 2019.

Laurent Oudre, Rémi Barrois-Müller, Thomas Moreau, Charles Truong, Aliénor Vienne-Jumeau, Damien Ricard, Nicolas Vayatis, and Pierre-Paul

Vidal. Template-based step detection with inertial measurement units. *Sensors*, 18(11):4033, 2018.

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

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

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

Hyung-Chul Lee, Yoonsang Park, Soo Bin Yoon, Seong Mi Yang, Dongnyeok Park, and Chul-Woo Jung. Vitaldb, a high-fidelity multi-parameter vital signs database in surgical patients. *Scientific Data*, 9(1):279, 2022.

Tom Dupré la Tour, Thomas Moreau, Mainak Jas, and Alexandre Gramfort. Multivariate convolutional sparse coding for electromagnetic brain signals, 2018.

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

Dominique Makowski, Tam Pham, Zen J. Lau, Jan C. Brammer, François Lespinasse, Hung Pham, Christopher Schölzel, and S. H. Annabel Chen. NeuroKit2: A Python toolbox for neurophysiological signal processing, February 2021. URL https://github.com/neuropsychology/NeuroKit. original-date: 2019-10-29T05:39:37Z.

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.

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

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

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

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

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

CHECKLIST

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A APPENDIX

A.1 UNHAP IN THE MULTIVARIATE CASE

In the appendix, we consider an explicit multivariate version of the proposed framework, where we have explicit event types, denoted with index $i \in \{1, \dots, D\}$. It is equivalent to having $\mathcal{K} = \widetilde{\mathcal{K}} \times \{1, \dots, D\}$ and separating the dependencies on each domain. The univariate setting from the manuscript's main body can be retrieved by considering $D = 1$ and dropping the index i in the sums. We expose below UNHaP's framework in the multivariate case.

A.1.1 BACKGROUND ON MULTIVARIATE 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, Sec. 6) 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 $\mathscr{F}_T^i = \{(t_n^i, \kappa_n^i) : \; \kappa_n^i \in \mathcal{K},\; t_n^i \in [0, T]\}$ 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 $N_i(T) = \int_{[0,T]\times\mathbb{R}_+} \mathbf{N}_i(\mathrm{d}t, \mathrm{d}\kappa) = \sum\limits_{n\geq 1} \mathbf{1}_{t_n^i \leq T}$.

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 instantaneous rate of events occurring at time t, taking into account the marks of previous events occurring before t. 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

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

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 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 Empirical Risk Minimization (ERM)-inspired least squares loss (refer to Eq. (II.4) in (Bompaire, 2019, Chapter 2)). The goal is to minimize

$$\mathcal{L}\left(\boldsymbol{\theta}, \mathscr{F}_T\right) = \sum_{i=1}^{D} \left(\int_0^T\!\!\int_{\mathcal{K}} \lambda_i(s, \kappa; \boldsymbol{\theta})^2 \;\mathrm{d}\kappa\mathrm{d}s - 2 \sum_{(t_n^i, \kappa_n^i)\in\mathscr{F}_T^i} \lambda_i\left(t_n^i, \kappa_n^i; \boldsymbol{\theta}\right) \right), \tag{6}$$

where $\boldsymbol{\theta} = \{\mu_i, \phi_{ij}, \omega_{ij}\}_{i=1}^D$ and $\mathscr{F}_T^i = \left\{(t_n^i, \kappa_n^i) : \ \kappa_n^i \in \mathcal{K}, \ t_n^i \in [0,T]\right\}$ are the ground truth events of event type i. 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).

A.1.2 UNMIXING NOISE FROM HAWKES PROCESSES

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 $\mathscr{F}_{T,0}$.

If 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 $\boldsymbol{\theta}_1 = \{\mu_i, \eta_{ij}\}_{i,j=1}^D$. 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

$$\begin{aligned}\mathcal{L}^i\left(\boldsymbol{\theta}; \mathcal{Y}_T, \mathscr{F}_T\right) = &\int_0^T\!\!\int_{\mathcal{K}} \lambda_i^0(t, \kappa; \boldsymbol{\theta}_0)^2 \ \mathrm{d}\kappa \mathrm{d}t + \int_0^T\!\!\int_{\mathcal{K}} \lambda_i^1(t, \kappa; \boldsymbol{\theta}_1)^2 \ \mathrm{d}\kappa \mathrm{d}t \\ &- 2 \sum_{e_n^i \in \mathscr{F}_T^i} (1 - Y_n^i) \lambda_i^0(t_n^i, \kappa_n^i; \boldsymbol{\theta}_0) + Y_n^i \lambda_i^1(t_n^i, \kappa_n^i; \boldsymbol{\theta}_1).\end{aligned} \tag{7}$$

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

Algorithm 2 UNHaP solver in the multivariate case.

```
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) $\boldsymbol{\rho}^{(\ell)} = \underset{\boldsymbol{\rho}}{\operatorname{argmin}} \sum_{i=1}^{D} \bar{\mathcal{L}}_{\mathcal{G}}^{i}(\boldsymbol{\rho}; \boldsymbol{\theta}^{(\ell-1)}, \mathscr{F}_T)$
    (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) $\boldsymbol{\theta}^{(\ell)} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathcal{L}_{\mathcal{G}}(\boldsymbol{\theta}; \mathcal{Y}_T^{(\ell)}, \mathscr{F}_T)$ initialized $\boldsymbol{\theta}$ at $\boldsymbol{\theta}^{(\ell-1)}$.
  end for
output  $\boldsymbol{\theta}^{(n_{\text{iter}})}, \boldsymbol{\rho}^{(n_{\text{iter}})}$.
```

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 event. Specifically, we perform the following approximation

$$p(\mathbf{Y}; \mathscr{F}_T) = \prod_{i=1}^{D} p(Y^i; \mathscr{F}_T^i) \approx \prod_{i=1}^{D} \prod_{n=1}^{N_T^i} q(Y_n^i; \rho_n^i), \tag{8}$$

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

$$\begin{aligned}\bar{\mathcal{L}}^i\left(\boldsymbol{\theta}, \boldsymbol{\rho}; \mathscr{F}_T\right) = &\int_0^T\!\!\int_{\mathcal{K}} \lambda_i^0(t, \kappa)^2 \,\mathrm{d}\kappa \mathrm{d}t + \int_0^T\!\!\int_{\mathcal{K}} \bar{\lambda}_i^1(t, \kappa)^2 \,\mathrm{d}\kappa \mathrm{d}t + \boldsymbol{C}(\boldsymbol{\rho}) \\ &- 2 \sum_{n, t_n^i \in \mathscr{F}_T^i} \left((1 - \rho_n^i) \lambda_i^0\left(t_n^i, \kappa_n^i\right) + \rho_n^i \bar{\lambda}^1\left(t_n^i, \kappa_n^i\right) \right),\end{aligned} \tag{9}$$

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 $\mathbb{E}[Y_n^i Y_l^i] = \rho_n^i \rho_l^i$ 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.

Based on this mean-field approximation, we propose a variant of the classification EM algorithm (CEM; (Celeux and Govaert, 1992)) summarized in Algorithm 1. 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 (7) and (9). For details on adapting this framework to our unmixing problem, we refer the reader to Section A.2.

Minimization steps. The **E** and **M** steps of Algorithm 1 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.5. 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.

A.2 DETAILS OF UNHAP LOSS WITH DISCRETIZATION

In the following, we assume that the functions $\omega_{ij}(\cdot)$ are identical for $1 \le i, j \le D$ 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 $\tilde{z}_j[s] = \sum_{t_n^j} \rho_n^j \; \omega(\kappa_n^j) \; \mathbf{1}_{\left\{|t_n^j - s\Delta| \le \frac{\Delta}{2}\right\}}$. 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 and 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

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

where $H^\ell_i = \int_{\mathcal{K}} (f^\ell_i(\kappa))^2 \ \mathrm{d}\kappa$ for $\ell \in \{0, 1\}$ and $\widetilde{\Phi}_j(\tau; G) = \sum_{s=1}^{G} \tilde{z}_j[s - \tau]$, $\widetilde{\Psi}_{jk}(\tau, \tau'; G) = \sum_{s=1}^{G} \tilde{z}_j[s - \tau] \tilde{z}_k[s - \tau']$, $\widetilde{\Xi}_j(\tau; G) = \sum_{s=1}^{G} \left(z^2_j[s - \tau] \rho^j[s - \tau] - \tilde{z}^2_j[s - \tau] \right)$ and $\widetilde{\Phi}_j(\tau; \widetilde{\mathscr{F}}^i_T) = \sum_{(\tilde{t}^i_n, \kappa^i_n) \in \widetilde{\mathscr{F}}^i_T} f^1_i(\kappa^i_n) \rho^i \left[\frac{\tilde{t}^i_n}{\Delta}\right] \tilde{z}_j \left[\frac{\tilde{t}^i_n}{\Delta} - \tau\right]$. 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^i_n by $Y^i_n = \mathbb{I}\{\rho^i_n > 1/2\}$ and removing the fourth term.

A.3 UNHAP'S COMPLEXITY AND SCALABILITY COMPARED TO OTHER APPROACHES

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 at every maximization step. As the complexity of FaDIn is $\mathcal{O}(\max(n_{\text{iter}} D^3 L^2, D^3 L^2 G))$, with $G \approx TL$, UNHaP's complexity is $\mathcal{O}(\frac{n_{\text{iter}}}{b} D^3 L^3 T)$, where D is the number of event types, G is the number of elements of the time grid $\mathcal{G}$, $L = \lfloor W/\Delta \rfloor$ is the number of elements of the grid used for the kernel discretization, and T is the duration of the signal.

In contrast, classical continuous parametric approaches can be computed in $O(n_{\text{iter}} D N^2)$ for general kernels. The important distinction here is the quadratic dependency in the number of events N (or T), which is often critical for scalability, while FaDIn and thus UNHaP only depend linearly on T due to the finite support assumption.

Finally, for TPPSelect (Zhang et al., 2021), the critical part is the greedy assignation procedure, which requires fitting multiple models for each event associated with the baseline. The overall complexity is thus $\mathcal{O}(N_{ev} C)$, where N_{ev} is the number of noisy events and C the complexity of the underlying solver (that we consider as FaDIn). For $N_{ev} = \alpha N$, Tripp's overall complexity is also in N^2 and scales poorly.

To conclude, usual methods methods have a quadratic complexity in the number of events or duration of the signal. In contrast, UNHaP has a linear complexity in the number of events and is, therefore, more scalable than usual methods.

A.4 INITIALIZATION WITH MOMENT 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 $N_{g_1}, \ldots, N_{g_D}$ 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.*

$$
N_{g_i}(T) = \mathbb{E}[N_{g_i}(T)] = \int_0^T \lambda_{g_i}(t) \ \mathrm{d}t. \tag{10}
$$

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^i_n are equal to $\frac{1}{2}$. With this, we get $N^0_{g_i}(T) = \frac{N_{g_i}(T)}{2}$ and thus we can compute a moment matching value $\tilde{\mu}^{\mathrm{m}}_i$ since

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

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

$$\frac{N_{g_i}(T)}{2} = \mu_i T + \sum_{j=1}^{D} \alpha^{\mathrm{m}}_{i,j} \sum_{(\tilde{t}^j_n, \kappa^j_n) \in \widetilde{\mathscr{F}}^j_T} \omega(\kappa^j_n).$$

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

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

and

$$\alpha^{\mathrm{m}}_{i,j} = \frac{N_{g_i}(T)}{2T(D+1)\sum_{(\tilde{t}^j_n, \kappa^j_n) \in \widetilde{\mathscr{F}}^j_T} \omega(\kappa^j_n)}.$$

Replacing these values for $\tilde{\mu}^m_i, \mu^m_i$, and $\alpha^m_{i,j}$ into (10) 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 $\delta t^{i,j}_n$ the delay between t^i_n and the time of occurrence of the last event in channel j before t^i_n

$$\delta t^{i,j}_n = t^i_n - \max\{t | t \in \mathscr{F}^j_T, W < t < t^i_n\}. \tag{11}$$

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

$$m^{\mathrm{m}}_{i,j} = \frac{1}{N_{g_i}(T)} \sum_{t^i_n \in \mathscr{F}^i_T} \delta t^{i,j}_n,$$
$$\sigma^{\mathrm{m}}_{i,j} = \sqrt{\frac{\sum_{t^i_n \in \mathscr{F}^i_T} (\delta t^{i,j}_n - m^{\mathrm{m}}_{i,j})^2}{N_{g_i}(T) - 1}}.$$

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

$$u^{\mathrm{m}}_{i,j} = \max(0, m^{\mathrm{m}}_{i,j} - \sigma^{\mathrm{m}}_{i,j}),$$
$$s^{\mathrm{m}}_{i,j} = \sigma^{\mathrm{m}}_{i,j}.$$

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 B.4.

For very noisy settings, where noisy events are very close to Hawkes process events in time, using the $\delta t^{i,j}_n$ defined in (11) leads to poor performance of UNHaP. This is because $\delta t^{i,j}_n$ 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 $\delta t^{i,j}_n$ with a mean instead of a maximum.

$$\delta t^{i,j}_n = t^i_n - \frac{1}{\#\{t \in \mathscr{F}^j_T, W < t < t^i_n\}} \sum_{t \in \mathscr{F}^j_T, W < t < t^i_n} t. \tag{12}$$

A.5 GRADIENTS OF THE UNHAP LOSS

This part presents 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

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

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 (5), 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 B.1 (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 B.1 (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 B.1 (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.

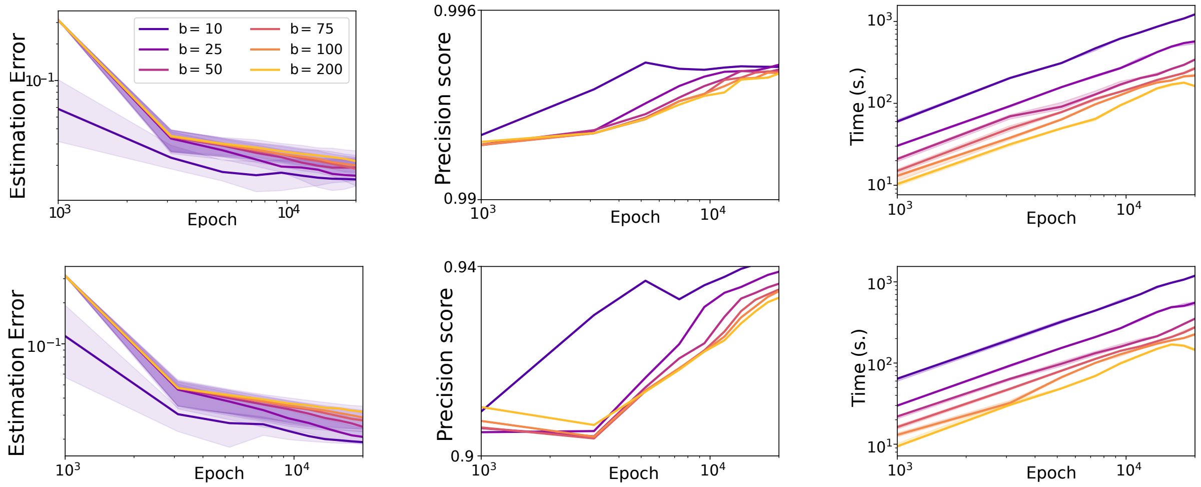

Figure B.1: Inference comparison regarding the batch size of Hawkes parameters gradients between each ρ update. The error estimation on Hawkes parameters (left), the Precision score on the ρ recovering (middle) and the associated computational time (right) are displayed for non-noisy (top) and noisy settings (bottom).

B.2 FURTHER EXPERIMENTS ON THE RECOVERY OF THE NOISE STRUCTURE

Figure B.2 displays the same experiment as in Section 4.1 but with different distribution on the marks: linear (*left*) and uniform (*right*). Figure B.3 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.4 over the classical random ones. The simulation study is conducted as follows. Relying on an Immigration-Birth algorithm (Møller and 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

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

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

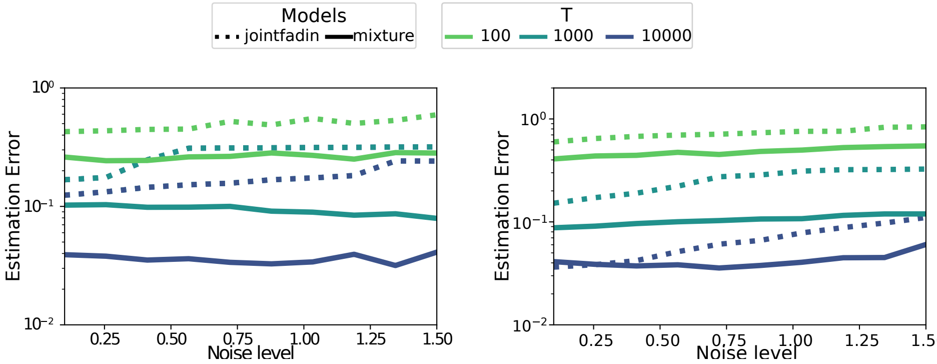

Figure B.2: Parameters estimation errors for UNHaP and "jointfadin" for varying T w.r.t. different values of $\tilde{\mu}$ with linear (*left*) and uniform (*right*) distributions on noisy marks.

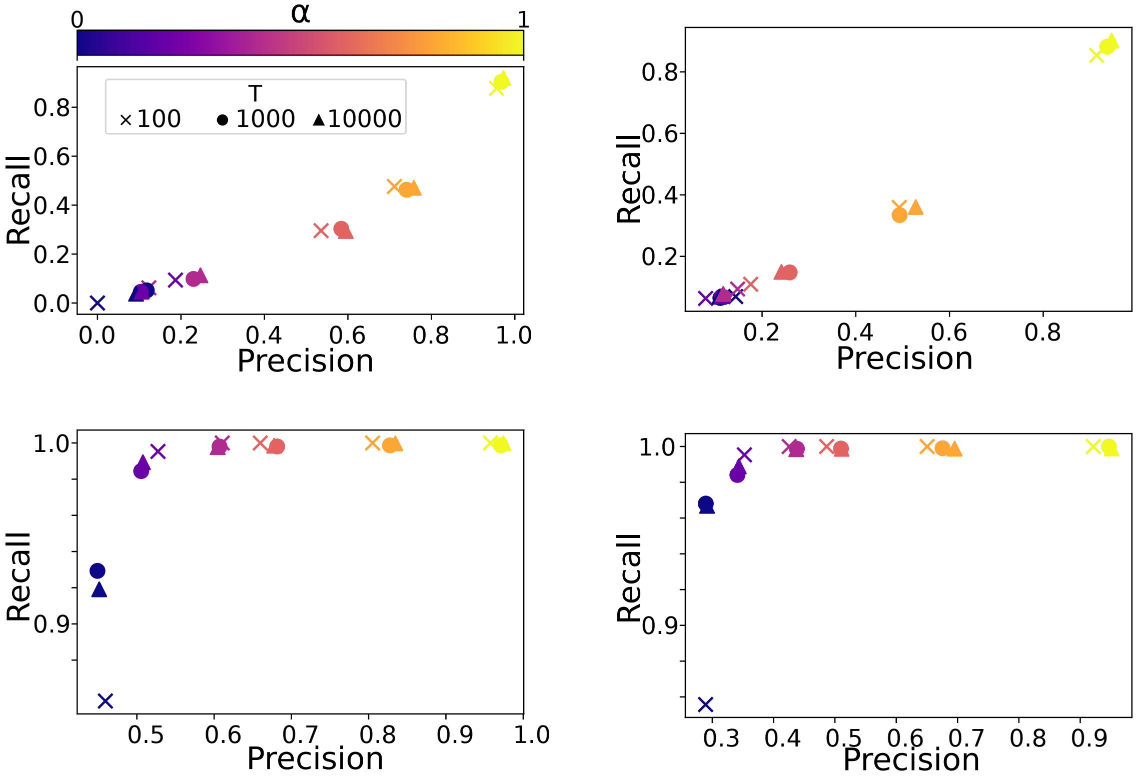

Figure B.3: Precision/Recall values for the estimation of $\boldsymbol{\rho}$ for different values of T w.r.t. the auto-excitation parameter α with uniform (top) and linear (bottom) distributions on noisy marks for $\tilde{\mu} = 0.5$ (left) and $\tilde{\mu} = 1.$ (right).

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

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

with $\eta = (u, \sigma)$. In contrast to the truncated Gaussian, the support of this kernel directly depends on its

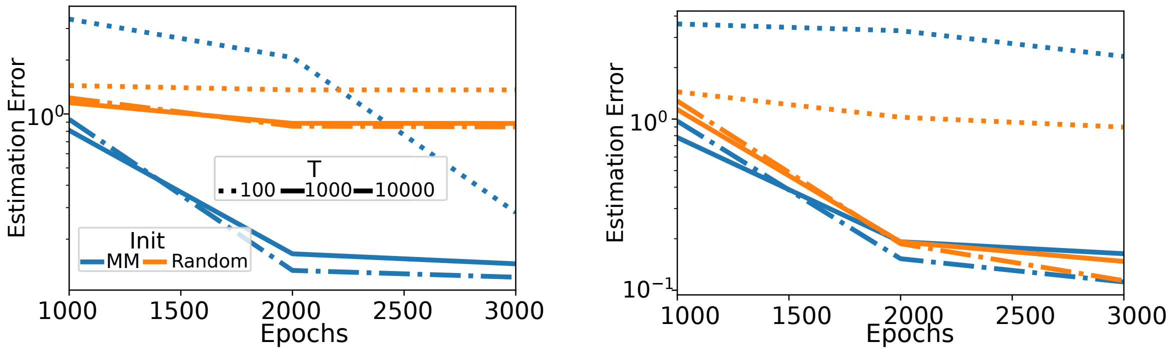

Figure B.4: Hawkes parameters estimation error using UNHaP with a raised cosine (left) and a truncated Gaussian (right) kernels, Moments Matching (blue), and Random (orange) initializations for varying size sequences.

Table B.1: Hawkes Parameter Errors of UNHaP and TPPSelect on Simulated Data

Model	NLL	μ error	α error	η error	Classification accuracy
TPPSelect	0.95	0.84	0.92	0.15	0.81
UnHaP	**0.45**	**0.06**	**0.04**	**0.09**	**0.89**

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 ERROR OF UNHAP AND TPPSELECT ON SIMULATED DATA

Some methods inferred the Hawkes process parameters and/or classified events. We compare the errors made by UNHaP and these methods on these two tasks.

We simulated data using the same pipeline as in Section 4.2, with $\tilde{\mu} = 1$, $\mu = 0.1$, $\alpha = 1$ and $\eta = (0.5, 0.1)$. After fitting UNHaP and TPPSelect to the data, we estimated their inference error on the Hawkes parameters. The results, displayed in Table B.1, show UNHaP's clear superiority on Hawkes parameters inference.

B.3.2 UNHAP AND TPPSELECT PERFORMANCE WITH POISSON DISK NOISE

Poisson noise arose from our application to remove spurious event detections that are unstructured and independent. This is a natural choice in the literature (Bonnet et al., 2024) and leads to a simple and useful algorithm. However, spurious events can follow more complex distributions. To evaluate empirically the impact of the noise model on the current algorithm, we evaluate UNHaP performances for a process where the noise is generated with a Disk Poisson noise. We compare the parameter estimation with UnHaP and TPPSelect (Zhang et al., 2021).

Table B.2: Errors Made by UNHaP and TPPSelect on Mixed Data with Disk Poisson Noise.

Model	NLL	μ error	α error	η error	Accuracy
TPPSelect	0.592	0.235	0.546	0.634	0
UNHaP	**0.387**	**0.066**	**0.517**	**0.007**	**0.420**

We can see that UnHaP loses some accuracy but still manages to accurately estimate most parameters, except the α parameter strongly impacted by the accuracy. In contrast, TPPSelect fails to distinguish between the two processes, resulting in more significant parameter estimation errors. This suggests that UNHaP could perform well in a more general setting than a Poisson process for noisy events.

B.3.3 BENCHMARK OF 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.

Table B.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.

	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

The events are simulated similarly to in Section 4.2. We simulate a Marked Hawkes process. Its intensity function is defined as in (5). 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 B.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 and 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 and Tompkins, 1985; Elgendi, 2013; Hamilton, 2002). However, raw ECG signals usually

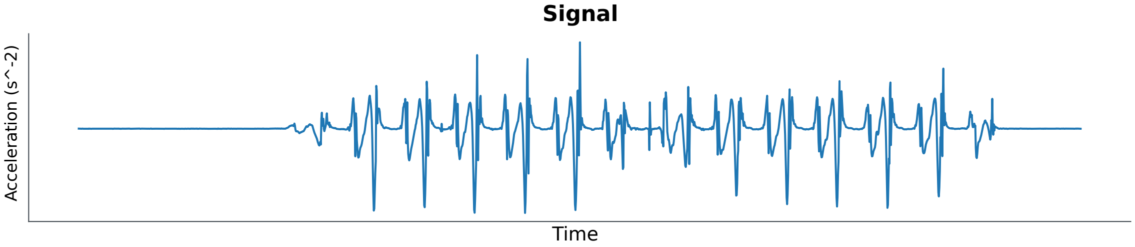

Figure B.5: Example of gait inertial measurement unit data.

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 and 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 MHP 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` (Dupré la 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.

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.

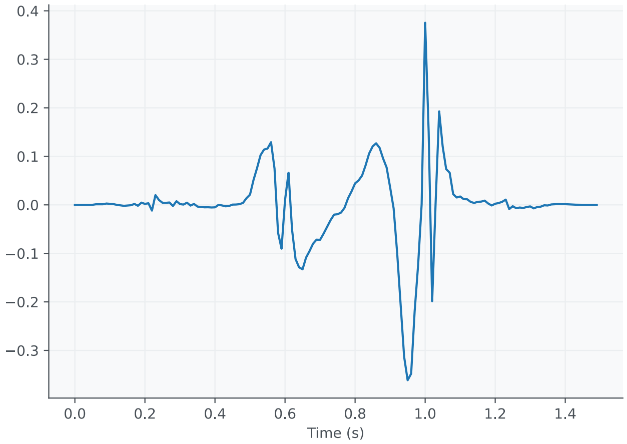

Figure B.6: Example of gait atom extracted using CDL.

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` (Dupré la 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.