Kronecker-Structured Nonparametric Spatiotemporal Point Processes

Zhitong Xu[1] **Qiwei Yuan**[1] **Yinghao Chen**[1] **Yan Sun**[2] **Bin Shen**[3] **Shandian Zhe**[1]

[1]Kahlert School of Computing, The University of Utah
[2]College of Arts & Sciences, Utah State University
[3]Celonis AI

Abstract

Events in spatiotemporal domains arise in numerous real-world applications, where uncovering event relationships and enabling accurate prediction are central challenges. Classical Poisson and Hawkes processes rely on restrictive parametric assumptions that limit their ability to capture complex interaction patterns, while recent neural point process models increase representational capacity but integrate event information in a black-box manner, hindering interpretable relationship discovery. To address these limitations, we propose a Kronecker-Structured Nonparametric Spatiotemporal Point Process (KSTPP) that enables transparent event-wise relationship discovery while retaining high modeling flexibility. We model the background intensity with a spatial Gaussian process (GP) and the influence kernel as a spatiotemporal GP, allowing rich interaction patterns including excitation, inhibition, neutrality, and time-varying effects. To enable scalable training and prediction, we adopt separable product kernels and represent the GPs on structured grids, inducing Kronecker-structured covariance matrices. Exploiting Kronecker algebra substantially reduces computational cost and allows the model to scale to large event collections. In addition, we develop a tensor-product Gauss-Legendre quadrature scheme to efficiently evaluate intractable likelihood integrals. Extensive experiments demonstrate the effectiveness of our framework. The code is released at `https://github.com/BayesianAIGroup/KSTPP`.

1 INTRODUCTION

Spatiotemporal events arise in many real-world domains, including weather dynamics, traffic accidents, natural disasters, epidemics, and population migration. Modeling such events for relationship discovery and predictive analysis is crucial. Understanding interactions among events facilitates uncovering the underlying mechanisms driving these phenomena, while accurate prediction enables risk monitoring, early warning, and timely intervention.

Existing spatiotemporal point process models — though widely used — face several limitations. Classical Poisson processes assume event independence and thus ignore mutual influences. Hawkes processes [Hawkes, 1971] introduce self-excitation via triggering effects from past events but typically rely on parametric kernels (e.g., exponential forms), which restrict their ability to capture diverse temporal patterns and inhibitory interactions.

At the other extreme, recent neural point process models directly parameterize the conditional intensity using deep architectures. For example, Neural Hawkes processes [Mei and Eisner, 2017] and Recurrent Marked Point Processes [Du et al., 2016] encode event histories via recurrent neural networks; Neural Spatial Temporal Point Processes [Chen et al., 2021] and Neural Jump Stochastic Differential Equations [Jia and Benson, 2019] incorporate continuous latent dynamics; and Transformer Hawkes Processes [Zuo et al., 2020] and Self-Attentive Hawkes Processes [Zhang et al., 2020] leverage transformer-based encoders. Although these approaches substantially enhance representational capacity, event interactions are encoded implicitly within latent states, hindering explicit and interpretable relationship discovery.

To address these limitations, we propose KSTPP, a Kronecker-Structured Nonparametric Spatiotemporal Point Process. Our framework enables transparent and explicit discovery of event-wise relationships while retaining high modeling flexibility to capture complex interaction patterns,

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:7594–7612.

including excitation, inhibition, neutrality, and time-varying effects. Our main contributions are summarized as follows:

- **Model.** We model the background intensity with a spatial Gaussian process (GP) and the influence kernel as a spatiotemporal GP. The conditional intensity is defined as the superposition of the background intensity and the aggregated influence from past events, followed by a positive link function. This formulation flexibly captures spontaneous occurrences and heterogeneous interaction patterns while maintaining explicit and interpretable representations of event relationships.
- **Algorithm.** To enable efficient maximum likelihood training and predictive inference, we employ separable product kernels and represent each GP on structured grids using inducing points, inducing Kronecker-structured covariance matrices. By exploiting Kronecker algebra, covariance operations decompose across input dimensions, substantially reducing computational complexity and enabling scalability to large event collections. To address the intractable integrals arising in likelihood evaluation and predictive density computation, we further develop a tensor-product Gauss-Legendre quadrature scheme. Leveraging the same Kronecker structure, we efficiently evaluate the GPs on structured quadrature grids, enabling tractable numerical evaluation of the required intensity integrals.
- **Experiments.** Experiments on three real-world benchmark datasets show that our method consistently outperforms state-of-the-art neural point process models in next-event prediction and achieves competitive performance relative to diffusion-based generative approaches. Synthetic experiments demonstrate accurate recovery of the underlying intensity functions and interaction patterns. Furthermore, analysis on a real-world earthquake dataset reveals meaningful and interpretable influence structures. Extensive ablation studies demonstrate the effectiveness and robustness of our approach with respect to its key design choices.

2 PRELIMINARIES

Spatiotemporal point processes naturally extend temporal point processes [Daley and Vere-Jones, 2008]. A spatiotemporal point process models random events occurring over a temporal domain $[0, T]$ and a spatial domain $\mathcal{S} \subset \mathbb{R}^2$. An observed event sequence is represented as $\Gamma = \{(t_n, \mathbf{s}_n)\}_{n=1}^N$, where $0 < t_1 < \cdots < t_N \leq T$ denote the event times and each $\mathbf{s}_n = (x_n, y_n) \in \mathcal{S}$ denotes the spatial location of the n-th event.

In this work, we assume a rectangular spatial domain,

$$\mathcal{S} = [a_x, b_x] \times [a_y, b_y], \tag{1}$$

although the framework naturally extends to three-dimensional spatial domains when needed. The process is characterized by its conditional intensity function $\lambda(t, \mathbf{s} \mid \mathcal{H}_t)$, defined through the infinitesimal event probability,

$$\begin{aligned}&\mathbb{P}\big(\text{an event occurs in } [t, t+dt) \times \mathrm{d}\mathbf{s} \mid \mathcal{H}_t\big)\\ &= \lambda(t, \mathbf{s} \mid \mathcal{H}_t)\, \mathrm{d}t\, \mathrm{d}\mathbf{s},\end{aligned}$$

where $\mathrm{d}\mathbf{s}$ denotes an infinitesimal spatial area element and $\mathcal{H}_t$ represents the history of events prior to time t, i.e.,

$$\mathcal{H}_t = \{(t_n, \mathbf{s}_n) | t_n < t\}. \tag{2}$$

Intuitively, $\lambda(t, \mathbf{s} \mid \mathcal{H}_t)$ corresponds to the instantaneous event rate at time t and location $\mathbf{s}$ given the past history. Under standard regularity conditions, the log-likelihood of observing Γ is given by

$$\sum\nolimits_{n=1}^{N} \log \lambda(t_n, \mathbf{s}_n \mid \mathcal{H}_{t_n}) - \int_0^T \int_{\mathcal{S}} \lambda(t, \mathbf{s} \mid \mathcal{H}_t)\, \mathrm{d}\mathbf{s}\, \mathrm{d}t.$$

3 METHODOLOGY

3.1 MODEL

To enable explicit event-wise relationship discovery while retaining high flexibility to capture complex interaction patterns, we model the conditional intensity as

$$\begin{aligned}&\lambda(t, x, y \mid \mathcal{H}_t) \\ &= \sigma\left(g(x, y) + \sum_{t_n < t} f(t - t_n, x - x_n, y - y_n)\right)\end{aligned} \tag{3}$$

where $g(x, y)$ denotes a latent spatial baseline function capturing spontaneous event occurrences across locations, and $f(\Delta t, \Delta x, \Delta y)$ denotes the influence kernel that characterizes how each past event affects the intensity at (t, x, y).

Unlike classical Hawkes processes, we do not restrict the influence kernel to be nonnegative. Our formulation allows f to take arbitrary values: $f > 0$ corresponds to excitation, $f < 0$ corresponds to inhibition, and $f \approx 0$ indicates negligible (neutral) influence. Because f is defined over temporal and spatial differences, it flexibly models how interaction strength evolves across both time and space.

To ensure positivity of the conditional intensity, we apply a positive link function $\sigma(\cdot)$ to the superposition of the baseline and influence terms. In this work, we use the SoftPlus function, $\sigma(z) = \frac{1}{\beta} \log\big(1 + e^{\beta z}\big)$,

where $\beta > 0$ controls the sharpness of the transformation.

To flexibly estimate g and f, we place Gaussian process (GP) priors [Williams and Rasmussen, 2006] on both functions:

$$\begin{aligned}g(x, y) &\sim \mathcal{GP}\left(0, \rho(\cdot, \cdot)\right),\\ f(\Delta t, \Delta x, \Delta y) &\sim \mathcal{GP}\left(0, \kappa(\cdot, \cdot)\right),\end{aligned} \tag{4}$$

where ρ and κ are covariance (kernel) functions for g and f, respectively.

3.2 ALGORITHM

Due to the GP priors over g and f, the joint distribution of their values evaluated at observed event times and locations, and at all auxiliary points required for likelihood evaluation (e.g., those for the integral terms) follows a multivariate Gaussian distribution with large covariance matrices defined by ρ and κ. Direct training and inference are therefore computationally prohibitive, requiring $\mathcal{O}(\overline{N}^3)$ time and $\mathcal{O}(\overline{N}^2)$ memory complexity, where $\overline{N}$ denotes the total number of function evaluations involved. The computational cost prevents the model from handling large-scale event datasets.

Kronecker-structured inducing representation. To overcome this challenge, we construct separable product kernels for both g and f:

$$\begin{aligned}&\rho\left((x,y),(x',y')\right)=\rho_1(x,x')\cdot\rho_2(y,y'),\\&\kappa\left((\Delta_t,\Delta_x,\Delta_y),(\Delta_t',\Delta_x',\Delta_y')\right)=\\&\kappa_0(\Delta_t,\Delta_t')\cdot\kappa_1(\Delta_x,\Delta_x')\cdot\kappa_2(\Delta_y,\Delta_y').\end{aligned}\tag{5}$$

The widely used squared exponential (SE) kernel is already separable. More generally, different kernels can be chosen along each dimension and multiplied together. This construction corresponds to performing high-dimensional feature mappings along each dimension and taking their tensor product in the latent feature space.

Next, we introduce a structured grid of inducing points. Taking f as an example, we define the mesh $\mathcal{M}_f = \boldsymbol{\gamma}_0 \times \boldsymbol{\gamma}_1 \times \boldsymbol{\gamma}_2 \subset [0,T]\times[a_x,b_x]\times[a_y,b_y]$, where each $\boldsymbol{\gamma}_k = \{z_1^k,\dots,z_{m_k}^k\}$ contains m_k points along dimension k. Let $\mathcal{F}$ denote the tensor of function values of f evaluated on $\mathcal{M}_f$. Since $f\sim\mathcal{GP}$, $\mathrm{vec}(\mathcal{F})$ follows a multivariate Gaussian prior. Under the separable kernel construction (5), the covariance matrix admits a Kronecker product structure:

$$p(\mathcal{F})=\mathcal{N}(\mathrm{vec}(\mathcal{F})\mid\mathbf{0},\mathbf{K}_0\otimes\mathbf{K}_1\otimes\mathbf{K}_2),\tag{6}$$

where $\mathbf{K}_i=\kappa_i(\boldsymbol{\gamma}_i,\boldsymbol{\gamma}_i)$ for $i=0,1,2$. Exploiting Kronecker algebra [Kolda, 2006], the log prior decomposes as:

$$\begin{aligned}\log p(\mathcal{F})=&-\frac{1}{2}\sum\nolimits_{i=0}^{2}\frac{m}{m_i}\log|\mathbf{K}_i|\\&-\frac{1}{2}\mathrm{vec}(\mathcal{F})^\top\mathrm{vec}(\mathcal{F}\times_0\mathbf{K}_0^{-1}\times_1\mathbf{K}_1^{-1}\times_2\mathbf{K}_2^{-1}),\end{aligned}\tag{7}$$

where $m=\prod_{k=0}^{2}m_k$ is the total number of mesh points, and $\times_i$ denotes the tensor-matrix multiplication at mode i.

Given $\mathcal{F}$, we evaluate f at arbitrary inputs via GP conditional mean (interpolation):

$$\begin{aligned}&f(\Delta t,\Delta x,\Delta y)=\kappa_0(\Delta t,\boldsymbol{\gamma}_0)\kappa_1(\Delta x,\boldsymbol{\gamma}_1)\kappa_2(\Delta y,\boldsymbol{\gamma}_2)\\&\cdot(\mathbf{K}_0\otimes\mathbf{K}_1\otimes\mathbf{K}_2)^{-1}\mathrm{vec}(\mathcal{F})\\&=\mathcal{F}\times_0\boldsymbol{\eta}_0\times_1\boldsymbol{\eta}_1\times_2\boldsymbol{\eta}_2,\end{aligned}\tag{8}$$

where $\boldsymbol{\eta}_0=\kappa_0(\Delta t,\boldsymbol{\gamma}_0)\mathbf{K}_0^{-1}$, $\boldsymbol{\eta}_1=\kappa_1(\Delta x,\boldsymbol{\gamma}_1)\mathbf{K}_1^{-1}$ and $\boldsymbol{\eta}_2=\kappa_2(\Delta y,\boldsymbol{\gamma}_2)\mathbf{K}_2^{-1}$. Importantly, neither (7) nor (8) requires explicit computation of the full $m\times m$ covariance matrix. All computations are confined to the per-dimensional kernel matrices $\{\mathbf{K}_i\}$, reducing the time complexity from $\mathcal{O}(\prod_k m_k^3)$ to $\mathcal{O}(\sum_k m_k^3)$, and the memory complexity from $\mathcal{O}(\prod_k m_k^2)$ to $\mathcal{O}(\sum_k m_k^2)$. This enables our method to scale to large event datasets encountered in practice. The same construction applies to g, using a spatial mesh $\mathcal{M}_g=\mathbf{v}_1\times\mathbf{v}_2$ with corresponding tensor values $\mathcal{G}$.

Training objective. Given observed event sequences $\mathcal{D}$, the training maximizes the joint log probability:

$$\begin{aligned}&\log p(\mathcal{F},\mathcal{G},\mathcal{D})=\log p(\mathcal{F})+\log p(\mathcal{G})\\&+\sum\nolimits_{\Gamma\in\mathcal{D}}\log p(\Gamma|\mathcal{F},\mathcal{G}).\end{aligned}\tag{9}$$

For each sequence $\Gamma=\{(t_n,x_n,y_n)\}_{n=1}^{N}\in\mathcal{D}$, the log likelihood is

$$\begin{aligned}&\log p(\Gamma|\mathcal{F},\mathcal{G})=\sum\nolimits_{n=1}^{N}\log\lambda(t_n,x_n,y_n|\mathcal{H}_{t_n})\\&-\sum_{n=0}^{N}\int_{t_n}^{t_{n+1}}\int_{a_x}^{b_x}\int_{a_y}^{b_y}\lambda(t,x,y|\mathcal{H}_{t_{n+1}})\mathrm{d}t\mathrm{d}x\mathrm{d}y,\end{aligned}\tag{10}$$

where $t_0=0$ and $t_{N+1}=T$. Note that the temporal integral must be evaluated piecewise over each interval (t_n,t_{n+1}), since the event history updates immediately after each observed event, causing a discontinuous jump in the conditional intensity λ.

Tensor-product Gauss-Legendre quadrature. The integral terms in the likelihood (10) are intractable and do not admit closed-form solutions. While crude Monte Carlo approximation may suffice for stochastic training, reliable prediction of future events (see Section 3.3) requires high-accuracy evaluation of these integrals. Obtaining such accuracy with Monte Carlo would require a prohibitively large number of samples due to its inherent variance. To address this challenge, we propose a tensor-product Gauss-Legendre quadrature scheme for efficient and high-order numerical integration.

Specifically, for one-dimensional quadrature rules, the nodes and weights depend only on the integration interval and the chosen rule, and are independent of the integrand. This property allows us to compute the triple integral dimension by dimension using a tensor-product construction. In particular,

$$
\begin{aligned}
&\int_{t_n}^{t_{n+1}} \lambda(t,x,y|\mathcal{H}_{t_{n+1}})\mathrm{d}t\mathrm{d}x\mathrm{d}y \\
&\approx \sum_i w_i^0 \int_{a_x}^{b_x}\int_{a_y}^{b_y} \lambda(\hat{t}_i,x,y|\mathcal{H}_{t_{n+1}})\mathrm{d}x\mathrm{d}y \\
&\approx \sum_i w_i^0 \sum_j w_j^1 \int_{a_y}^{b_y} \lambda(\hat{t}_i,\hat{x}_j,y|\mathcal{H}_{t_{n+1}})\mathrm{d}y \\
&\approx \sum_i w_i^0 \sum_j w_j^1 \sum_k w_k^2 \lambda(\hat{t}_i,\hat{x}_j,\hat{y}_k|\mathcal{H}_{t_{n+1}}) \\
&= \sum_{i,j,k} w_i^0 w_j^1 w_k^2 \cdot \lambda(\hat{t}_i,\hat{x}_j,\hat{y}_k|\mathcal{H}_{t_{n+1}}), \qquad (11)
\end{aligned}
$$

where $\{w_i^0, \hat{t}_i\}$, $\{w_j^1, \hat{x}_j\}$, and $\{w_k^2, \hat{y}_k\}$ denote the quadrature nodes and weights along the temporal and spatial dimensions, respectively. Because the nodes and weights are separable across dimensions and independent of λ, the multi-dimensional integral reduces to evaluating λ on the structured quadrature grid,

$$
\mathcal{Q} = \hat{\mathbf{t}} \times \hat{\mathbf{x}} \times \hat{\mathbf{y}},\ \hat{\mathbf{t}} = \{\hat{t}_i\},\ \hat{\mathbf{x}} = \{\hat{x}_j\},\ \hat{\mathbf{y}} = \{\hat{y}_k\},
$$

followed by a tensor-weighted inner product with the separable weight tensor $\{w_i^0 w_j^1 w_k^2\}_{i,j,k}$.

Evaluating λ on the quadrature grid $\mathcal{Q}$ requires computing the baseline component g and the influence kernel f at all grid locations. This can be carried out efficiently by leveraging the Kronecker structure again. In particular, $f(\mathcal{Q}) = \mathcal{F} \times_0 \kappa_0(\hat{\mathbf{t}}, \boldsymbol{\gamma}_0)\mathbf{K}_0^{-1} \times_1 \kappa_1(\hat{\mathbf{x}}, \boldsymbol{\gamma}_1)\mathbf{K}_1^{-1} \times_2 \kappa_2(\hat{\mathbf{y}}, \boldsymbol{\gamma}_2)\mathbf{K}_2^{-1}$, and the evaluation of g on $\mathcal{Q}$ follows analogously using its spatial mesh.

We adopt Gauss-Legendre rules along each dimension due to their high accuracy for smooth integrands, typically requiring only a small number of nodes. For a general interval $[a, b]$, the quadrature nodes and weights are obtained via a linear transformation of the standard rule on $[-1, 1]$: $\hat{z}_k = \frac{b-a}{2}\xi_k + \frac{b+a}{2}, w_k = \frac{b-a}{2}\alpha_k$, where $\{\xi_k\}$ and $\{\alpha_k\}$ denote the standard Gauss-Legendre nodes and weights on $[-1, 1]$.

Combining the Kronecker-structured GP representation with the tensor-product quadrature scheme allows efficient computation of both the log prior and the log likelihood for each event sequence in (9). Training is performed using stochastic mini-batch optimization over randomly sampled sequences to estimate $\mathcal{F}$, $\mathcal{G}$, and kernel parameters. The full model pipeline is summarized in Appendix Figure 8 and Section D.

Computational Complexity. The time complexity for processing a mini-batch of B sequences, each of length N, is

$$
\mathcal{O}\left(\sum_{k=0}^{2} m_k^3 + \sum_{k=1}^{2} v_k^3 + BN\left(m(\sum_{i=0}^{2} q_i) + v(\sum_{i=1}^{2} q_i)\right)\right),
$$

where $m = \prod_{k=0}^{2} m_k$ denotes the total number of mesh points in $\mathcal{M}_f$, $v = \prod_{k=1}^{2} v_k$ denotes the total number of mesh points in $\mathcal{M}_g$, and q_i is the number of quadrature nodes along each dimension i. The memory complexity is

$$
\mathcal{O}(\sum_{k=0}^{2} m_k^2 + \sum_{k=1}^{2} v_k^2 + m + v + BNq),
$$

where $q = \prod_i q_i$ is the total number of nodes in the tensor-product grid $\mathcal{Q}$. The dominant memory cost arises from storing the per-dimension kernel matrices and maintaining function evaluations at mesh points, quadrature nodes, and observed events. Overall, the cubic terms depend only on per-dimension mesh sizes rather than the total number of events, enabling scalable mini-batch training and inference.

3.3 PREDICTION

Given a sequence of N observed events $\mathcal{H} = \{(t_1, x_1, y_1), \ldots, (t_N, x_N, y_N)\}$, we aim to predict the time and location of the next event $(t_{N+1}, x_{N+1}, y_{N+1})$.

Predicting the next event time. The conditional density of the next arrival time is given by the standard point process formulation:

$$
p(t_{N+1} \mid \mathcal{H}) = \lambda(t_{N+1}|\mathcal{H}) \exp\left(-\int_{t_N}^{t_{N+1}} \lambda(t|\mathcal{H})\mathrm{d}t\right),
$$

where the temporal marginal intensity is

$$
\lambda(t|\mathcal{H}) = \int_{a_x}^{b_x}\int_{a_y}^{b_y} \lambda(t,x,y|\mathcal{H})\mathrm{d}x\mathrm{d}y. \qquad (12)
$$

We use the posterior mean as the point prediction of t_{N+1}. Let $\tau = t_{N+1} - t_N > 0$. Then

$$
\mathbb{E}[t_{N+1}|\mathcal{H}] = t_N + \mathbb{E}[\tau|\mathcal{H}]. \qquad (13)
$$

Using integration by parts, the conditional expectation of τ can be written as

$$
\mathbb{E}[\tau|\mathcal{H}] = \int_0^\infty e^{-\Lambda(\tau)}\mathrm{d}\tau, \qquad (14)
$$

$$
\Lambda(\tau) = \int_0^\tau \lambda(t_N + u|\mathcal{H})\mathrm{d}u. \qquad (15)
$$

The evaluation of $\lambda(t \mid \mathcal{H})$ and the inner integral (15) is performed using the same tensor-product quadrature method described in Section 3.2.

To compute the improper integral in (14), we apply the transformation

$$
u = \frac{\tau}{1+\tau}, \quad \tau = \frac{u}{1-u},
$$

which maps $\tau \in (0, \infty)$ to $u \in (0, 1)$. This yields

$$
\mathbb{E}[\tau|\mathcal{H}] = \int_0^1 \frac{e^{-\Lambda(u/(1-u))}}{(1-u)^2}\mathrm{d}u. \qquad (16)
$$

We then apply Gauss-Legendre quadrature to evaluate (16). To further improve time prediction accuracy and mitigate approximation errors introduced by numerical quadrature, we optionally learn a separate MLP head for inter-arrival time prediction, similar in spirit to the approach of Zuo et al. [2020]. Specifically, we evaluate the integrand in (16) at the quadrature nodes and use these values as input to an MLP. A Softplus activation is applied to the network output to ensure positivity. This prediction head is trained separately after the main model has converged. Intuitively, the MLP learns a data-driven correction to the numerical quadrature, compensating for residual approximation errors that are difficult to capture analytically. Empirically, we find that this refinement consistently yields a modest improvement in next-event time prediction over the plain quadrature estimator in (16).

Predicting the event location. Given the predicted time t_{N+1}, the conditional spatial density is

$$p(x, y|t_{N+1}) = \frac{\lambda(t_{N+1}, x, y \mid \mathcal{H})}{\lambda(t_{N+1} \mid \mathcal{H})}.$$

We use the conditional expectations as point predictions:

$$\mathbb{E}[x|t_{N+1}] = \int_{a_x}^{b_x} \int_{a_y}^{b_y} x\, p(x, y|t_{N+1}) \mathrm{d}x\mathrm{d}y$$
$$\mathbb{E}[y|t_{N+1}] = \int_{a_x}^{b_x} \int_{a_y}^{b_y} y\, p(x, y|t_{N+1}) \mathrm{d}x\mathrm{d}y. \qquad (17)$$

These integrals are evaluated using the same tensor-product quadrature scheme described in Section 3.2.

4 RELATED WORK

A rich body of work has been devoted to temporal point processes. Early developments include Poisson processes [Lloyd et al., 2015] and their applications in tensor decomposition [Schein et al., 2015, 2016, 2019]. Hawkes processes (HPs) [Hawkes, 1971] subsequently gained significant attention due to their ability to capture mutual excitation among events, e.g., [Blundell et al., 2012, Du et al., 2015, Wang et al., 2017, Yang et al., 2017, Xu et al., 2018, Zhe and Du, 2018]. Recent works have extended classical Hawkes processes with more flexible influence kernels capable of modeling inhibitory effects [Pan et al., 2020, Wang et al., 2020]. Pan et al. [2021] proposed a nonparametric kernel that captures general temporal influence decay patterns beyond standard exponential decay. More recently, Xu et al. [2026] introduced a product-form neural kernel combining a signed interaction network with a delay-aware monotonic network, enabling flexible modeling of both temporal decay and delayed influences.

In parallel, conditional intensities have been modeled using deep neural architectures. Neural Hawkes Processes (NHP) [Mei and Eisner, 2017] encode event history via LSTM states [Hochreiter and Schmidhuber, 1997], with the intensity parameterized as a function of the hidden state. Recurrent Marked Temporal Point Processes (RMTPP) [Du et al., 2016] adopt a similar recurrent architecture while explicitly modeling event marks (types). Transformer-based approaches [Zhang et al., 2020, Zuo et al., 2020] treat each event as a token and use causal attention mechanisms to aggregate historical information, with the conditional intensity parameterized on top of token representations.

Extending classical Poisson and Hawkes processes to the spatiotemporal setting is conceptually straightforward. Recently, Chen et al. [2021] proposed a neural spatiotemporal point process model that, similar to NHP and RMTPP, models intensity jumps using recurrent neural networks. To enable continuous-time evolution between events, they incorporate neural ordinary differential equations (ODEs) [Chen et al., 2018]. In contrast, Jia and Benson [2019] model hidden state dynamics using neural stochastic differential equations (SDEs). Zhou et al. [2022], Zhou and Yu [2023] extended spatiotemporal Hawkes processes. The model of Zhou et al. [2022] preserves the parametric Hawkes structure while estimating the triggering kernel parameters via transformer encodings of historical events. The work of Zhou and Yu [2023] introduces a neural triggering kernel based on monotonic networks [Sill, 1997] allowing exact likelihood integration. However, both approaches remain within the Hawkes framework and are therefore limited to modeling excitatory interactions, making them unable to capture inhibitory event dynamics commonly observed in practice. More recently, Yuan et al. [2023] proposed bypassing intensity modeling entirely by using diffusion-based generative models [Ho et al., 2020] to directly generate event sequences. While demonstrating strong predictive performance, such approaches sacrifice the ability to explicitly model and calibrate the conditional intensity, which is critical for applications such as risk monitoring and survival analysis — central objectives in point process modeling.

The computational advantages of Kronecker product structures have been widely recognized in scalable Gaussian process and kernel methods [Saatcci, 2012, Wilson and Nickisch, 2015, Izmailov et al., 2018, Zhe et al., 2019]. For example, Xu et al. [2012], Zhe et al. [2016] exploited Kronecker structure in nonparametric tensor factorization models. More recently, physics-informed machine learning methods [Fang et al., 2024, Xu et al., 2025] have leveraged Kronecker structure for efficient nonlinear PDE solving. To the best of our knowledge, our work is the first to integrate Kronecker-structured Gaussian process representations into spatiotemporal point process modeling, enabling scalable training and flexible intensity modeling.

5 EXPERIMENTS

5.1 SYNTHETIC DATA

We first evaluated KSTPP on synthetic datasets designed to validate its ability to capture both excitation and inhibition effects. We constructed two spatiotemporal point processes, each incorporating excitation and inhibition mechanisms driven by past events. The conditional intensity follows a form resembling a spatiotemporal Hawkes process:

$$\lambda(t, x, y|\mathcal{H}_t) = \lambda_0 + \sum_{t_n<t} c_n e^{-\beta(t-t_n)} \frac{1}{2\pi\sigma^2} e^{-\frac{d_n^2}{2\sigma^2}} \quad (18)$$

where $d_n = \sqrt{(x-x_n)^2 + (y-y_n)^2}$ and $\beta, \sigma > 0$. In the first process, denoted as **SYN1**, the influence strength c_n depends on the temporal lag. When $t - t_n < 1$, we set $c_n = 1.0$ to induce excitation; when $t - t_n \geq 1$, we set $c_n = -2.0$, introducing inhibitory effects. The decay parameter and spatial bandwidth were set to $\beta = 2.0$ and $\sigma = 0.3$, respectively. In the second process, denoted as **SYN2**, the interaction type depends on spatial distance. When $d_n > 1.0$, we set $c_n = 1.0$, enabling excitation from distant events. When $d_n < 1.0$, we set $c_n = -0.3$, modeling local inhibition. The other parameters were set to $\beta = 1.5$ and $\sigma = 0.5$. For both processes, the time horizon was set to $T = 50$ with base rate $\lambda_0 = 2$. We generated 2,300 sequences for training, 100 sequences for validation, and another 100 sequences for testing, using Ogata's thinning algorithm [Ogata, 1981].

We compared KSTPP against several popular and state-of-the-art point process models. (1) Spatiotemporal Hawkes Process (STHP): adopts the same conditional intensity form as in (18) but restricts all parameters to be positive (excitation-only). Parameters are optimized in the log domain to enforce positivity. (2) Neural Spatiotemporal Point Process (NSTPP) [Chen et al., 2021]: integrates historical events via RNN states, modeling continuous-time dynamics with neural ODEs and event-triggered updates through GRU-style gating. (3) Deep Spatiotemporal Point Process (DeepSTPP) [Zhou et al., 2022]: retains a parametric excitation-only Hawkes intensity while using a transformer encoder to estimate kernel parameters. We also include two neural temporal point process models: (4) Neural Hawkes Process (NHP) [Mei and Eisner, 2017], which encodes event history using a continuous-time LSTM; and (5) Transformer Hawkes Process (THP) [Zuo et al., 2020], which applies a transformer to model temporal dependencies.

Our method was implemented in PyTorch and trained using the Adam optimizer with a learning rate of 10^{-3}. The mini-batch size was set to one. We set $\beta = 1$ in the SoftPlus transformation. We used 12 Gauss-Legendre quadrature nodes per dimension and adopted the squared exponential (SE) kernel for the influence function f. STHP was trained using Adam with the same learning rate. For the remaining baselines, we used the official open-source implementations and default hyperparameter settings.

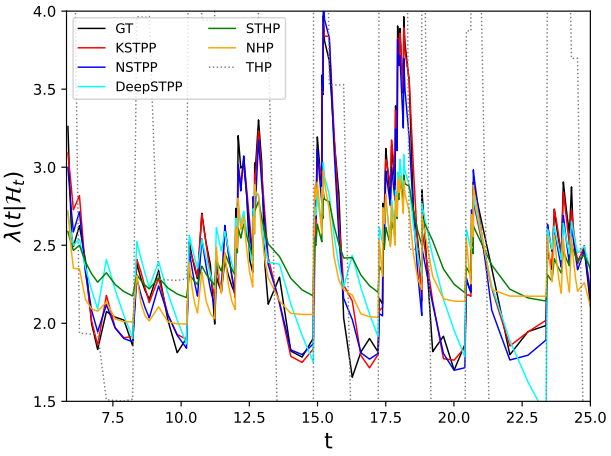

Figure 1: Temporal conditional intensity on example test sequence from **SYN1**.

Table 1: Relative L_2 error of the learned marginal intensity $\lambda(t|\mathcal{H}_t)$ on synthetic datasets. The smallest error is shown in bold.

	SYN1	**SYN2**
STHP	1.45e-01 $\pm_{2.41e-02}$	8.42e-02 $\pm_{1.87e-02}$
NSTPP	5.57e-02 $\pm_{6.50e-03}$	2.99e-02 $\pm_{4.78e-03}$
DeepSTPP	1.25e-01 $\pm_{1.89e-02}$	7.77e-02 $\pm_{1.52e-02}$
NHP	1.35e-01 $\pm_{3.08e-02}$	2.34e-02 $\pm_{4.37e-03}$
THP	8.14e-01 $\pm_{7.10e-02}$	1.91e-01 $\pm_{2.87e-02}$
KSTPP	**4.44e-02** $\pm_{3.96e-03}$	**2.00e-02** $\pm_{3.50e-03}$

Intensity Recovery. We first examined whether each method could recover the ground-truth intensity. Since NHP and THP are designed for purely temporal point processes, we compared the marginal conditional intensity $\lambda(t \mid \mathcal{H}_t)$ across all methods. Specifically, for each test sequence, we evaluated the conditional intensity at every observed event time as well as at three equally spaced time points between successive events. For each sequence, we computed the relative L_2 error with respect to the ground-truth intensity, and report the mean and standard deviation across all test sequences. As shown in Table 1, KSTPP consistently achieves

Table 2: Relative L_2 error of the learned spatiotemporal intensity $\lambda(t, x, y|\mathcal{H}_t)$ on synthetic datasets.

	SYN1	**SYN2**
STHP	1.77e-01 $\pm_{7.44e-02}$	1.23e-01 $\pm_{4.73e-02}$
NSTPP	8.68e-01 $\pm_{9.41e-03}$	8.64e-01 $\pm_{3.26e-03}$
DeepSTPP	3.78e-01 $\pm_{3.65e-02}$	3.87e-01 $\pm_{2.47e-02}$
KSTPP	**6.52e-02** $\pm_{4.80e-03}$	**3.67e-02** $\pm_{1.43e-02}$

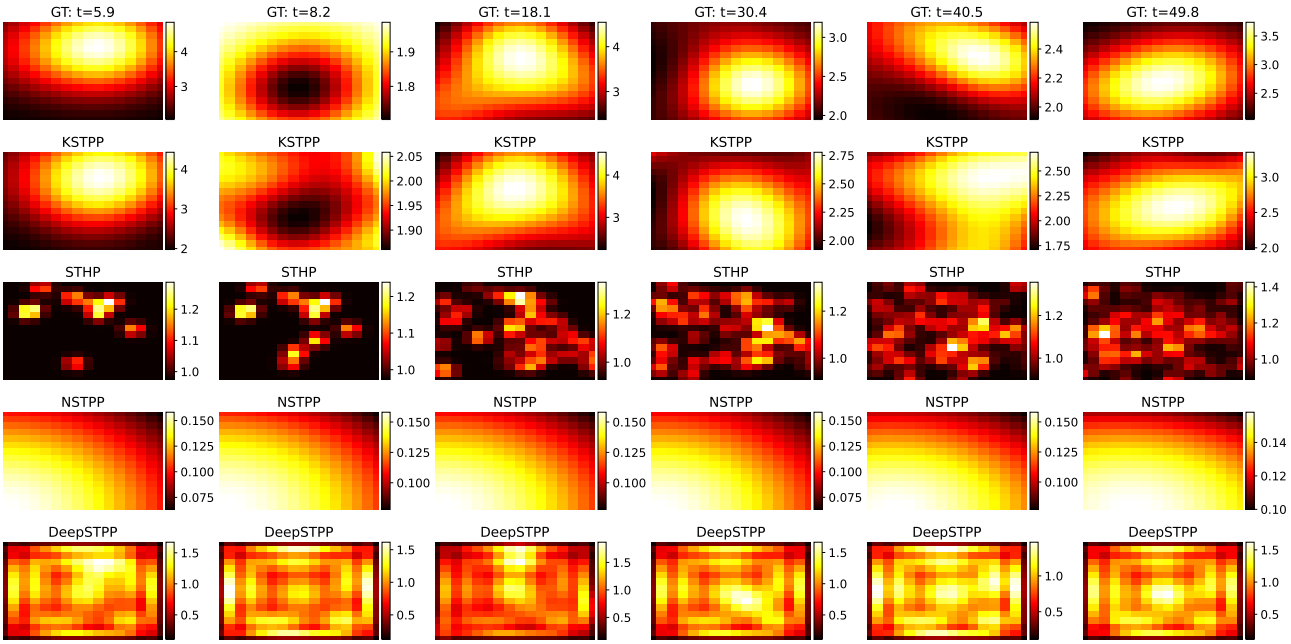

Figure 2: Spatial conditional intensity $p(x, y \mid t, \mathcal{H}_t) = \lambda(t, x, y \mid \mathcal{H}_t)/\lambda(t \mid \mathcal{H}_t)$ on an example test sequence from **SYN1**. GT denotes the ground-truth.

the lowest relative L_2 error, indicating superior intensity recovery. Figure 1 and Appendix Figure 5 visualize the recovered temporal intensity for an example test sequence from each synthetic dataset. In **SYN1**, the intensity curves estimated by STHP and DeepSTPP roughly capture the overall shape of the ground truth, albeit with noticeable inaccuracies. However, in **SYN2**, their estimates deviate substantially from the true intensity. This difference may be explained by the strength of inhibition in the two datasets. In **SYN1**, the inhibition effect is relatively weak — it appears only when $t > 1$ in (18), and the magnitude of the inhibitory kernel is small. In contrast, **SYN2** exhibits strong local inhibition within a small spatial range ($d < 1$). Since STHP and DeepSTPP rely on fixed kernel forms, model misspecification under strong inhibition can lead to degraded intensity estimates. NHP and NSTPP demonstrate strong approximation capability, reflecting their expressive modeling capacity. In contrast, THP exhibits substantial deviations from the ground-truth intensity. Our method consistently produces intensity estimates that closely match the ground-truth across both datasets. Moreover, unlike excitation-only models, KSTPP is capable of identifying both excitation and inhibition effects, as demonstrated later in the influence kernel estimation results.

We next evaluate recovery of the spatial conditional intensity. For each dataset (**SYN1** and **SYN2**), we randomly select a test sequence and examine the spatial conditional intensity at several representative time points (Figure 2 and Appendix Figure 6). As shown, KSTPP closely matches both the shape and magnitude of the ground-truth spatial intensity. In contrast, STHP, NSTPP, and DeepSTPP exhibit noticeable deviations and fail to recover key spatial structures. In particular, STHP and DeepSTPP produce multiple spurious local modes, resembling mixture-like densities. This behavior may stem from their use of additive, strictly positive parametric triggering kernels. Although NSTPP yields smoother spatial intensity estimates, its recovered structure and scale do not align well with the ground truth.

Table 2 reports the average relative L_2 error of the full spatiotemporal intensity evaluated on a 16×16 uniform spatial grid. KSTPP achieves the lowest error across both datasets, quantitatively confirming its improved intensity recovery. Overall, these results highlight the advantage of KSTPP in recovering both the temporal and spatial components of the underlying intensity.

Influence Kernel Estimation. We then evaluated the learned influence kernel f produced by KSTPP. For both **SYN1** and **SYN2**, we select three representative time lags and visualize the corresponding learned kernels. For comparison, we also report the kernel estimated by STHP. As shown in Figure 3 and Appendix Figure 7, KSTPP effectively recovers both excitation and inhibition patterns. Because our model applies a SoftPlus transformation to enforce nonnegativity of the conditional intensity, the learned kernel — under this nonlinear link — does not exactly coincide with the ground-truth kernel specified under a purely linear additive formulation. Nevertheless, the key structural characteristics, including the sign, modal location, and spatial spread of the influence, are well preserved. In contrast, although STHP adopts an additive formulation consistent

Table 3: Root-mean-square error (RMSE) and Euclidean distance for next-event time and location prediction. The best two results are highlighted in bold.

Model	Earthquake		COVID-19		Citibike	
	Spatial ↓	Temporal ↓	Spatial ↓	Temporal ↓	Spatial ↓	Temporal ↓
RMTPP	-	0.424±0.009	-	1.32±0.024	-	2.07±0.015
NHP	-	1.86±0.023	-	2.13±0.100	-	2.36±0.056
THP	-	2.44±0.021	-	0.611±0.008	-	1.46±0.009
Poisson	9.45±0.000	0.412±0.000	0.818±0.000	0.113±0.000	0.452±0.000	0.239±0.000
STHP	8.35±0.252	0.424±0.018	0.422±0.000	**0.100**±0.001	0.032±0.000	0.633±0.126
NJSDE	9.98±0.024	0.465±0.009	0.641±0.009	0.137±0.001	0.707±0.001	0.264±0.005
NSTPP	8.11±0.000	0.547±0.010	0.560±0.000	0.145±0.002	0.705±0.000	0.355±0.013
DeepSTPP	9.20±0.000	**0.341**±0.000	0.687±0.000	0.197±0.000	0.044±0.000	0.234±0.000
DSTPP	**6.77**±0.193	0.375±0.001	**0.419**±0.001	**0.093**±0.000	**0.031**±0.000	**0.200**±0.002
KSTPP (Ours)	**6.72**±0.014	**0.372**±0.000	**0.392**±0.000	**0.100**±0.000	**0.031**±0.000	**0.206**±0.000

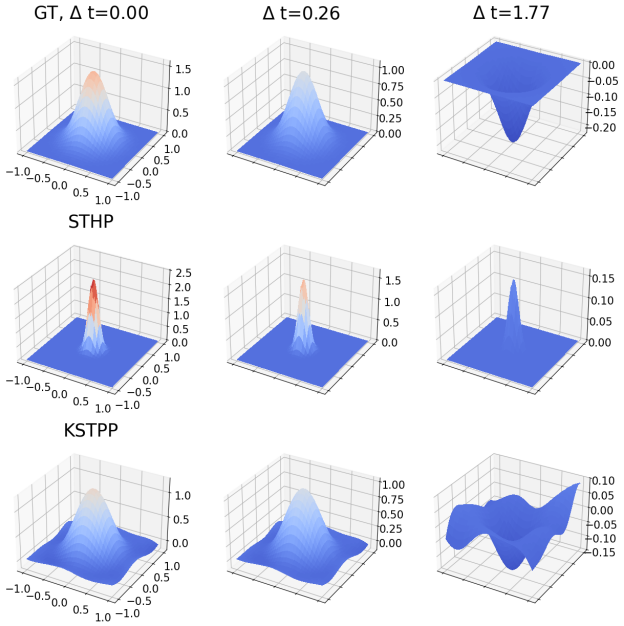

Figure 3: Learned influence kernel from **SYN1**.

with the data-generating process, it is restricted to excitation-only mechanisms. Consequently, its learned kernel exhibits substantial deviation from the ground truth, particularly in regions where inhibitory effects dominate.

5.2 PREDICTIVE PERFORMANCE

Next, we evaluate the predictive performance of KSTPP on three real-world benchmark datasets: **Earthquake** [Chen et al., 2021], **Covid-19** [Chen et al., 2021], and **Citibike** [Yuan et al., 2023]. These datasets cover seismic activity, epidemic spread, and urban mobility, providing diverse spatiotemporal event dynamics. Detailed descriptions, data splits, and statistics are provided in Appendix B.

In addition to the methods introduced in Section 5.1, we included: (6) Diffusion Spatiotemporal Point Process (DSTPP) [Yuan et al., 2023], a recent diffusion-based generative model that directly generates event sequences without explicitly modeling the conditional intensity. (7) Neural Jump Stochastic Differential Equations (NJSDE) [Jia and Benson, 2019], which summarizes event history into latent states and models state dynamics via neural SDEs with jumps induced by event arrivals. (8) Homogeneous Poisson Process (PP) as a classical baseline. For KSTPP, the covariance functions for g and f were selected from either the SE kernel or Matérn kernel with smoothness parameter $\nu = 5/2$. The number of quadrature nodes per dimension was chosen from $\{8, 12, 16\}$. We further employed a lightweight two-layer MLP prediction head with ReLU or GELU activations for next-event time prediction. Full implementation details and hyperparameter choices are provided in Appendix C.

We adopted the same training, validation, and test splits as provided in [Yuan et al., 2023] for all three datasets. Following [Chen et al., 2021, Yuan et al., 2023], each experiment was repeated five times with different random initializations. We evaluated: Root-Mean-Square Error (RMSE) for next-event time prediction, and Euclidean distance for next-event location prediction. We report the mean and standard deviation across runs in Table 3. For baselines other than PP and STHP, results were directly taken from [Yuan et al., 2023]. Although we conducted additional hyperparameter tuning for these methods, we were unable to outperform the reported results. To ensure fairness, we therefore compare against the optimized results reported in [Yuan et al., 2023].

As shown in Table 3, KSTPP consistently achieves top performance in both event time and location prediction. Its predictive accuracy is comparable to the diffusion-based model DSTPP and often outperforms other neural point process methods by a substantial margin. In particular, KSTPP achieves the highest location prediction accuracy across all datasets and the second-best performance in time prediction. These results demonstrate that, although KSTPP adopts a

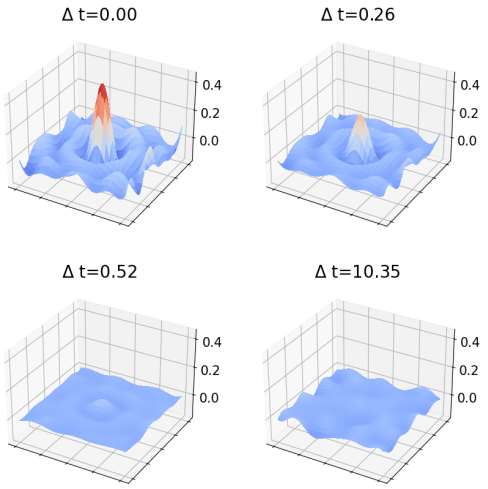

Figure 4: Learned influence kernel by KSTPP on **Earthquake**.

more structured modeling framework aimed at improved event relationship discovery, it maintains strong predictive performance that is competitive with, and in many cases superior to, existing neural point process models.

5.3 PATTERN DISCOVERY

Finally, we investigated the learned influence kernel of KSTPP on the real-world Earthquake dataset to examine whether the model reveals meaningful spatiotemporal interaction patterns.

We visualize the learned kernel $f(\Delta t, \Delta x, \Delta y)$ at four representative time lags, $\Delta t \in \{0, 0.26, 0.52, 10.35\}$, across the spatial domain. As shown in Figure 4, when Δt is small, the influence exhibits strong excitation in nearby regions (i.e., small $|\Delta x|$ and $|\Delta y|$). As the time lag increases, the overall magnitude of the influence decays. When $\Delta t \geq 0.52$, the kernel values across spatial distances approach zero. This behavior is consistent with well-established seismic dynamics: earthquakes tend to trigger short-term aftershocks in spatially proximate regions, while such triggering effects decay over time, as described by the Omori-Utsu law and ETAS models [Utsu et al., 1995, Ogata, 1988]. Interestingly, the learned kernel also reveals localized inhibition effects at small time lags. Specifically, at certain spatial offsets, the kernel takes negative values, indicating reduced intensity in more distant regions immediately following a seismic event. This pattern qualitatively aligns with stress redistribution effects (so-called stress shadows), which can lead to relative suppression of seismicity in specific areas [King et al., 1994, Harris, 1998]. Such effects cannot be captured by traditional Hawkes process models that restrict interactions to purely excitatory kernels. Overall, these findings suggest that our model is capable of uncovering nuanced excitation-inhibition structures and extracting interpretable spatiotemporal interaction patterns from real-world data.

5.4 ADDITIONAL ANALYSES

Beyond the results above, we provided several additional analyses in the Appendix. We conducted ablation studies on the quadrature resolution, inducing-grid resolution, and kernel choice, and found that KSTPP is robust across these settings (Appendix E). We further reported an empirical runtime and memory analysis as a function of the number of events and grid resolution (Appendix F), along with a per-epoch training-time comparison to neural baselines, where KSTPP attains competitive efficiency (Appendix G). To quantify interpretability, we measured influence-kernel recovery through its correlation with the ground-truth kernel on synthetic data and a sparsity metric on real data (Appendix H); we additionally verified on a synthetic benchmark with a nonseparable ground-truth kernel that KSTPP faithfully recovers nonseparable influence structure despite its product-kernel covariance (Appendix I).

6 CONCLUSION

We have presented KSTPP, a Kronecker-structured nonparametric spatiotemporal point process model that enables explicit event-wise relationship discovery while maintaining high modeling flexibility. Empirical evaluations on both synthetic datasets and real-world benchmarks demonstrate promising performance of KSTPP.

Currently, our method is restricted to rectangular spatial domains due to the tensor-product quadrature construction. In future work, we plan to extend our framework to irregular spatial domains by incorporating triangular finite-element discretizations and adaptive quadrature schemes. Such extensions would enable efficient inference over complex geometries while preserving the numerical stability and computational efficiency. Additionally, exploring sparse-grid quadrature methods may further improve scalability in higher-dimensional settings.

ACKNOWLEDGMENTS

SZ acknowledges support from NSF CAREER Award IIS-2046295 and NSF CSSI-2311685 (Elements: A Convergent Physics-based and Data-driven Computing Platform for Building Modeling), and NSF DMS-2529112 (Collaborative Research: MATH-DT: Computationally efficient hypercomplex variable-based sensitivity methods for rapid Digital Twin model updating).

References

Charles Blundell, Jeff Beck, and Katherine A Heller. Modelling reciprocating relationships with hawkes processes. In *Advances in Neural Information Processing Systems*, pages 2600–2608, 2012.

Ricky T. Q. Chen, Brandon Amos, and Maximilian Nickel. Neural spatio-temporal point processes. In *International Conference on Learning Representations*, 2021.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume II: general theory and structure*. Springer, 2008.

Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J Smola, and Le Song. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 219–228. ACM, 2015.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1555–1564, 2016.

Shikai Fang, Madison Cooley, Da Long, Shibo Li, Robert Kirby, and Shandian Zhe. Solving high frequency and multi-scale pdes with Gaussian processes. In *International Conference on Learning Representation*, 2024.

Ruth A. Harris. Introduction to special section: Stress triggers, stress shadows, and implications for seismic hazard. *Journal of Geophysical Research: Solid Earth*, 103(B10): 24347–24358, 1998.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Pavel Izmailov, Alexander Novikov, and Dmitry Kropotov. Scalable gaussian processes with billions of inducing inputs via tensor train decomposition. In *International Conference on Artificial Intelligence and Statistics*, pages 726–735, 2018.

Junteng Jia and Austin R Benson. Neural jump stochastic differential equations. *Advances in Neural Information Processing Systems*, 32, 2019.

Geoffrey C.P. King, Ross S. Stein, and Jian Lin. Static stress changes and the triggering of earthquakes. *Bulletin of the Seismological Society of America*, 84(3):935–953, 1994.

Tamara Gibson Kolda. *Multilinear operators for higher-order decompositions*, volume 2. United States. Department of Energy, 2006.

Chris Lloyd, Tom Gunter, Michael Osborne, and Stephen Roberts. Variational inference for gaussian process modulated poisson processes. In *International Conference on Machine Learning*, pages 1814–1822. PMLR, 2015.

Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*, pages 6754–6764, 2017.

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

Yosihiko Ogata. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical Association*, 83(401):9–27, 1988.

Zhimeng Pan, Zheng Wang, and Shandian Zhe. Scalable nonparametric factorization for high-order interaction events. In *International Conference on Artificial Intelligence and Statistics*, pages 4325–4335. PMLR, 2020.

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

Yunus Saat**c**ci. *Scalable inference for structured Gaussian process models*. PhD thesis, Citeseer, 2012.

Aaron Schein, John Paisley, David M Blei, and Hanna Wallach. Bayesian poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1045–1054. ACM, 2015.

Aaron Schein, Mingyuan Zhou, David M. Blei, and Hanna Wallach. Bayesian poisson tucker decomposition for learning the structure of international relations. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 2810–2819. JMLR.org, 2016. URL `http://dl.acm.org/citation.cfm?id=3045390.3045686`.

Aaron Schein, Scott Linderman, Mingyuan Zhou, David Blei, and Hanna Wallach. Poisson-randomized gamma dynamical systems. In *Advances in Neural Information Processing Systems*, pages 782–793, 2019.

Joseph Sill. Monotonic networks. *Advances in neural information processing systems*, 10, 1997.

Tokuji Utsu, Yosihiko Ogata, and Ritsuko S. Matsuura. The centenary of the omori formula for a decay law of aftershock activity. *Journal of Physics of the Earth*, 43(1): 1–33, 1995.

Yichen Wang, Xiaojing Ye, Hongyuan Zha, and Le Song. Predicting user activity level in point processes with mass transport equation. In *Advances in Neural Information Processing Systems*, pages 1644–1654, 2017.

Zheng Wang, Xinqi Chu, and Shandian Zhe. Self-modulating nonparametric event-tensor factorization. In *International Conference on Machine Learning*, pages 9857–9867. PMLR, 2020.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning*, pages 1775–1784, 2015.

Hongteng Xu, Dixin Luo, Xu Chen, and Lawrence Carin. Benefits from superposed hawkes processes. In *International Conference on Artificial Intelligence and Statistics*, pages 623–631. PMLR, 2018.

Zenglin Xu, Feng Yan, and Yuan Qi. Infinite Tucker decomposition: nonparametric Bayesian models for multiway data analysis. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1675–1682, 2012.

Zhitong Xu, Da Long, Yiming Xu, Guang Yang, Shandian Zhe, and Houman Owhadi. Toward efficient kernel-based solvers for nonlinear pdes. In *Forty-second International Conference on Machine Learning*, 2025.

Zhitong Xu, Qiwei Yuan, Yinghao Chen, Shandian Zhe, and Bin Shen. Structured neural marked point processes for interpretable event interaction modeling. *arXiv preprint arXiv:2605.17568*, 2026.

Jiasen Yang, Vinayak A Rao, and Jennifer Neville. Decoupling homophily and reciprocity with latent space network models. In *UAI*, 2017.

Yuan Yuan, Jingtao Ding, Chenyang Shao, Depeng Jin, and Yong Li. Spatio-temporal diffusion point processes. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 3173–3184, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030. doi: 10.1145/3580305.3599511. URL https://doi.org/10.1145/3580305.3599511.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes process. In *International Conference on Machine Learning*, pages 11183–11193. PMLR, 2020.

Shandian Zhe and Yishuai Du. Stochastic nonparametric event-tensor decomposition. In *Advances in Neural Information Processing Systems*, pages 6856–6866, 2018.

Shandian Zhe, Yuan Qi, Youngja Park, Zenglin Xu, Ian Molloy, and Suresh Chari. Dintucker: Scaling up Gaussian process models on large multidimensional arrays. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Shandian Zhe, Wei Xing, and Robert M Kirby. Scalable high-order gaussian process regression. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2611–2620, 2019.

Zihao Zhou and Rose Yu. Automatic integration for spatiotemporal neural point processes. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 50237–50253. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9d30c2def27b5c6a5fb21a9aa5c16f8f-Paper-Conferen pdf.

Zihao Zhou, Xingyi Yang, Ryan Rossi, Handong Zhao, and Rose Yu. Neural point process for learning spatiotemporal event dynamics. In Roya Firoozi, Negar Mehr, Esen Yel, Rika Antonova, Jeannette Bohg, Mac Schwager, and Mykel Kochenderfer, editors, *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, volume 168 of *Proceedings of Machine Learning Research*, pages 777–789. PMLR, 23–24 Jun 2022. URL https://proceedings.mlr.press/v168/zhou22a.html.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International Conference on Machine Learning*, pages 11692–11702. PMLR, 2020.

Appendix

Zhitong Xu[1] **Qiwei Yuan**[1] **Yinghao Chen**[1] **Yan Sun**[2] **Bin Shen**[3] **Shandian Zhe**[1]

[1]Kahlert School of Computing, The University of Utah
[2]College of Arts & Sciences, Utah State University
[3]Celonis AI

A COMPUTATIONAL COMPLEXITY

The time complexity for processing a mini-batch of B sequences, each of length N, is

$$\mathcal{O}\left(\sum_{k=0}^{2} m_k^3 + \sum_{k=1}^{2} v_k^3 + BN\left(m(\sum_{i=0}^{2} q_i) + v(\sum_{i=1}^{2} q_i)\right)\right),$$

where $m = \prod_{k=0}^{2} m_k$ denotes the total number of mesh points in $\mathcal{M}_f$, $v = \prod_{k=1}^{2} v_k$ denotes the total number of mesh points in $\mathcal{M}_g$, and q_i is the number of quadrature nodes along each dimension i. The memory complexity is

$$\mathcal{O}(\sum_{k=0}^{2} m_k^2 + \sum_{k=1}^{2} v_k^2 + m + v + BNq),$$

where $q = \prod_i q_i$ is the total number of nodes in the tensor-product grid $\mathcal{Q}$. The dominant memory cost arises from storing the per-dimension kernel matrices and maintaining function evaluations at mesh points, quadrature nodes, and observed events. Overall, the cubic terms depend only on per-dimension mesh sizes rather than the total number of events, enabling scalable mini-batch training and inference.

B DATASET DETAILS

All the datasets, including the training, validation, and test splits, were downloaded from `https://github.com/tsinghua-fib-lab/Spatio-temporal-Diffusion-Point-Processes/tree/main/dataset`.

- **Earthquake** [Chen et al., 2021]. This dataset contains the time and locations of earthquakes and aftershocks in Japan from 1990 to 2020. Each sequence corresponds to events within one month, with time horizon $T = 30$ (time unit: days). The dataset contains 1,050 sequences in total, with sequence lengths ranging from 18 to 543. We used 950 sequences for training, 50 for validation, and 50 for testing.
- **COVID-19** [Chen et al., 2021]. This dataset records daily COVID-19 cases across counties in New Jersey from March 2020 to July 2020. Each sequence represents events within one week ($T = 7$). The dataset contains 1,650 sequences, with sequence lengths varying from 5 to 287. We used 1,450 sequences for training, 100 for validation, and 100 for testing.
- **Citibike** [Yuan et al., 2023]. This dataset contains bike-sharing records from April to August 2019 in New York City. The time unit is hours, and each sequence represents events within one day ($T = 24$). We used 2,440 sequences for training, 300 for validation, and 320 for testing.

C IMPLEMENTATION DETAILS AND HYPERPARAMETERS

The inducing grid is constructed by taking evenly spaced points along each input dimension over the corresponding range and forming their Cartesian product. For all experiments, each kernel is chosen as either the squared-exponential (SE)

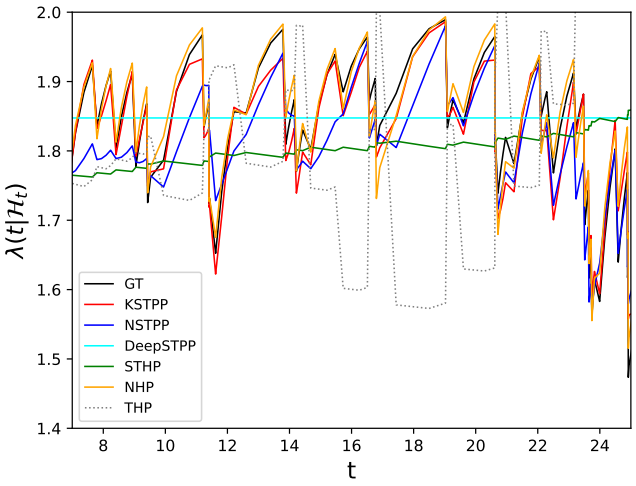

Figure 5: Temporal conditional intensity on example test sequences from **SYN2**.

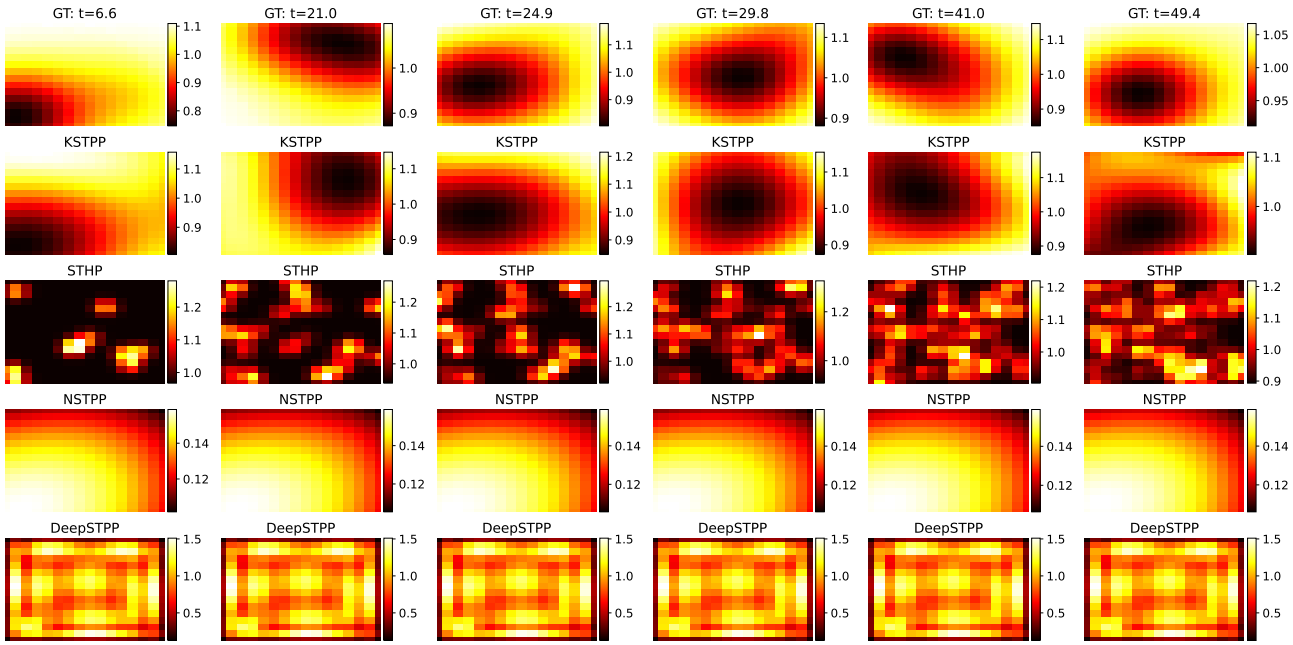

Figure 6: Spatial conditional intensities on an example test sequence from **SYN2** at different time points.

kernel or the Matérn-$5/2$ kernel; the kernel hyperparameters are not manually tuned or selected by cross-validation, but are instead learned jointly with the remaining model parameters during training. All models are optimized with Adam using a learning rate of 10^{-3}. Table 4 summarizes the hyperparameters used by KSTPP on each real-world dataset. Here F denotes the spatiotemporal inducing grid used for the influence-kernel GP and G denotes the spatial inducing grid used for the background-rate GP. We used ReLU activation for the MLP head on the Earthquake dataset, and GELU activation on COVID and Citibike datasets.

D MODEL PIPELINE

Figure 8 illustrates the overall KSTPP pipeline. Starting from the observed spatiotemporal event sequences, the conditional intensity combines a background rate with the aggregated influence of past events through a softplus link function. The background rate and the influence kernel are each modeled by a Gaussian process whose product kernel induces a Kronecker-structured covariance on the corresponding inducing grid. The intractable likelihood integral is approximated

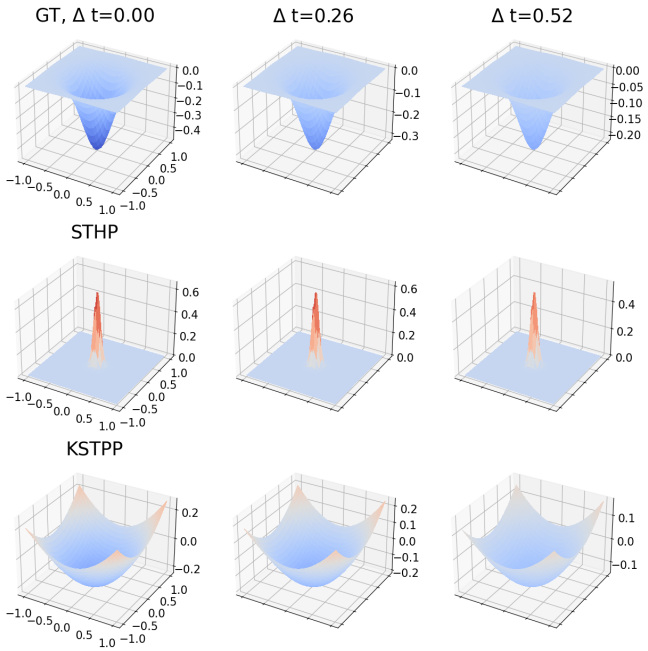

Figure 7: Learned influence kernel from **SYN2** dataset.

Dataset	Kernel	Inducing Grid Size	Quadrature Grid Size	Optimizer / LR
Earthquake	SE	F: 32×32×32; G: 12×12	8×12×12	Adam / 10^{-3}
COVID-19	Matérn-5/2	F: 32×32×32; G: 16×16	8×16×16	Adam / 10^{-3}
Citibike	Matérn-5/2	F: 32×32×32; G: 12×12	8×12×12	Adam / 10^{-3}

Table 4: Hyperparameters used by KSTPP on each real-world dataset.

with a tensor-product Gauss–Legendre quadrature grid, and the GP priors act as regularizers. Maximizing the resulting posterior yields the trained model, which supports predictive intensity estimation, next-event prediction, and interpretable influence-pattern discovery.

E SENSITIVITY AND ABLATION STUDIES

We studied the sensitivity of KSTPP to the quadrature resolution, the inducing-grid resolution, and the kernel choice on the COVID-19 dataset. In each study, we varied one factor while holding the others fixed. We reported the spatial prediction error (*Distance*) and the event-time prediction error (*Time RMSE*); lower is better for both.

E.1 QUADRATURE RESOLUTION

We varied the quadrature grid size while fixing the inducing grids to F: 32×32×32 and G: 16×16 and using the Matérn-5/2 kernel. As shown in Table 5, an overly coarse quadrature grid (e.g., 4×4×4) yielded a noticeably worse spatial prediction error, and the spatial performance improved steadily as the quadrature resolution increased. In contrast, the time RMSE was relatively stable across resolutions. A likely explanation is that the inter-event time intervals in this dataset are relatively short, so a moderate number of quadrature nodes is already sufficient to approximate the temporal contribution to the likelihood.

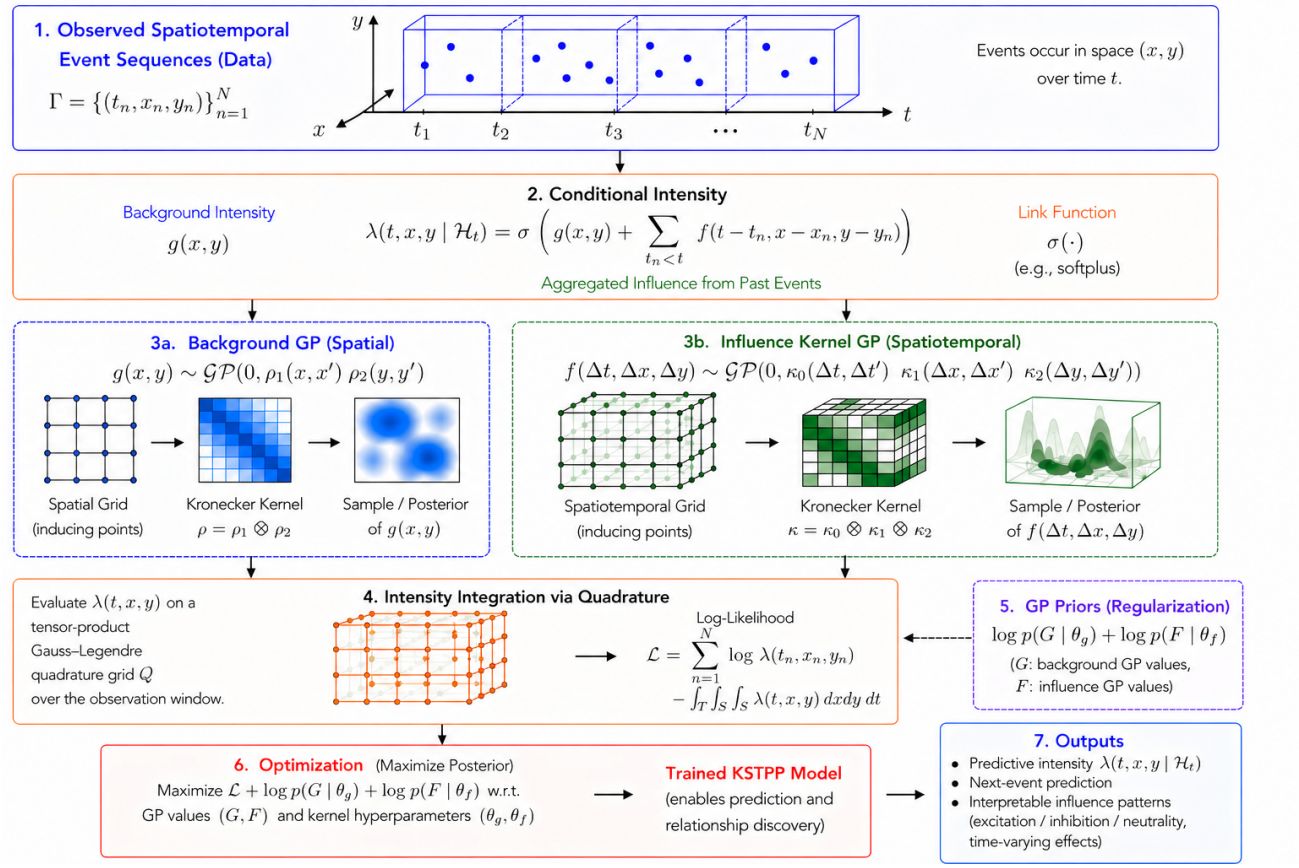

Figure 8: Overall KSTPP model pipeline: (1) observed spatiotemporal event sequences; (2) conditional intensity with a background rate, aggregated influence from past events, and a softplus link function; (3a) background spatial GP and (3b) influence-kernel spatiotemporal GP, both represented on Kronecker-structured inducing grids; (4) intensity integration via a tensor-product Gauss–Legendre quadrature grid for the log-likelihood; (5) GP priors for regularization; (6) optimization of the posterior to obtain the trained KSTPP model; and (7) outputs, including predictive intensity, next-event prediction, and interpretable influence patterns.

E.2 INDUCING-GRID RESOLUTION

We varied the inducing-grid resolution while fixing the quadrature grid to $16\times16\times16$ and using the Matérn-$5/2$ kernel. The spatial resolution of G was set equal to that of F, so we do not list G separately. Table 6 shows that performance was relatively robust to the inducing-grid resolution, suggesting that the target latent functions (the background rate and the influence function) are relatively smooth on this dataset, so that even a coarse grid such as $4\times4\times4$ provided sufficient coverage. Increasing the resolution from $4\times4\times4$ to $8\times8\times8$ gave a slight improvement, while further increasing the resolution brought no additional gain and could slightly degrade performance, likely due to the added optimization burden of a larger number of inducing variables.

E.3 KERNEL CHOICE

We compared the SE and Matérn-$5/2$ kernels while fixing the inducing grids to F: $32\times32\times32$ and G: 16×16 and the quadrature grid to $8\times16\times16$ (the first dimension is temporal and the last two are spatial). As shown in Table 7, the SE kernel performed slightly worse than the Matérn-$5/2$ kernel in both spatial and temporal prediction, but the differences were small. This indicates that both kernels work well on this dataset and that predictive performance is only mildly sensitive to this choice.

Quadrature Resolution	Distance	Time RMSE
$4\times4\times4$	0.445	0.103
$8\times8\times8$	0.400	0.101
$12\times12\times12$	0.397	0.102
$16\times16\times16$	0.393	0.102

Table 5: Effect of quadrature resolution on prediction error (COVID-19).

Inducing Grid Resolution	Distance	Time RMSE
$4\times4\times4$	0.393	0.0986
$8\times8\times8$	0.391	0.0985
$12\times12\times12$	0.391	0.0999
$16\times16\times16$	0.394	0.101

Table 6: Effect of inducing-grid resolution on prediction error (COVID-19).

F RUNTIME AND MEMORY ANALYSIS

We reported empirical runtime and memory usage as a function of the number of events and the grid resolution. Runtime is measured in seconds and memory usage in gigabytes (GB). All experiments were conducted on a Linux workstation with an NVIDIA H200 GPU. We reported the per-epoch training time, the total prediction time, and the peak training and test memory.

F.1 EFFECT OF THE NUMBER OF EVENTS

To isolate the effect of sequence length, we simulated 100 training sequences and 10 test sequences of equal length, varying the length over $\{50, 100, 150, 200, 250\}$ while fixing both the inducing grid and the quadrature grid to 8×8. As shown in Table 8, both runtime and memory grew with the sequence length, because longer sequences require more likelihood evaluations and more event-history-dependent computation during training and prediction.

F.2 EFFECT OF GRID RESOLUTION

We next varied the grid resolution at a fixed sequence length of 100. First we varied the quadrature grid size with the inducing grid fixed to 8×8 (Table 9); then we varied the inducing grid size with the quadrature grid fixed to 8×8 (Table 10).

Increasing either grid resolution raised memory usage and could also increase runtime, but the runtime growth with grid size was moderate compared with the growth caused by longer sequences. This is likely because the tensor-product quadrature computations, the local kernel matrices, and the Kronecker-algebra operations on the inducing grid parallelize well on the GPU, partially offsetting the cost of finer grids. In contrast, the number of events had a more pronounced effect on both runtime and memory, indicating that sequence length is the major practical driver of computational cost.

G TRAINING-TIME COMPARISON WITH NEURAL BASELINES

Table 11 reports the per-epoch training time (seconds) of KSTPP and the neural baselines on the COVID-19 and Earthquake datasets. The per-epoch training time of KSTPP was comparable to NSTPP and substantially lower than DSTPP, the diffusion-based approach. DeepSTPP was the fastest, as expected, because it uses a simpler excitation-only Hawkes-process intensity. Overall, KSTPP attained competitive training efficiency relative to the neural baselines while additionally providing flexible and transparent event-wise relationship discovery.

Kernel	Distance	Time RMSE
Matérn-$5/2$	0.392	0.100
SE	0.393	0.101

Table 7: Effect of kernel choice on prediction error (COVID-19).

Seq. Length	Training Time	Prediction Time	Training Memory	Test Memory
50	1.39	0.40	1.03	0.96
100	2.43	1.18	1.72	1.44
150	4.12	3.91	3.07	2.39
200	6.82	6.81	4.32	3.20
250	10.37	10.61	6.37	4.61

Table 8: Runtime (s) and memory (GB) under different event sequence lengths.

H QUANTITATIVE INFLUENCE-KERNEL RECOVERY

H.1 CORRELATION WITH THE GROUND TRUTH ON SYNTHETIC DATA

For the synthetic datasets SYN1 and SYN2, we quantified influence-kernel recovery by the correlation coefficient between the learned influence kernel and the ground-truth influence kernel, evaluated at several temporal lags Δt corresponding to the slices visualized in the main paper. Tables 12 and 13 report the results for KSTPP and the STHP baseline.

KSTPP attained substantially higher correlation with the ground truth than STHP, quantitatively confirming better recovery of the influence function. Moreover, KSTPP obtained positive correlations across all selected Δt values, indicating stable recovery of the influence-function shape. In contrast, STHP produced negative correlations on SYN1 at $\Delta t = 1.77$ and on SYN2 at all selected Δt values, indicating that it can recover an incorrect influence-function shape. SYN2 was more challenging than SYN1 for influence-function estimation, which was reflected in its lower correlation coefficients.

H.2 SPARSITY STRUCTURE ON REAL DATA

For the real-world Earthquake dataset the ground-truth influence kernel is unavailable, so we instead reported a sparsity/structure metric for the learned influence function. Following the Δt slices shown in the main paper, we set a sparsity threshold of 0.05 and defined the *sparsity ratio* as the proportion of learned influence-function values whose absolute value falls below this threshold. Table 14 reports the sparsity ratio at several Δt values.

The consistently high sparsity ratio indicated that the learned influence function was highly structured and localized. As Δt increased, the sparsity ratio increased until nearly all function values became very small, showing that the influence of past events decays over time and becomes negligible after a sufficiently long interval.

I RECOVERY OF A NONSEPARABLE INFLUENCE KERNEL

The product-kernel covariance used by KSTPP induces a separable covariance structure, but this does not imply that the represented latent function is separable. To verify this empirically, we constructed a synthetic dataset whose influence kernel is explicitly nonseparable. Following the generative form used for the other synthetic datasets, we set the influence kernel to a zero-mean Gaussian-shaped function with a dense covariance matrix,

$$f(\Delta t, \Delta x, \Delta y) = \mathcal{N}\big([\Delta t, \Delta x, \Delta y] \mid \mathbf{0}, \Sigma\big), \qquad \Sigma = \begin{bmatrix} 0.08 & 0.04 & 0.02 \\ 0.04 & 0.20 & 0.05 \\ 0.02 & 0.05 & 0.20 \end{bmatrix}. \tag{19}$$

Because Σ has nonzero off-diagonal entries, i.e., cross-dimensional dependencies between the temporal and spatial displacements, this influence kernel is nonseparable.

Quadrature Size	Training Time	Prediction Time	Training Memory	Test Memory
4	1.54	1.05	1.79	1.72
8	2.14	1.03	2.00	1.78
12	3.20	1.20	2.27	1.74
16	3.61	1.10	2.70	1.93
24	5.39	1.20	3.91	2.72

Table 9: Effect of quadrature resolution on runtime (s) and memory (GB).

Inducing Grid Size	Training Time	Prediction Time	Training Memory	Test Memory
4	0.86	0.10	1.49	1.28
12	0.96	0.20	2.30	1.58
16	0.95	0.20	2.50	1.84
24	0.99	0.20	3.31	1.96

Table 10: Effect of inducing-grid resolution on runtime (s) and memory (GB).

We simulated the same number of training sequences as in SYN1 and SYN2, trained KSTPP on this dataset, and examined the recovered influence kernel. Figure 9 compares the ground-truth influence kernel (top row) with the kernel recovered by KSTPP (bottom row) across a range of temporal lags Δt: KSTPP closely reproduced the ground-truth shape and its temporal decay. In addition, the relative L_2 error for recovering the intensity function on the held-out test sequences was 0.0361, which is on the same order as the relative L_2 errors reported for SYN1 and SYN2 in the main paper. These results provided direct evidence that KSTPP can capture nonseparable influence functions despite using a product-kernel covariance structure.

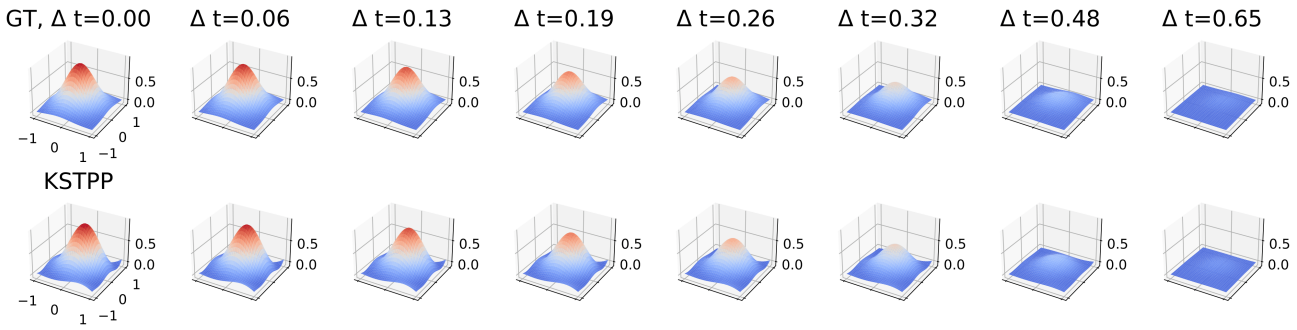

Figure 9: Recovery of a nonseparable influence kernel. Top row: the ground-truth (GT) influence kernel $f(\Delta t, \Delta x, \Delta y)$ shown as spatial surface slices at increasing temporal lags $\Delta t \in \{0.00, 0.06, 0.13, 0.19, 0.26, 0.32, 0.48, 0.65\}$. Bottom row: the corresponding influence kernel recovered by KSTPP at the same Δt slices. KSTPP closely matches the ground-truth shape and its temporal decay despite using a product-kernel covariance structure.

Method	COVID-19	Earthquake
NSTPP	21.3	25.5
DeepSTPP	1.3	0.8
DSTPP	43.9	29.3
KSTPP	24.5	19.7

Table 11: Per-epoch training time (seconds).

Method	$\Delta t = 0.00$	$\Delta t = 0.26$	$\Delta t = 1.77$
STHP	0.6409	0.6409	-0.6409
KSTPP	0.9932	0.9932	0.8706

Table 12: Correlation between the learned and ground-truth influence kernel on SYN1.

Method	$\Delta t = 0.00$	$\Delta t = 0.26$	$\Delta t = 0.52$
STHP	-0.4379	-0.4379	-0.4379
KSTPP	0.7502	0.7503	0.7507

Table 13: Correlation between the learned and ground-truth influence kernel on SYN2.

Metric	$\Delta t = 0.00$	$\Delta t = 0.26$	$\Delta t = 0.52$	$\Delta t = 10.35$
Sparsity Ratio	88.6%	96.7%	99.1%	100%

Table 14: Sparsity ratio of the learned influence function on the Earthquake dataset.