FaDIn: Fast Discretized Inference for Hawkes Processes with General Parametric Kernels

Guillaume Staerman [*1] **Cédric Allain** [*1] **Alexandre Gramfort** [1] **Thomas Moreau** [1]

Abstract

Temporal point processes (TPP) are a natural tool for modeling event-based data. Among all TPP models, Hawkes processes have proven to be the most widely used, mainly due to their adequate modeling for various applications, particularly when considering exponential or non-parametric kernels. Although non-parametric kernels are an option, such models require large datasets. While exponential kernels are more data efficient and relevant for specific applications where events immediately trigger more events, they are ill-suited for applications where latencies need to be estimated, such as in neuroscience. This work aims to offer an efficient solution to TPP inference using general parametric kernels with finite support. The developed solution consists of a fast ℓ_2 gradient-based solver leveraging a discretized version of the events. After theoretically supporting the use of discretization, the statistical and computational efficiency of the novel approach is demonstrated through various numerical experiments. Finally, the method's effectiveness is evaluated by modeling the occurrence of stimuli-induced patterns from brain signals recorded with magnetoencephalography (MEG). Given the use of general parametric kernels, results show that the proposed approach leads to an improved estimation of pattern latency than the state-of-the-art.

1. Introduction

The statistical framework of Temporal Point Processes (TPPs; see *e.g.,* Daley & Vere-Jones 2003) is well adapted for modeling event-based data. It offers a principled way to predict the rate of events as a function of time and the previous events' history. TPPs are historically used to model intervals between events, such as in renewal theory, which studies the sequence of intervals between successive replacements of a component susceptible to failure. TPPs find many applications in neuroscience, in particular, to model single-cell recordings and neural spike trains (Truccolo et al., 2005; Okatan et al., 2005; Kim et al., 2011; Rad & Paninski, 2011), occasionally associated with spatial statistics (Pillow et al., 2008) or network models (Galves & Löcherbach, 2015). Multivariate Hawkes processes (MHP; Hawkes 1971) are likely the most popular, as they can model interactions between each univariate process. They also have the peculiarity that a process can be self-exciting, meaning that a past event will increase the probability of having another event in the future on the same process. The conditional intensity function is the key quantity for TPPs. With MHP, it is composed of a baseline parameter and kernels. It describes the probability of occurrence of an event depending on time. The kernel function represents how processes influence each other or themselves. The most commonly used inference method to obtain the baseline and the kernel parameters of MHP is the maximum likelihood (MLE; see *e.g.,* Daley & Vere-Jones, 2007 or Lewis & Mohler, 2011). One alternative and often overlooked estimation criterion is the least squares ℓ_2 error, inspired by the theory of empirical risk minimization (ERM; Reynaud-Bouret & Rivoirard 2010; Hansen et al. 2015; Bacry et al. 2020).

A key feature of MHP modeling is the choice of kernels that can be either non-parametric or parametric. In the non-parametric setting, kernel functions are approximated by histograms (Lewis & Mohler, 2011; Lemonnier & Vayatis, 2014), by a linear combination of pre-defined functions (Zhou et al., 2013a; Xu et al., 2016) or, alternatively, by functions lying in a RKHS (Yang et al., 2017). In addition to the frequentist approach, many Bayesian approaches, such as Gibbs sampling (Ishwaran & James, 2001) or (stochastic) variational inference (Hoffman et al., 2013), have been adapted to MHP in particular to fit non-parametric kernels. Bayesian methods also rely on the modeling of the kernel by histograms (*e.g.,* Donnet et al., 2020) or by a linear combination of pre-defined functions

[*]Equal contribution [1]Université Paris-Saclay, Inria, CEA, Palaiseau, 91120, France. Correspondence to: Guillaume Staerman <guillaume.staerman@inria.fr>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

(*e.g.,* Linderman & Adams, 2015). These approaches are designed whether in continuous-time (Rasmussen, 2013; Zhang et al., 2018; Donnet et al., 2020; Sulem et al., 2021) or in discrete-time (Mohler et al., 2013; Linderman & Adams, 2015; Zhang et al., 2018; Browning et al., 2022). These functions allow great flexibility for the shape of the kernel, yet this comes at the risk of poor estimation of it when only a small amount of data is available (Xu et al., 2017). Another approach to estimating the intensity function is to consider parametrized kernels. Although it can introduce a potential bias by assuming a particular kernel shape, this approach has several benefits. First, it reduces inference burden, as the parameter, say η, is typically lower dimensional than non-parametric kernels. Moreover, for kernels satisfying the Markov property (Bacry et al., 2015), computing the conditional intensity function is linear in the total number of timestamps/events. The most popular kernel belonging to this family is the exponential kernel (Ogata, 1981). It is defined by $\eta = (\alpha, \gamma) \mapsto \alpha\gamma \exp(-\gamma t)$, where α and γ are the scaling and the decay parameters, respectively (Veen & Schoenberg, 2008; Zhou et al., 2013b). However, as pointed out by Lemonnier & Vayatis (2014), the maximum likelihood estimator for MHP with exponential kernels is efficient only if the decay γ is fixed. Thus, only the scaling parameter α is usually inferred. This implies that the hyperparameter γ must be chosen in advance, usually using a grid search, a random search, or Bayesian optimization. This leads to a computational burden when the dimension of the MHP is high. The second option is to define a γ decay parameter common to all kernels, which results in a loss of expressiveness of the model. In both cases, the relevance of the exponential kernel relies on the choice of the decay parameter, which may not be adapted to the data (Hall & Willett, 2016). For more general parametric kernels which do not verify the Markov property, the inference procedure with both MLE or ℓ_2 loss scales poorly as they have quadratic computational scaling with the number of events, making their use limited in practice (see *e.g.,* Bompaire, 2019, Chapter 1). Recently, neural network-based MHP estimation has been introduced, offering, with sufficient data, relevant models at the cost of high computational cost (Mei & Eisner, 2017; Shchur et al., 2019; Pan et al., 2021). These limitations for parametric and non-parametric kernels prevent their usage in some applications, as pointed out by Carreira (2021) in finance or Allain et al. (2021) in neuroscience. A strong motivation for this work is also neuroscience applications.

The quantitative analysis of electrophysiological signals such as electroencephalography (EEG) or magnetoencephalography (MEG) is a challenging modern neuroscience research topic (Cohen, 2014). By giving a non-invasive way to record human neural activity with a high temporal resolution, EEG and MEG offer a unique opportunity to study cognitive processes as triggered by controlled stimulation (Baillet, 2017). Convolutional dictionary learning (CDL) is an unsupervised algorithm recently proposed to study M/EEG signals (Jas et al., 2017; Dupré la Tour et al., 2018). It consists in extracting patterns of interest in M/EEG signals. It learns a combination of time-invariant patterns – called *atoms* – and their activation function to reconstruct the signal sparsely. However, while CDL recovers the local structure of signals, it does not provide any global information, such as interactions between patterns or how their activations are affected by stimuli. Atoms typically correspond to transient bursts of neural activity (Sherman et al., 2016) or artifacts such as eye blinks or heartbeats. By offering an event-based perspective on non-invasive electromagnetic brain signals, CDL makes Hawkes processes amenable to M/EEG-based studies. Given the estimated events, one important goal is to uncover potential temporal dependencies between external stimuli presented to the subject and the appearance of the atoms in the data. More precisely, one is interested in statistically quantifying such dependencies, *e.g.,* by estimating the mean and variance of the neural response latency following a stimulus. In Allain et al. (2021), the authors address this precise problem. Their approach is based on an EM algorithm and a Truncated Gaussian kernel, which can cope with only a few brain data, as opposed to non-parametric kernels, which are more data-hungry. Beyond neuroscience, Carreira (2021) uses a likelihood-based approach using exponential kernels to model order book events. Their approach uses high-frequency trading data, considering the latency at hand in the proposed loss.

This paper proposes a new inference method – named FaDIn – to estimate any parametric kernels for Hawkes processes. Our approach is based on two key features. First, we use finite-support kernels and a discretization applied to the ERM-inspired least-squares loss. Second, we propose to employ some precomputations that significantly reduce the computational cost. We then show, empirically and theoretically, that the implicit bias induced by the discretization procedure is negligible compared to the statistical error. Further, we highlight the efficiency of FaDIn in computation and statistical estimation over the non-parametric approach. Finally, we demonstrate the benefit of using a general kernel with MEG data. The flexibility of FaDIn allows us to model neural response to external stimuli with a much better-adapted kernel than the existing method derived in Allain et al. (2021).

2. Fast Discretized Inference for Hawkes processes (FaDIn)

After recalling key notions of Hawkes processes, we introduce our proposed framework FaDIn.

2.1. Hawkes processes

Given a stopping time $T \in \mathbb{R}_+$ and an observation period $[0, T]$, a temporal point process (TPP) is a stochastic process whose realization consists of a set of distinct timestamps $\mathscr{F}_T = \{t_n,\ t_n \in [0, T]\}$ occurring in continuous time. The behavior of a TPP is fully characterized by its *intensity function* that corresponds to the expected infinitesimal rate at which events are occurring at time $t \in [0, T]$. The values of this function may depend on time (*e.g.,* inhomogeneous Poisson processes) or rely on past events such as self-exciting processes (see Daley & Vere-Jones 2003 for an excellent account of TPP). For the latter, the occurrence of one event will modify the probability of having a new event in the near future. The conditional intensity function $\lambda : [0, T] \to \mathbb{R}_+$ has the following form:

$$\lambda\left(t|\mathscr{F}_t\right) \coloneqq \lim_{\mathrm{d}t \to 0} \frac{\mathbb{P}\left(N_{t+\mathrm{d}t} - N_t = 1|\mathscr{F}_t\right)}{\mathrm{d}t},$$

where $N_t \coloneqq \sum_{n \geq 1} \mathbf{1}_{t_n \leq t}$ is the counting process associated to the PP. Among this family, Multivariate Hawkes processes (MHP; Hawkes, 1971) model the interactions of $p \in \mathbb{N}_*$ self-exciting TPPs. Given p sets of timestamps $\mathscr{F}_T^i = \{t_n^i,\ t_n^i \in [0, T]\}_{n=1}^{N_T^i}, i = 1, \ldots, p$, each process is described by the following intensity function:

$$\lambda_i(t) = \mu_i + \sum_{j=1}^{p} \int_0^t \phi_{ij}(t - s)\, \mathrm{d}N_s^j, \tag{1}$$

where μ_i is the baseline parameter, $N_t = [N_t^1, \ldots, N_t^p]$ the associated multivariate counting process and $\phi_{ij} : [0, T] \to \mathbb{R}_+$ the excitation function – called *kernel* – representing the influence of j-th process' past events onto i-th process' future events. From an inference perspective, the goal is to estimate the baseline and kernels associated with the MHP from the data. In this paper, we focus on the ERM-inspired least squares loss. Assuming a class of parametric kernel parametrized by η, the objective is to find parameters that minimize (see *e.g.,* Eq. (I.2) in Bompaire, 2019, Chapter 1):

$$\mathcal{L}\left(\theta, \mathscr{F}_T\right) = \frac{1}{N_T} \sum_{i=1}^{p} \left(\int_0^T \lambda_i(s)^2\, \mathrm{d}s - 2 \sum_{t_n^i \in \mathscr{F}_T^i} \lambda_i\left(t_n^i\right) \right), \tag{2}$$

where $N_T = \sum_{i=1}^p N_T^i$ is the total number of timestamps, and where $\theta \coloneqq (\mu, \eta)$. Interestingly, when used with an exponential kernel, this loss benefits from some precomputations of complexity $O(N_T)$, making the subsequent iterative optimization procedure independent of N_T. This computational ease is the main advantage of the loss $\mathcal{L}$ over the log-likelihood function. However, when using a general parametric kernel, these precomputations require $O((N_T)^2)$ operations killing the computational benefit of the ℓ_2 loss $\mathcal{L}$ over the log-likelihood. It is worth noting that this loss differs from the quadratic error minimized between the counting processes and the integral of the intensity function, as used in Wang et al. (2016); Eichler et al. (2017) and Xu et al. (2018).

2.2. FaDIn

The approach we propose in this paper fills the need for general parametric kernels in many applications. We provide a computationally and statistically efficient solver – coined FaDIn – that works with many parametric kernels using gradient-based algorithms. Precisely, it relies on three key ideas: (i) the use of parametric finite-support kernels, (ii) a discretization of the time interval $[0, T]$, and (iii) precomputations allowing an efficient optimization procedure detailed below.

Finite support kernels A core bottleneck for MLE or ℓ_2 estimation of parametric kernels is the need to compute the intensity function for all events. For general kernels, the intensity function usually requires $O((N_T)^2)$ operations, which makes it intractable for long-time-length processes. To make this computation more efficient, we consider finite support kernels. Using a finite support kernel amounts to setting a limit in time for the influence of a past event on the intensity, *i.e.,* $\forall t \notin [0\,, W]\,, \phi_{ij}(t) = 0$, where W denotes the length of the kernel's support. This assumption matches applications where an event cannot have influence far in the future, such as in neuroscience (Krumin et al., 2010; Eichler et al., 2017; Allain et al., 2021), genetics (Reynaud-Bouret & Schbath) or high-frequency trading (Bacry et al., 2015; Carreira, 2021). The intensity function (1) can then be reformulated as a convolution between the kernel ϕ_{ij} and the sum of Dirac functions $z_i \coloneqq \sum_{t_n^i \in \mathscr{F}_T^i} \delta_{t_n^i}$ located at the event occurrences t_n^i:

$$\lambda_i(t) = \mu_i + \sum_{j=1}^{p} \phi_{ij} * z_j(t), \quad t \in [0\,, T]\,.$$

As ϕ_{ij} has finite support, the intensity can be computed efficiently with this formula. Indeed, only events in the interval $[t - W\,, t]$ need to be considered. See Section B.2 for more details.

Discretization To make these computations even more efficient, we propose to rely on discretized processes. Most Hawkes processes estimation procedures involve a continuous paradigm to minimize (2) or its log-likelihood counterpart. Discretization has been investigated so far for non-parametric kernels (Kirchner, 2016; Kirchner & Bercher, 2018; Kurisu, 2016). The discretization of a TPP consists in projecting each event t_n^i on a regular grid $\mathcal{G} = \{0, \Delta, 2\Delta, \ldots, G\Delta\}$, where $G = \left\lfloor \frac{T}{\Delta} \right\rfloor$. We refer to Δ as the stepsize of the discretization. Here $\lfloor \cdot \rfloor$ denotes the floor function. Let $\widetilde{\mathscr{F}_T^i}$ be the set of projected timestamps of $\mathscr{F}_T^i$

on the grid $\mathcal{G}$. The intensity function of the i-th process of our discretized MHP is defined as:

$$\begin{aligned}\tilde{\lambda}_i[s] &= \mu_i + \sum_{j=1}^{p} \sum_{\tilde{t}_m^j \in \widetilde{\mathscr{F}}_{s\Delta}^j} \phi_{ij}(s\Delta - \tilde{t}_m^j) \\ &= \mu_i + \sum_{j=1}^{p} \underbrace{\sum_{\tau=1}^{L} \phi_{ij}^{\Delta}[\tau] z_j[s-\tau]}_{(\phi_{ij}^{\Delta} * z_j)[s]}, \quad s \in [\![0\,, G]\!]\,, \quad (3)\end{aligned}$$

where $L = \lfloor \frac{W}{\Delta} \rfloor$ denotes the number of points on the discretized support, $\phi_{ij}^{\Delta}[s] = \phi_{ij}(s\Delta)$ is the kernel value on the grid and $z_i[s] = \#\left\{|t_n^i - s\Delta| \leq \frac{\Delta}{2}\right\}$ denotes the number of events projected on the grid timestamp s. From now and throughout the rest of the paper, we denote $\phi_{ij}(\cdot) : \mathbb{R}_+ \to \mathbb{R}_+$ as a function while $\phi_{ij}^{\Delta}[\cdot]$ represents the discrete vector $\phi_{ij}^{\Delta} \in \mathbb{R}_+^L$. Compared to the continuous formulation, the intensity function can be computed more efficiently as one can rely on discrete convolutions, whose worst-case complexity scales as $O(N_T L)$. It can also be further accelerated using Fast Fourier Transform when N_T is large. Another benefit of the discretization is that for kernels whose values are costly to compute, at most L values need to be calculated. This can have a strong computational impact when $N_T \gg L$ as all values can be precomputed and stored.

While discretization improves the computational efficiency, it also introduces a bias in the computation of the intensity function and, thus potentially, in estimating the kernel parameters. The impact of the discretization on the estimation is considered in Section 2.3 and Section 3.1. Note that this bias is similar to the one incurred by quantizing the kernel as histograms for non-parametric estimators.

Loss and precomputations FaDIn aims at minimizing the discretized ℓ_2 loss, which approximates the integral on the left part of (2) by a sum on the grid $\mathcal{G}$ after projecting timestamps of $\mathscr{F}_T$ on it. It boils down to optimizing the following loss $\mathcal{L}_{\mathcal{G}}\left(\theta, \widetilde{\mathscr{F}_T}\right)$ defined as:

$$\frac{1}{N_T} \sum_{i=1}^{p} \left(\Delta \sum_{s \in [\![0\,,G]\!]} \left(\tilde{\lambda}_i[s]\right)^2 - 2 \sum_{\tilde{t}_n^i \in \widetilde{\mathscr{F}}_T^i} \tilde{\lambda}_i \left[\frac{\tilde{t}_n^i}{\Delta}\right] \right). \quad (4)$$

To find the parameters of the intensity function θ, FaDIn minimizes $\mathcal{L}_{\mathcal{G}}$ using a first-order gradient-based algorithm. The computational bottleneck of the proposed algorithm is thus the computation of the gradient $\nabla \mathcal{L}_{\mathcal{G}}$ regarding parameters θ. Using the discretized finite-support kernel, this gradient can be computed using convolution, giving the same computational complexity as the computation of the intensity function $O(N_T L)$. However, gradient computation can still be too expensive for long processes with many events to get reasonable inference times. Using the least squares error of the process (4), one can further reduce the complexity of computing the gradient by precomputing some constants $\Phi_j(\tau; G)$, $\Psi_{j,k}(\tau, \tau'; G)$ and $\Phi_j(\tau; \widetilde{\mathscr{F}_T^i})$ that do not depend on the parameter θ. Indeed, by developing and rearranging the terms in (4), one obtains:

$$\begin{aligned}N_T\, \mathcal{L}_{\mathcal{G}}\left(\theta, \widetilde{\mathscr{F}_T}\right) &= 2\Delta \sum_{i=1}^{p} \mu_i \sum_{j=1}^{p} \sum_{\tau=1}^{L} \phi_{ij}^{\Delta}[\tau] \underbrace{\left(\sum_{s=1}^{G} z_j[s-\tau]\right)}_{\Phi_j(\tau;G)} \\ &+ \Delta \sum_{i,j,k} \sum_{\tau=1}^{L} \sum_{\tau'=1}^{L} \phi_{ij}^{\Delta}[\tau] \phi_{ik}^{\Delta}[\tau'] \underbrace{\left(\sum_{s=1}^{G} z_j[s-\tau]\, z_k[s-\tau']\right)}_{\Psi_{j,k}(\tau,\tau';G)} \\ &- 2\left(\sum_{i=1}^{p} N_T^i \mu_i + \sum_{i,j} \sum_{\tau=1}^{L} \phi_{ij}^{\Delta}[\tau] \underbrace{\left(\sum_{\tilde{t}_n^i \in \widetilde{\mathscr{F}_T^i}} z_j \left[\frac{\tilde{t}_n^i}{\Delta} - \tau\right]\right)}_{\Phi_j(\tau;\widetilde{\mathscr{F}_T^i})}\right).\end{aligned}$$

The term $\Psi_{j,k}(\tau, \tau'; G)$ dominates the computational cost of our precomputations. It requires $O(G)$ operations for each tuples (τ, τ') and (j, k). Thus, it has a total complexity of $O(p^2 L^2 G)$ and is the bottleneck of the precomputation phase. For any $m \in \{1, \ldots, p\}$, the gradient of the loss w.r.t. the baseline parameter is given by:

$$N_T \frac{\partial \mathcal{L}_G}{\partial \mu_m} = 2T\mu_m - 2N_T^m + 2\Delta \sum_{j=1}^{p} \sum_{\tau=1}^{L} \phi_{mj}^{\Delta}[\tau] \Phi_j(\tau; G).$$

For any tuple $(m, l) \in \{1, \ldots, p\}^2$, the gradient of η_{ml} is:

$$\begin{aligned}N_T \frac{\partial \mathcal{L}_G}{\partial \eta_{ml}} &= 2\Delta\mu_m \sum_{\tau=1}^{L} \frac{\partial \phi_{m,l}^{\Delta}[\tau]}{\partial \eta_{m,l}}\, \Phi_l(\tau; G) \\ &+ 2\Delta \sum_{k=1}^{p} \sum_{\tau=1}^{L} \sum_{\tau'=1}^{L} \phi_{mk}^{\Delta}[\tau'] \frac{\partial \phi_{m,l}^{\Delta}[\tau]}{\partial \eta_{m,l}}\, \Psi_{l,k}(\tau, \tau'; G) \\ &- 2 \sum_{\tau=1}^{L} \frac{\partial \phi_{m,l}^{\Delta}[\tau]}{\partial \eta_{m,l}}\, \Phi_l(\tau; \widetilde{\mathscr{F}_T^m}).\end{aligned}$$

Gradients of kernel parameters dominate the computational cost of gradients. The complexity is of $O(pL^2)$ for each kernel parameter, leading to a total complexity of $O(p^3 L^2)$ and is independent of the number of events N_T. Thus, a trade-off can be made between the precision of the method and its computational efficiency when varying the size of the kernel's support or the discretization.

Remark 2.1. *The primary motivation for the ℓ_2 loss is the presence of terms that can be precomputed in contrast to the log-likelihood (Reynaud-Bouret & Rivoirard, 2010; Reynaud-Bouret et al., 2014; Bacry et al., 2020). An experiment is performed in Section B.1 where it is shown that the statistical error of the ℓ_2 loss and the log-likelihood are equivalent. However, the computation time of FaDIn is advantageous compared to the one with log-likelihood, as FaDIn leverages efficient precomputations.*

Optimization The inference is then conducted using gradient descent for the ℓ_2 loss $\mathcal{L}_\mathcal{G}$. FaDIn thus allows for very general parametric kernels, as exact gradients for each parameter involved in the kernels can be derived efficiently as long as the kernel is differentiable and has finite support. Gradient-based optimization algorithms can, therefore, be used without limitation, in contrast with the EM algorithm, which requires a close-form solution to zero the gradient, which is difficult for many kernels. A critical remark is that the problem is generally non-convex and may converge to a local minimum.

2.3. Impact of the discretization

While discretization allows for efficient computations, it also introduces a perturbation in the loss value. In this section, we quantify the impact of this perturbation on the parameter estimation when Δ goes to 0. Through this section, we observe a process $\mathscr{F}_T$ whose intensity function is given by the parametric form $\lambda(\cdot;\theta^*)$. Note that if the process $\mathscr{F}_T$'s intensity is not in the parametric family $\lambda(\cdot;\theta)$, θ^* is defined as the best approximation of its intensity function in the ℓ_2 sense. The goal of the inference process is thus to recover the parameters θ^*.

When working with the discrete process $\widetilde{\mathscr{F}_T}$, the events t_n^i of the original process are replaced with a projection on a grid $\tilde{t}_n^i = t_n^i + \delta_n^i$. Here, δ_n^i is uniformly distributed on $[-\Delta/2, \Delta/2]$. We consider the discrete FaDIn estimator $\widehat{\theta}_\Delta$ defined as $\widehat{\theta}_\Delta = \arg\min_\theta \mathcal{L}_\mathcal{G}(\theta)$. We can upper-bound the error incurred by $\widehat{\theta}_\Delta$ by the decomposition:

$$\left\|\widehat{\theta}_\Delta - \theta^*\right\|_2 \leq \underbrace{\left\|\widehat{\theta}_c - \theta^*\right\|_2}_{(*)} + \underbrace{\left\|\widehat{\theta}_\Delta - \widehat{\theta}_c\right\|_2}_{(**)}, \tag{5}$$

where $\widehat{\theta}_c = \arg\min_\theta \mathcal{L}(\theta)$ is the reference estimator for θ^* based on the standard ℓ_2 estimator for continuous point processes. This decomposition involves the statistical error $(*)$ and the bias error $(**)$ induced by the discretization. The statistical term measures how far the parameters obtained by minimizing the ℓ_2 continuous loss having access to a finite amount of data are from the true ones. In contrast, the term $(**)$ represents the discretization bias induced by minimizing the discrete loss (4) instead of the continuous one (2). In the following proposition, we focus on the discretization error $(**)$, which is related to the computational trade-off offered by our method and not on the statistical error of the continuous ℓ_2 estimator $(**)$. Our work showcases that this disregarded estimator can be efficiently computed, and we hope it will promote research to describe its asymptotic behavior. We now study the perturbation of the loss due to discretization.

Proposition 2.2. *Let $\mathscr{F}_T$ and $\widetilde{\mathscr{F}_T}$ be respectively a MHP process and its discretized version on a grid $\mathcal{G}$ with step-size Δ. Assume that the intensity function of $\mathscr{F}_T$ possesses continuously differentiable finite support kernels on $[0, W]$. Thus, assuming $\Delta < \min_{t_n^i, t_m^j \in \mathscr{F}_T} |t_n^i - t_m^j|$, for any $i \in [\![1, p]\!]$, it holds:*

$$\widetilde{\lambda}_i[s] = \lambda_i(s\Delta) - \sum_{j=1}^{p} \sum_{t_m^j \in \mathscr{F}_{s\Delta}^j} \delta_m^j \frac{\partial \phi_{ij}}{\partial t}(s\Delta - t_m^j; \theta) + O(\Delta^2),$$

and

$$\mathcal{L}_\mathcal{G}(\theta) = \frac{2}{N_T} \sum_{i,j} \sum_{\substack{t_n^i \in \mathscr{F}_T^i \\ t_m^j \in \mathscr{F}_{s\Delta}^j}} (\delta_m^j - \delta_n^i) \frac{\partial \phi_{ij}}{\partial t}(t_n^i - t_m^j; \theta) + \mathcal{L}(\theta) + \Delta . h(\theta) + O(\Delta^2).$$

The technical proof is deferred to Section A.1 in the Appendix. The first result is a direct application of the Taylor expansion of the intensity for the kernels. For the loss, the first perturbation term $\Delta.h(\theta)$ comes from approximating the integral with a finite Euler sum (Tasaki, 2009) while the second one derives from the perturbation of the intensity. This proposition shows that, as the discretization step Δ goes to 0, the perturbed intensity and ℓ_2 loss are good estimates of their continuous counterpart. We now quantify the discretization error $(**)$ as Δ goes to 0.

Proposition 2.3. *We consider the same assumption as in Proposition 2.2. Then, if the estimators $\widehat{\theta}_c = \arg\min_\theta \mathcal{L}(\theta)$ and $\widehat{\theta}_\Delta = \arg\min_\theta \mathcal{L}_\mathcal{G}(\theta)$ are uniquely defined, $\widehat{\theta}_\Delta$ converges to $\widehat{\theta}_c$ as $\Delta \to 0$. Moreover, if $\mathcal{L}$ is C^2 and its hessian $\nabla^2 \mathcal{L}\left(\widehat{\theta}_c\right)$ is positive definite with $\varepsilon > 0$ its smallest eigenvalue, then $\left\|\widehat{\theta}_\Delta - \widehat{\theta}_c\right\|_2 \leq \frac{\Delta}{\varepsilon} g\left(\widehat{\theta}_\Delta\right)$, with $g\left(\widehat{\theta}_\Delta\right) = O(1)$.*

This proposition shows that asymptotically on Δ, the estimator $\widehat{\theta}_\Delta$ is equivalent to $\widehat{\theta}_c$. It also shows that the discrete estimator converges to the continuous one at the same speed as Δ decreases. This is confirmed experimentally by results shown in Figure B.6 in the Appendix. Thus, one would need to select Δ so that the discretization error is small compared to the statistical one. Notice that assumptions from Proposition 2.3 are not too restrictive. Indeed, they require the existence of a unique minimizer of $\mathcal{L}$, $\mathcal{L}_\mathcal{G}$ and $\mathcal{L}$. Moreover, if $\mathcal{L}$ is C^2 in $\widehat{\theta}_c$, the previous hypothesis also implies the strong local convexity at this point.

3. Numerical experiments

We present various synthetic data experiments to support the advantages of the proposed approach. To begin, we investigate the bias induced by the discretization in Section 3.1. Afterwards, the statistical and computational efficiency of FaDIn is highlighted through a benchmark with

popular non-parametric approaches Section 3.2. Due to the space limitation, sensitivity analysis regarding the parameter W and additional non-parametric comparisons are provided in Appendices B.2 and B.3, respectively.

3.1. Consistency of Discretization

In order to study the estimation bias due to discretization, we run two experiments and report the results in Figure 1 (details and further experiments are presented in Section B.4 and Section B.5 in the Appendix).

The general paradigm is a one-dimensional TPP with intensity parametrized as in (1) with a Truncated Gaussian kernel of mean $m \in \mathbb{R}$ and standard deviation $\sigma > 0$, with fixed support $[0\,, W] \subset \mathbb{R}^+$, $W > 0$. It corresponds to $\phi(\cdot) = \alpha\kappa(\cdot), \alpha \geq 0$ with

$$\kappa(\cdot) := \kappa\left(\cdot; m, \sigma, W\right) = \frac{1}{\sigma}\frac{f\left(\frac{\cdot - m}{\sigma}\right)}{F\left(\frac{W-m}{\sigma}\right) - F\left(\frac{-m}{\sigma}\right)}\mathbf{1}_{0\leq\cdot\leq W},$$

where f (resp. F) is the probability density function (resp. cumulative distribution function) of the standard normal distribution. Hence, the parameters to estimate are μ and $\eta = (\alpha, m, \sigma)$.

In both experiments, for multiple process length T, the discrete estimates are computed for varying grid stepsize Δ, from 10^{-1} to 10^{-3}. The parameter W is set to 1. The ℓ_2 norm of the difference between estimates and the true parameter values – the ones used for data simulation – is computed and reported. We first computed the parameter estimates with our FaDIn method for $T \in \{10^3, 10^5, 10^4, 10^6\}$, for 100 simulations each time. Second, since we wish to separate discretization bias from statistical bias, we compute the estimates with an EM algorithm, both continuously and discretely, and that for 50 random data simulations.

One can observe that the ℓ_2 errors between discrete estimates and true parameters tend towards zero as T increases. For T fixed, one can see plateaus starting for stepsize values that are not particularly small, indicating that the discretization bias is limited. The second experiment with the EM algorithm shows that when plateau is reached, it corresponds to some statistical error. In other words, even for a reasonably coarse stepsize, the bias induced by the discretization is slight compared to the statistical error.

3.2. Statistical and computational efficiency of FaDIn

We compare FaDIn with non-parametric and parametric methods by assessing approaches' statistical and computational efficiency. To learn the non-parametric kernel, we select various existing methods. The first benchmarked method uses histogram kernels and relies on the EM algorithm, provided in Zhou et al. (2013a) and implemented in the `tick` library (Bacry et al., 2017). The kernel is set with one basis function. The three other approaches involve a linear combination of pre-defined raised cosine functions as non-parametric kernels. The inference is made either by stochastic gradient descent algorithm (Non-param SGD; Linderman & Adams, 2014) or by Bayesian approaches such as Gibbs sampling (Gibbs) or Variational Inference (VB) from Linderman & Adams (2015). These algorithms are implemented in the `pyhawkes` library[1]. In the following experiments, we set the number of basis to five for each method. The parametric approach we compare with is the Neural Hawkes Process (NeuralHawkes; Mei & Eisner, 2017) where authors represents the intensity function by a LSTM module. The latter is calculated on a GPU. The experiment is conducted as follows. We simulate a two-dimensional Hawkes process (repeated ten times) using the `tick` library with baseline $\boldsymbol{\mu} = [0.1, 0.2]$ and Raised Cosine kernels:

$$\phi_{i,j}(\cdot) = \alpha_{i,j}\left[1 + \cos\left(\frac{\cdot - u_{i,j}}{\sigma_{i,j}}\pi - \pi\right)\right], (i,j) \in \{1,2\}^2$$

on the support $[u_{i,j}, u_{i,j} + 2\sigma_{i,j}]$ and zero outside with parameters $\boldsymbol{\alpha} = \begin{bmatrix}1.5 & 0.1\\ 0.1 & 1.5\end{bmatrix}$, $\mathbf{u} = \begin{bmatrix}0.1 & 0.3\\ 0.3 & 0.3\end{bmatrix}$ and $\boldsymbol{\sigma} = \begin{bmatrix}0.3 & 0.25\\ 0.3 & 0.3\end{bmatrix}$. Further, we infer the intensity function of these underlying Hawkes processes using FaDIn and the four previously mentioned methods setting $\Delta = 0.01$ for all these discrete approaches. The parameter W of FaDIn is set to 1. This experiment is repeated for varying values of $T \in \{10^3, 10^4, 10^5\}$. The averaged (over the ten runs) normalized ℓ_1 error on the intensity (evaluated on the same discrete grid), as well as the associated computation time, are reported in Figure 2. Due to the high computational times of NeuralHawkes, this approach is performed once and is not applied for $T = 10^5$.

From a statistical perspective, we can observe the advantages of FaDIn inference for varying T over the benchmarked methods. It is worth noting that this result is expected by a parametric approach when the used kernel belongs to the same family as the one with which events have been simulated. Also, only one (long) sequence of data has been used, explaining the poor statistical results of the Neural Hawkes, which is efficient on many repetitions of short sequences due to the massive amount of parameters to infer. From a computational perspective, FaDIn is very efficient compared to benchmarked approaches. Indeed, it scales very well with an increasing time T and then with a growing number of events. In contrast, other methods depend on the number of events and scale linearly with the time T.

[1] https://github.com/slinderman/pyhawkes

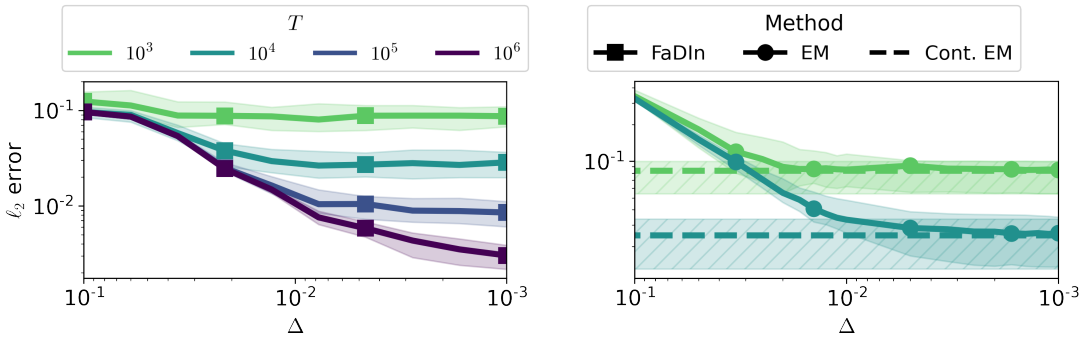

Figure 1. Median and interquartile error bar of the ℓ_2 norm between true parameters and parameter estimates computed with FaDIn (left) and with EM algorithm (right), continuously and discretely, w.r.t. the stepsize of the grid Δ.

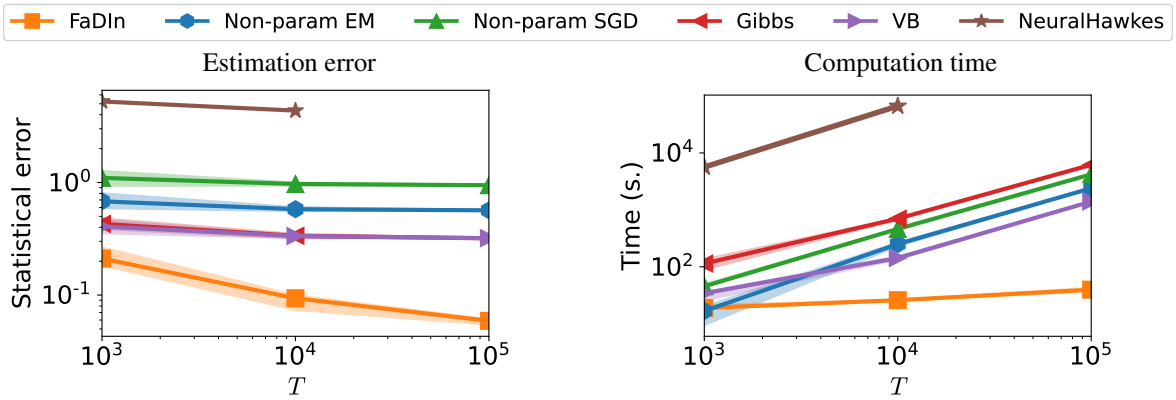

Figure 2. Comparison of the statistical and computational efficiency of FaDIn with five benchmarked methods. The averaged (over ten runs) statistical error on the intensity function (left) and the computational time (right) are computed regarding the time T (and thus the number of events).

4. Application to MEG data

The response's latency related to a stimulus has been identified as a Biomarker of ageing (Price et al., 2017) and many diseases such as epilepsy (Kannathal et al., 2005), Alzheimer's (Dauwels et al., 2010), Parkinson's (Tanaka et al., 2000) or multiple sclerosis (Gil et al., 1993). Therefore, obtaining information on such a feature after auditory or visual stimuli is critical to characterize and eventually detect the presence of a specific disease for a given subject. FaDIn allows fitting a statistical model on this latency by inferring a model on the latency of these responses through Hawkes processes kernels (see Section B.6). This approach characterizes the delays' distribution more finely compared to the latency estimates.

Electrophysiology signals recorded with M/EEG contain recurring prototypical waveforms that can be related to human behavior (Shin et al., 2017). Convolutional Dictionary Learning (CDL; Jas et al. 2017) is an unsupervised method to efficiently extract such patterns and study them in a quantitative way. With CDL, multivariate neural signals are represented by a set of spatio-temporal patterns, called *atoms*, with their respective onsets, called *activations*. Here, we make use of the `alphacsc` software for CDL with rank-1 constraint (Dupré la Tour et al., 2018), as it includes physical priors for the patterns to recover, namely that the spatial propagation of the signal from the brain to sensors is linear and instantaneous. The schema in the Appendix in Figure B.13 presents how CDL applies to MEG recordings.

Experiments on MEG data were run on two datasets from the `MNE` Python package (Gramfort et al., 2013; 2014): the *sample* dataset and the somatosensory (*somato*) dataset[2].

These datasets were selected as they elicit two distinct types of event-related neural activations: evoked responses which are time-locked to the onsets of the stimulation, and induced responses which exhibit larger random jitters. The *sample* dataset contains M/EEG recordings of a human subject presented with audio and visual stimuli. This experiment presents checkerboard patterns to the subject in the left and right visual field, interspersed with tones to the

[2]Both available at `https://mne.tools/stable/overview/datasets_index.html`

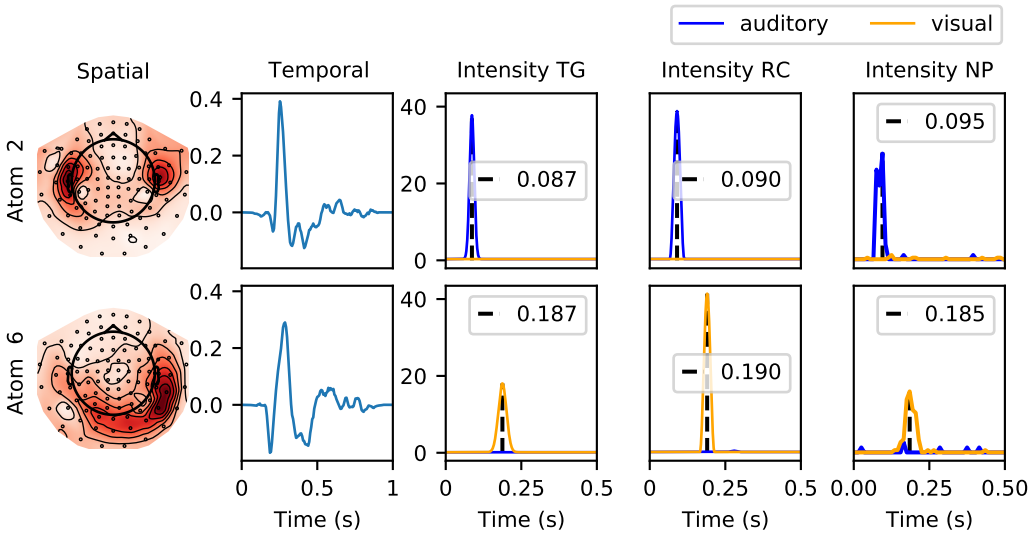

Figure 3. Spatial and temporal patterns of 2 atoms from *sample* dataset, and their respective estimated intensity functions after a stimulus (cue at time = 0 s), for auditory and visual stimuli with non-parametric (NP), Truncated Gaussian (TG) and Raised Cosine (RC) kernels.

left or right ear. The experiment lasts about $4.6\,\mathrm{min}$, and approximately 70 stimuli per type are presented to the subject. For the *somato* dataset, a human subject is scanned with MEG during $15\,\mathrm{min}$, while 111 stimulations of his left median nerve were made. For both datasets, raw data are first preprocessed as done by Allain et al. (2021), and CDL is then applied: 40 atoms of duration $1\,\mathrm{s}$ each are extracted on the *sample* dataset, and 20 atoms of duration $0.53\,\mathrm{s}$ for the *somato* dataset. Finally, each dataset is represented by two sets of Temporal Point Processes: a set of stochastic ones representing atoms' activations, and a set of deterministic ones coding for external stimuli events.

The main goal of applying the TPP framework to such data is to characterize directly when and how each stimulus is responsible for the occurrence of neural responses, especially by estimating the distribution of latencies. We are interested in the paradigm of Driven Point Process (DriPP; Allain et al. 2021) and for every extracted atom, its intensity function related to the corresponding stimuli is estimated using a non-parametric kernel (NP) and two kernel parametrizations: Truncated Gaussian (TG) and Raised Cosine (RC). Results on the *sample* (resp. *somato*) dataset are presented in Figure 3 (resp. Figure B.14 in the Appendix), where only the kernel related to each type of stimulus is plotted, for the sake of clarity. See Appendix B.6.

Results show that all three kernels agree on a peak latency around $90\,\mathrm{ms}$ for the auditory condition and $190\,\mathrm{ms}$ for the visual condition. Due to the limited number of events, one can observe that the non-parametric kernel estimated is less smooth, with spurious peaks later in the interval. Overall, these results on real MEG data demonstrate that our approach with a RC kernel parametrization can recover correct latency estimates even with the discretization of stepsize 0.02. Furthermore, the usage of RC allows us to have sharper peaks in intensity compared to TG, enforcing the link between the external stimulus and the atom's activation. This difference mainly comes from the fact that RC does not need pre-determined support. This advantage is even more pronounced in the case of induced responses, such as in the *somato* dataset (see Figure B.14), where the range of possible latency values is more difficult to determine beforehand.

5. Discussion

This work proposed an efficient approach to infer general parametric kernels for Multivariate Hawkes processes. Our method makes the use of parametric kernels computationally tractable, beyond exponential kernels. The development of FaDIn is based on three key features: (i) finite-support kernels, (ii) timeline discretization, and (iii) precomputations reducing the computational cost of the gradients. These allow for a computationally efficient gradient-based approach, improving state-of-the-art methods while providing flexible use of kernels well-fitted to the considered applications. Moreover, this work shows that the bias induced by the discretization is negligible, both theoretically and numerically. By allowing the use of a general parametric kernel in Hawkes processes, this contribution opens new possibilities for many applications. This is the case with M/EEG data, where estimating information about the rate and latency of occurrences of brain signal patterns is at the core of neuroscience questions. Therefore, FaDIn makes it possible to use a Raised Cosine kernel, allowing for efficient retrieval of these parameters.

References

Allain, C., Gramfort, A., and Moreau, T. DriPP: Driven point processes to model stimuli induced patterns in M/EEG signals. In *International Conference on Learning Representations*, 2021.

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

Bacry, E., Bompaire, M., Deegan, P., Gaïffas, S., and Poulsen, S. V. Tick: A python library for statistical learning, with an emphasis on hawkes processes and time-dependent models. *Journal of Machine Learning Research*, 18(1):7937–7941, 2017.

Bacry, E., Bompaire, M., Gaïffas, S., and Muzy, J.-F. Sparse and low-rank multivariate Hawkes processes. *Journal of Machine Learning Research*, 21(50):1–32, 2020.

Baillet, S. Magnetoencephalography for brain electrophysiology and imaging. *Nature Neuroscience*, 20:327 EP –, 02 2017.

Bompaire, M. *Machine learning based on Hawkes processes and stochastic optimization*. Theses, Université Paris Saclay (COmUE), July 2019.

Browning, R., Rousseau, J., and Mengersen, K. A flexible, random histogram kernel for discrete-time Hawkes processes. *arXiv preprint arXiv:2208.02921*, 2022.

Carreira, M. C. S. Exponential Kernels with Latency in Hawkes Processes: Applications in Finance. *preprint ArXiv*, 2101.06348, 2021.

Cohen, M. X. *Analyzing Neural Time Series Data: Theory and Practice*. The MIT Press, 01 2014. ISBN 9780262319553. doi: 10.7551/mitpress/9609.001.0001.

Daley, D. J. and Vere-Jones, D. *An introduction to the theory of point processes. Volume I: Elementary theory and methods*. Probability and Its Applications. Springer-Verlag New York, 2003.

Daley, D. J. and Vere-Jones, D. *An introduction to the theory of point processes. Volume II: general theory and structure*. Probability and Its Applications. Springer-Verlag New York, 2007.

Dauwels, J., Vialatte, F., and Cichocki, A. Diagnosis of alzheimer's disease from eeg signals: where are we standing? *Current Alzheimer Research*, 7(6):487–505, 2010.

Donnet, S., Rivoirard, V., and Rousseau, J. Nonparametric Bayesian estimation for multivariate Hawkes processes. *The Annals of Statistics*, 48(5):2698 – 2727, 2020.

Dupré la Tour, T., Moreau, T., Jas, M., and Gramfort, A. Multivariate convolutional sparse coding for electromagnetic brain signals. *Advances in Neural Information Processing Systems*, 31:3292–3302, 2018.

Eichler, M., Dahlhaus, R., and Dueck, J. Graphical modeling for multivariate Hawkes processes with nonparametric link functions. *Journal of Time Series Analysis*, 38 (2):225–242, 2017.

Galves, A. and Löcherbach, E. Modeling networks of spiking neurons as interacting processes with memory of variable length. *arXiv preprint arXiv:1502.06446*, 2015.

Gil, R., Zai, L., Neau, J.-P., Jonveaux, T., Agbo, C., Rosolacci, T., Burbaud, P., and Ingrand, P. Event-related auditory evoked potentials and multiple sclerosis. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 88(3):182–187, 1993.

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Goj, R., Jas, M., Brooks, T., Parkkonen, L., et al. MEG and EEG data analysis with MNE-Python. *Frontiers in neuroscience*, 7:267, 2013.

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Parkkonen, L., and Hämäläinen, M. S. MNE software for processing MEG and EEG data. *Neuroimage*, 86:446–460, 2014.

Grosse, R., Raina, R., Kwong, H., and Ng, A. Y. Shift-invariant sparse coding for audio classification. In *23rd Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 149–158. AUAI Press, 2007. ISBN 0-9749039-3-0.

Hall, E. C. and Willett, R. M. Tracking dynamic point processes on networks. *IEEE Transactions on Information Theory*, 62(7):4327–4346, 2016.

Hansen, N. R., Reynaud-Bouret, P., Rivoirard, V., et al. Lasso and probabilistic inequalities for multivariate point processes. *Bernoulli*, 21(1):83–143, 2015.

Hari, R. Action–perception connection and the cortical mu rhythm. *Progress in brain research*, 159:253–260, 2006.

Hawkes, A. G. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 33(3):438–443, 1971.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.

Ishwaran, H. and James, L. F. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001.

Jas, M., Dupré La Tour, T., Şimşekli, U., and Gramfort, A. Learning the morphology of brain signals using alpha-stable convolutional sparse coding. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1–15, 2017.

Kannathal, N., Acharya, U. R., Lim, C. M., and Sadasivan, P. Characterization of EEG—a comparative study. *Computer methods and Programs in Biomedicine*, 80(1): 17–23, 2005.

Kim, S., Putrino, D., Ghosh, S., and Brown, E. N. A granger causality measure for point process models of ensemble neural spiking activity. *PLoS Comput Biol*, 7 (3):e1001110, 2011.

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

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

Krumin, M., Reutsky, I., and Shoham, S. Correlation-based analysis and generation of multiple spike trains using hawkes models with an exogenous input. *Frontiers in computational neuroscience*, 4:147, 2010.

Kurisu, D. Discretization of self-exciting peaks over threshold models. 2016. doi: 10.48550/ARXIV.1612.06109.

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

Lewis, E. and Mohler, G. A nonparametric EM algorithm for multiscale Hawkes processes. *Journal of Nonparametric Statistics*, 1(1):1–20, 2011.

Linderman, S. and Adams, R. Discovering latent network structure in point process data. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pp. 1413–1421, 2014.

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

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

Mohler, G. et al. Modeling and estimation of multi-source clustering in crime and security data. *The Annals of Applied Statistics*, 7(3):1525–1539, 2013.

Ogata, Y. On lewis' simulation method for point processes. *IEEE transactions on information theory*, 27(1):23–31, 1981.

Okatan, M., Wilson, M. A., and Brown, E. N. Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity. *Neural computation*, 17(9):1927–1961, 2005.

Pan, Z., Wang, Z., Phillips, J. M., and Zhe, S. Self-adaptable point processes with nonparametric time decays. *Advances in Neural Information Processing Systems*, 34:4594–4606, 2021.

Pillow, J. W., Shlens, J., Paninski, L., Sher, A., Litke, A. M., Chichilnisky, E. J., and Simoncelli, E. P. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999, 2008.

Price, D., Tyler, L. K., Neto Henriques, R., Campbell, K. L., Williams, N., Treder, M. S., Taylor, J. R., and Henson, R. Age-related delay in visual and auditory evoked responses is mediated by white-and grey-matter differences. *Nature communications*, 8(1):15671, 2017.

Rad, K. and Paninski, L. Information rates and optimal decoding in large neural populations. *Advances in neural information processing systems*, 24:846–854, 2011.

Rasmussen, J. G. Bayesian inference for hawkes processes. *Methodology and Computing in Applied Probability*, 15 (3):623–642, 2013.

Reynaud-Bouret, P. and Rivoirard, V. Near optimal thresholding estimation of a poisson intensity on the real line. *Electronic journal of statistics*, 4:172–238, 2010.

Reynaud-Bouret, P. and Schbath, S. Adaptive estimation for Hawkes processes; application to genome analysis. *The Annals of Statistics*, 38(5):2781 – 2822.

Reynaud-Bouret, P., Rivoirard, V., Grammont, F., and Tuleau-Malot, C. Goodness-of-fit tests and nonparametric adaptive estimation for spike train analysis. *The Journal of Mathematical Neuroscience*, 4(1):1–41, 2014.

Shchur, O., Biloš, M., and Günnemann, S. Intensity-free learning of temporal point processes. *arXiv preprint arXiv:1909.12127*, 2019.

Sherman, M. A., Lee, S., Law, R., Haegens, S., Thorn, C. A., Hämäläinen, M. S., Moore, C. I., and Jones, S. R. Neural mechanisms of transient neocortical beta rhythms: Converging evidence from humans, computational modeling, monkeys, and mice. *Proceedings of the*

National Academy of Sciences, 113(33):E4885–E4894, 2016.

Shin, H., Law, R., Tsutsui, S., Moore, C. I., and Jones, S. R. The rate of transient beta frequency events predicts behavior across tasks and species. *eLife*, 6:e29086, nov 2017.

Sulem, D., Rivoirard, V., and Rousseau, J. Bayesian estimation of nonlinear Hawkes process. *arXiv preprint arXiv:2103.17164*, 2021.

Tanaka, H., Koenig, T., Pascual-Marqui, R. D., Hirata, K., Kochi, K., and Lehmann, D. Event-related potential and EEG measures in parkinson's disease without and with dementia. *Dementia and geriatric cognitive disorders*, 11(1):39–45, 2000.

Tasaki, H. Convergence rates of approximate sums of Riemann integrals. *Journal of Approximation Theory*, 161 (2):477–490, 2009. ISSN 0021-9045.

Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., and Brown, E. N. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of neurophysiology*, 93(2):1074–1089, 2005.

Veen, A. and Schoenberg, F. P. Estimation of space–time branching process models in seismology using an em–type algorithm. *Journal of the American Statistical Association*, 103(482):614–624, 2008.

Wang, Y., Xie, B., Du, N., and Song, L. Isotonic hawkes processes. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pp. 2226–2234. PMLR, 2016.

Xu, H., Farajtabar, M., and Zha, H. Learning Granger causality for Hawkes processes. In *International conference on machine learning*, pp. 1717–1726, 2016.

Xu, H., Luo, D., and Zha, H. Learning hawkes processes from short doubly-censored event sequences. In *International Conference on Machine Learning*, pp. 3831–3840. PMLR, 2017.

Xu, H., Luo, D., Chen, X., and Carin, L. Benefits from superposed Hawkes processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 623–631. PMLR, 2018.

Yang, Y., Etesami, J., He, N., and Kiyavash, N. Online learning for multivariate hawkes processes. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Zhang, R., Walder, C., Rizoiu, M.-A., and Xie, L. Efficient non-parametric bayesian hawkes processes. *arXiv preprint arXiv:1810.03730*, 2018.

Zhou, K., Zha, H., and Song, L. Learning triggering kernels for multi-dimensional Hawkes processes. In *International conference on machine learning*, pp. 1301–1309. PMLR, 2013a.

Zhou, K., Zha, H., and Song, L. Learning social infectivity in sparse low-rank networks using multi-dimensional Hawkes processes. In *Artificial Intelligence and Statistics*, pp. 641–649. PMLR, 2013b.

A. Technical details

This part presents proofs of the theoretical results proposed in the core paper.

A.1. Proof of Proposition 2.2

Recall that by definition,

$$\lambda_i(s\Delta) = \mu_i + \sum_{j=1}^{p} \sum_{t_m^j \in \mathscr{F}_{s\Delta}^j} \phi_{ij}(s\Delta - t_m^j),$$

and

$$\begin{aligned}\tilde{\lambda}_i[s] &= \mu_i + \sum_{j=1}^{p} \sum_{\tilde{t}_m^j \in \widetilde{\mathscr{F}}_{s\Delta}^j} \phi_{ij}(s\Delta - \tilde{t}_m^j) \\ &= \mu_i + \sum_{j=1}^{p} \sum_{t_m^j \in \mathscr{F}_{s\Delta}^j} \phi_{ij}(s\Delta - t_m^j - \delta_m^j), \end{aligned} \tag{6}$$

where (6) is a consequence of hypothesis $\Delta < \min_{t_n^i, t_m^j \in \mathscr{F}_T} |t_n^i - t_m^j|$ which ensures that no event collapses on the same bin of the grid and that $\widetilde{\mathscr{F}}_{s\Delta}^j = \mathscr{F}_{s\Delta}^j$. Note that this hypothesis also implies that the intensity function is smooth for all points on the grid $\mathcal{G}$. Applying the first-order Taylor expansion to the kernels ϕ_{ij} in $s\Delta - t_m^j$ and bounding the perturbation δ_n^i by Δ yields the first result of the proposition.

For the perturbation of the loss $\mathcal{L}_{\mathcal{G}}$, we have:

$$\begin{aligned}\mathcal{L}_{\mathcal{G}}\left(\theta, \widetilde{\mathscr{F}}_T\right) &= \frac{1}{N_T} \sum_{i=1}^{p} \left(\Delta \sum_{s \in [\![0, G]\!]} \left(\tilde{\lambda}_i[s]\right)^2 - 2 \sum_{\tilde{t}_n^i \in \widetilde{\mathscr{F}}_T^i} \tilde{\lambda}_i \left[\frac{\tilde{t}_n^i}{\Delta}\right] \right) \\ &= \mathcal{L}(\theta) + \frac{1}{N_T} \sum_{i=1}^{p} \Bigg(\underbrace{\Delta \sum_{s=0}^{G} \tilde{\lambda}_i[s]^2 - \int_0^T \lambda_i(t)^2 \mathrm{d}t}_{(*)} \underbrace{- 2 \sum_{t_n^i \in \mathscr{F}_T^i} \widetilde{\lambda}_i \left[\frac{\widetilde{t}_n^i}{\Delta}\right] - \lambda_i\left(t_n^i\right)}_{(**)} \Bigg).\end{aligned}$$

The first term $(*)$ is the error of a Riemann approximation of the integral. Theorem 1.2 in Tasaki (2009) shows that asymptotically with $\Delta \to 0$,

$$\Delta \sum_{s=0}^{G} \tilde{\lambda}_i[s]^2 - \int_0^T \lambda_i(t)^2 \mathrm{d}t = \Delta . h_i(\theta) + O(\Delta^2), \tag{7}$$

where $h_i(\theta) = \frac{1}{2}\Big(\int_0^T |\lambda_i(t;\theta) \frac{\partial \lambda_i}{\partial t}(t;\theta)|^{1/2} \mathrm{d}t \Big)^2$ and we denote $h(\theta) = \frac{1}{N_T} \sum_{i=1}^{p} h_i(\theta)$.

For the second term $(**)$, we re-use the expression from (6) but use a Taylor expansion in $t_n^i - t_m^j$. The perturbation becomes $\delta_m^j - \delta_n^i$,

$$\sum_{t_n^i \in \mathscr{F}_T^i} \widetilde{\lambda}_i \left[\frac{\widetilde{t}_n^i}{\Delta}\right] - \lambda_i\left(t_n^i\right) = \sum_{t_n^i \in \mathscr{F}_T^i} \left(\delta_n^i - \delta_m^j\right) \frac{\partial \phi_{ij}}{\partial t}\left(t_n^i - t_m^j; \theta\right) + O\left(\Delta^2\right). \tag{8}$$

Summing (7) and (8) concludes the proof.

A.2. Proof of Proposition 2.3

We consider the two estimators $\widehat{\theta}_\Delta = \arg\min_\theta \mathcal{L}_{\mathcal{G}}(\theta)$ and $\widehat{\theta}_c = \arg\min_\theta \mathcal{L}(\theta)$. With the loss approximation from Proposition 2.2, we have a pointwise convergence of $\mathcal{L}_{\mathcal{G}}(\theta)$ towards $\mathcal{L}(\theta)$ for all $\theta \in \Theta$ as Δ goes to 0. By continuity of

$\mathcal{L}_{\mathcal{G}}$, we have that the limit of $\widehat{\theta}_\Delta$ when Δ goes to 0 exists and is equal to $\widehat{\theta}_c$. This proves that the discretized estimator converges to the continuous one as Δ decreases.

We now characterize its asymptotic speed of convergence. The KKT conditions impose that:

$$\nabla \mathcal{L}_{\mathcal{G}}\left(\widehat{\theta}_\Delta\right) = 0 \qquad \text{and} \qquad \nabla \mathcal{L}\left(\widehat{\theta}_c\right) = 0. \tag{9}$$

Using the approximation from Proposition 2.2, one gets in the limit of small Δ:

$$\nabla \mathcal{L}_{\mathcal{G}}\left(\widehat{\theta}_\Delta\right) = \nabla \mathcal{L}\left(\widehat{\theta}_\Delta\right) + \Delta.\frac{\partial h}{\partial \theta}\left(\widehat{\theta}_\Delta\right) + O\left(\Delta^2\right) + \frac{2}{N_T}\sum_{i=1}^{p}\sum_{t_n^i \in \widetilde{\mathscr{F}}_T^i}\sum_{j=1}^{p}\sum_{t_m^j \in \widetilde{\mathscr{F}}_{s\Delta}^j}\left(\delta_m^j - \delta_n^i\right)\frac{\partial^2 \phi_{ij}}{\partial t \partial \theta}\left(t_n^i - t_m^j; \widehat{\theta}_\Delta\right).$$

Combining this with (9), we get:

$$\nabla \mathcal{L}\left(\widehat{\theta}_\Delta\right) = -\Delta.\frac{\partial h}{\partial \theta}\left(\widehat{\theta}_\Delta\right) + \frac{2}{N_T}\sum_{i=1}^{p}\sum_{t_n^i \in \widetilde{\mathscr{F}}_T^i}\sum_{j=1}^{p}\sum_{t_m^j \in \widetilde{\mathscr{F}}_{s\Delta}^j}\left(\delta_n^i - \delta_m^j\right)\frac{\partial^2 \phi_{ij}}{\partial t \partial \theta}\left(t_n^i - t_m^j; \widehat{\theta}_\Delta\right) + O\left(\Delta^2\right),$$

and

$$\begin{aligned}
\left\|\nabla \mathcal{L}(\widehat{\theta}_\Delta) - \nabla \mathcal{L}\left(\widehat{\theta}_c\right)\right\|_2 &= \left\|-\Delta.\frac{\partial h}{\partial \theta}\left(\widehat{\theta}_\Delta\right) + \frac{2}{N_T}\sum_{i,j=1}^{p}\sum_{t_n^i \in \widetilde{\mathscr{F}}_{s\Delta}^i}\sum_{t_m^j \in \widetilde{\mathscr{F}}_{s\Delta}^j}\left(\delta_n^i - \delta_m^j\right)\frac{\partial^2 \phi_{ij}}{\partial t \partial \theta}\left(t_n^i - t_m^j; \widehat{\theta}_\Delta\right)\right\|_2 + O\left(\Delta^2\right) \\
&\leq \Delta\left\|\frac{\partial h}{\partial \theta}\left(\widehat{\theta}_\Delta\right) + \frac{2}{N_T}\sum_{i,j=1}^{p}\sum_{t_n^i \in \widetilde{\mathscr{F}}_{s\Delta}^i}\sum_{t_m^j \in \widetilde{\mathscr{F}}_{s\Delta}^j}\frac{\partial^2 \phi_{ij}}{\partial t \partial \theta}\left(t_n^i - t_m^j; \widehat{\theta}_\Delta\right)\right\|_2 + O\left(\Delta^2\right) \\
&\leq \Delta.g\left(\widehat{\theta}_\Delta\right),
\end{aligned}$$

where $g(\theta)$ is equal to $\|\frac{\partial h}{\partial \theta}(\widehat{\theta}_\Delta) + \frac{2}{N_T}\sum_{i,j=1}^{p}\sum_{t_n^i \in \widetilde{\mathscr{F}}_{s\Delta}^i}\sum_{t_m^j \in \widetilde{\mathscr{F}}_{s\Delta}^j}\frac{\partial^2 \phi_{ij}}{\partial t \partial \theta}(t_n^i - t_m^j; \widehat{\theta}_\Delta)\|_2 + O(\Delta)$. This function is a $O(1)$. Using the hypothesis that the hessian $\nabla^2 \mathcal{L}(\widehat{\theta}_c)$ exists and is positive definite with smallest eigenvalue ε, we have:

$$\begin{aligned}
\varepsilon\left\|\widehat{\theta}_\Delta - \widehat{\theta}_c\right\|_2^2 &\leq \left\|\nabla \mathcal{L}\left(\widehat{\theta}_\Delta\right) - \nabla \mathcal{L}\left(\widehat{\theta}_c\right)\right\|_2^2 \\
\textit{i.e.,} \qquad \left\|\widehat{\theta}_\Delta - \widehat{\theta}_c\right\|_2^2 &\leq \frac{\Delta}{\varepsilon} g\left(\widehat{\theta}_\Delta\right).
\end{aligned}$$

This concludes the proof.

B. Additional Experiments

This section presents additional experimental results supporting the claims of the paper. We compare the ℓ_2 loss involved in FaDIn with the popular negative Log-Likelihood in Section B.1. A sensitivity analysis of the kernel length W is provided in Section B.2. Additional comparisons with popular non-parametric approaches are presented in Section B.3. The consistency of the discretization is supported in Section B.4 and Section B.5. Finally, we explain the methodology we employed on real MEG data and provide complementary results in Section B.6.

B.1. Comparison of FaDIn with the negative log-likelihood loss

We compare both approaches' statistical and computational efficiency to highlight the benefit of using the ℓ_2 loss in FaDIn over the log-likelihood (LL). Precisely, we compare the accuracy of the obtained parameter estimators from FaDIn and the minimization of the negative log-likelihood in the same setting as our approach (discretization and finite-support kernels). We conduct the experiment as follows. We place ourselves in the univariate setting for computational simplicity. We

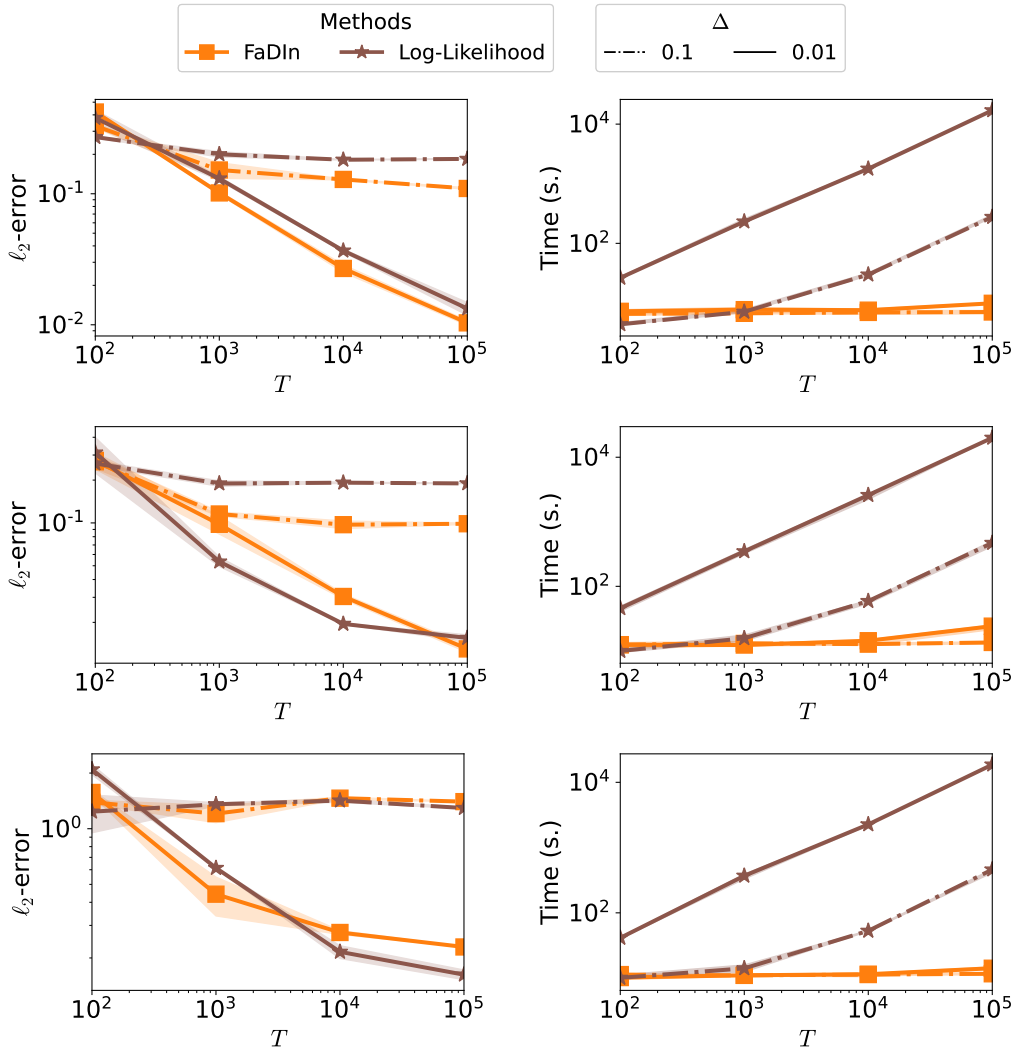

Figure B.1. Comparison of the statistical and computational efficiency of FaDIn with the Log-Likelihood loss. The averaged (over ten runs) statistical error on the intensity function (left) and the computational time (right) are computed regarding the time T for the Raised Cosine (top), the Truncated Gaussian (middle) and the Truncated Exponential (bottom).

sample a set of events in continuous time through the `tick` library. Three sets are sampled from the kernel shapes: Raised Cosine, Truncated Gaussian, and Truncated Exponential. The parameters are set as $\mu = 0.3$, $\alpha = 0.8$, $(u, \sigma) = (0.2, 0.3)$ for the Raised Cosine, $(m, \sigma) = (0.5, 0.3)$ for the Truncated Gaussian and $\gamma = 5$ for the Truncated Exponential. We set the kernel length W to 1 for each setting. Further, we estimate the parameters of the intensity of sampled events using both FaDIn and LL approaches. The experiment is repeated ten times. The median and 25-75% quantiles of the statistical accuracy and the computation time are reported in Figure B.1 for the three different kernels. We can observe an equivalent accuracy of the parameter estimation for both methods along the different kernels, stepsize and number of events. In contrast, the computational performance of FaDIn outperforms the LL approach. Indeed, the computational time is divided by ≈ 5 in a low data regime with $T = 10^2$ and by ≈ 1000 when $\Delta = 0.01$ and $T = 10^5$. This experiment clearly shows the advantages of using the ℓ_2 loss in FaDIn rather than the log-likelihood.

B.2. Kernel length on FaDIn estimates

To study the estimation bias induced by the finite support kernels, we conduct an experiment using FaDIn with a (Truncated) Exponential kernel. The general framework is a one-dimensional TPP with intensity parametrized as in (1) with a

Truncated Exponential kernel having a decay parameter γ, with fixed support $[0\,, W] \subset \mathbb{R}^+$, $W > 0$. It corresponds to $\phi(\cdot) = \alpha\kappa(\cdot), \alpha \geq 0$ with

$$\kappa(\cdot) := \kappa\left(\cdot; \gamma, a, b\right) = \frac{h\left(\cdot\right)}{H\left(b\right) - H\left(a\right)} \mathbf{1}_{a \leq \cdot \leq b},$$

where here h (resp. H) is the probability density function of parameter γ (resp. cumulative distribution function) of the exponential distribution. Therefore, when $W \to \infty$, this Truncated Exponential kernel converges to the standard exponential kernel, *i.e.,* $t \mapsto \alpha\gamma \exp(-\gamma t)$. The parameters to estimate are μ and $\eta = (\alpha, \gamma)$. The experiment is conducted as follows. We simulate events (10 repetitions) from a Hawkes process with baseline $\mu = 1.1$ and a standard Exponential kernel (non-truncated) with $\alpha = 0.8$, $\gamma = 0.5$ for varying $T \in \{10^3, 10^4, 10^5, 10^6\}$ using the `tick` Python library. FaDIn is then computed on each of these sets of events using a Truncated Exponential kernel of length $W \in [1, 100]$ and a stepsize $\Delta = 0.01$. The averaged (over ten runs) and the 25-75% quantiles statistical ℓ_2-error of parameters (left) and computational time (right) are displayed w.r.t. the stepsize of the grid Δ in Figure B.2. On the one hand, one can observe that the ℓ_2-error converges to a plateau once $W > 10$, *i.e.,* the bias induced by the finite support length is reduced. On the other hand, the computational time increase when W increases. Interestingly, for each T, the computational time is close when W is high enough (close to 100). Indeed, optimizing the loss becomes the bottleneck of FaDIn since the grid size ($G = TL + 1$) only intervenes in the precomputation part.

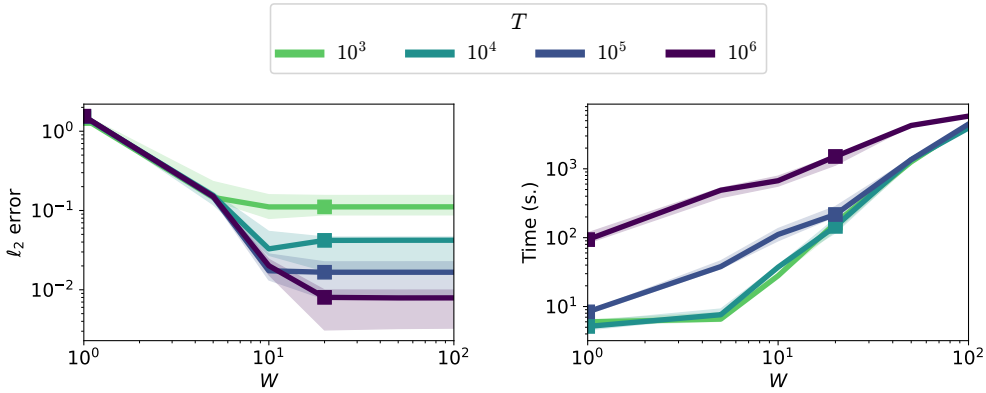

Figure B.2. Comparison of the influence of the kernel support size W on the parameter estimation of FaDIn for a Truncated Exponential kernel. The averaged (over 10 runs) statistical ℓ_2-error (left) and computational time (right) are displayed w.r.t. the stepsize of the grid Δ.

B.3. Statistical and computational efficiency of FaDIn

This part presents additional non-parametric comparisons.

B.3.1. Qualitative Comparison with a non-parametric approach

We compare FaDIn with the use of a non-parametric kernel by assessing the statistical and computational efficiency of both approaches. To learn the non-parametric kernel, we select the EM algorithm, provided in Zhou et al. (2013a) and implemented in the `tick` library (Bacry et al., 2017). The kernel is set with one basis function. In addition, we display the running time when computing gradients using PyTorch and automatic differentiation applied to the $\mathcal{L}_G$ discretized loss (4).

The experiment is conducted as follows. We fix $p = 1$ for simplicity, set $\mu = 1.1$ and choose a Raised Cosine kernel defined by:

$$\phi(\cdot) = \alpha \left[1 + \cos\left(\frac{\cdot - u}{\sigma}\pi - \pi\right)\right] \mathbb{I}\left\{\cdot \in [u;\ u + 2\sigma]\right\},$$

setting parameters $\alpha = 0.8$, $u = 0.2$ and $\sigma = 0.3$. We simulate events in a continuous time using the `tick` library (Bacry et al., 2017). FaDIn and the non-parametric kernel are optimized over 800 iterations (with an early stopping for the EM algorithm). The RMSprop algorithm is used in FaDIn. The discretization size of the non-parametric kernel is settled as in FaDIn. This experiment is done varying $T \in \{10^3, 10^5, 10^6\}$.

On the one hand, in a relatively small data regime where $T = 10^3$, we evaluate the statistical accuracy of the estimated kernel of both methods with the discretization parameter $\Delta = 0.01$. As we can see in Figure B.3 (top left), the non-parametric approach fails to recover the structure of the kernel. The non-parametric approach results in noisy kernel estimates, with probability mass where the kernel is zero. In contrast, FaDIn can recover the kernel parameters used to simulate data even with a small number of events. On the other hand, we evaluate the computational times varying the discretization steps in a large data regime where $T = 10^5$ and $T = 10^6$ with the same simulation parameters. Figure B.3 (bottom left) reports the average computational times (over 10 runs) regarding the discretization stepsize Δ and the dimension p. Although both approaches can recover the kernel under which we simulate data (see Figure B.3, top right), FaDIn is a great deal more computationally efficient than the non-parametric and the automatic differentiation implementations, improving the computational speed by ≈ 100 when $\Delta \in [0.1, 0.01]$ and by ≈ 10 when $\Delta \approx 0.001$. The computation speed regarding the dimension of the MHP is improved by ≈ 10. It is worth noting that the ℓ_2-Autodiff explodes in memory when $\Delta > 0.01$ or when the dimension grows. Additional shapes of kernels are displayed in Figure B.4 for the Truncated Gaussian and in Figure B.5 for the Truncated Exponential kernels.

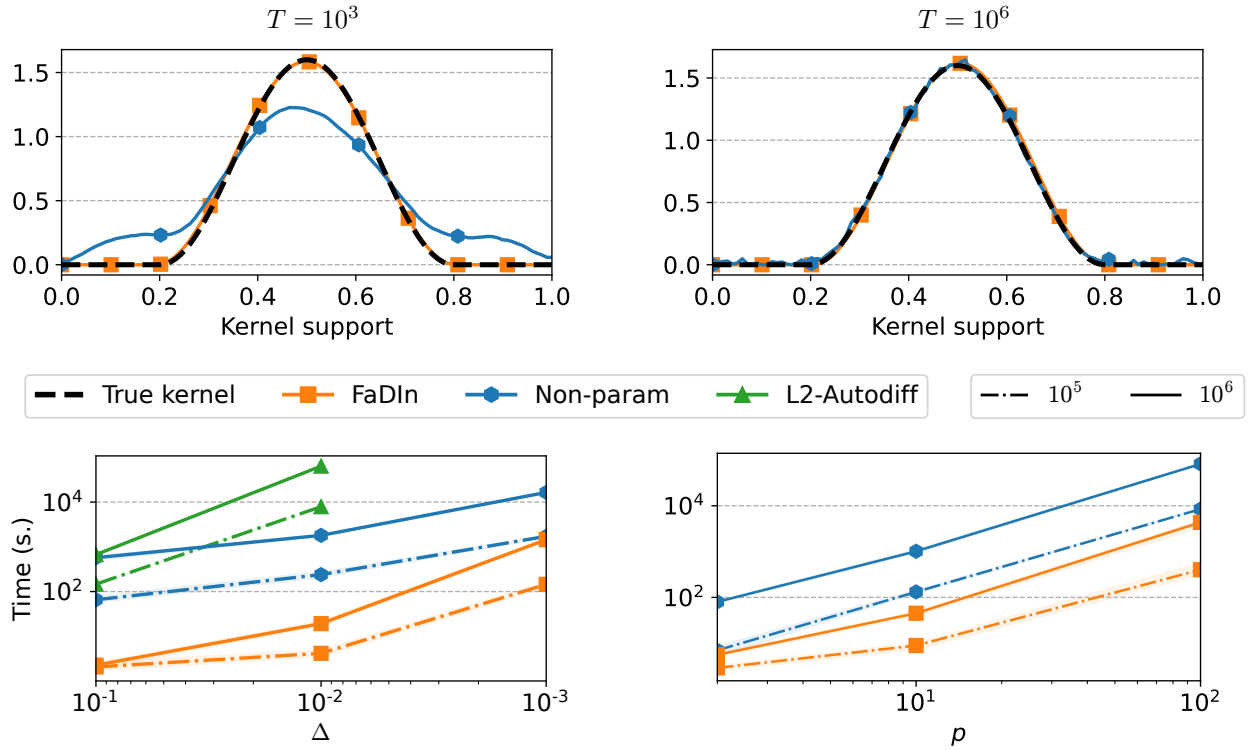

Figure B.3. Comparison between our approach FaDIn and non-parametric approach. Estimated kernels with $\Delta = 0.01$ in a relatively small data setting with $T = 10^3$ (top left), in a large data setting with $T = 10^6$ (top right), and computation time in a large data setting with $T \in \{10^5, 10^6\}$ w.r.t. the stepsize Δ (bottom left) and the dimension p (bottom right). In contrast to non-parametric kernels, FaDIn estimates well the true kernel in a small regime while it is computationally faster than non-parametric kernels in a large regime.

B.4. Discretization on EM estimates (DriPP)

Figure B.6 displays the convergence of the estimator $\widehat{\theta}_\Delta$ towards $\widehat{\theta}_c$ as Δ goes to 0 in the same experimental setup as the right part of Figure 1.

Figure B.7 presents the detailed results, *i.e.,* parameter-wise, of the experiment shown in Figure 1 (right). In this experiment, we are interested in the context of Driven PP (Allain et al., 2021) with an exogenous homogeneous PP. The simulation parameter of the latter is set to 0.5, meaning that on average, 1 event occurs every 2 seconds on the driver.

Figure B.8 presents the results of the same experiment with Poisson parameter set to 0.1 which represents roughly five times less events.

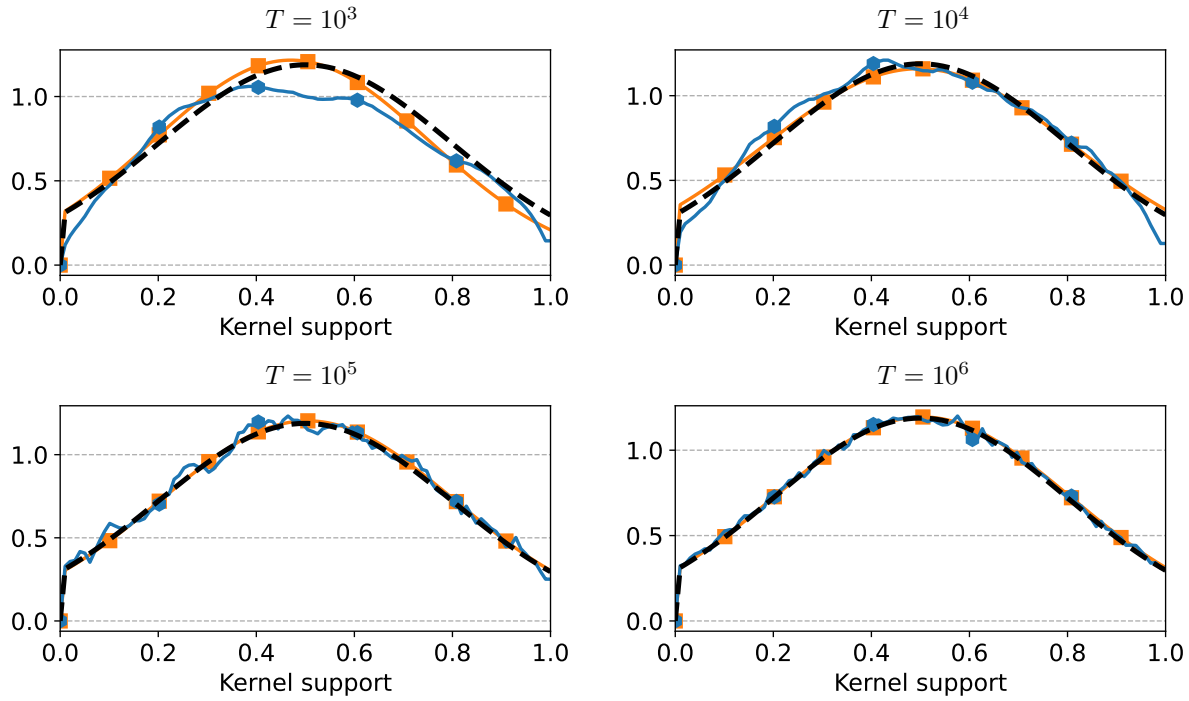

Figure B.4. Comparison between our approach FaDIn and non-parametric approach for a Truncated Gaussian kernel. Estimated kernels with $\Delta = 0.01$ and $T \in \{10^3, 10^4, 10^5, 10^6\}$. The true kernel, FaDIn and the non-parametric approach are depicted in black, orange and blue, respectively.

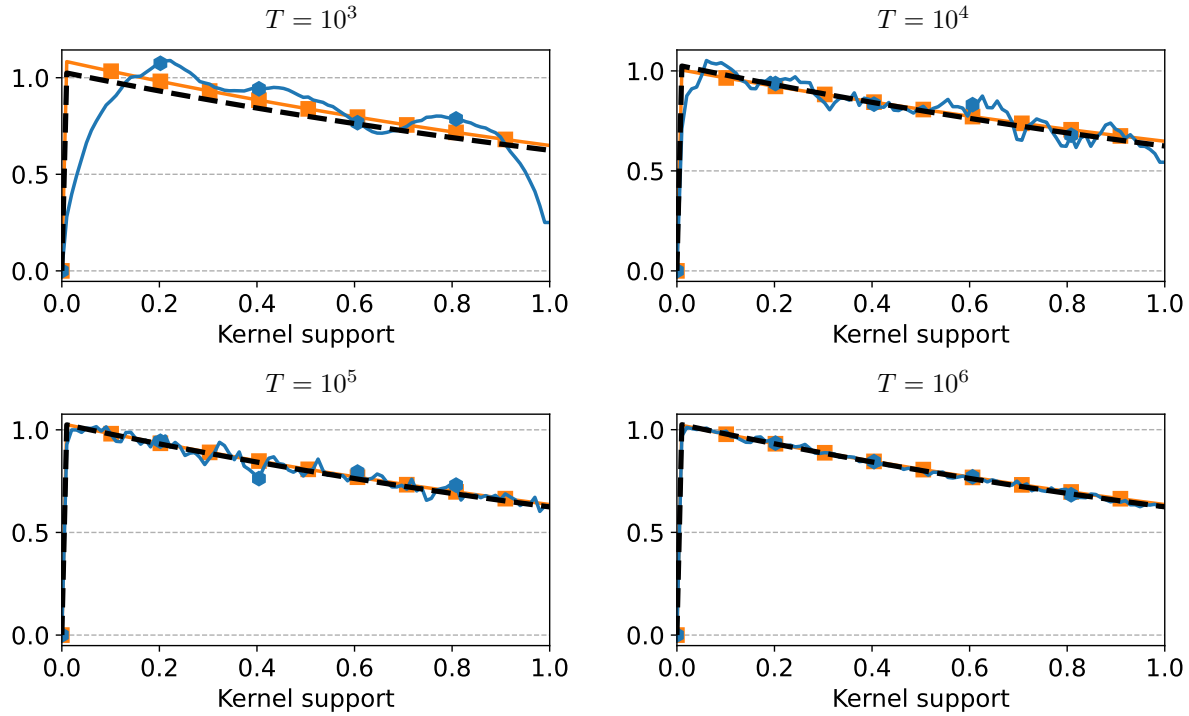

Figure B.5. Comparison between our approach FaDIn and non-parametric approach for a Truncated Exponential kernel. Estimated kernels with $\Delta = 0.01$ and $T \in \{10^3, 10^4, 10^5, 10^6\}$. The true kernel, FaDIn and the non-parametric approach are depicted in black, orange and blue, respectively.

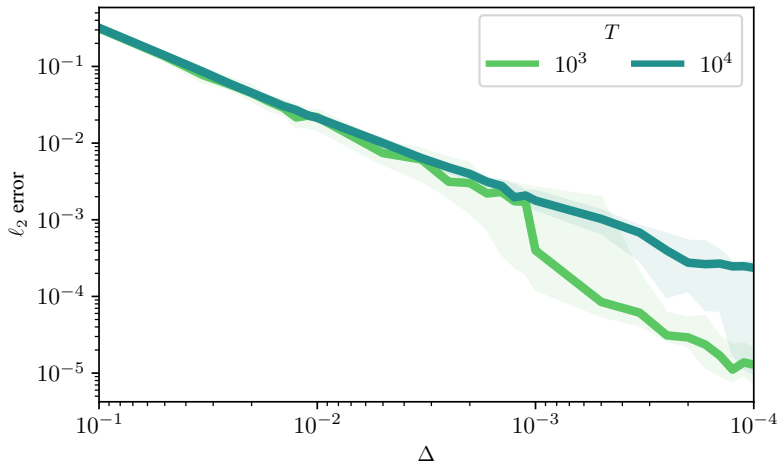

Figure B.6. Median and interquartile error bar of the ℓ_2 norm between the parameters estimated computed with EM algorithm, continuously and discretely, w.r.t. the stepsize Δ. This figure confirms the results from Proposition 2.2; that is, that the convergence of $\widehat{\theta}_\Delta$ towards $\widehat{\theta}_c$ is linear with respect to Δ.

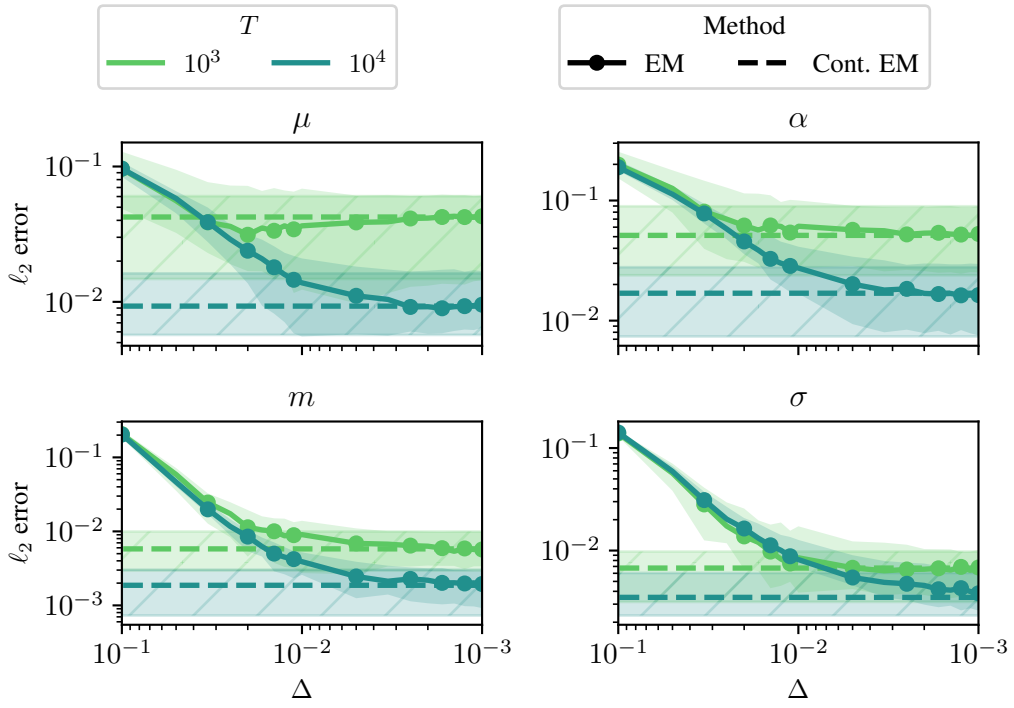

Figure B.7. Median and interquartile error bar of the ℓ_2 norm between true parameters and parameter estimates computed with EM algorithm, continuously and discretely, w.r.t. the stepsize Δ.

B.5. Discretization effect on FaDIn estimates

This section presents additional results related to the Section 3.1. We reproduce the experiments of this section with FaDIn and two other kernels: Raised Cosine and Truncated Exponential.

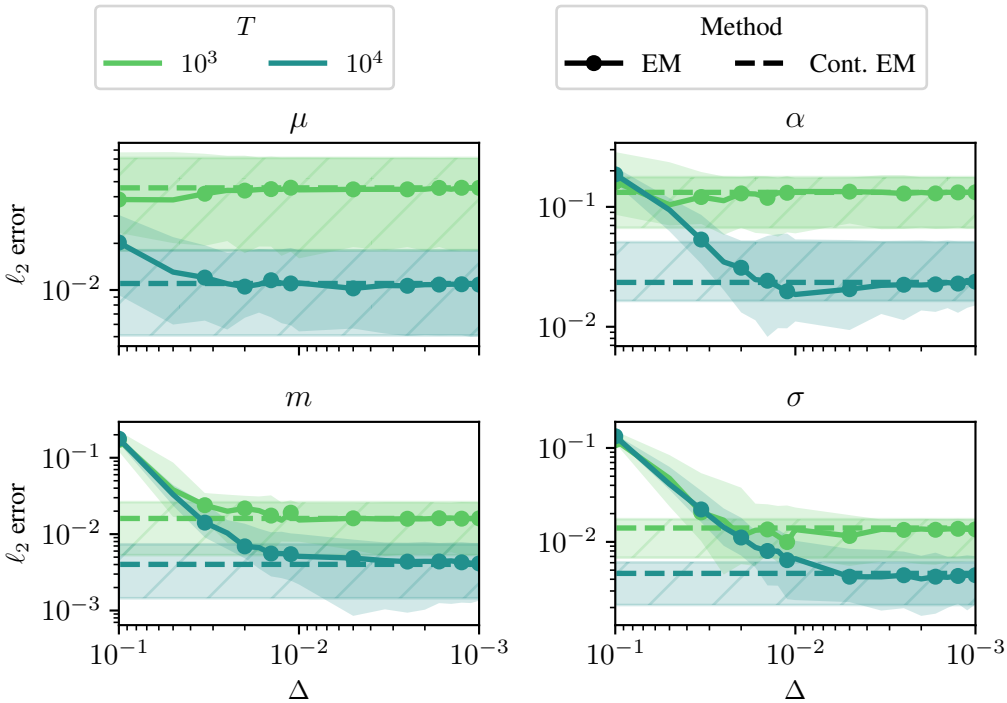

Figure B.8. Median and interquartile error bar of the ℓ_2 norm between true parameters and parameter estimates computed with EM algorithm, continuously and discretely, w.r.t. the stepsize Δ.

The Raised Cosine kernel is defined by:

$$\phi(\cdot) = \alpha \left[1 + \cos\left(\frac{\cdot - u}{\sigma} \pi - \pi \right) \right] \mathbb{I}\left\{ \cdot \in [u;\ u + 2\sigma] \right\}.$$

The parameters to estimate are μ, α, u and σ. The Truncated Exponential kernel of decay parameter $\gamma \in \mathbb{R}_+$, with fixed support $[a\,, b] \subset \mathbb{R}^+$, $b > a$ is defined as $\phi(\cdot) = \alpha\kappa(\cdot), \alpha \geq 0$ with

$$\kappa(\cdot) \coloneqq \kappa\left(\cdot; \gamma, a, b\right) = \frac{h\left(\cdot\right)}{H\left(b\right) - H\left(a\right)} \mathbf{1}_{a \leq \cdot \leq b},$$

where here h (resp. H) is the probability density function of parameter γ (resp. cumulative distribution function) of the exponential distribution. The parameters to estimate are μ, α and γ.

Estimation results (median and 20-80% quantiles) are displayed in Figure B.9 and confirm the conclusion presented in Section 3.1 about the consistency of the discretization for FaDIn. In addition, we display the quadratic error for each parameter separately in Figure B.10 for the Truncated Gaussian, Figure B.11 for the Raised Cosine and Figure B.12 for the Truncated Exponential kernels.

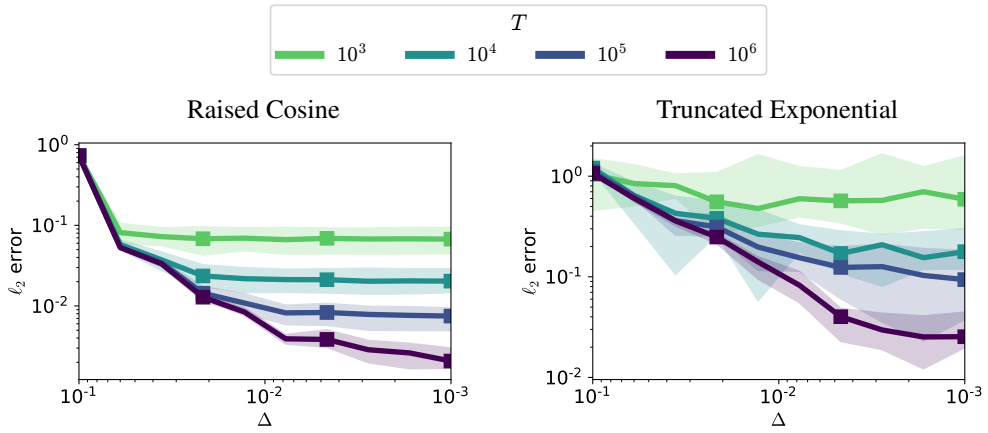

Figure B.9. Comparison of the influence of the discretization on the parameter estimation of FaDIn for a Raised Cosine kernel (left) and an Exponential kernel (right) w.r.t. the stepsize of the grid Δ.

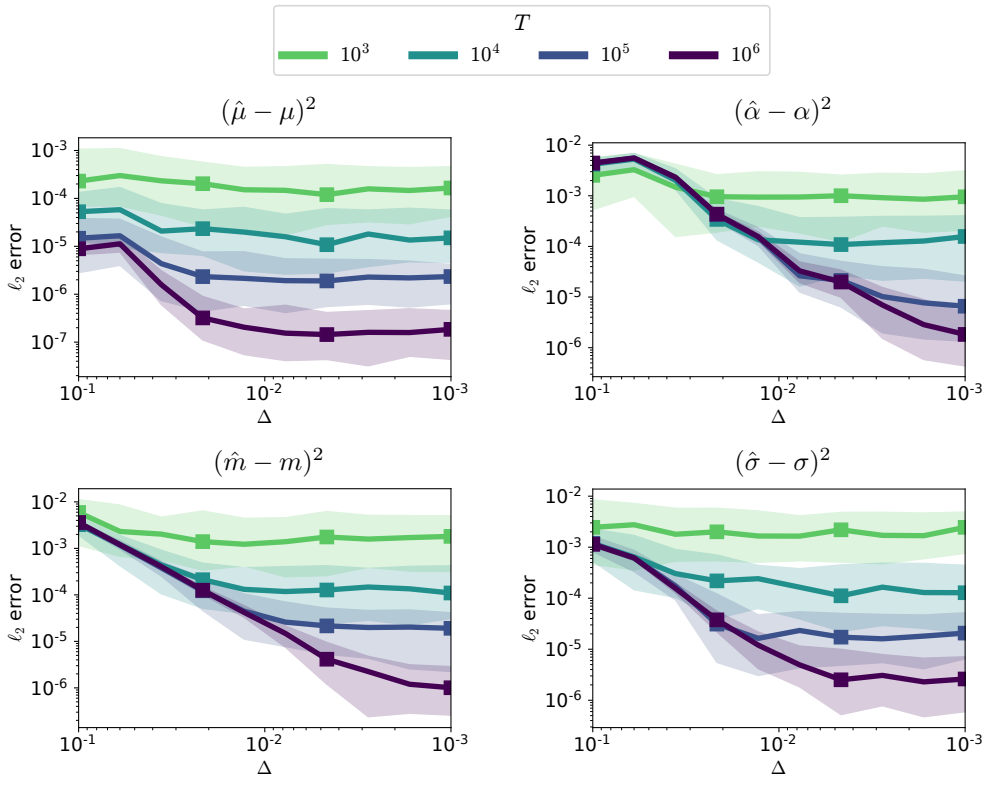

Figure B.10. Error on parameters for the Truncated Gaussian kernel as a function of T and Δ.

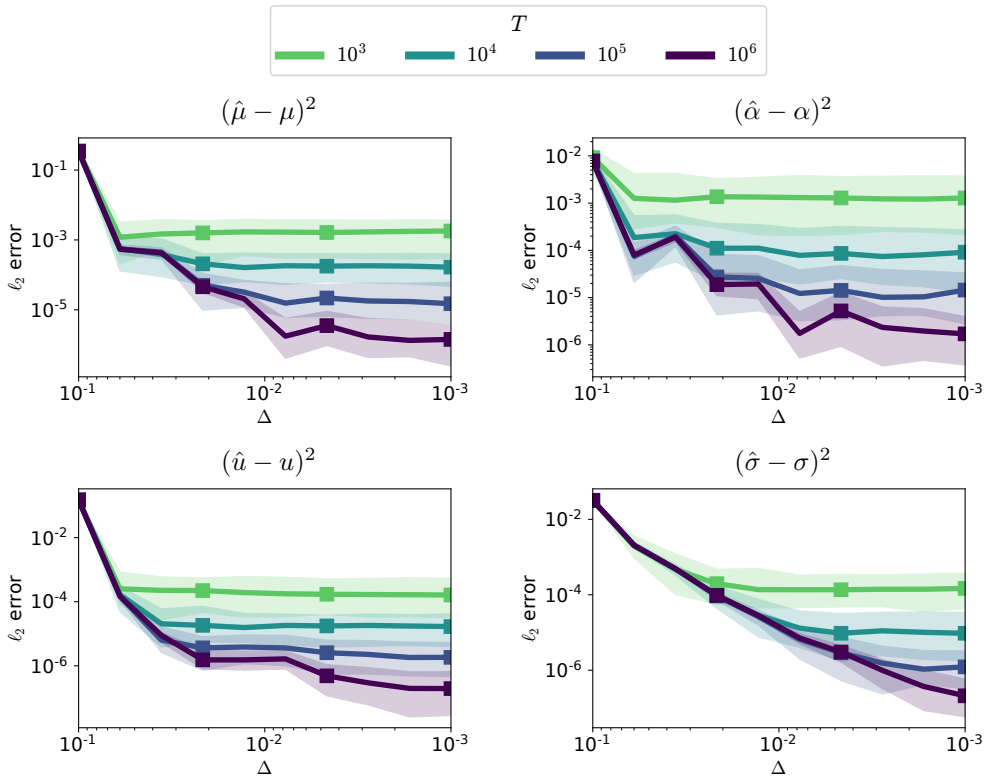

Figure B.11. Error on parameters for the Raised Cosine kernel as a function of T and Δ.

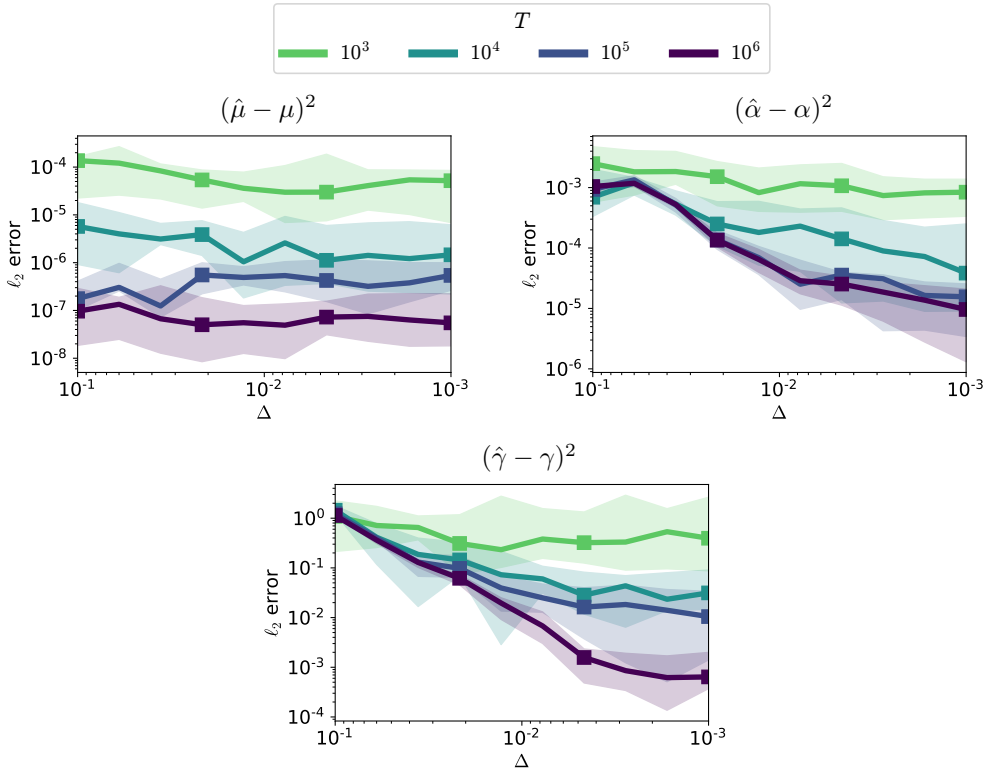

Figure B.12. Error on parameters for the Truncated Exponential kernel as a function of T and Δ.

B.6. Other experiments on real data

After detailing the procedure we use in MEG experiments and adding extra information about used datasets, we provide a second experiment on the somatosensory dataset.

Background on CDL The objective of CDL is to decompose a signal as the convolution between a translationally invariant pattern called atom and its sparse activation vector (Grosse et al., 2007). To do this, we minimize the following objective function:

$$\min_{D_k, z_k^n} \sum_{n=1}^{N} \frac{1}{2} \left\| X^n - \sum_{k=1}^{K} z_k^n * D_k \right\|_F^2 + \lambda \sum_{k=1}^{K} \|z_k^n\|_1 \,, \text{ s.t. } \|D_k\|_F^2 \leq 1 \text{ and } z_k^n \geq 0$$

where $\{X^n\}_{n=1}^N \subset \mathbb{R}^{P \times T}$ is the observed signals, $\{D_k\}_{k=1}^K \subset \mathbb{R}^{P \times L}$ is the dictionaries of atoms, $\{z_k^n\}_{k=1}^K \subset \mathbb{R}^{\widetilde{T}}$ the sparse activations associated with X^n, $\widetilde{T} = T - L + 1$, and $\lambda > 0$ the regularization parameter. Here $\|\cdot\|_F$ stands for the Frobenius norm.

In the application to M/EEG signals, we add a rank-1 constraint on the dictionary to account for the physics of the signals (Dupré la Tour et al., 2018): $D_k = u_k {v_k}^\top \in \mathbb{R}^{P \times L}$, where $u_k \in \mathbb{R}^P$ is the pattern over the channels (sensors) and $v_k \in \mathbb{R}^L$ the pattern over time. The new minimization problem is as follows:

$$\min_{u_k, v_k, z_k^n} \sum_{n=1}^{N} \frac{1}{2} \left\| X^n - \sum_{k=1}^{K} z_k^n * \left(u_k v_k^\top\right) \right\|_F^2 + \lambda \sum_{k=1}^{K} \|z_k^n\|_1 \,, \text{ s.t. } \|u_k\|_2^2 \leq 1, \|v_k\|_2^2 \leq 1 \text{ and } z_k^n \geq 0$$

The optimization is done by block coordinate descent, alternating the optimization over atoms and activations. Figure B.13 presents in a schematic way the functioning of CDL on MEG data.

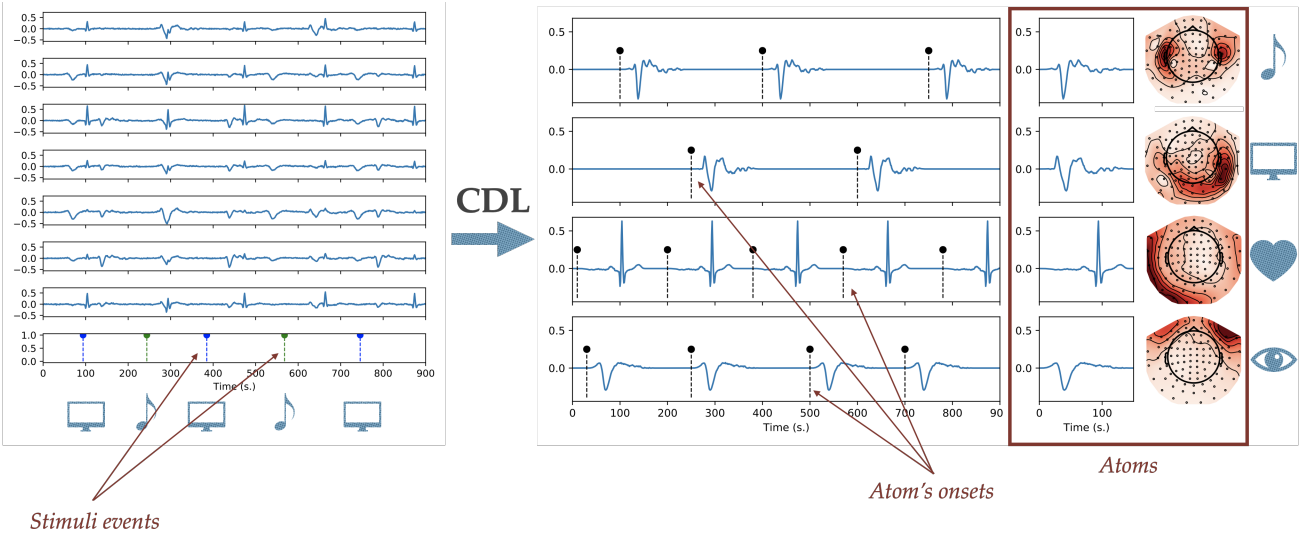

Figure B.13. Schematic operation of the CDL on MEG signals. Raw MEG signals alongside timestamps of external stimuli of type visual and auditory (left). CDL output composed of a set of spatio-temporal atoms alongside their respective onsets (right). One may claim to associate each atom to a physical phenomenom, *i.e.,* heartbeat or eye blink artifact, auditory or visual neural response.

Extra information about used datasets Table B.1 presents the main information related to real MEG datasets that we used, both available with the `MNE` Python package (Gramfort et al., 2013; 2014). Regarding the *sample* dataset, as mentioned before, four external stimuli are presented to the subject during the MEG recording session: auditory left and right and visual left and right. Each type of stimulus leads to a so-called "deterministic" point process, where each event denotes the exact time the stimulus was presented to the subject. Once the CDL was applied, and 40 atoms of

Dataset	# Atoms	Duration of Atoms (s.)	# Atom's events	# Drivers	# Driver's events	Sequence length (min.)
sample	40	1	≈ 401.025	4	≈ 72.25	4.6
somatosensory	20	0.53	10408	1	111	15

Table B.1. Statistics of each dataset. $\approx N$ denotes that N is the average number.

duration 1 s were extracted from the signal, a quick visual inspection of the atoms revealed that, among atoms that could be linked to audio-visual stimuli, there were mostly bimodal atoms. Thus, this observation led us to consider both auditory (resp. visual) stimuli as one, and the two corresponding point processes were merged. For the somatosensory dataset, as previously mentioned, 20 atoms of duration 0.53 seconds are extracted from the MEG signal corresponding to 15 minutes of recording during which the single subject has received 111 stimulations of his left median nerve at the hand level. For both datasets, intensities functions for the EM with a Truncated Gaussian kernel were obtained similarly as Allain et al. (2021), always between one atom's point process and the considered stimuli. For the intensities with a Raised Cosine kernel, however, they were obtained using the method presented in this paper, with a grid discretization Δ equal to data re-sampling rate of 150 Hz (*i.e.,* $\Delta = 1/150$). Indeed, setting a Δ smaller than the discretization imposed by the data would not lead to better estimation. Finally, the intensities estimated with the non-parametric (NP) method were obtained using `Tick` Python package (Bacry et al., 2017), with the same grid discretization parameters to have accurate comparisons.

Experiment on somatosensory dataset Figure B.14 presents results on three atoms estimated from the *somato* dataset. All three atoms elicit classical induced responses and have waveforms with a prototypical μ-shape (sharp trough) (Hari, 2006). Remark that in the *somato* paradigm, the subject receives only one type of external stimulus. Similarly, as in Figure 3, for each atom, the intensity related to the stimulus is learned with a non-parametric kernel (NP) and two kernel parametrizations: Truncated Gaussian (TG) and Raised Cosine (RC). The non-parametric kernel cannot characterize the link between stimulus and neural response.

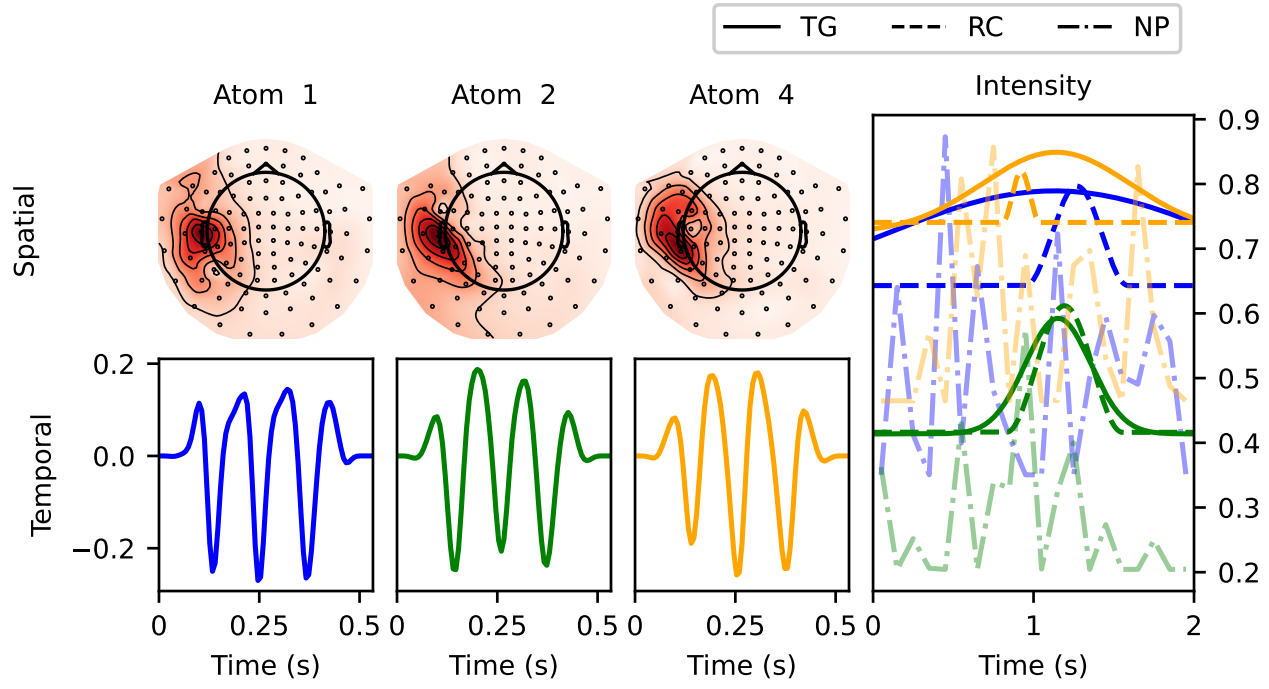

Figure B.14. Spatial and temporal patterns of three μ-wave atoms from *somato* dataset, and their respective estimated intensity functions following a stimulus (cue at time = 0 s), for somatosensory stimuli with non-parametric kernel (NP) and two parametrized kernels: Truncated Gaussian (TG) and Raised Cosine (RC).