Journal of Machine Learning Research 25 (2024) 1-67 Submitted 2/24; Revised 11/24; Published 12/24

Virtual-Event-Based Posterior Sampling and Inference for Neyman-Scott Processes

Chengkuan Hong CKHONG@MAIL.TSINGHUA.EDU.CN
Department of Computer Science and Technology, BNRist Center
Tsinghua Institute for AI, Tsinghua-Bosch Joint Center for ML
Tsinghua University
Beijing 100084, China

Christian R. Shelton CSHELTON@CS.UCR.EDU
Department of Computer Science and Engineering
University of California
Riverside, CA 92521, USA

Jun Zhu [*] DCSZJ@TSINGHUA.EDU.CN
Department of Computer Science and Technology, BNRist Center
Tsinghua Institute for AI, Tsinghua-Bosch Joint Center for ML
Tsinghua University
Beijing 100084, China

Editor: Stephan Mandt

Abstract

Neyman-Scott processes (NSPs) are a class of Cox processes constructed by stacking layers of Poisson processes into a deep structure. While a lot of research has been conducted regarding the posterior sampling and inference for NSPs, most of the existing methods only work for shallow NSPs (*i.e.*, NSPs with one layer of latent Poisson processes). In this paper, we present virtual-event-based posterior sampling and inference algorithms for NSPs. The algorithms work for both deep NSPs and shallow NSPs. Moreover, we show that deep NSPs can be viewed as branching processes or a limiting case of probabilistic graphical models. We conduct a theoretical analysis of the convergence of our algorithms and provide the condition for the convergence to hold. In doing so, we also prove the convergence of virtual-event-based sampling inference algorithms for other point process models with missing information (Markov jump processes, piecewise-constant intensity models, and Hawkes processes). Like NSPs, the latent variables of these models with missing information are also point processes. Our experimental results demonstrate that the prediction based on our sampling and inference algorithms for NSPs can achieve good prediction performance compared with state-of-the-art methods.

Keywords: Markov chain Monte Carlo, variational inference, point processes, hierarchical model, Neyman-Scott processes

1. Introduction

There has been a long history of developing hierarchical models (*e.g.*, deep neural networks (LeCun et al., 2015), graph neural networks (Scarselli et al., 2008), and probabilistic graphi-

[*]. The corresponding author

cal models (Koller and Friedman, 2009)) to solve real-world problems, and they have achieved great success in various fields, such as computer vision (He et al., 2016), natural language processing (Gales et al., 2008), and biology (Fout et al., 2017). However, little attention has been given to inventing hierarchical models built solely with point processes, though point processes without such kind of hierarchical structures have been used in different fields like finance (Bauwens and Hautsch, 2009), cosmology (Stoica et al., 2014), and crime modeling (Shelton et al., 2018). Amidst all point process models, Neyman-Scott processes (NSPs) (Neyman and Scott, 1958), a class of networks with each node representing a point process, have shown a promising future of working as a backbone model to capture the hierarchy.

NSPs were first proposed as a class of stochastic point processes to model the clustering behavior of galaxies. Since the appearance of NSPs, they have been widely used in different areas, such as ecology (Waagepetersen and Guan, 2009), neuroscience (Williams et al., 2020), and pandemics (Park et al., 2022). The central assumption of NSPs is that the observed events are triggered by hidden events, and the hidden events themselves are modeled as random events triggered by other hidden events. Each event is a point. Then, this hierarchical triggering behavior allows us to stack multiple point processes to form a deep structure and constitute the primary generative mechanism of NSPs. The points that can induce another set of points are called parent points, while the points induced by the parent points are called child points of these parent points. The child points themselves can serve as parent points to generate another generation of child points. If the parent points of an NSP are not child points of any other points, then this NSP is called a shallow NSP (SNSP), otherwise it is called a deep NSP (DNSP). In the original paper proposing NSPs, Neyman and Scott (1958) only describe the hierarchical clustering behavior of the model without giving specific formulas. We first use DNSPs to name these kinds of deep structures and give specific formulas for the intensity functions of each Poisson process in NSPs. We present that DNSPs can be constructed as branching processes by treating each layer of DNSPs as a generation of branching processes. We also show that DNSPs are a limiting case of probabilistic graphical models when the number of random variables of probabilistic graphical models approaches infinity.

Various posterior sampling and inference techniques (*e.g.*, Møller and Waagepetersen, 2003; Waagepetersen and Guan, 2009; Andersen et al., 2018; Williams et al., 2020; Wang et al., 2023) have been developed for NSPs though most of them are only applied to SNSPs. To the best of our knowledge, Andersen et al. (2018) present the only work that infers the parameters for a DNSP, but their method is not able to estimate the posterior distribution of the points in the hidden space (*i.e.*, parent points). The difficulty of designing efficient posterior sampling and inference algorithms for NSPs lies in the fact that the hidden space of NSPs contains an unbounded number of points and it is hard to find an analytical expression for the posterior distribution of these hidden points. Moreover, as we stack more point processes into a deep NSP, the dependencies between different point processes become increasingly complex and the search space of the posterior distribution of the hidden points becomes increasingly large. Thus, we need to design a novel algorithm to help trim the search space such that we can spend more time searching at the places where more points would occur in the posterior distribution.

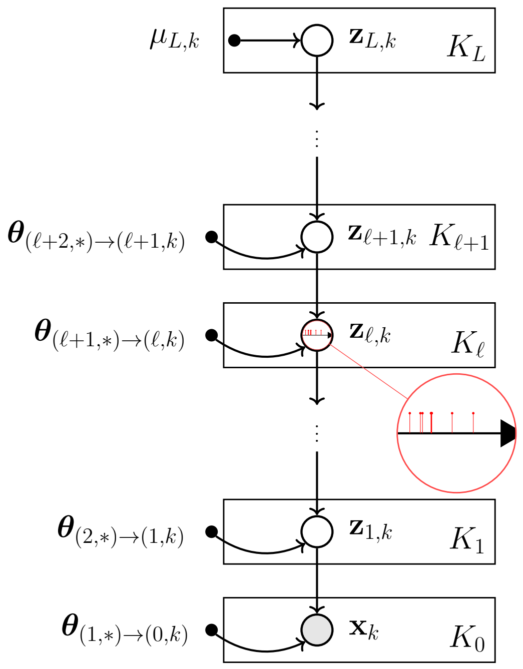
We propose virtual-event-based posterior sampling and inference algorithms for NSPs in this paper. The proposed algorithms work for both shallow and deep NSPs. Virtual-event-based posterior sampling algorithms are based on reversible jump Markov chain Monte Carlo (RJMCMC) (Green, 1995) and have been applied to various kinds of models, such as Markov jump processes (MJPs) (Rao and Teh, 2011, 2013), piecewise-constant conditional intensity models (PCIMs) (Gunawardana et al., 2011; Qin and Shelton, 2015), and Hawkes processes (Hawkes, 1971; Shelton et al., 2018). Virtual events are auxiliary variables that can help suggest possible locations of the events in the posterior distribution. The virtual-event-based posterior sampling for NSPs differs significantly from the existing virtual-event-based sampling algorithms in that the virtual events in our sampler can form a hierarchical structure themselves. Moreover, we show that the hierarchical structures formed by the virtual events in our algorithm can work as an approximation for the true posterior distribution of the hidden events. The distribution of the virtual events is learned through the optimization of the inclusive Kullback–Leibler (KL) divergence, which is the key element of our virtual-event-based inference algorithm. Different from the variational inference algorithm in Naesseth et al. (2020), which minimizes inclusive KL divergence *w.r.t.* the variational parameters of the Markov kernels, our inference algorithm optimizes the inclusive KL divergence for the variational parameters of the auxiliary variables. Besides, our MCMC has an unbounded number of dimensions, while the MCMC algorithm in Naesseth et al. (2020) has a fixed number of dimensions. Although the virtual-event-based inference algorithm is only applied to NSPs currently, we expect it can be generalized to other models (*e.g.*, MJPs, PCIMs, and Hawkes processes) as well if an approximate variational expression for the virtual events can be found.

Though virtual-event-based posterior sampling algorithms have been applied to multiple models, including ours, the analysis of the convergence is typically insufficient. The previous work (Rao and Teh, 2011, 2013; Qin and Shelton, 2015; Shelton et al., 2018; Hong and Shelton, 2022, 2023) did not exclude a null set from which convergence may fail. By studying Ψ-irreducibility, aperiodicity, and Harris recurrence, we provide a sufficient condition for the convergence of the RJMCMC to hold. With the convergence of the posterior sampling and appropriate assumptions, the convergence of the optimization for inclusive KL divergence can be proved to hold as well.

The paper is organized as follows. We introduce DNSPs in section 2. Then we describe the posterior sampling in section 3 and the variational inference in section 4. The analysis of the convergence of MCMC is performed in section 5 and the analysis of the convergence of variational inference is discussed in section 6. With our novel algorithm, we are able to make predictions for sequences and the prediction procedure is described in section 7. The experimental results for earthquakes, homicides, and retweets are presented in section 8. Finally, we conclude in section 9. Part of this paper builds on conference proceedings in Hong and Shelton (2022, 2023) and we greatly extend the analysis of virtual-event-based posterior sampling and inference algorithms to provide rigorous convergence guarantees for both NSPs and other similar samplers (*i.e.*, the samplers for MJPs, PCIMs, and Hawkes processes).

2. Deep Neyman-Scott Processes

The formal introduction of the hierarchical structure and the generative mechanism for DNSPs is given in this section. We should notice that SNSPs are DNSPs with the number of layers equaling 1. The main focus of this paper is on temporal point processes, but most of the results can be generalized to spatio-temporal point processes easily.



$\mu_{L,k}$
$\mathbf{z}_{L,k}$
K_L
$\boldsymbol{\theta}_{(\ell+2,*)\to(\ell+1,k)}$
$\mathbf{z}_{\ell+1,k}$
$K_{\ell+1}$
$\boldsymbol{\theta}_{(\ell+1,*)\to(\ell,k)}$
$\mathbf{z}_{\ell,k}$
K_ℓ
$\boldsymbol{\theta}_{(2,*)\to(1,k)}$
$\mathbf{z}_{1,k}$
K_1
$\boldsymbol{\theta}_{(1,*)\to(0,k)}$
$\mathbf{x}_k$
K_0

Figure 1: Structure of a DNSP

As depicted in Figure 1, the generative procedure for DNSPs can be described in a top-down manner. Poisson processes are stacked together to form a network, with each node of this network as a sample from a Poisson process. We usually use intensity functions to describe Poisson processes:

Definition 1 (intensity function) *The intensity function $\lambda(t)$ for a Poisson process in an Euclidean space $\mathbb{R}^n$ is defined as*

$$\lambda(t) = \lim_{\delta \to 0} \frac{\mathbb{E}[\textit{the number of points inside } t_\delta]}{\delta},$$

where t_δ is a ball centered at t with radius δ.

Different from the presentation of NSPs in Neyman and Scott (1958), we depict DNSPs with specific intensity functions of each node similar to deep neural networks or probabilistic graphical models. The Poisson processes at the top are homogeneous Poisson processes and the intensities are constant numbers, while the intensities of the other Poisson processes are fully determined by the samples from the Poisson processes stacked above. There are $L+1$ layers of Poisson processes (L hidden layers and 1 observed layer) in Figure 1, with the Poisson processes at layer ℓ denoted as $\mathbf{Z}_\ell = \{Z_{\ell,k}\}_{k=1}^{K_\ell}$ (the realizations are denoted as $\mathbf{z}_\ell = \{\mathbf{z}_{\ell,k}\}_{k=1}^{K_\ell}$), where K_ℓ is the number of Poisson processes at layer ℓ and $Z_{\ell,k}$ is the k-th hidden Poisson process (k-th type for multivariate point processes or k-th mark in marked point processes) at layer ℓ. When $\ell = 0$, $\mathbf{z}_{0,k} = \mathbf{x}_k$ is observed. The intensity $\lambda_{\ell,k}$ of $Z_{\ell,k}$ is fully determined by the samples of $\mathbf{Z}_{\ell+1}$ through kernel functions

$$\phi_{\boldsymbol{\theta}_{(\ell+1,\cdot)\to(\ell,k)}} = \{\phi_{\boldsymbol{\theta}_{(\ell+1,i)\to(\ell,k)}}\}_{i=1}^{K_{\ell+1}}$$

for $\ell < L$, while the intensity for $Z_{L,k}$ is a constant number $\mu_k > 0$. A SNSP is a DNSP with $L = 1$.

The Poisson processes at the bottom generate samples that can be observed, while the samples generated by inner-layer Poisson processes cannot be directly detected. Thus, the processes that are not at the bottom are hidden processes.

Generative Mechanism The generative mechanism of DNSPs is an iterative procedure. We first draw a sample from each of the homogeneous Poisson processes at the top layer, and the intensity functions for these Poisson processes are denoted as

$$\lambda_{L,k} = \mu_k > 0.$$

Then we sample from the nonhomogeneous Poisson processes conditioned on the Poisson processes stacked on the layer above. Given the samples from $\mathbf{Z}_{\ell+1}$, the intensity function for $Z_{\ell,k}$ is

$$\lambda_{\ell,k}(t) = \sum_{i=1}^{K_{\ell+1}} \sum_{j=1}^{m_{\ell+1,i}} \phi_{\boldsymbol{\theta}_{(\ell+1,i)\to(\ell,k)}}(t - t_{\ell+1,i,j}), \tag{1}$$

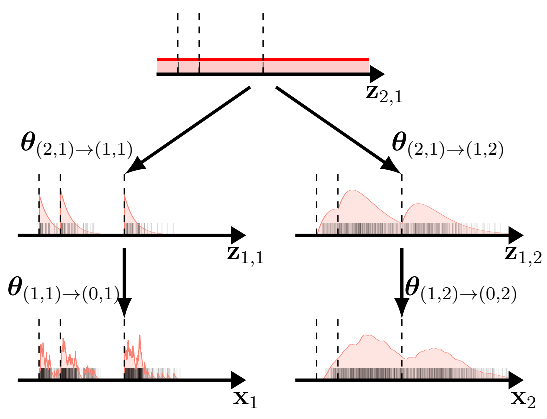
where $m_{\ell+1,i}$ is the number of points for the sample drawn from $Z_{\ell+1,i}$, $t_{\ell+1,i,j}$ represents the location of the j-th point of the sample drawn from $Z_{\ell+1,i}$, and $\phi_{\boldsymbol{\theta}_{(\ell+1,i)\to(\ell,k)}}(\cdot)$ is a bounded function in a bounded space.

We let the time interval (denoted as $[0, T]$) for each point process be the same and ignore the edge effect to simplify our model.

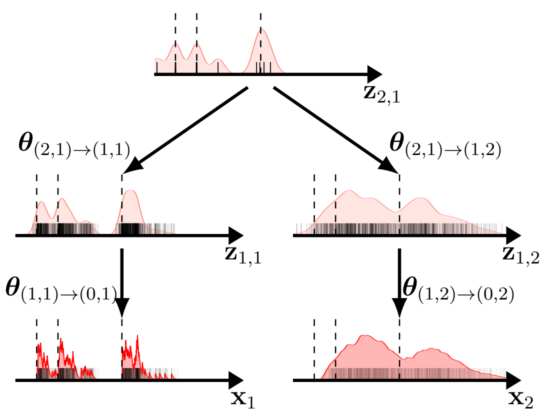
Figure 2 is an example of the distribution of the events for a temporal DNSP with two hidden layers. Every Poisson process resides in the same time interval. The sampled events in Figure 2(a) are from forward sampling, and the sampled events in Figure 2(b) are from posterior sampling.

During the forward sampling process in Figure 2(a), the realization $\mathbf{z}_{2,1}$ has 3 events sampled from the homogeneous Poisson process at the top. The dashed lines at the top represent the positions of these 3 events. Then, conditioned on the events at the top, we can draw the realizations $\mathbf{z}_{1,1}$ and $\mathbf{z}_{1,2}$. Similarly, $\mathbf{x}_1$ is drawn conditional on $\mathbf{z}_{1,1}$ and $\mathbf{x}_2$ is drawn conditional on $\mathbf{z}_{1,2}$. The intensity for $Z_{2,1}$ is given and is a constant number, drawn as a red line at the top. The other pink plots show the intensity functions generated using the samples from the layer above. The short black lines sticking to the horizontal axes at the bottom indicate the positions of the events from the realizations.

For posterior sampling (one of the main focuses of this paper), demonstrated in Figure 2(b), $\mathbf{x}_1$ and $\mathbf{x}_2$ are observations and they are fixed. The samples for the observations are collected from forward sampling. The samples for the hidden point processes ($\mathbf{z}_{1,1}$, $\mathbf{z}_{1,2}$, and $\mathbf{z}_{2,1}$) are drawn by performing the posterior sampling conditional on $\mathbf{x}_1$ and $\mathbf{x}_2$. The dashed lines indicate the positions of the 3 events generated by the homogeneous Poisson process in Figure 2(a). The pink plots are generated by kernel density estimation using Gaussian kernels.


$\mathbf{z}_{2,1}$
$\boldsymbol{\theta}_{(2,1)\to(1,1)}$
$\boldsymbol{\theta}_{(2,1)\to(1,2)}$
$\mathbf{z}_{1,1}$
$\mathbf{z}_{1,2}$
$\boldsymbol{\theta}_{(1,1)\to(0,1)}$
$\boldsymbol{\theta}_{(1,2)\to(0,2)}$
$\mathbf{x}_1$
$\mathbf{x}_2$

(a) Forward Sampling


$\mathbf{z}_{2,1}$
$\boldsymbol{\theta}_{(2,1)\to(1,1)}$
$\boldsymbol{\theta}_{(2,1)\to(1,2)}$
$\mathbf{z}_{1,1}$
$\mathbf{z}_{1,2}$
$\boldsymbol{\theta}_{(1,1)\to(0,1)}$
$\boldsymbol{\theta}_{(1,2)\to(0,2)}$
$\mathbf{x}_1$
$\mathbf{x}_2$

(b) Posterior Sampling

Figure 2: Distribution of events: (a) demonstration of the samples generated by forward sampling; (b) demonstration of the samples generated by posterior sampling.

The Kernel Function Kernel functions are used to propagate the information from the top layer to the lower layers. The desirable kernel functions should be as flexible as possible with a small number of parameters. Moreover, it is advantageous to have an analytical inverse of the integral of the kernel functions, to allow efficient sampling of the Poisson processes using inversion sampling. Here we give two examples of temporal point processes.

Example 1 (Gamma kernel for temporal point processes)

$$\phi_{\boldsymbol{\theta}}(x) = \begin{cases} p \cdot \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} & \textit{for } x > 0,\ p, \alpha, \beta > 0, \\ 0 & \textit{for } x \leq 0, \end{cases}$$

where $\boldsymbol{\theta} = \{p, \alpha, \beta\}$ *and* $\Gamma(\alpha)$ *is the Gamma function.*

Example 2 (Weibull kernel for temporal point processes)

$$\phi_{\boldsymbol{\theta}}(x) = \begin{cases} p \cdot \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & \textit{for } x > 0,\ k, \lambda > 0, \\ 0 & \textit{for } x \leq 0, \end{cases}$$

where $\boldsymbol{\theta} = \{p, k, \lambda\}$.

Both the Gamma and Weibull kernels decrease monotonically or behave like a Gaussian function. They approach 0 when $x \to \infty$. The Weibull kernel has analytical forms of the gradients for the kernel function itself and the integral of the kernel function, while the gradient of the integral of the Gamma kernel is numerically unstable and expensive to calculate because it involves the Meijer G-function.

Remark 2 *Although we only discuss the kernel functions for temporal point processes, we can extend the kernel in this paper to the kernel for spatio-temporal point processes, simply by defining the functional forms of the kernel over the whole space, and not forcing the values to be 0 for* $x \leq 0$.

Connections to Branching Processes Similar to the branching structure of Hawkes processes in Møller and Rasmussen (2006), DNSPs can also be defined as branching processes. Layer ℓ can be viewed as generation $L-\ell$ and each event with time $t_{\ell,i,j}$ can be viewed as an individual of generation $L-\ell$. Each event of generation $L-\ell$ can generate offspring for generation $L-\ell+1$. The offspring of the event with time $t_{\ell,i,j}$ are distributed as Poisson processes with the intensity function

$$\phi_{\ell,i,j}(t) = \sum_{r=1}^{K_{\ell-1}} \phi_{\boldsymbol{\theta}_{(\ell,i)\to(\ell-1,r)}}(t - t_{\ell,i,j}).$$

Connections to Probabilistic Graphical Models DNSPs can be viewed as a limiting case of probabilistic graphical models (more specifically, deep exponential families (DEFs) (Ranganath et al., 2015)) when the number of random variables of DEFs approaches infinity.

Following the conventions of DEFs, we consider a DEF with L hidden layers. The data is at layer 0 (the bottom layer) and the hidden layer directly connected to layer 0 is layer 1 (the bottom hidden layer). The top hidden layer is layer L. The random variables at layer ℓ are denoted as $\mathbf{G}_\ell = \{\{G_{\ell,i,j}\}_{i=1}^{K_\ell}\}_{j=1}^{\hat{m}_{\ell,i}}$ (corresponding to the hidden variables $\mathbf{z}_\ell$ in Ranganath et al. (2015)) and they are distributed as Bernoulli distributions, where $\hat{m}_{\ell,i}$ is the number of random variables corresponding to the i-th Poisson process at layer ℓ of DEFs. For each i, we can divide the interval $[0,T]$ into equally sized subintervals

$$\{[T_{\ell,i,j}, T_{\ell,i,j+1})\}_{j=0}^{\hat{m}_{\ell,i}-2} \cup [T_{\ell,i,\hat{m}_{\ell,i}-1}, T_{\ell,i,\hat{m}_{\ell,i}}] = [0,T],$$

where $T_{\ell,i,0} = 0$ and $T_{\ell,i,\hat{m}_{\ell,i}} = T$.

$G_{\ell,i,j}$ corresponds to the number of events at subinterval $\hat{T}_{\ell,i,j} = [T_{\ell,i,j-1}, T_{\ell,i,j})$ for $j \le \hat{m}_{\ell,i} - 1$ and $\hat{T}_{\ell,i,j} = [T_{\ell,i,\hat{m}_{\ell,i}-1}, T_{\ell,i,\hat{m}_{\ell,i}}]$ for $j = \hat{m}_{\ell,i}$ in DNSPs. The size of each subinterval at layer ℓ is denoted as δ_ℓ. The kernel $\phi_{\boldsymbol{\theta}}(t)$ satisfies $0 \le \phi_{\boldsymbol{\theta}}(t) \le \hat{\phi}$ for a positive number $\hat{\phi} > 1$ and $t \in [-T,T]$. The probability of $G_{\ell,i,j} = 1$ is $p(G_{\ell,i,j} = 1) = \lambda_{\ell,i}(\xi_{\ell,i,j}) \cdot \delta_\ell$ and $p(G_{\ell,i,j} = 0) = 1 - p(G_{\ell,i,j} = 1)$, where $\xi_{\ell,i,j}$ is from a uniform distribution with the support as $\hat{T}_{\ell,i,j}$, $\lambda_{\ell,k}(t) = \sum_{i=1}^{K_{\ell+1}} \sum_{j=1}^{\hat{m}_{\ell+1,i}} G_{\ell+1,i,j} \cdot \phi_{\boldsymbol{\theta}_{(\ell+1,i)\to(\ell,k)}}(t - \xi_{\ell+1,i,j})$ for $\ell \le L-1$, and $\lambda_{L,k}(t) = \mu_k$. The size of subinterval δ_ℓ satisfies $\delta_\ell < 1/(K_{\ell+1} \cdot (T/\delta_{\ell+1}) \cdot \hat{\phi})$ for $\ell \le L-1$ and $\delta_L < 1/\max_k\{\mu_k\}$, such that $p(G_{\ell,i,j} = 1) = \lambda_{\ell,i}(\xi_{\ell,i,j}) \cdot \delta_\ell < 1$ for $\ell = 0,1,2,\cdots,L$ and $\delta_\ell < \delta_{\ell+1}$.

Following the same notations of the hidden variables $\mathbf{z}$ and the weights $\mathbf{W}$ in DEFs (Ranganath et al., 2015), we let $\mathbf{z}_{\ell+1} = \left[[G_{\ell+1,i,j}]_{i=1}^{K_\ell}\right]_{j=1}^{\hat{m}_{\ell,i}}$ and $\mathbf{w}_{\ell,k,r} = \left[[\phi_\theta(\xi_{\ell,k,r} - \xi_{\ell+1,i,j})]_{i=1}^{K_\ell}\right]_{j=1}^{\hat{m}_{\ell,i}}$. Notice that we only use $\mathbf{z}$ for hidden variables in DEFs here. For the rest of this paper, $\mathbf{z}$ refers to the hidden points processes in DNSPs. Like convolutional DEFs (Hong and Shelton, 2021), the construction of the weights here also enjoys a convolutional paradigm with the filter size equal to infinity since the weights $\mathbf{w}_{\ell,k,r}$ only depend on the distances $\{\{\xi_{\ell,k,r} - \xi_{\ell+1,i,j}\}_{i=1}^{K_\ell}\}_{j=1}^{\hat{m}_{\ell,i}}$ between different hidden variables in adjacent layers. For the top layer, the distribution of $z_{L,k,r} \equiv G_{L,k,r}$ is

$$\begin{aligned} p(z_{L,k,r}) =& \text{Bernoulli}(z_{L,k,r}; \mu_k \cdot \delta_L) \\ =& z_{L,k,r} \cdot \mu_k \cdot \delta_L + (1 - z_{L,k,r}) \cdot (1 - \mu_k \cdot \delta_L). \end{aligned}$$

The distribution of the random variable $z_{\ell,k,r} \equiv G_{\ell,k,r}$ for $\ell \leq L-1$ is

$$\begin{aligned} p(z_{\ell,k,r} \mid \mathbf{z}_{\ell+1}, \mathbf{w}_{\ell,k,r}) &= \text{Bernoulli}(z_{\ell,k,r}; g(\mathbf{z}_{\ell+1}^T \mathbf{w}_{\ell,k,r})) \\ &= z_{\ell,k,r} \cdot g_\ell(\mathbf{z}_{\ell+1}^T \mathbf{w}_{\ell,k,r}) + (1 - z_{\ell,k,r}) \cdot (1 - g(\mathbf{z}_{\ell+1}^T \mathbf{w}_{\ell,k,r})), \end{aligned}$$

where the link function $g_\ell(\cdot)$ is given as $g_\ell(x) = x \cdot \delta_\ell$.

With the above construction, it is easy to find that the joint density, the conditional density, and the marginal density of DEFs converge to the joint density, conditional density, and the marginal density of DNSPs when $\delta_L \to 0$. The definition and more rigorous discussions of the density of DNSPs can be found in Section 3.2 and Theorem 7.

3. Posterior Sampling

By contrast to straightforward forward sampling, posterior sampling is generally considered computationally intense, as it involves an unbounded number of events and there is no analytical expression for the posterior distribution of the hidden Poisson processes. A lot of research (*e.g.*, Møller and Waagepetersen (2003); Kopecký and Mrkvička (2016); Williams et al. (2020); Wang et al. (2023)) has been conducted on the posterior sampling for those NSPs with one single hidden layer. However, little work has been done to perform Bayesian posterior sampling for NSPs with more than one hidden layer. In this paper, we provide a posterior sampling algorithm, which can be applied to NSPs with any finite number of hidden layers. Moreover, this posterior sampling algorithm serves as the foundation for both the likelihood-based inference in section 4.1 and the variational inference in section 4.2.

The posterior sampling is based on RJMCMC equipped with auxiliary variables (or auxiliary point processes), which we call virtual point processes (VPPs). In contrast to the VPPs, we call the point processes belonging to our model (*i.e.*, the point processes that are not auxiliary variables) the real point processes (RPPs). The events belonging to VPPs are called virtual events, and the events belonging to RPPs are called real events.

The intentions of the introduction of virtual events are (1) to use the virtual events, instead of the real events, to explore the space of the real events more easily, as the sampling of the virtual events is very efficient by inversion; (2) to serve as variational approximations of the posterior point processes; (3) to help put more computational resources into searching where there are more events; and (4) to propagate the information from the observation to every hidden point process. Although virtual-event-based posterior sampling has been applied to MJPs, PCIMs, and Hawkes processes, the virtual-event-based posterior sampling for NSPs requires a new design of the sampler moves and the distribution of the virtual events, as our virtual events form a hierarchical structure which propagates the information from the data to the latent space.

3.1 Virtual Events

The VPPs are generative models as well. In correspondence with the DNSPs depicted in Figure 1, there are L layers of VPPs $\tilde{\mathbf{Z}}_{1:L} = \{\tilde{\mathbf{Z}}_1, \cdots, \tilde{\mathbf{Z}}_L\}$, where $\tilde{\mathbf{Z}}_\ell = \{\tilde{Z}_{\ell,k}\}_{k=1}^{K_\ell}$.

3.1.1 Generative Mechanism

The VPPs are generated from bottom to top. We generate $\tilde{\mathbf{Z}}_1$ first and $\tilde{\mathbf{Z}}_1$ is assumed to be a class of Poisson processes. The functional form of the intensity function for $\tilde{Z}_{1,k}$ is

$$\tilde{\lambda}_{1,k}(t) = q_{1,k}(t; \tilde{\mathbf{z}}_0 = \mathbf{x}, \tilde{\boldsymbol{\theta}}_{1,k}(t)),$$

where $q_{1,k}(t; \mathbf{x}, \tilde{\boldsymbol{\theta}}_{1,k}(t))$ is a function of t with some parameters characterized by the events from the observation $\mathbf{x}$, and the other parameters $\tilde{\boldsymbol{\theta}}_{1,k}(t)$ that can evolve with time t.

Next, the samples $\tilde{\mathbf{z}}_\ell$ for $\tilde{\mathbf{Z}}_\ell$ are drawn conditionally on the samples $\tilde{\mathbf{z}}_{\ell-1}$, and the intensity function for $\tilde{Z}_{\ell,k}$ is

$$\tilde{\lambda}_{\ell,k}(t) = q_{\ell,k}(t; \tilde{\mathbf{z}}_{\ell-1}, \tilde{\boldsymbol{\theta}}_{\ell,k}(t)),$$

where $q_{\ell,k}(t; \tilde{\mathbf{z}}_{\ell-1}, \tilde{\boldsymbol{\theta}}_{\ell,k}(t))$ is a function of t with some parameters characterized by the events from the sample $\tilde{\mathbf{z}}_{\ell-1}$, and the other parameters $\tilde{\boldsymbol{\theta}}_{\ell,k}(t)$ that can evolve with time t.

3.1.2 Examples for Virtual Point Processes

The generative mechanism guarantees that the information can be transported from the observation to the top. Additionally, we need to design functional forms for $q_{\ell,k}$ such that the inversion method for sampling can be efficiently applied and the VPPs are close enough to the posterior point processes. For this purpose, we give two examples of the functional forms for $q_{\ell,k}$: upward Neyman-Scott processes (UNSPs) and upward self-attention processes (USAPs).

Example 3 (Upward Neyman-Scott Processes) *The VPPs are NSPs evolving in an upward direction. In this case, the intensity function is*

$$\tilde{\lambda}_{\ell,k}(t) = q_{\ell,k}\left(t; \tilde{\mathbf{z}}_{\ell-1}, \tilde{\boldsymbol{\theta}}_{\ell,k}\right) = \tilde{\mu}_{\ell,k} + \sum_{i=1}^{K_{\ell-1}} \sum_{t_{\ell-1,i,j}} \phi_{\tilde{\boldsymbol{\theta}}_{(\ell-1,i)\to(\ell,k)}}(\tilde{t}_{\ell-1,i,j} - t), \tag{2}$$

where $\tilde{\boldsymbol{\Theta}}_{\ell,k} = \left\{\tilde{\mu}_{\ell,k}, \left\{\tilde{\boldsymbol{\theta}}_{(\ell-1,i)\to(\ell,k)}\right\}_{i=1}^{K_{\ell-1}}\right\}$ *and* $\tilde{\mu}_{\ell,k} > 0$.

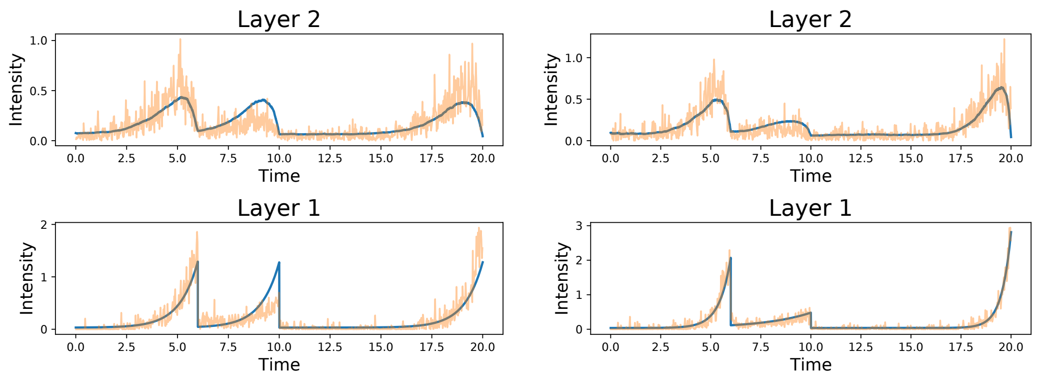
Example 4 (Upward Self-Attention Processes) *To leverage the power of neural networks to construct more flexible approximate point processes, we adopt self-attention to encode the event information like other work (e.g., Zuo et al., 2020; Zhang et al., 2020; Chen et al., 2021). Each event* $t_{\ell-1,i,j}$ *is mapped to a hidden vector* $\boldsymbol{h}_{\ell-1,i,j} = f_{Attn}(t_{\ell-1,i,j}, \mathbf{z}^I_{\ell-1} = \{\{t_{\ell-1,i,j}\}_{j=1}^{m_{\ell-1,i}}\}_{i=1}^{K_{\ell-1}}\})$ *through self-attention. The intensity function for* $t_{\ell-1,i,j-1} < t \leq t_{\ell-1,i,j}$ *becomes*

$$\tilde{\lambda}_{\ell,k}(t) = q_{\ell,k}\left(t; \tilde{\mathbf{z}}_{\ell-1}, \tilde{\boldsymbol{\theta}}_{\ell,k}\right) = \tilde{\mu}_{\ell,k} + \phi_{\tilde{\boldsymbol{\theta}}_{\ell-1,i,j}}(\tilde{t}_{\ell-1,i,j} - t), \tag{3}$$

where $\tilde{\boldsymbol{\Theta}}_{\ell,k} = \left\{\tilde{\mu}_{\ell,k}, \left\{\left\{\tilde{\boldsymbol{\theta}}_{\ell-1,i,j}\right\}_{i=1}^{K_{\ell-1}}\right\}_{j=1}^{m_{\ell-1,i}}\right\}$, $\tilde{\mu}_{\ell,k} > 0$, *and* $\tilde{\boldsymbol{\theta}}_{\ell,k}$ *is the output of a neural network applied to the hidden vector* $\boldsymbol{h}_{\ell-1,i,j}$.

Comparison To establish an intuitive understanding of how close our examples of variational point processes can approximate the posterior point processes, we demonstrate two clean and typical examples for UNSPs and USAPs respectively in Figure 3. The intensity functions obtained from variational inference for UNSPs and USAPs are compared with the approximate intensity functions obtained from MCMC. The details of how to do variational inference and MCMC will be discussed in the following sections.

For simplicity, we consider a DNSP with 2 hidden layers, each with one Poisson process. 3 events (times at 6, 10, 20) are picked manually and fixed at the observation. The model parameters (*i.e.*, $\boldsymbol{\theta}$ and μ) are fixed, and we only change the parameters for the VPPs (*i.e.*, $\tilde{\boldsymbol{\theta}}$ and $\tilde{\mu}$). The yellow lines represent the approximate intensity functions and we estimate these lines with the samples from MCMC. The whole interval is split into many subintervals, and the intensity for a subinterval is the number of events at that subinterval divided by its volume. The blue lines are the approximate intensity functions estimated by using the samples from VPPs. The approximate intensity functions for layer 1 are just $q_{1,0}(\cdot)$ and we can simply draw this function. For layer 2, we do not have a direct way to represent the approximate intensity functions. Thus, we first generate samples for the VPPs at layer 1. Then, conditioned on the samples for layer 1, we can get some intensity functions for layer 2. The approximate intensity functions for layer 2 are just the mean of all these intensity functions depending on the samples drawn from the VPPs at layer 1. It is easy to see from Figure 3 that the USAP fits more closely to the approximate intensity functions for MCMC than UNSP.



(a) UNSP vs. MCMC

(b) USAP vs. MCMC

Figure 3: Comparison between UNSP and USAP

USAPs approximate better because (1) self-attention enables the adaptation of the parameters for the kernels independently for each interval, while the parameters for the kernels of UNSPs are tied for the same pair of connected Poisson processes; (2) USAPs can take account of the events happening both before time t and after time t when producing the intensity function at time t, while UNSPs can only use the information of the events with the times greater than t.

3.2 Density

We focus on realizations of point processes that have finite cardinality in this paper, as it is not realistic to have an infinite number of events in the observation. Each realization $\mathbf{x}$ is a countable subset of a space S and S is chosen to be a subset of a Euclidean space with d dimensions $\mathbb{R}^d$. These realizations are called *locally finite point configurations* (Møller and Waagepetersen, 2003), which is defined in the following as

Definition 3 (Locally Finite Point Configurations) *Suppose $S \subseteq \mathbb{R}^d$, then N_{lf} is defined as*

$$N_{lf} = \{\mathbf{x} \subseteq S : n(\mathbf{x}_B) < \infty \text{ for all bounded } B \subseteq S\},$$

where $\mathbf{x}_B = \mathbf{x} \cap B$ and $n(\mathbf{x})$ denotes the cardinality of $\mathbf{x}$. Elements of N_{lf} are called locally finite point configurations.

Unlike random variables, which usually have probability density functions defined *w.r.t.* the Lebesgue measure, we can only define the density for a finite point process *w.r.t.* a Poisson process. (Following the standard approach (Møller and Waagepetersen, 2003), we choose the reference Poisson process as the homogeneous Poisson process with the intensity function as 1 in this paper. The choice of the reference does not affect the correctness of our derivation.) The density can be defined as follows by using the Radon-Nikodym theorem

Definition 4 (Density (Møller and Waagepetersen, 2003)) *By the Radon-Nikodym theorem, there exists a function $f : N_{lf} \to [0, \infty]$ such that*

$$P(X_1 \in F) = \mathbb{E}[\mathbb{1}[X_2 \in F] f(X_2)],\ F \subseteq N_{lf}.$$

We call f a density for X_1 with respect to X_2.

Complete Density of NSPs For NSPs equipped with virtual events, the complete density *w.r.t.* Poisson$(S, 1)$ is

$$f(\mathbf{x}, \mathbf{Z}_{1:L} = \mathbf{z}_{1:L}, \tilde{\mathbf{Z}}_{1:L} = \tilde{\mathbf{z}}_{1:L}) = f(\mathbf{z}_L) \prod_{\ell=0}^{L-1} f(\mathbf{z}_\ell \mid \mathbf{z}_{\ell+1}) \tilde{f}(\tilde{\mathbf{z}}_{\ell+1} \mid \mathbf{z}_\ell), \tag{4}$$

where

$$\begin{aligned}
\mathbf{Z}_{1:L} &= \{\mathbf{Z}_1, \mathbf{Z}_2, \cdots, \mathbf{Z}_L\}, \\
\mathbf{z}_{1:L} &= \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_L\}, \\
\tilde{\mathbf{Z}}_{1:L} &= \{\tilde{\mathbf{Z}}_1, \tilde{\mathbf{Z}}_2, \cdots, \tilde{\mathbf{Z}}_L\}, \\
\tilde{\mathbf{z}}_{1:L} &= \{\tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2, \cdots, \tilde{\mathbf{z}}_L\}, \\
f(\mathbf{z}_L) &= \prod_{k=1}^{K_L} f(\mathbf{z}_{L,k}), \\
f(\mathbf{z}_\ell \mid \mathbf{z}_{\ell+1}) &= \prod_{k=1}^{K_\ell} f(\mathbf{z}_{\ell,k} \mid \mathbf{z}_{\ell+1}), \\
\tilde{f}(\tilde{\mathbf{z}}_\ell \mid \mathbf{z}_{\ell-1}) &= \prod_{k=1}^{K_\ell} \tilde{f}(\tilde{\mathbf{z}}_{\ell,k} \mid \mathbf{z}_{\ell-1}) \ .
\end{aligned}$$

The density of $\mathbf{z}_{L,k}$ is

$$f(\mathbf{z}_{L,k}) = \exp\left(T - \mu_{L,k} T\right) \mu_{L,k}^{m_{L,k}}, \tag{5}$$

where $m_{L,k}$ is the number of events drawn from $Z_{L,k}$.

The density of $\mathbf{z}_{\ell,k}$ for $0 \leq \ell \leq L-1$ is

$$f(\mathbf{z}_{\ell,k} \mid \mathbf{z}_{\ell+1}) = \exp\left(\sum_{t_{\ell,k,j} \leq T} \log \lambda_{\ell,k}(t_{\ell,k,j}) + T - \int_0^T \lambda_{\ell,k}(t)\, dt\right). \tag{6}$$

The density of $\tilde{\mathbf{z}}_{\ell,k}$ for $1 \leq \ell \leq L$ is

$$\tilde{f}(\tilde{\mathbf{z}}_{\ell,k} \mid \mathbf{z}_{\ell-1}) = \exp\left(\sum_{\tilde{t}_{\ell,k,j} \leq T} \log \tilde{\lambda}_{\ell,k}(\tilde{t}_{\ell,k,j}) + T - \int_0^T \tilde{\lambda}_{\ell,k}(t)\, dt\right). \tag{7}$$

Remark 5 *Although we only derive equations for point processes in one-dimensional space, most of the results can be generalized to higher-dimensional spaces. We often omit T in densities since T is a constant and it does not affect the derivation.*

Marginal Density The marginal density $f(\mathbf{x})$ *w.r.t.* $\text{Poisson}(S = [0,T]^{\sum_{\ell=1}^{L} K_\ell}, 1)$ of the observation is

$$f(\mathbf{x}) = \mathbb{E}_{\mathbf{Z}_{1:L}}\left[f(\mathbf{x} \mid \mathbf{Z}_{1:L})\right] = \mathbb{E}_{\mathbf{z}_{1:L} \sim \text{Poisson(S,1)}}\left[f(\mathbf{x}, \mathbf{z}_{1:L})\right]. \tag{8}$$

The marginal density $f(\mathbf{x})$ is well-defined as explained in the following proposition:

Proposition 6 *The marginal density $f(\mathbf{x})$ is finite.*

The proof of Proposition 6 can be found in Appendix A.

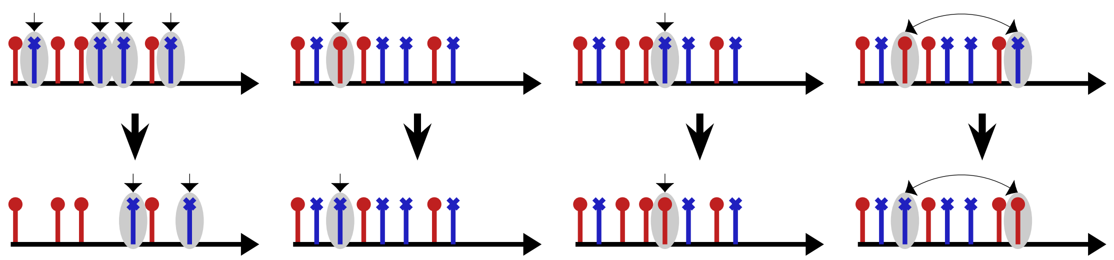
Density of DNSPs as a Limiting Case of Density of DEFs As we discussed earlier in Section 2, the joint density, the conditional density, and the marginal density of DNSPs can also be derived as the limiting value of the joint density, the conditional density, and the marginal density of DEFs when $\delta_L \to 0$.

Theorem 7 *For a realization $(\mathbf{x}, \mathbf{z}_{1:L})$, the joint density of DEFs $p(\mathbf{x}, \mathbf{z}_{1:L})$ converges to the joint density of DNSPs $f(\mathbf{x}, \mathbf{z}_{1:L})$, the conditional density of DEFs $p(\mathbf{x} \mid \mathbf{z}_{1:L})$ converges to the conditional density of DNSPs $f(\mathbf{x} \mid \mathbf{z}_{1:L})$, and the marginal density of DEFs $p(\mathbf{x})$ converges to the marginal density of DEFs $f(\mathbf{x})$ when $\delta_L \to 0$.*

The proof of Theorem 7 can be found in Appendix B.

3.3 Sampler Moves

There are 3 kinds of moves for our RJMCMC sampler, namely, re-sample, flip, and swap, as illustrated in Figure 4 and detailed below. For each MCMC step, we first uniformly select a hidden Poisson process. Then, with some pre-determined probabilities, we randomly select a move from the 3 types of moves and propose to apply this move to our randomly picked



(a) Re-sample (b) Flip: real to virtual (c) Flip: virtual to real (d) Swap

Figure 4: Examples for sampler moves. ● represents a real event. × represents a virtual event.

hidden Poisson process. According to the Hastings ratio, we accept this move randomly. We repeat the process until the sampler reaches a stable distribution of the hidden events.

Move 1: Re-sample. We re-sample the virtual events when this move is selected. It is worth noting that the determinant of the Jacobian matrix, which is the correction for the changes of variables, is 1, since the newly sampled virtual events only depend on the real events. Thus, the probability of accepting this move is always 1.

Remark 8 *During re-sampling, the intensity functions for VPPs depend on the real events on the layer immediately below, instead of the virtual events as described in the generative mechanism for VPPs. We do this because the generated virtual events depending on the real events are closer to the true posterior distribution, and this will not invalidate our posterior sampling or inference algorithms.*

Move 2: Flip. We first uniformly select an event from all of the real and virtual events. Then, we propose to flip the type of the selected event. If the selected event is a real event, then we propose to change the type to virtual, and vice versa.

Move 3: Swap. For this move, we choose two events uniformly from the set of real events and the set of virtual events respectively. For these two events, we propose to swap the types such that the type of the real event is proposed to become virtual, and vice versa.

We denote the proposals for $\mathbf{z}_{\ell,k}$ and $\tilde{\mathbf{z}}_{\ell,k}$ as $\mathbf{z}'_{\ell,k}$ and $\tilde{\mathbf{z}}'_{\ell,k}$ respectively. Then the likelihood ratio is

$$\mathcal{P} = \frac{f(\mathbf{z}'_{\ell,k} \mid \mathbf{z}_{\ell+1})\tilde{f}(\tilde{\mathbf{z}}'_{\ell,k} \mid \mathbf{z}_{\ell-1})}{f(\mathbf{z}_{\ell,k} \mid \mathbf{z}_{\ell+1})\tilde{f}(\tilde{\mathbf{z}}_{\ell,k} \mid \mathbf{z}_{\ell-1})} \cdot \frac{f(\mathbf{z}_{\ell-1} \mid \mathbf{z}'_{\ell})\tilde{f}(\tilde{\mathbf{z}}_{\ell+1} \mid \mathbf{z}'_{\ell})}{f(\mathbf{z}_{\ell-1} \mid \mathbf{z}_{\ell})\tilde{f}(\tilde{\mathbf{z}}_{\ell+1} \mid \mathbf{z}_{\ell})},$$

where $f(\mathbf{z}_{\ell-1} \mid \mathbf{z}_\ell) = \prod_{k=1}^{K_{\ell-1}} f(\mathbf{z}_{\ell-1,k} \mid \mathbf{z}_\ell)$, and $\tilde{f}(\tilde{\mathbf{z}}_{\ell+1} \mid \mathbf{z}_\ell) = \prod_{k=1}^{K_{\ell+1}} \tilde{f}(\tilde{\mathbf{z}}_{\ell+1,k} \mid \mathbf{z}_\ell)$.

The ratio for the proposal probability

$$\mathcal{Q} = \frac{f(\tilde{\mathbf{z}}_{\ell,k} \mid \mathbf{z}_{\ell-1})}{f(\tilde{\mathbf{z}}'_{\ell,k} \mid \mathbf{z}_{\ell-1})}$$

is 1 for both *Move 2* and *Move 3*, as the proposed change would not change the number of all events and the probability of choosing an event remains the same. According to the detailed balance, the acceptance probability for *Move 2* and *Move 3* is

$$\alpha = \min(1, \mathcal{P} \cdot 1).$$

4. Inference

Instead of updating the parameters of Markov kernels as in the inference algorithm proposed by Naesseth et al. (2020), our algorithm gradually updates the parameters of VPPs to minimize the KL divergence. As we mentioned in Section 3.1, VPPs serve as our variational approximation. We briefly outline our algorithm in Algorithm 1. Our inference algorithm approximately maximizes the marginal likelihood of our model (line 13 in Algorithm 1), *w.r.t.* the model parameters $\mathbf{\Theta}$, and minimizes the inclusive KL divergence (line 14 in Algorithm 1) between the posterior distribution and variational distribution, *w.r.t.* the variational parameters $\tilde{\mathbf{\Theta}}$, at the same time. The maximization of the marginal likelihood is performed by stochastic gradient ascent, and the gradient is an unbiased estimation of the gradient of the marginal likelihood (see Theorem 9). The minimization of the inclusive KL divergence is also implemented by stochastic gradient ascent, with the gradient estimated by using Theorem 10. The function for re-sample, flip, and swap can be found in Algorithm 2, 3, and 4 respectively. To make the notations more concise, we let $\hat{\mathbf{\Theta}} = \left[\mathbf{\Theta}, \tilde{\mathbf{\Theta}}\right]$.

Algorithm 1 Inference for NSPs

Input: data $\mathbf{x}$ and model $\mathcal{M}$.
Initialization: parameters for the model $\mathbf{\Theta}_0$, parameters for the VPPs $\tilde{\mathbf{\Theta}}_0$, number of samples for each iteration $\mathcal{S}$, initial sample for RPPs $\mathbf{z}^{(0,\mathcal{S})} = \{\mathbf{z}_1^{(0,\mathcal{S})}, \cdots, \mathbf{z}_L^{(0,\mathcal{S})}\}$, initial sample for VPPs $\tilde{\mathbf{z}}^{(0,\mathcal{S})} = \{\tilde{\mathbf{z}}_1^{(0,\mathcal{S})}, \cdots, \tilde{\mathbf{z}}_L^{(0,\mathcal{S})}\}$, and iterations N.
Output: $\mathbf{\Theta}_N \approx \mathbf{\Theta}^*$, $\tilde{\mathbf{\Theta}}_N \approx \tilde{\mathbf{\Theta}}^*$

```
1: for $n = 1$ to $N$ do
2:     $\mathbf{z}^{(n,0)} = \mathbf{z}^{(n-1,\mathcal{S})}$
3:     for $s = 1$ to $\mathcal{S}$ do
4:         draw $R \sim$ Uniform$([0,1])$
5:         if $R < p$(re-sample) then
6:             $\mathbf{z}^{(n,s)}, \tilde{\mathbf{z}}^{(n,s)} \leftarrow$ Re-sample$(\mathbf{z}^{(n,s-1)}, \tilde{\mathbf{z}}^{(n,s-1)}, \tilde{\mathbf{\Theta}}_{n-1})$
7:         else if $R < p$(re-sample) $+ p$(flip) then
8:             $\mathbf{z}^{(n,s)}, \tilde{\mathbf{z}}^{(n,s)} \leftarrow$ Flip$(\mathbf{z}^{(n,s-1)}, \tilde{\mathbf{z}}^{(n,s-1)}, \mathbf{\Theta}_{n-1}, \tilde{\mathbf{\Theta}}_{n-1})$
9:         else
10:            $\mathbf{z}^{(n,s)}, \tilde{\mathbf{z}}^{(n,s)} \leftarrow$ Swap$(\mathbf{z}^{(n,s-1)}, \tilde{\mathbf{z}}^{(n,s-1)}, \mathbf{\Theta}_{n-1}, \tilde{\mathbf{\Theta}}_{n-1})$
11:        end if
12:    end for
13:    $\mathbf{\Theta}_n \leftarrow \mathbf{\Theta}_{n-1} + \eta_n \frac{1}{\mathcal{S}} \sum_{s=1}^{\mathcal{S}} \nabla_{\mathbf{\Theta}} \log f(\mathbf{x}, \mathbf{z}^{(n,s)}; \mathbf{\Theta}_{n-1})$
14:    $\tilde{\mathbf{\Theta}}_n \leftarrow \tilde{\mathbf{\Theta}}_{n-1} + \tilde{\eta}_n \frac{1}{\mathcal{S}} \sum_{s=1}^{\mathcal{S}} \nabla_{\tilde{\mathbf{\Theta}}} \log \tilde{f}(\mathbf{z}^{(n,s)}; \tilde{\mathbf{\Theta}}_{n-1})$
15: end for
```

Algorithm 2 Re-sample

```
Input: current sample $\mathbf{z}$ and $\tilde{\mathbf{z}}$, parameters $\tilde{\boldsymbol{\Theta}}$.
Output: updated sample $\mathbf{z}'$ and $\tilde{\mathbf{z}}'$.
1: function Re-sample($\mathbf{z}, \tilde{\mathbf{z}}, \tilde{\boldsymbol{\Theta}}$)
2:     draw $(\ell, k) \sim \text{Uniform}\left(\{\{(w,s)\}_{w=1}^{L}\}_{s=1}^{K_w}\right)$
3:     draw $\tilde{\xi} \sim \tilde{\lambda}_{\ell,k}(\cdot)$        ▷ The parameters for $\tilde{\lambda}_{\ell,k}$ are determined by $\tilde{\boldsymbol{\Theta}}$
4:     $\begin{cases} \tilde{\mathbf{z}}'_{\ell,k} = \tilde{\xi} \\ \tilde{\mathbf{z}}'_{\ell',k'} = \tilde{\mathbf{z}}_{\ell',k'} \text{ for } (\ell',k') \neq (\ell,k) \\ \mathbf{z}'_{\ell'',k''} = \mathbf{z}_{\ell'',k''} \text{ for all } (\ell'',k'') \end{cases}$
5:     return $\mathbf{z}'$, $\tilde{\mathbf{z}}'$
6: end function
```

Algorithm 3 Flip

```
Input: current sample $\mathbf{z}$ and $\tilde{\mathbf{z}}$, parameters $\boldsymbol{\Theta}$ and $\tilde{\boldsymbol{\Theta}}$.
Output: updated sample $\mathbf{z}'$ and $\tilde{\mathbf{z}}'$.
1: function Flip($\mathbf{z}, \tilde{\mathbf{z}}, \boldsymbol{\Theta}, \tilde{\boldsymbol{\Theta}}$)
2:     draw $(\ell, k) \sim \text{Uniform}\left(\{\{(w,s)\}_{w=1}^{L}\}_{s=1}^{K_w}\right)$
3:     draw $\xi \sim \text{Uniform}(\{\mathbf{z}_{\ell,k} = \{t_{\ell,k,i}\}_{i=1}^{m_{\ell,k}}, \tilde{\mathbf{z}}_{\ell,k} = \{\tilde{t}_{\ell,k,i}\}_{i=1}^{\tilde{m}_{\ell,k}}\})$
4:     Calculate the acceptance probability $\alpha$
5:     draw $R' \sim \text{Uniform}([0,1])$
6:     if $R' < \alpha$ then
7:         if $\xi = t_{\ell,k,j}$ then
8:             $\begin{cases} \mathbf{z}'_{\ell,k} = \{t_{\ell,k,i}\}_{i=1}^{m_{\ell,k}} \backslash t_{\ell,k,j} \\ \tilde{\mathbf{z}}'_{\ell,k} = \{\tilde{t}_{\ell,k,i}\}_{i=1}^{\tilde{m}_{\ell,k}} \cup t_{\ell,k,j} \end{cases}$
9:         else if $\xi = \tilde{t}_{\ell,k,j}$ then
10:            $\begin{cases} \mathbf{z}'_{\ell,k} = \{t_{\ell,k,i}\}_{i=1}^{m_{\ell,k}} \cup \tilde{t}_{\ell,k,j} \\ \tilde{\mathbf{z}}'_{\ell,k} = \{\tilde{t}_{\ell,k,i}\}_{i=1}^{\tilde{m}_{\ell,k}} \backslash \tilde{t}_{\ell,k,j} \end{cases}$
11:        end if
12:    else
13:        $\begin{cases} \mathbf{z}'_{\ell,k} = \mathbf{z}_{\ell,k} \\ \tilde{\mathbf{z}}'_{\ell,k} = \tilde{\mathbf{z}}_{\ell,k} \end{cases}$
14:    end if
15:    $\begin{cases} \mathbf{z}'_{\ell',k'} = \mathbf{z}_{\ell',k'} \text{ for } (l',k') \neq (l,k) \\ \tilde{\mathbf{z}}'_{\ell',k'} = \tilde{\mathbf{z}}_{\ell',k'} \text{ for } (l',k') \neq (l,k) \end{cases}$
16:    return $\mathbf{z}'$, $\tilde{\mathbf{z}}'$
17: end function
```

Algorithm 4 Swap

Input: current sample $\mathbf{z}$ and $\tilde{\mathbf{z}}$, parameters $\boldsymbol{\Theta}$ and $\tilde{\boldsymbol{\Theta}}$.
Output: updated sample $\mathbf{z}'$ and $\tilde{\mathbf{z}}'$.

```
1: function Flip($\mathbf{z}, \tilde{\mathbf{z}}, \boldsymbol{\Theta}, \tilde{\boldsymbol{\Theta}}$)
2:     draw $(\ell, k) \sim \text{Uniform}\left(\{\{(w,s)\}_{w=1}^{L}\}_{s=1}^{K_w}\right)$
3:     draw $\xi \sim \text{Uniform}(\{\mathbf{z}_{\ell,k} = \{t_{\ell,k,i}\}_{i=1}^{m_{\ell,k}})$
4:     draw $\tilde{\xi} \sim \text{Uniform}(\{\tilde{\mathbf{z}}_{\ell,k} = \{\tilde{t}_{\ell,k,i}\}_{i=1}^{\tilde{m}_{\ell,k}}\})$
5:     Calculate the acceptance probability $\alpha$
6:     draw $R' \sim \text{Uniform}([0,1])$
7:     if $R' < \alpha$ then
8:         $\begin{cases} \mathbf{z}'_{\ell,k} = \{t_{\ell,k,i}\}_{i=1}^{m_{\ell,k}} \backslash \xi \cup \tilde{\xi} \\ \tilde{\mathbf{z}}'_{\ell,k} = \{\tilde{t}_{\ell,k,i}\}_{i=1}^{\tilde{m}_{\ell,k}} \backslash \tilde{\xi} \cup \xi \end{cases}$
9:     else
10:        $\begin{cases} \mathbf{z}'_{\ell,k} = \mathbf{z}_{\ell,k} \\ \tilde{\mathbf{z}}'_{\ell,k} = \tilde{\mathbf{z}}_{\ell,k} \end{cases}$
11:    end if
12:    $\begin{cases} \mathbf{z}'_{\ell',k'} = \mathbf{z}_{\ell',k'} \text{ for } (l',k') \neq (l,k) \\ \tilde{\mathbf{z}}'_{\ell',k'} = \tilde{\mathbf{z}}_{\ell',k'} \text{ for } (l',k') \neq (l,k) \end{cases}$
13:    return $\mathbf{z}'$, $\tilde{\mathbf{z}}'$
14: end function
```

4.1 Likelihood-based Inference for the Model Parameters

Similar to Ou and Song (2020); Naesseth et al. (2020), we notice that the numerical approximation $\frac{1}{\mathcal{S}}\sum_{s=1}^{\mathcal{S}} \nabla_{\boldsymbol{\Theta}} \log f(\mathbf{x}, \mathbf{z}^{(n,s)}; \boldsymbol{\Theta}_{n-1})$ at line 13 in Algorithm 1 is an unbiased estimate of the gradient of the marginal likelihood, as stated in the following theorem:

Theorem 9 *Suppose* $\left|\hat{\boldsymbol{\Theta}}\right| \leq r$ *for some* $r > 0$*, then*

$$\mathbb{E}_{\mathbf{z}\sim f(\mathbf{Z}|\mathbf{x})}[\nabla_{\boldsymbol{\Theta}} \log f(\mathbf{x}, \mathbf{z})] = \nabla_{\boldsymbol{\Theta}} \log f(\mathbf{x}).$$

We leave the proof of Theorem 9 to Section C.1. Theorem 9 implies that the update of the parameters based on stochastic gradient ascent at line 13 in Algorithm 1 is equivalent to approximately maximizing the marginal likelihood.

4.2 Variational Inference for the Variational Parameters

Notice that the inclusive KL divergence between the true posterior $\mathbf{Z} \mid \mathbf{x}$ and the approximate posterior (VPPs) $\tilde{\mathbf{Z}}$ can be written as

$$\mathrm{KL}(f(\mathbf{Z} \mid \mathbf{x}; \boldsymbol{\Theta}) \parallel \tilde{f}(\mathbf{Z}; \tilde{\boldsymbol{\Theta}})) = \mathbb{E}_{\mathbf{z}\sim f(\mathbf{Z}|\mathbf{x})}\left[\log f(\mathbf{z} \mid \mathbf{x}; \boldsymbol{\Theta}) - \log \tilde{f}(\mathbf{z}; \tilde{\boldsymbol{\Theta}})\right], \tag{9}$$

where $\tilde{f}(\mathbf{z}; \tilde{\boldsymbol{\Theta}})$ is the density of the VPPs, and

$$\tilde{f}(\mathbf{z}; \tilde{\boldsymbol{\Theta}}) = \prod_{\ell=1}^{L} \tilde{f}(\mathbf{z}_\ell \mid \mathbf{z}_{\ell-1}; \tilde{\boldsymbol{\Theta}}),$$

where $\mathbf{z}_0$ is equal to $\mathbf{x}$.

We also find that the gradient of the inclusive KL divergence in Equation 9 can be represented as an expectation of the gradient of the densities of the samples from posterior distributions, formally described in the following theorem:

Theorem 10 *Suppose* $\left|\hat{\mathbf{\Theta}}\right| \leq r$ *for some* $r > 0$*, then*

$$\mathbb{E}_{\mathbf{z} \sim f(\mathbf{Z}|\mathbf{x};\mathbf{\Theta})}\left[\nabla_{\tilde{\mathbf{\Theta}}} \log \tilde{f}(\mathbf{z}; \tilde{\mathbf{\Theta}})\right] = \nabla_{\tilde{\mathbf{\Theta}}} \mathbb{E}_{\mathbf{z} \sim f(\mathbf{Z}|\mathbf{x};\mathbf{\Theta})}\left[\log \tilde{f}(\mathbf{z}; \tilde{\mathbf{\Theta}})\right].$$

We leave the proof of Theorem 10 to Section C.2.

As described in this section, Algorithm 1 performs maximum likelihood estimation for the model parameters and variational inference for the variational parameters simultaneously. Moreover, as we will show in section 6, the convergence of the optimization for the marginal likelihood and the inclusive KL divergence depends on the convergence of RJMCMC. Thus, we first prove the convergence of RJMCMC in the following section.

5. Convergence of RJMCMC

We analyze the aperiodicity, Ψ-irreducibility, and Harris recurrence for RJMCMC. A sufficient condition for the convergence is given in this section and we demonstrate how to verify the convergence by applying the analysis to NSPs, Markov jump processes (MJPs) (Rao and Teh, 2013), piecewise-constant conditional intensity models (PCIMs) (Qin and Shelton, 2015), and Hawkes processes (Shelton et al., 2018).

The analysis of previous work (Rao and Teh, 2013; Qin and Shelton, 2015; Shelton et al., 2018; Hong and Shelton, 2022) implies that the posterior distribution exists and there is a *positive probability density* moving from any point configuration to any other. However, these results cannot guarantee the algorithm can move from any point configuration to any other set with a probability measure greater than 0 and it may get stuck within a null set with a measure of 0. Harris recurrence provides additional conditions such that the algorithm will not always stay within a null set.

5.1 Virtual-Event-Based RJMCMC

We first describe a virtual-event-based RJMCMC. We denote the trans-dimensional Markov chain as $\mathcal{C} = (\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \cdots)$ with the state space as Ω. The real events represent the events in our desired posterior distribution, and the virtual events are auxiliary events that only serve as the candidates for the real events. Given that the chain is in state $\mathcal{C}_n$, it proposes to move to a state $\mathcal{C}'_{n+1}$ with proposal probability $g(\mathcal{C}_n, \mathcal{C}'_{n+1})$. We accept this move with probability $\alpha(\mathcal{C}_n, \mathcal{C}'_{n+1})$ and reject this move with probability $1 - \alpha(\mathcal{C}_n, \mathcal{C}'_{n+1})$. The proposal of Markov chain $\mathcal{C}$ can be categorized into two types: (1) re-sampling virtual events from some Poisson processes, which involves the dimension changing; (2) changing the states of the events (virtualness, marks or the parent-child relationships in Hawkes processes), which does not involve the changing of the number of dimensions.

Remark 11 *According to the design of the virtual-event-based RJMCMC, the real events can only appear where the virtual events occur.*

Suppose we have an unnormalized target density function for real and virtual point processes $f_{r+v} : (\Omega_r^R, \Omega_v^V) \to (0, \infty)$ with $\int_{(\Omega_r^R, \Omega_v^V)} f_{r+v} < \infty$, where $r, v \in \mathbb{N}$ and (Ω_r^R, Ω_v^V) is of dimension $r + v$. We combine different spaces with different dimensions into a single state space

$$\Omega = \cup_{r=1}^{\infty} \cup_{v=1}^{\infty} \{((\{r\} \times \Omega_r^R), (\{v\} \times \Omega_v^V))\},$$

where r represents the number of points for a realization of RPPs, Ω_r^R represents the space of dimension r where RPPs reside, x_r is a realization of RPPs, v represents the number of points for a realization of virtual point processes, Ω_v^V represents the space of dimension v where virtual point processes reside, and x_v is a realization of virtual point processes.

The Markov chain $\mathcal{C}$ is reversible if the detailed balance is satisfied, which also implies that the invariant distribution Π exists. Following Møller and Waagepetersen (2003), we let

$$Q^m(F) = \mathbb{E}P^m(\mathcal{C}_0, F) = P(\mathcal{C}_m \in F), \ F \subseteq \Omega,$$

denote the marginal distribution of $\mathcal{C}_m$, where $P^m(\mathcal{C}_0, F) = P(\mathcal{C}_m \in F \mid \mathcal{C}_0)$ is the m-step transition probability. Then, we use the total variation norm to measure the distance between Q^m and the invariant distribution Π, and the total variation norm is defined as

Definition 12 (total variation norm (Møller and Waagepetersen, 2003)) *The total variation norm for any two probability distributions μ and ν defined on Ω is*

$$\|\mu - \nu\|_{\text{TV}} = \sup_{F \subseteq \Omega} |\mu(F) - \nu(F)|.$$

We first define *drift combination* in Definition 13. Then, we use drift combination to help describe the main result of the convergence analysis in this section in Theorem 14.

Definition 13 (drift combination) *A drift combination is the combination of moves that change a real event to a virtual event followed by a re-sampling of virtual events.*

Theorem 14 *For a Ψ-irreducile and aperiodic Markov chain described in virtual-event-based RJMCMC, if the re-sampling of virtual events and the acceptance of the drift combination for each real event happen eventually, then*

$$\lim_{m \to \infty} \|Q^m - \Pi\|_{\text{TV}} = 0$$

holds for this Markov chain, where Π is the invariant distribution.

We provide a sketch of proof for Theorem 14 in Section 5.2 and more details can be found in Appendix D.

Now it becomes much easier to see whether a Markov chain can converge, since we only need to check aperiodicity, Ψ-irreducibility, and whether the re-sampling of virtual events and a drift combination for each real event will be accepted eventually with probability 1.

5.2 Markov Chains and Harris Recurrence

Definition 15 (Ψ-irreducible (Møller and Waagepetersen, 2003)) *For any $x \in \Omega$ and $F \subseteq \Omega$, if there exists a nonzero measure Ψ on Ω such that $\Psi(F) > 0$, $P^m(x, F) > 0$ for some $m \in \mathbb{N}$, then this Markov chain is Ψ-irreducible.*

Definition 16 (aperiodicity (Møller and Waagepetersen, 2003)) *For a Ψ-irreducible chain, if Ω can be partitioned into d disjoint subsets $D_0, \cdots, D_{d-1}$ with $\Psi(D_i) > 0$, such that $P(x, D_{i+1}) = 1$ for all $x \in D_i$ $(1 \leq i \leq D-1)$ and $P(x, D_0) = 1$ for all $x \in D_{d-1}$. If there exists a partition such that $d > 1$, the chain is periodic, else it is aperiodic.*

If a Markov chain is Ψ-irreducible and aperiodic, then this Markov chain converges to the invariant distribution Π from almost every starting point. However, this kind of convergence is not good enough for our RJMCMC algorithm, since there may exist a null set of point configurations where the Markov chain cannot converge with any point configuration as a starting point. To exclude the null set, we consider Harris recurrence:

Definition 17 (Harris recurrence (Møller and Waagepetersen, 2003))
A Ψ-irreducible Markov chain is Harris recurrent if for some probability measure Ψ, and that for all $x \in \Omega$ and all $F \subseteq \Omega$ with $\Psi(F) > 0$,

$$P(Y_m \in F \text{ for some } m \mid Y_0 = x) = 1.$$

With Harris recurrence, we can use the following proposition to prove our RJMCMC algorithm converges to Π without any null sets involved:

Proposition 18 (Møller and Waagepetersen (2003, Proposition 7.7 (iii)))
The Markov chain $\mathcal{C}$ is Harris recurrent and aperiodic if and only if

$$\lim_{m \to \infty} \|Q^m - \Pi\|_{\mathrm{TV}} = 0$$

regardless of the initial distribution.

From Proposition 18, we know that the only remaining tasks for the proof of convergence are the verifications of aperiodicity and Harris recurrence.

The verification of aperiodicity is pretty easy and will be discussed in Section 5.3. The verification of Harris recurrence, however, is not straightforward and we would like to use the following theorem to find a criterion that can be used to check Harris recurrence practically:

Theorem 19 (Roberts and Rosenthal (2006, Theorem 6(vi))) *For a Markov chain which is Ψ-irreducible with invariant probability distribution Π and period $d \geq 1$, the chain is Harris recurrent if and only if for all $x \in \Omega$ and all $F \subseteq \Omega$ with $\Pi(F) = 0$,*

$$P(\forall n \; \mathcal{C}_n \in F \mid \mathcal{C}_0 = x) = 0.$$

Theorem 19 provides us a weak condition under which a Ψ-irreducible Markov chain with invariant distribution is Harris recurrent. Following this condition, we only need to show that there exists $x \in \Omega$, $F \subseteq \Omega$ with $\Pi(F) = 0$, and a number $n_0 \in \mathbb{N}$ such that $P(\mathcal{C}_{n_0} \in F \mid \mathcal{C}_0 = x) = 0$, then it follows immediately that

$$P(\forall n\ \mathcal{C}_n \in F \mid \mathcal{C}_0 = x) \leq P(\mathcal{C}_{n_0} \in F \mid \mathcal{C}_0 = x) = 0.$$

We use $\Pi(r, x_r, v, x_v)$ to denote the invariant probability distribution for the posterior point processes, and it can be set as

$$\Pi(r, A_r, v, A_v) = p(r, v) \frac{\int_{(A_r, A_v)} f_{r+v}(x) \mathcal{U}_{r+v}(dx)}{\int_{(\Omega_r, \Omega_v)} f_{r+v}(x) \mathcal{U}_{r+v}(dx)}, \tag{10}$$

where $p : (\mathbb{N}, \mathbb{N}) \to (0, 1)$ with $\sum_r \sum_v p(r, v) = 1$, $\mathcal{U}_{r+v}(\cdot)$ is the Lebesgue measure on $\mathbb{R}^{d_r + d_v}$. $\Pi(\cdot)$ could be 0 for impossible point configurations. For example, if the intensity for an interval is 0, then there would never be any event in this interval.

To find under which condition the Harris recurrence can be established, we introduce *drift* here:

Definition 20 (drift) *A drift for an event is the behavior in which not only the state of the event is changed, but the location is also changed, including completely removed from the realization.*

Proposition 21 *Let $M_i(x, \cdot)$ be the i-th transition kernel that proposes to move a point configuration $x \in \Omega$ to another point configuration $x' \in \Omega$, and $(i_1, i_2, \cdots, i_n)$ be any sequence of transition kernels. Assume that each of the point of x accepts at least one drift in $(i_1, i_2, \cdots, i_n)$, then*

$$(M_{i_1} M_{i_2} \cdots M_{i_n})(x, A) = 0$$

for all $x \in \Omega$ and $A \subseteq \Omega$ with the Lebesgue measure of A is 0.

The following corollary follows immediately from Proposition 21:

Corollary 22 *If the Lebesgue measure of A is 0, then*

$$P[X_n \in A \mid X_0 = x_0] \leq P[D_n],$$

where $X_n \in \Omega$, and D_n is the event that, by time n, the chain has not yet accepted at least one drift for each event of the original starting realization.

Combining Corollary 22 and Theorem 19, the following proposition follows immediately:

Proposition 23 *If a Ψ-irreducible Markov chain eventually accepts at least one drift for each real and virtual event, then this Markov chain is Harris recurrent.*

Proposition 23 gives us a condition under which Harris recurrence can hold. If the re-sampling and the drift combination described in Theorem 14 happen eventually, then the probability of accepting a drift for each real and virtual event eventually would be 1, which leads to Harris recurrence and Theorem 14 follows easily from it.

5.3 Verification of Aperiodicity

The verification of aperiodicity is simple, as we only need to check whether a Markov chain can leave and go back to the same state, *i.e.*, $P(x, \{x\}) > 0$ for some $x \in \Omega$.

5.4 Verification of Ψ-irreducibility

For the verification of Ψ-irreducibility, we need to find a probability measure Ψ which meets the requirement for Ψ-irreducibility. Here we give one way of finding such probability measure Ψ, and we need to introduce a new concept:

Definition 24 (continuous set of point configurations) *Let y be the evidence, x be a point configuration sampled from MCMC for the missing information, $l(x, y)$ be the joint likelihood of the evidence y and the sample x, $U_r^R \subseteq \Omega_r^R$, and $U_v^V \subseteq \Omega_v^V$. A continuous set of point configurations is defined as*

$$U = \{((\{r\} \times U_r^R), (\{v\} \times U_v^V)) : l(x = (x_r \in U_r, x_v \in U_v), y) > 0, r \geq 0, v \geq 0\},$$

if U_r^R is either an empty set or has the Lebesgue measure greater than 0, and the same for U_v^V. Notice that $U_0^V = U_0^R = \emptyset$.

A Ψ-irreducible Markov chain needs to have a positive probability of moving any point configuration x to a set with a positive measure. From the definition of the intensity function and density, it is easy to see that the probability of generating a single point configuration is 0. This implies that we need a set of point configurations such that the probability of generating this set of point configurations is greater than 0, which leads to the introduction of the continuous set of point configurations and the following theorem follows immediately.

Theorem 25 *If a Markov chain is Ψ-irreducible, then there exists a continuous set of point configurations U for this Markov chain.*

However, the existence of a continuous set of point configurations cannot guarantee Ψ-irreducibility, because there is an additional requirement implied by the definition of Ψ-irreducibility that the probabilities of transitions with a finite number of steps from any state x to the continuous set of point configurations U should be greater than 0.

Thus, we need to check whether a continuous set of point configurations exists and whether any state can be moved to the continuous set of point configurations for each specific algorithm. If both of the above conditions are satisfied, we can just let

$$\Psi(F) = \mathbb{1}[U \in F], \ F \subseteq \Omega,$$

where $\mathbb{1}[\cdot]$ is the indicator function.

5.5 Examples of Convergence Analysis

Based on the above analysis of the convergence, we would like to demonstrate how to verify the convergence of virtual-event-based MCMC algorithms with some examples, including the convergence for NSPs, MJPs, PCIMs, and Hawkes processes.

5.5.1 Neyman-Scott Processes

Proposition 26 *The Markov chain $\mathcal{C}$ generated by Algorithm 1 is Ψ-irreducible and aperiodic. The continuous set of point configurations can be chosen as*

$$U = \left\{ \left(\left(\left\{ r = \sum_{\ell} K_\ell \right\} \times U_r^R \right), (\{0\} \times U_0^V) \right) \right\},$$

where U_r^R is the set of points such that $\mathbf{z}_{\ell,k} = \{t_{\ell,k,0} : 0 < t_{\ell,k,0} < \min_{i,j}\{t_{\ell-1,i,j}\}\}$.

Remark 27 *We have the restriction in Proposition 26 that each hidden point process has one real event because the intensities for RPPs are solely dependent on the real events. This restriction can be removed if we add a constant base rate to the intensity in Equation 1, like the intensity functions for VPPs. In such a case, U can be set as*

$$\{(\{0\} \times U_0^R), (\{0\} \times U_0^V)\}.$$

Proposition 28 *The re-sampling of virtual events and the drift combination for each real event happen eventually for the Markov chain generated by Algorithm 1.*

The proofs can be found in Appendix D.1, and the convergence then follows from Proposition 26, Proposition 28, and Theorem 14.

5.5.2 Markov Jump Processes (MJPs) and Extensions

Rao and Teh (2013) propose an MCMC algorithm for MJPs and their extensions. An MJP is a stochastic process with random paths that depict jumps from one state to another state. An MJP path can be specified by a 3-element tuple (s_0, S, T) where s_0 is the initial state, $T = (t_1, t_2, \cdots, t_n)$ are the ordered times where the state changes happen, and $S = (s_1, s_2, \cdots, s_n)$ are the corresponding states after the changing of the states.

Remark 29 *In the semantics of virtual-event-based RJMCMC, each point x for MJP is represented as $x = (t, s)$, where t represents the time and s represents the state. We exclude the initial state $(0, s_0)$ from our RJMCMC state space as the initial time is always set as 0 and only the initial state s_0 can be changed.*

Rao and Teh (2013) propose a uniformization-based Gibbs sampling algorithm for MJP-based models. The Markov chain first re-samples a set of virtual jumps as auxiliary variables, and then samples a new MJP path conditional on the times of the union of the newly sampled virtual jumps and the jumps from the current MJP path. The algorithm repeats the above two sampling steps alternately until convergence. The virtual jumps are jumps that do not change the state of the MJP, which are equivalent to virtual events described in this paper. The normal jumps, corresponding to the real events, leave the current states and move to other states.

When implementing the uniformization-based Gibbs sampling algorithm, Rao and Teh (2013) sample the times of the virtual jumps $J_{\mathcal{T}}$ from an inhomogeneous Poisson process with intensity $R(t) = \lambda_\Omega + A_{S(t)}$, where $A_{S(t)}$ represents the transition matrix when MJP is at state $S(t)$ at time t and $\lambda_\Omega \geq \max_s |A_s|$ is a positive constant. The states of the virtual jumps $J_\mathcal{S}$ are determined by $J_\mathcal{T}$ and the current MJP trajectory (s_0, S, T). The union of $J_\mathcal{T}$ and the times of the current MJP T is equivalent to the samples drawn from a Poisson process with rate λ_Ω, and we use $W = (w_1, \cdots, w_{|W|})$ to denote the times of this union. Given W, we can sample an MJP path by assigning each w_i as a virtual jump or a normal jump by performing the forward-filtering backward-sampling algorithm with $1 + |W|$ steps. The likelihood of state s at step i is given as

$$L_i(s) = p(O_{[w_i, w_{i+1})} \mid S(t) = s \text{ for } t \in [w_i, w_{i+1})]),$$

where $O_{[w_i,w_{i+1})}$ is the observations in the interval $[w_i, w_{i+1})$.

Proposition 30 *If there exists a continuous set of point configurations U, then the Markov chain for the MCMC algorithm proposed in Rao and Teh (2013) is convergent.*

From Proposition 30, we know the convergence of MCMC for an MJP depends on the likelihood $L_i(s)$, as the joint likelihood of the MJP states and the observations depends on $L_i(s)$ and the set U requires the joint likelihood to be positive. More details can be found in Appendix D.2. Here is an example:

Example 5 (Markov-Modulated Poisson Processes) *The likelihood at step i is given by*

$$L_i(s) = (\lambda_s)^{|O_i|} \exp(-\lambda_s(w_{i+1} - w_i)),$$

where λ_s is a nonnegative constant associated with the state s, $|O_i|$ is the number of events of the observation in the interval $[w_i, w_{i+1})$. If there exists a state s^* such that $\lambda_{s^*} > 0$, then the probability of staying at state s^* for the MJP is greater than 0 and U can be

$$\{((\{0\} \times U_0^R), (\{0\} \times U_0^V))\}.$$

Then, the convergence is implied by Proposition 30.

5.5.3 Piecewise-Constant Conditional Intensity Models (PCIMs)

A PCIM is a model that uses decision trees to determine the piecewise-constant conditional intensity functions. Different tests are performed at each internal node to propagate the information to the leaves. Different leaves have different constant intensity functions and form the final piecewise-constant conditional intensity functions. Qin and Shelton (2015) propose an MCMC for PCIM to infer the missing events in an unobserved time period. A PCIM assumes the events can be represented as a sequence $x = \{(t_i, l_i)\}_{i=1}^n$, where $0 < t_1 < \cdots < t_n$ and l_i is from a finite label set L. The data likelihood for x is

$$p(x) = \prod_{l \in L} \prod_{s \in \Sigma_l} \lambda_{ls}^{c_{ls}(x)} e^{-\lambda_{ls} d_{ls}(x)},$$

where $\sum_l$ is the set of leaves corresponding to label l, $c_{ls}(x)$ is the number of times label l occurs in x for leaf s, $d_{ls}(x)$ is the total duration when the decision tree maps the event sequence x to leaf s for label l.

Similar to Rao and Teh (2013), Qin and Shelton (2015) first re-samples a set of auxiliary variables (virtual events), and then samples a new event sequence by re-specifying the virtualness for each event from the union of the newly sampled virtual events and the current event sequence. The convergence analysis is given as Proposition 31, which is similar to Proposition 30 and we omit the proof.

Proposition 31 *If there exists a continuous set of point configurations U, then the Markov chain for the MCMC algorithm proposed in Qin and Shelton (2015) is convergent.*

5.5.4 Hawkes Processes

Hawkes processes are a class of point processes whose intensity functions for label l are written as

$$\lambda_l = \mu_l + \sum_{t_i : t_i < t} \phi_{l_i,l}(t - t_i),$$

where $\mu_l > 0$ is the base rate for events with label l and $\phi_{l_i,l}(\cdot)$ is a kernel function that indicates the sudden increase of the intensity function triggered by the events from the history. Possible functional forms for $\phi_{l_i,l}(\cdot)$ include the exponential function ($\phi(t) = e^{-\beta t}$ with $\beta > 0$) and the power-law function ($\phi(t) = (t + \gamma)^{-(1+\beta)}$ with $\beta > 0$ and $\gamma > 0$).

Shelton et al. (2018) propose an MCMC algorithm for Hawkes processes with missing data. 3 moves are designed for the MCMC algorithm: virtual children, virtualness, and parent. The virtual children move proposes to replace the current set of virtual children with a new set of virtual children for a given index; the virtualness move proposes to switch the type of event between real and virtual for a given event; the parent move proposes to re-sample the parent for a chosen event. Given the observed data, we uniformly select a move at random at each MCMC step until convergence. We choose an empty set for the continuous set of point configurations for Hawkes processes because, in contrast to the intensities we design for the NSPs, the Hawkes processes allow the extraneous intensity to be 0, as explained in Remark 27. Thus, we have the following propositions and the convergence follows directly. More details of the proofs can be found in Appendix D.3.

Proposition 32 *The Markov chain generated by the MCMC algorithm proposed in Shelton et al. (2018) is Ψ-irreducible and aperiodic. The continuous set of point configurations can be*

$$U = \{((\{0\} \times U_0^R), (\{0\} \times U_0^V))\}.$$

Proposition 33 *The re-sampling of virtual events and the drift combination for each real event happen eventually for the Markov chain in Shelton et al. (2018).*

6. Convergence of Optimization

The general algorithm for the optimization in Algorithm1 is of the form

$$\mathbf{\Theta}_{n+1} = \mathbf{\Theta}_n + \gamma_{n+1} H(\mathbf{\Theta}_n, X_{n+1}), \tag{11}$$

and

$$\tilde{\mathbf{\Theta}}_{n+1} = \tilde{\mathbf{\Theta}}_n + \gamma_{n+1} \tilde{H}(\tilde{\mathbf{\Theta}}_n, X_{n+1}), \tag{12}$$

where $\boldsymbol{\Theta}_n$ and $\tilde{\boldsymbol{\Theta}}_n$ are in $\mathbb{R}^d$ and X_n represents the point configurations in the whole space, $H(\boldsymbol{\Theta}_n, X_{n+1})$ and $\tilde{H}(\tilde{\boldsymbol{\Theta}}_n, X_{n+1})$ are approximations of

$$\mathbb{E}_{\mathbf{z}\sim f(\mathbf{Z}|\mathbf{x};\boldsymbol{\Theta}_n)}[\nabla_{\boldsymbol{\Theta}_n} \log f(\mathbf{x}, \mathbf{z}; \boldsymbol{\Theta}_n)]$$

and

$$\mathbb{E}_{\mathbf{z}\sim f(\mathbf{Z}|\mathbf{x};\boldsymbol{\Theta}_n)}[\nabla_{\tilde{\boldsymbol{\Theta}}_n} \log \tilde{f}(\mathbf{z}; \mathbf{x}, \tilde{\boldsymbol{\Theta}}_n)]$$

respectively. We get $\mathcal{S}$ samples from the MCMC with target distribution as $f(\mathbf{z} \mid \mathbf{x}; \boldsymbol{\Theta}_n)$. The samples are denoted as $\mathbf{z}^1, \mathbf{z}^2, \cdots, \mathbf{z}^{\mathcal{S}}$. Then

$$H(\boldsymbol{\Theta}_n, X_{n+1}) = \frac{1}{\mathcal{S}} \sum_{i=1}^{\mathcal{S}} \nabla_{\boldsymbol{\Theta}_n} \log f(\mathbf{x}, \mathbf{z}^i; \boldsymbol{\Theta}_n)$$

and

$$\tilde{H}(\tilde{\boldsymbol{\Theta}}_n, X_{n+1}) = \frac{1}{\mathcal{S}} \sum_{i=1}^{\mathcal{S}} \nabla_{\tilde{\boldsymbol{\Theta}}_n} \log \tilde{f}(\mathbf{z}^i; \mathbf{x}, \tilde{\boldsymbol{\Theta}}_n).$$

The proof of convergence of this optimization that follows is largely modified from Benveniste et al. (2012, Chapter 5).

Assumption 34 *Given the history $F_n = \{X_i, \boldsymbol{\Theta}_i, \tilde{\boldsymbol{\Theta}}_i\}_{i=1}^n$ until the n-th step, we assume the approximations approach the true posterior distribution, i.e.,*

$$\mathbb{E}[H(\boldsymbol{\Theta}_n, X_{n+1}) \mid F_n] = \mathbb{E}_{\mathbf{z}\sim f(\mathbf{Z}|\mathbf{x};\boldsymbol{\Theta}_n)}[\nabla_{\boldsymbol{\Theta}_n} \log f(\mathbf{x}, \mathbf{z}; \boldsymbol{\Theta}_n)] \tag{13}$$

$$\mathbb{E}[\tilde{H}(\tilde{\boldsymbol{\Theta}}_n, X_{n+1}) \mid F_n] = \mathbb{E}_{\mathbf{z}\sim f(\mathbf{Z}|\mathbf{x};\boldsymbol{\Theta}_n)}[\nabla_{\tilde{\boldsymbol{\Theta}}_n} \log \tilde{f}(\mathbf{z}; \mathbf{x}, \tilde{\boldsymbol{\Theta}}_n)] \tag{14}$$

Remark 35 *Theoretically, when the number of samples $\mathcal{S}$ goes to infinity, the approximation approaches the true posterior, i.e.,*

$$\lim_{\mathcal{S}\to\infty} H(\boldsymbol{\Theta}_n, X_{n+1}) = \mathbb{E}_{\mathbf{z}\sim f(\mathbf{Z}|\mathbf{x};\boldsymbol{\Theta}_n)}[\nabla_{\boldsymbol{\Theta}_n} \log f(\mathbf{x}, \mathbf{z}; \boldsymbol{\Theta}_n)]$$

and

$$\lim_{\mathcal{S}\to\infty} \tilde{H}(\tilde{\boldsymbol{\Theta}}_n, X_{n+1}) = \mathbb{E}_{\mathbf{z}\sim f(\mathbf{Z}|\mathbf{x},\boldsymbol{\Theta}_n)}[\nabla_{\tilde{\boldsymbol{\Theta}}_n} \log \tilde{f}(\mathbf{z}; \mathbf{x}, \tilde{\boldsymbol{\Theta}}_n)].$$

In practice, we choose the number of samples large enough to make our algorithm stable.

Let $\hat{\boldsymbol{\Theta}}_* = \left[\boldsymbol{\Theta}_*, \tilde{\boldsymbol{\Theta}}_*\right]$ and $\hat{H}(\hat{\boldsymbol{\Theta}}, X) = \left[H(\boldsymbol{\Theta}, X), \tilde{H}(\tilde{\boldsymbol{\Theta}}, X)\right]$, then we have the following proposition and theorem.

Proposition 36

$$\forall r > 0, \sup_{|\hat{\boldsymbol{\Theta}}|\le r} \mathbb{E}_{\mathbf{z}\sim f(\mathbf{Z}|\mathbf{x};\boldsymbol{\Theta})} \left[|\hat{H}(\hat{\boldsymbol{\Theta}}, X)|^2\right] < \infty, \tag{15}$$

$$\forall r > 0, \sup_{|\hat{\boldsymbol{\Theta}}|\le r} h(\hat{\boldsymbol{\Theta}}) = \mathbb{E}_{\mathbf{z}\sim f(\mathbf{Z}|\mathbf{x};\boldsymbol{\Theta})} \left[|\hat{H}(\hat{\boldsymbol{\Theta}}, X)|\right] < \infty, \tag{16}$$

where $|\cdot|$ represents the length of a vector.

Proof When the sample size $\mathcal{S}$ goes to infinity, H converges to the expectation, and we can choose $\mathcal{S}$ such that for any $\epsilon > 0$,

$$H(\mathbf{\Theta}_n, X_{n+1}) = \mathbb{E}_{\mathbf{z} \sim f(\mathbf{Z}|\mathbf{x};\mathbf{\Theta}_n)}[\nabla_{\mathbf{\Theta}_n} \log f(\mathbf{x}, \mathbf{z}; \mathbf{\Theta}_n)] + \epsilon$$

and

$$\tilde{H}(\tilde{\mathbf{\Theta}}_n, X_{n+1}) = \mathbb{E}_{\mathbf{z} \sim f(\mathbf{Z}|\mathbf{x},\mathbf{\Theta}_n)}[\nabla_{\tilde{\mathbf{\Theta}}_n} \log \tilde{f}(\mathbf{z}; \mathbf{x}, \tilde{\mathbf{\Theta}}_n)] + \epsilon.$$

Thus, we only need to verify the following equations are true.

$$\forall r > 0, \sup_{|\hat{\mathbf{\Theta}}| \le r} \left|\mathbb{E}_{\mathbf{z} \sim f(\mathbf{Z}|\mathbf{x};\mathbf{\Theta}_n)}[\nabla_{\mathbf{\Theta}_n} \log f(\mathbf{x}, \mathbf{z}; \mathbf{\Theta}_n)]\right| < \infty \tag{17}$$

$$\forall r > 0, \sup_{|\hat{\mathbf{\Theta}}| \le r} \left|\mathbb{E}_{\mathbf{z} \sim f(\mathbf{Z}|x;\mathbf{\Theta}_n)}[\nabla_{\tilde{\mathbf{\Theta}}_n} \log \tilde{f}(\mathbf{z}; \mathbf{x}, \tilde{\mathbf{\Theta}}_n)]\right| < \infty \tag{18}$$

$$\forall r > 0, \sup_{|\hat{\mathbf{\Theta}}| \le r} \left|\mathbb{E}_{\mathbf{z} \sim f(\mathbf{Z}|\mathbf{x};\mathbf{\Theta}_n)}[\nabla_{\mathbf{\Theta}_n} \log f(\mathbf{x}, \mathbf{z}; \mathbf{\Theta}_n)]\right|^2 < \infty \tag{19}$$

$$\forall r > 0, \sup_{|\hat{\mathbf{\Theta}}| \le r} \left|\mathbb{E}_{\mathbf{z} \sim f(\mathbf{Z}|\mathbf{x};\mathbf{\Theta}_n)}[\nabla_{\tilde{\mathbf{\Theta}}_n} \log \tilde{f}(\mathbf{z}; \mathbf{x}, \tilde{\mathbf{\Theta}}_n)]\right|^2 < \infty \tag{20}$$

Equations 17 and 19 have been verified in Theorem 9. Equations 18 and 20 have been verified in Theorem 10. ■

Assumption 37 *Let* $h(\hat{\mathbf{\Theta}}) = \mathbb{E}[\hat{H}(\hat{\mathbf{\Theta}}, X)]$*, then*

$$\exists \hat{\mathbf{\Theta}}_* \text{ s.t.} \sup_{\varepsilon \le |\hat{\mathbf{\Theta}} - \hat{\mathbf{\Theta}}_*| \le \frac{1}{\varepsilon}} (\hat{\mathbf{\Theta}} - \hat{\mathbf{\Theta}}_*)^T h(\hat{\mathbf{\Theta}}) < 0 \text{ for all } 1 > \varepsilon > 0.$$

Assumption 37 is a stability condition and it describes the phenomenon that $\hat{\mathbf{\Theta}}$ would move towards $\hat{\mathbf{\Theta}}_*$.

Assumption 38 $\sum \eta_n = \infty$*,* $\sum \eta_n^2 < \infty$*.*

Following the construction of $\widetilde{H}$ in Benveniste et al. (2012, Sec 5.3.4), we let ϕ be a function mapping from $\mathbb{R}_+$ into $[0, 1]$. ϕ is equal to 1 on $[0, R]$ and 0 on (R, ∞) for some R, and set

$$\widetilde{\hat{H}}\left(\hat{\mathbf{\Theta}}, X\right) = \phi\left(\left|\hat{\mathbf{\Theta}} - \hat{\mathbf{\Theta}}_*\right|\right) \hat{H}\left(\hat{\mathbf{\Theta}}, X\right) - \left(1 - \phi\left(\left|\hat{\mathbf{\Theta}} - \hat{\mathbf{\Theta}}_*\right|\right)\right)\left(\hat{\mathbf{\Theta}} - \hat{\mathbf{\Theta}}_*\right)$$

Theorem 39 *If the Assumptions 34, 37, and 38 hold, then* $\hat{\mathbf{\Theta}}_n$ *converges a.s. to* $\hat{\mathbf{\Theta}}_*$ *on the set* $\{\sup_n |\hat{\mathbf{\Theta}}_n| < \infty\}$*.*

Proof Notice that

$$\widetilde{\hat{H}}\left(\hat{\mathbf{\Theta}}, X\right) = \begin{cases} \hat{H}\left(\hat{\mathbf{\Theta}}, X\right) \text{ if } \left|\hat{\mathbf{\Theta}} - \hat{\mathbf{\Theta}}_*\right| \in [0, R], \\ -\left(\hat{\mathbf{\Theta}} - \hat{\mathbf{\Theta}}_*\right) \text{ if } \left|\hat{\mathbf{\Theta}} - \hat{\mathbf{\Theta}}_*\right| \in (R, \infty), \end{cases} \tag{21}$$

then it follows easily (by Proposition 36) that

$$\widetilde{\sigma}^2(\hat{\mathbf{\Theta}}) = \mathbb{E}\left[|\widetilde{\hat{H}}(\hat{\mathbf{\Theta}}, X)|^2\right] \leq C(1 + |\hat{\mathbf{\Theta}}|^2)$$

for some constant C.

$$\widetilde{h}(\hat{\mathbf{\Theta}}) = \mathbb{E}\left[\widetilde{\hat{H}}(\hat{\mathbf{\Theta}}, X)\right]$$

exists, and

$$\sup_{\varepsilon \leq |\hat{\mathbf{\Theta}} - \hat{\mathbf{\Theta}}_*| \leq \frac{1}{\varepsilon}} (\hat{\mathbf{\Theta}} - \hat{\mathbf{\Theta}}_*)^T \widetilde{h}(\hat{\mathbf{\Theta}}) < 0 \text{ for all } 1 > \varepsilon > 0.$$

Following Benveniste et al. (2012, Section 5.3.4), $\widetilde{\hat{\mathbf{\Theta}}}_n$ converges *a.s.* to $\hat{\mathbf{\Theta}}_*$ for the algorithm

$$\widetilde{\hat{\mathbf{\Theta}}}_{n+1} = \widetilde{\hat{\mathbf{\Theta}}}_n + \eta_{n+1}\widetilde{\hat{H}}\left(\hat{\mathbf{\Theta}}_n, X_{n+1}\right).$$

But for all n, $\widetilde{\hat{\mathbf{\Theta}}}_n = \hat{\mathbf{\Theta}}_n$ on $\{\tau = +\infty\}$, where $\tau = \inf\{n \geq 0 : |\hat{\mathbf{\Theta}}_n - \hat{\mathbf{\Theta}}_*| > R\}$ for arbitrarily large R, which proves the result stated above. ■

7. Prediction

As described in Hong and Shelton (2022, 2023), the prediction is performed by utilizing the samples from the approximate posterior distribution (MCMC or virtual point processes) of the hidden points. The procedure of the prediction is briefly described in Algorithm 5. We adopt the Weibull kernel as in Example 2. The implementation for the inference is a little different than Algorithm 1. We do not optimize the constant intensity function on the top layer by stochastic gradient ascent. Instead, we directly estimate the constant intensity by Monte Carlo expectation maximization (MCEM). In this way, the constant intensity for the top layer is just the expectation of the number of points divided by the length of the interval where NSPs reside. In practice, we found it efficient and stable. We should also notice that we assume different sequences have different constant intensity functions for the top layer. Thus, during the training process, we only learn and fix the parameters for the kernels of the model parameters. For the prediction, we update the constant intensity function for the top layer for each test sequence by MCEM as well.

In terms of the sampling for the next future event, we can extend the intensity function to the future and then sample the next future event. The events for all layers at the interval $[0, t_n]$ are fixed from the posterior sample. The events at the interval (t_n, ∞) for the top layer (layer L) are sampled from the constant intensity function μ_k for each k. Then, the events at the interval (t_n, ∞) for layer $L-1$ are sampled based on Equation 1 determined by the events from the posterior sampling (at the interval $[0, t_n]$) and the forward sampling (at the interval (t_n, ∞)) at layer L. We can keep this sampling process from top to bottom until we reach the evidence. The event at the bottom layer with the smallest time at the interval (t_n, ∞) is the next future event.

Algorithm 5 Prediction with MCMC or approximation

Input: observed data $\mathbf{x} = \{e_1, e_2, \cdots, e_n\}$, where $e_i = (t_i, k_i)$ with t_i representing the time and k_i representing the type, sampler $G(\mathbf{x}, \mathbf{\Theta}^G)$, and model $\mathcal{M}$.

Initialization: the model parameters $\mathbf{\Theta}$ (the kernel parameters $\boldsymbol{\theta}$ are from the training process, and the constant intensity functions $\{\mu_k\}_{k=1}^{K_L}$ are set as the mean of the constant intensity functions for all training sequences), the variational parameters $\tilde{\mathbf{\Theta}}$, the number of iterations for MCEM C (C can be set as 1 in practice), and the sample size $\mathcal{S}$.

Output: time prediction $\hat{t}_{n+1}$, type prediction $\hat{k}_{n+1}$.

```
▷ If we are predicting using MCMC,
    $G(\mathbf{x}, \mathbf{\Theta}^G) = f(\mathbf{z} \mid \mathbf{x}; \mathbf{\Theta})$, where $\mathbf{\Theta}^G = \mathbf{\Theta}$.
▷ If we are predicting using approximation,
    $G(\mathbf{x}, \mathbf{\Theta}^G) = \tilde{f}(\mathbf{z}; \mathbf{x}, \tilde{\mathbf{\Theta}})$, where $\mathbf{\Theta}^G = \tilde{\mathbf{\Theta}}$.

 1: for $c = 1$ to $C$ do
 2:     for $s = 1$ to $\mathcal{S}$ do
 3:         $\underline{\text{sample } \mathbf{z}^s \sim G(\mathbf{x}, \mathbf{\Theta}^G)}$
 4:     end for
 5:     for $k = 1$ to $K_L$ do
 6:         $\mu_k = \frac{1}{\mathcal{S}} \sum_{s=1}^{\mathcal{S}} m_{L,k}^s / T$ ▷ $m_{L,k}^s$ is the number of points within $s$-th sample for $Z_{L,k}$
 7:     end for
 8: end for
 9: for $s = 1$ to $\mathcal{S}$ do
10:     $\underline{\text{sample } \mathbf{z}^s \sim G(\mathbf{x}, \mathbf{\Theta}^G)}$
11:     sample the future time $\hat{t}_{n+1}^s$ based on $\mathbf{z}^s$
12:     sample the future type $\hat{k}_{n+1}^s$ based on $\mathbf{z}^s$
13: end for
14: $\hat{t}_{n+1} = \frac{1}{\mathcal{S}} \sum_{s=1}^{\mathcal{S}} \hat{t}_{n+1}^s$
15: $\hat{k}_{n+1} = \arg\max_{k \in \{1, \ldots, K_0\}} \sum_{s=1}^{\mathcal{S}} \mathbb{1}_k(\hat{k}_{n+1}^s)$
```

Remark 40 *Notice that we do not need to sample until ∞ in practice. The sampling for the intensity function $\sum_{i=1}^{K_{\ell+1}} \sum_{t_{\ell+1,i,j}} \phi_{\boldsymbol{\theta}_{(\ell+1,i)\to(\ell,k)}}(t - t_{\ell+1,i,j})$ as a whole is equivalent to sampling independently for each $\phi_{\boldsymbol{\theta}_{(\ell+1,i)\to(\ell,k)}}(t - t_{\ell+1,i,j})$ and combining the samples at the end. Thus, we can first sample an event at the interval (t_n, ∞) for the evidence with the intensity function determined by a part of the events (the events from the posterior sampling and some events from the forward sampling) and set the time of this sampled event as the largest possible time for the next future event. Suppose the largest possible time is $t_{largest}$, then we only need to sample the events at the interval $(t_n, t_{largest})$ for all layers.*

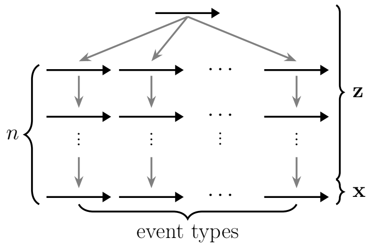
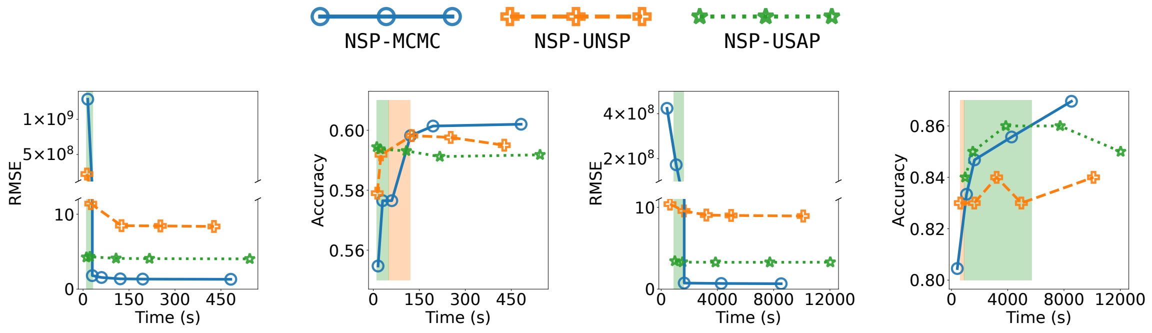


Figure 5: 1-hidden ($n = 1$) and 2-hidden ($n = 2$)



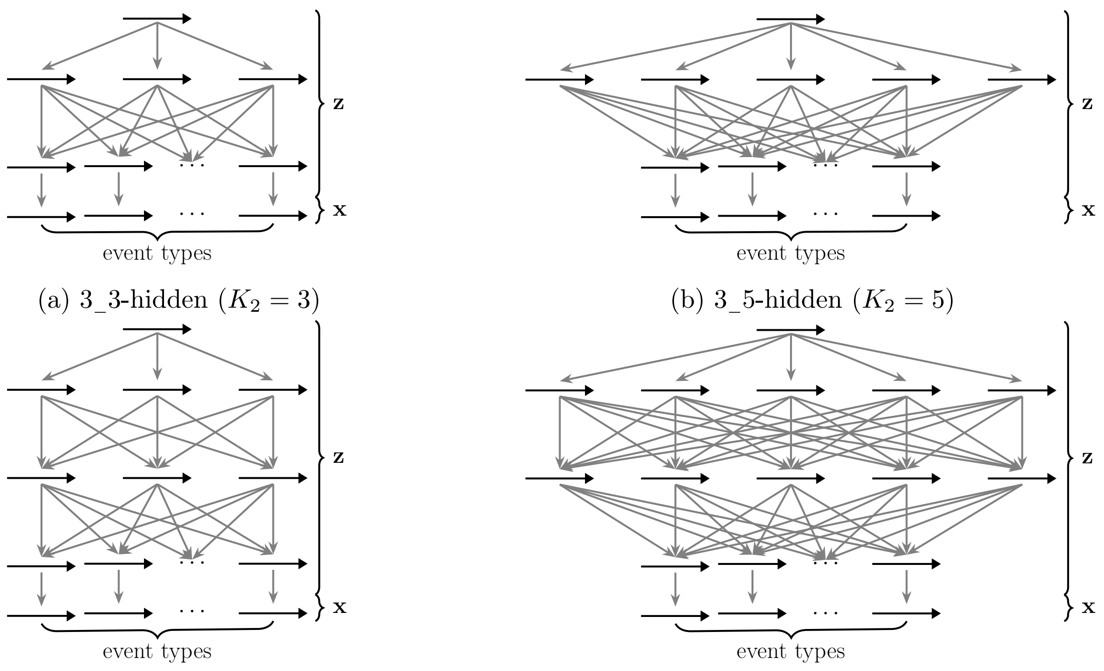
(a) 1-hidden RMSE (b) 1-hidden accuracy (c) 2-hidden RMSE (d) 2-hidden accuracy

Figure 6: Results on the synthetic dataset

8. Experiments

The metrics we use to compare the performance of different models and algorithms are root mean squared error (RMSE) for the time prediction and accuracy for the type prediction. Experiments are performed for both synthetic and real-world datasets. We split each dataset into training, validation, and test sets. The training sets are used to train the model parameters and the variational parameters. The validation sets are used to stop early. The test sets are used to calculate the metrics for performance comparison. Multiple samples are needed to calculate the prediction results in Algorithm 5. More samples typically lead to better results, though consuming more computational time.

Our experiments demonstrate that when the time budget is limited, UNSPs and USAPs perform better than MCMC in terms of time prediction and type prediction. NSPs-based methods are better than non-NSPs-based methods for all the real-world datasets (retweet, earthquake, and homicide) appearing in Section 8.2. We have also constructed NSPs with various numbers of layers and numbers of Poisson processes per layer to compare the prediction performance for the real-world datasets. More hidden layers and hidden Poisson processes can be beneficial in some instances. However, it is not guaranteed that more hidden Poisson processes lead to better predictions as shown in Section 8.2. Theoretically, NSPs with more hidden Poisson processes are more expressive than NSPs with less hidden Poisson processes, which should help improve the performance. However, many factors can prevent the models from reaching their full potential, such as the initialization of the param-



(a) 3_3-hidden ($K_2 = 3$)
(b) 3_5-hidden ($K_2 = 5$)
(c) 4_3-hidden ($K_2 = K_3 = 3$)
(d) 4_5-hidden ($K_2 = K_3 = 5$)

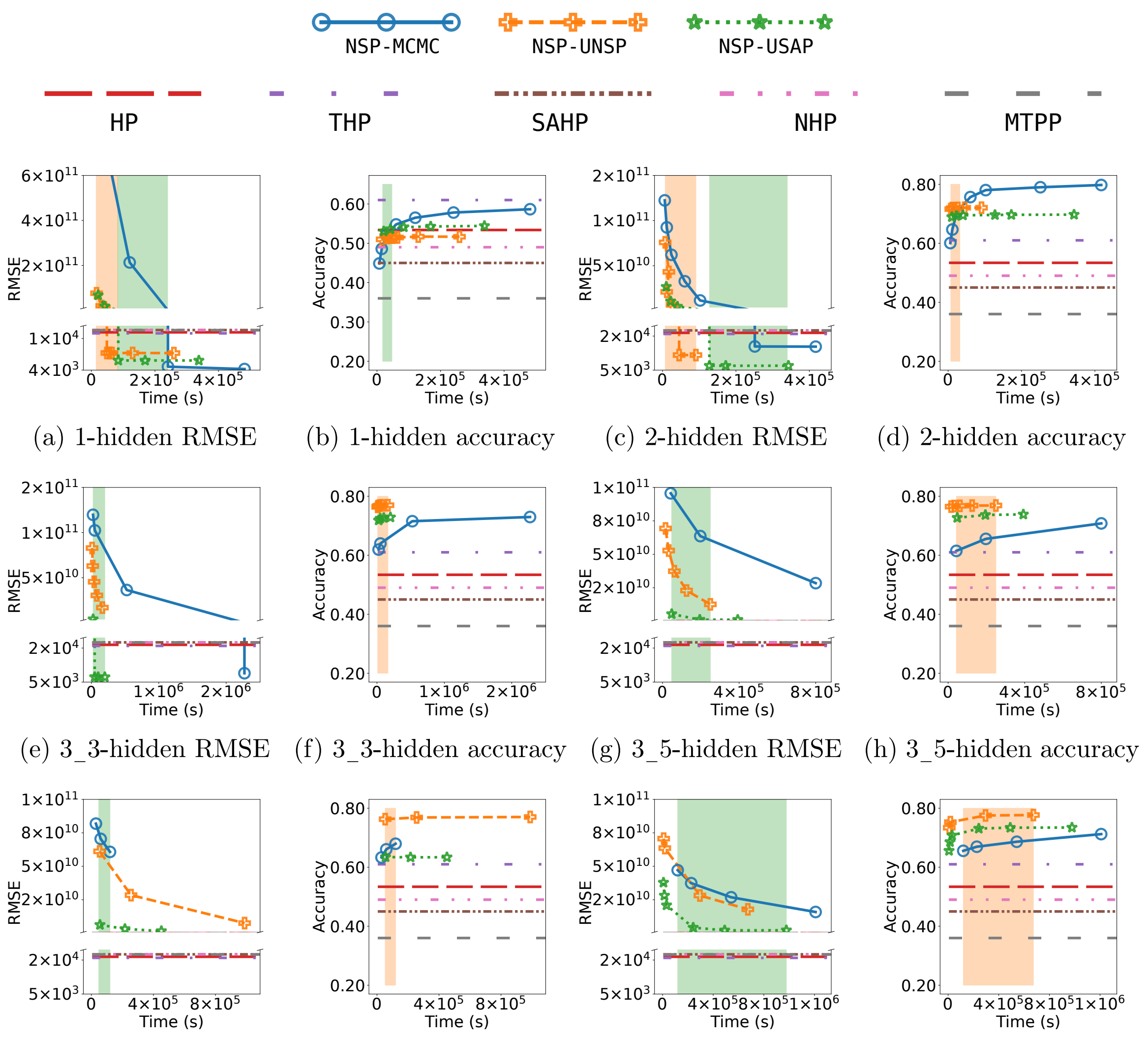
Figure 7: NSPs with more hidden layers and more hidden processes: (a) NSPs with 3 hidden layers with $K_2 = 3$; (b) NSPs with 3 hidden layers with $K_2 = 5$; (c) NSPs with 4 hidden layers with $K_2 = K_3 = 3$; (d) NSPs with 4 hidden layers with $K_2 = K_3 = 5$.

eters, local optima, and the architectures of the models. Regarding how to choose suitable architectures for specific datasets, more research is still needed.

8.1 Synthetic Data Experiments

The two synthetic datasets are generated by 1-hidden and 2-hidden models of NSPs (shown in Figure 5) separately. The NSPs used to generate synthetic data have 2 event types. During synthetic data generation, model parameters are fixed and the data is produced from top to bottom like what we do in Figure 2(a).

Figure 6 summarizes the prediction results for the synthetic datasets. Different algorithms are represented by different line styles with different colors, where NSP-MCMC represents the predictions made by using MCMC for NSPs, NSP-UNSP represents the predictions made by using approximation with UNSPs for NSPs, NSP-USAP represents the predictions made by using approximation with USAPs. The vertical axes indicate RMSE or accuracy. The horizontal axes indicate the time for sampling with different sample sizes for the predictions of the events from all sequences. Different computational time budgets correspond to different numbers of samples used to predict the next future event. Amidst UNSPs, USAPs, and MCMC, USAPs perform the best in the areas filled with light green, and UNSPs perform the best in the areas filled with light orange. It is easy to find that NSP-UNSP and NSP-USAP behave better than NSP-MCMC when the computational time budget is tight.

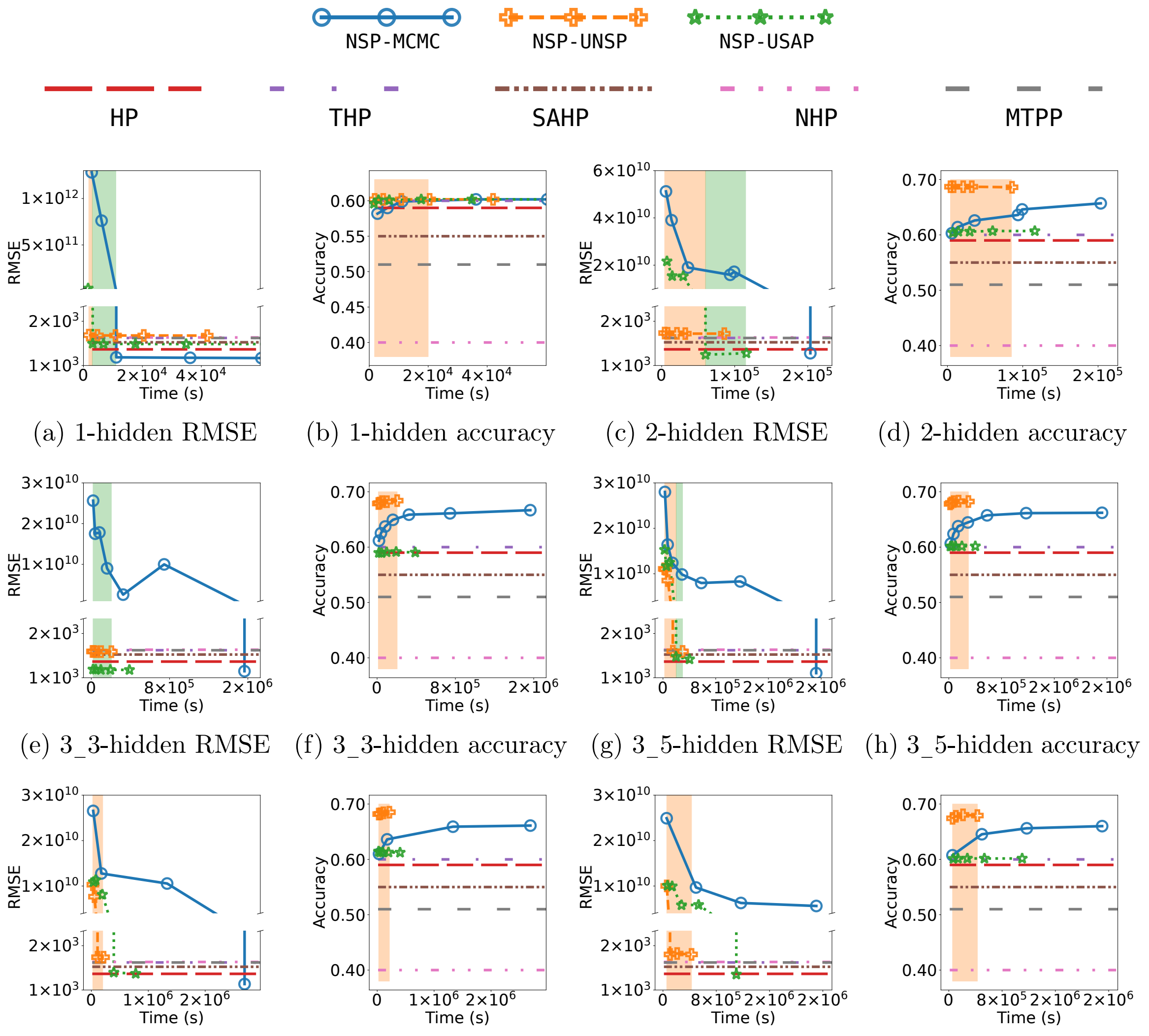


(a) 1-hidden RMSE (b) 1-hidden accuracy (c) 2-hidden RMSE (d) 2-hidden accuracy

(e) 3_3-hidden RMSE (f) 3_3-hidden accuracy (g) 3_5-hidden RMSE (h) 3_5-hidden accuracy

(i) 4_3-hidden RMSE (j) 4_3-hidden accuracy (k) 4_5-hidden RMSE (l) 4_5-hidden accuracy

Figure 8: Results on the retweet dataset

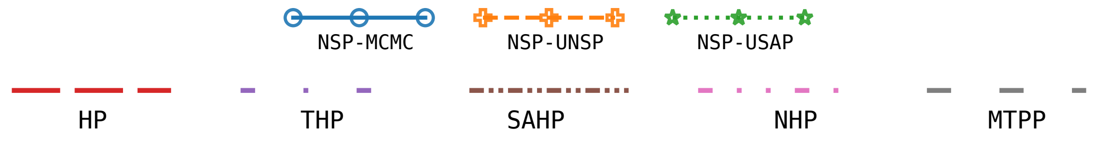


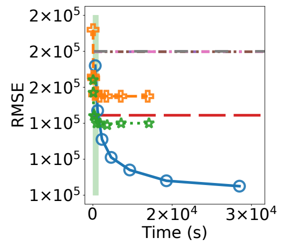
(a) 1-hidden RMSE (b) 1-hidden accuracy (c) 2-hidden RMSE (d) 2-hidden accuracy

(e) 3_3-hidden RMSE (f) 3_3-hidden accuracy (g) 3_5-hidden RMSE (h) 3_5-hidden accuracy

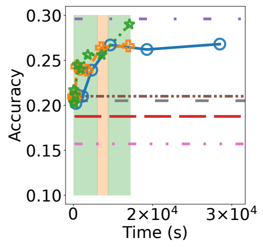
(i) 4_3-hidden RMSE (j) 4_3-hidden accuracy (k) 4_5-hidden RMSE (l) 4_5-hidden accuracy

Figure 9: Results on the earthquake dataset

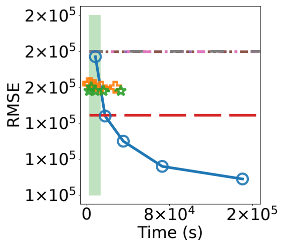




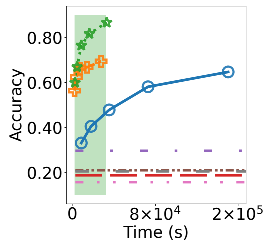
(a) 1-hidden RMSE



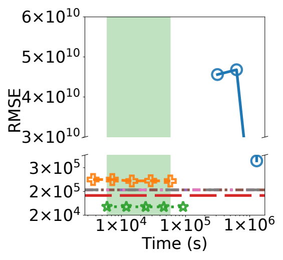
(b) 1-hidden accuracy



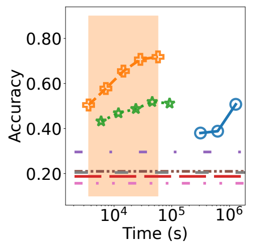
(c) 2-hidden RMSE



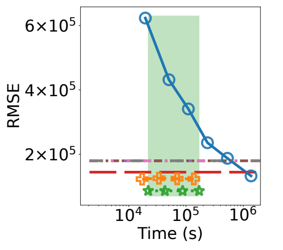
(d) 2-hidden accuracy



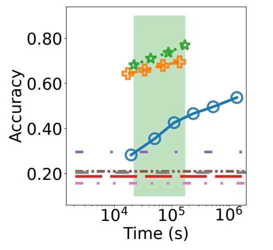
(e) 3_3-hidden RMSE



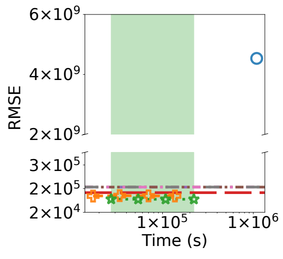
(f) 3_3-hidden accuracy



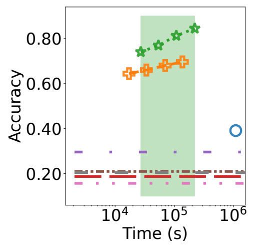
(g) 3_5-hidden RMSE



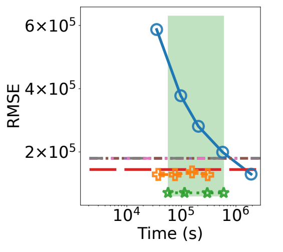
(h) 3_5-hidden accuracy



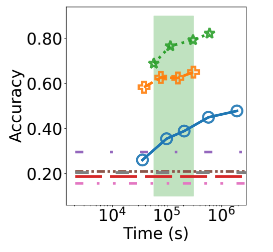
(i) 4_3-hidden RMSE



(j) 4_3-hidden accuracy



(k) 4_5-hidden RMSE



(l) 4_5-hidden accuracy

Figure 10: Results on the homicide dataset

Table 1: RMSE comparison for the retweet dataset

Model	Time(s)				
	6e4	1.2e5	2.4e5	4.8e5	2e6
1-hidden	9.42e+03	7.05e+03	7.04e+03	**4.34e+03**	**4.34e+03**
2-hidden	9.42e+03	9.42e+03	**6.28e+03**	6.29e+03	6.29e+03
3_3-hidden	**6.49e+03**	**6.47e+03**	6.46e+03	6.46e+03	6.46e+03
3_5-hidden	5.01e+09	3.28e+09	1.13e+09	1.10e+09	1.10e+09
4_3-hidden	6.16e+09	5.02e+09	3.02e+09	1.59e+09	1.59e+09
4_5-hidden	1.86e+10	1.39e+10	4.33e+09	2.21e+09	2.18e+09
HP			1.60e+04		
THP			1.56e+04		
SAHP			1.67e+04		
NHP			1.66e+04		
MTPP			1.66e+04		

Table 2: Accuracy comparison for the retweet dataset

Model	Time(s)				
	6e4	1.2e5	2.4e5	4.8e5	2e6
1-hidden	5.48e-01	5.65e-01	5.78e-01	5.87e-01	5.87e-01
2-hidden	7.56e-01	**7.81e-01**	**7.89e-01**	**7.98e-01**	**7.98e-01**
3_3-hidden	**7.69e-01**	7.69e-01	7.70e-01	7.70e-01	7.70e-01
3_5-hidden	7.68e-01	7.68e-01	7.69e-01	7.69e-01	7.69e-01
4_3-hidden	7.63e-01	7.65e-01	7.68e-01	7.69e-01	7.70e-01
4_5-hidden	7.56e-01	7.61e-01	7.71e-01	7.76e-01	7.77e-01
HP			5.34e-01		
THP			6.07e-01		
SAHP			4.55e-01		
NHP			4.89e-01		
MTPP			3.60e-01		

8.2 Real-World Data Experiments

Other than 1-hidden and 2-hidden in Figure 5, we also run the experiments for NSPs with more hidden layers and more hidden Poisson processes as in Figure 7. For the models in Figure 7, the numbers of Poisson processes in layers 0 and 1 are both the number of event types. The layers except for the bottom 2 layers are fully connected. Each Poisson process

Table 3: RMSE comparison for the earthquake dataset

Model	Time(s)				
	1e4	5e4	2.5e5	1.6e6	3.2e6
1-hidden	**1.73e+03**	**1.26e+03**	**1.25e+03**	1.25e+03	1.25e+03
2-hidden	2.09e+03	2.08e+03	1.42e+03	1.42e+03	1.42e+03
3_3-hidden	\	1.28e+03	1.27e+03	**1.22e+03**	1.22e+03
3_5-hidden	\	1.02e+10	1.69e+03	1.63e+03	**1.16e+03**
4_3-hidden	\	8.13e+09	2.11e+03	1.55e+03	1.20e+03
4_5-hidden	\	\	2.23e+03	1.53e+03	1.53e+03
HP			1.55e+03		
THP			1.93e+03		
SAHP			1.79e+03		
NHP			1.95e+03		
MTPP			1.93e+03		

Table 4: Accuracy comparison for the earthquake dataset

Model	Time(s)				
	1e4	5e4	2.5e5	1.6e6	3.2e6
1-hidden	6.02e-01	6.02e-01	6.02e-01	6.02e-01	6.02e-01
2-hidden	**6.87e-01**	**6.87e-01**	**6.86e-01**	**6.86e-01**	**6.86e-01**
3_3-hidden	\	6.82e-01	6.84e-01	6.84e-01	6.84e-01
3_5-hidden	\	6.81e-01	6.83e-01	6.82e-01	6.82e-01
4_3-hidden	\	6.82e-01	6.85e-01	6.85e-01	6.85e-01
4_5-hidden	\	\	6.80e-01	6.79e-01	6.79e-01
HP			5.90e-01		
THP			6.02e-01		
SAHP			5.54e-01		
NHP			4.00e-01		
MTPP			5.14e-01		

in layer 0 is only connected to one corresponding Poisson process in layer 1. For the models in Figures 7(a) and 7(b), the number of Poisson processes in layer 3 is 1, the number of Poisson processes in layer 2 is 3 for 3-hidden ($K_2 = 3$) and 5 for 3-hidden ($K_2 = 5$). For the models in Figures 7(c) and Figure 7(d), the number of Poisson processes in layer 4 is 1, the

Table 5: RMSE comparison for the homicide dataset

Model	Time(s)				
	1e4	2e4	2.4e5	4.8e5	2e6
1-hidden	1.12e+05	1.07e+05	1.05e+05	1.05e+05	1.05e+05
2-hidden	1.58e+05	1.42e+05	1.09e+05	1.09e+05	1.09e+05
3_3-hidden	**7.30e+04**	**7.26e+04**	7.23e+04	7.23e+04	7.23e+04
3_5-hidden	\	1.23e+05	8.63e+04	8.63e+04	8.63e+04
4_3-hidden	\	1.23e+05	1.03e+05	1.03e+05	1.03e+05
4_5-hidden	\	\	**7.04e+04**	**7.04e+04**	**7.05e+04**
HP			1.44e+05		
THP			1.80e+05		
SAHP			1.80e+05		
NHP			1.80e+05		
MTPP			1.80e+05		

Table 6: Accuracy comparison for the homicide dataset

Model	Time(s)				
	1e4	2e4	2.4e5	4.8e5	2e6
1-hidden	2.82e-01	2.90e-01	2.90e-01	2.90e-01	2.90e-01
2-hidden	**7.77e-01**	**8.29e-01**	**8.67e-01**	**8.67e-01**	**8.67e-01**
3_3-hidden	6.07e-01	6.75e-01	7.18e-01	7.18e-01	7.18e-01
3_5-hidden	\	6.47e-01	7.70e-01	7.70e-01	7.70e-01
4_3-hidden	\	6.47e-01	8.45e-01	8.45e-01	8.45e-01
4_5-hidden	\	\	7.85e-01	8.11e-01	8.22e-01
HP			1.88e-01		
THP			2.96e-01		
SAHP			2.10e-01		
NHP			1.57e-01		
MTPP			2.05e-01		

number of Poisson processes in layers 2 and 3 are 3 for 4-hidden ($K_2 = K_3 = 3$) and 5 for 4-hidden ($K_2 = K_3 = 5$).

The real-world datasets we use are retweets (Zhao et al., 2015), earthquakes (NCEDC, 2014; BDSN, 2014; HRSN, 2014; BARD, 2014), and homicides (COC, 2022) as in Hong and Shelton (2022, 2023). We compare NSPs with other state-of-the-art models represented

Table 7: Training time for the retweet dataset

Model	Time(d)		Hardware
	UNSP	USAP	
1-hidden	1.2	5.5	2 cores from an Intel® Xeon® Silver 4214 CPU @ 2.20GHz
			1 NVIDIA® GeForce® RTX 2080 Ti
2-hidden	2.9	6.4	2 cores from an Intel® Xeon® Silver 4214 CPU @ 2.20GHz
			1 NVIDIA® GeForce® RTX 2080 Ti
3_3-hidden	24.5	30.7	1 Intel® Xeon® Gold 6330 CPU @ 2.00GHz shared with other programs
			1 NVIDIA® GeForce® RTX 4090
3_5-hidden	32.0	32.7	1 Intel® Xeon® Gold 6330 CPU @ 2.00GHz shared with other programs
			1 NVIDIA® GeForce® RTX 4090
4_3-hidden	21.8	33.0	1 Intel® Xeon® Gold 6330 CPU @ 2.00GHz shared with other programs
			1 NVIDIA® GeForce® RTX 4090
4_5-hidden	23.4	25.3	1 Intel® Xeon® Gold 6330 CPU @ 2.00GHz shared with other programs
			1 NVIDIA® GeForce® RTX 4090

by various line styles in Figures 8, 9 and 10, where HP is for Hawkes processes(Hawkes, 1971) with an exponential kernel (the parametric form of the kernel is $\phi(t) = \alpha \exp(-\alpha \cdot \beta t)$ with the parameters $\alpha > 0$ and $\beta > 0$), THP is for transformer Hawkes processes (Zuo et al., 2020), SAHP is for self-attentive Hawkes processes (Zhang et al., 2020), NSP is for neural Hawkes processes (Mei and Eisner, 2017), and MTPP is for a multitask point process model (Lian et al., 2015). Similar to the synthetic data experiments, the vertical axes represent RMSE or accuracy. The horizontal axes represent the time for sampling. The results demonstrate that NSPs-based methods are better than non-NSPs-based methods for all these real-world datasets regarding RMSE and accuracy. We also find that the approximations (UNSPs and USAPs) are clearly better than MCMC if there is a limit for the time budget.

Table 8: Training time for the earthquake dataset

Model	Time(d)		Hardware
	UNSP	USAP	
1-hidden	4.0	1.1	2 cores from an Intel® Xeon® Silver 4214 CPU @ 2.20GHz
			1 NVIDIA® GeForce® RTX 2080 Ti
2-hidden	5.2	6.8	2 cores from an Intel® Xeon® Silver 4214 CPU @ 2.20GHz
			1 NVIDIA® GeForce® RTX 2080 Ti
3_3-hidden	12.9	11.8	1 AMD EPYC 7763 64-Core Processor shared with other programs
			1 NVIDIA® GeForce® RTX 3090
3_5-hidden	12.8	14.8	1 AMD EPYC 7763 64-Core Processor shared with other programs
			1 NVIDIA® GeForce® RTX 3090
4_3-hidden	16.3	17.0	1 AMD EPYC 7763 64-Core Processor shared with other programs
			1 NVIDIA® GeForce® RTX 3090
4_5-hidden	20.2	25.9	1 AMD EPYC 7763 64-Core Processor shared with other programs
			1 NVIDIA® GeForce® RTX 3090

Tables 1, 2, 3, 4, 5, and 6 compare the performance of NSPs with different architectures and the baseline models at some fixed time (measured in seconds) points. The experiments are run on different machines with different computational devices. Then, we scale the running time to the running time of the machines with the same computational devices, making sure the running time scales linearly with the sample sizes. The results of each NSPs-based model for each time point reported in these tables are the best predictions among NSP-MCMC, NSP-UNSP, and NSP-USAP, where "\" means there is no result for this entry. The best results among the NSPs with different architectures for each fixed time point are shown in bold text. 2-hidden NSPs achieve the best type prediction results for almost all of the fixed time points. For the time prediction, the effect of adding more hidden Poisson processes is more complicated. For the earthquake and homicide datasets, NSPs with more hidden Poisson processes tend to behave better when we keep increasing the number of samples. For the retweet dataset, more hidden Poisson processes help improve the performance when the number of samples used for prediction is small, while do not boost the performance when we continue to increase the number of samples used for time prediction. One possible reason is that the retweet dataset is much larger than the other two datasets, so the time prediction for the retweet dataset may require many more samples

Table 9: Training time for the homicide dataset

Model	Time(d)		Hardware
	UNSP	USAP	
1-hidden	2.1	3.6	2 cores from an Intel® Xeon® Silver 4214 CPU @ 2.20GHz
			1 NVIDIA® GeForce® RTX 2080 Ti
2-hidden	2.2	1.9	2 cores from an Intel® Xeon® Silver 4214 CPU @ 2.20GHz
			1 NVIDIA® GeForce® RTX 2080 Ti
3_3-hidden	7.7	9.5	1 Intel® Xeon® Silver 4210 CPU @ 2.20GHz shared with other programs
			1 NVIDIA® GeForce® RTX 2080Ti
3_5-hidden	4.1	\	1 Intel® Xeon® Silver 4210 CPU @ 2.20GHz shared with other programs
			1 NVIDIA® GeForce® RTX 2080Ti
3_5-hidden	\	5.9	1 Intel® Xeon® Silver 4214R CPU @ 2.40GHz shared with other programs
			1 NVIDIA® A40
4_3-hidden	8.8	\	1 Intel® Xeon® Silver 4210 CPU @ 2.20GHz shared with other programs
			1 NVIDIA® GeForce® RTX 2080Ti
4_3-hidden	\	23.7	1 Intel® Xeon® Silver 4214R CPU @ 2.40GHz shared with other programs
			1 NVIDIA® A40
4_5-hidden	5.2	13.9	1 Intel® Xeon® Silver 4214R CPU @ 2.40GHz shared with other programs
			1 NVIDIA® A40

to get good results. The time spent on training (measured in days) is shown in Tables 7, 8, and 9.

9. Conclusion

We revisit the classical work of NSPs (Neyman and Scott, 1958) and introduce DNSPs in a more modern formulation. DNSPs are formed by stacking Poisson processes into a deep network. The inherent hierarchical structure allows DNSPs to capture the triggering mechanism of random events. However, the deep and multivariate structure of DNSPs brings huge challenges to the design of posterior sampling and inference algorithms and very few work has touched this problem.

We design virtual-event-based posterior sampling and inference algorithms for NSPs. Our posterior sampling algorithm uses virtual events to help accelerate the mixing, and the distribution of the virtual events is learned through variational inference. Our algorithms combine RJMCMC, marginal likelihood maximization, and variational inference together. The virtual events not only guide the proposal of RJMCMC but also work as approximations for variational inference. We conduct a theoretical analysis of the convergence and provide a sufficient condition for RJMCMC and variational inference to converge. We further extend the analysis to other models, such as MJPs, PCIMs, and Hawkes processes. We have also run experiments to demonstrate the efficacy of NSPs in prediction by using our algorithms. NSPs can predict future events better than baselines and the variational approximation achieves better performance than RJMCMC when the computational time available is limited.

The main limitation of the algorithms proposed in this paper is that the computational cost of the training and prediction is high. During the training process, a large number of samples are needed to be drawn from RJMCMC for each iteration. Further investigation is needed to develop new inference algorithms without RJMCMC involved, such as variational inference through the optimization of exclusive KL divergence. For the prediction algorithm, a large number of samples need to be drawn for the prediction of each future event. To accelerate the prediction, we may consider training a function to map the history to the predicted value directly.

Our work has demonstrated the potential of NSPs in both theory and applications. From the theoretical aspect, there is still much room left for the analysis of the convergence. Our work can only be used to check whether the convergence can be reached. More research is needed to find the convergence rate of our algorithm. With the convergence rate, we may be able to find the bottleneck of our algorithm and accelerate it accordingly. For applications, we can extend our algorithm to spatio-temporal point processes. This extension requires the kernel to be non-causal (*i.e.*, a kernel that is not only positive for $t > 0$ like in Examples 1 and 2, but positive in the whole space like a Gaussian function) and new network architectures may be needed to design a new variational approximation, as the self-attention network used in USAPs cannot be directly applied to a space with 3 or more dimensions.

Acknowledgments

This work was supported by the NSFC Projects (Nos. 62350080, 62106120), Tsinghua Institute for Guo Qiang, and the High Performance Computing Center, Tsinghua University. CH was also supported by Shuimu Tsinghua Scholar Program, and JZ was also supported by the XPlorer Prize.

The data for the earthquakes came from the High Resolution Seismic Network (HRSN), doi:10.7932/HRSN; the Berkeley Digital Seismic Network (BDSN), doi:10/7932/BDSN; and the Bay Area Regional Deformation Network (BARD), doi:10.7932/BARD, all operated by the UC Berkeley Seismological Laboratory and archived at the Northern California Earthquake Data center (NCEDC), doi: 10/7932/NCEDC.

Appendix A. Marginal Density

Lemma 41 *Suppose the parameters for $\lambda_{\ell,k}(\cdot)$ are in a bounded space, then*

$$\lambda_{\ell,k}(t) \leq m_{\ell+1} \cdot a + b,$$

where $a > 0$, $b > 0$ are some constant numbers, and $m_{\ell+1} = \sum_i m_{\ell+1,i}$ is the number of points at layer $\ell + 1$.

Proof It is easy to see from the functional form of the intensity function as in Equation 1. ■

Lemma 42

$$\sum_{n=0}^{\infty} \frac{k^n}{n!}(n+\beta)^{\alpha} = O(e^{Ck}),$$

where the O notation is w.r.t. $k > 0$, and $\alpha \geq 0, \beta \geq 0, C > 1$ are constants.

Proof Let $a_n = \frac{k^n}{n!}(n+\beta)^{\alpha}$, and we can write a_n as $a_n = \frac{(Ck)^n}{n!}\frac{(n+\beta)^{\alpha}}{C^n}$.

Then, it follows that

$$\begin{aligned}
\sum_{n=0}^{\infty} a_n =& \sum_{n=0}^{\infty} \frac{(Ck)^n}{n!}\frac{(n+\beta)^{\alpha}}{C^n} && (22)\\
=& \sum_{n=0}^{N} \frac{(Ck)^n}{n!}\frac{(n+\beta)^{\alpha}}{C^n} + \sum_{n=N+1}^{\infty} \frac{(Ck)^n}{n!}\frac{(n+\beta)^{\alpha}}{C^n} && (23)\\
\leq& e^{Ck} \sum_{n=0}^{N} \frac{(n+\beta)^{\alpha}}{C^n} + \sum_{n=N+1}^{\infty} \frac{(Ck)^n}{n!} && (24)\\
\leq& e^{Ck} \left(\sum_{n=0}^{N} \frac{(n+\beta)^{\alpha}}{C^n} + 1 \right) && (25)\\
=& O(e^{Ck}), && (26)
\end{aligned}$$

where $N > 0$ in Equation 23 is a number *s.t.* $C^n > (n+\beta)^{\alpha}$ for $n \geq N$, the first inequality follows from the fact that $\frac{(Ck)^n}{n!} < \sum_{n=0}^{\infty} \frac{(Ck)^n}{n!} = \exp(Ck)$ and $\frac{(n+\beta)^{\alpha}}{C^n} < 1$, and the last equality follows from the fact that $\left(\sum_{n=0}^{N} \frac{(n+\beta)^{\alpha}}{C^n} + 1\right)$ is a constant number *w.r.t.* k. ■

Proposition 6 *The marginal density $f(\mathbf{x})$ is finite.*

Proof We can write the marginal density as a sequence

$$\begin{aligned} f(\mathbf{x}) =& \mathbb{E}_{\mathbf{z}\sim \text{Poisson(S,1)}}\left[f(\mathbf{x},\mathbf{z})\right] \\ =& \mathbb{E}_{\mathbf{z}\sim \text{Poisson(S,1)}}\left[f(\mathbf{z}_L)\prod_{\ell=1}^{L} f(\mathbf{z}_{\ell-1}\mid \mathbf{z}_\ell)\right] \\ =& \mathbb{E}_{\mathbf{z}_L}\left[\mathbb{E}_{\mathbf{z}_{L-1}}\left[\cdots \mathbb{E}_{\mathbf{z}_1}\left[f(\mathbf{z}_L)\prod_{\ell=1}^{L} f(\mathbf{z}_{\ell-1}\mid \mathbf{z}_\ell)\right]\right]\right] \\ =& \mathbb{E}_{\mathbf{z}_L}\left[f(\mathbf{z}_L)\mathbb{E}_{\mathbf{z}_{L-1}}\left[f(\mathbf{z}_{L-1}\mid \mathbf{z}_L)\cdots \mathbb{E}_{\mathbf{z}_1}\left[f(\mathbf{z}_1\mid \mathbf{z}_2)f(\mathbf{z}_0\mid \mathbf{z}_1)\right]\right]\right], \end{aligned} \tag{27}$$

where $S=[0,T]^{\sum K_\ell}$.

Let

$$q_{\ell-1}(\mathbf{z}_\ell)=\mathbb{E}_{\mathbf{z}_{\ell-1}}\left[f(\mathbf{z}_{\ell-1}\mid \mathbf{z}_\ell)\mathbb{E}_{\mathbf{z}_{\ell-2}}\left[f(\mathbf{z}_{\ell-2}\mid \mathbf{z}_{\ell-1})\cdots \mathbb{E}_{\mathbf{z}_1}\left[f(\mathbf{z}_1\mid \mathbf{z}_2)f(\mathbf{z}_0\mid \mathbf{z}_1)\right]\right]\right], \tag{28}$$

then it follows from Equation 27 that

$$f(\mathbf{x}) = \sum_{m_L=0}^{\infty}\frac{\exp(-K_L\cdot T)}{m_L!}\int\left(\prod_{k=1}^{K_L}\exp(-\mu_k\cdot T)\mu_k^{m_{L,k}}\right)\cdot q_{L-1}(\mathbf{z}_L)d\mathbf{z}_{|m_L} \tag{29}$$

$$\leq \sum_{m_L=0}^{\infty}\frac{\mu^{m_L}}{m_L!}\int q_{L-1}(\mathbf{z}_L)d\mathbf{z}_{|m_L}, \tag{30}$$

where $m_L=\sum_{k=1}^{K_L} m_{L,k}$, $\mu=\max_k\{\mu_k\}$, and $\mathbf{z}_{|m_\ell}$ means $\mathbf{z}$ is restricted to the dimension of m_ℓ.

It also follows that

$$q_{\ell-1}(\mathbf{z}_\ell) = \sum_{m_{\ell-1}=0}^{\infty}\frac{\exp(-K_{\ell-1}\cdot T)}{m_{\ell-1}!}\int\left(\prod_{k=1}^{K_{\ell-1}} f(\mathbf{z}_{\ell-1,k}\mid \mathbf{z}_\ell)\right)\cdot q_{\ell-2}(\mathbf{z}_{\ell-1})d\mathbf{z}_{|m_{\ell-1}} \tag{31}$$

$$\leq \sum_{m_{\ell-1}=0}^{\infty}\frac{(a\cdot m_\ell+b)^{m_{\ell-1}}}{m_{\ell-1}!}\int q_{\ell-2}(\mathbf{z}_{\ell-1})d\mathbf{z}_{|m_{\ell-1}}, \tag{32}$$

where $f(\mathbf{z}_{\ell-1,k}\mid \mathbf{z}_\ell)=\exp\left(-\int \lambda_{\ell-1,k}(t;\mathbf{z}_\ell)dt\right)\prod_{i=1}^{m_{\ell-1,k}}\lambda_{\ell-1,k}(t_{\ell-1,k,i};\mathbf{z}_\ell)$.

According to Lemma 41,

$$f(\mathbf{z}_0\mid \mathbf{z}_1)\leq (a\cdot m_1+b)^{m_0}, \tag{33}$$

where m_0 is the number of points in the observation.

Then,

$$f(\mathbf{x})\leq \sum_{m_L=0}^{\infty}\frac{\mu^{n_L}}{m_L!}\left(\sum_{m_{L-1}=0}^{\infty}\frac{(a\cdot m_L+b)^{m_{L-1}}}{m_{L-1}!}\left(\cdots\sum_{m_1=0}^{\infty}\frac{(a\cdot m_2+b)^{m_1}}{m_1!}(a\cdot m_1+b)^{m_0}\right)\right). \tag{34}$$

Let

$$\begin{aligned} h_{\ell-1} = &\sum_{m_{\ell-1}=0}^{\infty} \frac{(a \cdot m_\ell + b)^{m_{\ell-1}}}{m_{\ell-1}!} \cdot \\ &\left(\sum_{m_{\ell-2}=0}^{\infty} \frac{(a \cdot m_{\ell-1} + b)^{m_{\ell-2}}}{m_{\ell-2}!} \left(\cdots \sum_{m_1=0}^{\infty} \frac{(a \cdot m_2 + b)^{m_1}}{m_1!} (a \cdot m_1 + b)^{m_0} \right) \right), \end{aligned} \tag{35}$$

and we want to prove

$$h_{\ell-1} = O(e^{c \cdot m_\ell}), \text{ where } O \text{ is } \textit{w.r.t. } m_\ell, \text{ and } c > 0 \text{ is a constant number} \tag{36}$$

by induction.

When $\ell = 2$,

$$\begin{aligned} h_1 = &\sum_{m_1=0}^{\infty} \frac{(a \cdot m_2 + b)^{m_1}}{m_1!} (a \cdot m_1 + b)^{m_0} \\ =&a^{m_0} \sum_{m_1=0}^{\infty} \frac{(a \cdot m_2 + b)^{m_1}}{m_1!} (m_1 + \frac{b}{a})^{m_0} \\ =&a^{m_0} O(e^{c_1 \cdot (a \cdot m_2 + b)}) \\ =&O(e^{c_1 \cdot a \cdot m_2}), \end{aligned}$$

where $c_1 > 0$ is a constant number and the third equality follows from Lemma 42. It is easy to see Equation 36 holds for $\ell = 2$.

Suppose Equation 36 holds for $\ell = n$, then

$$\begin{aligned} h_{n+1} = &\sum_{m_{n+1}=0}^{\infty} \frac{(a \cdot m_{n+2} + b)^{m_{n+1}}}{m_{n+1}!} h_n \\ \leq &M \cdot \sum_{m_{n+1}=0}^{\infty} \frac{(a \cdot m_{n+2} + b)^{m_{n+1}}}{m_{n+1}!} \left(e^{c \cdot m_{n+1}}\right) \text{ for a constant number } M > 0 \\ \leq &M \cdot \sum_{m_{n+1}=0}^{\infty} \frac{((a \cdot m_{n+2} + b) \cdot e^c)^{m_{n+1}}}{m_{n+1}!} \\ \leq &M \cdot e^{(a \cdot m_{n+2} + b) \cdot e^c} \\ = &O(e^{c' \cdot m_{n+2}}), \end{aligned} \tag{37}$$

where $c' = a \cdot e^c$ is a constant number *w.r.t.* m_{n+2}.

It follows from derivation of Equation 37 that

$$h_{L-1} = O(e^{c \cdot m_L}),$$

where the O is *w.r.t.* m_L and $c > 0$ is a constant number.

Thus,

$$
\begin{aligned}
f(\mathbf{x}) \leq & M' \cdot \sum_{m_L=0}^{\infty} \frac{\mu^{m_L}}{m_L!} e^{c \cdot m_L} \text{ for a constant number } M' > 0 \\
= & M' \cdot e^{\mu e^c},
\end{aligned}
$$

which shows that the marginal density $f(\mathbf{x})$ exists and is finite. ■

Appendix B. Density of DNSPs as the Limiting Value of DEFs

Lemma 43 *For a realization $(\mathbf{x}, \mathbf{z}_{1:L})$, the joint density of DEFs $p(\mathbf{x}, \mathbf{z}_{1:L})$ converges to the joint density of DNSPs $f(\mathbf{x}, \mathbf{z}_{1:L})$ when $\delta_L \to 0$.*

Proof For the realization $(\mathbf{x}, \mathbf{z}_{1:L})$, we can build a corresponding DEF such that $G_{\ell,i,j} = 1$ if there is an event in $\hat{T}_{\ell,i,j}$ and $G_{\ell,i,j} = 0$ if there is no event in $\hat{T}_{\ell,i,j}$, where $G_{\ell,i,j}$ represents a hidden variable in DEFs as explained in Section 2.

$$
\begin{aligned}
\lim_{\delta_L \to 0} p(\mathbf{x}, \mathbf{z}_{1:L}) &= \lim_{\delta_L \to 0} \left(\left(\prod_{G_{\ell,i,j}=0} (1 - \lambda^G_{\ell,i}(\xi_{\ell,i,j})\delta_\ell) \right) \left(\prod_{G_{\ell,i,j}=1} \frac{\lambda^G_{\ell,i}(\xi_{\ell,i,j})\delta_\ell}{\delta_\ell} \right) \right) && (38) \\
&= \lim_{\delta_L \to 0} \left(\left(\prod_{G_{\ell,i,j}=0} \exp\left(-\lambda^G_{\ell,i}(\xi_{\ell,i,j}) \cdot \delta_\ell + o(\delta_\ell)\right) \right) \left(\prod_{G_{\ell,i,j}=1} \lambda^G_{\ell,i}(\xi_{\ell,i,j}) \right) \right) && (39) \\
&= \exp\left(-\sum_{\ell,i} \int_0^T \lambda_{\ell,i}(t)dt \right) \prod_{t_{\ell,i,j}} \lambda_{\ell,i}(t_{\ell,i,j}) && (40) \\
&= f(\mathbf{x}, \mathbf{z}_{1:L}) && (41)
\end{aligned}
$$

■

Lemma 44 *For a realization $(\mathbf{x}, \mathbf{z}_{1:L})$, the conditional density of DEFs $p(\mathbf{x} \mid \mathbf{z}_{1:L})$ converges to the conditional density of DNSPs $f(\mathbf{x} \mid \mathbf{z}_{1:L})$ when $\delta_L \to 0$.*

Proof It can be proved similarly to Lemma 43. ■

Lemma 45 *For a realization $(\mathbf{x}, \mathbf{z}_{1:L})$, the marginal density of DEFs $p(\mathbf{x})$ converges to the marginal density of DEFs $f(\mathbf{x})$ when $\delta_L \to 0$.*

Proof Similar to the proof of Lemma 43, we can build a corresponding DEF such that

$$
p(\mathbf{x}) = \mathbb{E}_{\mathbf{g}_L}\left[p(\mathbf{g}_L)\mathbb{E}_{\mathbf{g}_{L-1}}\left[p(\mathbf{g}_{L-1} \mid \mathbf{g}_L) \cdots \mathbb{E}_{\mathbf{g}_1}\left[p(\mathbf{g}_1 \mid \mathbf{g}_2) \cdot p(\mathbf{x} \mid \mathbf{g}_1)\right]\right]\right],
$$

where $\mathbf{g}_\ell$ is a realization for $\mathbf{G}_\ell = \left\{ \{G_{\ell,i,j}\}_{i=1}^{K_\ell} \right\}_{j=1}^{\hat{m}_{\ell,i}}$.

Let

$$p_{\ell-1}(\mathbf{g}_\ell) = \mathbb{E}_{\mathbf{g}_{\ell-1}} \left[p(\mathbf{g}_{\ell-1} \mid \mathbf{g}_\ell) \mathbb{E}_{\mathbf{g}_{\ell-2}} \left[p(\mathbf{g}_{\ell-2} \mid \mathbf{g}_{\ell-1}) \cdots \mathbb{E}_{\mathbf{g}_1} \left[p(\mathbf{g}_1 \mid \mathbf{g}_2) p(\mathbf{x} \mid \mathbf{g}_1) \right] \right] \right],$$

we first prove

$$\lim_{\delta_L \to 0} p_1(\mathbf{g}_2) = q_1(\mathbf{z}_2),$$

where $q_{\ell-1}(\mathbf{z}_\ell)$ is defined in Equation 28 and $\mathbf{g}_2$ is the realization of DEF corresponding to $\mathbf{z}_2$ as in Lemma 43.

Notice that when $\ell = L+1$, we let $\mathbf{g}_{L+1} = \mathbf{z}_{L+1} = \emptyset$, $p(\mathbf{g}_L \mid \mathbf{g}_{L+1}) = p(\mathbf{g}_L)$ and $f(\mathbf{z}_L \mid \mathbf{z}_{L+1}) = f(\mathbf{z}_L)$.

$$\begin{aligned}
p_1(\mathbf{g}_2) =& \mathbb{E}_{\mathbf{g}_1} \left[p(\mathbf{g}_1 \mid \mathbf{g}_2) p(\mathbf{x} \mid \mathbf{g}_1) \right] && (42)\\
=& \sum_{\mathbf{g}_1} p(\mathbf{g}_1 \mid \mathbf{g}_2) p(\mathbf{g}_1 \mid \mathbf{g}_2) p(\mathbf{x} \mid \mathbf{g}_1) && (43)\\
=& \sum_{n=0}^{K_1 \cdot T/\delta_1} \sum_{\mathbf{g}_1 = 1 \mid n} p(\mathbf{g}_1 \mid \mathbf{g}_2) p(\mathbf{g}_1 \mid \mathbf{g}_2) p(\mathbf{x} \mid \mathbf{g}_1) && (44)\\
=& \sum_{n=0}^{K_1 \cdot T/\delta_1} \sum_{\mathbf{g}_1 = 1 \mid n} \left[\left[\exp\left(- \sum_{\mathbf{g}_{1,i,j}=0} \lambda_{1,i}^G(\xi_{1,i,j}) + o(\delta_1) \right) \prod_{\mathbf{g}_{1,i,j}=1} \lambda_{1,i}^G(\xi_{1,i,j}) \right]^2 \cdot \right. && (45)\\
& \left. \left[\exp\left(- \sum_{\mathbf{g}_{0,i,j}=0} \lambda_{0,i}^G(\xi_{0,i,j}) + o(\delta_0) \right) \prod_{\mathbf{g}_{0,i,j}=1} \lambda_{0,i}^G(\xi_{0,i,j}) \right] \right], && (46)
\end{aligned}$$

where, with $i=1$, $\sum_{\mathbf{g}_i}$ represents the sum *w.r.t.* different realizations for $\mathbf{G}_i$, n represents the number of random variables of $\mathbf{G}_i$ equal to 1, and $\mathbf{g}_i = 1 \mid n$ represents the different combinations of $\mathbf{g}_i$ with n random variables equal to 1.

It is easy to find that

$$\begin{aligned}
& \lim_{\delta_L \to 0} \sum_{n=0}^{K_1 \cdot T/\delta_1} \sum_{\mathbf{g}_1 = 1 \mid n} \left[\left[\exp\left(- \sum_{\mathbf{g}_{1,i,j}=0} \lambda_{1,i}^G(\xi_{1,i,j}) + o(\delta_1) \right) \prod_{\mathbf{g}_{1,i,j}=1} \lambda_{1,i}^G(\xi_{1,i,j}) \right]^2 \cdot \right. && (47)\\
& \left. \left[\exp\left(- \sum_{\mathbf{g}_{0,i,j}=0} \lambda_{0,i}^G(\xi_{0,i,j}) + o(\delta_0) \right) \prod_{\mathbf{g}_{0,i,j}=1} \lambda_{0,i}^G(\xi_{0,i,j}) \right] \right] && (48)\\
=& \sum_{n=0}^{K_1 \cdot T/\delta_1} \sum_{\mathbf{z}_1 = 1 \mid n} \left[\left[\exp\left(- \int_0^T \lambda_{1,i}(t) dt \right) \prod_{t_{1,i,j}} \lambda_{1,i}(t_{1,i,j}) \right]^2 \cdot \right. && (49)\\
& \left. \left[\exp\left(- \int_0^T \lambda_{0,i}(t_{0,i,j}) \right) \prod_{t_{0,i,j}} \lambda_{0,i}(t_{0,i,j}) \right] \right], && (50)
\end{aligned}$$

which implies that $\forall \epsilon' > 0$, $\exists \delta' > 0$, such that for all $\delta_1 < \delta'$,

$$\left| \sum_{n=0}^{K_1 \cdot T/\delta_1} \sum_{\mathbf{g}_1 = 1|n} \left[\left[\exp\left(- \sum_{\mathbf{g}_{1,i,j}=0} \lambda_{1,i}^G(\xi_{1,i,j}) + o(\delta_1) \right) \prod_{\mathbf{g}_{1,i,j}=1} \lambda_{1,i}^G(\xi_{1,i,j}) \right]^2 \cdot \right. \right. \tag{51}$$

$$\left. \left[\exp\left(- \sum_{\mathbf{g}_{0,i,j}=0} \lambda_{0,i}^G(\xi_{0,i,j}) + o(\delta_0) \right) \prod_{\mathbf{g}_{0,i,j}=1} \lambda_{0,i}^G(\xi_{0,i,j}) \right] \right] - \tag{52}$$

$$\sum_{n=0}^{K_1 \cdot T/\delta_1} \sum_{\mathbf{z}_1 = 1|n} \left[\left[\exp\left(- \int_0^T \lambda_{1,i}(t) dt \right) \prod_{t_{1,i,j}} \lambda_{1,i}(t_{1,i,j}) \right]^2 \cdot \right. \tag{53}$$

$$\left. \left. \left[\exp\left(- \int_0^T \lambda_{0,i}(t_{0,i,j}) \right) \prod_{t_{0,i,j}} \lambda_{0,i}(t_{0,i,j}) \right] \right] \right| < \epsilon'. \tag{54}$$

We can also find that $\forall \epsilon'' > 0$, $\exists \delta'' > 0$, such that for all $\delta_1 < \delta''$,

$$\sum_{n=K_1 \cdot T/\delta_1}^{\infty} \sum_{\mathbf{z}_1 = 1|n} \left[\left[\exp\left(- \int_0^T \lambda_{1,i}(t) dt \right) \prod_{t_{1,i,j}} \lambda_{1,i}(t_{1,i,j}) \right]^2 \cdot \right. \tag{55}$$

$$\left. \left[\exp\left(- \int_0^T \lambda_{0,i}(t_{0,i,j}) \right) \prod_{t_{0,i,j}} \lambda_{0,i}(t_{0,i,j}) \right] \right] < \epsilon'', \tag{56}$$

since the density of the corresponding DNSP is finite.

Hence, for all $\delta_1 < \min\{\delta', \delta''\}$,

$$\left| \sum_{n=0}^{K_1 \cdot T/\delta_1} \sum_{\mathbf{g}_1 = 1|n} \left[\left[\exp\left(- \sum_{\mathbf{g}_{1,i,j}=0} \lambda_{1,i}^G(\xi_{1,i,j}) + o(\delta_1) \right) \prod_{\mathbf{g}_{1,i,j}=1} \lambda_{1,i}^G(\xi_{1,i,j}) \right]^2 \cdot \right. \right. \tag{57}$$

$$\left. \left[\exp\left(- \sum_{\mathbf{g}_{0,i,j}=0} \lambda_{0,i}^G(\xi_{0,i,j}) + o(\delta_0) \right) \prod_{\mathbf{g}_{0,i,j}=1} \lambda_{0,i}^G(\xi_{0,i,j}) \right] \right] - \tag{58}$$

$$\sum_{n=0}^{\infty} \sum_{\mathbf{z}_1 = 1|n} \left[\left[\exp\left(- \int_0^T \lambda_{1,i}(t) dt \right) \prod_{t_{1,i,j}} \lambda_{1,i}(t_{1,i,j}) \right]^2 \cdot \right. \tag{59}$$

$$\left. \left. \left[\exp\left(- \int_0^T \lambda_{0,i}(t_{0,i,j}) \right) \prod_{t_{0,i,j}} \lambda_{0,i}(t_{0,i,j}) \right] \right] \right| < \epsilon' + \epsilon'', \tag{60}$$

which implies $\lim_{\delta_L \to 0} p_1(\mathbf{g}_2) = q_1(\mathbf{z}_2)$.

Next, we assume

$$\lim_{\delta_L \to 0} p_{\ell-1}(\mathbf{g}_\ell) = q_{\ell-1}(\mathbf{z}_\ell)$$

holds for $\ell = i$. Then we need to prove it also holds for $\ell = i+1$.

When $\ell = i+1$,

$$\begin{aligned} p_{\ell-1}(\mathbf{g}_\ell) =& \mathbb{E}_{\mathbf{g}_i}\left[p(\mathbf{g}_i \mid \mathbf{g}_{i+1}) p_{i-1}(\mathbf{g}_i)\right] && (61)\\ =& \sum_{\mathbf{g}_i} p(\mathbf{g}_i \mid \mathbf{g}_{i+1}) p(\mathbf{g}_i \mid \mathbf{g}_{i+1}) p_{i-1}(\mathbf{g}_i) && (62)\\ =& \sum_{n=0}^{K_i \cdot T/\delta_i} \sum_{\mathbf{g}_i|n} p^2(\mathbf{g}_i \mid \mathbf{g}_{i+1}) p_{i-1}(\mathbf{g}_i) && (63)\\ =& \sum_{n=0}^{K_i \cdot T/\delta_i} \sum_{\mathbf{g}_i|n} \left(\exp\left(-\sum_{\mathbf{g}_{i,s,k}=0} \lambda^G_{i,s}(\xi_{i,s,k}) + o(\delta_i)\right) \prod_{\mathbf{g}_{i,s,k}=1} \lambda^G_{i,s}(\xi_{i,s,k})\right)^2 p_{i-1}(\mathbf{g}_i). && (64) \end{aligned}$$

Notice that when $i = L$, Equation 61 becomes $p_{\ell-1}(\mathbf{g}_\ell) = \mathbb{E}_{\mathbf{g}_L}\left[p(\mathbf{g}_L) p_{L-1}(\mathbf{g}_L)\right]$ and the following arguments still hold.

Similar to the argument of proving $\lim_{\delta_L \to 0} p_1(\mathbf{g}_2) = q_1(\mathbf{z}_2)$, $\forall \epsilon''' > 0$, we can find $\delta''' > 0$, such that for all $\delta_i < \delta'''$,

$$\left| \sum_{n=0}^{K_i \cdot T/\delta_i} \sum_{\mathbf{g}_i=1|n} \left[\left[\exp\left(-\sum_{\mathbf{g}_{i,s,k}=0} \lambda^G_{i,s}(\xi_{i,s,k}) + o(\delta_i)\right) \prod_{\mathbf{g}_{i,s,k}=1} \lambda^G_{i,s}(\xi_{i,s,k})\right]^2 \cdot p_{i-1}(\mathbf{g}_i) \right] - \right. \tag{65}$$

$$\left. \sum_{n=0}^{K_1 \cdot T/\delta_1} \sum_{\mathbf{z}_i=1|n} \left[\left[\exp\left(-\int_0^T \lambda_{i,s}(t)dt\right) \prod_{t_{i,s,k}} \lambda_{i,s}(t_{i,s,k})\right]^2 \cdot q_{i-1}(\mathbf{z}_i) \right] \right| < \epsilon'''. \tag{66}$$

It also follows that $\forall \epsilon'''' > 0$, $\exists \delta'''' > 0$, such that for all $\delta_i < \delta''''$,

$$\sum_{n=K_i \cdot T/\delta_i}^{\infty} \sum_{\mathbf{z}_i=1|n} \left[\left[\exp\left(-\int_0^T \lambda_{i,s}(t)dt\right) \prod_{t_{i,s,k}} \lambda_{i,s}(t_{i,s,k})\right]^2 \cdot q_{i-1}(\mathbf{z}_i) \right] < \epsilon''''. \tag{67}$$

Thus,

$$\lim_{\delta_L \to 0} p_{\ell-1}(\mathbf{g}_\ell) = q_{\ell-1}(\mathbf{z}_\ell)$$

holds for $\ell = i+1$ and for all $\ell \le L+1$, which implies that

$$\lim_{\delta_L \to 0} p(\mathbf{x}) = f(\mathbf{x}).$$

■

Theorem 7 *For a realization $(\mathbf{x}, \mathbf{z}_{1:L})$, the joint density of DEFs $p(\mathbf{x}, \mathbf{z}_{1:L})$ converges to the joint density of DNSPs $f(\mathbf{x}, \mathbf{z}_{1:L})$, the conditional density of DEFs $p(\mathbf{x} \mid \mathbf{z}_{1:L})$ converges to the conditional density of DNSPs $f(\mathbf{x} \mid \mathbf{z}_{1:L})$, and the marginal density of DEFs $p(\mathbf{x})$ converges to the marginal density of DEFs $f(\mathbf{x})$ when $\delta_L \to 0$.*

Proof It follows directly from Lemmas 43, 44, and 45. ■

Appendix C. Interchange of the Expectation and the Gradient

The following theorem tells us under which conditions a sequence of functions would uniformly converge.

Theorem 46 (Rudin et al. (1976, Theorem 7.10)) *Suppose $\{f_n\}$ is a sequence of functions defined on E, and suppose*

$$|f_n(x)| \leq M_n \ (x \in E, \ n = 1, 2, 3, \cdots).$$

Then $\sum f_n$ converges uniformly on E if $\sum M_n$ converges.

The uniform convergence of a sequence of functions provides a nice property for interchanging the derivative and the limit of a sequence of functions, which can be described as follows.

Theorem 47 (Rudin et al. (1976, Theorem 7.17)) *Suppose $\{f_n\}$ is a sequence of functions, differentiable on $[a, b]$ and such that $\{f_n(x_0)\}$ converges for some point x_0 on $[a, b]$. If $\{f'_n\}$ converges uniformly on $[a, b]$, then $\{f_n\}$ converges uniformly on $[a, b]$, to a function f, and*

$$f'(x) = \lim_{n\to\infty} f'_n(x) \ (a \leq x \leq b).$$

The following corollary follows easily from Theorem 47[1]:

Corollary 48 *Suppose $\{f_n\}$ is a sequence of functions, differentiable on a convex set $\mathcal{C} \subseteq \mathcal{X} \subseteq \mathbb{R}^n$ and such that $\{f_n\}$ converges pointwise to a function f on $\mathcal{C}$. If $\{\nabla f_n\}$ converges uniformly on $\mathcal{C}$, then $\{f_n\}$ converges uniformly on $\mathcal{C}$, to a function f, and*

$$\nabla f = \lim_{n\to\infty} \nabla f_n.$$

C.1 Unbiased Estimate of the Gradient of the Marginal Likelihood

We can write the gradient of the marginal likelihood as

$$\begin{aligned}\nabla_{\boldsymbol{\Theta}} \log f(\mathbf{x}; \boldsymbol{\Theta}) &= \frac{\nabla_{\boldsymbol{\Theta}} f(\mathbf{x})}{f(\mathbf{x})} \\ &= \frac{\nabla_{\boldsymbol{\Theta}} \sum_{n=0}^{\infty} \frac{\exp(-|S|)}{n!} \int f(\mathbf{x}, \mathbf{z}) d\mathbf{z}_{|n}}{f(\mathbf{x})},\end{aligned} \tag{68}$$

1. `https://math.stackexchange.com/questions/2511866/multivariable-uniform-convergence-and-differentiation`

where $S = [0,T]^{\sum_\ell K_\ell}$, $\mathbf{z}_{|n}$ indicates that $\mathbf{z}$ is restricted in a space with dimension of n.

The expectation of the gradient of the joint log of density can be written as

$$\begin{aligned}
\mathbb{E}_{\mathbf{z}\sim f(\mathbf{Z}|\mathbf{x})}\left[\nabla_{\mathbf{\Theta}} \log f(\mathbf{x},\mathbf{z})\right] =& \sum_{n=0}^{\infty} \frac{\exp(-|S|)}{n!} \int f(\mathbf{z} \mid \mathbf{x}) \nabla_{\mathbf{\Theta}} \log f(\mathbf{x},\mathbf{z}) d\mathbf{z}_{|n} \\
=& \sum_{n=0}^{\infty} \frac{\exp(-|S|)}{n!} \int \frac{f(\mathbf{z},\mathbf{x})}{f(\mathbf{x})} \frac{\nabla_{\mathbf{\Theta}} f(\mathbf{x},\mathbf{z})}{f(\mathbf{x},\mathbf{z})} d\mathbf{z}_{|n} \\
=& \sum_{n=0}^{\infty} \frac{\exp(-|S|)}{n!} \int \frac{\nabla_{\mathbf{\Theta}} f(\mathbf{x},\mathbf{z})}{f(\mathbf{x})} d\mathbf{z}_{|n}, \qquad (69)
\end{aligned}$$

where $S = [0,T]^{\sum_\ell K_\ell}$.

It is easy to see that $\int \nabla_{\mathbf{\Theta}} f(\mathbf{x},\mathbf{z}) d\mathbf{z}_{|n} = \nabla_{\mathbf{\Theta}} \int f(\mathbf{x},\mathbf{z}) d\mathbf{z}_{|n}$, where the equality follows from the dominated convergence theorem. Because $f(\mathbf{x},\mathbf{z})$ is integrable *w.r.t.* $\mathbf{z}$ and differentiable *w.r.t.* $\mathbf{\Theta}$, and $\nabla_{\mathbf{\Theta}} f(\mathbf{x},\mathbf{z})$ is bounded.

Let

$$w_m(\mathbf{\Theta}) = \sum_{n=0}^{m} \frac{\exp(-|S|)}{n!} \int f(\mathbf{x},\mathbf{z}) d\mathbf{z}_{|n},$$

it follows that

$$\nabla_{\mathbf{\Theta}} w_m(\mathbf{\Theta}) = \nabla_{\mathbf{\Theta}} \sum_{n=0}^{m} \frac{\exp(-|S|)}{n!} \int f(\mathbf{x},\mathbf{z}) d\mathbf{z}_{|n} = \sum_{n=0}^{m} \frac{\exp(-|S|)}{n!} \int \nabla_{\mathbf{\Theta}} f(\mathbf{x},\mathbf{z}) d\mathbf{z}_{|n}.$$

We want to show that $\nabla_{\mathbf{\Theta}} w_m(\mathbf{\Theta})$ converges uniformly, then we can use Theorem 47 to demonstrate that Equation 68 is equal to Equation 69.

The gradient of the joint density is

$$\begin{aligned}
\nabla_{\mathbf{\Theta}} f(\mathbf{x},\mathbf{z}) =& \nabla_{\mathbf{\Theta}} \left(\prod_{\ell=1}^{L} f(\mathbf{z}_{\ell-1} \mid \mathbf{z}_\ell) \right) \\
=& \sum_{i=1}^{L} \left(\nabla_{\mathbf{\Theta}} f(\mathbf{z}_{i-1} \mid \mathbf{z}_i) \left(\prod_{j=1, j\neq i}^{L} f(\mathbf{z}_{j-1} \mid \mathbf{z}_j) \right) \right).
\end{aligned}$$

and,

$$\begin{aligned}
\nabla_{\mathbf{\Theta}} f(\mathbf{z}_\ell \mid \mathbf{z}_{\ell+1}) =& \nabla_{\mathbf{\Theta}} \prod_k f(\mathbf{z}_{\ell,k} \mid \mathbf{z}_{\ell+1}) \\
=& \nabla_{\mathbf{\Theta}} \exp\left(\sum_{k=1}^{K_\ell} \left(\sum_{j=1}^{m_{\ell,k}} \log \lambda_{\ell,k}(t_{\ell,k,j}) - \int_0^T \lambda_{\ell,k}(t) dt \right) \right) \\
=& \left(\sum_k \sum_i \left(\nabla_{\mathbf{\Theta}} \lambda_{\ell,k}(t_{\ell,k,i})\right) \cdot \left(\prod_{p\neq k, j\neq i} \lambda_{\ell,p}(t_{\ell,p,j}) \right) \right) \cdot \exp\left(-\sum_k \int_0^T \lambda_{\ell,k}(t) dt \right) \\
&+ \left(\prod_k \prod_i \lambda_{\ell,k}(t_{\ell,k,i}) \exp\left(-\int \lambda_{\ell,k}(t) dt \right) \right) \cdot \left(-\sum_k \int \nabla_{\mathbf{\Theta}} \lambda_{\ell,k}(t) dt \right)
\end{aligned}$$

Since the parameters are in a compact set ($\left|\hat{\mathbf{\Theta}}\right| \leq r$ for some $r > 0$), we can find a bound (by Lemma 41) for $\lambda_{\ell,k}(t)$ *s.t.* $|\lambda_{\ell,k}(t)| \leq m_{\ell+1}a + b$, where $a > 0$ and $b > 0$ are constant numbers. There also exists a bound for $|\nabla_{\mathbf{\Theta}}\lambda_{\ell,k}(t)|$ *s.t.* $|\nabla_{\mathbf{\Theta}}\lambda_{\ell,k}(t)| \leq m_{\ell+1}a_g + b_g$, where $a_g > 0$ and $b_g > 0$. We also notice that $\left|\frac{\nabla_{\mathbf{\Theta}}\lambda_{\ell,k}(t)}{\lambda_{\ell,k}(t)}\right| \leq C_r$ for some constant $C_r > 0$, because $\left|\frac{\nabla_{\mathbf{\Theta}}\phi(\cdot)}{\phi(\cdot)}\right|$ is bounded.

Thus,

$$\begin{aligned}
|\nabla_{\mathbf{\Theta}} f(\mathbf{z}_\ell \mid \mathbf{z}_{\ell+1})| \leq & (m_\ell C_r + (m_{\ell+1}a_g + b_g)K_\ell T)\left(\prod_k\prod_i \lambda_{\ell,k}(t_{\ell,k,i}) \cdot \exp\left(-\int_0^T \lambda_{\ell,k}(t)dt\right)\right) \\
\leq & (m_\ell C_r + (m_{\ell+1}a_g + b_g)K_\ell T)\left(\prod_k\prod_i \lambda_{\ell,k}(t_{\ell,k,i})\right). \qquad (70)
\end{aligned}$$

Notice that

$$\lim_{m\to\infty} \nabla_{\mathbf{\Theta}} w_m(\mathbf{\Theta}) = \mathbb{E}_{\mathbf{z}\sim\text{Poisson}([0,T]^{\sum_\ell K_\ell},1)}\left[\nabla_{\mathbf{\Theta}} f(\mathbf{x},\mathbf{z})\right],$$

we can write down the equation similar to Equation 34, when $\ell = 0$,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{z}\sim\text{Poisson}([0,T]^{\sum_\ell K_\ell},1)}\left[\left|\left(\prod_{i=\ell+1}^{L-1} f(\mathbf{z}_i \mid \mathbf{z}_{i+1})\right)\left(\nabla_{\mathbf{\Theta}} f(\mathbf{z}_\ell \mid \mathbf{z}_{\ell+1})\right)\left(\prod_{j=1}^{\ell} f(\mathbf{z}_{j-1} \mid \mathbf{z}_j)\right)\right|\right] \\
&\leq \sum_{m_L=0}^{\infty} \frac{\mu^{m_L}}{m_L!} \\
&\quad\left(\sum_{m_{L-1}=0}^{\infty} \frac{(a\cdot m_L + b)^{m_{L-1}}}{m_{L-1}!}\left(\cdots \sum_{m_1=0}^{\infty} \frac{(a\cdot m_2 + b)^{m_1}}{m_1!}(a\cdot m_1 + b)^{m_0}\left(C_0'\cdot m_0 + C_0''\cdot m_1 + C_0'''\right)\right)\right),
\end{aligned}$$

where $C_0' > 0, C_0'' > 0, C_0''' > 0$ are constant numbers *w.r.t.* $\{m_\ell\}_\ell$ based on Equation 70.

When $\ell > 0$,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{z}\sim\text{Poisson}([0,T]^{\sum_\ell K_\ell},1)}\left[\left|\left(\prod_{i=\ell+1}^{L-1} f(\mathbf{z}_i \mid \mathbf{z}_{i+1})\right)\left(\nabla_{\mathbf{\Theta}} f(\mathbf{z}_\ell \mid \mathbf{z}_{\ell+1})\right)\left(\prod_{j=1}^{\ell} f(\mathbf{z}_{j-1} \mid \mathbf{z}_j)\right)\right|\right] \\
&\leq \sum_{m_L=0}^{\infty} \frac{\mu^{m_L}}{m_L!}\left(\sum_{m_{L-1}=0}^{\infty} \frac{(a\cdot m_L + b)^{m_{L-1}}}{m_{L-1}!}\left(\cdots\right.\right. \\
&\quad\left.\left.\left(\sum_{m_\ell=0}^{\infty} \frac{(a\cdot m_{\ell+1} + b)^{m_\ell}}{m_\ell!}(C_\ell'\cdot m_\ell + C_\ell''\cdot m_{\ell+1} + C_\ell''')\left(\cdots \sum_{m_1=0}^{\infty} \frac{(a\cdot m_2 + b)^{m_1}}{m_1!}(a\cdot m_1 + b)^{m_0}\right)\right)\right)\right),
\end{aligned}$$

where $C_\ell' > 0, C_\ell'' > 0, C_\ell''' > 0$ are constant numbers *w.r.t.* $\{m_\ell\}_\ell$ based on Equation 70.

Let

$$h_{L-1}^{(1)} = \sum_{m_{L-1}=0}^{\infty} \frac{(a \cdot m_L + b)^{m_{L-1}}}{m_{L-1}!} \left(\cdots \sum_{m_1=0}^{\infty} \frac{(a \cdot m_2 + b)^{m_1}}{m_1!} (a \cdot m_1 + b)^{m_0} \cdot m_0 \right),$$

$$h_{L-1}^{(2)} = \sum_{m_{L-1}=0}^{\infty} \frac{(a \cdot m_L + b)^{m_{L-1}}}{m_{L-1}!} \left(\cdots \sum_{m_1=0}^{\infty} \frac{(a \cdot m_2 + b)^{m_1}}{m_1!} (a \cdot m_1 + b)^{m_0} \cdot m_1 \right),$$

$$h_{L-1}^{(3)} = \sum_{m_{L-1}=0}^{\infty} \frac{(a \cdot m_L + b)^{m_{L-1}}}{m_{L-1}!} \left(\cdots \sum_{m_1=0}^{\infty} \frac{(a \cdot m_2 + b)^{m_1}}{m_1!} (a \cdot m_1 + b)^{m_0} \right),$$

$$h_{L-1}^{(4)} = \sum_{m_{L-1}=0}^{\infty} \frac{(a \cdot m_L + b)^{m_{L-1}}}{m_{L-1}!} \cdot \left(\cdots \left(\sum_{m_\ell=0}^{\infty} \frac{(a \cdot m_{\ell+1} + b)^{m_\ell}}{m_\ell!} \cdot m_\ell \left(\cdots \sum_{m_1=0}^{\infty} \frac{(a \cdot m_2 + b)^{m_1}}{m_1!} (a \cdot m_1 + b)^{m_0} \right) \right) \right),$$

$$h_{L-1}^{(5)} = \sum_{m_{L-1}=0}^{\infty} \frac{(a \cdot m_L + b)^{m_{L-1}}}{m_{L-1}!} \cdot \left(\cdots \left(\sum_{m_\ell=0}^{\infty} \frac{(a \cdot m_{\ell+1} + b)^{m_\ell}}{m_\ell!} \cdot m_{\ell+1} \left(\cdots \sum_{m_1=0}^{\infty} \frac{(a \cdot m_2 + b)^{m_1}}{m_1!} (a \cdot m_1 + b)^{m_0} \right) \right) \right),$$

and it follows that

$$\mathbb{E}_{\mathbf{z} \sim \text{Poisson}([0,T]^{\sum_\ell K_\ell}, 1)} \left[\left| \left(\prod_{i=\ell+1}^{L-1} f(\mathbf{z}_i \mid \mathbf{z}_{i+1}) \right) (\nabla_{\mathbf{\Theta}} f(\mathbf{z}_\ell \mid \mathbf{z}_{\ell+1})) \left(\prod_{j=1}^{\ell} f(\mathbf{z}_{j-1} \mid \mathbf{z}_j) \right) \right| \right]$$
$$\leq \sum_{m_L=0}^{\infty} \frac{\mu^{m_L}}{m_L!} \left(C_0' h_{L-1}^{(1)} + C_0'' h_{L-1}^{(2)} + C_0''' h_{L-1}^{(3)} \right) \text{ for } \ell = 0,$$

$$\mathbb{E}_{\mathbf{z} \sim \text{Poisson}([0,T]^{\sum_\ell K_\ell}, 1)} \left[\left| \left(\prod_{i=\ell+1}^{L-1} f(\mathbf{z}_i \mid \mathbf{z}_{i+1}) \right) (\nabla_{\mathbf{\Theta}} f(\mathbf{z}_\ell \mid \mathbf{z}_{\ell+1})) \left(\prod_{j=1}^{\ell} f(\mathbf{z}_{j-1} \mid \mathbf{z}_j) \right) \right| \right]$$
$$\leq \sum_{m_L=0}^{\infty} \frac{\mu^{m_L}}{m_L!} \left(C_\ell' h_{L-1}^{(4)} + C_\ell'' h_{L-1}^{(5)} + C_\ell''' h_{L-1}^{(3)} \right) \text{ for } \ell > 0.$$

Then we only need to show that $\sum_{m_L=0}^{\infty} \frac{\mu^{m_L}}{m_L!} h_{L-1}^{(i)}$ is bounded for $i = 1, 2, 3, 4, 5$. Notice that h_ℓ is defined in Equation 35.

Lemma 49

$$\sum_{m_L=0}^{\infty} \frac{\mu^{m_L}}{m_L!} h_{L-1}^{(2)} \textit{ is bounded.}$$

Proof We first show that

$$h_\ell^{(2)} = O(m_{\ell+1} e^{C \cdot m_{\ell+1}}), \tag{71}$$

where $C > 0$ is a constant number.

For $\ell = 1$,

$$\begin{aligned}
h_1^{(2)} =& \sum_{m_1=0}^{\infty} \frac{(a \cdot m_2 + b)^{m_1}}{m_1!}(a \cdot m_1 + b)^{m_0} \cdot m_1 \\
=& (a \cdot m_2 + b) \sum_{m_1=1}^{\infty} \frac{(a \cdot m_2 + b)^{m_1-1}}{(m_1-1)!}(a \cdot m_1 + b)^{m_0} \\
=& (a \cdot m_2 + b) \sum_{m_1=0}^{\infty} \frac{(a \cdot m_2 + b)^{m_1}}{m_1!}(a \cdot (m_1+1) + b)^{m_0} \\
\leq& (a \cdot m_2 + b) \cdot a^{m_0} \cdot e^{c \cdot (a \cdot m_2 + b)} \text{ for some constant number } c > 1 \\
=& O(m_2 e^{C \cdot m_2}),
\end{aligned}$$

where $C = c \cdot a$ and the inequality follows from Lemma 42.

Suppose, $h_\ell^{(2)} = O(m_{\ell+1} e^{C' \cdot m_{\ell+1}})$ holds for $\ell = i$ and a constant number $C' > 0$. For $\ell = i + 1$,

$$h_{i+1}^{(2)} = \sum_{m_{i+1}=0}^{\infty} \frac{(a \cdot m_{i+2} + b)^{m_{i+1}}}{m_{i+1}!} h_i^{(2)} \tag{72}$$

$$\leq M \cdot \sum_{m_{i+1}=0}^{\infty} \frac{(a \cdot m_{i+2} + b)^{m_{i+1}}}{m_{i+1}!} m_{i+1} e^{C' \cdot m_{i+1}} \text{ for some constant number } M > 0 \tag{73}$$

$$\leq M \cdot \sum_{m_{i+1}=1}^{\infty} \frac{(a \cdot m_{i+2} + b)^{m_{i+1}}}{(m_{i+1}-1)!} e^{C' \cdot m_{i+1}} \tag{74}$$

$$= M \cdot \sum_{m_{i+1}=0}^{\infty} \frac{(a \cdot (m_{i+2}+1) + b)^{m_{i+1}+1}}{m_{i+1}!} e^{C' \cdot (m_{i+1}+1)} \tag{75}$$

$$= M \cdot e^{C'} \cdot (a \cdot (m_{i+2}+1) + b) \sum_{m_{i+1}=0}^{\infty} \frac{((a \cdot (m_{i+2}+1) + b) \cdot e^{C'})^{m_{i+1}}}{m_{i+1}!} \tag{76}$$

$$= M \cdot e^{C'} \cdot (a \cdot (m_{i+2}+1) + b) \cdot e^{(a \cdot (m_{i+2}+1)+b) \cdot e^{C'}}, \tag{77}$$

where $M > 0$ is a constant number.

Thus, $h_{i+1}^{(2)} = O(m_{i+2} e^{C \cdot m_{i+2}})$, where $C = a \cdot e^{C'}$, and we can conclude that

$$h_\ell^{(2)} = O(m_{\ell+1} e^{C \cdot m_{\ell+1}}),$$

holds for some constant number $C > 0$.

It then follows that

$$\begin{aligned}
&\sum_{m_L=0}^{\infty}\frac{\mu^{m_L}}{m_L!}h_{L-1}^{(2)}\\
\le& M'\cdot\sum_{m_L=0}^{\infty}\frac{\mu^{m_L}}{m_L!}m_L e^{C\cdot m_L}\text{ for some constant } M'>0\\
=& M'\cdot\sum_{m_L=1}^{\infty}\frac{\mu^{m_L}}{(m_L-1)!}e^{C\cdot m_L}\\
=& M'\cdot\sum_{m_L=1}^{\infty}\frac{(\mu e^C)^{m_L}}{(m_L-1)!}\\
=& M'\cdot\mu e^C\sum_{m_L=0}^{\infty}\frac{(\mu e^C)^{m_L}}{(m_L)!}\\
=& M'\cdot\mu e^C e^{\mu e^C}.
\end{aligned}$$

■

Lemma 50

$$\sum_{m_L=0}^{\infty}\frac{\mu^{m_L}}{m_L!}h_{L-1}^{(4)}\ \textit{is bounded.}$$

Proof

$$\begin{aligned}
h_{L-1}^{(4)}=&\sum_{m_{L-1}=0}^{\infty}\frac{(a\cdot m_L+b)^{m_{L-1}}}{m_{L-1}!}\cdot\\
&\left(\cdots\left(\sum_{m_\ell=0}^{\infty}\frac{(a\cdot m_{\ell+1}+b)^{m_\ell}}{m_\ell!}\cdot m_\ell\left(\cdots\sum_{m_1=0}^{\infty}\frac{(a\cdot m_2+b)^{m_1}}{m_1!}(a\cdot m_1+b)^{m_0}\right)\right)\right)\\
=&\sum_{m_{L-1}=0}^{\infty}\frac{(a\cdot m_L+b)^{m_{L-1}}}{m_{L-1}!}\left(\cdots\left(\sum_{m_\ell=0}^{\infty}\frac{(a\cdot m_{\ell+1}+b)^{m_\ell}}{m_\ell!}\cdot m_\ell\cdot h_{\ell-1}\right)\right)\\
\le& M\cdot\sum_{m_{L-1}=0}^{\infty}\frac{(a\cdot m_L+b)^{m_{L-1}}}{m_{L-1}!}\left(\cdots\left(\sum_{m_\ell=0}^{\infty}\frac{(a\cdot m_{\ell+1}+b)^{m_\ell}}{m_\ell!}\cdot m_\ell\cdot e^{c\cdot m_\ell}\right)\right),
\end{aligned}$$

where the second equality follows from Equation 35, the third inequality follows Equation 36, and $c>0, M>0$ are constant numbers.

It follows from Equation 77 that

$$h_{L-1}^{(4)}=O(m_L e^{C\cdot m_L})\text{ for some constant number } C>0,$$

and from Lemma 49 that $\sum_{m_L=0}^{\infty}\frac{\mu^{m_L}}{m_L!}h_{L-1}^{(4)}$ is bounded. ■

Lemma 51
$$\sum_{m_L=0}^{\infty} \frac{\mu^{m_L}}{m_L!} h_{L-1}^{(5)} \textit{ is bounded.}$$

Proof

$$\begin{aligned}
h_{L-1}^{(5)} = &\sum_{m_{L-1}=0}^{\infty} \frac{(a\cdot m_L + b)^{m_{L-1}}}{m_{L-1}!} \cdot \\
&\left(\cdots \left(\sum_{m_\ell=0}^{\infty} \frac{(a\cdot m_{\ell+1}+b)^{m_\ell}}{m_\ell!} \cdot m_{\ell+1} \left(\cdots \sum_{m_1=0}^{\infty} \frac{(a\cdot m_2+b)^{m_1}}{m_1!} (a\cdot m_1+b)^{m_0}\right)\right)\right) \\
= &\sum_{m_{L-1}=0}^{\infty} \frac{(a\cdot m_L + b)^{m_{L-1}}}{m_{L-1}!} \left(\cdots \left(\sum_{m_\ell=0}^{\infty} \frac{(a\cdot m_{\ell+1}+b)^{m_\ell}}{m_\ell!} \cdot m_{\ell+1} \cdot h_{\ell-1}\right)\right) \\
\le &M \cdot \sum_{m_{L-1}=0}^{\infty} \frac{(a\cdot m_L + b)^{m_{L-1}}}{m_{L-1}!} \left(\cdots \left(m_{\ell+1} \cdot \sum_{m_\ell=0}^{\infty} \frac{(a\cdot m_{\ell+1}+b)^{m_\ell}}{m_\ell!} \cdot e^{c\cdot m_\ell}\right)\right) \\
= &M \cdot \sum_{m_{L-1}=0}^{\infty} \frac{(a\cdot m_L + b)^{m_{L-1}}}{m_{L-1}!} \left(\cdots \sum_{m_{\ell+1}=0}^{\infty} \frac{(a\cdot m_{\ell+2}+b)^{m_{\ell+1}}}{m_{\ell+1}!} \left(m_{\ell+1} \cdot e^{(a\cdot m_{\ell+1}+b)e^c}\right)\right),
\end{aligned}$$

where $M > 0$ and $c > 0$ are constant numbers. It also follows from Equation 77 that

$$h_{L-1}^{(5)} = O(m_L e^{C\cdot m_L}) \text{ for some constant number } C > 0,$$

and from Lemma 49 that $\sum_{m_L=0}^{\infty} \frac{\mu^{m_L}}{m_L!} h_{L-1}^{(5)}$ is bounded. ■

According to Equations 68 and 69, it is easy to see Theorem 9 follows from the following lemma.

Lemma 52 *Suppose* $\left|\hat{\mathbf{\Theta}}\right| \le r$ *for some* $r > 0$,

$$\nabla_{\mathbf{\Theta}} \sum_{n=0}^{\infty} \frac{\exp(-|S|)}{n!} \int f(\mathbf{x}, \mathbf{z}) d\mathbf{z}_{|n} = \sum_{n=0}^{\infty} \frac{\exp(-|S|)}{n!} \int \nabla_{\mathbf{\Theta}} f(\mathbf{x}, \mathbf{z}) d\mathbf{z}_{|n}.$$

Proof The statement that $\sum_{m_L=0}^{\infty} \frac{\mu^{m_L}}{m_L!} h_{L-1}^{(3)}$ is bounded has been proved in Equation 36, and it follows easily that $\sum_{m_L=0}^{\infty} \frac{\mu^{m_L}}{m_L!} h_{L-1}^{(1)}$ is bounded since m_0 is a constant number. Combining Lemmas 49,50, and 51, it follows that $\nabla_{\mathbf{\Theta}} w_m(\mathbf{\Theta})$ converges uniformly.

$w_m(\mathbf{\Theta})$ converges for some $\mathbf{\Theta}$ follows from Proposition 6.

Thus, it follows from Corollary 48 that

$$\nabla_{\mathbf{\Theta}} \sum_{n=0}^{\infty} \frac{\exp(-|S|)}{n!} \int f(\mathbf{x}, \mathbf{z}) d\mathbf{z}_{|n} = \sum_{n=0}^{\infty} \frac{\exp(-|S|)}{n!} \int \nabla_{\mathbf{\Theta}} f(\mathbf{x}, \mathbf{z}) d\mathbf{z}_{|n}.$$

■

Theorem 9 *Suppose $\left|\hat{\boldsymbol{\Theta}}\right| \leq r$ for some $r > 0$, then*

$$\mathbb{E}_{\mathbf{z}\sim f(\mathbf{Z}|\mathbf{x})}[\nabla_{\boldsymbol{\Theta}} \log f(\mathbf{x},\mathbf{z})] = \nabla_{\boldsymbol{\Theta}} \log f(\mathbf{x}).$$

Proof This theorem is an immediate result of Lemma 52, Equations 68 and 69. ■

C.2 Interchange of the Expectation and the Gradient for Virtual Point Processes

Similar to the proof of Theorem 9, we prove Theorem 10 by Theorem 46 and Corollary 47.

Lemma 53 *Suppose $\left|\hat{\boldsymbol{\Theta}}\right| \leq r$ for some $r > 0$ and each point process is restricted in a compact set B(e.g.,[0,T]), then the expected number of events $\bar{\Lambda}_{\ell,k}(B) = \mathbb{E}_{\mathbf{z}\sim f(\mathbf{z})}[N]$, where N represents the number of hidden points, for a point process $Z_{\ell,k}$ in a DNSP is bounded, and*

$$\bar{\Lambda}_{\ell,k}(B) \leq \sum_{h=\ell+1}^{L} C^{h-\ell} \cdot \max_i\{\mu_{h,i} \cdot v(B)\}, \tag{78}$$

where $C > 0$ is a constant number, and $v(B)$ is the volume of the compact set B.

Proof The expected intensity function $\bar{\lambda}_{\ell,k}$ for $Z_{\ell,k}$ is

$$\bar{\lambda}_{\ell,k}(t) = \mathbb{E}_{Z_{\ell+1,i}}\left[\mathbb{E}_{t_{\ell+1,i,j}\sim Z_{\ell+1,i}}\left[\mu_{\ell,k} + \sum_{i=1}^{K_{\ell+1}} \sum_{t_{\ell+1,i,j}} \phi_{\boldsymbol{\theta}_{(\ell+1,i)\to(\ell,k)}}(t - t_{\ell+1,i,j})\right]\right], \tag{79}$$

where $\mu_{\ell,k} = 0$ for $\ell \neq L$ and $\bar{\lambda}_{L,k}(t) = \mu_{L,k}$.

By Campbell's theorem,

$$\bar{\lambda}_{\ell,k}(t) = \mu_{\ell,k} + \sum_{i=1}^{K_{\ell+1}} \int \phi_{\boldsymbol{\theta}_{(\ell+1,i)\to(\ell,k)}}(\tau)\bar{\lambda}_{\ell+1,i}(\tau)d\tau. \tag{80}$$

It follows that

$$\bar{\Lambda}_{\ell,k}(B) = \int_B \bar{\lambda}_{\ell,k}(t)dt. \tag{81}$$

By $\left|\hat{\boldsymbol{\Theta}}\right| \leq r$ for some $r > 0$,

$$\bar{\Lambda}_{\ell,k}(B) \leq C \cdot \max_i\{\bar{\Lambda}_{\ell+1,i}(B)\} + \mu_{\ell,k} \cdot v(B), \tag{82}$$

where the constant number C exists because there exists a bound for the parameters such that the kernel function $\phi(\cdot)$ is also bounded, and the number of point processes per layer is also bounded. Then it follows

$$\bar{\Lambda}_{\ell,k}(B) \leq \sum_{h=\ell}^{L} C^{h-\ell} \cdot \max_i\{\mu_{L,i} \cdot v(B)\}. \tag{83}$$

■

Lemma 54 *Suppose* $\left|\hat{\mathbf{\Theta}}\right| \leq r$ *for some* $r > 0$, $\mathbb{E}_{\mathbf{z}\sim f(\mathbf{z}|\mathbf{x})}[N]$ *is finite, where* N *represents the number of hidden points.*

Proof We shall prove this by contradiction. Suppose $\mathbb{E}_{\mathbf{z}\sim f(\mathbf{z}|\mathbf{x})}[N] = \infty$, then there exists a posterior point process for $Z_{\ell,i}$, *s.t.* $\exists \varepsilon_0 > 0, \forall C_0 > 0, \sum_{n=m_0}^{\infty} p(N_{\ell,i} = n \mid \mathbf{x}) \cdot n > \varepsilon_0$, where $N_{\ell,i}$ represents the number of points for $Z_{\ell,i}$.

However, we know

$$\sum_{n=C_0}^{\infty} p(N_{\ell,i} = n \mid \mathbf{x}) \cdot n \tag{84}$$

$$= \sum_{n=C_0}^{\infty} \frac{p(N_{\ell,i} = n, \mathbf{x})}{f(\mathbf{x})} \cdot n \tag{85}$$

$$\leq \sum_{n=C_0}^{\infty} \frac{p(N_{\ell,i} = n)}{f(\mathbf{x})} \cdot n, \tag{86}$$

where $f(\mathbf{x})$ is the density of $\mathbf{x}$. Lemma 53 tells us that $\exists C_1 > 0$, $\forall \varepsilon_1 > 0$, $\sum_{n=C_1}^{\infty} p(N_{\ell,i} = n) \cdot n \leq \varepsilon_1$. Let $\varepsilon_1 = f(\mathbf{x})\varepsilon_0$ and $C_0 = C_1$, we can get (by Equation 86)

$$\sum_{n=C_0}^{\infty} p(N_{\ell,i} = n \mid \mathbf{x}) \cdot n \leq \epsilon_0, \tag{87}$$

which contradicts our assumption. ■

Lemma 55 *Suppose* $\left|\hat{\mathbf{\Theta}}\right| \leq r$ *for some* $r > 0$, *then there exists* $\tilde{a} > 0$, $\tilde{b} > 0$, $\tilde{a}_g > 0$, *and* $\tilde{b}_g > 0$ *such that*

$$|\tilde{\lambda}_{\ell,k}| \leq n_m \cdot \tilde{a} + \tilde{b},$$

where $n_m = \max\{n_o, n\}$, n_o *is the number of points in the observation, and* n *is the number of hidden points of a realization for a real point process.*

Proof Since $\tilde{\lambda}_{\ell,k}(t) = \tilde{\mu}_{\ell,k} + \sum \phi(\cdot)$ or $\tilde{\lambda}_{\ell,k}(t)$ is represented as a self-attention map and the parameters are in a compact set ($\left|\hat{\mathbf{\Theta}}\right| \leq r$ for some $r > 0$), we can find a bound for $\tilde{\lambda}(t)$ *s.t.* $|\tilde{\lambda}(t)| \leq n_m \cdot \tilde{a} + \tilde{b}$, where $\tilde{a} > 0$ and $\tilde{b} > 0$. ■

Theorem 10 *Suppose* $\left|\hat{\mathbf{\Theta}}\right| \leq r$ *for some* $r > 0$, *then*

$$\mathbb{E}_{\mathbf{z}\sim f(\mathbf{Z}|\mathbf{x};\mathbf{\Theta})}\left[\nabla_{\tilde{\mathbf{\Theta}}} \log \tilde{f}(\mathbf{z}; \tilde{\mathbf{\Theta}})\right] = \nabla_{\tilde{\mathbf{\Theta}}} \mathbb{E}_{\mathbf{z}\sim f(\mathbf{Z}|\mathbf{x};\mathbf{\Theta})}\left[\log \tilde{f}(\mathbf{z}; \tilde{\mathbf{\Theta}})\right].$$

Proof We can write the expectation as a sequence,

$$\mathbb{E}_{f(\mathbf{z}|\mathbf{x};\mathbf{\Theta})}[-\log \tilde{f}(\mathbf{z}; \mathbf{x}, \tilde{\mathbf{\Theta}})] = -\lim_{m\to\infty} g_m(\tilde{\mathbf{\Theta}}), \tag{88}$$

where $g_m(\tilde{\boldsymbol{\Theta}}) = \sum_{n=0}^{m} p(n) \int \log \tilde{f}(\mathbf{z}; \mathbf{x}, \tilde{\boldsymbol{\Theta}}) f(\mathbf{z}) d\mathbf{z}_{|n}$, $p(n)$ is the probability that the number of points of a realization $\mathbf{z}$ from the posterior point processes equals n, and $f(\mathbf{z})$ is the density of the realization $\mathbf{z}$ given the number of points n.

Because we can interchange the gradient with the summation if the series has finite terms, we can get

$$\begin{aligned}\nabla_{\tilde{\boldsymbol{\Theta}}} g_m(\tilde{\boldsymbol{\Theta}}) =& \sum_{n=0}^{m} p_{\mathbf{z}|\mathbf{x}}(n) \nabla_{\tilde{\boldsymbol{\Theta}}} \int \log \tilde{f}(\mathbf{z}; \mathbf{x}, \tilde{\boldsymbol{\Theta}}) \tilde{f}(\mathbf{z}) d\mathbf{z}_{|n} \\ =& \sum_{n=0}^{m} p_{\mathbf{z}|\mathbf{x}}(n) \int f(\mathbf{z}) \nabla_{\tilde{\boldsymbol{\Theta}}} \log \tilde{f}(\mathbf{z}; \mathbf{x}, \tilde{\boldsymbol{\Theta}}) d\mathbf{z}_{|n}, \end{aligned} \tag{89}$$

where the second equality follows from dominated convergence theorem.

By Lemma 55, $|\tilde{\lambda}_{\ell,k}| \leq n_m \cdot \tilde{a} + \tilde{b}$. Similarly, there also exists a bound for the gradient $|\nabla \lambda^I(t)|$ *s.t.* $|\nabla \lambda^I(t)| \leq n_m \cdot \tilde{a}_g + \tilde{b}_g$. We also notice that there exists a constant C *s.t.*

$$\left| \frac{\nabla_{\tilde{\boldsymbol{\Theta}}} \tilde{\lambda}(t)}{\tilde{\lambda}(t)} \right| \leq C, \tag{90}$$

because (1) if $\tilde{\lambda}(t) = \tilde{\mu} + \sum \phi(\cdot)$, then $\nabla_{\tilde{\boldsymbol{\Theta}}} \tilde{\lambda}(t) = \sum \nabla_{\tilde{\boldsymbol{\Theta}}} \phi(\cdot)$ and $\left| \frac{\nabla_{\tilde{\boldsymbol{\Theta}}} \phi(\cdot)}{\phi(\cdot)} \right|$ is bounded; (2) if $\tilde{\lambda}(t)$ is an output from a self-attention network, then it is obvious both $\nabla_{\tilde{\boldsymbol{\Theta}}} \tilde{\lambda}(t)$ and $\tilde{\lambda}(t)$ are bounded.

We notice that

$$\left| \nabla_{\tilde{\boldsymbol{\Theta}}} \log \tilde{f}(\mathbf{z}; \mathbf{x}, \tilde{\boldsymbol{\Theta}}) \right| = \left| \nabla_{\tilde{\boldsymbol{\Theta}}} \sum_{\ell=1}^{L} \sum_{k=1}^{K_\ell} \left(-\int \tilde{\lambda}_{\ell,k}(t) dt + \sum_{i=1}^{\tilde{m}_{\ell,k}} \log \tilde{\lambda}_{\ell,k}(\tilde{t}_{\ell,k,i}) \right) \right| \tag{91}$$

$$\leq \sum_{\ell=1}^{L} \sum_{k=1}^{K_\ell} \left(\left| \int \nabla_{\tilde{\boldsymbol{\Theta}}} \tilde{\lambda}_{\ell,k}(t) dt \right| + \sum_{i=1}^{\tilde{m}_{\ell,k}} \left| \frac{\nabla_{\tilde{\boldsymbol{\Theta}}} \tilde{\lambda}_{\ell,k}(t)}{\tilde{\lambda}_{\ell,k}(t)} \right| \right) \tag{92}$$

$$\leq (n_m \cdot \tilde{a}_g + \tilde{b}_g) \cdot (\sum K_\ell) \cdot T + n_m C \tag{93}$$

Let $g_n^* = p_{\mathbf{z}|\mathbf{x}}(n) \int f(\mathbf{z}) \nabla_{\tilde{\boldsymbol{\Theta}}} \log \tilde{f}(\mathbf{z}; \mathbf{x}, \tilde{\boldsymbol{\Theta}}) d\mathbf{z}_{|n}$, and we know

$$|g_n^*| \leq p_{\mathbf{z}|\mathbf{x}}(n) \left((n_m \cdot \tilde{a}_g + \tilde{b}_g) \cdot (\sum K_\ell) \cdot T + n_m C \right),$$

where $p_{\mathbf{z}|\mathbf{x}}(n)$ is the probability of the number of hidden points equal to n.

It also follows that

$$\sum_n p_{\mathbf{z}|\mathbf{x}}(n) \cdot n_m = \mathbb{E}_{\mathbf{z} \sim f(\mathbf{z}|\mathbf{x})}[N] + \sum_{i=0}^{n_o} p_{\mathbf{z}|\mathbf{x}}(i) \cdot (n_o - i),$$

as $n_m = n_o$ when $n < n_o$.

Let $M_n^* = p_{\mathbf{z}|\mathbf{x}}(n) \left((n_m \cdot \tilde{a}_g + \tilde{b}_g) \cdot (\sum K_\ell) \cdot T + n_m C \right)$, $\sum_n M_n^*$ converges (by Lemma 54) and $\sum_n M_n^* = (\mathbb{E}_{\mathbf{z} \sim f(\mathbf{z}|\mathbf{x})}[N] - \sum_{i=0}^{n_o} p(i) \cdot (n_o - i))(\tilde{a}_g \cdot (\sum_\ell K_\ell) \cdot T + C) + \tilde{b}_g \cdot (\sum_\ell K_\ell) \cdot T$.

Then it follows (by Theorem 46) $\nabla_{\tilde{\mathbf{\Theta}}} g_m(\tilde{\mathbf{\Theta}}) = \sum_{n=0}^{m} g_n^*(\tilde{\mathbf{\Theta}})$ converges uniformly. It is easy to see $g_m(x_0)$ converges for some point x_0.

The uniform convergence implies, by Corollary 48, that $\lim_{m\to\infty} g_m$ exists, and

$$\lim_{m\to\infty} \nabla_{\tilde{\mathbf{\Theta}}} g_m(\tilde{\mathbf{\Theta}}) = \nabla_{\tilde{\mathbf{\Theta}}} \lim_{m\to\infty} g_m(\tilde{\mathbf{\Theta}}). \tag{94}$$

By Equations 89 and 88, we know $\nabla_{\tilde{\mathbf{\Theta}}} \mathbb{E}_{f(\mathbf{z}|\mathbf{x};\mathbf{\Theta})}\left[\log \tilde{f}(\mathbf{z};\mathbf{x},\tilde{\mathbf{\Theta}})\right] = \mathbb{E}_{f(\mathbf{z}|\mathbf{x};\mathbf{\Theta})}\left[\nabla_{\tilde{\mathbf{\Theta}}} \log \tilde{f}(\mathbf{z};\mathbf{x},\tilde{\mathbf{\Theta}})\right]$.
■

Appendix D. Convergence of MCMC

Proposition 21 *Let $M_i(x,\cdot)$ be the i-th transition kernel that proposes to move a point configuration $x \in \Omega$ to another point configuration $x' \in \Omega$, and $(i_1, i_2, \cdots, i_n)$ be any sequence of transition kernels. Assume that each of the point of x accepts at least one drift in $(i_1, i_2, \cdots, i_n)$, then*

$$(M_{i_1} M_{i_2} \cdots M_{i_n})(x, A) = 0$$

for all $x \in \Omega$ and $A \subseteq \Omega$ with the Lebesgue measure of A is 0.

Proof For each re-sampling, we re-sample a set of virtual events, it is easy to see, from Equation 10, if the re-sampled events fall into a set with measure 0, then the probability of generating these re-sampled events is 0.

The real events can only appear where the virtual events appear, so the measure of the set of real events is also 0.

Thus, we complete the proof. ■

Corollary 22 *If the Lebesgue measure of A is 0, then*

$$P[X_n \in A \mid X_0 = x_0] \le P[D_n],$$

where $X_n \in \Omega$, and D_n is the event that, by time n, the chain has not yet accepted at least one drift for each event of the original starting realization.

Proof

$$\begin{aligned}
&P[X_n \in A \mid X_0 = x_0] \\
=&P[X_n \in A, D_n \mid X_0 = x_0] + P[X_n \in A, \neg D_n \mid X_0 = x_0] \\
=&P[X_n \in A, D_n \mid X_0 = x_0] + 0 \ \text{(by Proposition 21)} \\
\le&P[X_n \in A, D_n \mid X_0 = x_0] + P[X_n \notin A, D_n \mid X_0 = x_0] \\
=&P[D_n]
\end{aligned}$$

■

Proposition 23 *If a Ψ-irreducible Markov chain eventually accepts at least one drift for each real and virtual event, then this Markov chain is Harris recurrent.*

Proof If $\Pi(r, A_r, v, A_v) = 0$, then $\mathcal{U}_{r+v}(A_r, A_v) = 0$ because $f_{r+v}(x) > 0$ for any $x \in (\Omega_r, \Omega_v)$.

By Corollary 22, we have

$$P[X_n \in A \ \forall n \mid X_0 = x] \leq \lim_{n\to\infty} P[X_n \in A \mid X_0 = x] \leq \lim_{n\to\infty} P[D_n \mid X_0 = x] = 0,$$

where X_n represents the point configuration of the real and virtual point processes at state $\mathcal{C}_n$, $A = (A_r, A_v)$, the last equality follows from the fact that the Markov chain accepts at least one drift eventually with probability 1.

Thus, the result follows from Theorem 19. ■

Theorem 14 *For a Ψ-irreducile and aperiodic Markov chain described in virtual-event-based RJMCMC, if the re-sampling of virtual events and the acceptance of the drift combination for each real event happen eventually, then*

$$\lim_{m\to\infty} \|Q^m - \Pi\|_{\mathrm{TV}} = 0$$

holds for this Markov chain, where Π is the invariant distribution.

Proof It follows directly from Proposition 23. ■

D.1 Neyman-Scott Processes

Proposition 26 *The Markov chain $\mathcal{C}$ generated by Algorithm 1 is Ψ-irreducible and aperiodic. The continuous set of point configurations can be chosen as*

$$U = \left\{ \left(\left(\left\{ r = \sum_{\ell} K_\ell \right\} \times U_r^R \right), \left(\{0\} \times U_0^V \right) \right) \right\},$$

where U_r^R is the set of points such that $\mathbf{z}_{\ell,k} = \{t_{\ell,k,0} : 0 < t_{\ell,k,0} < \min_{i,j}\{t_{\ell-1,i,j}\}\}$.

Proof Let $x = ((\{r\} \times \{u_1^R, u_2^R, \cdots, u_r^R\}), (\{v\} \times \{u_1^V, u_2^V, \cdots, u_v^V\})) \in \Omega$ be a point configuration to be moved and $m = r$ be the number of transition steps.

If $m = r = \sum K_\ell$, then for each virtual point process, we draw a new set of points and this set is an empty set, which makes sure that each virtual point process has an empty set of virtual events. For this case, $\sum K_\ell$ is the total number of virtual point processes, and the probability of drawing an empty set of points for $\tilde{Z}_{\ell,k}$ is equal to $\tilde{P}_{\ell,k}(\emptyset) = \exp\left(-\int \tilde{\lambda}_{\ell,k}(t)dt\right)$. According to Remark 27, each real point process has one real event. The probability of moving any x to U_r^R is

$$P^m(x, U_r^R) = \prod_{\ell,k} \left(\frac{1}{\sum K_\ell} P_{re-sample} \cdot \tilde{P}_{\ell,k}(\emptyset) \right) > 0, \qquad (95)$$

where $\frac{1}{\sum K_\ell}$ is the probability of choosing the point process $\tilde{Z}_{\ell,k}$, and $P_{re-sample}$ is the probability of resampling virtual events.

If $m > \sum K_\ell$, then one possible way to ensure x is transited to U_r^R is to do flips $(m - \sum K_\ell)$ times first, and draw an empty set of virtual events similar to the case $m = r$.

$$P^m(x, U_r^R) \geq \left(\prod_{t=1}^{m-\sum K_\ell} \frac{1}{\sum K_\ell} P_{flip} \cdot \frac{1}{M(t)} \cdot \min\{1, \mathcal{P}_r^t\} \right) \tag{96}$$

$$\cdot \prod_{\ell,k} \left(\frac{1}{\sum K_\ell} P_{re-sample} \cdot \tilde{P}_{\ell,k}(\emptyset) \right) > 0, \tag{97}$$

where P_{flip} is the probability of choosing the flip move, $M(t)$ is the total number of real and virtual events when performing the t-th flip, and $\mathcal{P}_r^t$ is the Hastings ratio for performing the t-th flip, $P_{re-sample}$ is the probability of re-sampling virtual events (Hong and Shelton, 2023, Appx. B.3). The inequality comes from the fact that we can either re-sample at first or at last.

If $\Psi(F) > 0$, then $P^m(x, F) \geq P^m(x, U_r^R) > 0$ for any x for $m \geq 1$, which implies the chain is Ψ-irreducible.

If $x \in U_r^R$, $P(x, \{x\}) > 0$, which implies the chain is aperiodic. ■

Lemma 56 *For a sequence of random variables $\{A_n\}$, if the probability $P(A_{n_i} = a) > 0$ for a subsequence $\{A_{n_i}\}$, then there exists $N > 0$ s.t. $A_N = a$ with probability 1.*

Proof The probability of the event that $A_n \neq a$ for all n is equal to

$$\lim_{i \to \infty} \prod_i P(A_{n_i} \neq a) = 0.$$

■

Definition 57 (non-empty-event region) *The non-empty-event region is the region where there must be at least one real event for a realization of NSPs of a Markov state.*

Lemma 58 *The probability of having a newly generated real event (i.e., the real event is flipped from a virtual event) in a non-empty-event region infinitely often is 1.*

Proof Suppose there exists a non-empty-event region B, where the number of times a real event can appear is bounded. Then there exists some $N \in \mathbb{N}$ such that for all Markov states $Y_n (n > N)$, Y_n does not have any real events inside B. We denote this situation as event E and try to calculate the probability of E.

We have the following situations and calculate the probability P_1, P_2, P_3 for each of them respectively:

1. **No virtual events are generated inside B for all $Y_n(n > N)$**

   If there are no virtual events ever generated inside B, then there would be no real events. For $Y_n(n > N)$, there exists a subsequence $\{Y_{i_j}\}_j$ which only consists of re-sampling. Every Y_{i_j} does not generate any virtual event. We calculate the probability P_1 for such sequences,

   $$P_1 \leq \prod_{j=1}^{\infty} \exp\left(-\int_B \tilde{\mu} dt\right) = 0,$$

   where $\tilde{\mu}$ is the base intensity for virtual point processes inside B, $\exp\left(-\int_B \tilde{\mu} dt\right)$ is the upper bound for the probability that no virtual events can be generated inside B.

2. **A bounded number of states of $Y_n(n > N)$ have virtual events inside B**

   Similar to P_1, the probability P_2 of this situation is 0.

3. **Virtual events occur infinitely often inside B. However, only a bounded number of virtual events can be changed to real events for $Y_n(n > N)$**

   A flip or swap needs to be accepted to change a virtual event to a real event, which means that rejections for flips or swaps appear infinitely often. For a non-empty-event region, the probability of rejecting a flip for a virtual event or a swap is less than 1, and the probability of having infinitely often rejection is 0 (by Lemma 56). Thus, the probability P_3 of this situation is 0.

According to the analysis of the above situations, the probability of having a real event inside B infinitely often is 1. ■

Proposition 28 *The re-sampling of virtual events and the drift combination for each real event happen eventually for the Markov chain generated by Algorithm 1.*

Proof It is obvious the re-sampling of each virtual point process occurs with a positive probability an infinite number of times, so the re-sampling of virtual point processes eventually happens.

For each real event at each step, the probability of accepting a drift combination is positive or 0. If the probability of accepting a drift combination for a real event is positive, then the drift combination is accepted eventually, because otherwise rejecting a drift combination happens with a positive probability an infinite number of times, which is a contradiction by Lemma 56.

The only situation that the probability of accepting a drift combination for a real event equals 0 is that the intensity of the real events from a layer below entirely depends on this real event.

We prove in the following that each real event accepts a drift combination eventually.

Suppose, until eventually ($n \to \infty$), the probability of not accepting a drift combination for $t_{\ell,i,0}$ is greater than 0, then there exists a $N \in \mathbb{N}$ such that the probability of rejection is 1 for $Y_n(n > N)$, which implies there must be events at layer $\ell - 1$ fully depend on $t_{\ell,i,0}$ for $Y_n(n > N)$. We denote the smallest time for the event at layer $\ell - 1$ that fully depends on $t_{\ell,i,0}$ as $t_{\ell-1,k,\delta}(t_{\ell-1,k,\delta} > t_{\ell,i,0})$.

Then $[t_{\ell,i,0}, t_{\ell-1,k,\delta})$ at layer ℓ is a non-empty-event region. According to Lemma 58, the probability of having a newly generated real event infinitely often inside $[t_{\ell,i,0}, t_{\ell-1,k,\delta})$ is 1, which implies that even the event with the smallest time $(t_{\ell-1,k,\delta})$ does not fully depend on $t_{\ell,i,0}$ infinitely often. This contradicts the above argument that there must be events at layer $\ell - 1$ fully depend on $t_{\ell,i,0}$ for $Y_n(n > N)$. Thus, the probability of accepting a drift combination eventually $(n \to \infty)$ for each real event is 1. ■

D.2 Markov Jump Processes and Extensions

Proposition 30 *If there exists a continuous set of point configurations U, then the Markov chain for the MCMC algorithm proposed in Rao and Teh (2013) is convergent.*

Proof After every 2 steps of sampling for the virtual jumps and the new MJP path, any state x can be moved to any other state inside U with positive probability, as the probability of generating a point configuration inside U is greater than 0. This implies Ψ-irreducibility.

For any x with virtual jumps dropped, we have

$$P^m(x, \{x\}) > 0,$$

where $m = 2$. Because we can simply generate no events from the Poisson process with piecewise-constant rate

$$R(t) = (\Omega + A_{S(t)}),$$

and then the states can stay at the same state with probability greater than 0 after performing the forward-filtering backward-sampling algorithm. This implies aperiodicity.

Virtual jumps are re-sampled every 2 steps and each existing jump accepts a drift combination with a positive probability when implementing the forward-filtering backward-sampling algorithm, which implies Harris recurrence and convergence by Theorem 14. ■

D.3 Hawkes Processes

Proposition 32 *The Markov chain generated by the MCMC algorithm proposed in Shelton et al. (2018) is Ψ-irreducible and aperiodic. The continuous set of point configurations can be*

$$U = \{((\{0\} \times U_0^R), (\{0\} \times U_0^V))\}.$$

Proof Let $x = ((\{r\} \times \{u_1^R, u_2^R, \cdots, u_r^R\}), (\{v\} \times \{u_1^V, u_2^V, \cdots, u_v^V\})) \in \Omega$ be a point configuration to be moved, $n_{evidence}$ be the number of events in the evidence. Different than the setting of the real events in Shelton et al. (2018), the root event and evidence events are not included in the real events. m represents the number of transition steps and we set $m = 2(r + n_{evidence}) + 1$.

We first propose to resample the parents for the real events to let the parents of the real events and the evidence events be root events, and the number of moves is $r + n_{evidence}$ because each of these events needs a resampling. The probability of choosing the Parent move at each MCMC step is $\frac{1}{3}$, and the probability of selecting the root event as the parent for the real events and the evidence events is greater than 0.

Then, we propose to change the real events, which only have virtual children, to virtual events with r moves of Virtualness, because the events in x currently only have virtual children after the resampling of the parents. Similarly, the probability of these moves is greater than 0.

Finally, we propose to resample the virtual children for the root events and evidence events, with $1 + n_{evidence}$ moves. The probability of generating an empty set of virtual events is greater than 0.

Thus, $m = r + n_{evidence} + r + 1 + n_{evidence} = 2(r + n_{evidence}) + 1$, and the probability of moving x to $\{((\{0\} \times U_0^R), (\{0\} \times U_0^V))\}$ is

$$P^m(x, \{((\{0\} \times U_0^R), (\{0\} \times U_0^V))\}) > 0,$$

and if $\Psi(F) > 0$, then $P^m(x, F) \geq P^m(s, \{((\{0\} \times U_0^R), (\{0\} \times U_0^V))\}) > 0$.

We can also find that if $x = \{((\{0\} \times U_0^R), (\{0\} \times U_0^V))\}$, then $P(x, \{x\}) > 0$, which demonstrates that the chain is aperiodic. ■

Proposition 33 *The re-sampling of virtual events and the drift combination for each real event happen eventually for the Markov chain in Shelton et al. (2018).*

Proof Similar to Proposition 28, it is obvious at least one re-sampling of virtual events will be accepted.

A real event can only have virtual children, or have both real and virtual children. When a real event only has virtual children, we can first propose to move this real event to a virtual event and then perform a Virtual children move for the parent of this newly moved virtual event. After accepting the Virtual children move, the real event accepts a drift combination. The probability of accepting a Virtualness move followed by a Virtual Children move for this real event is greater than 0 at any Markov chain state. Similarly, when a real event with both real and virtual children, the probability of accepting a Parent move for each real child, which removes the real children for this real event, then a Virtualness move followed by a Virtual Children move is greater than 0 at any Markov chain state. Thus, a drift combination for each real event will be accepted eventually. ■

References

Ina Trolle Andersen, Ute Hahn, Eva C Arnspang, Lene Niemann Nejsum, and Eva B Vedel Jensen. Double Cox cluster processes—with applications to photoactivated localization microscopy. *Spatial statistics*, 27:58–73, 2018.

BARD. Bay area regional deformation network. UC Berkeley Seismological Laboratory. dataset., 2014.

Luc Bauwens and Nikolaus Hautsch. Modelling financial high frequency data using point processes. In *Handbook of financial time series*, pages 953–979. Springer, 2009.

BDSN. Berkeley digital seismic network. UC Berkeley Seismological Laboratory. dataset., 2014.

Albert Benveniste, Michel Métivier, and Pierre Priouret. *Adaptive algorithms and stochastic approximations*, volume 22. Springer Science & Business Media, 2012.

Ricky T. Q. Chen, Brandon Amos, and Maximilian Nickel. Neural spatio-temporal point processes. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=XQQA6-So14`.

COC. City of Chicago, Crimes - 2001 to present. `https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2`, 2022. Accessed: 2022-08-14.

Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using graph convolutional networks. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/f507783927f2ec2737ba40afbd17efb5-Paper.pdf`.

Mark Gales, Steve Young, et al. The application of hidden Markov models in speech recognition. *Foundations and Trends® in Signal Processing*, 1(3):195–304, 2008.

Peter J Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

Asela Gunawardana, Christopher Meek, and Puyang Xu. A model for temporal dependencies in event streams. *Advances in neural information processing systems*, 24, 2011.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Chengkuan Hong and Christian Shelton. Deep Neyman-Scott processes. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 3627–3646. PMLR, 28–30 Mar 2022. URL `https://proceedings.mlr.press/v151/hong22a.html`.

Chengkuan Hong and Christian Shelton. Variational inference for Neyman-Scott processes. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 2002–2018. PMLR, 25–27 Apr 2023.

Chengkuan Hong and Christian R Shelton. Convolutional deep exponential families. *arXiv preprint arXiv:2110.14800*, 2021.

HRSN. High resolution seismic network. UC Berkeley Seismological Laboratory. dataset., 2014.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Jiří Kopeckỳ and Tomáš Mrkvička. On the Bayesian estimation for the stationary Neyman-Scott point processes. *Applications of Mathematics*, 61(4):503–514, 2016.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Wenzhao Lian, Ricardo Henao, Vinayak Rao, Joseph Lucas, and Lawrence Carin. A multi-task point process predictive model. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2030–2038, Lille, France, 07–09 Jul 2015. PMLR.

Hongyuan Mei and Jason M Eisner. The neural Hawkes process: A neurally self-modulating multivariate point process. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/6463c88460bd63bbe256e495c63aa40b-Paper.pdf`.

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

Jesper Møller and Rasmus Plenge Waagepetersen. *Statistical inference and simulation for spatial point processes*. CRC Press, 2003.

Christian Naesseth, Fredrik Lindsten, and David Blei. Markovian score climbing: Variational inference with KL(p || q). In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15499–15510. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/b20706935de35bbe643733f856d9e5d6-Paper.pdf`.

NCEDC. Northern California earthquake data center. UC Berkeley Seismological Laboratory. dataset., 2014.

Jerzy Neyman and Elizabeth L Scott. Statistical approach to problems of cosmology. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(1):1–29, 1958.

Zhijian Ou and Yunfu Song. Joint stochastic approximation and its application to learning discrete latent variable models. In Jonas Peters and David Sontag, editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 929–938. PMLR, 03–06 Aug 2020.

Jaewoo Park, Won Chang, and Boseung Choi. An interaction Neyman–Scott point process model for coronavirus disease-19. *Spatial Statistics*, 47:100561, 2022. ISSN 2211-6753. doi: https://doi.org/10.1016/j.spasta.2021.100561. URL `https://www.sciencedirect.com/science/article/pii/S2211675321000671`.

Zhen Qin and Christian R Shelton. Auxiliary Gibbs sampling for inference in piecewise-constant conditional intensity models. In *UAI*, pages 722–731, 2015.

Rajesh Ranganath, Linpeng Tang, Laurent Charlin, and David Blei. Deep exponential families. In *Artificial intelligence and statistics*, pages 762–771. PMLR, 2015.

Vinayak Rao and Yee Whye Teh. Fast MCMC sampling for Markov jump processes and continuous time Bayesian networks. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence, UAI 2011*, 2011.

Vinayak Rao and Yee Whye Teh. Fast MCMC sampling for Markov jump processes and extensions. *The Journal of Machine Learning Research*, 14(1):3295–3320, 2013.

Gareth O. Roberts and Jeffrey S. Rosenthal. Harris recurrence of Metropolis-within-Gibbs and trans-dimensional Markov chains. *The Annals of Applied Probability*, 16(4):2123–2139, 2006. ISSN 10505164.

Walter Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1976.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1): 61–80, 2008.

Christian R Shelton, Zhen Qin, and Chandini Shetty. Hawkes process inference with missing data. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

R.S. Stoica, E. Tempel, L.J. Liivamägi, G. Castellan, and E. Saar. Spatial patterns analysis in cosmology based on marked point processes. *European Astronomical Society Publications Series*, 66:197–226, 2014. doi: 10.1051/eas/1466013.

Rasmus Waagepetersen and Yongtao Guan. Two-step estimation for inhomogeneous spatial point processes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 71(3):685–702, 2009.

Yixin Wang, Anthony Degleris, Alex Williams, and Scott W. Linderman. Spatiotemporal clustering with Neyman-Scott processes via connections to Bayesian nonparametric mixture models. *Journal of the American Statistical Association*, 0(ja):1–27, 2023. doi: 10.1080/01621459.2023.2257896. URL `https://doi.org/10.1080/01621459.2023.2257896`.

Alex Williams, Anthony Degleris, Yixin Wang, and Scott Linderman. Point process models for sequence detection in high-dimensional neural spike trains. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14350–14361. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/a5481cd6d7517aa3fc6476dc7d9019ab-Paper.pdf`.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive Hawkes process. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11183–11193. PMLR, 13–18 Jul 2020.

Qingyuan Zhao, Murat A Erdogdu, Hera Y He, Anand Rajaraman, and Jure Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1513–1522, 2015.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer Hawkes process. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11692–11702. PMLR, 13–18 Jul 2020.