Sequential Inference Methods for Non-homogeneous Poisson Processes with State-space Prior

Chenhao Li, Simon Godsill
University of Cambridge
Department of Engineering
Cambridge, CB2 1PZ, UK
Email: cl557@cam.ac.uk, sjg@eng.cam.ac.uk

***Abstract*—The non-homogeneous Poisson process provides a generalised framework for the modelling of random point data by allowing the intensity of point generation to vary across its domain of interest (time or space). The use of non-homogeneous Poisson processes have arisen in many areas of signal processing and machine learning, but application is still largely limited by its intractable likelihood function and the lack of computationally efficient inference schemes, although some methods do exist for the batch data case. In this paper, we propose for the first time a sequential framework for intensity inference which combines the non-homogeneous model of Poisson data with continuous-time state-space models for their time-varying intensity. This approach enables us to design efficient online inference schemes, for which we propose a novel sequential Markov chain Monte Carlo (SMCMC) algorithm, as is demanded by many applications where point data arrive sequentially and decisions need to be made with low latency. The proposed approach is compared with competing methods on synthetic datasets and tested with high-frequency financial order book data, showing in general improved performance and better computational efficiency than the main batch-based competitor algorithm, and better performance than a simple baseline kernel estimation scheme.**

I. Introduction

The Poisson process stands out for its ability to model point data in both temporal and spatial settings. Intensity inference for Poisson processes under various settings provides valuable insights into both the behaviour of the stochastic process itself and the prediction of future events.

Poisson processes have been widely applied in many disciplines, for example modelling of neuronal spike trains as Poisson processes [1] and the study of earthquake sequences [2]. In finance, transactions of orders in an open limit order book market are frequently modelled as Poisson processes [3]; and thanks to its convenient mathematical properties, the process is also combined with other sophisticated models such as state-space models (SSMs) in order to capture salient features of dynamical systems with jumps [4], [5].

The non-homogeneous Poisson process (NHPP), as an important variant of the standard Poisson process [6], provides extra flexibility by allowing process intensity to vary across time and/or space, which gives more general and realistic modelling of real-world data. However, intensity inference for the NHPP is non-trivial and has been discussed in many literatures. The early frequentist approach proposed in [7] uses kernel densities to construct an intensity estimator for the NHPP. The method developed in [8] assumes a piecewise-linear form of the intensity function and estimates linear function parameters via regression. Both methods have achieved relatively fast estimation of intensity but the inference accuracy is rather sensitive to the choice of hyperparameters and model assumptions. The authors of [9] proposed the first tractable approach to perform Bayesian intensity inference on Cox processes as a type of NHPP with a sigmoid transformation function. Based on the ideas in [9], the model was later extended in [10] to include variational sampling of hyperparameters and parallel inference for multiple correlated processes. However, both approaches scale poorly with the size of the dataset due to the high complexity of the Gaussian process prior and Bayesian computation. Inspired by sparse Gaussian process models, [11] uses generative *inducing points* and the convolution process to perform tractable Bayesian intensity inference on multiple correlated NHPPs, which achieves reduced complexity compared to the naive implementation.

A short conference paper summarising an early version of our work was published as [12] which includes a much more limited theoretical discussion and experimental evaluation, does not include full details of the our SMCMC scheme, and relies on a much more basic version of SMCMC, not including for example the rejection sampling (RS) approach or the Gibbs-like refinement steps that are newly proposed in this paper. The contributions of this paper are:

- We propose a novel model of NHPP in which the intensity function is governed by a continuous-time state-space model.
- We then develop a new point process version of the SMCMC algorithm [13], [14] as the inference method to allow sequential online Bayesian inference for the process intensity.
- We propose the use of a mixture sampling scheme and sequential batch scheme to improve inference accuracy and computational efficiency including a new *rejection sampling* method.
- An empirical study of inference performance on both

synthetic and real datasets in comparison with the frequentist kernel density estimation (KDE) approach [7] and Sigmoidal Gaussian Cox Process (SGCP) approach [9].

The remainder of the paper is organised as follows. We first introduce general properties and simulation methods for the NHPP in Section **II** before reviewing the SGCP model [9]. Sections **III** and **IV** introduce the newly proposed model of a SSM-governed NHPP and its corresponding SMCMC inference algorithm, respectively. Experimental results are shown in Section **V** including comparisons between models, convergence tests and hyperparameter analyses.

II. Background

In this section, we briefly introduce the NHPP and its generative procedures for data when the intensity function is known. We then review the SGCP method [9], which forms a starting point for our proposed approach.

A. Non-homogeneous Poisson Processes

In contrast with the standard Poisson process, the non-homogeneous variant generalises the process by allowing a time-varying intensity function. The NHPP itself can be defined in several equivalent ways, each fitting the general definitions of a Poisson process. For the application to our setting, we present an intuitive definition of the NHPP as introduced in [6]:

Definition 1. *Non-homogeneous Poisson Process: For a domain $\mathcal{S} = \mathbb{R}^D$, we define a NHPP with an intensity function $\lambda(s) \geq 0, s \in \mathcal{S}$, and the counting measure $N(\mathcal{T})$ (i.e. number of occurrences) in any bounded region $\mathcal{T} \subset \mathcal{S}$ s.t.:*

1) $N(\emptyset) = 0$
2) $\{N(\mathcal{T}_i)\}_i$ *are independent for any disjoint subsets* $\{\mathcal{T}_i\} \subset \mathcal{S}$
3) $N(\mathcal{T}) \sim \text{Poisson}(\Lambda)$, *with* $\Lambda = \int_{\mathcal{T}} \lambda(s)\, \mathrm{d}s$

Let $\{s_k\}_{k=1}^K$ be a set of K events/occurrences in a region $\mathcal{T}$; then with the definition above we can write the likelihood function of the NHPP with intensity $\lambda(s)$ as the product of three probabilities: (1) the Poisson probability of observing K events in $\mathcal{T}$: $\frac{e^{-\Lambda}\Lambda^K}{K!}$; (2) the density of the events $\{s_k\}_{k=1}^K$: $\prod_{k=1}^K \frac{\lambda(s_k)}{\Lambda}$; and (3) the $K!$ number of possibilities of ordering the K events. Thus the likelihood can be expressed as:

$$
\begin{aligned}
p(\{s_k\}_{k=1}^K \,|\, \lambda(s), \mathcal{T}) &= \frac{e^{-\Lambda}\Lambda^K}{K!} \times \prod_{k=1}^K \frac{\lambda(s_k)}{\Lambda} \times K! \\
&= \exp\Big\{ - \int_{\mathcal{T}} \lambda(s) \mathrm{d}s \Big\} \prod_{k=1}^K \lambda(s_k)
\end{aligned} \tag{1}
$$

It is clear that the direct computation of the above likelihood requires not only pointwise evaluations of the intensity function at event times/locations $\{s_k\}_{k=1}^K$ but also an integration of the intensity function $\lambda(s)$ over the region of interest $\mathcal{T}$. The integration is in general intractable, except for simple known forms of intensity function. Such intractability inhibits the inference of intensity function via direct likelihood-based or Bayesian inference approaches. It is of course possible to perform intensity inference (either Bayesian or non-Bayesian) assuming tractable functional forms for the intensity function, but this leads to restrictive modelling constraints.

B. Thinning and Simulation

Despite the intractability of the likelihood, exact simulation of fairly general classes of NHPP can be performed tractably using *thinning* methods as introduced in [15]. The thinning operation entails removing point(s) from an existing point process based on some predefined rules in order to produce a new point process.

Here, we focus on independent thinning, where the decision about removing each point is made by an independent Bernoulli trial and interactions between points has no effect on this decision [16]. The following thinning theorem is fundamental to the NHPP simulation:

Theorem 1. *[15] Consider a homogeneous Poisson process (HPP) with constant intensity λ^* over a domain $S = \mathbb{R}^D$, so that the counting measure over any bounded region $\mathcal{T} \subset \mathcal{S}$ is $N^*(\mathcal{T}) \sim$ Poisson$(\lambda^*|\mathcal{T}|)$, where $|\mathcal{T}|$ is the Lebesgue measure of $\mathcal{T}$. If the points of this process undergo an independent thinning operation with a spatially varying deletion probability $1 - p(s)$, the remaining points form a NHPP with intensity function $\lambda^* p(s)$ within region $\mathcal{T}$.*

Following **Theorem 1** to generate a NHPP with desired intensity $\lambda(s)$, we can simply start with a HPP having intensity λ^* and perform the described thinning operation with spatially varying Bernoulli (retaining) probabilities:

$$
p(s) = \frac{\lambda(s)}{\lambda^*} \ , \ \lambda^* \geq \sup_{s \in \mathcal{S}} \{\lambda(s)\} \tag{2}
$$

The value of λ^* should be chosen such that it serves as an upper bound on $\lambda(s)$. It is also worth noting that the starting process (of counting measure $N(\mathcal{T})$) is not restricted to HPP and the method of generating non-homogeneous realisations via thinning also generalises to NHPP as long as the initial intensity $\lambda^*(s)$ is an envelope of the desired intensity $\lambda(s)$ such that $\lambda^*(s) \geq \lambda(s), \forall s \in \mathcal{T}$ [15]. From this, we observe a close relation between *thinning* and *rejection sampling* (RS), as the tighter the envelope $\lambda^*(s)$ is, the more efficient simulation will be.

Based on the above, without requiring any form of integration we can thus simulate any NHPP whose intensity function is upper-bounded and which can be evaluated point-wise. A set of N homogeneous Poisson points $\{s_n\}_{n=1}^N$ are generated by first drawing the variable N from a Poisson distribution with parameter $(\lambda^*|\mathcal{T}|)$, followed by N independent uniform random draws within $\mathcal{T}$. The homogeneous points are then thinned with Bernoulli probabilities $(1 - \lambda(s)/\lambda^*)$ to provide the NHPP events having the desired intensity function $\lambda(s)$. **Algorithm 1** shows the detailed procedure of simulating a NHPP, accompanied by a graphical illustration in **Fig. 1**.

It can be noted that the application of **Theorem 1** requires

Algorithm 1 Simulation of a NHPP

```
Inputs: domain of interest $\mathcal{T} \subset \mathcal{S}$, intensity function $\lambda(s)$, upper-bound intensity $\lambda_{\max}$
Outputs: A set of (random) $K$ NHPP events $\Phi = \{s_k\}_{k=1}^{K}$ within $\mathcal{T}$

$N \sim \text{Poisson}(\lambda_{\max}|\mathcal{T}|)$                ▷ No. events in HPP
$\{s_n\}_{n=1}^{N} \sim \text{Uniform}(\mathcal{T})$             ▷ Sample event locations
$\Phi \leftarrow \emptyset$                           ▷ An empty set for NHPP events
for $n = 1 : N$ do
    $\rho_n \sim \text{Uniform}(0, 1)$
    if $\rho_n \le \lambda(s_n)/\lambda_{\max}$ then                 ▷ Thinning decision
        $\Phi \leftarrow \Phi \cup s_n$
    end if
end for
Return $\Phi$
```

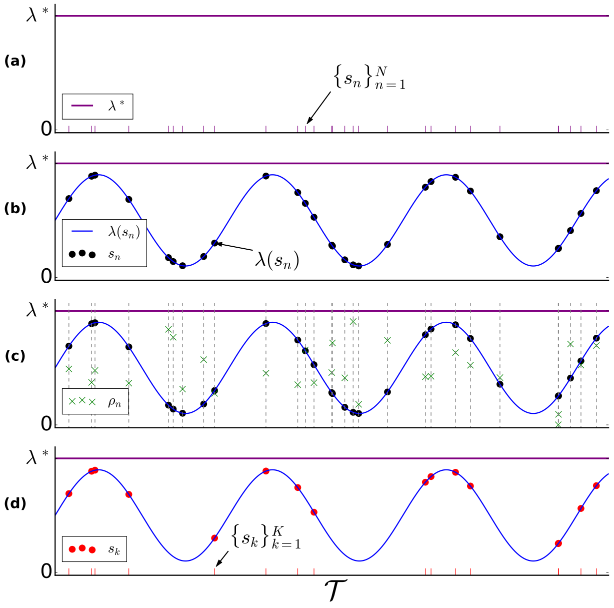


Fig. 1: The figure shows the generative procedure of the NHPP with a periodical intensity function: (a) simulation of HPP with intensity λ^*; (b) point-wise evaluation of $\lambda(s)$ at locations $\{s_n\}_n$; (c) sample variates $\rho_n \sim \text{Uniform}(0, \lambda^*)$; (d) retain points with $\rho_n \le \lambda(s_n)$

an assumption of an attainable maximum intensity λ^* of the NHPP. Although this assumption may seem restrictive, it can generally be satisfied in most real-world applications especially as λ^* is not necessarily a *tight upper bound* or a *supremum* of the intensity function $\lambda(s)$. In practice, an intuitive value of λ^* is often inherently defined by the modelled systems e.g. the maximum number of phone calls that can be processed by the call center at one time; and in the real-data experiment presented later in Section **V-B**, the maximum number of limit order arrivals is also restricted by the exchange. However, this intuitive bound may not be the best choice for λ^* as we discussed above the tightness of this upper bound is closely related to algorithm efficiency (for both simulation and intensity inference). The value of λ^* should be tuned to accommodate the specific application.

C. Sigmoidal Gaussian Cox Process

Inference for the NHPP usually involves a non-preconceived functional form of the intensity function, which incites the development of doubly-stochastic Poisson process where the varying intensity function $\lambda(s)$ is governed by another stochastic process $\{g(s), s \in \mathcal{T}\}$. The SGCP model introduced in [9] uses a Gaussian process ($\mathcal{GP}$) [17] as the prior, which is then mapped to the non-negative intensity function through a scaled sigmoid function $\lambda(s) = \lambda^*\sigma(g(s))$, with $\sigma(x) = (1+e^{-x})^{-1}$. Inspired by the constructive generative process of the NHPP, the SGCP model achieves tractability by considering the observed NHPP as the output from a thinning operation applied to a latent HPP. Define $i_n \in \{0, 1\}$ as an indicator associated with each Poisson event, taking value 0 for an observed data point and 1 for a 'thinned' point. We thus have the retaining probability for an observed point as $\frac{\lambda(s)}{\lambda^*} = \sigma(g(s))$; and an augmented joint probability for NHPP:

$$p(\{s_n\}_{n=1}^{N}, \mathbf{g}_{1:N}, \{i_n\}_{n=1}^{N} \mid \lambda^*, \mathcal{T})$$
$$= \underbrace{(\lambda^*)^N \, \mathrm{e}^{-\lambda^*|\mathcal{T}|}}_{(1)} \underbrace{p(\mathbf{g}_{1:N}|\{s_n\}_{n=1}^{N})}_{(2)} \underbrace{\prod_{n=1}^{N} \sigma\{(-1)^{i_n} g(s_n)\}}_{(3)} \quad (3)$$

where $\mathbf{g}_{1:N}$ is the concatenated vector with $\mathbf{g}_{1:N} = [g(s_1), g(s_2), \ldots, g(s_N)]^T$. The formulation of Eq. (3) follows similar steps to the simulation in **Algorithm 1**, with the addition of random simulation from the intensity function: (1) represents the probability of generating N ordered points in $\mathcal{T}$ according to the upper-bound intensity λ^*; (2) is the probability of generating stochastic process values $\mathbf{g}_{1:N}$ at times $\{s_n\}_{n=1}^{N}$ from the prior; and (3) is the probability of the Bernoulli trials, since $1-\sigma(x)=\sigma(-x)$. Inference for the SGCP model is achieved in [9] by offline batch-based MCMC samplers for each latent variable which alternate in a Gibbs sampling manner.

The SGCP model has illustrated a tractable inference procedure for NHPPs. However the practical application of the model is limited by the $\mathcal{O}(N^3)$ complexity arising from the $\mathcal{GP}$ prior and the corresponding MCMC inference methods. The value of N is the total number of homogeneous points, and this can be much larger compared to the number of input points when the intensity function has regions of low intensity compared to the upper-bound intensity λ^*. Furthermore, the batch-based MCMC sampler proposed provides only retrospective knowledge of the NHPP and cannot readily be adapted to a sequential, online setting.

III. New Model

In order to alleviate these limitations, a new state-space intensity model is proposed in this section under the generative thinning framework, which in the meantime allows efficient sequential Bayesian inference.

A. Continuous-time State Space Model

In order to construct a computationally tractable sequential framework for non-homogeneous point process data while allowing flexibility in choice of prior characteristics, we employ a continuous-time SSM, as is commonly used in tracking applications, to replace the fully correlated Gaussian process prior. Denoting $\mathbf{g}(t)$ as the state vector at time t, we formulate the SSM as the following stochastic differential equation (SDE):

$$d\mathbf{g}(t) = \mathbf{A}\mathbf{g}(t)\ dt + \mathbf{h}\ dW_t \tag{4}$$

where $\{W_t\}$ is a Wiener process. Such a model, which is linear and Gaussian, can be discretised exactly in closed form using Itô calculus. The solution of the SDE, integrating from time P to Q for $Q \geq P$, is:

$$\mathbf{g}(Q) = e^{A(Q-P)}\Big[\mathbf{g}(P) + \int_0^{Q-P} e^{-A\tau}\mathbf{h}\ dW_\tau\Big] \tag{5}$$

and the conditional transition density $p\big(\mathbf{g}(Q)|\mathbf{g}(P)\big)$ can be readily computed to be Gaussian and Markovian:

$$p\big(\mathbf{g}(Q) \mid \mathbf{g}(P)\big) \sim \mathcal{N}\big(\mathbf{g}(Q) \mid \mu(Q,P), C(Q,P)\big) \tag{6}$$

with mean and covariance calculated directly from the stochastic integral in Eq. (5):

$$\mu(Q,P) = \mathbb{E}\{\mathbf{g}(Q) \mid \mathbf{g}(P)\} = e^{A(Q-P)}\mathbf{g}(P) \tag{7}$$

$$\begin{aligned} C(Q,P) &= \mathbb{E}\Big\{\big[\mathbf{g}(Q) - \mu(Q,P)\big]\big[\mathbf{g}(Q) - \mu(Q,P)\big]'\Big\} \\ &= e^{A(Q-P)}K(Q,P)\big(e^{A(Q-P)}\big)' \end{aligned} \tag{8}$$

where

$$K(Q,P) = \int_0^{(Q-P)} e^{-A\tau}\mathbf{h}\mathbf{h}'(e^{-A\tau})'d\tau \tag{9}$$

The computation of $K(Q,P)$ is non-trivial and can be obtained using matrix fraction decomposition [18] or approximated by series expansion of the exponential functions [19]. With the above definition of the conditional transition density, and a Gaussian initial state prior, we can obtain the joint probability of all or part of the state vector by conditioning and probability chain rule, thanks to the Markovian property.

While any Gaussian SSM could in principle be applied in our framework, we have adopted for illustration a Langevin dynamical model that is similar to that used in [5]. In such a model, the state vector $\mathbf{g}_t = [g_{1,t}, g_{2,t}]^T$ contains a value term $g_{1,t}$ and a stochastic trend term $g_{2,t}$ at time t. The general SDE in (4) is reformulated as:

$$d\begin{bmatrix} g_{1,t} \\ g_{2,t} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \theta \end{bmatrix}\begin{bmatrix} g_{1,t} \\ g_{2,t} \end{bmatrix} dt + \begin{bmatrix} 0 \\ \sigma \end{bmatrix} \mathrm{d}W(t) \tag{10}$$

where θ is the non-positive reversion coefficient and $\sigma > 0$ is the scale of the trend process. The Langevin dynamics have the advantage of being analytically tractable while allowing either long term or short term trend behaviours of the intensity depending on the choice of θ. We denote the joint prior under this model as:

$$p(\mathbf{g}_{1:N} \mid \{s_n\}_{n=1}^N) = \mathcal{LD}(\mathbf{g}_{1:N} \mid \{s_n\}_{n=1}^N) \tag{11}$$

where $\mathbf{g}_n = \big[g_{1,s_n}\ ,\ g_{2,s_n}\big]^T$ and $\{s_n\}_{n=1}^N$ are the time stamps.

B. Doubly-stochastic Process with SSM Dynamics

Under the Gaussian assumption the process values $g_{1,s}$ can lie anywhere on the real line. Hence, as in [9], a sigmoidal mapping onto $[0, \lambda^*]$ is adopted:

$$\lambda(t) = \lambda^*\sigma(g_{1,t}) \tag{12}$$

although we note that any suitable function $\sigma(.)$ that maps to $[0,1]$ could be used in place of the sigmoidal function. Under this formulation we may use **Algorithm 1** to generate realisations of the NHPP with dynamics specified in (10). With sampled event timestamps $\{s_n\}_{n=1}^N$ proposed from the homogeneous Poisson process, the state vectors $\{\mathbf{g}_n\}_{n=1}^N$ and corresponding intensities $\{\lambda(s_n)\}_{n=1}^N$ are evaluated sequentially through the conditional transition density (6) and the mapping function (12). Second row of **Fig. 3** shows a typical realisation of a NHPP generated with Langevin governing stochastic intensity (in solid purple line). Such a realisation is later used in testing of the proposed model and methods.

IV. Sequential Inference

As in the SGCP model [9], our model works with the tractable augmented joint probability in Eq. (3) but with the prior being the SSM. However unlike the $\mathcal{GP}$ in [9], the Markovian property of the SSM allows efficient sequential inference for the intensity. By inputting short batches of data delineated by times t_k, $k = 0, 1, ...$, inference is updated with arriving data for each batch. The time intervals $\mathcal{T}_k = (t_{k-1}, t_k] \subset \mathcal{T}$ could be e.g. regularly spaced, $t_k = k\delta_T$, or spaced according to the timings of the input (observed) points $\{s_n; i_n = 0\}$.

We further define the notation $x_k = \{s_n, \mathbf{g}_n, i_n;\ s_n \in \mathcal{T}_k\}$ as the locations, the state vectors, and the indicators corresponding to all events in the interval $\mathcal{T}_k$. Note that x_k includes both unobserved (latent) and observed components of the model. We write the recursion for the joint distribution as:

$$p(x_{1:k} \mid \lambda^*, \mathcal{T}_{1:k}) = p(x_{1:k\text{–}1} \mid \lambda^*, \mathcal{T}_{1:k\text{–}1})\, p(x_k \mid \lambda^*, x_{1:k\text{–}1}, \mathcal{T}_k) \tag{13}$$

where the second term of the conditional propagation can be conveniently factorised based on Eq. (3) as:

$$\begin{aligned} & p(x_k \mid \lambda^*, x_{1:k\text{–}1}, \mathcal{T}_k) \\ &= p(x_k \mid \lambda^*, x_{k\text{–}1}, \mathcal{T}_k) \\ &= (\lambda^*)^{N_k}\, e^{-\lambda^*|\mathcal{T}_k|} \times \mathcal{LD}(\{\mathbf{g}\}_{\mathcal{T}_k} | \{\mathbf{g}\}_{\mathcal{T}_{k\text{–}1}}, \{s\}_{\mathcal{T}_k}) \\ &\quad \times \prod_{n:s_n \in \mathcal{T}_k} \sigma\{(-1)^{i_n} g_{1,n}\} \end{aligned} \tag{14}$$

and $N_k = |\{n\,; s_n \in \mathcal{T}_k\}|$ is the total number of events in $\mathcal{T}_k$. Note that the number of events N_k in $\mathcal{T}_k$ is itself a random variable, therefore extra care is needed in performing the correct sequential inference. A suitable scheme is proposed below.

Suppose that at time interval k–1 we have a large collection of random and possibly weighted samples ('particles') drawn from the posterior joint distribution at k–1 with the pth particle and its corresponding weight denoted as:

$$\big\{x^p_{1:k-1}, w^p_{k-1}\big\} \sim p(x_{1:k\text{–}1}|\lambda^*, \mathcal{T}_{1:k\text{–}1}), \quad p = 1, ..., N_p \tag{15}$$

We can therefore approximate $p(x_{1:k-1}|\lambda^*, \mathcal{T}_{1:k-1})$ (the 'smoothing' distribution at k–1) with the empirical distribution of the particles:

$$p(x_{1:k-1}|\lambda^*, \mathcal{T}_{1:k-1}) \approx \sum_{p=1}^{N_p} w_{k-1}^p \, \delta_{x_{1:k-1}^p}(x_{1:k-1}) \tag{16}$$

with $w_{k-1}^p \geq 0$ and $\sum_p w_{k-1}^p = 1$. Combining the above approximated posterior with the factorised joint recursion of Eq. (13) & (14), we obtain an updated particle posterior distribution at interval k:

$$\begin{aligned} p(x_{1:k}|\lambda^*, \mathcal{T}_{1:k}) \approx \sum_{p=1}^{N_p} & w_{k-1}^p \, \delta_{x_{1:k-1}^p}(x_{1:k-1}) \\ & \times p(x_k|\lambda^*, x_{k-1}^p, \mathcal{T}_k) \end{aligned} \tag{17}$$

The above equation gives a mixed discrete-continuous distribution containing the point masses for the "past history" variables $x_{k-1} = x_{k-1}^p = \{s_n, \mathbf{g}_n, i_n;\, s_n \in \mathcal{T}_{k-1}\}^p$ and the conditional distributions for the "new variables" x_k. We can now propose samples jointly from this entire approximated distribution of past and new variables and compute the importance weights or MCMC acceptance probabilities, leading to standard particle filtering methods or SMCMC procedures respectively. In either case, it is helpful to keep in view that the samples produced are *joint* samples approximating the posterior for all $\mathcal{T}_{1:k}$.

The posterior propagation proceeds by selecting one particle, say $p = \tilde{p}$, randomly from the smoothing distribution in Eq. (16) represented empirically by the 'history' collection at the end of interval $\mathcal{T}_{k-1}$. Based on the drawn particle $x_{1:k-1}^{\tilde{p}}$, we propose new sets of variables x_k from either priors or pre-assigned proposal distributions and compute the corresponding weight/acceptance ratio accordingly. Enough repetitions of this procedure during interval $\mathcal{T}_k$ will yield a set of importance-weighted/converged samples from the joint smoothing distribution $p(x_{1:k}|\lambda^*, \mathcal{T}_{1:k})$. Note that for mathematical convenience, we have treated the observed events $\{s_n; i_n = 0\}$ jointly with the latent events as random variables whose values are known with probability 1, and hence simply chosen deterministically in the proposal step. We note that the particle approximation of Eq. (17) approximates the joint distribution of input points $\{s_n; i_n = 0\}$ and all the remaining unknowns in the system. Since this joint distribution is directly proportional to the posterior distribution of the unknown state elements, conditional on a particular realisation of the input points (the 'data'), we obtain posterior Monte Carlo samples simply by extracting the Monte Carlo samples of the unknowns and excluding the known fixed input points. **Fig. 2** shows graphically the scheme of propagation as described in Eq. (17) across batches for different particles in the algorithm.

We have so far described a scheme that can be easily implemented with the variable-rate particle filter (VRPF) [4]. Such an approach however is not especially effective for this task as the factorisation in Eq. (14) requires the proposal of multiple latent variables of varying dimension in a single propagation step, which will inevitably result in the inherent weight degeneracy problem of the particle filter. Instead, we address this high-dimensional proposal with a SMCMC algorithm which targets sequentially the joint distributions of Eq. (17), using both local and global Metropolis-Hastings (MH) accept-reject moves instead of importance sampling or resampling [20], [14], [21], [22].

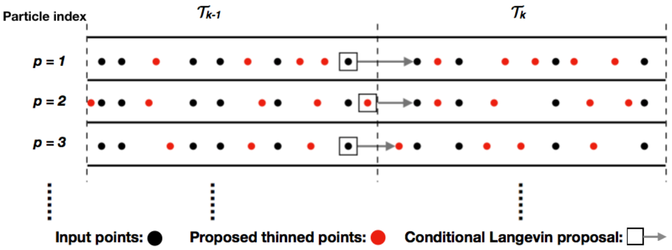


Fig. 2: Propagation scheme of the SMCMC algorithm across the batch boundary between $\mathcal{T}_{k-1}$ and $\mathcal{T}_k$. Number and locations of 'thinned' events are proposed independently for each particle p.

Furthermore, a mixture sampling procedure is adopted in this paper: at each MCMC iteration, a decision is made on performing either a joint MH proposal step with probability P_J or a sequence of individual refinement Metropolis-within-Gibbs (MwG) transitions with probability $1 - P_J$. Such a scheme provides an effective trade-off between the speed and accuracy of the inference via adjusting the value of P_J. In Section **IV-A**, we briefly cover the generic algorithm of SMCMC before proceeding to the detailed design of joint proposal and refinement steps in Section **IV-B**, **IV-C** and **IV-D**.

A. A Generic SMCMC Algorithm

The generic algorithm (SMCMC) also tries to approximate the posterior density with an empirical representation of the particles. However in contrast to a standard particle filter, the particles in SMCMC are not weighted and each is guaranteed to be a representative sample of the posterior density with the assumption that the MCMC run has converged.

Targeting the posterior joint density e.g. Eq. (17) at each propagation, the SMCMC algorithm selects historic particles from $\{x_{1:k-1}^p\}_{p=1}^{N_p}$ and conditionally proposes new latent variables x_k from one joint MCMC kernel or several conditional kernels in turn. The samples accepted after a burn-in period are included into the particle collection for the next propagation step requiring neither weight assignments nor a resampling step in the manner of the regular particle filter. However, we should note that any of the MCMC moves that involve proposing the sequence $x_{1:k-1}$ (or $x_{0:k-1}$ if involves a random initialisation), i.e. the discretely approximated ancestor states, are in some sense equivalent to a resampling operation. Thus we can expect some degree of path/time degeneracy in the SMCMC method, as for regular particle filtering. We are not aware of theory that proves which method would be less degenerate. However, we would postulate that the wide range of MCMC moves available in a single SMCMC scheme may improve on path degeneracy in the SMCMC case compared with particle filtering, although a full exploration of this is left for future work.

Both generic algorithms of particle filter and SMCMC are shown in **Algorithm 2** and **3** respectively. SMCMC may be favourable in the case of high-dimensional latent variables, as

Algorithm 2 A generic particle filter (PF) algorithm

```
Inputs: A set of (observed) events $\{s_k\}_{k=1}^{K}$ in $\mathcal{T}$.
Outputs: A collection of weighted particles $\{x_{0:K}^p\}_{p=1}^{N_p}$ with normalised weights $\{w_K^p\}_{p=1}^{N_p}$

Initialisation: ($k = 0$) Sample $N_p$ particles $\{x_0^p\}_{p=1}^{N_p}$ from the prior $p(x_0)$ and assign uniform weights $w_0^p = 1/N_p$.
for $k = 1 : K$ do
    for particle $p = 1 : N_p$ do
        Sample $x_k^p \sim q(x_k \,|\, x_{0:k-1}^p)$
        Compute weight $\widetilde{w}_k^p = w_{k-1}^p \times \frac{p(x_k^p \,|\, x_{0:k-1}^p)}{q(x_k^p \,|\, x_{0:k-1}^p)}$
    end for
    Normalise weights: $w_k^p = \widetilde{w}_k^p \,/\, \sum_{p'=1}^{N_p} \widetilde{w}_k^{p'}$
    Resample if necessary
end for
Return: $\{x_{0:K}^p\}_{p=1}^{N_p}$, $\{w_K^p\}_{p=1}^{N_p}$
```

Algorithm 3 A generic SMCMC algorithm

```
Inputs: A set of (observed) events $\{s_k\}_{k=1}^{K}$ in $\mathcal{T}$.
Outputs: A collection of (unweighted) particles $\Omega$.

Initialisation: ($k = 0$) Sample $N_p$ particles $\{x_0^p\}_{p=1}^{N_p}$ from the prior $p(x_0)$ to form the initial particle collection $\Omega_0$.
for $k = 1 : K$ do
    $\Omega_k = \emptyset$
    for iteration $p = 1 : (N_p + N_{\text{burn}})$ do
        Sample $x_{0:k}^* \sim q(x_k \,|\, x_{0:k-1})\, q(x_{0:k-1})$
        if $p = 1$ then
            $x_{0:k}^p = x_{0:k}^*$                                  ▷ Accept the initial condition
        else
            Compute $\rho = \min\Big\{1, \frac{p(x_{0:k}^*) q(x_k^{p-1} | x_{0:k-1}^{p-1})\, q(x_{0:k-1}^{p-1})}{p(x_{0:k}^{p-1}) q(x_k^* | x_{0:k-1}^*)\, q(x_{0:k-1}^*)}\Big\}$
            Draw $u \sim \text{Uniform}(0, 1)$
            if $u < \rho$ then                                      ▷ MH accept-reject
                $x_{0:k}^p = x_{0:k}^*$
            else
                $x_{0:k}^p = x_{0:k}^{p-1}$
            end if
        end if
        if $p > N_{\text{burn}}$ then
            $\Omega_k \leftarrow \Omega_k \cup x_k^p$
        end if
    end for
end for
Return: $\Omega = \Omega_0 \cup \Omega_1 \cup \cdots \cup \Omega_K$
```

in practice the proposal $q(x_k \,|\, x_{0:k-1})\, q(x_{0:k-1})$ will be split up into a number of blockwise Gibbs steps and Metropolis-within-Gibbs (MwG) moves. This allows us to adopt suitable MCMC schemes based on domain knowledge, which is likely to lead to better convergence [23]. In later sections, we will show how to design both joint MH samplers and MwG samplers to give good inference performance with the SMCMC algorithm.

B. Joint Proposal of Latent Variables

The first MCMC move described is a joint MH kernel that provides fast proposals of the 'new' latent variables x_k in each interval $\mathcal{T}_k$. This consists of a discrete uniform draw of the converged sample $x_{1:k-1}$ from the 'past' particle collection obtained from the previous step at the end of interval $\mathcal{T}_{k-1}$, followed by proposals of x_k conditioned on the sampled particle $x_{1:k-1}^p$.

More specifically, this latter proposal step is split into three sampling sub-steps, applied in sequence: 1) the total number of thinned events $\widetilde{M}$ in $\mathcal{T}_k$ sampled from a Poisson distribution; 2) the locations of thinned events $\{\tilde{s}_m\}_{m=1}^{\widetilde{M}}$ sampled uniformly within $\mathcal{T}_k$; and 3) the state vectors $\{\mathbf{g}\}_{\mathcal{T}_k}$ of all events (both observed and latent) in $\mathcal{T}_k$ sampled from the $\mathcal{LD}$ prior conditioned on the events' locations and the sampled particle $x_{1:k-1}^p$. This gives an overall proposal density as:

$$q_J(x_k) = \frac{\text{Poisson}(\widetilde{M} \,|\, \lambda^*, \mathcal{T}_k)}{|\mathcal{T}_k|^{\widetilde{M}}}\, \mathcal{LD}\big(\{\mathbf{g}\}_{\mathcal{T}_k} | \{\mathbf{g}\}_{\mathcal{T}_{k-1}}, \{s\}_{\mathcal{T}_k}\big) \tag{18}$$

Note that since the thinned events and input events jointly contribute to form the prior homogeneous Poisson process, there is no tractable prior distribution for the number of thinned events $\widetilde{M}$. However, we can still sample it from the Poisson distribution with the upper-bound intensity λ^* (or an arbitrary discounted intensity) as the MH acceptance probability will adjust for the values of $\widetilde{M}$ proposed. Incorporating Eq. (14), we can write down the MH acceptance probability for joint latent variable x_k at the pth MCMC iteration:

$$\rho_J = \min\Bigg\{1\,,\, \frac{(\lambda^*)^{N_k^*} \mathcal{LD}\big(\{\mathbf{g}\}_{\mathcal{T}_k}^* \,|\, \{\mathbf{g}\}_{\mathcal{T}_{k-1}}^*, \{s\}_{\mathcal{T}_k}^*\big)}{(\lambda^*)^{N_k^{p-1}} \mathcal{LD}\big(\{\mathbf{g}\}_{\mathcal{T}_k}^{p-1} \,|\, \{\mathbf{g}\}_{\mathcal{T}_{k-1}}^{p-1}, \{s\}_{\mathcal{T}_k}^{p-1}\big)} \times \frac{\prod_n \sigma\big\{(-1)^{i_n^*}\, g_{1,n}^*\big\}\; q_J(x_k^{p-1})}{\prod_n \sigma\big\{(-1)^{i_n^{p-1}}\, g_{1,n}^{p-1}\big\}\; q_J(x_k^*)}\Bigg\} \tag{19}$$

where the superscript '$*$' indicates the samples proposed in the current iteration and '$p-1$' indicates the previous iteration of the MCMC. Note that the nature (i.e. latent or observed) of the events are known a priori, hence the indicators $\{i\}_{\mathcal{T}_k}$ of the events in $\mathcal{T}_k$ are assigned determinstically with values of either 0 or 1.

As is common practice, it is suggested to take N_{burn} iterations before including any MCMC output into the new particle set, in order to neglect non-converged MCMC samples. **Algorithm 4** outlines the pseudo-code of the general scheme for performing SMCMC inference with the joint proposal. Tuning of the proposal intensity and alternative proposals incorporating domain knowledge could improve the convergence rate and inference performance, although this is not investigated here.

C. Refinement Metropolis-within-Gibbs (MwG)

A second step in the MCMC procedure involves refinement moves using MwG. The joint proposal of the previous section can sometimes result in low acceptance rate and consequently low particle diversity, which is analogous to the weight degeneracy encountered in importance sampling particle filters [24], although we stress that the SMCMC procedure operates entirely without particle weights. Therefore we here propose to use a MwG refinement step in conjunction with the joint MH kernel, as shown in **Algorithm 4**.

We choose to split the refinement moves into: a reversible-jump step for adding or removing thinned events and their positions; a MwG step for refining the positions of the thinned events; and finally a MwG step for moving the state vectors $\{\mathbf{g}\}_{\mathcal{T}_k}$ using a Metropolis-adjusted-Langevin-algorithm (MALA) procedure. While these three Gibbs sampling steps are likely to make smaller moves than those of the joint MH

Algorithm 4 SMCMC algorithm for sequential intensity inference

```
Inputs: A set of events $\{s_k\}_{k=1}^{K}$ in $\mathcal{T}$.
Outputs: Posterior filtering samples of underlying intensity.

Initialisation: ($k = 0$) Create a particle collection $\Omega_0$ of $N_p$ particles from the prior.
for batch $\mathcal{T}_k = \mathcal{T}_1 : \mathcal{T}_K$ do
    Initialise a new (empty) particle collection $\Omega_k = \emptyset$
    for iteration $p = 1 : (N_p + N_{\text{burn}})$ do
        if $p = 1$ then                                                        ▷ Initial condition
            Draw a sample $x_{k-1}^{*}$ discretely from collection $\Omega_{k-1}$
            Propose No. thinned points $\widetilde{M}^{*} \sim$ Poisson$\{\lambda^{*}|\mathcal{T}_k|\}$
            Propose positions of thinned points $\{\tilde{s}_m^{*}\}_{m=1}^{\widetilde{M}^{*}} \sim \widetilde{M}^{*} \times$ Uniform$(\mathcal{T}_k)$
            Propose $\{\mathbf{g}\}_{\mathcal{T}_k}^{*}$ from $\mathcal{LD}$ prior (6) conditioned on $x_{k-1}^{*}$
            $x_k^{p} = \{s, \mathbf{g}, i\}_{\mathcal{T}_k}^{*}$
        else
            $u \sim$ Uniform$(0, 1)$
            if $u < P_J$ then                                                  ▷ A joint proposal
                Draw a sample $x_{k-1}^{*}$ discretely from collection $\Omega_{k-1}$
                Propose No. thinned points $\widetilde{M}^{*} \sim$ Poisson$\{\lambda^{*}|\mathcal{T}_k|\}$
                Propose positions of thinned points $\{\tilde{s}_m^{*}\}_{m=1}^{\widetilde{M}^{*}} \sim \widetilde{M}^{*} \times$ Uniform$(\mathcal{T}_k)$
                Propose $\{\mathbf{g}\}_{\mathcal{T}_k}^{*}$ from $\mathcal{LD}$ prior (6) conditioned on $x_{k-1}^{*}$
                $x_k^{*} = \{s, \mathbf{g}, i\}_{\mathcal{T}_k}^{*}$
                Compute MH acceptance probability $\rho_J$ from Eq. (19)
                if Uniform$(0, 1) < \rho_J$ then
                    $x_k^{p} = x_k^{*}$                                        ▷ Accept proposed variables
                else
                    $x_k^{p} = x_k^{p-1}$                                      ▷ Reject proposed variables
                end if
            else                                                               ▷ Metropolis-within-Gibbs
                Perform MwG refinement moves
            end if
        end if
        if $p > N_{\text{burn}}$ then
            $\Omega_k \leftarrow \Omega_k \cup x_k^{p}$                        ▷ Include the converged sample into particle collection
        end if
    end for
end for
Map all posterior state vector samples to intensity $\lambda^{p}(s_k)$ with Eq. (12)
Return: Intensity samples $\{\lambda^{p}(s_k)\}_{p=1}^{N_p}$ at each input event $s_k$
```

sampler, they are also able to achieve more local exploration of the latent sample space through higher acceptance probabilities. A similar MwG construction was adopted in the non-sequential SGCP model of [9]. We now detail these three substeps.

1) Reversible-jump MCMC for $\widetilde{M}$: The value of $\widetilde{M}$ determines the number of thinned event locations and state vectors that need to be proposed in the other two MwG samplers. Therefore, we use the reversible-jump Markov chain Monte Carlo [25] algorithm to navigate the variable dimension of the sample space.

The sampler first makes a Bernoulli decision on whether to insert or delete a latent event. An insertion proposal q_{ins} consists of a uniform proposal of the event location $\tilde{s}'$ in $\mathcal{T}_k$, followed by a draw of its corresponding state vector $\mathbf{g}(\tilde{s}')$ from the $\mathcal{LD}$ prior conditioned on the state vectors of the two events immediately preceding and following $\tilde{s}'$, whilst a deletion proposal q_{del} simply consists of a uniform random selection and removal of an existing latent event, say $\tilde{s}_m$, out from a total of $\widetilde{M}$ events. Thus, we obtain the following proposal densities:

$$q_{\text{ins}}(\widetilde{M}+1 \leftarrow \widetilde{M}) = \frac{P_B}{|\mathcal{T}_k|}\,\mathcal{LD}(\mathbf{g}(\tilde{s}') \mid \tilde{s}', \{\mathbf{g}\}_{\mathcal{T}_k}) \tag{20}$$

$$q_{\text{del}}(\widetilde{M}-1 \leftarrow \widetilde{M}) = \frac{1-P_B}{\widetilde{M}} \tag{21}$$

where P_B is the Bernoulli probability of making an insertion move which we set to 0.5. Incorporating the joint recursion in (13) and (14), we obtain the acceptance ratio for both moves:

$$\rho_{\text{ins}} = \min\Big\{1\,,\ \frac{(1-P_B)\,|\mathcal{T}_k|\lambda^{*}}{P_B\,(\widetilde{M}+1)(1+\exp\{g_1(\tilde{s}')\})}\Big\} \tag{22}$$

$$\rho_{\text{del}} = \min\Big\{1\,,\ \frac{P_B\,\widetilde{M}\,(1+\exp\{g_1(\tilde{s}_m)\})}{(1-P_B)\,|\mathcal{T}_k|\,\lambda^{*}}\Big\} \tag{23}$$

Algorithm 5 shows the pseudo-code for performing one iteration of the reversible-jump move. It is found advisable to perform several iterations of this MH kernel before proceeding to the other two samplers.

2) Metropolis-Hastings for $\{\tilde{s}_m\}_{m=1}^{\widetilde{M}}$: Conditioned on the total number of thinned events $\widetilde{M}$, the posterior thinned event locations are sampled from a standard MH kernel. For each thinned event $\tilde{s}_m$, a new location $\tilde{s}_m'$ is proposed from a pre-assigned conditional transition kernel $q_{\text{loc}}(\tilde{s}_m' \leftarrow \tilde{s}_m)$ followed

Algorithm 5 Single-iteration reversible-jump MCMC for $\widetilde{M}$

```
Inputs: Event positions $\{s\}_{\mathcal{T}_k}$ and state vectors $\{\mathbf{g}\}_{\mathcal{T}_k}$ in $\mathcal{T}_k$; the number of thinned events $\widetilde{M}$
Outputs: Updated number of thinned events $\widetilde{M}$ (and corresponding event positions and state vectors).

Draw $u \sim$ Uniform$(0, 1)$
if $u < P_B$ then                                          ▷ Insertion
    Draw $\tilde{s}' \sim$ Uniform$(\mathcal{T}_k)$
    Draw $\mathbf{g}(\tilde{s}') \sim \mathcal{LD}(\mathbf{g}(\tilde{s}')|\tilde{s}', \{\mathbf{g}\}_{\mathcal{T}_k})$
    Compute $\rho_{\text{ins}}$ from Eq. (22)
    if Uniform$(0, 1) < \rho_{\text{ins}}$ then
        Accept $\tilde{s}'$ and $\mathbf{g}(\tilde{s}')$ as a new thinned event
        $\widetilde{M} = \widetilde{M} + 1$
    end if
else                                                       ▷ Deletion
    Draw $\tilde{s}_m$ discretely uniformly from $\{m = 1, 2, ..., \widetilde{M}\}$
    Compute $\rho_{\text{del}}$ from Eq. (23)
    if Uniform$(0, 1) < \rho_{\text{del}}$ then
        Remove $\tilde{s}_m$ and $\mathbf{g}(\tilde{s}_m)$ from the thinned events
        $\widetilde{M} = \widetilde{M} - 1$
    end if
end if
Return: $\widetilde{M}$
```

by a draw of the new state vector $\mathbf{g}(\tilde{s}'_m)$ at time $\tilde{s}'_m$ from the conditional Langevin prior:

$$\mathcal{LD}\Big(\mathbf{g}(\tilde{s}'_m) \mid \tilde{s}'_m,\ \{\mathbf{g}\}_{\mathcal{T}_k \backslash \mathbf{g}(\tilde{s}_m)}\Big) \tag{24}$$

where $\{\mathbf{g}\}_{\mathcal{T}_k \backslash \mathbf{g}(\tilde{s}_m)}$ stands for all state vectors in the batch $\mathcal{T}_k$ except for the one at $\tilde{s}_m$. We can therefore write out the acceptance probability as:

$$\rho_{\text{loc}} = \min\Big\{1\,,\ \frac{q_{\text{loc}}(\tilde{s}_m \leftarrow \tilde{s}'_m)(1 + \exp\{g_1(\tilde{s}_m)\})}{q_{\text{loc}}(\tilde{s}'_m \leftarrow \tilde{s}_m)(1 + \exp\{g_1(\tilde{s}'_m)\})}\Big\} \tag{25}$$

In the case where q_{loc} is symmetric, the acceptance probability is further reduced to the ratio of two sigmoidal thinning probabilities.

3) Metropolis-adjusted-Langevin-algorithm (MALA) for state vectors: Conditioned on the number and locations of the thinned events, we can also sample the posterior state vectors of all events within the batch. The exploration of the state vectors takes place in a multi-dimensional continuous space and hence requires a well-tuned sampling method to ensure fast convergence. Based on the conditional propagation equation shown in Eq. (14), we can write the *Log*-posterior of the state vector subject to an additive constant (normalising constant):

$$\begin{aligned}
&\mathcal{L}\Big(\{\mathbf{g}\}_{\mathcal{T}_k} \,|\, x_{k-1}, \{s\}_{\mathcal{T}_k}, \{i\}_{\mathcal{T}_k}, \lambda^*, \mathcal{T}_k\Big) \\
&= \ln\Big\{\mathcal{LD}(\{\mathbf{g}\}_{\mathcal{T}_k} | \{\mathbf{g}\}_{\mathcal{T}_{k-1}}, \{s\}_{\mathcal{T}_k})\Big\} \\
&\quad - \sum_{n=1}^{N_k} \ln\big[1 + (-1)^{i_n} \exp\{g_{1,n}\}\big] + \text{const.}
\end{aligned} \tag{26}$$

As Eq. (6) shows the conditional progression of the state vectors, one can concatenate $\{\mathbf{g}\}_{\mathcal{T}_k}$ into a $2N_k$-dimensional multivariate Gaussian vector $\mathbf{G}_k$:

$$p(\{\mathbf{g}\}_{\mathcal{T}_k} | \{\mathbf{g}\}_{\mathcal{T}_{k-1}}, \{s\}_{\mathcal{T}_k}) = \mathcal{N}(\mathbf{G}_k \mid \hat{\boldsymbol{\mu}},\ \hat{\boldsymbol{\Sigma}}) \tag{27}$$

which essentially allows further simplification over the *Log*-posterior:

$$\begin{aligned}
&\mathcal{L}\Big(\{\mathbf{g}\}_{\mathcal{T}_k} \,|\, x_{k-1}, \{s\}_{\mathcal{T}_k}, \{i\}_{\mathcal{T}_k}, \lambda^*, \mathcal{T}_k\Big) \\
&= -\frac{1}{2}(\mathbf{G}_k - \hat{\boldsymbol{\mu}})^T\, \hat{\boldsymbol{\Sigma}}^{-1} (\mathbf{G}_k - \hat{\boldsymbol{\mu}}) \\
&\quad - \sum_{n=1}^{N_k} \ln\big[1 + \exp\{(-1)^{i_n} g_{1,n}\}\big] + \text{const.}
\end{aligned} \tag{28}$$

We take advantage of the *Log*-gradient information and use MALA to accelerate the convergence by proposing from a gradient-adjusted transition kernel:

$$q(\mathbf{G}_k^* | \mathbf{G}_k^{p-1}) = \mathcal{N}\Big(\mathbf{G}_k^* \mid \mathbf{G}_k^{p-1} + \frac{\epsilon^2}{2}\Sigma\nabla\log\tilde{\pi}(\mathbf{G}_k^{p-1}),\ \epsilon^2\Sigma_{\text{c}}\Big) \tag{29}$$

where $\nabla\log\tilde{\pi}(.)$ is the gradient of Eq. (28), ϵ is the integration step size and Σ_{c} is a pre-defined (constant) covariance matrix. MALA is then completed with a standard accept/reject step with acceptance probability:

$$\rho_{\text{MALA}} = \min\Bigg\{1,\ \frac{\tilde{\pi}(\mathbf{G}_k^*) q(\mathbf{G}_k^{p-1} | \mathbf{G}_k^*)}{\tilde{\pi}(\mathbf{G}_k^{p-1}) q(\mathbf{G}_k^* | \mathbf{G}_k^{p-1})}\Bigg\} \tag{30}$$

Additional, we perform the gradient calculation and MALA diffusion over the 'whitened' space of the variable $\mathbf{G}_k$. This is achieved by applying Cholesky decomposition on the precision matrix $\hat{\boldsymbol{\Sigma}}^{-1} = \mathbf{L}\mathbf{L}^T$ and rewrite the *Log*-posterior of Eq. (28) as:

$$\begin{aligned}
&\mathcal{L}\Big(\{\mathbf{g}\}_{\mathcal{T}_k} \,|\, x_{k-1}, \{s\}_{\mathcal{T}_k}, \{i\}_{\mathcal{T}_k}, \lambda^*, \mathcal{T}_k\Big) \\
&= -\frac{1}{2}(\mathbf{G}_k - \hat{\boldsymbol{\mu}})^T\, \mathbf{L}\mathbf{L}^T (\mathbf{G}_k - \hat{\boldsymbol{\mu}}) \\
&\quad - \sum_{n=1}^{N_k} \ln\big[1 + \exp\{(-1)^{i_n} g_{1,n}\}\big] + \text{const.} \\
&= -\frac{1}{2}(\mathbf{G}_k^w - \hat{\boldsymbol{\mu}}^w)^T\, (\mathbf{G}_k^w - \hat{\boldsymbol{\mu}}^w) \\
&\quad - \sum_{n=1}^{N_k} \ln\Big[1 + \exp\big\{(-1)^{i_n} [\mathbf{L}^{-T}\mathbf{G}_k^w]_{2n-1}\big\}\Big] + \text{const.}
\end{aligned} \tag{31}$$

where $\mathbf{G}_k^w = \mathbf{L}^T\mathbf{G}_k$, $\hat{\boldsymbol{\mu}}^w = \mathbf{L}^T\hat{\boldsymbol{\mu}}$ and the subscript $2n-1$ represents the $(2n-1)$th element of the $2N_k$-dimensional vector. This allows us to carry out the same MH routine of MALA on the whitened variable $\mathbf{G}_k^w$ instead of $\mathbf{G}_k$, which gives a better-conditioned covariance matrix $(\mathbf{I})$ and fastens the convergence.

D. Refinement with Rejection Sampling (RS)

Recall that we can derive the thinning operation in point process from rejection sampling, in this paper we also try to use RS as a possible refinement approach to obtain the posterior approximation of latent variables. RS is a technique used to generate samples from distributions that cannot be sampled directly. Denoting the target density as $f(x)$ and the proposal density as $q(x)$, the rejection sampling compute the acceptance probability $\rho(x)$ as:

$$\rho(x) = \frac{f(x)}{B \times q(x)} \tag{32}$$

Algorithm 6 Rejection Sampling

```
**Inputs:** target density $f(x)$, proposal density $q(x)$, bound $B$
**Outputs:** $X$ as a sample from $f(x)$

flag = False
while not flag do
    Draw $X \sim q(x)$
    Compute $\rho(X) = \frac{f(X)}{B \times q(X)}$
    Draw $u \sim \text{Uniform}(0, 1)$
    if $u < \rho(X)$ then
        Accept sample $X$
        flag = True
    end if
end while
Return: $X$
```

where $1 \leq B < \infty$ is a finite bound over the probability ratio $f(x)/q(x)$, i.e. $f(x) \leq Bq(x), \forall x$. Therefore RS is usually more restrictive to use than other MCMC methods as it requires the calculation of a tractable bound B. However, RS does not require any burn-in for convergence as the generated samples are guaranteed to come from the target distribution. **Algorithm 6** shows the standard RS procedure to generate one sample from the target distribution, the general structure of which is used repeatedly in the refinement step below.

To apply RS in the refinement step, we again use three separate rejection samplers in a Gibbs manner as in MwG: (i) sampling the number thinned events $\widetilde{M}$; (ii) sampling the location of the thinned events $\{\tilde{s}_m\}_{m=1}^{\widetilde{M}}$; and (iii) sampling the state vectors of all events in the batch $\{\mathbf{g}\}_{\mathcal{T}_k}$.

(i): Denoting the number of observed events in the batch $\mathcal{T}_k$ as $\hat{K}$, these together with the $\widetilde{M}$ thinned events should constitute a HPP with counting measure $N(\mathcal{T}_k) \sim \text{Poisson}(\lambda^*|\mathcal{T}_k|)$. The target density can thus be written as:

$$f(\widetilde{M}) = \text{Poisson}\Big((\widetilde{M} + \hat{K}) \mid \lambda^*|\mathcal{T}_k|\Big) \tag{33}$$

With a simple proposal of $\widetilde{M}$ from a discretely uniform distribution with $G \in \mathbb{N}^+$ bins i.e. $\{0, 1, 2..., G{–}1\}$, the bound $B(\widetilde{M})$ and corresponding acceptance ratio $\rho(\widetilde{M})$ are:

$$B(\widetilde{M}) = \frac{(\lambda^*|\mathcal{T}_k|)^{\lfloor \lambda^*|\mathcal{T}_k| \rfloor} \exp\{-\lambda^*|\mathcal{T}_k|\}\, G}{(\lfloor \lambda^*|\mathcal{T}_k| \rfloor)\, !} \tag{34}$$

$$\rho(\widetilde{M}) = \frac{\lfloor \lambda^*|\mathcal{T}_k| \rfloor\, ! \times (\lambda^*|\mathcal{T}_k|)^{\widetilde{M} + \hat{K} - \lfloor \lambda^*|\mathcal{T}_k| \rfloor}}{(\widetilde{M} + \hat{K})\, !} \tag{35}$$

Note that unlike the corresponding MH sampler in MwG, the described rejection sampler does not propose locations and state vectors of the $\widetilde{M}$ thinned events.

(ii): As it is impossible to 'fill' thinned events into a NHPP to make it homogeneous without knowledge of intensity function (or state vectors), we propose jointly the location $\tilde{s}_m$ and the state vector $\mathbf{g}(\tilde{s}_m)$ for $\widetilde{M}$ times conditioned on the existing state vectors in the batch (for input points and already proposed latent points).

To this end, we can write the target density and the proposal density respectively as:

$$\begin{aligned} f\big(\tilde{s}_m,\ \mathbf{g}(\tilde{s}_m)\big) =& \frac{1}{|\mathcal{T}|} \times \mathcal{LD}\Big\{\mathbf{g}(\tilde{s}_m) | \tilde{s}_m, \{\mathbf{g}\}_{\mathcal{T}_k}\Big\} \\ & \times \sigma(-g_1(\tilde{s}_m)) \end{aligned} \tag{36}$$

$$q\big(\tilde{s}_m,\ \mathbf{g}(\tilde{s}_m)\big) = \frac{1}{|\mathcal{T}|} \times \mathcal{LD}\Big\{\mathbf{g}(\tilde{s}_m) | \tilde{s}_m, \{\mathbf{g}\}_{\mathcal{T}_k}\Big\} \tag{37}$$

As the additional term in the target density is always less than 1, the proposal itself is already a finite bound of the target with $B = 1$, giving the acceptance ratio $\rho\big(\tilde{s}_m, \mathbf{g}(\tilde{s}_m)\big) = \sigma(-g_1(\tilde{s}_m))$

(iii): With the locations of both input and latent events in the batch fixed, the proposal of state vectors $\{\mathbf{g}\}_{\mathcal{T}_k}$ can simply be the conditional prior of the Langevin dynamics. In this case, we achieve a simplified acceptance ratio as the product of Bernoulli probability of each event, with the bound B equal to unity. Hence, one joint proposal of $\{\mathbf{g}\}_{\mathcal{T}_k}$ will give an acceptance probability:

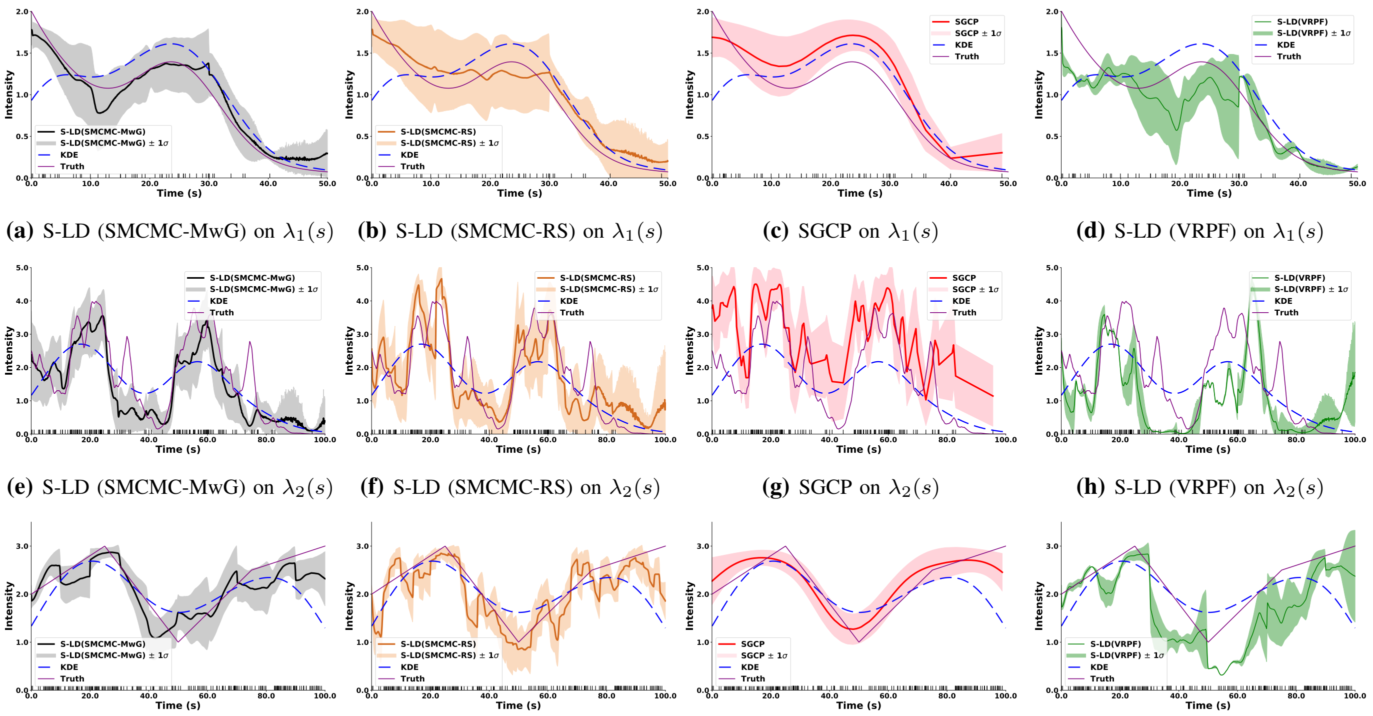
$$\rho(\{\mathbf{g}\}_{\mathcal{T}_k}) = \prod_{n: s_n \in \mathcal{T}_k} \sigma\{(-1)^{i_n} g_{1,n}\} \tag{38}$$

It is worth noting that the product of multiple sigmoid functions could result in an extremely low acceptance ratio which stagnates the algorithm and increases computation. To work around this issue, the rejection sampler is applied individually to each state vector conditional on all others e.g. propose from the conditional prior as in Eq. (24). Such rejection sampler provides a better acceptance ratio as a single sigmoid function at the cost of reduced mixing among state vectors.

Furthermore, this RS process can be regarded as the reverse of the thinning operation: instead of deciding whether the point should be thinned or retained given intensity, we now try to find the appropriate intensity (state vector) value given the fact that a point is either latent ($i_n = 1$) or observed ($i_n = 0$).

E. Sequential Batch Scheme

With the interval $\mathcal{T}_k$ defined earlier in this section, the choice of how to delineate the entire domain of interest is largely arbitrary. However, the most intuitive choice that fixes these intervals to correspond to the observed event times, may not yield best algorithmic performance. We propose the use of regular sized batches of duration δT in this paper. With the appropriate choice of δT, the scheme recovers the temporal correlation among points within the same batch and thus tends to improve the sequential inference accuracy. Moreover, the batch scheme provides the possibility to replace the global maximum intensity λ^* with maxima λ_k^* that vary with batch number k and these can be updated individually in a Gibbs manner with a Gamma prior specified for each λ_k^*. The Gibbs conditional parameters for the posterior are: $\alpha_{\text{post}} = \alpha_{\text{prior}} + N_k$, $\beta_{\text{post}} = \beta_{\text{prior}} + |\mathcal{T}_k|$. This local maximum intensity considerably reduces the number of latent variables proposed for inference and thus enables computational savings.



(a) S-LD (SMCMC-MwG) on $\lambda_1(s)$ **(b)** S-LD (SMCMC-RS) on $\lambda_1(s)$ **(c)** SGCP on $\lambda_1(s)$ **(d)** S-LD (VRPF) on $\lambda_1(s)$

(e) S-LD (SMCMC-MwG) on $\lambda_2(s)$ **(f)** S-LD (SMCMC-RS) on $\lambda_2(s)$ **(g)** SGCP on $\lambda_2(s)$ **(h)** S-LD (VRPF) on $\lambda_2(s)$

(i) S-LD (SMCMC-MwG) on $\lambda_3(s)$ **(j)** S-LD (SMCMC-RS) on $\lambda_3(s)$ **(k)** SGCP on $\lambda_3(s)$ **(l)** S-LD (VRPF) on $\lambda_3(s)$

Fig. 3: Figure shows the intensity inference results estimated using different methods on three synthetic datasets. True intensity curves and KDE results are displayed in all panels. Three Bayesian approaches are shown individually with corresponding means in solid lines and $\pm 1\sigma$ (68%) confidence intervals as shaded regions. The occurrences of input events are shown as scatter lines on the x-axes

TABLE I: Hyperparameters settings for each inference approach on synthetic datasets

	S-LD (SMCMC)	**SGCP**	**S-LD (VRPF)**
$\boldsymbol{\lambda_1(s)}$	$\theta = -0.7$, $\sigma = 0.5$, $K_{\text{MwG}} = K_{\text{RS}} = 5$, $P_j = 0.5$, $N_{\text{burn}} = 200$, $N_p = 200$	$N_{\text{iter}} = 400$, $N_{\text{burn}} = 200$, $l_k = 2.0$	$\theta = -0.7$, $\sigma = 0.5$, $K = 5$, $N_p = 800$
$\boldsymbol{\lambda_2(s)}$	$\theta = -0.5$, $\sigma = 0.8$, $K_{\text{MwG}} = K_{\text{RS}} = 20$, $P_j = 0.7$, $N_{\text{burn}} = 200$, $N_p = 200$	$N_{\text{iter}} = 400$, $N_{\text{burn}} = 200$, $l_k = 1.0$	$\theta = -0.5$, $\sigma = 0.8$, $K = 20$, $N_p = 800$
$\boldsymbol{\lambda_3(s)}$	$\theta = -0.2$, $\sigma = 0.2$, $K_{\text{MwG}} = 10$, $K_{\text{RS}} = 50$, $P_j = 0.5$, $N_{\text{burn}} = 200$, $N_p = 200$	$N_{\text{iter}} = 400$, $N_{\text{burn}} = 200$, $l_k = 15.0$	$\theta = -0.2$, $\sigma = 0.2$, $K = 10$, $N_p = 1500$

TABLE II: Numerical results for models. **Bold** is the best.

		S-LD (SMCMC-MwG)	S-LD (SMCMC-RS)	KDE	SGCP	S-LD (VRPF)
$\lambda_1(s)$	MSE	**0.0257**	0.0342	0.129	0.0704	0.187
	$\mathcal{L}(p)$	**1.825**	-6.379	–	-9.440	-5498
	Time (s)	15.86	36.46	**0.01**	60.23	14.89
$\lambda_2(s)$	MSE	0.6531	**0.6018**	0.8599	1.5257	1.7004
	$\mathcal{L}(p)$	-248.1	**-233.85**	–	-326.6	-9.3×10^{34}
	Time (s)	60.05	490.4	**0.05**	1326.28	64.25
$\lambda_3(s)$	MSE	0.0986	0.1157	0.2166	**0.0637**	0.4286
	$\mathcal{L}(p)$	-69.20	-74.54	–	**-28.34**	-5.93×10^{29}
	Time (s)	100.3	125.3	**0.05**	522.2	98.38

V. Results and Discussion

In this section, we present empirical performance analysis of the proposed sequential-Langevin (S-LD) model. Section **V-A** assesses the relative performance of S-LD, SGCP and a baseline kernel density estimation (KDE) method [7] on three synthetic datasets of distinct intensity functions $\lambda(s)$ with ground truth available. In the same section, the S-LD model and the KDE approach are further tested on 4 realisations of the doubly-stochastic Poisson process which are found computationally challenging for the SGCP model. The S-LD model is then applied to a real financial dataset with high-frequency input events in Section **V-B**. Finally, we examine the convergence behaviour of the SMCMC algorithm under different refinement schemes (MwG or RS); and the effect of hyperparameters on the S-LD model performance.

A. Synthetic Data

Three sets of one-dimensional data are generated using **Algorithm 1** with the following intensity functions:

1) A sum of an exponential and a Gaussian bump: $\lambda_1(s) = 2\exp\{-s/15\} + \exp\{-((s-25)/10)^2\}$ on the interval [0, 50] with 55 events.
2) A doubly-stochastic process with $\lambda_2(s)$ governed by Langevin dynamics with parameters $\theta = -0.5$, $\sigma = 0.5$ and $\lambda^* = 5$ on interval [0, 100] with 156 events.
3) A piece-wise linear intensity function $\lambda_3(s)$ on interval [0, 100] with 230 events

Synthetic datasets similar to 1) and 3) were also used in the original SGCP paper [9]; while 2) is the dataset generated from the matching prior model. Furthermore, the three synthetic intensity functions also test the models' ability in generalising to underlying intensities not drawn from the assumed prior structure of the model.

In these experiments, we infer the S-LD model intensity functions using both the proposed SMCMC algorithm and a batch-based variable-rate particle filter (VRPF). Additionally, the SMCMC algorithm is tested separately using both MwG refinement and RS refinement schemes. The number of particles used in VRPF is tuned to roughly match the computational cost of SMCMC algorithm. The results are compared with those obtained by the SGCP model using a square-exponential covariance function (with lengthscale l_k) and by the KDE approach with Gaussian smoothing kernel [26]. **Table I** lists the hyperparameter values used by each approach, except for KDE, for all three synthetic cases. The hyperparameters are heuristically tuned to provide representative results for comparison purpose. **Fig. 3** shows the graphical results of the four approaches and **Table II** quantitatively reports the performance averaged across 10 trials in terms of the computational time, the mean squared error (MSE) to the true intensity function, and a probabilistic metric $\mathcal{L}(p)$. The log-probability $\mathcal{L}(p)$ is computed as follows:

$$\mathcal{L}(p) = \sum_{k=1}^{K} \log\Big\{\mathcal{N}\big(\lambda(s_k)\,|\,\hat{\mu}_k, \hat{\sigma}_k^2\big)\Big\} \tag{39}$$

where $\lambda(s_k)$ is the true intensity value at s_k; $\hat{\mu}_k$ and $\hat{\sigma}_k^2$ are mean and variance empirically approximated by the particles obtained from the inference algorithm. In addition to the mean error, $\mathcal{L}(p)$ also quantifies the uncertainty of different Bayesian approaches.

Inferred with SMCMC, the proposed S-LD model is able to outperform the other two models in both MSE and $\mathcal{L}(p)$ for the first two synthetic datasets while giving satisfactory accuracy for the third dataset. VRPF on the other hand fails to provide good inference for the S-LD model due to particle weight degeneracy as discussed in Section **IV**, which can be seen from the extremely low $\mathcal{L}(p)$ values. Computationally, the KDE always gives the fastest run-time because of its algorithmic simplicity, but gives no confidence intervals as a frequentist approach. The SGCP model is significantly more expensive and scales poorly with λ^*, which can be observed from the results for datasets 2 and 3. Comparing between MwG refinement and RS refinement, both schemes give similar estimation accuracy whilst MwG refinement requires lower computational cost than RS. It is typically more difficult to maintain a salable computational cost using RS algorithms due to its inherent sampling mechanism. The MwG refinement on S-LD model shows roughly linear computational cost with the number of input events.

Visually from **Fig. 3**, despite the prior mismatches for $\lambda_1(s)$ and $\lambda_3(s)$ compared with the model used in inference, the S-LD model inferred with both refinements of the SMCMC algorithm still captures the overall shape of the true intensity function and gives a reasonable estimate of uncertainty. The SGCP model on the other hand, was found to be sensitive to the choice of hyperparameters e.g. different forms of covariance functions and values of lengthscale. The KDE method tends to over-smooth the intensity and ignores short-term variations. From the figure, one can again notice the degeneracy in particles for the S-LD(VRPF) as it provides overly narrow confidence intervals.

We also apply the S-LD model (inferred by SMCMC-MwG) to additional 4 realisations of the doubly-stochastic Poisson process with the same parameters used to generate $\lambda_2(s)$. **Fig. 4** shows the results obtained using the same set of inference hyperparameters for $\lambda_2(s)$ as listed in **Table I**. The S-LD model is compared to KDE only as the SGCP model was found to be impractically slow. In terms of both accuracy and computational time, the results are consistent with those obtained from earlier three experiments on synthetic datasets.

B. Application to Order Book Data

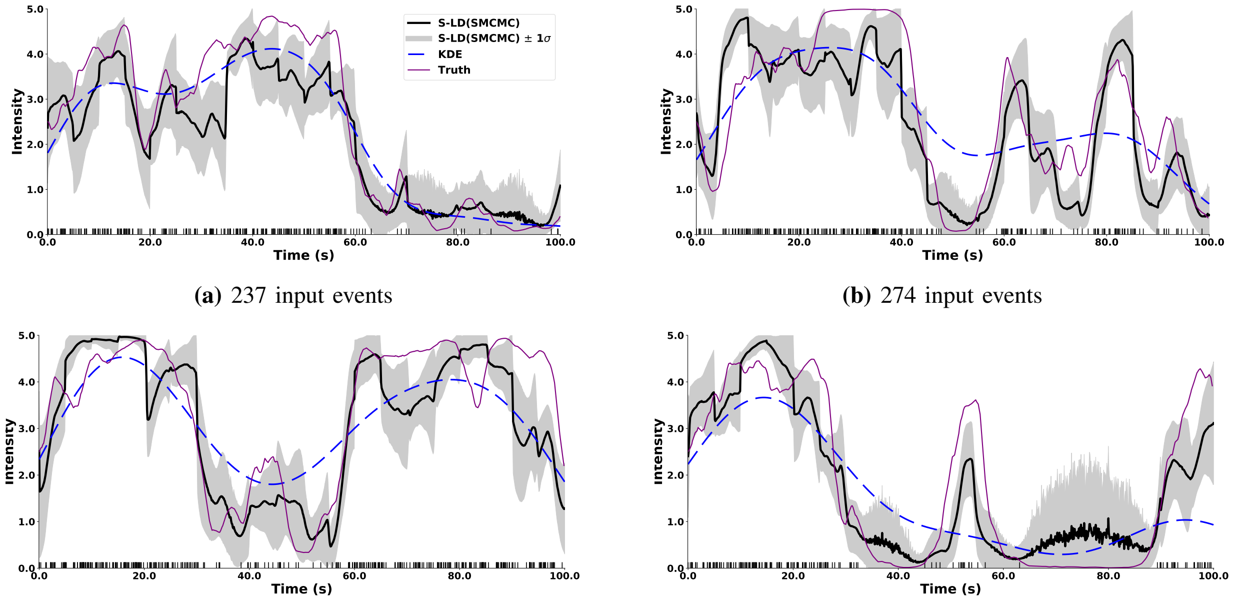
Limit order books [27] in modern financial markets, record and display order operations performed by market participants all over the world. With momentum strategy as a common technique used by the traders in high-frequency finance [28], being able to infer the intensities of limit order arrivals, cancellations and executions provides crucial insights into the future market structure and price trends. This demands the development of a computational-efficient online intensity inference method for market analysis based on time-of arrival trading data.

We apply the S-LD model on a set of LOB data collected from the EUR-USD FOREX market on the 2nd of September 2015[1]. The tick data used in the experiments is a record of all limit order arrivals at 51 different price levels ('ticks') around the mid-price for a duration of 5 minutes (19:35–19:40) from one of the busiest hours of the day.

We construct 51 independent S-LD models for the 51 price levels of interest. A volatile Langevin prior with $\sigma = 1.0$, $\theta = -0.7$ and $\lambda^* = 5.0$ is assigned to all models to accommodate possible drastic changes of the intensity curve in the highly stochastic market. The SMCMC algorithm with MwG refinement is used for inference with $P_J = 0.7$, $K = 20$, $N_p = 400$ and $N_{\text{burn}} = 400$ to ensure convergence.

Fig. 5 shows the inference results presented as a 3D surface plot and a heatmap. Both plots exhibit reasonable behaviours

[1] The authors would like to thank Cambridge Capital Management for providing the datasets for these experiments.



(**a**) 237 input events

(**b**) 274 input events

(**c**) 353 input events

(**d**) 180 input events

Fig. 4: Additional tests of the S-LD model on doubly-stochastic Poisson processes. In comparison to KDE, the S-LD model is able to achieve an average MSE of **0.6806** and an average computational time of **0.863**s per input point; while KDE only provides an average MSE of **1.032** with an average computational time of $\mathbf{1.576 \times 10^{-4}}$s per input point.

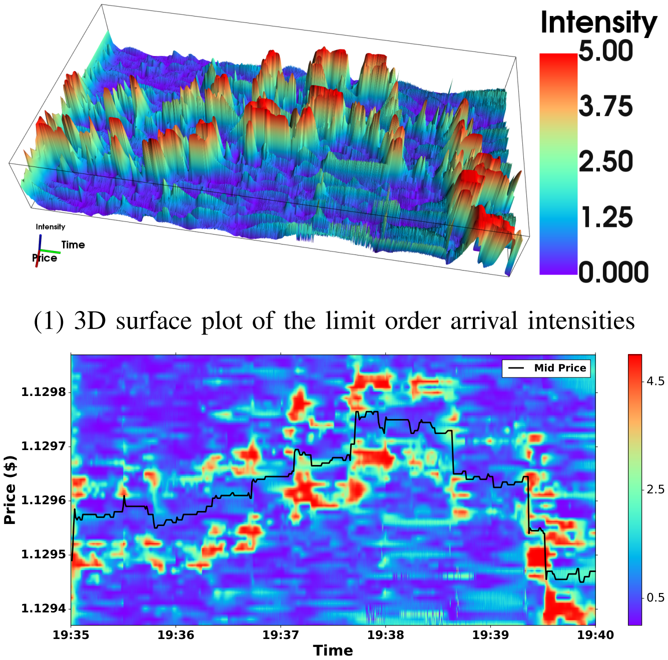
with the mid-price lying in a ‘valley’ formed between the high-intensity ridges and peaks at prices above and below the mid price: this price region close to the mid-price is where the market orders are typically placed and matched (executed) immediately, presenting little interest for limit-order traders. In contrast, the limit order arrivals have high intensity a few *ticksizes* away from the mid-price as these levels are most likely to become the best prices in future market fluctuations. We can also observe from the heatmap that the high intensity of bid or ask arrivals exerts pressure on the mid-price to go in the opposite direction, as would be expected from the market supply-demand relationship.

Fig. 6 presents a detailed view of the inferred arrival intensity at the price of $1.12960 with a reference to the mid-price shown in the bottom panel. The figure again demonstrates how intensity changes in relation to the mid-price movements and during ask-bid transitions.

Based on the results obtained from this example, it is reasonable to conclude that the proposed S-LD model may be useful for providing predictive inference about market behaviours, although we leave a full investigation of this for future work. The SGCP model was not tested on this dataset due to its impractical computational cost.

C. Convergence Evaluation

When it comes to an iterative sampling method such as SMCMC, it is important to ensure its convergence while maintaining reasonable computational efficiency. In this section, we evaluate the convergence behaviours of different SMCMC refinement setups by computing their corresponding integrated autocorrelation times (IACTs).



(1) 3D surface plot of the limit order arrival intensities

(2) 2D intensity heatmap

Fig. 5: The top panel shows the surface plot of the inferred (filtering) intensity with time, price and intensity on x, y and z axes respectively; the bottom panel shows the intensity heatmap with market mid-price plotted in black solid line.

The IACT for a sequence $f(t)$ is defined as:

$$\tau_f = \sum_{k=-\infty}^{\infty} \rho_f(k) = 1 + 2\sum_{k=1}^{\infty} \rho_f(k) \tag{40}$$

TABLE III: IACT values computed from RMSE sequence and intensity sequences obtained from 6000 iterations of SMCMC run on $\lambda_1(s)$. Intensity IACT values are averaged across all input points/events.

Refinement method		*Joint move ratio*		
		$\boldsymbol{P_J = 0.1}$	$\boldsymbol{P_J = 0.5}$	$\boldsymbol{P_J = 0.9}$
MCMC (MwG)	RMSE IACT	37.23	7.46	18.37
	Intensity avg. IACT	43.91	15.88	29.88
RS	RMSE IACT	49.17	18.92	12.97
	Intensity avg. IACT	50.08	15.24	15.08

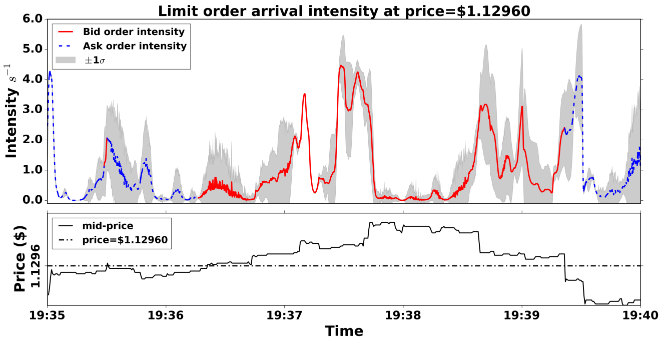


Fig. 6: Top panel shows the S-LD model intensity inference result on limit order (both bid and ask) arrival data at price $1.12960. Bottom panel shows the market mid-price for the same duration.

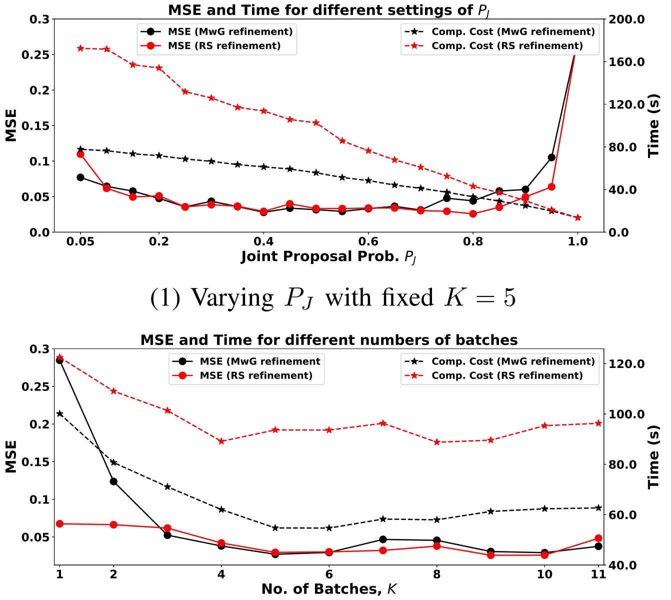
where $\rho_f(k)$ is the normalised autocorrelation function (ACF):

$$\rho_f(k) = \frac{\mathbb{E}\big[\big(f(t) - \mu_f\big)\big(f(t+k) - \mu_f\big)\big]}{\sigma_f^2} \tag{41}$$

with μ_f and σ_f^2 being the mean and variance of the sequence. With $f(t)$ being the sequence output (e.g. intensities) from the SMCMC algorithm, τ_f quantifies the factor by which MCMC chain's Monte Carlo error is degraded comparing to standard i.i.d. Monte Carlo from the target posterior distribution. Therefore, the SMCMC refinement setup with better mixing and faster convergence would give a smaller value of τ_f. We approximate the IACT with a finite summation using the method suggested in [29].

Running the SMCMC algorithm for 6000 iterations (i.e. particles+burn-in) on synthetic dataset $\lambda_1(s)$, **Table III** reports the IACTs of root-mean-square errors (RMSE) and intensities for different configurations of refinement step. Both RS and MCMC refinement show relatively poor convergence at low P_J due to the lack of resampling of the 'ancestor' particle. At high P_J, MCMC suffers from inadequate mixing of Markov chains (i.e. MwG); while RS provides lower IACTs as it is guaranteed to produce representative posterior samples and it only requires convergence in the conditional Gibbs step. An intermediate value of P_J yields the best result for MCMC and outperforms all other configurations.

In summary, the refinement moves used in the SMCMC algorithm provide improved mixing of the Markov chains without dramatic increase in computation. Meanwhile, it is also necessary to maintain enough resampling of the 'ancestor' particles from the previous batch. Each sampling step of the RS refinement scheme provides better/more representative posterior samples, which means that the RS refinement can achieve good mixing of the Markov chains without requiring a large amount of expensive Gibbs-like refinements. The overall low values of IACT computed in this experiment suggest that the proposed SMCMC algorithm requires only a moderate amount of particles and burn-in for inference.



(1) Varying P_J with fixed $K = 5$

(2) Varying K with fixed $P_J = 0.5$

Fig. 7: The S-LD model performances under different settings of hyperparameters using synthetic dataset $\lambda_1(s)$

D. Hyperparameter Settings

Tuning of hyperparameters is crucial in Bayesian inference and learning. The hyperparameters in the SSM determine loosely the prior dynamics of the state vector diffusion, and hence case-specific domain knowledge should be incorporated to improve the fit of the prior model to the data. Alternatively, SSM hyperparameters can be learned directly from the data with an extension of a variational structure [30] or particle-MCMC methods [31]. In this section however, we present a focused analysis on the algorithm-related hyperparameters.

As introduced in **Algorithm 4**, the SMCMC inference routine is controlled by two hyperparameters: joint proposal ratio P_J and batch size $\mathcal{T}_k$. Both values affect the inference accuracy and computational speed. We run the S-LD model on the same set of NHPP realisations from $\lambda_1(s)$ with the same SSM settings described in **Table I**. Algorithm-wise, we use both MwG and RS refinement while setting $N_p = 200$ and $N_{\text{burn}} = 800$, which constitute to a total of 1000 iterations to ensure convergence. Each result presented here is averaged across 10 random runs of the SMCMC algorithm.

Fig. 7 shows two plots of computational cost and MSEs under a range of values of P_J and K. The top panel shows that the inference time reduces linearly with the increase of P_J as both MwG and RS refinements require more computation especially in the MALA step and rejection sampling steps. MSE shows weaker correlation with P_J but deteriorates drastically without refinement (i.e. $P_J = 1$), which emphasises the importance of refinement steps in the SMCMC algorithm. At low values of P_J, the MSEs for both refinement schemes rise slightly due to the lack of resampling of 'ancestor' particle. Despite the small difference, RS refinement gives lower MSEs at high values of P_J which supports the findings in Section **V-C**.

With MALA being the computational bottleneck of MwG refinement, the sequential batch scheme changes the complexity from $\mathcal{O}(N^3)$ to roughly $\mathcal{O}(\frac{N^3}{K^3})\times\mathcal{O}(K)$. This gives obvious drops in computational cost especially upon using the batch scheme (i.e. K changes from 1 to 2) as shown in the bottom plot of **Fig. 7**. RS refinement's cost is relatively less sensitive to this change as RS does not involve the $\mathcal{O}(N^3)$ matrix inversion. However, the low acceptance rate of the rejection sampler still renders it slower than MwG. The cost later increases slightly with K, as the $\mathcal{O}(K)$ part becomes more dominant. For MwG refinement, MSE generally improves with an increasing K value because the sequential batch scheme reduces the dimension of latent variables in each batch and hence improves MCMC samplers' performance. However, the rising trend in MSE that is just observable as K increases becomes more acute at larger K values (not shown on the plot), as the batches tend to de-correlate local location information of the input points. As for RS refinement, the problem of high-dimensional latent space is primarily reflected in the high computational cost (i.e. low acceptance probability in rejection samplers) and hence the small value of K has little influence on its MSE. But the de-correlation of input points will also have a negative impact on the MSEs for RS refinement at larger K values.

Based on above analyses, we can also conclude that despite higher computational cost, RS refinement is able to provide more robust inference results across a range of algorithmic tuning hyperparameters. A drawback of the current RS refinement scheme is its relative slowness compared to MwG. This could potentially be overcome by using more sophisticated proposals such as adaptive rejection sampling (ARS) [32], [33] and its variants. In practice, one may choose to use a mixture of both RS and MwG refinement moves to achieve the best trade-off between computational time per iteration and convergence over iterations.

VI. Summary and Future Work

In this paper, we have presented a novel approach of modelling the intensity function of a NHPP with a continuous-time SSM. In addition to using a generative prior and latent variables to mitigate the inherent intractability of NHPP, we further utilised the Markvoian property of the SSM and performed sequential Bayesian inference for the intensity function with a novel design of SMCMC algorithm. The proposed algorithm not only dealt with the degeneracy problem caused by high-dimensional latent variables, but was also favourable in practical applications that require online intensity estimations.

We also proposed two refinement schemes (MwG and RS) and a sequential batch scheme to further improve the inference performance by increasing Markov chain mixing. In comparison with the KDE and SGCP approaches on synthetic datasets, our model has demonstrated better inference accuracy and reasonable computational cost while maintaining a fully Bayesian framework. The results obtained using FOREX data again demonstrated that the proposed model is capable of handling real-world intensity inference tasks while giving plausible interpretations of the data.

In our current work we are investigating automated hyperparameter learning for the SSM and also extensions to our models for multiple correlated point processes by encapsulating them into a single SSM (see also [13] and [11]), which would be highly beneficial in many applications such as the financial order book examples considered earlier.

References

[1] D. Perkel, G. Gerstein, and G. Moore, "Neuronal spike trains and stochastic point processes: II. Simultaneous spike trains," *Biophysical journal*, vol. 7, no. 4, pp. 419–440, 1967.

[2] JK Gardner and L Knopoff, "Is the sequence of earthquakes in Southern California, with aftershocks removed, Poissonian?," *Bulletin of the seismological society of America*, vol. 64, no. 5, pp. 1363–1367, 1974.

[3] M. Avellaneda and S. Stoikov, "High-frequency trading in a limit order book," *Quantitative Finance*, vol. 8, no. 3, pp. 217–224, 2008.

[4] S. Godsill and J. Vermaak, "Variable rate particle filters for tracking applications," in *IEEE/SP 13th Workshop on Statistical Signal Processing, 2005*. IEEE, 2005, pp. 1280–1285.

[5] H. Christensen, J. Murphy, and S. Godsill, "Forecasting high-frequency futures returns using online Langevin dynamics," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 4, pp. 366–380, 2012.

[6] D. Cox and V. Isham, *Point processes*, vol. 12, CRC Press, 1980.

[7] P. Diggle, "A kernel method for smoothing point process data," *Applied statistics*, pp. 138–147, 1985.

[8] W. Massey, G. Parker, and W. Whitt, "Estimating the parameters of a nonhomogeneous Poisson process with linear rate," *Telecommunication Systems*, vol. 5, no. 2, pp. 361–388, 1996.

[9] R. Adams, I. Murray, and D. MacKay, "Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 9–16.

[10] C. Lloyd, T. Gunter, M. Osborne, and S. Roberts, "Variational inference for Gaussian process modulated Poisson processes," in *International Conference on Machine Learning*, 2015, pp. 1814–1822.

[11] T. Gunter, C. Lloyd, M. Osborne, and S. Roberts, "Efficient Bayesian nonparametric modelling of structured point processes," *arXiv preprint arXiv:1407.6949*, 2014.

[12] C. Li and S. Godsill, "Sequential inference methods for non-homogeneous Poisson processes with state-space prior," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2856–2860.

[13] S. K. Pang, J. Li, and S. Godsill, "Detection and tracking of coordinated groups," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 1, pp. 472–502, 2011.

[14] A. Golightly and D. Wilkinson, "Bayesian sequential inference for nonlinear multivariate diffusions," *Statistics and Computing*, vol. 16, no. 4, pp. 323–338, 2006.

[15] P. Lewis and G. Shedler, "Simulation of nonhomogeneous Poisson processes by thinning," *Naval Research Logistics (NRL)*, vol. 26, no. 3, pp. 403–413, 1979.

[16] S. Chiu, D. Stoyan, W. Kendall, and J. Mecke, *Stochastic geometry and its applications*, John Wiley & Sons, 2013.

[17] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*, vol. 1, MIT press Cambridge, 2006.

[18] S. Särkkä et al., *Recursive Bayesian inference on stochastic differential equations*, Helsinki University of Technology, 2006.

[19] S. Godsill, "Particle filters for continuous-time jump models in tracking applications," in *ESAIM: Proceedings*. EDP Sciences, 2007, vol. 19, pp. 39–52.

[20] C. Berzuini and W. Gilks, “RESAMPLE-MOVE filtering with cross-model jumps,” in *Sequential Monte Carlo Methods in Practice*, pp. 117–138. Springer, 2001.

[21] S. K. Pang, S. Godsill, J. Li, F. Septier, and S. Hill, *Sequential inference for dynamically evolving groups of objects*, chapter 12, pp. 245–276, Bayesian Time Series Models. 2011.

[22] F. Septier and G. Peters, “Langevin and Hamiltonian based sequential MCMC for efficient Bayesian filtering in high-dimensional spaces,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 312–327, 2016.

[23] A. Finke, A. Doucet, and A. Johansen, “Limit theorems for sequential MCMC methods,” *arXiv preprint arXiv:1807.01057*, 2018.

[24] O. Cappé, S. Godsill, and E. Moulines, “An overview of existing methods and recent advances in sequential Monte Carlo,” *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.

[25] P. Green, “Reversible jump Markov chain Monte Carlo computation and Bayesian model determination,” *Biometrika*, vol. 82, no. 4, pp. 711–732, 1995.

[26] D. Scott, *Multivariate density estimation: theory, practice, and visualization*, John Wiley & Sons, 2015.

[27] M. Gould, M. Porter, S. Williams, M. McDonald, D. Fenn, and S. Howison, “Limit order books,” *Quantitative Finance*, vol. 13, no. 11, pp. 1709–1742, 2013.

[28] L. Menkhoff, L. Sarno, M. Schmeling, and A. Schrimpf, “Currency momentum strategies,” *Journal of Financial Economics*, vol. 106, no. 3, pp. 660–684, 2012.

[29] A. Sokal, “Monte Carlo methods in statistical mechanics: foundations and new algorithms,” in *Functional integration*, pp. 131–192. Springer, 1997.

[30] Z. Ghahramani and G. Hinton, “Variational learning for switching state-space models,” *Neural computation*, vol. 12, no. 4, pp. 831–864, 2000.

[31] C. Andrieu, A. Doucet, and R. Holenstein, “Particle Markov chain Monte Carlo methods,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 3, pp. 269–342, 2010.

[32] W. Gilks and P. Wild, “Adaptive rejection sampling for Gibbs sampling,” *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 41, no. 2, pp. 337–348, 1992.

[33] D. Görür and Y. W. Teh, “Concave-convex adaptive rejection sampling,” *Journal of Computational and Graphical Statistics*, vol. 20, no. 3, pp. 670–691, 2011.