Neural Jump-Diffusion Temporal Point Processes

Shuai Zhang [1] **Chuan Zhou** [1 2] **Yang Liu** [1] **Peng Zhang** [3] **Xixun Lin** [4] **Zhi-Ming Ma** [1]

Abstract

We present a novel perspective on temporal point processes (TPPs) by reformulating their intensity processes as solutions to stochastic differential equations (SDEs). In particular, we first prove the equivalent SDE formulations of several classical TPPs, including Poisson processes, Hawkes processes, and self-correcting processes. Based on these proofs, we introduce a unified TPP framework called Neural Jump-Diffusion Temporal Point Process (NJDTPP), whose intensity process is governed by a neural jump-diffusion SDE (NJDSDE) where the drift, diffusion, and jump coefficient functions are parameterized by neural networks. Compared to previous works, NJDTPP exhibits model flexibility in capturing intensity dynamics without relying on any specific functional form, and provides theoretical guarantees regarding the existence and uniqueness of the solution to the proposed NJDSDE. Experiments on both synthetic and real-world datasets demonstrate that NJDTPP is capable of capturing the dynamics of intensity processes in different scenarios and significantly outperforms the state-of-the-art TPP models in prediction tasks.

1. Introduction

Many real-world scenarios often generate a large amount of asynchronous event sequences. Each event consists of a timestamp and a type mark, indicating when and what the event occurred. Examples include user activities on social media platforms (Farajtabar et al., 2017), electronic health records in healthcare (Liu & Hauskrecht, 2021), and transaction behaviors in e-commerce systems (Xue et al., 2022). Modeling such data has become increasingly important for tasks such as predicting the occurrence of future events (Du et al., 2016; Mei & Eisner, 2017; Zhang et al., 2020a; Zuo et al., 2020), detecting anomalies in event sequences (Liu & Hauskrecht, 2021; Shchur et al., 2021; Zhang et al., 2023), and performing causal inference on events (Xu et al., 2016; Zhang et al., 2020b; Gao et al., 2021).

Temporal point processes (TPPs) (Daley et al., 2003) serve as a useful mathematical tool for modeling sequences of discrete events in continuous time. Classical examples of TPPs include Poisson processes (Kingman, 1992), Hawkes processes (Hawkes, 1971), and self-correcting processes (Isham & Westcott, 1979). A central concept in TPPs is the intensity process[1] (Oakes, 1975), also known as the intensity function (Zhang et al., 2020b), which measures the expected rate of events occurrence given historical events. While these classical models exhibit favorable statistical properties, the fixed parametric form of their intensity functions prevents them from capturing complicated dynamics.

To enhance the capability of TPP models, there has been a surge in modeling the intensity function as a transformation of the hidden state of neural networks. Depending on the neural network structures, these TPP models can be divided into two categories, i.e., those based on either RNNs or Transformers (Du et al., 2016; Zuo et al., 2020; Yang et al., 2022), and those based on continuous-depth neural networks (Jia & Benson, 2019; Chen et al., 2020). While being more expressive than classical TPPs, the former models usually assume a specific functional form for the intensity function. For example, RMTPP (Du et al., 2016) assumes that the intensity exponentially decreases or increases between events. However, relying on such an assumption would limit model expressiveness when the employed assumption deviates from reality (Omi et al., 2019). The latter models represent the hidden state as the solution to a neural jump stochastic differential equation (Jia & Benson, 2019). These models, however, provide no theoretical guarantee for the global existence and uniqueness of the solution.

In this paper, we provide a new view for TPPs by reformulating the intensity process as the solution to a stochastic differential equation (SDE) (Ikeda & Watanabe, 2014). Specifi-

[1]Academy of Mathematics and Systems Science, Chinese Academy of Sciences [2]School of Cyber Security, University of Chinese Academy of Sciences [3]Cyberspace Institute of Advanced Technology, Guangzhou University [4]Institute of Information Engineering, Chinese Academy of Sciences. Correspondence to: Chuan Zhou <zhouchuan@amss.ac.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]In this paper, we use intensity process and intensity function interchangeably.

cally, we first derive equivalent SDE formulations of several classical TPPs mentioned above. From these SDE formulations, we observe that the coefficient functions in SDE play a key role in shaping the evolution of intensity process over time and revealing the influences between events. Based on these observations, we introduce the Neural Jump-Diffusion Temporal Point Process (NJDTPP), whose intensity process is governed by a neural jump-diffusion SDE (NJDSDE). The drift, diffusion, and jump coefficient functions in NJDSDE are parameterized by three neural networks, i.e., the drift net, diffusion net, and jump net. Concretely, the drift net captures the intrinsic evolution of the intensity process, the diffusion net models the Gaussian noise with the Brownian motion (Wang et al., 2017; 2018), and the jump net captures the influences between events, such as the excitatory and inhibitory influences. Remarkably, our NJDTPP model does not require a specific functional form for the intensity function. Instead, by using the drift, diffusion, and jump nets, the solution to NJDSDE can implicitly determine a free-form intensity process consistent with the observed event data. We summarize our contributions as follows:

- **Theoretical Analysis.** We prove the equivalent SDE formulations of several classical TPPs. For the SDE formulation, we provide a sufficient condition for the existence of a unique positive solution. Moreover, we theoretically analyze the existence and uniqueness of the solution to the proposed neural jump-diffusion SDE.
- **Unified Framework.** By viewing the intensity process as the solution to an SDE, we propose a unified TPP framework NJDTPP which can learn a free-form intensity process consistent with the observed data. A number of classical TPPs can be interpreted as special cases of our framework with simple coefficient functions.
- **Extensive Experiments.** We conduct experiments on three synthetic and six real-world datasets to evaluate the performance of NJDTPP. Experimental results show that NJDTPP successfully captures the dynamics of intensity processes and achieves state-of-the-art results in the tasks of likelihood evaluation and event prediction.

2. Related Work

Neural Temporal Point Processes. Neural TPPs that combine TPPs with neural networks have received considerable attention (Du et al., 2016; Mei & Eisner, 2017; Zhang et al., 2020a; Zuo et al., 2020; Lin et al., 2021; Yang et al., 2022). While being more expressive than classical parametric ones, neural TPPs usually assume a specific functional form for the intensity function. For example, RMTPP (Du et al., 2016) assumes that the intensity exponentially decreases or increases between events; THP (Zuo et al., 2020) utilizes the softplus function so that the intensity between events is approximately linearly interpolated. However, relying on such an assumption can undermine model effectiveness if the employed assumption deviates from reality. In addition to the dominant paradigm of parameterizing intensity functions, alternative methods involve modeling cumulative intensity functions (Omi et al., 2019) and conditional density functions (Shchur et al., 2019). However, these methods may not fully capture the dynamics of the intensity process. In contrast to existing studies, our model formulates the intensity process as the solution to an SDE without relying on any specific functional form.

Neural Differential Equations. Neural differential equations (NDEs) (Kidger et al., 2021a) are defined as differential equations in which coefficient functions are parameterized by neural networks. Many NDEs, including neural ODE and its variants (Chen et al., 2018; Rubanova et al., 2019; Kidger et al., 2020; Herrera et al., 2020), as well as neural SDEs (Li et al., 2020; Kong et al., 2020; Kidger et al., 2021a;b), have been proposed for modeling time series. However, there is a distinction between time series and event sequences (Xiao et al., 2017). In time series, time serves only as the index to order the sequence of values for the target variable. In event sequences, time is regarded as a random variable representing the timestamp of asynchronous events, with time itself being the subject of research. Therefore, many existing NDE-based models are not directly suitable for modeling event sequences. While Jia & Benson (2019) and Chen et al. (2020) utilize NDEs to model event sequences, they actually capture the dynamics of the hidden state of neural networks. Besides, they solely focus on the jump term, neglecting the diffusion term associated with randomness driven by Brownian motion. In contrast, we incorporate Brownian motion to model Gaussian noise, and more importantly our proposed NJDSDE models the dynamics of the intensity process.

Equivalent SDE Formulations for TPPs. Wang et al. (2018) provided a jump-diffusion SDE framework for modeling user activities. They introduced the diffusion term to model the Gaussian noise, such as fluctuations in the dynamics caused by unobserved factors. However, their utilization of fixed linear coefficient functions in the SDE might not fully capture the actual intensity. On the contrary, we employ neural networks to parameterize coefficient functions, allowing for a more flexible modeling of the intensity that better aligns with the observed data. While De et al. (2016); Zarezade et al. (2017); Wang et al. (2018) established the equivalent SDE formulation for Hawkes processes, we provide a distinct proof method. Besides, we derive equivalent SDE formulations for several other classical TPPs, such as Poisson processes and self-correcting processes. Moreover, for the SDE formulation, we provide a sufficient condition for the existence of a unique positive solution.

3. Background

In this section, we provide a brief overview of temporal point processes and jump-diffusion stochastic differential equations.

3.1. Temporal Point Processes

A temporal point process (TPP) (Daley et al., 2003) is a stochastic process $\{t_i\}_{i=1}^{\infty}$, in which the non-negative random variable t_i represents the occurrence time of the i-th event and $t_i < t_{i+1}$. Such a process can be equivalently represented as a counting process $\{N_t\}_{t\geq 0}$, where N_t represents the number of events up to time t.

The most common way to characterize a TPP is via its intensity process (Oakes, 1975), also known as the intensity function. Specifically, the intensity process of $\{N_t\}_{t\geq 0}$ is a left-continuous with right-limits process $\{\lambda(t \mid \mathcal{F}_{t-})\}_{t\geq 0}$, denoted for simplicity as $\{\lambda_t\}_{t\geq 0}$, where λ_t measures the expected rate of events occurring in an infinitesimal window $(t, t+\mathrm{d}t]$ given the historical events up to time t. Formally,

$$\lambda_t \,\mathrm{d}t = \mathbb{P}\left(\mathrm{d}N_t = 1 \mid \mathcal{F}_{t-}\right) = \mathbb{E}\left[\mathrm{d}N_t \mid \mathcal{F}_{t-}\right], \quad (1)$$

where $\mathcal{F}_{t-} = \sigma(N_s : 0 \leq s < t)$ and the jump size $\mathrm{d}N_t = N_{t+\mathrm{d}t} - N_t \in \{0, 1\}$.

In the following, we review several classical TPPs, where the intensity function has a fixed parametric form.

Poisson processes (Kingman, 1992). The intensity function of the Poisson process $\{N_t\}_{t\geq 0}$ is independent of event history. The simplest case is a homogeneous Poisson process where the intensity is a positive constant:

$$\lambda_t = \lambda > 0. \quad (2)$$

For a more general inhomogeneous poisson process, the intensity is a function varying over time:

$$\lambda_t = g(t) > 0. \quad (3)$$

Hawkes processes (Hawkes, 1971). The Hawkes process $\{N_t\}_{t\geq 0}$ with the widely used exponential kernel assumes that events are self-exciting. The arrival of a new event results in a sudden increase in intensity, and this influence decays exponentially:

$$\lambda_t = \mu + \alpha \sum_{i:\, t_i<t} \exp\left(-\beta(t - t_i)\right), \quad (4)$$

where $\mu > 0$, $\alpha > 0$ and $\beta > 0$.

Self-correcting processes (Isham & Westcott, 1979). In contrast to the Hawkes process, the self-correcting process $\{N_t\}_{t\geq 0}$ assumes that a new event inhibits future events and the intensity grows exponentially over time:

$$\lambda_t = \exp\left(\mu t - \sum_{i:\, t_i<t} \alpha\right), \quad (5)$$

where $\mu > 0$ and $\alpha > 0$.

3.2. Jump-Diffusion Stochastic Differential Equations

One-dimensional autonomous jump-diffusion stochastic differential equations (JDSDE) (Hanson, 2007) with initial conditions are of the form

$$\begin{cases} \mathrm{d}X_t = f(X_t)\,\mathrm{d}t + g(X_t)\,\mathrm{d}W_t + h(X_t)\,\mathrm{d}N_t, \\ X_0 = x_0, \end{cases} \quad (6)$$

where $x_0 \in \mathbb{R}$ is the initial value, $f : \mathbb{R} \to \mathbb{R}$ is the drift coefficient function, $g : \mathbb{R} \to \mathbb{R}$ is the diffusion coefficient function, $h : \mathbb{R} \to \mathbb{R}$ is the jump coefficient function, $\{W_t\}_{t\geq 0}$ is a standard Brownian motion, and $\{N_t\}_{t\geq 0}$ is a counting process that jumps at times $\{t_i\}_{i=1}^{\infty}$. Suppose that $\{W_t\}_{t\geq 0}$ and $\{N_t\}_{t\geq 0}$ are independent. In this paper, it is essential to highlight that the process $\{N_t\}_{t\geq 0}$ in Eq.(6) is a general counting process introduced in Section 3.1, distinct from many previous works (Cyganowski et al., 2002; Hanson, 2007; Lamberton & Lapeyre, 2011) that focus on a Poisson process.

The JDSDE Eq.(6) is interpreted as a stochastic integral equation (Cyganowski et al., 2002):

$$X_t = x_0 + \int_0^t f(X_s)\,\mathrm{d}s + \int_0^t g(X_s)\,\mathrm{d}W_s + \int_0^t h(X_s)\,\mathrm{d}N_s,$$

where the first integral is a Riemann integral, the second is an Itô integral and the third is a Riemann–Stieltjes integral. In fact, Eq.(6) behaves as a normal Itô SDE (Cyganowski et al., 2002; Hanson, 2007) between jumps of $\{N_t\}_{t\geq 0}$. This can be expressed as:

$$\mathrm{d}X_t = f(X_t)\,\mathrm{d}t + g(X_t)\,\mathrm{d}W_t, \quad t \in (t_{i-1}, t_i].$$

On the other hand, at a jump time t_i, $\{N_t\}_{t\geq 0}$ has a jump size of $\Delta N_{t_i} = 1$, which implies that the process $\{X_t\}_{t\geq 0}$ will have a jump of size

$$\Delta X_{t_i} = X_{t_i+} - X_{t_i} = h(X_{t_i})\,\Delta N_{t_i} = h(X_{t_i}),$$

where $X_{t_i+} = \lim_{s\downarrow t_i} X_s$. Then $X_{t_i+} = X_{t_i} + h(X_{t_i})$.

4. Equivalent SDE Formulations for TPPs

In this section, we derive equivalent SDE formulations of several classical TPPs, which involves expressing their respective intensity process as a solution to the corresponding SDE. Then for the SDE formulation, we provide a sufficient condition for the existence of a unique positive solution.

Theorem 1. *The intensity processes of homogeneous and inhomogeneous Poisson processes can be equivalently expressed as solutions to the following ODEs, respectively. These ODEs can be viewed as degenerate forms of SDEs.*

$$\mathrm{d}\lambda_t = 0, \quad \lambda_0 = \lambda, \quad (7)$$

$$\mathrm{d}\lambda_t = g'(t)\,\mathrm{d}t, \quad \lambda_0 = g(0), \quad (8)$$

where $\lambda > 0$ and $g(t) > 0$ is assumed to be differentiable.

According to Eq.(2) and Eq.(3), Theorem 1 is evident. Subsequently, we establish equivalent SDE formulations for Hawkes processes and self-correcting processes.

Theorem 2. *The intensity process $\{\lambda_t\}_{t\geq 0}$ of the Hawkes process $\{N_t\}_{t\geq 0}$ can be equivalently expressed as the solution to the jump SDE*

$$\mathrm{d}\lambda_t = \beta(\mu - \lambda_t)\,\mathrm{d}t + \alpha\,\mathrm{d}N_t, \quad \lambda_0 = \mu. \tag{9}$$

Proof. See Appendix A.1. The proof sketch is as follows: Taking inspiration from (Björk, 2021), we now solve the above SDE. Let the jump times of $\{N_t\}_{t\geq 0}$ be $\{t_i\}_{i=1}^{\infty}$, then Eq.(9) behaves as an ODE $\mathrm{d}\lambda_t = \beta(\mu - \lambda_t)\,\mathrm{d}t$ between these jump points. And at a jump time t_i, the jump size is α, leading to $\lambda_{t_i+} = \lambda_{t_i} + \alpha$. Iteratively solving this ODE between jumps with the initial value $\lambda_{t_{i-1}+}$, we establish that the intensity process Eq.(4) satisfies Eq.(9). □

Theorem 3. *The intensity process $\{\lambda_t\}_{t\geq 0}$ of the self-correcting process $\{N_t\}_{t\geq 0}$ can be equivalently expressed as the solution to the jump SDE*

$$\mathrm{d}\lambda_t = \mu\lambda_t\,\mathrm{d}t + \left(e^{-\alpha} - 1\right)\lambda_t\,\mathrm{d}N_t, \quad \lambda_0 = 1. \tag{10}$$

The proof of this theorem is similar to the previous one and can be found in Appendix A.2. The following result shows that under certain conditions, there exits a unique positive solution to an SDE, which means that an SDE can determine an intensity process of a TPP.

Theorem 4. *Assume that the ODE*

$$\mathrm{d}y_t = \frac{f\left(e^{y_t}\right)}{e^{y_t}}\,\mathrm{d}t, \quad t \geq 0, \quad y_0 = y,$$

has a unique global solution for every $y \in \mathbb{R}$ and let $h(x) : \mathbb{R} \to \mathbb{R}$ be a chosen function such that $h(x) + x > 0$ for $x > 0$. Then the jump SDE

$$\mathrm{d}\lambda_t = f\left(\lambda_t\right)\mathrm{d}t + h\left(\lambda_t\right)\mathrm{d}N_t, \quad \lambda_0 = \lambda, \tag{11}$$

has a unique global positive solution for every $\lambda > 0$.

Appendix A.3 includes the detailed proof. Specially, according to this theorem, by setting $f(x) = \mu x$ and $h(x) = (e^{-\alpha} - 1)x$, it follows that Eq.(10) has a unique global positive solution.

From the above equivalent SDE formulations of several classical TPPs, we can clearly see that the coefficient functions within the SDE play a key role in shaping the evolution of intensity processes over time and revealing the influences between events. For example, in Hawkes processes (Eq.(9)), there exists excitatory influences between events, where each occurrence of an event leads to an instantaneous increase in intensity by α. This inspires us that by defining appropriate coefficient functions, it becomes feasible to construct an intensity process consistent with the observed data. These observations motivate us to propose our model Neural Jump-Diffusion Temporal Point Processes.

5. Neural Jump-Diffusion TPPs

In this section, for symbol simplicity and reader comprehension, we first model the intensity process of the univariate TPPs. Subsequently, we extend our method to address the multivariate TPPs, proposing a more comprehensive model.

5.1. Neural Jump-Diffusion Univariate Point Process

Unlike classical TPPs with known linear drift and jump coefficient functions, we consider a general problem where the dynamics of the intensity process are completely unknown. Specifically, we assume access to a large set of event sequences, each denoted as $\mathcal{S} = \{t_i\}_{i=1}^{n}$, representing independent realizations of a counting process $\{N_t\}_{t\geq 0}$. The objective is to identify the unknown dynamics governing the intensity process $\{\lambda_t\}_{t\geq 0}$ of $\{N_t\}_{t\geq 0}$.

To this end, we propose the Neural Jump-Diffusion Univariate Point Process whose intensity process is governed by a neural jump-diffusion SDE (NJDSDE). The drift, diffusion, and jump coefficient functions in the NJDSDE are parameterized by three neural networks which are called the drift net, diffusion net, and jump net, respectively. To ensure that the intensity $\{\lambda_t\}_{t\geq 0}$ remains positive, we introduce the log-intensity process $\eta_t := \log\lambda_t$. Then we formally present the NJDSDE for $\{\eta_t\}_{t\geq 0}$ as follows:

$$\begin{cases} \mathrm{d}\eta_t = \underbrace{f_{\theta_f}(\eta_t)}_{\text{drift net}}\,\mathrm{d}t + \underbrace{g_{\theta_g}(\eta_t)}_{\text{diffusion net}}\,\mathrm{d}W_t + \underbrace{h_{\theta_h}(\eta_t)}_{\text{jump net}}\,\mathrm{d}N_t, \\ \eta_0 = \log\lambda_0, \end{cases} \tag{12}$$

where $\eta_0 \in \mathbb{R}$ is the initial value, $f_{\theta_f} : \mathbb{R} \to \mathbb{R}$, $g_{\theta_g} : \mathbb{R} \to \mathbb{R}$, $h_{\theta_h} : \mathbb{R} \to \mathbb{R}$, $\{W_t\}_{t\geq 0}$ is a standard Brownian motion (Le Gall, 2016), and $\{N_t\}_{t\geq 0}$ is the counting process mentioned above, which records the occurrence of events. Suppose that $\{W_t\}_{t\geq 0}$ and $\{N_t\}_{t\geq 0}$ are independent. We explain each term in Eq.(12) in detail:

- The drift term $f_{\theta_f}(\eta_t)\,\mathrm{d}t$ captures the intrinsic evolution of $\{\eta_t\}_{t\geq 0}$.
- The diffusion term $g_{\theta_g}(\eta_t)\,\mathrm{d}W_t$ models the Gaussian noise with the Brownian motion. Inspired by (Wang et al., 2018), we add the diffusion term to model the impact of noise on the intensity process.
- The jump term $h_{\theta_h}(\eta_t)\,\mathrm{d}N_t$ represents the magnitude of the jump, capturing the influence of historical events up to time t. Its sign indicates whether the influence is excitatory or inhibitory.

The proposed NJDSDE Eq.(12) is a general framework. When the function f_{θ_f} is set to $\beta(\mu e^{-\eta_t} - 1)$, g_{θ_g} is set to 0, and h_{θ_h} is set to $\log(1 + \alpha e^{-\eta_t})$ in Eq.(12), the NJDSDE characterizes Hawkes processes. Similarly, when f_{θ_f} is set

to μ, g_{θ_g} is set to 0, and h_{θ_h} is set to $-\alpha$, Eq.(12) characterizes self-correcting processes. The similar results for Poisson processes are trivial. In other words, the proposed NJDSDE encompasses the classical TPPs mentioned above. In addition, a specific class (but not all) of log-Gaussian Cox processes (Møller et al., 1998) can also be incorporated into our modeling framework Eq.(12). The proofs for these conclusions are detailed in Appendix A.4.

We proceed to investigate the existence and uniqueness of the solution $\{\eta_t\}_{t\geq 0}$ to the proposed NJDSDE. The theoretical analysis in the following theorem provides insights into designing an effective network architecture for the drift net f_{θ_f}, diffusion net g_{θ_g}, and jump net h_{θ_h}.

Theorem 5. *Assuming that $f_{\theta_f}(x)$, $g_{\theta_g}(x)$, $h_{\theta_h}(x)$ are measurable functions $\mathbb{R}\to\mathbb{R}$, $h_{\theta_h}(x)$ is continuous, and there exists a positive constant C such that for all $x, y\in\mathbb{R}$,*

$$|f_{\theta_f}(x)-f_{\theta_f}(y)|+|g_{\theta_g}(x)-g_{\theta_g}(y)|\leq C|x-y|,$$

then for every $\lambda_0>0$, there exists a unique adapted left-continuous process $\{\eta_t\}_{t\geq 0}$ with right-limits that satisfies Eq.(12).

The proof is available in Appendix A.5. According to Theorem 5, if $f_{\theta_f}(x)$, $g_{\theta_g}(x)$ and $h_{\theta_h}(x)$ are uniformly Lipschitz continuous, then Eq.(12) has a unique strong solution. Thus, we utilize Lipschitz nonlinear activations, such as ReLU, sigmoid, and Tanh, within the network architectures, as highlighted in previous works (Anil et al., 2019; Kong et al., 2020; Oh et al., 2024; Lin et al., 2024). Moreover, in this paper, the drift net, diffusion net, and jump net are implemented as three multi-layer perceptrons (MLPs).

Remarks. We summarize the differences of our model compared to existing TPP models:

- Different from the SDE formulation of classical TPPs (e.g., Eq.(9)), the coefficient functions in our model are parameterized by neural networks rather than relying on fixed functions. This enables a more flexible modeling of the complex dynamics of the intensity process.
- Compared to neural TPPs (Du et al., 2016; Zuo et al., 2020), our model eliminates the need to assume a specific functional form for the intensity function. Instead, based on the NJDSDE, our model formulates the time evolution of the intensity process in a general manner.
- Furthermore, our model differs from previous TPP models based on neural differential equations (Jia & Benson, 2019; Chen et al., 2020; Song et al., 2024). In addition to incorporating the Brownian motion to model the Gaussian noise, a key distinction lies in our proposed NJDSDE, which models the dynamics of the intensity process rather than the hidden state.

5.2. Model Training

To learn model parameters in f_{θ_f}, g_{θ_g}, h_{θ_h}, and the initial value η_0, we perform the Maximum Likelihood Estimation (MLE). For an event sequence $\mathcal{S}=\{t_i\}_{i=1}^n$ over the time interval $[0,T]$, given its intensity λ_t, the log-likelihood function (Rasmussen, 2018) is

$$\ell(\mathcal{S})=\sum_{i=1}^n\log\lambda_{t_i}-\int_0^T\lambda_s\,\mathrm{d}s=\sum_{i=1}^n\eta_{t_i}-\int_0^T e^{\eta_s}\,\mathrm{d}s.$$

In general, the integral term does not have a closed-form computational method. Therefore, we apply numerical integration methods for approximate calculations, such as the trapezoidal rule (Zuo et al., 2020). This requires determining the value of η_t at the divided time points. Noting that this process $\{\eta_t\}_{t\geq 0}$ is governed by our proposed NJDSDE Eq.(12). That is, on the time interval $(t_{i-1},t_i]$, η_t is governed by the neural SDE

$$\mathrm{d}\eta_t=f_{\theta_f}(\eta_t)\,\mathrm{d}t+g_{\theta_g}(\eta_t)\,\mathrm{d}W_t. \tag{13}$$

And at a jump time t_i, the jump size of η_t is given by

$$\Delta\eta_{t_i}=\eta_{t_i+}-\eta_{t_i}=h_{\theta_h}(\eta_{t_i})\,\Delta N_{t_i}=h_{\theta_h}(\eta_{t_i}). \tag{14}$$

Then the right-limit of η_t at t_i is

$$\eta_{t_i+}=\eta_{t_i}+h_{\theta_h}(\eta_{t_i}). \tag{15}$$

Since the solution of neural SDEs (e.g., Eq.(13)) is generally analytically intractable, numerical approximation methods are often required (Kong et al., 2020; Kidger et al., 2021b). We adopt the Euler-Maruyama scheme (Kloeden & Platen, 1992) with fixed step size due to its computational efficiency. Under such a scheme, the time interval $(t_{i-1},t_i]$ is divided into N subintervals $t_{i-1}=\tau_0^i<\cdots<\tau_k^i<\cdots<\tau_N^i=t_i$ with stepsize $\Delta_k^i=\tau_{k+1}^i-\tau_k^i=(t_i-t_{i-1})/N$. Then we discretize Eq.(13) on $(t_{i-1},t_i]$ by the recursive equation

$$\eta_{\tau_{k+1}^i}=\eta_{\tau_k^i}+f_{\theta_f}(\eta_{\tau_k^i})\,\Delta_k^i+g_{\theta_g}(\eta_{\tau_k^i})\,\Delta W_k^i, \tag{16}$$

for $k=0,1,\ldots,N-1$ with $\eta_{\tau_0^i}=\eta_{t_{i-1}+}$. Here, $\Delta W_k^i=W_{\tau_{k+1}^i}-W_{\tau_k^i}$ is sampled from $\mathcal{N}(0,\Delta_k^i)$ for numerical computation. The advantage of introducing the log-intensity $\eta_t=\log\lambda_t$ is evident in obtaining a numerical solution of Eq.(12) over the entire real number space, rather than being restricted to the domain of positive real numbers.

Iteratively using Eq.(15) and Eq.(16), we can calculate the log-likelihood function as follows:

$$\ell(\mathcal{S})=\sum_{i=1}^n\eta_{t_i}-\sum_{i=1}^{n+1}\sum_{k=1}^{N}\frac{\tau_k^i-\tau_{k-1}^i}{2}\left(e^{\eta_{\tau_{k-1}^i}}+e^{\eta_{\tau_k^i}}\right), \tag{17}$$

where $\tau_0^1=0$, $\tau_N^{n+1}=T$, $\eta_{\tau_0^i}=\eta_{t_{i-1}+}$ and $\eta_{\tau_N^i}=\eta_{t_i}$. The complete algorithm of model training is described in Algorithm 1 in Appendix B.

5.3. Neural Jump-Diffusion Multivariate Point Process

An important example of multivariate TPPs is the multivariate Hawkes process $\boldsymbol{N}_t = (N_t^1, \ldots, N_t^M)^{\mathrm{T}}$, whose intensity process $\boldsymbol{\lambda}_t = (\lambda_t^1, \ldots, \lambda_t^M)^{\mathrm{T}}$ characterizes the past event influences on future ones in an excitatory manner (Hawkes, 1971):

$$\lambda_t^m = \mu_0^m + \sum_{l=1}^{M} \sum_{t_i<t, m_i=l} \alpha^{ml} \exp\left(-\beta\left(t - t_i\right)\right),\ m \in [M],$$

where $[M] := \{1, 2, \ldots, M\}$. For this intensity process, we derive the following equivalent SDE formulation.

Theorem 6. *The intensity process $\{\boldsymbol{\lambda}_t\}_{t\geq 0}$ of the multivariate Hawkes process $\{\boldsymbol{N}_t\}_{t\geq 0}$ can be equivalently expressed as the solution to the jump SDEs*

$$\begin{cases} \mathrm{d}\lambda_t^m = \beta\left(\mu_0^m - \lambda_t^m\right)\mathrm{d}t + \sum_{l=1}^{M} \alpha^{ml}\,\mathrm{d}N_t^l, \\ \lambda_0^m = \mu_0^m, \quad m \in [M]. \end{cases}$$

The proof follows a similar approach to that given earlier for Theorem 2, and is omitted here.

Subsequently, we extend our approach to identify the unknown dynamics of general multivariate point processes. Let $\mathcal{S} = \{(t_i, m_i)\}_{i=1}^n$ be a multi-type event sequence, where each event (t_i, m_i) indicates that the i-th event occurs at time $t_i \in \mathbb{R}_+$ and is of type $m_i \in [M]$. We still denote $\boldsymbol{N}_t = (N_t^1, \ldots, N_t^M)^{\mathrm{T}}$ the associated multivariate counting process. Similar to Section 5.1, we introduce the M-dimensional log-intensity process $\boldsymbol{\eta}_t := \log \boldsymbol{\lambda}_t = (\log \lambda_t^1, \ldots, \log \lambda_t^M)^{\mathrm{T}}$. For each $m \in [M]$, we propose the NJDSDE for $\{\eta_t^m\}_{t\geq 0}$ as follows:

$$\mathrm{d}\eta_t^m = f^m(\boldsymbol{\eta}_t)\mathrm{d}t + \sum_{k=1}^{K} g^{mk}(\boldsymbol{\eta}_t)\mathrm{d}W_t^k + \sum_{l=1}^{M} h^{ml}(\boldsymbol{\eta}_t)\mathrm{d}N_t^l,$$

where the superscripts are used for the indices of vectors and matrices, such as the function g^{mk} and h^{ml} are the (m,k)-th component of the $M \times K$-matrix $\boldsymbol{g}_{\boldsymbol{\theta}_g}$ and the (m,l)-th component of the $M \times M$-matrix $\boldsymbol{h}_{\boldsymbol{\theta}_h} = \left[\boldsymbol{h}_{\boldsymbol{\theta}_h}^1 | \cdots | \boldsymbol{h}_{\boldsymbol{\theta}_h}^M\right]$ with $\boldsymbol{h}_{\boldsymbol{\theta}_h}^l$ as its l-th column vector, respectively. Moreover, the components W_t^k of $\boldsymbol{W}_t = (W_t^1, \ldots, W_t^K)^{\mathrm{T}}$ are standard Brownian motions which are pairwise independent. Suppose that $\boldsymbol{W}_t$ and $\boldsymbol{N}_t$ are independent.

We rewrite the above componentwise expression into vector form, leading to the formulation of the NJDSDE for $\{\boldsymbol{\eta}_t\}_{t\geq 0}$ as follows:

$$\begin{cases} \mathrm{d}\boldsymbol{\eta}_t = \underbrace{\boldsymbol{f}_{\boldsymbol{\theta}_f}(\boldsymbol{\eta}_t)}_{\text{drift net}}\,\mathrm{d}t + \underbrace{\boldsymbol{g}_{\boldsymbol{\theta}_g}(\boldsymbol{\eta}_t)}_{\text{diffusion net}}\,\mathrm{d}\boldsymbol{W}_t + \underbrace{\boldsymbol{h}_{\boldsymbol{\theta}_h}(\boldsymbol{\eta}_t)}_{\text{jump net}}\,\mathrm{d}\boldsymbol{N}_t, \\ \boldsymbol{\eta}_0 = \log \boldsymbol{\lambda}_0, \end{cases} \tag{18}$$

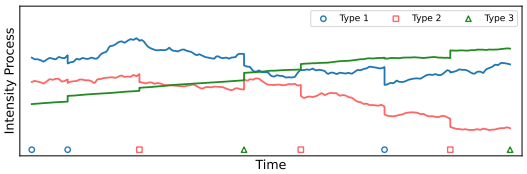

Figure 1. Illustration of the NJDSDE-governed intensity process with three event types. This example shows that the intensity process behaves as a diffusion process between event points, with a jump occurring at these specific points.

where $\boldsymbol{\eta}_0 \in \mathbb{R}^M$ is the initial value, the coefficient functions $\boldsymbol{f}_{\boldsymbol{\theta}_f} : \mathbb{R}^M \to \mathbb{R}^M$, $\boldsymbol{g}_{\boldsymbol{\theta}_g} : \mathbb{R}^M \to \mathbb{R}^{M\times K}$, and $\boldsymbol{h}_{\boldsymbol{\theta}_h} : \mathbb{R}^M \to \mathbb{R}^{M\times M}$ are the drift net, diffusion net, and jump net, respectively. In particular, when $M = K = 1$, this NJDSDE degenerates into Eq.(12). Essentially, Eq.(18) behaves as a neural SDE between event points, with a jump occurring at these points. This can be interpreted as:

$$\begin{cases} \mathrm{d}\boldsymbol{\eta}_t = \boldsymbol{f}_{\boldsymbol{\theta}_f}(\boldsymbol{\eta}_t)\,\mathrm{d}t + \boldsymbol{g}_{\boldsymbol{\theta}_g}(\boldsymbol{\eta}_t)\,\mathrm{d}\boldsymbol{W}_t, \quad t \in (t_{i-1}, t_i], \\ \boldsymbol{\eta}_{t_i+} = \boldsymbol{\eta}_{t_i} + \boldsymbol{h}_{\boldsymbol{\theta}_h}^{m_i}(\boldsymbol{\eta}_{t_i}), \quad t = t_i,\ m = m_i. \end{cases} \tag{19}$$

It is evident from this expression that the numerical value of l-th element in vector $\boldsymbol{h}_{\boldsymbol{\theta}_h}^{m_i}$ characterizes the influence magnitude of event type m_i on event type l, with the sign reflecting the excitatory or inhibitory influence.

As discussed in Section 5.1, we employ Lipschitz continuous activations in the neural networks $\boldsymbol{f}_{\boldsymbol{\theta}_f}$, $\boldsymbol{g}_{\boldsymbol{\theta}_g}$, and $\boldsymbol{h}_{\boldsymbol{\theta}_h}$. This ensures the existence of a unique strong solution to Eq.(18). We also apply MLE to learn model parameters and provide the complete training algorithm in Appendix B. Figure 1 presents a concrete example of the intensity process governed by the NJDSDE Eq.(18).

5.4. Event Prediction

With the proposed NJDSDE-governed intensity process, we aim to predict the next event time and next event type given the historical events $\mathcal{H}_{t_{n+1}} = \{(t_i, m_i)\}_{i=1}^n$.

The conditional density function of the next event time (Rasmussen, 2018) is that for $t \geq t_n$,

$$f_{t_{n+1}}(t) = \lambda_t^{\mathrm{g}} \exp(-\int_{t_n}^{t} \lambda_s^{\mathrm{g}}\,\mathrm{d}s), \tag{20}$$

where $\lambda_t^{\mathrm{g}} := \sum_{m=1}^M \lambda_t^m = \sum_{m=1}^M \exp\left(\eta_t^m\right)$. Then for the next event time prediction, we employ the formula

$$\widehat{t}_{n+1} = \int_{t_n}^{\infty} t\, f_{t_{n+1}}(t)\,\mathrm{d}t. \tag{21}$$

Note that after the jump time t_n, the process $\boldsymbol{\eta}_t$ governed by Eq.(18) will have the dynamics

$$\begin{cases} \mathrm{d}\boldsymbol{\eta}_t = \boldsymbol{f}_{\boldsymbol{\theta}_f}(\boldsymbol{\eta}_t)\,\mathrm{d}t + \boldsymbol{g}_{\boldsymbol{\theta}_g}(\boldsymbol{\eta}_t)\,\mathrm{d}\boldsymbol{W}_t, \quad t > t_n, \\ \boldsymbol{\eta}_{t_n+} = \boldsymbol{\eta}_{t_n} + \boldsymbol{h}_{\boldsymbol{\theta}_h}^{m_n}(\boldsymbol{\eta}_{t_n}). \end{cases} \tag{22}$$

Therefore, similar to the method discussed in Section 5.2 for computing the log-likelihood function, we utilize the Euler-Maruyama scheme to discretize Eq.(22), followed by numerical integration techniques to compute the integrals mentioned above. Here, the value of $\boldsymbol{\eta}_{t_n}$ required for discretizing Eq.(22) can be obtained by discretizing Eq.(18) over the interval $[0, t_n]$ using the Euler-Maruyama scheme and the historical events $\mathcal{H}_{t_{n+1}}$.

Following previous works (Zuo et al., 2020; Shi et al., 2023; Xue et al., 2024), the next event type prediction is given by

$$\widehat{m}_{n+1} = \operatorname{argmax}_m \lambda^m_{t_{n+1}} / \lambda^{\mathrm{g}}_{t_{n+1}}. \tag{23}$$

6. Experiments

We first test the flexibility of our NJDTPP model by recovering the ground truth dynamics of the intensity process of classical TPPs. Then, we evaluate the modeling capability for event sequences and the prediction performance of NJDTPP on six real-world datasets. Our code is available at https://github.com/Zh-Shuai/NJDTPP.

6.1. Intensity Process Recovery for Classical TPPs

Synthetic Datasets. We consider the following classical TPPs: (i) Poisson Process: the intensity is given by $\lambda_t = \lambda_0$, where $\lambda_0 = 1.0$; (ii) Hawkes Process: the intensity is given by $\lambda_t = \mu + \alpha \sum_{t_i<t} \exp\left(-\beta(t - t_i)\right)$, where $\mu = 0.2$, $\alpha = 0.8$, $\beta = 1.0$; and (iii) Self-Correcting Process: the intensity is given by $\lambda_t = \exp\left(\mu t - \sum_{t_i<t} \alpha\right)$, where $\mu = 0.5$, $\alpha = 0.2$. For each TPP, we simulate a dataset using the Ogata's thinning algorithm (Ogata, 1981). Each dataset contains 500 event sequences within the time interval $[0, 100]$. The train-validation-test data split is $3:1:1$.

Experimental Setup. We fit our NJDTPP model to each dataset using the training procedure described in Section 5.2. In this experiment, the drift, diffusion, and jump nets are implemented as three MLPs, each with 2 hidden layers of 32 units. The activation function chosen for these networks is Tanh. More training details are reported in Appendix C.5. For evaluation, we visually demonstrate the similarity between the estimated intensity from the learned NJDTPP and the ground truth intensity.

In addition, we compare the mean absolute percentage error (MAPE) of the estimated intensity of our model with the Poisson process (PP) model, the Hawkes process (HP) model, the self-correcting process (SC) model, an RNN-based model (Jia & Benson, 2019), and the Neural Jump SDE (NJSDE, Jia & Benson (2019)) model. The baseline results in Table 1 are extracted from (Jia & Benson, 2019). Similar to the calculation of the log-likelihood function, we first numerically solve Eq.(12) using the Euler-Maruyama scheme, and then compute MAPE through numerical integration. See Appendix C.4 for the definition of MAPE.

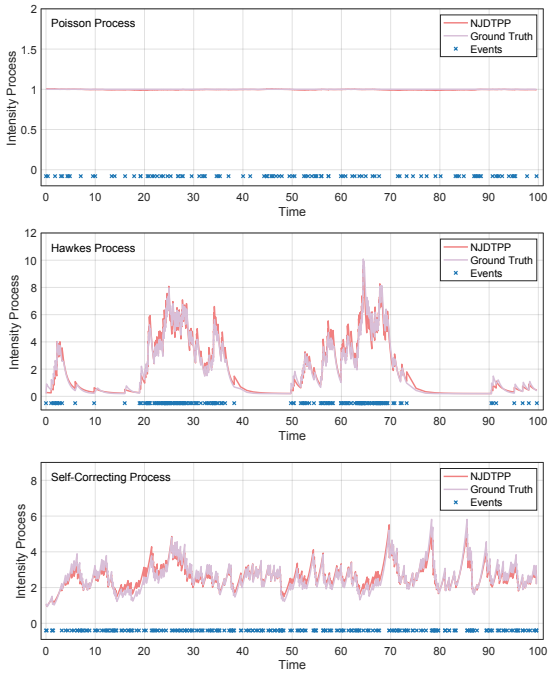

Figure 2. The estimated and ground truth intensity process of the Poisson Process, Hawkes Process, and Self-Correcting Process. Each blue cross represents an event at the corresponding time.

Table 1. The MAPE comparison of the estimated intensity. Each row represents a synthetic dataset. Each column represents a model.

	PP	HP	SC	RNN	NJSDE	NJDTPP
Poisson	**0.1**	0.3	98.7	3.2	1.3	**0.1**
Hawkes	188.2	3.5	101.0	22.0	5.9	**0.2**
Self-Correcting	29.1	29.1	1.6	24.3	9.3	**0.1**

Results. Figure 2 compares the estimated intensity process (red curve) of our NJDTPP model with the ground truth (grayish purple curve). It clearly shows that NJDTPP effectively recovers the dynamics of ground truth intensities. This also indicates that our model can capture the excitatory and inhibitory influences between events. Table 1 reports the estimated MAPE for NJDTPP and the baseline models. These values are the average results for all 100 event sequences in the test set. As shown, our model fits the data well and shows a substantial performance improvement compared to baselines.

Having observed that NJDTPP achieves superior experimental results, we now turn to analyze the two main factors that contributed to this success. Firstly, as discussed in Section 5.1, these classical TPPs are special cases of our modeling framework. Specifically, when the drift, diffusion, and jump functions take certain forms, our proposed NJDSDE Eq.(12) characterizes these classical TPPs. Secondly, we employ neural networks to parameterize the drift, diffusion, and jump functions in Eq.(12). Due to the powerful capability of neural networks, our model can fit the data well and effectively recover the ground truth intensity.

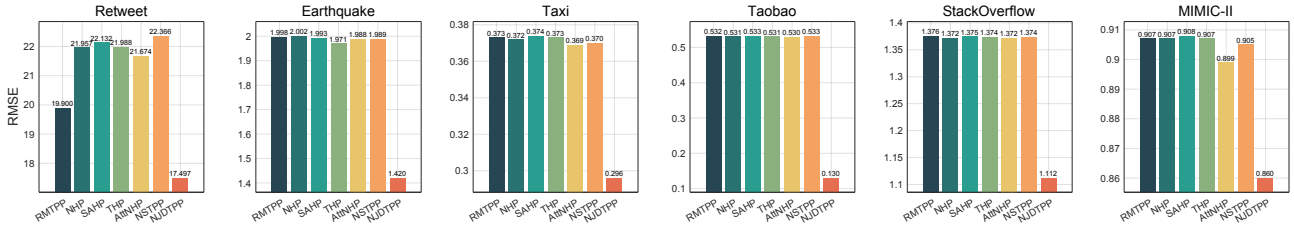

Figure 3. Event time prediction RMSE comparison.

Table 2. Event type prediction accuracy and F_1 comparison.

Model	Retweet		Earthquake		Taxi		Taobao		StackOverflow		MIMIC-II	
	Acc	F_1	Acc	F_1	Acc	F_1	Acc	F_1	Acc	F_1	Acc	F_1
RMTPP	0.503	0.415	0.214	0.082	0.836	0.815	0.431	0.429	0.437	0.265	0.812	0.600
NHP	0.601	0.573	0.451	0.283	0.890	0.886	0.463	0.447	0.467	0.315	0.832	0.680
SAHP	0.522	0.497	0.417	0.246	0.847	0.802	0.442	0.320	0.452	0.301	0.827	0.639
THP	0.536	0.375	0.451	0.281	0.865	0.828	0.467	0.406	0.463	0.309	0.853	0.596
AttNHP	0.592	0.575	0.452	0.283	0.761	0.724	0.458	0.386	0.465	0.310	0.856	0.817
NSTPP	0.513	0.482	0.354	0.165	0.884	0.837	0.437	0.289	0.449	0.293	0.840	0.799
NJDTPP	**0.608**	**0.584**	**0.472**	**0.313**	**0.908**	**0.891**	**0.486**	**0.452**	**0.469**	**0.327**	**0.861**	**0.823**

Table 3. Negative log-likelihood comparison.

Model	Retweet	Earthquake	Taxi	Taobao	StackOverflow	MIMIC-II
RMTPP	4.241	3.653	0.227	1.659	2.891	2.333
NHP	4.137	2.189	0.208	0.986	2.496	2.205
SAHP	5.009	3.941	0.478	1.640	2.952	3.394
THP	4.560	3.387	0.442	1.191	2.630	1.515
AttNHP	4.756	2.376	0.491	1.206	2.586	1.697
NSTPP	4.527	2.203	0.217	1.415	2.541	2.421
NJDTPP	**4.092**	**1.305**	**-0.293**	**-0.440**	**2.347**	**1.398**

Table 4. Performance of the NJDTPP variant on Earthquake and Taxi. NJDTPP-BM refers to the variant without Brownian motion.

Model	Earthquake				Taxi			
	NLL	RMSE	Acc	F_1	NLL	RMSE	Acc	F_1
NJDTPP-BM	1.594	1.507	0.470	0.306	-0.290	0.302	0.905	0.887
NJDTPP	**1.305**	**1.420**	**0.472**	**0.313**	**-0.293**	**0.296**	**0.908**	**0.891**

6.2. Likelihood Evaluation and Event Prediction

We use the negative log-likelihood (NLL) as a metric to evaluate the ability of NJDTPP in modeling event sequences on real-world datasets. Moreover, we evaluate the performance of NJDTPP in the standard next-event prediction task in TPPs, predicting every next event (t_i, m_i) given the history $\mathcal{H}_{t_i}$. We use Eq.(21) and Eq.(23) to predict the next event time and type, respectively. We evaluate event time prediction by Root Mean Square Error (RMSE) and event type prediction by accuracy and the weighted F_1 score. The training details and the configuration of our NJDTPP model are provided in Appendix C.5 and Appendix C.6.

Datasets. We evaluate our model on six real-world benchmark datasets: **Retweet** (Zhou et al., 2013), **Earthquake** (Xue et al., 2024), **Taxi** (Whong, 2014), **Taobao** (Xue et al., 2022), **StackOverflow** (Leskovec & Krevl, 2014), and **MIMIC-II** (Johnson et al., 2016). The **MIMIC-II** dataset is available at the public Github repository[2], and all other datasets are available at the public `EasyTPP`[3] library (Xue et al., 2024), an open benchmark for evaluating TPPs. See Appendix C.2 for dataset details.

[2] https://github.com/hongyuanmei/neurawkes

[3] https://github.com/ant-research/EasyTemporalPointProcess

Baselines. We compare NJDTPP with the following models. Two RNN-based models: Recurrent Marked Temporal Point Process (**RMTPP**, Du et al. (2016)) and Neural Hawkes Process (**NHP**, Mei & Eisner (2017)). Three attention-based models: Self-Attentive Hawkes Process (**SAHP**, Zhang et al. (2020a)), Transformer Hawkes Process (**THP**, Zuo et al. (2020)), and Attentive Neural Hawkes Process (**AttNHP**, Yang et al. (2022)). One TPP with the hidden state governed by a neural jump SDE: Neural Spatio-Temporal Point Process (**NSTPP**, Chen et al. (2020)). More details in Appendix C.3. For the implementation of baselines, we use the code from `EasyTPP`[3] (Xue et al., 2024), and extract partial baseline results from the same paper.

Results. Table 3 summarizes the per-event NLL of these models on each test set. From this table, we can observe that NJDTPP fits the data well and significantly outperforms baselines across all experiments. This demonstrates our model's capability to learn complex real-world intensity dynamics. The results for next event time and event type prediction are presented in Figure 3 and Table 2 respectively. It is evident that, in each dataset, NJDTPP outperforms all competing models, often by a substantial margin. This shows the superior performance of our model in prediction tasks. The success of NJDTPP can be attributed to its flexibility in capturing complex intensity dynamics and the influences between events.

6.3. Ablation Study

We conduct an ablation study on the Earthquake and Taxi datasets, investigating the variant of NJDTPP by removing the diffusion term. We evaluate models based on NLL and prediction performance. Table 4 reports the experimental results. As shown, the diffusion term contributes to model performance due to the fact that it models the Gaussian noise with the Brownian motion.

7. Conclusion

We have presented Neural Jump-Diffusion Temporal Point Processes, a unified TPP framework that can learn a free-form intensity process consistent with the observed event data. By modeling the intensity process as the solution to an SDE, our approach eliminates the need to pre-specify the functional form of the intensity function, thereby significantly enhancing the flexibility and capability of TPP models. Experimental results show that our model effectively captures intensity dynamics and the influences between events, as well as achieves state-of-the-art results on benchmark datasets in likelihood evaluation and event prediction tasks.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was partially supported by the National Key Research and Development Program of China (2021YFB3100600), the Strategic Priority Research Program of the Chinese Academy of Sciences (XDB0680101), the NSFC (62376064, U2336202), and the CAS Project for Young Scientists in Basic Research (YSBR-008).

Impact Statement

We develop a novel framework for modeling the intensity process of temporal point processes using neural jump-diffusion stochastic differential equations. We hope that our study will inspire new developments in the field of temporal point processes. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Anil, C., Lucas, J., and Grosse, R. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pp. 291–301. PMLR, 2019.

Bertoin, J. *Lévy processes*. Cambridge University Press, 1996.

Björk, T. *Point Processes and Jump Diffusions: An introduction with finance applications*. Cambridge University Press, 2021.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31, 2018.

Chen, R. T., Amos, B., and Nickel, M. Neural spatio-temporal point processes. *International Conference on Learning Representations*, 2020.

Cyganowski, S., Grüne, L., and Kloeden, P. E. *MAPLE for jump—diffusion stochastic differential equations in finance*. Springer, 2002.

Daley, D. J., Vere-Jones, D., et al. *An introduction to the theory of point processes: volume I: elementary theory and methods*. Springer, 2003.

De, A., Valera, I., Ganguly, N., Bhattacharya, S., and Gomez Rodriguez, M. Learning and forecasting opinion dynamics in social networks. *Advances in Neural Information Processing Systems*, 29, 2016.

Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., and Song, L. 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*, pp. 1555–1564, 2016.

Farajtabar, M., Wang, Y., Gomez-Rodriguez, M., Li, S., Zha, H., and Song, L. Coevolve: A joint point process model for information diffusion and network evolution. *Journal of Machine Learning Research*, 18(41):1–49, 2017.

Gao, T., Subramanian, D., Bhattacharjya, D., Shou, X., Mattei, N., and Bennett, K. P. Causal inference for event pairs in multivariate point processes. *Advances in Neural Information Processing Systems*, 34:17311–17324, 2021.

Hanson, F. B. *Applied stochastic processes and control for jump-diffusions: modeling, analysis and computation*. SIAM, 2007.

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

Herrera, C., Krach, F., and Teichmann, J. Neural jump ordinary differential equations: Consistent continuous-time prediction and filtering. *International Conference on Learning Representations*, 2020.

Ikeda, N. and Watanabe, S. *Stochastic differential equations and diffusion processes*. Elsevier, 2014.

Isham, V. and Westcott, M. A self-correcting point process. *Stochastic processes and their applications*, 8(3):335–347, 1979.

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

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., and Mark, R. G. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

Kidger, P., Morrill, J., Foster, J., and Lyons, T. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33: 6696–6707, 2020.

Kidger, P., Foster, J., Li, X., and Lyons, T. J. Neural sdes as infinite-dimensional gans. In *International Conference on Machine Learning*, pp. 5453–5463. PMLR, 2021a.

Kidger, P., Foster, J., Li, X. C., and Lyons, T. Efficient and accurate gradients for neural sdes. *Advances in Neural Information Processing Systems*, 34:18747–18761, 2021b.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Kingman, J. F. C. *Poisson processes*, volume 3. Clarendon Press, 1992.

Kloeden, P. E. and Platen, E. *Numerical solution of stochastic differential equations*. Springer-Verlag Berlin, 1992.

Kong, L., Sun, J., and Zhang, C. Sde-net: Equipping deep neural networks with uncertainty estimates. In *International Conference on Machine Learning*, pp. 5405–5415. PMLR, 2020.

Kuo, H.-H. *Stochastic differential equations*. Springer, 2006.

Lamberton, D. and Lapeyre, B. *Introduction to stochastic calculus applied to finance*. CRC press, 2011.

Le Gall, J.-F. *Brownian motion, martingales, and stochastic calculus*. Springer, 2016.

Leskovec, J. and Krevl, A. Snap datasets: Stanford large network dataset collection. 2014.

Li, X., Wong, T.-K. L., Chen, R. T., and Duvenaud, D. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pp. 3870–3882. PMLR, 2020.

Lin, X., Cao, J., Zhang, P., Zhou, C., Li, Z., Wu, J., and Wang, B. Disentangled deep multivariate hawkes process for learning event sequences. In *International Conference on Data Mining*, pp. 360–369. IEEE, 2021.

Lin, X., Zhang, W., Shi, F., Zhou, C., Zou, L., Zhao, X., Yin, D., Pan, S., and Cao, Y. Graph neural stochastic diffusion for estimating uncertainty in node classification. In *International Conference on Machine Learning*, 2024.

Liu, S. and Hauskrecht, M. Event outlier detection in continuous time. In *International Conference on Machine Learning*, pp. 6793–6803. PMLR, 2021.

Mei, H. and Eisner, J. M. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in Neural Information Processing Systems*, 30, 2017.

Møller, J., Syversveen, A. R., and Waagepetersen, R. P. Log gaussian cox processes. *Scandinavian journal of statistics*, 25(3):451–482, 1998.

Oakes, D. The markovian self-exciting process. *Journal of Applied Probability*, 12(1):69–77, 1975.

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

Oh, Y., Lim, D., and Kim, S. Stable neural stochastic differential equations in analyzing irregular time series data. In *International Conference on Learning Representations*, 2024.

Oksendal, B. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

Omi, T., Aihara, K., et al. Fully neural network based model for general temporal point processes. *Advances in Neural Information Processing Systems*, 32, 2019.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. 2017.

Rasmussen, J. G. Lecture notes: Temporal point processes and the conditional intensity function. *arXiv preprint arXiv:1806.00221*, 2018.

Rubanova, Y., Chen, R. T., and Duvenaud, D. K. Latent ordinary differential equations for irregularly-sampled time series. *Advances in Neural Information Processing Systems*, 32, 2019.

Shchur, O., Biloš, M., and Günnemann, S. Intensity-free learning of temporal point processes. *International Conference on Learning Representations*, 2019.

Shchur, O., Turkmen, A. C., Januschowski, T., Gasthaus, J., and Günnemann, S. Detecting anomalous event sequences with temporal point processes. *Advances in Neural Information Processing Systems*, 34:13419–13431, 2021.

Shi, X., Xue, S., Wang, K., Zhou, F., Zhang, J. Y., Zhou, J., Tan, C., and Mei, H. Language models can improve event prediction by few-shot abductive reasoning. *Advances in Neural Information Processing Systems*, 2023.

Song, Y., Donghyun, L., Meng, R., and Kim, W. H. Decoupled marked temporal point process using neural ordinary differential equations. *International Conference on Learning Representations*, 2024.

Wang, Y., Williams, G., Theodorou, E., and Song, L. Variational policy for guiding point processes. In *International Conference on Machine Learning*, pp. 3684–3693. PMLR, 2017.

Wang, Y., Theodorou, E., Verma, A., and Song, L. A stochastic differential equation framework for guiding online user activities in closed loop. In *International Conference on Artificial Intelligence and Statistics*, pp. 1077–1086. PMLR, 2018.

Whong, C. FOILing NYC's taxi trip data. 2014.

Xiao, S., Yan, J., Yang, X., Zha, H., and Chu, S. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Xu, H., Farajtabar, M., and Zha, H. Learning granger causality for hawkes processes. In *International Conference on Machine Learning*, pp. 1717–1726. PMLR, 2016.

Xue, S., Shi, X., Zhang, J., and Mei, H. Hypro: A hybridly normalized probabilistic model for long-horizon prediction of event sequences. *Advances in Neural Information Processing Systems*, 35:34641–34650, 2022.

Xue, S., Shi, X., Chu, Z., Wang, Y., Zhou, F., Hao, H., Jiang, C., Pan, C., Xu, Y., Zhang, J. Y., et al. Easytpp: Towards open benchmarking the temporal point processes. *International Conference on Learning Representations*, 2024.

Yang, C., Mei, H., and Eisner, J. Transformer embeddings of irregularly spaced events and their participants. In *International Conference on Learning Representations*, 2022.

Zarezade, A., De, A., Upadhyay, U., Rabiee, H. R., and Gomez-Rodriguez, M. Steering social activity: A stochastic optimal control point of view. *J. Mach. Learn. Res.*, 18:205–1, 2017.

Zhang, Q., Lipani, A., Kirnap, O., and Yilmaz, E. Self-attentive hawkes process. In *International Conference on Machine Learning*, pp. 11183–11193. PMLR, 2020a.

Zhang, S., Zhou, C., Zhang, P., Liu, Y., Li, Z., and Chen, H. Multiple hypothesis testing for anomaly detection in multi-type event sequences. In *International Conference on Data Mining*, pp. 808–817. IEEE, 2023.

Zhang, W., Panum, T., Jha, S., Chalasani, P., and Page, D. Cause: Learning granger causality from event sequences using attribution methods. In *International Conference on Machine Learning*, pp. 11235–11245. PMLR, 2020b.

Zhou, K., Zha, H., and Song, L. Learning triggering kernels for multi-dimensional hawkes processes. In *International Conference on Machine Learning*, pp. 1301–1309. PMLR, 2013.

Zuo, S., Jiang, H., Li, Z., Zhao, T., and Zha, H. Transformer hawkes process. In *International Conference on Machine Learning*, pp. 11692–11702. PMLR, 2020.

Appendix

A. Proofs

A.1. Proof of Theorem 2

Proof. To solve the SDE Eq.(9), we first denote the jump times of the Hawkes process $\{N_t\}_{t\geq 0}$ as $\{t_i\}_{i=1}^{\infty}$. On the time interval $(t_{i-1}, t_i]$, Eq.(9) behaves as an ODE $\mathrm{d}\lambda_t = \beta(\mu - \lambda_t)\,\mathrm{d}t$. On the other hand, at a jump time t_i, the jump size of $\{\lambda_t\}_{t\geq 0}$ is given by $\Delta\lambda_{t_i} = \lambda_{t_i+} - \lambda_{t_i} = \alpha\,\Delta N_{t_i} = \alpha$. The right-limit at t_i is then $\lambda_{t_i+} = \lambda_{t_i} + \alpha$.

Up to the first jump time t_1, the solution follows the ODE $\mathrm{d}\lambda_t = \beta(\mu - \lambda_t)\,\mathrm{d}t$ on $[0, t_1]$ with the initial value $\lambda_0 = \mu$. Solving this ODE yields $\lambda_t = \mu - C_1 e^{-\beta t}$, where C_1 is a constant. Using the initial condition $\lambda_0 = \mu$, we find that $C_1 = 0$. Thus, we have $\lambda_t = \mu$ for all $t \in [0, t_1]$. In particular, $\lambda_{t_1} = \mu$ and $\lambda_{t_1+} = \lambda_{t_1} + \alpha = \mu + \alpha$.

Subsequently, we solve the ODE $\mathrm{d}\lambda_t = \beta(\mu - \lambda_t)\,\mathrm{d}t$ on $(t_1, t_2]$ with $\lambda_{t_1+} = \mu + \alpha$ to obtain $\lambda_t = \mu - C_2 e^{-\beta t}$, where C_2 is a constant. Using $\lambda_{t_1+} = \mu + \alpha$ and taking the right-limit at t_1 in the above equation, we obtain $\lambda_{t_1+} = \mu + \alpha = \mu - C_2 e^{-\beta t_1}$. Then $C_2 = -\alpha e^{\beta t_1}$. Thus, we get $\lambda_t = \mu + \alpha e^{-\beta(t - t_1)}$ for $t \in (t_1, t_2]$. This allows us to determine $\lambda_{t_2} = \mu + \alpha e^{-\beta(t_2 - t_1)}$ and $\lambda_{t_2+} = \lambda_{t_2} + \alpha = \mu + \alpha e^{-\beta(t_2 - t_1)} + \alpha$.

Iterating this procedure, the solution is given by $\lambda_t = \mu + \alpha \sum_{i:\, t_i < t} \exp\left(-\beta\,(t - t_i)\right)$, which is the intensity process of the Hawkes process. □

A.2. Proof of Theorem 3

Proof. To solve the SDE Eq.(10), we first denote the jump times of the self-correcting process $\{N_t\}_{t\geq 0}$ as $\{t_i\}_{i=1}^{\infty}$. On the time interval $(t_{i-1}, t_i]$, Eq.(10) behaves as an ODE $\mathrm{d}\lambda_t = \mu\lambda_t\,\mathrm{d}t$. On the other hand, at a jump time t_i, the jump size of $\{\lambda_t\}_{t\geq 0}$ is given by $\Delta\lambda_{t_i} = \lambda_{t_i+} - \lambda_{t_i} = (e^{-\alpha} - 1)\,\lambda_{t_i}\,\Delta N_{t_i} = (e^{-\alpha} - 1)\,\lambda_{t_i}$. The right-limit at t_i is then $\lambda_{t_i+} = e^{-\alpha}\lambda_{t_i}$.

Up to the first jump time t_1, the solution follows the ODE $\mathrm{d}\lambda_t = \mu\lambda_t\,\mathrm{d}t$ on $[0, t_1]$ with the initial value $\lambda_0 = 1$. Solving this ODE yields $\lambda_t = C_1 e^{\mu t}$, where C_1 is a constant. Using the initial condition $\lambda_0 = 1$, we can find that $C_1 = 1$. Thus, we have $\lambda_t = e^{\mu t}$ for $t \in [0, t_1]$. In particular, $\lambda_{t_1} = e^{\mu t_1}$ and $\lambda_{t_1+} = e^{-\alpha}\lambda_{t_1} = e^{\mu t_1 - \alpha}$.

Subsequently, we solve the ODE $\mathrm{d}\lambda_t = \mu\lambda_t\,\mathrm{d}t$ on $(t_1, t_2]$ with $\lambda_{t_1+} = e^{\mu t_1 - \alpha}$ to obtain $\lambda_t = C_2 e^{\mu t}$, where C_2 is a constant. Using $\lambda_{t_1+} = e^{\mu t_1 - \alpha}$ and taking the right-limit at t_1 in the above equation, we obtain $\lambda_{t_1+} = e^{\mu t_1 - \alpha} = C_2 e^{\mu t_1}$. Then $C_2 = e^{-\alpha}$. Thus, we get $\lambda_t = e^{\mu t - \alpha}$ for $t \in (t_1, t_2]$. This allows us to determine $\lambda_{t_2} = e^{\mu t_2 - \alpha}$ and $\lambda_{t_2+} = e^{-\alpha}\lambda_{t_2} = e^{\mu t_2 - 2\alpha}$.

Iterating this procedure, the solution is given by $\lambda_t = \exp(\mu t - \sum_{i:\, t_i < t} \alpha)$, which is the intensity process of the self-correcting process. □

A.3. Proof of Theorem 4

Proof. **Step 1**: For every $\lambda > 0$, we shall prove the existence and uniqueness of the solution to the SDE given by

$$\begin{cases} \mathrm{d}\eta_t = \dfrac{f\left(e^{\eta_t}\right)}{e^{\eta_t}}\,\mathrm{d}t + \log\dfrac{e^{\eta_t} + h\left(e^{\eta_t}\right)}{e^{\eta_t}}\,\mathrm{d}N_t, \\ \eta_0 = \log\lambda, \end{cases} \tag{24}$$

where the logarithmic function is well-defined due to the condition $h(x) + x > 0$ for all $x > 0$.

To initiate the proof, we denote the jump times of $\{N_t\}_{t\geq 0}$ as $\{t_i\}_{i=1}^{\infty}$. Since the ODE

$$\begin{cases} \mathrm{d}y_t = \dfrac{f\left(e^{y_t}\right)}{e^{y_t}}\,\mathrm{d}t, \quad t \geq 0, \\ y_0 = y, \end{cases} \tag{25}$$

has a unique global solution for every $y \in \mathbb{R}$, we can solve the ODE $\mathrm{d}\eta_t = \frac{f(e^{\eta_t})}{e^{\eta_t}}\,\mathrm{d}t$ on the interval $[0, t_1]$ with the initial value $\eta_0 = \log\lambda$ for every $\lambda > 0$. In particular, we can obtain the value of η_{t_1} and then $\eta_{t_1+} = \eta_{t_1} + \log\frac{e^{\eta_{t_1}} + h\left(e^{\eta_{t_1}}\right)}{e^{\eta_{t_1}}} = \log(e^{\eta_{t_1}} + h(e^{\eta_{t_1}}))$.

Since the ODE Eq.(25) is autonomous, we can similarly solve the ODE $\mathrm{d}\eta_t = \frac{f(e^{\eta_t})}{e^{\eta_t}}\,\mathrm{d}t$ on $(t_1, t_2]$ with the initial value η_{t_1+}. This yields η_{t_2} and then $\eta_{t_2+} = \eta_{t_2} + \log\frac{e^{\eta_{t_2}}+h\left(e^{\eta_{t_2}}\right)}{e^{\eta_{t_2}}} = \log(e^{\eta_{t_2}} + h(e^{\eta_{t_2}}))$.

Iterating this procedure, we have proved that Eq.(24) has a unique global solution for every $\lambda > 0$.

Step 2: We now aim to prove that $\lambda_t = e^{\eta_t}$ is the unique global positive solution of the SDE Eq.(11), where $\{\eta_t\}_{t\geq 0}$ is the solution of Eq.(24).

Note that $\lambda_t = e^{\eta_t}$ ensures that λ_t is positive and $\lambda_0 = e^{\eta_0} = \lambda$. According to Eq.(24), between the jumps of $\{N_t\}_{t\geq 0}$, the process $\{\eta_t\}_{t\geq 0}$ follows the dynamics $\mathrm{d}\eta_t = \frac{f(e^{\eta_t})}{e^{\eta_t}}\,\mathrm{d}t$. Therefore, the dynamics of λ_t can be expressed as follows:

$$\mathrm{d}\lambda_t = \mathrm{d}e^{\eta_t} = e^{\eta_t}\,\mathrm{d}\eta_t = f(e^{\eta_t})\,\mathrm{d}t = f(\lambda_t)\,\mathrm{d}t. \tag{26}$$

On the other hand, at a jump time t, the process $\{N_t\}_{t\geq 0}$ has a jump size of $\Delta N_t = N_{t+} - N_t = 1$, implying that the process $\{\eta_t\}_{t\geq 0}$ will have a jump of size $\Delta\eta_t = \log\frac{e^{\eta_t}+h(e^{\eta_t})}{e^{\eta_t}}\Delta N_t = \log\frac{e^{\eta_t}+h(e^{\eta_t})}{e^{\eta_t}}$. Then $\eta_{t+} = \eta_t + \Delta\eta_t = \eta_t + \log\frac{e^{\eta_t}+h(e^{\eta_t})}{e^{\eta_t}} = \log(e^{\eta_t} + h(e^{\eta_t}))$. Since $\lambda_t = e^{\eta_t}$, the induced jump size of λ_t is given by

$$\Delta\lambda_t = e^{\eta_{t+}} - e^{\eta_t} = e^{\log(e^{\eta_t}+h(e^{\eta_t}))} - e^{\eta_t} = h(e^{\eta_t}) = h(\lambda_t). \tag{27}$$

Combining Eq.(26) and Eq.(27), the equation holds: $\mathrm{d}\lambda_t = f(\lambda_t)\,\mathrm{d}t + h(\lambda_t)\,\mathrm{d}N_t$. Therefore, we establish that the SDE Eq.(11) given by

$$\begin{cases} \mathrm{d}\lambda_t = f\left(\lambda_t\right)\mathrm{d}t + h\left(\lambda_t\right)\mathrm{d}N_t, \\ \lambda_0 = \lambda, \end{cases}$$

has a unique global positive solution for every $\lambda > 0$. □

A.4. Proofs of Special Cases of Our Proposed NJDSDE

Proof. We first provide the proof for the case of Hawkes processes, and the proofs for self-correcting processes and Poisson processes are similar.

According to Theorem 2, between the jumps of the Hawkes process $\{N_t\}_{t\geq 0}$, the intensity process $\{\lambda_t\}_{t\geq 0}$ follows the dynamics $\mathrm{d}\lambda_t = \beta(\mu - \lambda_t)\,\mathrm{d}t$. Therefore, the dynamics of $\eta_t := \log(\lambda_t)$ can be expressed as follows:

$$\mathrm{d}\eta_t = \mathrm{d}\log(\lambda_t) = \frac{1}{\lambda_t}\,\mathrm{d}\lambda_t = \frac{\beta(\mu-\lambda_t)}{\lambda_t}\,\mathrm{d}t = \frac{\beta(\mu - e^{\eta_t})}{e^{\eta_t}}\,\mathrm{d}t = \beta(\mu e^{-\eta_t} - 1)\,\mathrm{d}t. \tag{28}$$

On the other hand, at a jump time t, the intensity process $\{\lambda_t\}_{t\geq 0}$ has a jump of size $\Delta\lambda_t = \alpha$. Then $\lambda_{t+} = \lambda_t + \Delta\lambda_t = \lambda_t + \alpha$. Since $\eta_t = \log(\lambda_t)$, the induced jump size of η_t is given by

$$\Delta\eta_t = \log(\lambda_{t+}) - \log(\lambda_t) = \log(\frac{\lambda_t+\alpha}{\lambda_t}) = \log(\frac{e^{\eta_t}+\alpha}{e^{\eta_t}}) = \log(1 + \alpha e^{-\eta_t}). \tag{29}$$

Combining Eq.(28) and Eq.(29), the equation holds: $\mathrm{d}\eta_t = \beta(\mu e^{-\eta_t} - 1)\,\mathrm{d}t + \log(1+\alpha e^{-\eta_t})\,\mathrm{d}N_t$.

Therefore, when the function $f_{\theta_f}(\eta_t)$ is set to $\beta(\mu e^{-\eta_t} - 1)$, $g_{\theta_g}(\eta_t)$ is set to 0, and $h_{\theta_h}(\eta_t)$ is set to $\log(1+\alpha e^{-\eta_t})$ in Eq.(12), the proposed NJDSDE characterizes Hawkes processes.

Now, we prove that a specific class (but not all) of log-Gaussian Cox processes (LGCPs) can be incorporated into our modeling framework Eq.(12).

Specifically, when f_{θ_f} takes 0, g_{θ_g} takes 1, and h_{θ_h} takes 0 in Eq.(12), the NJDSDE reduces to $\mathrm{d}\eta_t = \mathrm{d}W_t$. Given the initial value $\eta_0 = \log\lambda_0$ and $W_0 = 0$, we have $\eta_t = W_t + \log\lambda_0$. Since $\lambda_t = \exp(\eta_t)$, it follows that $\lambda_t = \lambda_0\exp(W_t)$. Taking λ_0 as 1, we obtain $\lambda_t = \exp(W_t)$, which indeed represents a specific class of LGCPs, since the Brownian motion W_t is a Gaussian process with independent stationary increments (Bertoin, 1996). In turn, since a Gaussian process is not necessarily a Brownian motion, for example when the increments of a Gaussian process do not satisfy independence, our framework Eq.(12) cannot incorporate all LGCPs. □

A.5. Proof of Theorem 5

Proof. The following proof is adapted from the Theorem 9.1 in (Ikeda & Watanabe, 2014).

Consider a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ on which we define a Brownian motion $W = \{W_t\}_{t\geq 0}$ and a counting process $N = \{N_t\}_{t\geq 0}$ that jumps at the times $\{t_i\}_{i=1}^{\infty}$. Suppose that W and N are independent. We define the filtration $\{\mathcal{F}_t\}_{t\geq 0}$ as the augmented natural filtration of W and N, i.e., for all $t \geq 0$ we set $\mathcal{F}_t = \sigma\left(\{(W_s, N_s) : s \leq t\} \cup \mathcal{N}\right)$, where $\mathcal{N} = \{A \in \mathcal{F} : \mathbb{P}(A) = 0\}$. With this, $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, \mathbb{P})$ is a filtered probability space that satisfies the usual conditions.

It is easy to see that the jump times $\{t_i\}_{i=1}^{\infty}$ are stopping times of $\{\mathcal{F}_t\}_{t\geq 0}$ since $\{t_i \leq t\} = \{N_t \geq i\} \in \mathcal{F}_t$, and $\lim_{i\to\infty} t_i = \infty$ a.s. First we shall show the existence and uniqueness of solutions in the time interval $[0, t_1]$. For this, consider the following equation

$$\mathrm{d}X(t) = f_{\theta_f}(X(s))\,\mathrm{d}s + g_{\theta_g}(X(s))\,\mathrm{d}W_s, \quad X(0) = \eta_0. \tag{30}$$

The functions f_{θ_f} and g_{θ_g} in the above equation depend only on the variable x and are independent of the time variable t. In this case, note that the Lipschitz condition $|f_{\theta_f}(x) - f_{\theta_f}(y)| + |g_{\theta_g}(x) - g_{\theta_g}(y)| \leq C|x - y|$ implies $|f_{\theta_f}(x) - f_{\theta_f}(y)| \leq C|x - y|$. Then we derive

$$|f_{\theta_f}(x)| \leq |f_{\theta_f}(x) - f_{\theta_f}(0)| + |f_{\theta_f}(0)| \leq C|x| + |f_{\theta_f}(0)| \leq D(1 + |x|), \tag{31}$$

where $D = \max\{C, |f_{\theta_f}(0)|\}$. Therefore, the linear growth condition automatically follows from the Lipschitz condition (Kuo, 2006). According to the Theorem 5.2.1 in (Oksendal, 2013), we know that the solution $X(t)$ of Eq.(30) exists uniquely. This solution is a measurable function of $X(0)$, W and N in the obvious sense. Using the continuity of $h_{\theta_h}(x)$, we can set

$$\eta_1(t) = X(t), \quad t \in [0, t_1], \tag{32}$$

and

$$\eta_1(t_1+) = X(t_1) + h_{\theta_h}(X(t_1)). \tag{33}$$

The process $\{\eta_1(t)\}_{t\in[0,t_1]}$ is clearly the unique solution of Eq.(12) in the time interval $[0, t_1]$. Next, set $\widetilde{X}(0) = \eta_1(t_1+)$, $\widetilde{W} = \{\widetilde{W}_t\}_{t\geq 0}$ where $\widetilde{W}_t = W_{t+t_1} - W_{t_1}$, and $\widetilde{N} = \{\widetilde{N}_t\}_{t\geq 0}$ where $\widetilde{N}_t = N_{t+t_1} - N_{t_1}$. Since the SDE Eq.(30) is autonomous, we can determine the process $\tilde{\eta}_2(t)$ on $\left[0, \tilde{t}_1\right]$ with respect to $\widetilde{X}(0)$, $\widetilde{W}$ and $\widetilde{N}$ in the same way as $\eta_1(t)$. Clearly $\tilde{t}_1$, defined with respect to $\widetilde{N}$, coincides with $t_2 - t_1$. Define $\{\eta(t)\}_{t\in[0,t_2]}$ by

$$\eta(t) = \begin{cases} \eta_1(t), & t \in [0, t_1]\,, \\ \tilde{\eta}_2\left(t - t_1\right), & t \in (t_1, t_2]\,. \end{cases} \tag{34}$$

It is easy to see that $\{\eta(t)\}_{t\in[0,t_2]}$ is the unique solution of Eq.(12) in the time interval $[0, t_2]$. Continuing this process successively, $\eta(t)$ is determined uniquely in the time interval $[0, t_i]$ for every i and hence $\eta(t)$ is determined globally. This completes the proof. □

B. Training Algorithm

The pseudo-codes of training algorithm of the Neural Jump-Diffusion Univariate Point Process (NJDUPP) and the Neural Jump-Diffusion Multivariate Point Process (NJDMPP) are presented in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1 Training of the NJDUPP.

Input: Model parameter $\theta = \{\theta_f, \theta_g, \theta_h, \eta_0\}$, start time $t_0 = 0$, event sequences $\{\mathcal{S}^l\}_{l=1}^L$, where $\mathcal{S}^l = \{t_i^l\}_{i=1}^{n_l}$
Initialize f_{θ_f}, g_{θ_g}, h_{θ_h}, $\eta_{t_0+} = \eta_0$, $\mathcal{L} = 0$
while *Not Converge* **do**
  for *each sequence* $\mathcal{S}^l$ *in batch* **do**
    for $i = 1, \ldots, n_l + 1$ **do**
      $\{\eta_{\tau_k^i}\}_{k=1}^N = \text{SDESolve}\left(f_{\theta_f}, g_{\theta_g}, \eta_{t_{i-1}^l+}, (\tau_0^i = t_{i-1}^l, \ldots, \tau_k^i, \ldots, \tau_N^i = t_i^l)\right)$ ▷ using Eq.(16)
      $\eta_{t_i^l+} = \eta_{t_i^l} + h_{\theta_h}(\eta_{t_i^l})$ ▷ right-limit at $t_i^l = \tau_N^i$
    end
    $\mathcal{L}^l = -\ell\big(\{\eta_{\tau_k^i}\}_{\substack{k=0 \\ i=1}}^{N, n_l+1}; \theta\big)$ ▷ compute the NLL Eq.(17)
    $\mathcal{L} \mathrel{+}= \mathcal{L}^l$
  end
  back-propagate with gradient $\nabla_\theta \mathcal{L}$
  update model parameters by Adam optimizer
end

Algorithm 2 Training of the NJDMPP.

Input: Model parameter $\theta = \{\boldsymbol{\theta}_f, \boldsymbol{\theta}_g, \boldsymbol{\theta}_h, \boldsymbol{\eta}_0\}$, $t_0 = 0$, multi-type event sequences $\{\mathcal{S}^l\}_{l=1}^L$, where $\mathcal{S}^l = \{(t_i^l, m_i^l)\}_{i=1}^{n_l}$
Initialize $\boldsymbol{f}_{\boldsymbol{\theta_f}}$, $\boldsymbol{g}_{\boldsymbol{\theta_g}}$, $\boldsymbol{h}_{\boldsymbol{\theta_h}}$, $\boldsymbol{\eta}_{t_0+} = \boldsymbol{\eta}_0$, $\mathcal{L} = 0$
while *Not Converge* **do**
  for *each sequence* $\mathcal{S}^l$ *in batch* **do**
    for $i = 1, \ldots, n_l + 1$ **do**
      $\{\boldsymbol{\eta}_{\tau_k^i}\}_{k=1}^N = \text{SDESolve}\left(\boldsymbol{f}_{\boldsymbol{\theta_f}}, \boldsymbol{g}_{\boldsymbol{\theta_g}}, \boldsymbol{\eta}_{t_{i-1}^l+}, (\tau_0^i = t_{i-1}^l, \ldots, \tau_k^i, \ldots, \tau_N^i = t_i^l)\right)$ ▷ using Euler-Maruyama scheme
      $\boldsymbol{\eta}_{t_i^l+} = \boldsymbol{\eta}_{t_i^l} + \boldsymbol{h}_{\boldsymbol{\theta_h}}^{m_i^l}(\boldsymbol{\eta}_{t_i^l})$
    end
    $\mathcal{L}^l = -\ell\big(\{\boldsymbol{\eta}_{\tau_k^i}\}_{\substack{k=0 \\ i=1}}^{N, n_l+1}; \theta\big)$ ▷ compute the NLL Eq.(36)
    $\mathcal{L} \mathrel{+}= \mathcal{L}^l$
  end
  back-propagate with gradient $\nabla_\theta \mathcal{L}$
  update model parameters by Adam optimizer
end

C. Experimental Details

C.1. Experimental Environment

The experiments are conducted on a Linux server with eight GPUs (NVIDIA RTX 2080 Ti * 8). We implement our model and all baselines with the deep learning library `PyTorch` (Paszke et al., 2017).

C.2. Dataset Descriptions

- **Retweet** (Zhou et al., 2013). The dataset consists of sequences of time-stamped user retweet events, categorized into three types based on the users' following sizes: "small", "medium", and "large".

- **Earthquake** (Xue et al., 2024). This dataset contains timestamped earthquake events over the Conterminous U.S from 1996 to 2023. The seven event types are defined based on the magnitude of earthquakes.

Table 5. Statistics of the used datasets.

Dataset	# Types	# Sequences			Sequence Length			# Events		
		Train	Dev	Test	Min	Mean	Max	Train	Dev	Test
Retweet	3	9, 000	1, 535	1, 520	10	40	97	369, 731	62, 823	61, 154
Earthquake	7	3000	400	896	11	16	18	49, 363	6, 612	14, 748
Taxi	10	1, 400	200	400	36	37	38	51, 854	7, 404	14, 820
Taobao	17	1, 300	200	500	32	57	64	75, 205	11, 737	28, 455
StackOverflow	22	1, 401	401	401	41	65	101	90, 497	25, 762	26, 518
MIMIC-II	75	527	58	65	2	4	33	1, 930	252	237

- **Taxi** (Whong, 2014). This dataset contains time-stamped taxi pick-up and drop-off events throughout the five boroughs of New York city. Each combination of borough, whether it's a pick-up or drop-off, defines an event type, resulting in a total of 10 event types.

- **Taobao** (Xue et al., 2022). This dataset includes the time-stamped click behavior of users in Taobao platform from November 25 to December 3, 2017. Each user has a sequence of product click events, where each event contains a timestamp and the product category.

- **StackOverflow** (Leskovec & Krevl, 2014). This dataset contains two years of user-awarded collections from the question-answering website. Each user is awarded a sequence of badges, with a total of 22 different badge types.

- **MIMIC-II** (Johnson et al., 2016). This dataset includes timestamped de-identified clinical visit events of Intensive Care Unit patients for seven years. Each patient has a sequence of hospital visit events, and each event records its timestamp and disease diagnosis.

Table 5 shows statistics about each dataset mentioned above.

C.3. Baseline Descriptions

We provide detailed descriptions of the used baselines as follows:

- **RMTPP** (Du et al., 2016). RMTPP leverages RNNs to learn a hidden representation of event history, and then applies an exponential transformation on this representation for defining the intensity function.

- **NHP** (Mei & Eisner, 2017). NHP proposes a continuous-time LSTM to encode event sequences. The intensity function of NHP can decay over time and does not need to encode inter-event times as numerical inputs to the LSTM.

- **SAHP** (Zhang et al., 2020a). SAHP uses a self-attention mechanism to aggregate historical events, which enhances the expression ability of the intensity function.

- **THP** (Zuo et al., 2020). To capture the long-term dependence of events, THP proposes to model the intensity function using a Transformer architecture.

- **AttNHP** (Yang et al., 2022). AttNHP generalizes the Transformer architecture for modeling event sequences. Its architecture builds rich embeddings of actual and possible events at any given time, based on lower-level representations of these events and their context.

- **NSTPP** (Chen et al., 2020). With the goal of modeling high-fidelity distributions in continuous time and space, NSTPP uses the Neural ODE framework to parameterize the spatio-temporal TPP by combining ideas from Neural Jump SDEs (Jia & Benson, 2019) and continuous-time normalizing flows (Chen et al., 2018).

C.4. Evaluation Metrics

We formulate the metrics used in this paper as follows:

- The mean absolute percentage error (MAPE) of the estimated intensity is given by

$$\text{MAPE} = \frac{1}{T}\int_0^T \left|\frac{\lambda_t^{\text{model}} - \lambda_t^{\text{GT}}}{\lambda_t^{\text{GT}}}\right| \mathrm{d}t \times 100\%, \tag{35}$$

  where T is the observation length, λ_t^{model} is the trained model intensity, and λ_t^{GT} is the ground truth intensity.

- The negative log-likelihood (NLL) of a multivariate point process over a time interval $[0, T]$ is

$$\text{NLL} = -\sum_{i=1}^{n} \log \lambda_{t_i}^{m_i} + \sum_{m=1}^{M} \left(\int_0^T \lambda_s^m \,\mathrm{d}s\right) = -\sum_{i=1}^{n} \eta_{t_i}^{m_i} + \sum_{m=1}^{M} \left(\int_0^T \exp\left(\eta_s^m\right) \mathrm{d}s\right). \tag{36}$$

- The root mean square error (RMSE) is

$$\text{RMSE}(\boldsymbol{t}, \hat{\boldsymbol{t}}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(t_i - \hat{t}_i\right)^2}. \tag{37}$$

- The accuracy of multiclass classification is the fraction of correct classifications, that is

$$\text{Accuracy} = \frac{\text{\# correct classifications}}{\text{\# all classifications}}. \tag{38}$$

- The formula for the weighted F_1 score, accounting for class imbalance, is expressed as:

$$F_{1\text{weighted}} = \sum_{i=1}^{C} w_i \cdot F_{1i}, \tag{39}$$

  where C is the number of classes, w_i represents the sample weight for class i, and F_{1i} is the F_1 score for class i.

C.5. Training Details

We train our NJDTPP model for all experiments by minimizing the negative log-likelihood of training sequences, as described in Appendix B. The drift net, diffusion net, and jump net of NJDTPP are implemented as three multi-layer perceptrons (MLPs) with the same network structure. The activation function chosen for these networks is Tanh. Optimizer is Adam (Kingma & Ba, 2015) with a weight decay of 10^{-5}. The MLP parameters and the initial value η_0 are initialized by the Gaussian distribution.

C.6. Hyper-parameter Setting

We employ the "diagonal" noise in the diffusion term of Eq.(18), i.e., $\boldsymbol{g}_{\boldsymbol{\theta}_g}$ is a diagonal matrix. In this case, the dimension of the Brownian motion $\boldsymbol{W}_t$ is equal to the total number of event types, i.e., $K = M$. Grid search is used to determine other hyper-parameters: the learning rate is selected from $\{0.001, 0.01, 0.1\}$, the hidden layer number is selected from $\{1, 2, 3\}$, and the hidden layer size is selected from $\{16, 32, 64\}$.