Neural Hawkes: Non-Parametric Estimation in High Dimension and Causality Analysis in Cryptocurrency Markets

Timothée Fabre[1,2] and Ioane Muni Toke[1]

[1]Laboratoire MICS, CentraleSupélec, Université Paris-Saclay, France.
[2]Sun Zu Lab, Paris, France.

January 9, 2024

Abstract

We propose a novel approach to marked Hawkes kernel inference which we name the moment-based neural Hawkes estimation method. Hawkes processes are fully characterized by their first and second order statistics through a Fredholm integral equation of the second kind. Using recent advances in solving partial differential equations with physics-informed neural networks, we provide a numerical procedure to solve this integral equation in high dimension. Together with an adapted training pipeline, we give a generic set of hyperparameters that produces robust results across a wide range of kernel shapes. We conduct an extensive numerical validation on simulated data. We finally propose two applications of the method to the analysis of the microstructure of cryptocurrency markets. In a first application we extract the influence of volume on the arrival rate of BTC-USD trades and in a second application we analyze the causality relationships and their directions amongst a universe of 15 cryptocurrency pairs in a centralized exchange.

Keywords – Hawkes process, Non-parametric estimation, Physics-informed neural network, Cryptocurrency

Contents

1 Introduction **2**
1.1 Literature review of non-parametric estimation methods . . . 2
1.2 Contribution and organization of the paper . . . 3

2 Moment-based non-parametric neural Hawkes estimation **3**
2.1 Framework and notations . . . 3
2.2 Physics-Informed Neural Networks . . . 5
2.3 Moment-based neural Hawkes estimation procedure . . . 6
2.4 The practitioner's corner . . . 7

3 Numerical validation **9**
3.1 Numerical validation for various shapes of kernels . . . 9
3.2 Convergence rate with respect to the number of observations . . . 15
3.3 Robustness with respect to the choice of hyperparameters . . . 16

4 Application to high-frequency cryptocurrency data **17**
4.1 Data . . . 17
4.2 The impact of volume on the arrival rate . . . 17
4.3 Cryptocurrency spillover: a causality analysis of transactions . . . 18

5 Conclusion **21**

1 Introduction

Hawkes processes are a class of point process that models a sequence of arrivals of events that can cluster over time. Their properties can display self-exciting or self-inhibiting behaviors in the unidimensional case, and mutual-exciting or mutual-inhibiting behaviors for multidimensional processes. In the recent years, their application has been explored in many areas such as earthquakes (Wang et al., 2012; Kwon et al., 2023), neuroscience (Reynaud-Bouret et al., 2013), criminality (Olinde and Short, 2020) and in finance (Muni Toke and Pomponio, 2012; Bacry et al., 2013, 2015; Rambaldi et al., 2017; Gao et al., 2018; Jusselin and Rosenbaum, 2020). A D-variate linear Hawkes process $N = (N^1, \dots, N^D)$ is a counting process with conditional intensity vector $\lambda_t = (\lambda_t^1, \dots, \lambda_t^D)$ such that for any $i = 1, \dots, D$,

$$\lambda_t^i = \mu^i + \sum_{j=1}^{D} \int_{(-\infty,t)} \phi^{ij}(t-s)\,\mathrm{d}N_s^j, \tag{1}$$

where $\mu = (\mu^1, \dots, \mu^D)$ is the baseline intensity which corresponds to the exogenous component and $\Phi = (\phi^{ij})_{1\le i,j\le D}$ is the kernel matrix. For any $i = 1, \dots, D$, $j = 1, \dots, D$, the element ϕ^{ij} is a function that gives the impact each past event has over the instantaneous probability of observing a new event at time t. This interpretability of Hawkes processes justifies their widespread use especially in causality analysis (Xu et al., 2016; Achab et al., 2017). In the common framework where ϕ is positive we say that the processes $N^1, \dots, N^D$ are self- and mutually exciting. The use of Hawkes processes that exhibit kernels with negative parts provides more flexibility and better fits to inhibitory behaviors that are often observed in the data, at the price of a lesser mathematical tractability (Lu and Abergel, 2018; Duval et al., 2022).

In general the estimation of such processes is carried on by specifying a parametric form of the influence kernel and then calibrating the parameters via likelihood maximization (Lee and Seo, 2017; Lu and Abergel, 2018; Bonnet et al., 2023, among recent contributions). Specifying an exponentially decreasing kernel preserves the Markovian property and is therefore a common choice. But even the estimation of exponential kernels can sometimes be tricky — especially in large dimension — since the likelihood is strongly non-convex. Moreover, much more specific forms of Hawkes kernels are required to fit the data in practice. For example, latency effects may cause the kernel to be non-monotonic when dealing with high-frequency financial data. Furthermore, the dynamic of the point process may depend on an additional component such as the past realizations of marks which makes the kernel even more complex to specify. The application of a non-parametric method is thus essential in this case.

1.1 Literature review of non-parametric estimation methods

Non-parametric inference has been explored through different methodologies. An expectation-maximization (EM) algorithm was designed using the cluster representation of Hawkes and Oakes (1974) for univariate kernel estimation (Lewis and Mohler, 2011) and was later extended to the multivariate case (Zhou et al., 2013). Nevertheless this method has three drawbacks which are particularly exacerbated in the context of high-frequency financial data, as outlined in Bacry and Muzy (2016):

- high computational cost: the evaluation of the log-likelihood has a quadratic time complexity in the number of observations ;
- tail-modulated speed of convergence: the EM algorithm slowly converges when the kernel is power-law decaying, and such kernels are often observed in financial data ;
- removal of inhibition effects: as the kernel is constrained to take positive values, inhibitory effects cannot be quantified.

The exploitation of an infinite autoregressive representation has been proposed as well in Eichler et al. (2017). Similarly, the INAR representation of Hawkes processes is used in Kirchner (2017). Note that other kernel estimation methods have been proposed in the literature and either explore least squares estimation (Bacry et al., 2020) or Bayesian inference perspectives (Zhang et al., 2018; Rousseau et al., 2018; Sulem et al., 2021). However, these methods are unsuitable to general multivariate marked kernels. Other methods focus on the estimation of the norm of the kernels rather than their shape as in Achab et al. (2017).

The use of neural networks has been explored in Joseph et al. (2020); Joseph and Jain (2023). The authors train a neural network model that outputs the kernel matrix and aims at maximizing the associated log-

likelihood over event time observations. This procedure solves the problem of high dimension but shares the drawbacks that are directly linked to a Hawkes likelihood optimization, *i.e.* the potentially large number of local minima and a high computational cost.

To overcome these disadvantages, another class of non-parametric methods encompassing moment-based procedures has been made popular in the literature. Based on the Laplace transform of the autocovariance function, the inference of symmetric kernels is developed in Bacry et al. (2012). This methodology is extended in Bacry and Muzy (2016) by relaxing the symmetry assumption. The authors demonstrate a characterization equation that links the kernel matrix of any stationary linear Hawkes process to its second order statistics. The resulting system of equations involves Fredholm integral equations of the second kind that are numerically solved as a Wiener-Hopf system by discretizing the time domain with quadratures.

Although the Wiener-Hopf (WH) method of Bacry and Muzy (2016) seems to provide a convenient framework as both power-laws and inhibitory effects, as well as marked kernels are inferred, its practical application can be laborious as it is prone to the curse of dimensionality when D increases. Solving such a system involves the inversion of a matrix that can be arbitrarily large depending on the dimension of the Hawkes process, on the size of the support of the mark process and on the number of quadrature points. Moreover estimation issues involving oscillatory behaviours have been reported — see for example Cartea et al. (2021) — and the optimal number of quadratures might be highly problem-dependent.

The problem of the Wiener-Hopf approach does not lie in the methodology itself since solving the characterization equation should lead to the desired solution, but rather in the numerical solving procedure. Supported by recent advances in the literature about physics-informed neural networks (PINNs), we propose in this work to infer the solution by training a neural network to solve the characterization integral equation. Originally developed in Psichogios and Ungar (1992); Dissanayake and Phan-Thien (1994); Lagaris et al. (1998) for differential equations and initial/boundary value problems that arise in physics, this idea has gained much more attention in the past years due to the rising availability of computational resources.

1.2 Contribution and organization of the paper

Our main contributions are the following.

- We introduce a novel non-parametric estimation method for the kernel of marked Hawkes processes in high dimension. Our method leverages some of the latest advances in PINNs training to train a robust kernel learner that solves the characterization equation of Bacry and Muzy (2016).
- We provide a general configuration of hyperparameters and network architecture that approximates the solution with excellent accuracy with respect to the characterization equation residual.
- Using our estimation method on high-frequency cryptocurrency data, we extract the impact of volume over the arrival rate of BTC-USD trades and we analyze spillover effects within a high-dimensional universe of cryptocurrency pairs. We define two ratios that quantify the direction of causal relationships.

The structure of this paper is as follows. Section 2 first defines our framework and notations, then introduces the characterization equation of Bacry and Muzy (2016) and finally presents the neural Hawkes non-parametric estimation method. A fully detailed training procedure is provided. Section 3 validates the estimation method on simulated data and provides insights about hyperparameter tuning and robustness. Section 4 finally applies the estimation method to high-frequency cryptocurrency data.

2 Moment-based non-parametric neural Hawkes estimation

2.1 Framework and notations

Let N be a D-multidimensional linear marked Hawkes process such that for $i \in \{1, \dots, D\}$, $(N_t^i)_{t\geq 0}$ has the intensity process $(\lambda_t^i)_{t\geq 0}$ and mark process $(\xi_t^i)_{t\geq 0}$. We assume the non-zero marks are i.i.d. and $\{t, \xi_t^i \neq 0\} = \{t, N_t^i - N_{t^-}^i = 1\}$. We denote by p^i the probability density function of the random variable ξ_t^i conditionally on $\{N_t^i - N_{t^-}^i = 1\}$. We furthermore denote by $\mathcal{X}_i \subset (0, +\infty)$ the domain of p^i.

We denote by $\Phi := (\phi^{ij})_{(i,j)\in\{1,\dots,D\}}$ the kernel matrix of N and by μ^i its baseline intensity and we assume that for any $i = 1,\dots,D$, $j = 1,\dots,D$, ϕ^{ij} takes values in $\mathbb{R}_+$. For any $i \in \{1,\dots,D\}$, the arrival intensity of events of type i at time $t \geq 0$ is

$$\lambda_t^i = \mu^i + \sum_{j=1}^{D} \int_{(-\infty,t)} \phi^{ij}(t-s,\, \xi_s^j)\, \mathrm{d}N_s^j. \tag{2}$$

Let $\|\Phi\| := \left(\|\phi^{ij}\|\right)_{(i,j)\in\{1,\dots,D\}^2}$ be the matrix of L^1-norms of the kernel components, i.e. for all $(i,j) \in \{1,\dots,D\}^2$,

$$\|\phi^{ij}\| := \iint_{(0,+\infty)\times\mathcal{X}_j} \phi^{ij}(s,z)\, p^j(z)\mathrm{d}s\mathrm{d}z. \tag{3}$$

The endogeneity ratio of the process N is

$$\mathcal{R} := \max_{\lambda\in\mathcal{S}(\Phi)} |\lambda|, \tag{4}$$

where $\mathcal{S}(\Phi)$ is the spectrum of the matrix $\|\Phi\|$. The condition $\mathcal{R} < 1$ ensures the stationarity of N and consequently, the vector of first order statistics $\Lambda := \mathbb{E}(\lambda_t) = (\Lambda^i)_{i\in\{1,\dots,D\}}$ writes

$$\Lambda = (\mathbb{I} - \|\Phi\|)^{-1}\, \mu, \tag{5}$$

where $\mathbb{I}$ is the identity matrix of size D.

Let us now define two quantities that will be of interest in our paper. For all $(i,j) \in \{1,\dots,D\}^2$, we call *aggregated time kernel*, denoted φ^{ij}, the expectation of the kernel component (i,j) with respect to the mark distribution :

$$\varphi^{ij}(t) := \mathbb{E}\left(\phi^{ij}\left(t,\xi^j\right)\right) = \int_{\mathcal{X}_j} \phi^{ij}(t,z)\, p^j(z)\mathrm{d}z. \tag{6}$$

Similarly, we call *aggregated mark kernel*, denoted f^{ij}, the norm of the kernel component (i,j) along the time coordinate

$$f^{ij}(x) := \frac{1}{\|\phi^{ij}\|} \int_{(0,+\infty)} \phi^{ij}(s,x)\mathrm{d}s. \tag{7}$$

Note the normalization such that $\mathbb{E}\left(f^{ij}(\xi^j)\right) = 1$. The aggregated kernels are such that if the kernel is multiplicative in the sense that $\phi^{ij}(t,x) = \psi(t)h(x)$ for some functions ψ and h with $\mathbb{E}(h(\xi)) = 1$, then $\psi \equiv \varphi^{ij}$ and $h \equiv f^{ij}$.

The second-order statistics characterization equation of a point process N with intensity defined in Equation (2) is given by the following result.

Theorem 1 (Bacry and Muzy (2016)) *If G is the second-order statistics matrix defined by*

$$G^{ij}(t,x)\mathrm{d}t\mathrm{d}x := \mathbb{E}\left[\mathrm{d}N_t^i \,\middle|\, \mathrm{d}N_0^j = 1,\, \xi_0^j \in [x, x+\mathrm{d}x]\right] - \mathbb{1}_{\{i=j,\, t=0\}}\mathrm{d}x - \Lambda^i \mathrm{d}t\mathrm{d}x, \tag{8}$$

then for all $i \in \{1,\dots,D\}$, the cross-excitation kernels of the i-th event type $\phi^{i\cdot} := (\phi^{ij})_{j\in\{1,\dots,D\}}$ are the solutions of the following D-system of equations

$$G^{ij}(t,x) = \phi^{ij}(t,x) + \sum_{k=1}^{D} \iint_{(0,+\infty)\times\mathcal{X}_k} \phi^{ik}(s,z) H^{kj}(t-s,x,z) p^k(z)\, \mathrm{d}s\mathrm{d}z, \quad 1 \leq j \leq D \tag{9}$$

where

$$H^{kj}(t,x,z) := G^{kj}(t,x)\mathbb{1}_{\{t>0\}} + \frac{\Lambda^k}{\Lambda^j} G^{jk}(-t,z)\mathbb{1}_{\{t<0\}}. \tag{10}$$

The process is fully characterized by this result in the sense that once the kernel matrix is obtained from Equation (9), the baseline intensity μ is deduced using Equation (5).

In practice the mark universe is discretized which turns Equation (9) into a Fredholm equation of the second kind. The Wiener-Hopf method consists in discretizing the time domain using quadratures and inverting the corresponding matrix to deduce the values taken by the solution at the quadrature points.

Thus, the entire kernel matrix Φ is obtained by proceeding to the inversion of D matrices of size $Q \times M \times D$, where Q is the number of quadratures and M the size of the discretized mark universe. For a large Q — 100 to 1000 points —, a reasonably large M — 3 to 10 points — and a large dimension — 10 event types — the inversion becomes both computationally costly and noisy with a potentially high approximation error.

Note that depending on the sign of the second order statistics, inhibitory effects can be observed, *i.e.* a kernel that explores negative values. As discussed in Bacry and Muzy (2016); Rambaldi et al. (2017) this does not invalidate the non-parametric estimation method as long as $\mathbb{P}\left(\lambda_t^i < 0\right)$ is negligible for any $i = 1, \ldots, D$. In this case, the authors have shown the method leads to reliable results.

2.2 Physics-Informed Neural Networks

Assume $f : [0, T] \times \Omega \to \mathbb{R}$ is a solution to a partial differential equation (PDE) with representation

$$\partial_t f + \mathcal{D} f = 0, \quad t \in [0, T], \; x \in \Omega \subset \mathbb{R}, \tag{11}$$

where $\mathcal{D}$ is a differential operator, T the time horizon and Ω the spatial domain. Suppose that some initial and boundary conditions ensure the uniqueness of solution f:

$$f(0, x) = g(x), \quad x \in \Omega, \tag{12}$$

$$\mathcal{B} f(t, x) = 0, \quad t \in [0, T], \, x \in \partial\Omega, \tag{13}$$

where $\mathcal{B}$ is the boundary operator associated to Equation (11) and $\partial\Omega$ is the boundary of Ω.

A data-driven approach to solve this PDE is to specify the solution f as a neural network f_θ where θ is the set of trainable parameters. Since f is assumed continuous on $[0, T] \times \Omega$, the existence of this approximation is guaranteed by the universal approximation theorem (Hornik et al., 1989) provided there are enough hidden units and the activation function is continuous, unbounded. This physics-informed neural network (PINN) specification was formulated in Raissi et al. (2019) and have motivated many researchers in the past years. The idea is to train f_θ such that it minimizes the following loss function

$$\mathcal{L}(\theta, \mathcal{E}) := \frac{1}{N^c} \sum_{n=1}^{N^c} \left|\partial_t f_\theta(t_n, x_n) + \mathcal{D} f_\theta(t_n, x_n)\right|^2 + \frac{1}{N^i} \sum_{n=1}^{N^i} \left|f_\theta(0, x_n^i) - g(x_n^i)\right|^2 + \frac{1}{N^b} \sum_{n=1}^{N^b} \mathcal{B} f_\theta(t_n^b, x_n^b)^2, \tag{14}$$

where $\mathcal{E} := \{\{(t_n, x_n)\}_{1 \le n \le N^c}, \{x_n^i\}_{1 \le n \le N^i}, \{(t_n^b, x_n^b)\}_{1 \le n \le N^b}\}$ are either a reference grid used for validation or randomly generated data points at each iteration of a gradient descent algorithm. The notations N^c, N^i and N^b hold respectively for the number of collocation points, initial condition points and boundary condition points. The gradients and other derivatives are computed via automatic differentiation.

At the beginning of its development, the method presented a lot of training issues that have been tackled since, encompassing unbalanced gradients (Wang et al., 2021; Deguchi and Asai, 2023), the reproduction of time causality (Wang et al., 2022b), the complexity of the PINNs loss landscapes (Krishnapriyan et al., 2021) and the difficulty of optimizing the sum of multiple competitive loss functions (Bischof and Kraus, 2021). Although the literature about PINNs is still burgeoning, several good practices have already emerged and now allow researchers to train robust models.

This data-driven approach can be transposed from PDEs to integral equations (IEs). If f is a solution to an integral equation with representation

$$f + \mathcal{I} f = 0, \quad t \in [0, T], \; x \in \Omega \subset \mathbb{R}, \tag{15}$$

where $\mathcal{I}$ is an integral operator, T the time horizon and Ω the spatial domain, then the loss function of the associated PINN f_θ is written

$$\mathcal{L}(\theta, \mathcal{E}) := \frac{1}{N} \sum_{n=1}^{N} \left|f_\theta(t_n, x_n) + \mathcal{I} f_\theta(t_n, x_n)\right|^2. \tag{16}$$

For example, if f is the solution of the following Fredholm integral equation of the second kind

$$g(t) = f(t) + \int_0^T g(t - s) f(s) \mathrm{d}s, \quad t \in [0, T], \tag{17}$$

where g is a continuous function satisfying $\sup_{0\leq t\leq T}\int_0^T |g(t-s)|\mathrm{d}s < 1$ which ensures the uniqueness of the solution, we write for all $t \in [0,T]$

$$\mathcal{I}f(t) := -g(t) + \int_0^T g(t-s)f(s)\mathrm{d}s. \tag{18}$$

This time, the training of a PINN f_θ that approximates the solution of Equation (15) does not involve the estimation of gradients, but the computation of an integral via a quadrature method. The exploitation of neural networks in such frameworks has been proposed *e.g.* in Effati and Buzhabadi (2012); Zappala et al. (2022); Guan et al. (2022); Yuan et al. (2022). Inspired by the best practices from the literature concerning PINNs applied to PDEs and IEs, we now design a PINN model to solve the Hawkes characterization equation.

2.3 Moment-based neural Hawkes estimation procedure

Let $u_\theta : (t,x) \mapsto (u_\theta^{ij}(t,x))_{1\leq i,j\leq D}$ be a physics-informed neural network with trainable parameters θ. Our goal is to find θ^* that minimizes a loss function $\mathcal{L}$ over the domain of Equation (9). We truncate the time domain and define the kernels for $t \in [0,T]$ instead of $[0,+\infty)$. Without any loss of generality we assume that for all i, $\mathcal{X}_i = \mathcal{X} \subset (0,+\infty)$ and that $\mathcal{X}$ is bounded. In practice, it simplifies the computation of the loss function and we can always extend the sets $\mathcal{X}_i$ such that all D probability density functions p^i share the same domain $\mathcal{X}$.

2.3.1 Loss function

Consider a set of data points $\mathcal{E} := \{(t_n, x_n),\, 1 \leq n \leq N^c\}$ sampled over the domain $(0,T)\times\mathcal{X}$. We assume that the second-order statistics $G^{ij}(t_n,x_n)$ defined at Equation (8) are non-zero for all i,j and n. From the characterization equation of Theorem 1, we define the (i,j)-equation residual at the n-th data point by

$$\varepsilon_n^{ij}(\theta) := G^{ij}(t_n,x_n) - u_\theta^{ij}(t_n,x_n) - \sum_{k=1}^{D}\iint_{(0,T)\times\mathcal{X}} u_\theta^{ik}(s,z)H^{kj}(t_n-s,x_n,z)p_k(z)\,\mathrm{d}s\mathrm{d}z, \tag{19}$$

and define the quadratic loss function $\mathcal{L}$ with

$$\mathcal{L}(\theta,\mathcal{E}) := \sum_{n=1}^{N^c}\sum_{1\leq i,j\leq D}\varepsilon_n^{ij}(\theta)^2. \tag{20}$$

Following the recommendations of Wang et al. (2023) on the training of PINNs, we use a loss weighting procedure to better infer the time causality of the kernel matrix. This weighting procedure was introduced in Wang et al. (2022b). The authors noticed that neural networks may encounter issues in approximating the solution of PDEs over the time component t due to the propensity of the model to minimize all loss components at the same time. They proposed to create temporal weights that will not give importance to a loss component evaluated at any time t unless all the previous components — for all time $s < t$ — are sufficiently small. This can be particularly relevant for Hawkes kernels when the accuracy of estimation at short times is crucial.

We revisit this methodology by introducing a normalized weighting procedure that scales the loss and controls the weights' range of values. Assume the set $\mathcal{E}$ is built such that for all $n \in \{1,\dots,N^c\}$, $t_{n-1} < t_n$ and define the cumulative loss at time t_n:

$$\mathcal{S}_n^{ij} := \sum_{k=1}^{n}\varepsilon_k^{ij}(\theta)^2. \tag{21}$$

We define the (i,j)-temporal weight evaluated at the n-th data point

$$\omega_n^{ij}(\theta) := \begin{cases} 1 & \text{if } n=1, \\ \exp\left\{-\epsilon\dfrac{\mathcal{S}_{n-1}^{ij}}{\mathcal{S}_{N^c}^{ij}}\right\} & \text{otherwise,} \end{cases} \tag{22}$$

where ϵ is a a hyperparameter that controls the minimum value taken by the weights as they all take values in $]e^{-\epsilon}, 1]$. We finally define the weighted loss function as

$$\mathcal{L}^{\omega}(\theta, \mathcal{E}) := \sum_{n=1}^{N^c} \mathcal{L}_n^{\omega}(\theta), \tag{23}$$

where

$$\mathcal{L}_n^{\omega}(\theta, \mathcal{E}) := \sum_{1 \leq i,j \leq D} \omega_n^{ij}(\theta)\, \varepsilon_n^{ij}(\theta)^2. \tag{24}$$

2.3.2 Sampling procedure

We now detail the sampling procedure of the set of data points $\mathcal{E} := \{(t_n, x_n),\, 1 \leq n \leq N^c\}$. Our method uses a generic randomized sampling. The model did not show any significant improvement by using residual-based adaptive sampling such as the ones described in Wu et al. (2023) as standard sampling already provides accurate estimates of the kernel matrix.

As our practical applications are usually done for short time scales – such as high-frequency data – we are particularly concerned by the inference of kernels that may have large variations for $t < 10^{-3}$, corresponding to a millisecond. Therefore we adapt the randomized time sampling by generating more data points for small times. Let S be the proportion of time points taking values under a short-time scale threshold $h \ll T$. For a training set of size N^c, we sample $\lfloor SN^c \rfloor$ data points using a log-uniform law on the interval $(0, h)$, and $N^c - \lfloor SN^c \rfloor$ data points using a uniform law on the interval (h, T), $\lfloor . \rfloor$ being the floor function. In the rest of the paper, we set $S = 0.3$.

Finally, an integer uniform grid sampling is used for the mark data points.

2.3.3 Architecture and detailed training pipeline

The choice of the neural network architecture can be crucial when designing a PINN. For example in Wang et al. (2021), the authors propose a novel architecture to help reducing gradient flow issues during training. Furthermore, neural networks inspired from the long-short term memory cell may lead to better performance as argued in Sirignano and Spiliopoulos (2018). In the rest of the paper, we use the Deep Galerkin Method (DGM) network architecture introduced in Sirignano and Spiliopoulos (2018).

As discussed in Bacry and Muzy (2016), it is noteworthy that the system of D^2 Fredholm integral equations can be formulated as D independent systems of D Fredholm integral equations. Thus, the approximated solution can be splitted into D neural networks each predicting an output of size D — the kernels $\phi^{i\cdot}$ or in over words, the i-th row of the kernel matrix Φ. This splitting procedure involves the training of D models and presents two great benefits. The first one is the suitability to parallel computing leading to a much faster training. Secondly, separately minimizing D loss functions instead of minimizing their sum will much likely lead to an improved overall performance. To this extent, we use this methodology to learn the entire kernel matrix Φ. The loss functions of Equations (20) and (24) are decomposed as D loss functions such that for all i,

$$\mathcal{L}_i(\theta, \mathcal{E}) := \sum_{n=1}^{N^c} \sum_{j=1}^{D} \varepsilon_n^{ij}(\theta)^2, \tag{25}$$

and

$$\mathcal{L}_{i,n}^{\omega}(\theta, \mathcal{E}) := \sum_{j=1}^{D} \omega_n^{ij}(\theta)\, \varepsilon_n^{ij}(\theta)^2. \tag{26}$$

The detailed training pipeline is described in Algorithm 1.

2.4 The practitioner's corner

The aim of this section is to give a practical advice regarding training and potential questions that arise during the implementation step.

Algorithm 1 Training procedure of the moment-based neural Hawkes

1. Compute statistics over the time and mark domains that will be used for input normalization.
2. For $i = 1$ to D:
   (a) Represent the solution of the i-th characterization equation of Theorem 1 by a DGM neural network u^i_θ and initialize its weights θ with a Glorot scheme.
   (b) Set the number of epochs E, the size N^c of the training set $\mathcal{E}$, the batch size B such that $N^c = 0\,[B]$, a decreasing sequence of learning rates $(\gamma_e)_{e\geq 1}$ and a random sampling scheme, *e.g.* the procedure described above. We use $\gamma_e = \gamma_0 100^{-\frac{e}{E}}$.
   (c) For $e = 1$ to E:
      i. Sample a training set $\mathcal{E} = \{(t_n, x_n), 1 \leq n \leq N^c\}\}$ and a validation set $\bar{\mathcal{E}}$.
      ii. Apply a standardization and log-scaling to the data points.
      iii. Over the elements of $\mathcal{E}$: compute the equation residuals $(\varepsilon^{ij}_n(\theta))_{1\leq j\leq D}$ and the temporal weights $(\omega^{ij}_n(\theta))_{1\leq j\leq D}$.
      iv. For $s = 1$ to $\frac{N^c}{B}$:
         - Using Equation (26), update the weights via a mini-batch gradient descent

$$\theta \leftarrow \theta - \gamma_e \frac{1}{B} \sum_{n=(s-1)B+1}^{sB} \nabla_\theta \mathcal{L}^\omega_{i,n}(\theta). \tag{27}$$

      v. Using Equation (25), compute the unweighted validation loss $\mathcal{L}_i(\theta, \bar{\mathcal{E}})$.
   (d) **Outputs:** An estimate θ^i of the optimal weights for the i-th model and a list of validation losses over the epochs.
3. **Outputs:** D models $(u^i_{\theta^i})_{1\leq i\leq D}$, the i-th model representing the i-th row of the kernel matrix.

On the non-dimensionalization of the problem. In order to standardize the time and mark inputs of the neural network, we recommend the following transformations.

- **Time**: Apply a logarithmic transformation $t \mapsto \log_p t$, with $p = 10$ for example. This allows the network to better infer kernels that have large fluctuations over small time windows. In practice, as this is commonly observed in financial microstructure when studying the order flow at short time scales, we advise the practitioners to use this scaling by default. Furthermore, our numerical experiments have shown significant improvements by using this transformation.
- **Mark**: A good standardization methodology for the marks should depend on the data generating process. By default, we applied a z-score transformation $x \mapsto \frac{x-m}{\sigma}$ where m and σ are respectively the empirical average and empirical standard deviation computed over a sample of the training data.

On the activation functions of the DGM layer. We recommend using ReLU activation functions in place of hyperbolic tangent in order to improve the approximation quality. We found that discontinuous kernels are much better reproduced with this configuration.

On the estimation of the integral. The integral component of Equation (29) is numerically estimated with time quadratures. Although a large number of quadratures — larger than 1000 — may provide slightly better results, we did find that 500 quadratures were amply sufficient to obtain a good performance. One should keep in mind that the training procedure becomes costly if this number is too large as D^2M integrals need to be computed at each evaluation of the loss. In the numerical experiments, we use a quadrature scheme with a logarithmic grid.

On the choice of the loss function. As pointed out in Wang et al. (2022a), the L^∞-norm can be chosen instead of the L^2-norm in the training process as it may lead to more accurate solutions. We explored this idea and finally chose to stick to the original norm as the validation results presented in the next section did not improve with this change.

On the estimation of the second order statistics. A correct choice of the time grid over which the statistics G are estimated is crucial. As the number of observations increases, we can expect to get a more precise estimation and thus we can set a finer mesh. As the choice of the sample-grid may be highly problem-dependent, we will use a grid that is well suited for our scope of application. To this extent,

we shall use a scheme inspired by the work of Bacry et al. (2016) and Rambaldi et al. (2017) for the estimation of kernels with short time horizon. It consists in a linear scheme for very short times up to an intermediate step h, and then a logarithmic scheme from h to the time horizon T. This allows to account for the structure of high-frequency data as explained by the authors. We thus use the following rule. We set $h < T$, the number of points in the linear part n_{lin} and the number of points in the logarithmic part n_{log} and estimate the statistics on the following grid

$$\left[t_{\min}, t_{\min} + \frac{1}{n_{\text{lin}}}(h - t_{\min}), t_{\min} + \frac{2}{n_{\text{lin}}}(h - t_{\min}), \ldots, h, h\left(\frac{T}{h}\right)^{\frac{1}{n_{\text{log}}}}, h\left(\frac{T}{h}\right)^{\frac{2}{n_{\text{log}}}}, \ldots, T\right]. \tag{28}$$

On the continuity of the kernel at time T^-. Since the kernel matrix is defined for $t \leq T$, we may impose a continuity condition $u(T, x) = 0$, for all $x \in \mathcal{X}$ depending on the practical application. In the PINN literature, this boundary constraint is simply introduced to the training by adding an additional error term to the quadratic loss. A regularization weight ω_c is then applied to this loss term.

On kernels with different orders of magnitude. Approximation issues may arise when the values taken by each kernel component of the matrix explore very different orders of magnitude. In this case, the kernels with high values will be correctly estimated but the relative error made over the other kernels might be high. This inference problem is similar to the causality issue that was studied in Wang et al. (2022b) in the sense that the neural network averages the errors made over the statistics G. Thus, statistics that take high values will overshadow the ones taking small values. An idea to mitigate this problem is to use a weighted version of the loss function defined in Equation (29). To this extent, we propose the following weighted loss function

$$\mathcal{L}^{\omega}(u, G, \mathcal{E}) := \sum_{n=1}^{N^c} \sum_{1 \leq i,j \leq D} \zeta_{ij}(x_n) \omega_n^{ij}(\theta) \varepsilon_n^{ij}(\theta)^2, \tag{29}$$

with

$$\zeta_{ij}(x) := \frac{\left(\int_{(0,T)} \left|G^{ij}(t,x)\right| \mathrm{d}t\right)^{-1}}{\sum_{1 \leq i',j' \leq D} \left(\int_{(0,T)} \left|G^{i'j'}(t,x)\right| \mathrm{d}t\right)^{-1}}, \quad 1 \leq i,j \leq D. \tag{30}$$

This weighting procedure puts more weight to kernels that exhibit smaller absolute second-order statistics G with respect to other kernels. Obviously, this technique becomes irrelevant if all the components of the kernel matrix Φ take similar values.

On the training time. As the Wiener-Hopf approach, our non-parametric method has the benefit of having a computation time that is independent from the number of observations. In fact, the number of observations influences the computation time of second order statistics but not the training time itself. This strongly contrasts with common likelihood approaches.

3 Numerical validation

In this section we present several numerical experiments conducted over a wide range of simulated data. We show that the moment-based non-parametric neural Hawkes estimation method detailed in Section 2 provides accurate kernel estimates in various configurations of dimensions, kernel types and mark distributions. We measure the rate of convergence of the method with respect to the sample size and provide some elements regarding the robustness of the method with respect to hyperparameters.

3.1 Numerical validation for various shapes of kernels

We set up four numerical experiments. In each experiment, we simulate a sequence of timestamps generated by a multidimensional linear Hawkes process with kernel configurations that may appear in financial applications, e.g., with slowly decreasing tails and inhibition patterns. For each simulated event,

a mark ξ is independently generated from a discrete uniform law taking values in $\mathcal{X} = \{1, \dots, M\}$ where M is specified for each experiment.

More precisely, we use in all experiments a multiplicative kernel matrix Φ with components satisfying

$$\phi^{ij}(t,x) := \varphi^{ij}(t) f^{ij}(x), \quad 1 \le i,j \le D,\ t \in (0,+\infty),\ x \in \mathcal{X}. \tag{31}$$

Each experiment tests a specific form of aggregated time kernels φ^{ij}, namely exponentially decreasing kernels, slowly-decreasing kernels, kernels with latencies and kernels with inhibition.

As for marks, aggregated mark kernels can be either a constant mark kernel

$$f_0(x) = 1, \tag{32}$$

or a linear mark kernel

$$f_1(x) = \frac{2}{M+1}\, x, \tag{33}$$

or a quadratic mark kernel

$$f_2(x) = \frac{6}{(M+1)(2M+1)}\, x^2. \tag{34}$$

In each case $\mathbb{E}\left(f(X)\right) = 1$. We train a neural network following Algorithm 1 with the set of hyperparameters displayed in Table 1 and we estimate the kernel over the truncated time domain $(0, T)$.

Table 1: Hyperparameters configuration in the numerical experiment

Neurons	DGM depth	Learning rate	Quadratures	Batch size	Training size	Validation size	Epochs
64	1	10^{-3}	250	8	1024	128	1000

Exponentially decreasing kernel. Exponentially decreasing functions are amongst the most popular kernels found in the literature. The Markov property that results from such a specification provides mathematical tractability and a greatly reduced computation cost when simulating. The non-parametric estimation of these kernels from simulated data constitutes the first step of our validation procedure.

We set $D = 4$, $M = 10$ and simulate 10^7 events of a linear marked Hawkes process with kernel functions

$$\phi^{ij}(t,x) := \alpha_{ij} e^{-\beta_{ij} t} f^{ij}(x), \quad 1 \le i,j \le D,\ t \in [0,+\infty),\ x \in \mathcal{X}. \tag{35}$$

The parameters of the time component and the mark kernels are given in Table 2. The second order statistics are estimated using the grid of Equation (28) with parameters $h = 0.1$, $n_{\text{lin}} = 10$, $n_{\log} = 50$. The baseline intensity is $\mu = [0.5, 0.25, 0.75, 0.15]'$.

Table 2: *Exponentially decreasing kernel* — Parameter configuration of the simulation

(a) α

α_{ij}	1	2	3	4
1	1.5	1	2	0.75
2	1	2	2	1
3	1	1	2	1.5
4	2	1	2	0.5

(b) β

β_{ij}	1	2	3	4
1	8	5	10	8
2	5	15	8	5
3	8	10	8	8
4	5	5	8	4

(c) f

f^{ij}	1	2	3	4
1	f_1	f_2	f_1	f_1
2	f_1	f_2	f_2	f_0
3	f_0	f_0	f_2	f_0
4	f_2	f_1	f_2	f_1

The fits of aggregated time kernels and mark kernels are displayed in Figure 1 and 2. We observe a great fit for both components and the model is able to reproduce the various mark kernel functions even when $f \equiv 1$.

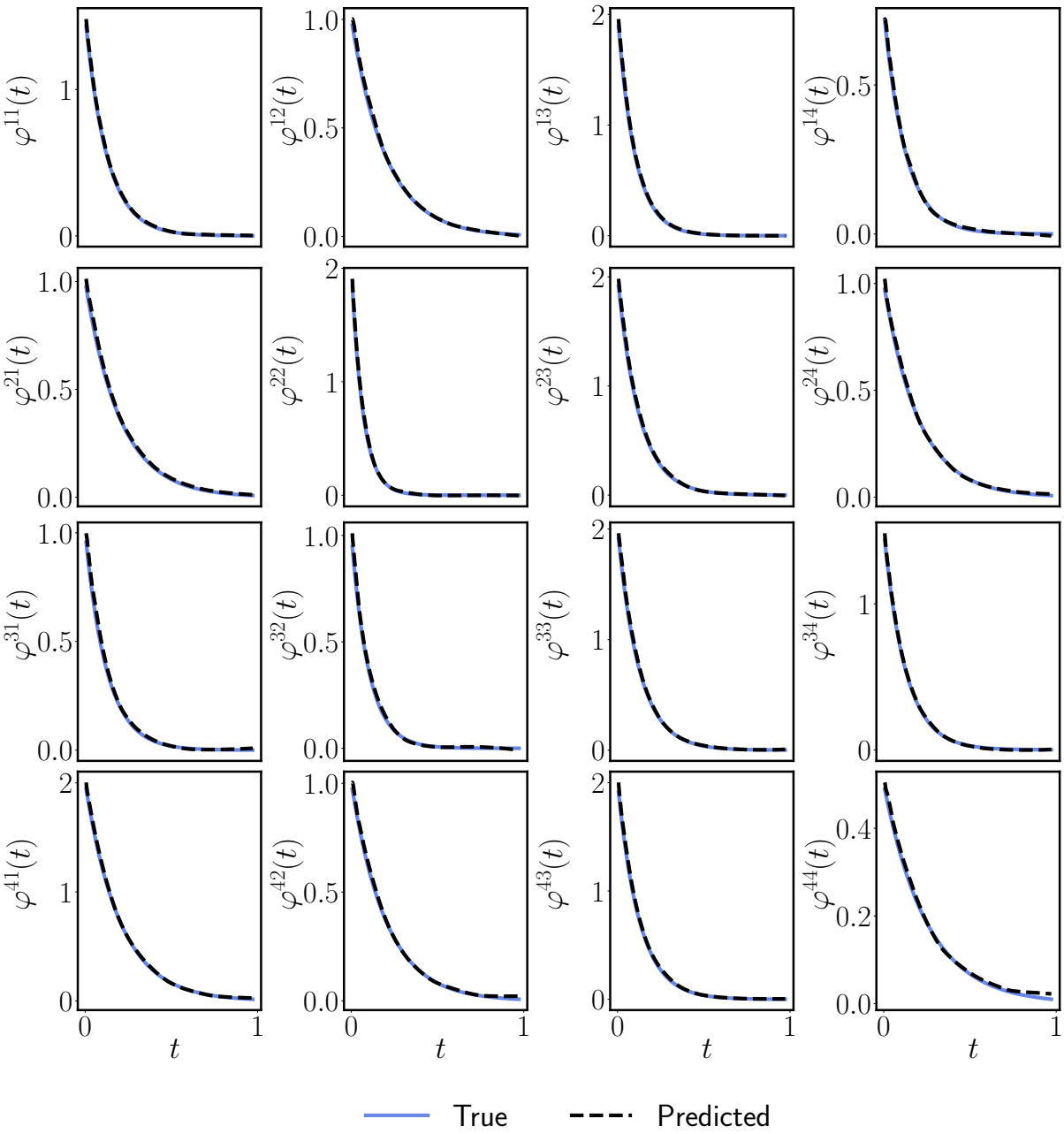


Figure 1: *Exponentially decreasing kernel* — Comparison of the predicted and exact aggregated time kernel matrix $(\varphi^{ij})_{1\leq i,j\leq D}$ defined in Equation (6).

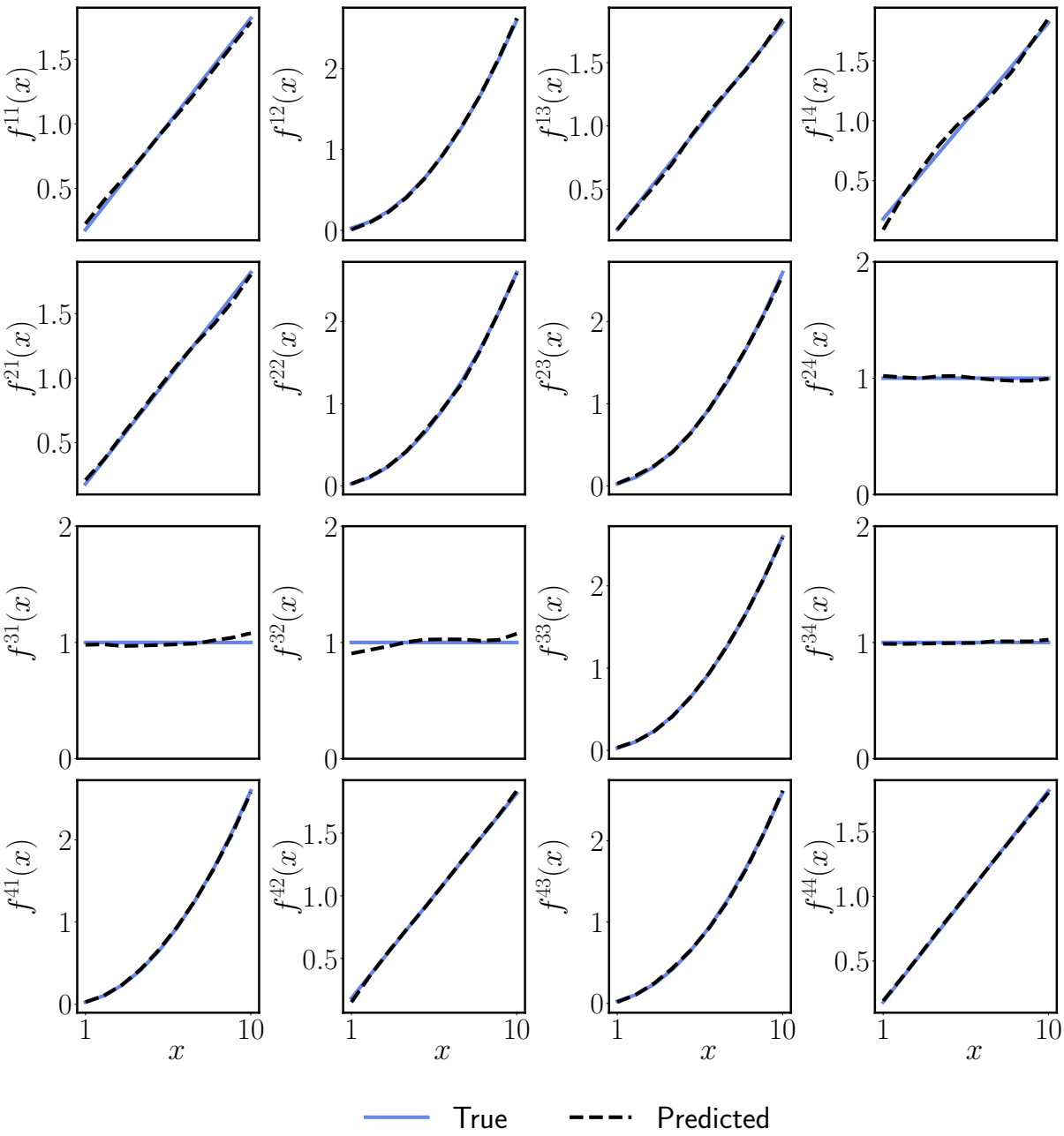
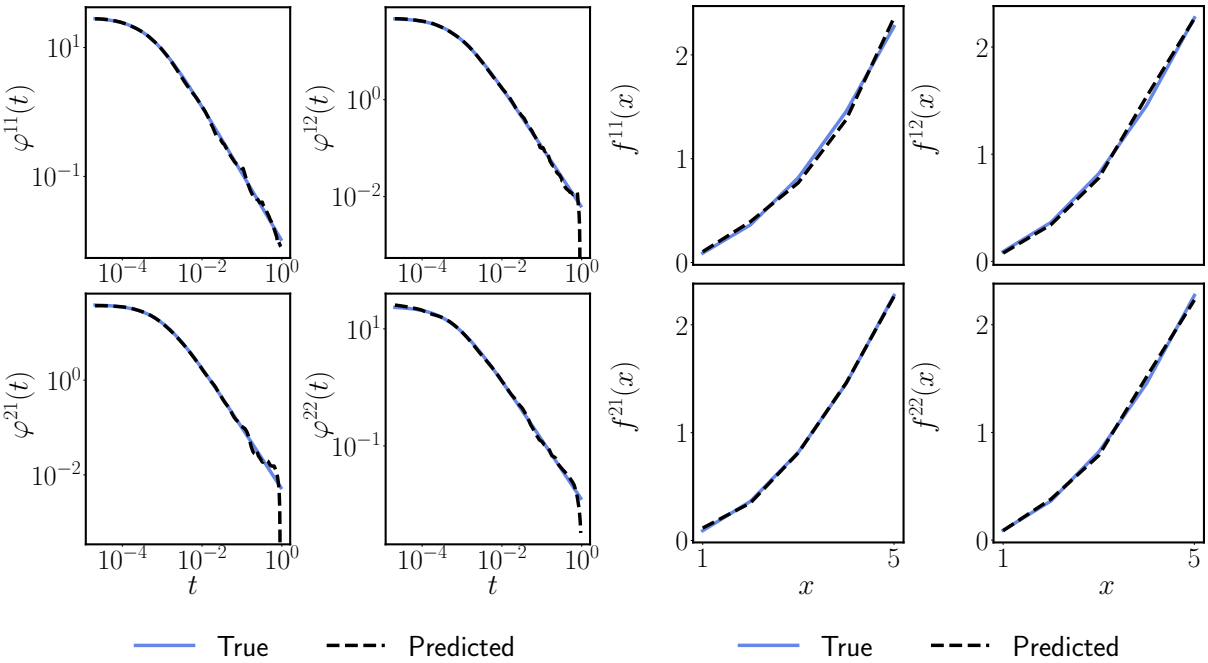


Figure 2: *Exponentially decreasing kernel* — Comparison of the predicted and exact aggregated mark kernel matrix $(f^{ij})_{1\leq i,j\leq D}$ defined in Equation (7).



(a) Time kernel $(\varphi^{ij})_{1\leq i,j\leq D}$

(b) Mark kernel $(f^{ij})_{1\leq i,j\leq D}$

Figure 3: *Slowly decreasing kernel* — Comparison of the predicted and exact aggregated kernel matrices defined in Equations (6) and (7).

Slowly decreasing kernel. Slowly decreasing kernels are often observed on high-frequency financial data after applying non-parametric estimation methods for Hawkes processes — see Bacry et al. (2012), Bacry et al. (2016), Fosset et al. (2022). Identifying the rates of decrease over time of self-excitation and cross-excitation components gives a better understanding of the market microstructure and the memory of order flow.

We set $D = 2$, $M = 5$ and simulate 2.10^6 events of a linear marked Hawkes process with kernel functions

$$\phi^{ij}(t,x) := \alpha_{ij}(\gamma_{ij}+t)^{-\beta_{ij}} f^{ij}(x), \quad 1 \leq i,j \leq D,\ t \in [0,+\infty),\ x \in \mathcal{X}. \tag{36}$$

with $f^{ij}(x) = f_2(x)$. The parameters of the time component are given in Table 3. The second order statistics are estimated using the grid of Equation (28) with parameters $h = 10^{-3}$, $n_{\text{lin}} = 25$, $n_{\log} = 75$. The baseline intensity is $\mu = [0.05, 0.05]'$.

Table 3: *Slowly decreasing kernel* — Parameter configuration of the simulation

(a) α

α_{ij}	1	2
1	0.01	0.006
2	0.005	0.012

(b) β

β_{ij}	1	2
1	1.05	1.25
2	1.3	1.025

(c) γ

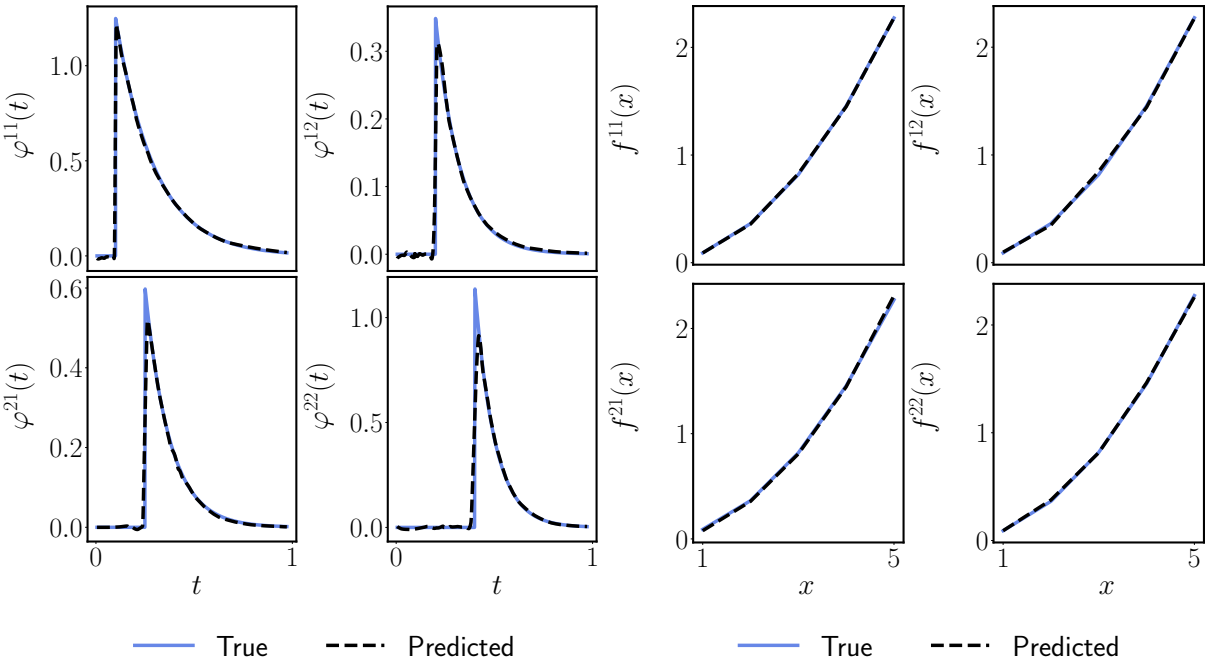
γ_{ij}	1	2
1	0.0005	0.00075
2	0.001	0.0006

The fits of aggregated time kernels and mark kernels are displayed in Figure 3. The model reproduces the power law over 5 decades and fits the mark component as well.

Delayed exponentially decreasing kernel. In practice, trading algorithms are exposed to transmission delays and latency. This phenomenon passes through to the formation of lagged kernels — see for example Carreira (2021).

We set $D = 2$, $M = 5$ and simulate 2.10^6 events of a linear marked Hawkes process with kernel functions

$$\phi^{ij}(t,x) := \alpha_{ij} e^{-\beta_{ij}(t-\ell_{ij})} f^{ij}(x)\, \mathbb{1}_{\{t\geq \ell_{ij}\}}, \quad 1 \leq i,j \leq D,\ t \in [0,+\infty),\ x \in \mathcal{X}. \tag{37}$$



(a) Time kernel $(\varphi^{ij})_{1\leq i,j\leq D}$ (b) Mark kernel $(f^{ij})_{1\leq i,j\leq D}$

Figure 4: *Delayed exponentially decreasing kernel* — Comparison of the predicted and exact aggregated kernel matrices defined in Equations (6) and (7).

with $f^{ij}(x) = f_2(x)$.

The parameters of the time component are given in Table 4. The second order statistics are estimated using the grid of Equation (28) with parameters $h = 0.1$, $n_{\text{lin}} = 10$, $n_{\log} = 50$. The baseline intensity is $\mu = [0.05, 0.05]'$.

Table 4: *Delayed exponentially decreasing kernel* — Parameter configuration of the simulation

(a) α

α^{ij}	1	2
1	1.25	0.35
2	0.6	1.15

(b) β

β^{ij}	1	2
1	5	8
2	8	10

(c) ℓ

ℓ^{ij}	1	2
1	0.1	0.2
2	0.25	0.4

The fits of aggregated time kernels and mark kernels are displayed in Figure 4. All different 4 latencies are correctly captured by the model and the mark kernels are correctly estimated as well.

Kernel with inhibition. Inhibition effects have been documented several times in the market microstructure literature - see for example Rambaldi et al. (2017), Lu and Abergel (2018). In this experiment, we combine both latency and inhibition effects in one single kernel.

We set $D = 2$, $M = 5$ and simulate 2.10^6 events of a linear marked Hawkes process with kernel functions

$$\phi_{ij}(t,x) := \left(\underline{\alpha}_{ij} e^{-\underline{\beta}_{ij} t} \mathbb{1}_{\{t<\ell_{ij}\}} + \overline{\alpha}_{ij} e^{-\overline{\beta}_{ij}(t-\ell_{ij})} \mathbb{1}_{\{t\geq \ell_{ij}\}}\right) f^{ij}(x), \quad 1 \leq i,j \leq D,\ t \in [0,+\infty),\ x \in \mathcal{X} \tag{38}$$

with $f^{ij}(x) = f_2(x)$ and we set the instantaneous self/cross-impact elements $\underline{\alpha}_{ij}$ and $\overline{\alpha}_{ij}$ such that

$$\text{sign}(\underline{\alpha}_{ij}) = -\text{sign}(\overline{\alpha}_{ij}). \tag{39}$$

The parameters of the time component are given in Table 5. The second order statistics are estimated using the grid of Equation (28) with parameters $h = 0.1$, $n_{\text{lin}} = 25$, $n_{\log} = 75$. The baseline intensity is $\mu = [3, 2.5]'$.

Table 5: *Kernel with inhibition* — Parameter configuration of the simulation

(a) $\underline{\alpha}$

$\underline{\alpha}_{ij}$	1	2
1	1	-0.25
2	-0.2	1.2

(b) $\underline{\beta}$

$\underline{\beta}_{ij}$	1	2
1	3	3
2	2	2

(c) $\overline{\alpha}$

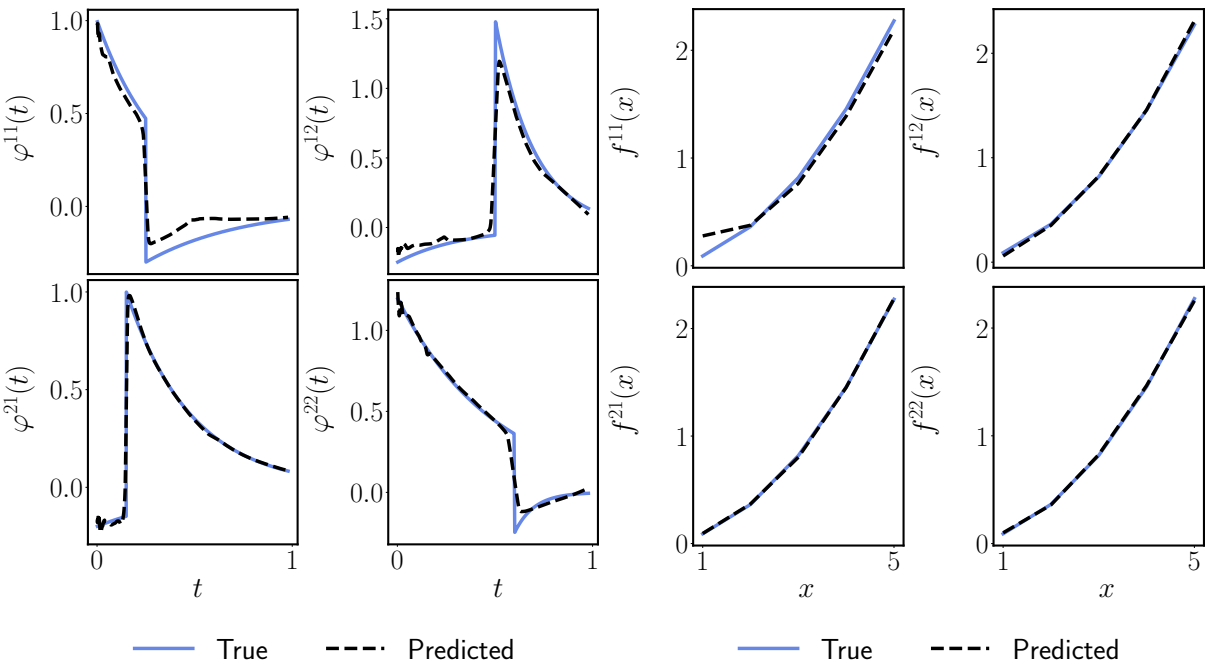
$\overline{\alpha}_{ij}$	1	2
1	-0.3	1.5
2	1	-0.25

(d) $\overline{\beta}$

$\overline{\beta}_{ij}$	1	2
1	2	5
2	3	10

(e) ℓ

ℓ_{ij}	1	2
1	0.25	0.5
2	0.15	0.6



(a) Time kernel $(\varphi^{ij})_{1\leq i,j\leq D}$ (b) Mark kernel $(f^{ij})_{1\leq i,j\leq D}$

Figure 5: *Kernel with inhibition* — Comparison of the predicted and exact aggregated kernel matrices defined in Equations (6) and (7).

The fits of aggregated time kernels and mark kernels are displayed in Figure 5. Note that the model is able to correctly capture the inhibition effects together with the latencies.

3.2 Convergence rate with respect to the number of observations

We measure the rate of decay of the estimation error of the moment-based non-parametric neural Hawkes method for smooth kernels with respect to the number of observations. We set $D = 2$, $M = 2$ and simulate a linear marked Hawkes process with exponential and power-law kernels. In the exponential case, kernels are given by

$$\phi^{ij}(t,x) := \alpha_{ij} e^{-\beta_{ij} t} f_1(x), \quad 1 \leq i,j \leq D,\ t \in [0,+\infty),\ x \in \mathcal{X}, \tag{40}$$

and numerical values of the parameters are given in Appendix in Table 8. In the power-law case, kernels are given by

$$\phi^{ij}(t,x) := \alpha_{ij}(\gamma_{ij} + t)^{-\beta_{ij}} f_1(x), \quad 1 \leq i,j \leq D,\ t \in [0,+\infty),\ x \in \mathcal{X}, \tag{41}$$

and numerical values of the parameters are given in Appendix in Table 9. For each case, we simulate a sample of N events and apply the moment-based non-parametric neural Hawkes estimation method. We then compute for the estimated kernels $\hat{\phi}_N^{i,j}$ a root mean squared error (using a regular subdivision of $[0,T]$ of size K):

$$\Delta_2(N) := \sqrt{\frac{1}{DM(K+1)} \sum_{i,j=1}^{D} \sum_{m=1}^{M} \sum_{k=0}^{K} \left(\hat{\phi}_N^{i,j}\left(\frac{kT}{K}, m\right) - \phi^{i,j}\left(\frac{kT}{K}, m\right)\right)^2}. \tag{42}$$

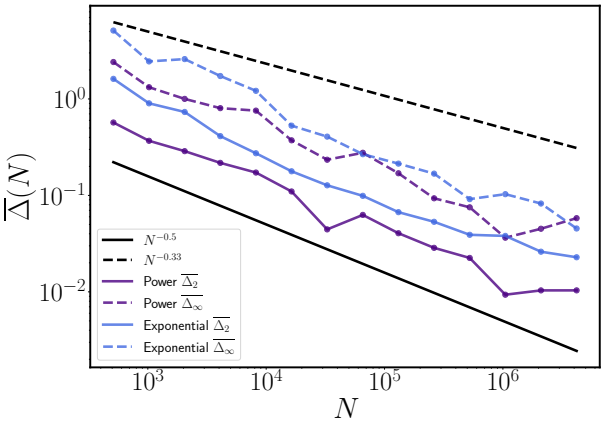


Figure 6: *Robustness experiment* — Decays of the normalized errors as functions of the number of events N for both power and exponential experiments. We observe a decay rate $\sim N^{-0.5}$ over the 4 decades.

For ease of comparison, we also define a normalized version $\overline{\Delta_2}(N)$ of this error by

$$\overline{\Delta_2}(N) := \frac{\Delta_2(N)}{\sup_{i,j,m} \|\phi^{ij}(\cdot, m)\|_\infty}. \tag{43}$$

Similarly, we define a L^∞-error

$$\Delta_\infty(N) := \sup_{i,j,m,k} \left| \hat{\phi}_N^{i,j}(kT/K, m) - \phi^{i,j}(kT/K, m) \right|, \tag{44}$$

and its corresponding normalized error

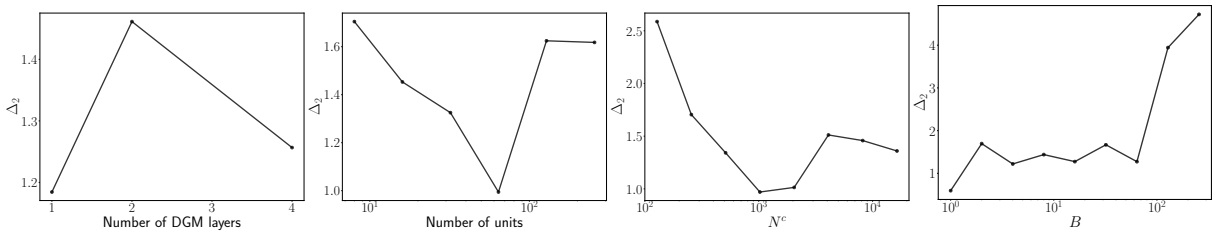
$$\overline{\Delta_\infty}(N) := \frac{\Delta_\infty(N)}{\sup_{i,j,m} \|\phi^{ij}(\cdot, m)\|_\infty}. \tag{45}$$

Figure 6 plots the normalized errors $\overline{\Delta_2}$ and $\overline{\Delta_\infty}$ with respect to the number of events N for both kernels. We empirically observe a decay rate $\sim N^{-0.5}$ over several decades. Note that these results are obtained with the standard configuration of Table 1 without any fine tuning. Interestingly, it was shown in Bacry and Muzy (2016) that the theoretical rate — empirically validated in the same paper — of the Wiener-Hopf method is rather $\sim N^{-0.33}$.

3.3 Robustness with respect to the choice of hyperparameters

We now test the robustness of the moment-based neural Hawkes estimation method with respect to the choice of various hyperparameters. We simulate the linear marked Hawkes process with power-law kernels defined at Equation (40) and Table 9 with $N = 2.10^6$ events and apply our estimation method with varying number of DGM cells (first experiment), varying number of neurons in the DGM cell (second experiment), varying size of the training set (third experiment) and varying batch size (fourth experiment).

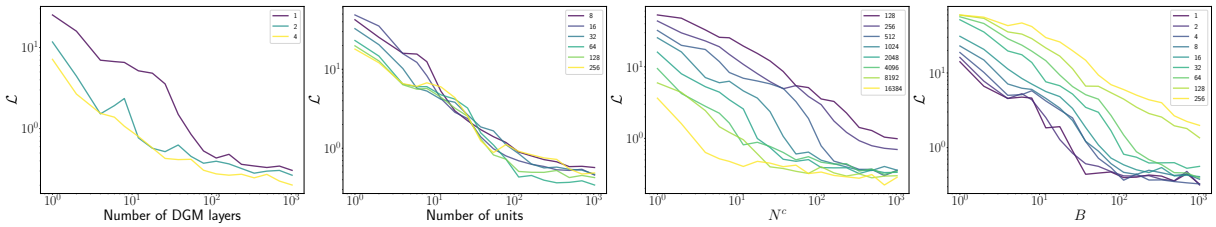
Figures 7 and 8 plot for each experiment the errors Δ_2 and the validation losses $\mathcal{L}$ of the trained model for the power law case. Additional material regarding the errors Δ_∞ is displayed in appendix and exhibits similar shapes as Δ_2 — see Figure 12. The aforementioned losses are the average of the validation losses of the two models $(\hat{\phi}^{1j})_{1\leq j\leq 2}$ and $(\hat{\phi}^{2j})_{1\leq j\leq 2}$, *i.e.* $\mathcal{L} = \frac{1}{2}(\mathcal{L}_1 + \mathcal{L}_2)$, across the epochs. We observe that neither the number of cells nor the number of units seem to have a significant impact on the quality of the model. Tuning the number of layers or the number of units does not necessary lead to a solid improvement of the overall performance, but a number of neurons in the range of $[50, 100]$ seems a good choice hence justifying our general hyperparameter configuration. Concerning the training set we can formulate a similar conclusion and set a good value in $[1000, 2000]$. By observing losses decays, we can expect that smaller values of the training set should be coupled with a larger number of epochs to ensure that the model explores and enough number of collocation points. Thus, the first three experiments do not suggest a high sensitivity of the model to these hyperparameters. Finally, we observe that large



(a) Number of DGM cells (b) Number of neurons (c) Size of the training set (d) Batch size

Figure 7: *Robustness experiment* — Error Δ_2 as a function of the hyperparameters in the four experiments.

values of batch size lowers the learning speed. This result contrasts with the findings of Sankaran et al. (2022) and may be specific to our problem as the authors explored the case of PDEs and not IEs. We also found that a very small batch size, *i.e.* in the range [1,8], generally improves the performance. All in all, it is very satisfying to observe that the moment-based neural Hawkes estimation method appears robust with respect to the choice of hyperparameters.



(a) Number of DGM cells (b) Number of neurons (c) Size of the training set (d) Batch size

Figure 8: *Robustness experiment* — Validation loss as a function of the number of epochs over the four experiments.

4 Application to high-frequency cryptocurrency data

4.1 Data

In this section we apply our estimation method to cryptocurrency trade data provided by SUN ZU Lab. The dataset is composed of trades executed between December 1st, 2023 and December 30th, 2023 on Coinbase for 15 cryptocurrency pairs. With every trade is given a timestamp that corresponds to the time at which the matching engine of the market processed the transaction. Timestamps are given with microsecond precision. For each pair, trades that share the same timestamp are aggregated and identified as one single event that was initiated either as a market order or as a marketable limit order. No feed handler disconnection was reported during the time period, ensuring a coherent estimation of the first and second order statistics. Descriptive statistics are displayed in Table 6.

4.2 The impact of volume on the arrival rate

To illustrate the stability of our moment-based neural Hawkes estimation method in the presence of a marked kernel, we analyze the influence of the traded size on the trade arrival rate. Note that a similar analysis was made in Rambaldi et al. (2017) on traditional financial markets using the Wiener-Hopf estimation method. We focus on the BTC-USD pair, set $D = 1$ and $M = 15$ and the mark random variable ξ encodes the volume of each trade. The hyperparameter configuration we use is the generic one of Table 1, we set $T = 1$ s and estimate the second order statistics with $h = 5.10^{-4} = 500$ µs, $n_{\text{lin}} = 50$ and $n_{\text{log}} = 100$. The discretization of the mark domain and the associated probability mass function p of ξ are displayed in Table 7. The adopted binning scheme uses a logarithmic grid from 100 USD to 100,000 USD. The analysis differs from the one of Rambaldi et al. (2017) as we focus on the shape of the aggregated mark kernel f and the asset classes, and hence the market microstructures, are different. However, our results are consistent in the sense that the shape of the kernel depends on the traded volume.

Table 6: Descriptive statistics of the dataset over the 1 month period.

Pair	Number of events	Λ^i (events.s^{-1})	Traded volume (millions USD)
BTC-USD	5,196,281	2.00	15,803.85
ETH-USD	3,579,423	1.38	8,058.49
SOL-USD	2,734,306	1.05	5,655.90
ICP-USD	569,019	0.22	464.61
SEI-USD	730,907	0.28	567.25
AVAX-USD	1,668,967	0.64	2,654,.66
XRP-USD	1,190,267	0.46	1,091.47
LINK-USD	1,237,588	0.48	1,275.48
XLM-USD	580,121	0.22	288.27
ADA-USD	1,436,622	0.55	1,031.34
MATIC-USD	896,950	0.35	753.90
DOT-USD	810,835	0.31	415.30
AAVE-USD	658,330	0.25	290.75
DOGE-USD	1,195,724	0.46	1,174.46
LTC-USD	967,231	0.37	405.78

Table 7: *The impact of volume* - Discretization of the mark domain and probability mass function of ξ.

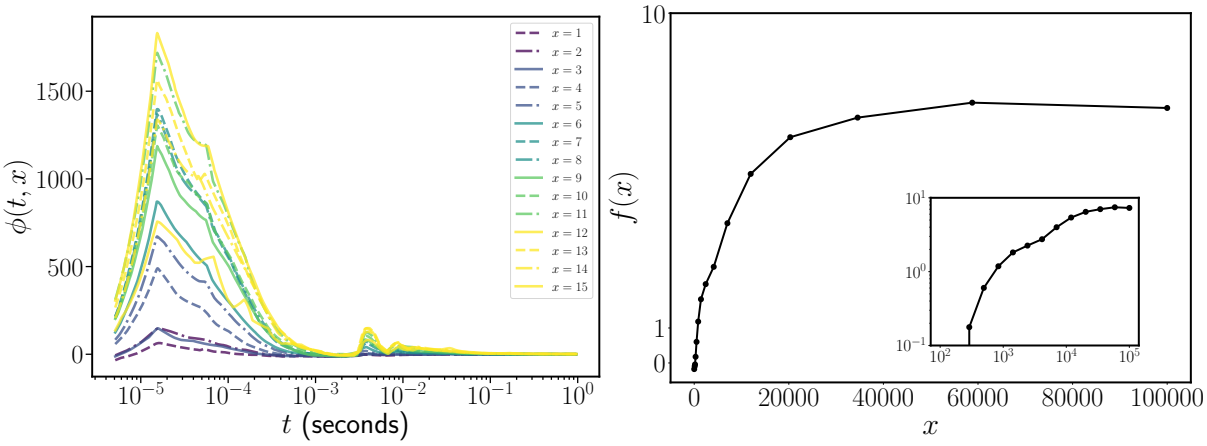
m	Volume interval (USD)	$p(m)$
1	(0 - 100]	0.46
2	(100 - 170]	0.046
3	(170 - 290]	0.053
4	(290 - 490]	0.046
5	(490 - 840]	0.043
6	(840 - 1,425]	0.053
7	(1,425 - 2,425]	0.051
8	(2,425 - 4,125]	0.057
9	(4,125 - 7,000]	0.078
10	(7,000 - 12,000]	0.055
11	(12,000 - 20,300]	0.028
12	(20,300 - 34,500]	0.015
13	(34,500 - 58,750]	0.008
14	(58,750 - 100,000]	0.003
15	(100,000 - $+\infty$]	0.002

Figure 9 shows the kernels $(\phi(.,m))_{m\in\mathcal{X}}$ and the aggregated mark kernel f computed using Equation (7). We observe that the aggregated kernel is a concave non-decreasing function of the volume and reaches a plateau for sizes that are larger than 50,000 USD. The monotony of the function is expected as the larger the size, the greater the amount of information interpreted by the market. A possible explanation of the concavity of the function and its plateau is that the agents consider there is less and less additional information when the size increases and once it reaches a specific value, *i.e.* here 50,000 USD, larger sizes do not bring any additional information.

This finding provides insights to practitioners regarding the choice of parametric kernels in order flow models. Concerning the kernel functions $(\phi(.,x))_{x\in\mathcal{X}}$ they exhibit different latency peaks depending on the traded volume. We see that they all globally share three latency peaks at around 10 µs, 4 to 5 ms and a smaller one near 10 ms. However, for large values of x, we observe two another peaks: one near 70 µs and one near 200 µs.

4.3 Cryptocurrency spillover: a causality analysis of transactions

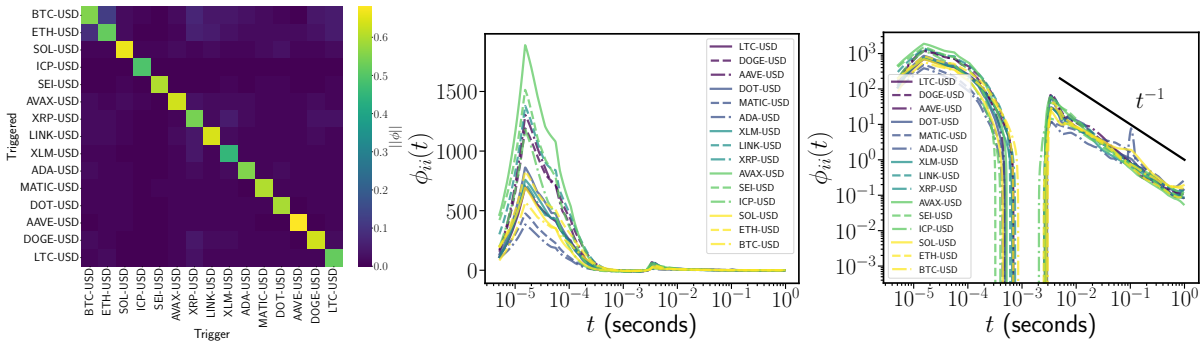
We analyze the causality relationships between arrival times of transactions amongst the 15 cryptocurrency pairs of Table 6. To this extent, we proceed to a moment-based non-parametric neural Hawkes estimation in $D = 15$ dimensions and set $M = 1$. The hyperparameter configuration we use is the generic one of Table 1, we set $T = 1$ s and estimate the second order statistics with $h = 5.10^{-4} = 500$ µs, $n_{\text{lin}} = 50$ and $n_{\log} = 100$. The kernel ϕ^{ij} characterizes the impact of the arrival of a transaction in the



(a) Kernels $(\phi(t,x))_{x\in\mathcal{X}}$ — x-axis in logarithmic scale.

(b) Aggregated mark kernel $(f(x))_{x\in\mathcal{X}}$.

Figure 9: *The impact of volume* — Time kernels as functions of time and aggregated mark kernel f as a function of the traded volume (USD). The mark kernel seems to reach a plateau for $x > 50,000$.



(a) Endogeneity matrix $(\|\phi^{ij}\|)_{1\le i,j\le D}$

(b) Kernels $(\phi^{ii}(t))_{1\le i\le D}$ — x-axis in logarithmic scale.

(c) Kernels $(\phi^{ii}(t))_{1\le i\le D}$ — Both axes in logarithmic scale.

Figure 10: *Cryptocurrency spillover* — Endogeneity ratios and diagonal of the kernel matrix. Note the near-critical power law above 3 milliseconds with exponent close to 1.

j-th pair on the intensity of the arrival rate of transactions in the i-th pair.

The endogeneity matrix $(\|\phi^{ij}\|)_{1\le i,j\le D}$ and the diagonal components $(\phi^{ii})_{1\le i\le D}$ of the kernel are displayed in Figure 10. The endogeneity ratio $\mathcal{R}$ of the system is 0.80. We observe that Bitcoin (BTC) and Ethereum (ETH) are not the most endogeneous components of the system as other less traded coins show a much larger endogeneity ratio for their self-excitation component, e.g., AAVE, DOGE, SOL or LINK. Furthermore, as observed for the BTC-USD pair in the previous analysis, the kernels exhibit mainly three peaks at around 10 µs, 2 to 5 ms and a smaller one at 10 ms. The self-excitation component of ADA shows a significant peak at 100 ms which may be due to some properties that are specific to this coin. We recover the near-critical power law behavior $\sim t^{-\gamma}$ with $\gamma \simeq 1$ of the kernel for $t > 3$ ms, consistent with the microstructure literature — *e.g.* see Bacry et al. (2012); Fosset et al. (2022). We provide the entire estimated kernel in Figure 13 in appendix. Notice that the results highlight the stability of the method as no fine tuning was made for this experiment.

If one denotes by $N^{i\leftarrow j}$ the counting process of events of type i that are caused by the arrival of events of type j, we have from Bacry et al. (2015)

$$\mathbb{E}\left(\mathrm{d}N_t^{i\leftarrow j}\right) = \Lambda^j \|\phi^{ij}\| \mathrm{d}t. \tag{46}$$

Therefore the ratio of the arrival rate of events of type i with a direct ancestor of type j to the overall arrival rate of events of type i is a measure of the causal relationship from j to i. We thus define the spillover ratio

$$S^{ij} := \frac{\Lambda^j}{\Lambda^i}\|\phi^{ij}\|\mathbb{1}_{\{i\neq j\}}, \quad 1 \le i,j \le D. \tag{47}$$

Questions that naturally arise when analyzing the microstructure of a highly correlated multi-asset framework are the following. Which asset does lead the others? In other words, which asset causes the others? Which asset does not? Can we sort the assets according to their causal strength amongst the selected trading universe? Is the traded volume a good indicator of such property?

To answer these questions and gain insights about the direction of causal relationships, we propose to look at two aggregated versions of the spillover ratio of Equation (47). The first measure is what we call the leader ratio

$$L^j := \frac{\Lambda^j}{\sum_{i \neq j} \Lambda^i} \sum_{i \neq j} \|\phi^{ij}\|, \quad 1 \leq j \leq D. \tag{48}$$

This measure is the ratio of the arrival rate of all the events — except those of type j — with a direct ancestor of type j to the overall arrival rate of all the events that are not of type j. The larger the ratio, the greater the causal strength of type j over the trading universe. In other words, a large ratio L^j indicates that a large proportion of the observed events that are not of type j is directly caused by an event of type j, suggesting that the j-th pair leads the other pairs. On the contrary, a small ratio $L^j \to 0$ indicates that very few events that are not of type j are directly caused by an event of type j, suggesting that the j-th pair does not lead the other pairs. The second measure we define is the receiver ratio

$$R^i := \frac{\sum_{j \neq i} \Lambda^j \|\phi^{ij}\|}{\Lambda^i}, \quad 1 \leq i \leq D. \tag{49}$$

It is the ratio of the arrival rate of the events of type i that are directly caused by events of other types than i to the overall arrival rate of events of type i. When $R^i \to 1$, the causal strength of other types of events over events of type i is large. Hence, a large ratio R^i indicates that a large proportion of the observed events of type i is directly caused by an event of another type, suggesting that the i-th pair is a receiver amongst the pairs. On the contrary, a small ratio $R^i \to 0$ indicates that very few events of type i are directly caused by an event of another type, suggesting that the i-th pair is not a receiver.

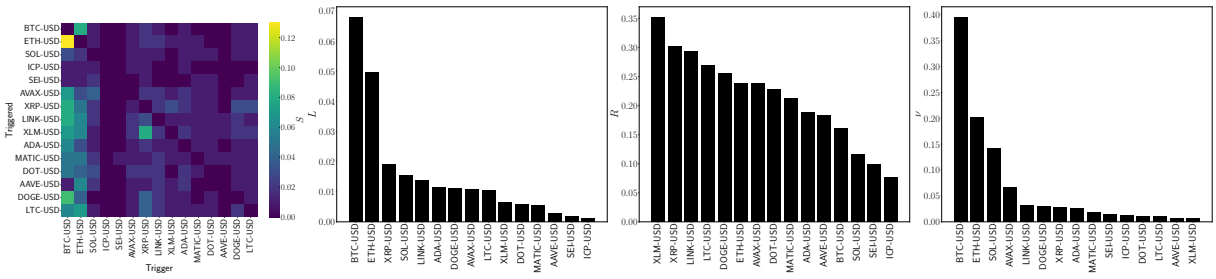
Usually, practitioners refer to the participation rate of the assets to gain insights about possible leader/lager relationships. For example, it is well-known — *e.g.* see Jia et al. (2023) — that the most traded cryptocurrencies, *i.e.* Bitcoin and Ethereum, lead the overall market. But the question can be trickier for less traded coins such as Cardano (ADA) and Ripple (XRP). We define the participation rate

$$\nu^j := \frac{V^j}{\sum_{i=1}^{D} V^i}, \tag{50}$$

where V^j is the traded volume of pair j and we compare the ranking with respect to this measure with the ranking with respect to the aggregated spillover ratio.

The results are displayed in Figure 11. First of all we clearly see discrepancies between the endogeneity ratio matrix of Figure 10 and the spillover ratio matrix which is due to the inherent differences in market activity amongst the coins. We observe a strong causality from BTC and ETH to other coins as expected. There is a strong bi-directional causality between BTC and ETH which is consistent with the conclusions of recent empirical studies — see for example Kumar and Anandarao (2019); Moratis (2021). Some of the least traded coins such as ICP, SEI and AAVE are strongly self-exciting but exhibit weak mutual excitation with other coins. Interestingly, we notice a bi-directional causality relationship between XRP and XLM, the strongest direction being from XRP to XLM. This finding gives credit to the widespread hypothesis that Ripple and Stellar (XLM) are two highly correlated cryptocurrencies[1]. In fact, the ideas behind these protocols are similar and the causal relationship between both prices is still a debate in the decentralized finance community. Furthermore, we observe that the top three leaders are BTC, ETH and XRP and the top three receivers are XLM, XRP and LINK. It shows that XRP is both a strong transmitter and a strong receiver which is consistent with the studies of Kim et al. (2021); Moratis (2021). Comparing the ratios with the participation rates, we see that high market activity does not necessarily mean high shock transmission. For example, the traded volume on AVAX is much greater than the one on XRP on this time period but XRP shows a significantly higher leader ratio. We also observe that least traded coins are not necessarily strong receivers, *e.g.* ICP. Finally, it is noteworthy that we trained several models with different hyperparameters configurations and all the results were mostly identical.

[1] Although we did not find any empirical analysis of the XRP/XLM relationship in the academic literature, we refer the reader to a blog discussion `https://u.today/intriguing-xrp-and-stellar-xlm-correlation-deciphered-heres-explanation` (Accessed on Jan. 9th, 2024).



(a) Spillover ratio matrix $(S^{ij})_{1\leq i,j\leq D}$

(b) Leader ratios $(L^j)_{1\leq j\leq D}$

(c) Receiver ratios $(R^i)_{1\leq i\leq D}$

(d) Participation rates $(\nu^j)_{1\leq j\leq D}$

Figure 11: *Cryptocurrency spillover* — Spillover ratios, ranking of the leader, receiver measures and participation rates amongst the 15 cryptocurrency pairs.

5 Conclusion

In this paper we proposed a new non-parametric estimation method for linear multivariate Hawkes processes that relies on two main tools:

- The second order statistics characterization theorem of multivariate linear Hawkes processes originally proven in Bacry and Muzy (2016);
- The physics-informed neural networks that are able to solve non-linear PDEs in high dimension, provided that they are properly trained – see Wang et al. (2023).

The Fredhlom equation that links second order statistics to the kernel function is solved using a neural network that is trained using the latest advances in PINNs literature. The benefits of our approach is threefold.

- The application of the Wiener-Hopf method requires the inversion of a matrix that can be arbitrarily large in high dimension, leading to potential computation issues and approximation errors. The neural network model is able to work around this problem as it is well suited to high dimension inference;
- The performance of the model can be evaluated by tracking the estimated second order statistics at any point, allowing us to evaluate its ability to fit the data;
- Once the model is trained, it is able to compute the value of the kernel at any given point. This contrasts with the Wiener-Hopf method for which the computation of the kernel between two quadrature points is done either by linear interpolation or by inversion of the characterization equation.

We provided a general hyperparameter configuration that can be used to fit a first non-parametric model before further tuning, as it showed robust performance across the numerical experiments. The model shows a great performance on simulated data with general kernels. We explored the influence of some hyperparameters on the final performance and found the model to be essentially sensitive to the batch size, the learning rate and the number of hidden layers. In general, a small batch size, a learning rate of 0.001 and 1 to 3 DGM cells provide very satisfactory results.

As an example of practical applications we conducted a non-parametric estimation of the kernel matrix of transactions on a centralized exchange in two settings. The first one is univariate marked Hawkes estimated on BTC-USD trades, mapping the traded volume to the mark process. The second one is a multivariate Hawkes estimated on a universe of 15 cryptocurrency pairs. We analyzed the shape of the mark component of the kernel and found it appears to behave as a concave function of the volume. We introduced a spillover measure and leader/receiver ratios that can be used for identifying the directions of the causal relationships of shocks amongst a correlated asset universe. The estimated metrics indicate that the traded volume is not necessarily representative of the leader-lager relationships. The use of our method can provide insights about the causality structure of large complex systems. In a forthcoming paper we will aim at applying this framework to high frequency cryptocurrency data and extract stylized facts about the role of marks in order flow dynamics or the spillover and lead-lag relationships in (very) large dimension.

References

Achab, M., Bacry, E., Gaıffas, S., Mastromatteo, I., and Muzy, J.-F. (2017). Uncovering causality from multivariate Hawkes integrated cumulants. In *International Conference on Machine Learning*, pages 1–10. PMLR.

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

Bacry, E., Dayri, K., and Muzy, J.-F. (2012). Non-parametric kernel estimation for symmetric Hawkes processes. application to high frequency financial data. *The European Physical Journal B*, 85:1–12.

Bacry, E., Delattre, S., Hoffmann, M., and Muzy, J.-F. (2013). Modelling microstructure noise with mutually exciting point processes. *Quantitative finance*, 13(1):65–77.

Bacry, E., Jaisson, T., and Muzy, J.-F. (2016). Estimation of slowly decreasing Hawkes kernels: application to high-frequency order book dynamics. *Quantitative Finance*, 16(8):1179–1201.

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

Bacry, E. and Muzy, J.-F. (2016). First-and second-order statistics characterization of Hawkes processes and non-parametric estimation. *IEEE Transactions on Information Theory*, 62(4):2184–2202.

Bischof, R. and Kraus, M. (2021). Multi-objective loss balancing for physics-informed deep learning. *arXiv preprint arXiv:2110.09813*.

Bonnet, A., Martinez Herrera, M., and Sangnier, M. (2023). Inference of multivariate exponential Hawkes processes with inhibition and application to neuronal activity. *Statistics and Computing*, 33(4):91.

Carreira, M. C. S. (2021). Exponential kernels with latency in Hawkes processes: Applications in finance. *arXiv preprint arXiv:2101.06348*.

Cartea, Á., Cohen, S. N., and Labyad, S. (2021). Gradient-based estimation of linear Hawkes processes with general kernels. *arXiv preprint arXiv:2111.10637*.

Deguchi, S. and Asai, M. (2023). Dynamic & norm-based weights to normalize imbalance in back-propagated gradients of physics-informed neural networks. *Journal of Physics Communications*, 7(7):075005.

Dissanayake, M. and Phan-Thien, N. (1994). Neural-network-based approximations for solving partial differential equations. *Communications in Numerical Methods in Engineering*, 10(3):195–201.

Duval, C., Luçon, E., and Pouzat, C. (2022). Interacting Hawkes processes with multiplicative inhibition. *Stochastic Processes and their Applications*, 148:180–226.

Effati, S. and Buzhabadi, R. (2012). A neural network approach for solving Fredholm integral equations of the second kind. *Neural Computing and Applications*, 21:843–852.

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

Fosset, A., Bouchaud, J.-P., and Benzaquen, M. (2022). Non-parametric estimation of quadratic Hawkes processes for order book events. *The European Journal of Finance*, 28(7):663–678.

Gao, X., Zhou, X., and Zhu, L. (2018). Transform analysis for Hawkes processes with applications in dark pool trading. *Quantitative Finance*, 18(2):265–282.

Guan, Y., Fang, T., Zhang, D., and Jin, C. (2022). Solving Fredholm integral equations using deep learning. *International Journal of Applied and Computational Mathematics*, 8(2):87.

Hawkes, A. G. and Oakes, D. (1974). A cluster process representation of a self-exciting process. *Journal of applied probability*, 11(3):493–503.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Jia, Y., Wu, Y., Yan, S., and Liu, Y. (2023). A seesaw effect in the cryptocurrency market: Understanding the return cross predictability of cryptocurrencies. *Journal of Empirical Finance*, 74:101428.

Joseph, S. and Jain, S. (2023). A neural network based model for multi-dimensional nonlinear Hawkes processes. *arXiv preprint arXiv:2303.03073*.

Joseph, S., Kashyap, L. D., and Jain, S. (2020). Shallow neural Hawkes: Non-parametric kernel estimation for Hawkes processes. *arXiv preprint arXiv:2006.02460*.

Jusselin, P. and Rosenbaum, M. (2020). No-arbitrage implies power-law market impact and rough volatility. *Mathematical Finance*, 30(4):1309–1336.

Kim, M. J., Canh, N. P., and Park, S. Y. (2021). Causal relationship among cryptocurrencies: A conditional quantile approach. *Finance Research Letters*, 42:101879.

Kirchner, M. (2017). An estimation procedure for the Hawkes process. *Quantitative Finance*, 17(4):571–595.

Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., and Mahoney, M. W. (2021). Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 34:26548–26560.

Kumar, A. S. and Anandarao, S. (2019). Volatility spillover in crypto-currency markets: Some evidences from GARCH and wavelet analysis. *Physica A: statistical mechanics and its applications*, 524:448–458.

Kwon, J., Zheng, Y., and Jun, M. (2023). Flexible spatio-temporal Hawkes process models for earthquake occurrences. *Spatial Statistics*, 54:100728.

Lagaris, I. E., Likas, A., and Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9(5):987–1000.

Lee, K. and Seo, B. K. (2017). Marked Hawkes process modeling of price dynamics and volatility estimation. *Journal of Empirical Finance*, 40:174–200.

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

Lu, X. and Abergel, F. (2018). High-dimensional Hawkes processes for limit order books: modelling, empirical analysis and numerical calibration. *Quantitative Finance*, 18(2):249–264.

Moratis, G. (2021). Quantifying the spillover effect in the cryptocurrency market. *Finance Research Letters*, 38:101534.

Muni Toke, I. and Pomponio, F. (2012). Modelling trades-through in a limit order book using Hawkes processes. *Economics*, 6(1):20120022.

Olinde, J. and Short, M. B. (2020). A self-limiting Hawkes process: interpretation, estimation, and use in crime modeling. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 3212–3219. IEEE.

Psichogios, D. C. and Ungar, L. H. (1992). A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10):1499–1511.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707.

Rambaldi, M., Bacry, E., and Lillo, F. (2017). The role of volume in order book dynamics: a multivariate Hawkes process analysis. *Quantitative Finance*, 17(7):999–1020.

Reynaud-Bouret, P., Rivoirard, V., and Tuleau-Malot, C. (2013). Inference of functional connectivity in neurosciences via Hawkes processes. In *2013 IEEE global conference on signal and information processing*, pages 317–320. IEEE.

Rousseau, J., Donnet, S., and Rivoirard, V. (2018). Nonparametric bayesian estimation of multivariate Hawkes processes. *arXiv preprint arXiv:1802.05975*.

Sankaran, S., Wang, H., Guilhoto, L. F., and Perdikaris, P. (2022). On the impact of larger batch size in the training of physics informed neural networks. In *The Symbiosis of Deep Learning and Differential Equations II*.

Sirignano, J. and Spiliopoulos, K. (2018). DGM: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364.

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

Wang, C., Li, S., He, D., and Wang, L. (2022a). Is L^2 physics informed loss always suitable for training physics informed neural network? *Advances in Neural Information Processing Systems*, 35:8278–8290.

Wang, S., Sankaran, S., and Perdikaris, P. (2022b). Respecting causality is all you need for training physics-informed neural networks. *arXiv preprint arXiv:2203.07404*.

Wang, S., Sankaran, S., Wang, H., and Perdikaris, P. (2023). An expert's guide to training physics-informed neural networks. *arXiv preprint arXiv:2308.08468*.

Wang, S., Teng, Y., and Perdikaris, P. (2021). Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081.

Wang, T., Bebbington, M., and Harte, D. (2012). Markov-modulated Hawkes process with stepwise decay. *Annals of the Institute of Statistical Mathematics*, 64:521–544.

Wu, C., Zhu, M., Tan, Q., Kartha, Y., and Lu, L. (2023). A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 403:115671.

Xu, H., Farajtabar, M., and Zha, H. (2016). Learning granger causality for Hawkes processes. In *International conference on machine learning*, pages 1717–1726. PMLR.

Yuan, L., Ni, Y.-Q., Deng, X.-Y., and Hao, S. (2022). A-PINN: Auxiliary physics informed neural networks for forward and inverse problems of nonlinear integro-differential equations. *Journal of Computational Physics*, 462:111260.

Zappala, E., Fonseca, A. H. d. O., Caro, J. O., and van Dijk, D. (2022). Neural integral equations. *arXiv preprint arXiv:2209.15190*.

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

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

Appendix

This appendix provides additional figures and tables that do not appear in the main text in order to enhance readability. Tables 8 and 9 give additional parameters used in Section 3. Figure 12 displays L_∞ errors of the robustness tests of Section 3. Finally, Figure 13 displays the full 15-dimensional kernel estimated in the empirical analysis of Section 4.

Table 8: A 2-variate exponentially decreasing kernel

(a) α

α_{ij}	1	2
1	1	0.25
2	0.5	0.75

(b) β

β_{ij}	1	2
1	2	1
2	1	1.5

Table 9: A 2-variate power-law decreasing kernel

(a) α

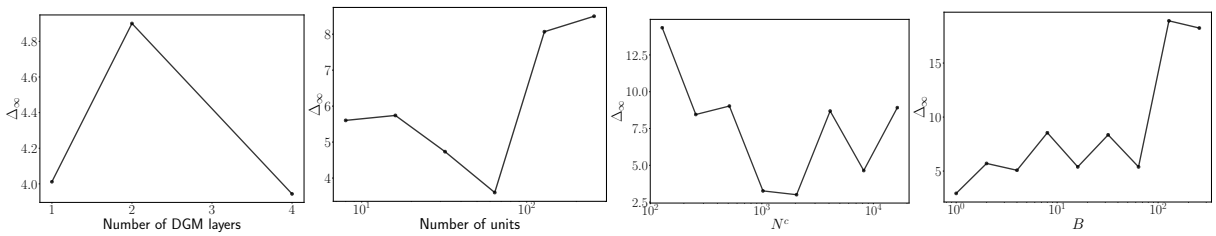
α_{ij}	1	2
1	0.012	0.008
2	0.004	0.005

(b) β

β_{ij}	1	2
1	1.3	1.3
2	1.3	1.3

(c) γ

γ_{ij}	1	2
1	0.0005	0.0005
2	0.0005	0.0005



(a) Number of DGM cells (b) Number of neurons (c) Size of the training set (d) Batch size

Figure 12: *Robustness experiment* — Error Δ_∞ as a function of the hyperparameters in the four experiments.

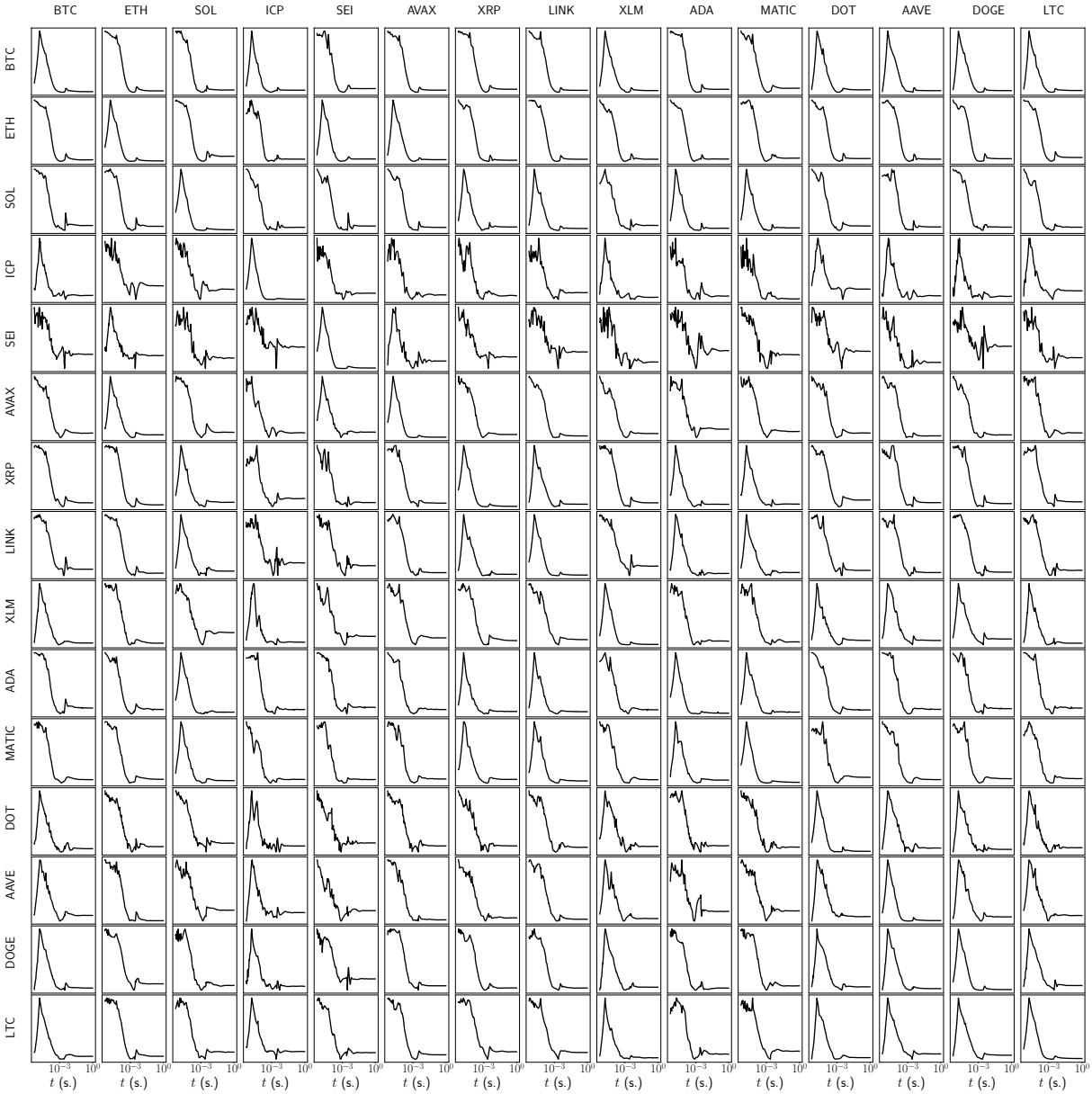


Figure 13: *Cryptocurrency spillover* — Moment-based neural Hawkes kernel $(\phi^{ij})_{1\leq i,j\leq D}$ as a function of time for a universe of 15 cryptocurrencies using trade data on Coinbase from 2023/12/01 to 2023/12/30. Note the scale of the y-axis might change from one kernel to another as they do not necessary share the same orders of magnitude. We chose to remove the ticks of the y-axis for illustration purposes.