# Flow-Based Delayed Hawkes Process

**Chao Yang**[1] **Wendi Ren**[1] **Shuang Li**[*1]

[1]School of Data Science , The Chinese University of Hong Kong (Shenzhen) , China

## Abstract

Multivariate Hawkes processes are classic temporal point process models for event data. These models are simple and parametric in nature, offering interpretability by capturing the triggering effects between event types. However, these parametric models often struggle with low model capacity, limiting their expressive power to capture heterogeneous data patterns influenced by latent variables. In this paper, we propose a simple yet powerful extension: the Flow-based Delayed Hawkes Process, which integrates Normalizing Flows as a generative model to parameterize the Hawkes process. By generating all model parameters through the flow-based network, our approach significantly improves flexibility and expressiveness while preserving interpretability. We provide theoretical guarantees by proving the identifiability of the model parameters and the consistency of the maximum likelihood estimator under mild assumptions. Extensive experiments on both synthetic and real-world datasets show that our model outperforms existing baselines in capturing intricate and heterogeneous event dynamics.

## 1 INTRODUCTION

Complex systems often produce voluminous event data with *stochastic* and *irregularly-spaced* occurrence times. Temporal point process (TPPs) provide an elegant tool for modeling the dynamics of these event sequences in *continuous time*, which directly treat the inter-event time as random variables [Daley et al., 2003]. Among various TPPs, Hawkes process is a classic and transparent model, with intensity functions are designed to capture the *triggering effects* from previous events. The intensity functions capture self-exciting and mutual-exciting triggering effects across event types, which can be interpreted—under certain conditions—as forming a Granger causality graph that reflects the underlying temporal dependencies [Eichler et al., 2017, Gao et al., 2021].

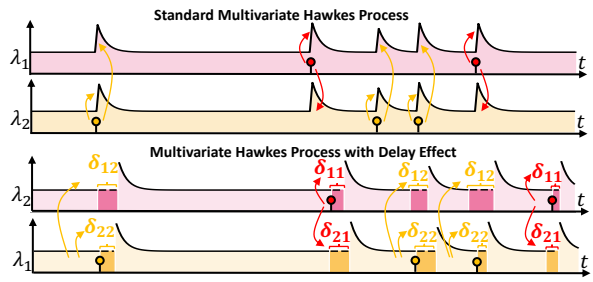


Figure 1: Illustration of a multivariate Hawkes process without (**top**) and with (**bottom**) delay effects. In the bottom case, the time lag $\delta_{uu'}$ captures the delayed triggering from dimension $u'$ to $u$—that is, an event in $u'$ affects the intensity of $u$ only after a delay of $\delta_{uu'}$, not immediately.

Hawkes processes are widely used with exponential kernels due to their simplicity and interpretability. To enhance modeling flexibility, various extensions have introduced alternative triggering kernels—such as nonparametric [Eichler et al., 2017] and Gaussian mixture kernels [Xu et al., 2016]. While these parametric or nonparametric variants improve expressiveness, they often struggle to capture the *heterogeneous triggering patterns* found in real-world data. To address this, *neural-based Hawkes processes* have been proposed [Du et al., 2016, Mei and Eisner, 2017, Zuo et al., 2020], offering increased expressiveness through data-driven modeling. Yet, their black-box nature sacrifices interpretability, raising important concerns about the trade-off between flexibility and transparency.

*Can we enhance model capacity while preserving interpretability through a simple, plug-and-play approach?*

To answer this, we propose a novel framework that leverages Normalizing Flows (NFs) [Dinh et al., 2016, Papamakarios et al., 2021] to *model distributions over Hawkes process*

[*]Corresponding Author: lishuang@cuhk.edu.cn.

*parameters*. Our approach achieves both flexibility and interpretability by combining a simple, parametric *main model* with expressive, data-driven *parameter generation*. Specifically, the main model is a delayed Hawkes process featuring time lags, where all parameters—including the base intensity and those of the exponential triggering kernel—are generated by a NF. To further enhance model capacity and mitigate mode collapse, we introduce an ensemble of multiple NFs, allowing the model to capture a broader range of behaviors.

Specifically, the **main model** adopts an exponential triggering kernel with delay:

$$g(t) = \alpha \exp(-\beta(t-\delta))\mathbb{1}\{t-\delta \geq 0\},$$

where $\alpha > 0$ controls the strength of the triggering effect, $\beta > 0$ governs how rapidly the effect decays, and $\delta \geq 0$ introduces a delay before the effect begins. Figure 1 illustrates the difference between a standard multivariate Hawkes process and our delayed variant. While traditional models assume immediate triggering, the delay parameter $\delta$ captures the time lag between an event and its influence on future occurrences. This modeling capability is essential in many real-world settings—for instance, during the COVID-19 pandemic, the concept of *incubation periods* helps explain why symptoms and infectiousness appear only several days after exposure [Quesada et al., 2021, Koyama et al., 2021]. Similarly, in chronic diseases such as cancer, environmental exposures or genetic mutations may not manifest clinically until years later. By explicitly modeling such delays, we enhance both the realism and expressiveness of the temporal process.

Our framework leverages flow-based generative models to generate Hawkes process parameters, allowing the model to flexibly capture *heterogeneous dynamics* driven by unobserved or latent factors. In domains like healthcare, hidden variables—such as patient age, comorbidities, drug resistance, or psychological status—can cause wide variation in how interventions affect outcomes. A fixed parameter model cannot account for this variability. Instead, our method learns a *rich, joint distribution* over all the key parameters of the Hawkes process, capturing both their marginal variability and their interdependencies. Using deep NFs, we are able to model complex, multimodal parameter distributions, which enables the system to represent diverse behaviors across individuals or subpopulations.

In summary, our contributions are threefold:
*i)* We propose a simple yet flexible framework that generates Hawkes process parameters using a deep generative model, allowing the capture of heterogeneous triggering patterns, including delay effects that are often overlooked.
*ii)* We provide theoretical analysis showing identifiability of the parameter distribution under our model and the consistency of the estimator.
*iii)* Empirical results on both synthetic and real-world datasets demonstrate the competitive performance of the model and the ability to handle complex event dynamics.

## 2 RELATED WORK

**Temporal Point Process (TPPs)** provide a principled framework for modeling the timing of discrete events in continuous time. Among them, the Hawkes process [Hawkes, 1971, Xu et al., 2016] is one of the most widely used, particularly for inferring inter-type Granger causality [Granger, 1969, Dahlhaus and Eichler, 2003]. The classic Hawkes model assumes that past events independently and additively increase the intensity of future events through a set of pairwise kernel functions. While the exponential kernel is most common, several studies have explored alternative parametric forms to increase modeling flexibility, such as the Gamma kernel [Lesage et al., 2022], Weibull kernel [Zhang et al., 2020a], and power-law kernel [Zhang, 2016].

More recently, neural-based TPP models have been proposed to improve expressiveness by parameterizing the intensity function directly with deep networks. These approaches include RNN- and LSTM-based models [Du et al., 2016, Mei and Eisner, 2017, Xiao et al., 2017, Mei et al., 2020] and Transformer-based architectures [Zuo et al., 2020, Zhang et al., 2020b, Zhu et al., 2021, Yang et al., 2021]. However, *these methods primarily focus on directly approximating the intensity function, which limits their ability to recover meaningful insights such as Granger causality or delay effects*.

In contrast, our work focuses on modeling the full distribution over the parameters of a Hawkes process. This *distributional view* enables the model to capture heterogeneous triggering patterns, while maintaining interpretability through the use of a simple parametric backbone.

**Parameter Estimation for TPPs** has been explored from both frequentist and Bayesian perspectives. Classical approaches such as maximum likelihood estimation (MLE) [Lewis et al., 2012] and the EM algorithm [Lewis and Mohler, 2011, Wheatley et al., 2014] *typically yield point estimates of model parameters*. Kernel-based and other nonparametric methods [Zhou et al., 2013, Joseph et al., 2020, Kirchner, 2017, Eichler et al., 2017] estimate intensity functions or kernel shapes without assuming specific parametric forms, *but generally provide functional or point estimates rather than full parameter distributions*.

Bayesian methods [Zhang et al., 2018, Santos et al., 2023] aim to infer posterior distributions over parameters, offering uncertainty quantification and limited modeling of heterogeneity. *However, these approaches often rely on simplifying assumptions or approximations that restrict their ability to capture complex, multimodal structures*.

More recently, generative models such as hypernetworks [Dubey et al., 2022, 2023], variational autoencoders (VAEs) [Mehrasa et al., 2019b], and NFs [Mehrasa et al., 2019a, Shchur et al., 2019] have been applied to TPPs—*primarily for modeling latent dynamics or inter-event time distributions—rather than directly learning distributions over the underlying process parameters*.

In contrast, our approach explicitly learns flexible joint distributions over key Hawkes process parameters—base intensity, triggering strength, decay rate, and delay—using NFs trained via a differentiable maximum likelihood objective. This enables direct optimization over expressive parameter families, capturing rich, multimodal patterns and better reflecting heterogeneity in real-world temporal dynamics.

# 3 MODEL: FLOW-BASED DELAYED HAWKES PROCESSES

Consider a $U$-dimensional temporal point process with event sequences $\{N_u(t)\}_{u=1}^{U}$, where $N_u(t)$ denotes the number of events in dimension $u$ up to time $t$. The corresponding event histories are defined as

$$\mathcal{H}_t = \{t_n^u : 1 \le n \le N_u(t), u = 1, \dots, U\}.$$

In our interpretable **main model**, the conditional intensity for dimension $u$ is defined by a Hawkes process with delayed triggering and exponentially decaying kernels:

$$f_u\left(t \mid \mathcal{H}_t; \boldsymbol{\theta}\right) = \mu_u + \sum_{u'=1}^{U} \sum_{n=1}^{N_u(t)} \alpha_{uu'} e^{-\beta\left(t - t_n^{u'} - \delta_{uu'}\right)} \mathbb{1}\{t - t_n^{u'} \ge \delta_{uu'}\} \quad (1)$$

where $\mu_u \in \mathbb{R}^+$ is the base intensity at which events occur spontaneously, $\alpha_{uu'} \ge 0$ (for all $u, u' \in [U]$) quantifies the strength of the triggering effect from events in dimension $u'$ to events in dimension $u$, and $\beta > 0$ controls the decay rate of this effect (with $\beta$ being shared across all event types). We further introduce $\delta_{uu'} \ge 0$ (for all $u, u' \in [U]$) to indicate the delay before the triggering effect becomes effective, such that the triggering kernel is active only when $t - t_n^{u'} \ge \delta_{uu'}$. Finally, we denote the complete set of parameters by

$$\boldsymbol{\theta} := \{\boldsymbol{\mu}, \boldsymbol{\alpha}, \beta, \boldsymbol{\delta}\}$$

where $\boldsymbol{\mu} := [\mu_u]$, $\boldsymbol{\alpha} := [\alpha_{uu'}]$, and $\boldsymbol{\delta} := [\delta_{uu'}]$.

To capture heterogeneity in the dynamics of the event sequences, we extend the main model by assuming that the parameters are not fixed but are drawn from a learnable distribution $p(\boldsymbol{\theta})$, i.e., $\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$, modeled via a NF. Accordingly, the expected (or marginal) intensity function becomes

$$\lambda_u\left(t \mid \mathcal{H}_t; p(\boldsymbol{\theta})\right) := \mathbb{E}_{\boldsymbol{\theta} \sim p(\boldsymbol{\theta})}[f_u(t; \boldsymbol{\theta})] \quad (2)$$

where we denote $f_u(t; \boldsymbol{\theta}) := f_u\left(t \mid \mathcal{H}_t; \boldsymbol{\theta}\right)$ for notation simplicity. This formulation marginalizes over a learned parameter distribution rather than relying on a fixed setting. It naturally captures heterogeneity across sequences, as different parameter samples induce different dynamics. Effectively, it acts like a mixture of Hawkes processes—each sample defines a component—allowing the model to flexibly represent diverse triggering patterns while retaining interpretability.

**Modeling Joint Dependencies with NFs** We explicitly model the **joint distribution** of the parameters $\boldsymbol{\theta}$ using a NF that captures their inherent dependencies. The concatenated main model parameters will be $\boldsymbol{\theta} \in \mathbb{R}^d$, where $d = 2U^2 + U + 1$.

We assume a latent variable $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ and learn an invertible transformation:

$$\boldsymbol{\theta} = F_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}), \quad \text{where } F_{\boldsymbol{\phi}} : \mathbb{R}^d \to \mathbb{R}^d.$$

Here, $F_{\boldsymbol{\phi}}$ is implemented using a flexible flow-based generative model (e.g., RealNVP [Dinh et al., 2016], Glow [Kingma and Dhariwal, 2018], or Neural Spline Flows [Durkan et al., 2019]). Specifically, we define the transformation as a composition of $K$ invertible layers:

$$F_{\boldsymbol{\phi}} = h_K \circ h_{K-1} \circ \cdots \circ h_1$$

where each $h_k$ is an invertible mapping with a tractable Jacobian determinant. For example, in a RealNVP-style flow, each layer $h_k$ may be defined by an affine coupling layer. In such a layer, the input is split into two parts $\boldsymbol{u}$ and $\boldsymbol{v}$; then one updates the output as

$$\boldsymbol{u}' = \boldsymbol{u}, \quad \boldsymbol{v}' = \boldsymbol{v} \odot \exp\left(s_k(\boldsymbol{u})\right) + b_k(\boldsymbol{u})$$

where $s_k(\cdot)$ and $b_k(\cdot)$ are neural networks parameterized by $\boldsymbol{\phi}$, and $\odot$ denotes elementwise multiplication. The invertibility of each $h_k$ is ensured, and the Jacobian determinant is easily computed since it is triangular. Using the change-of-variables formula, the target density $p_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ induced by the flow is given by:

$$p_{\boldsymbol{\phi}}(\boldsymbol{\theta}) = p_{\boldsymbol{\epsilon}}\left(F_{\boldsymbol{\phi}}^{-1}(\boldsymbol{\theta})\right) \cdot \left|\det\left(\frac{\partial F_{\boldsymbol{\phi}}^{-1}}{\partial \boldsymbol{\theta}}\right)\right|$$

where $p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})$ is the density of the base multivariate Gaussian. In practice, we compute the inverse $F_{\boldsymbol{\phi}}^{-1}(\boldsymbol{\theta})$ layer by layer, and accumulate the log-determinants of the Jacobian matrices from each transformation. This construction allows us to evaluate the target density $p_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ efficiently.

Using a single NF can sometimes lead to *mode collapse*. This is particularly problematic when modeling the joint distribution of the parameters in our Hawkes process, as the parameters (such as $\boldsymbol{\mu}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\delta}$) often exhibit complex, multimodal dependencies reflecting heterogeneous triggering behaviors. We propose to use a mixture of NFs (depicted in Figure 2) to address this challenge by combining several component flows, each of which can specialize in capturing different modes of the distribution. Concretely, the target density of the Hawkes parameters is represented as

$$p(\boldsymbol{\theta}) = \sum_{m=1}^{M} \pi_m p_m(\boldsymbol{\theta})$$

where each component $p_m(\boldsymbol{\theta})$ is modeled by its own NF, with parameters denoted as $\boldsymbol{\phi}_m$, and $\pi_m$ are the mixture weights (summing to 1). When computing the marginal intensity, the mixture formulation results in a weighted sum of expectations from each component (due to the linearity of expectation):

$$\lambda_u\left(t \mid \mathcal{H}_t; p(\boldsymbol{\theta})\right) = \sum_{m=1}^{M} \pi_m \mathbb{E}_{\boldsymbol{\theta} \sim p_m(\boldsymbol{\theta})}\left[f_u(t; \boldsymbol{\theta})\right] \quad (3)$$

ensuring that all modes contribute proportionally to the final intensity function according to their mixture weights.

Using the mixture model have been explored in generative models such as GANs, where multiple adversarial networks

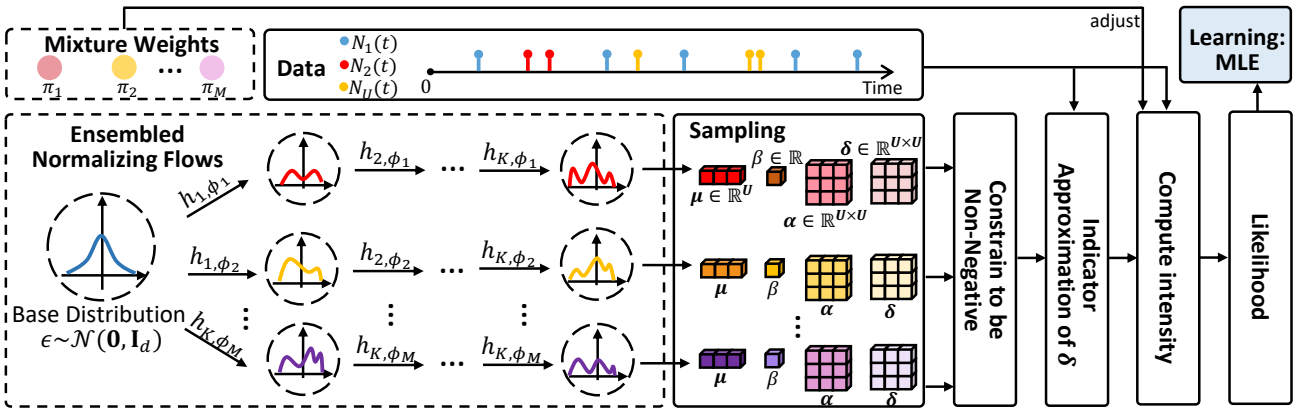


Figure 2: Model framework: the normalizing flow ensembles with mixture weights are presented in dashed boxes.

have been used to mitigate mode collapse and improve diversity [Hoang et al., 2018, Nguyen et al., 2017, Durugkar et al., 2016, Mordido et al., 2020]. Similarly, Berry and Meger [2023] extended this idea to normalizing flows. Building on this, we incorporate a mixture of NFs to improve mode coverage in our flow-based delayed Hawkes process.

## 4 MODEL LEARNING

The overall framework is shown in Figure 2, where the model first computes the marginal intensity function by averaging over sampled parameters from the flow-based generator, which is then used to evaluate the log-likelihood of the observed event sequences. Now the model parameters become $\boldsymbol{\phi} = [\boldsymbol{\phi}_m]$ and $\boldsymbol{\pi} = [\pi_m] \in \Delta^M$ (i.e., probability simplex) of the mixture NFs. We will learn $\boldsymbol{\phi}$ and $\boldsymbol{\pi}$ via maximizing the log-likelihood of the observed event sequences through:

$$\max_{\boldsymbol{\phi},\boldsymbol{\pi}\in\Delta^M} \mathcal{L}\left(p_{\boldsymbol{\phi},\boldsymbol{\pi}}(\boldsymbol{\theta})\right) \tag{4}$$

where the log-likleihood $\mathcal{L}\left(p_{\boldsymbol{\phi},\boldsymbol{\pi}}\right)$ is computed as

$$\sum_u \left[\sum_{n=1}^{N_u(T)} \log \lambda_u^*\left(t_n^u\right) - \int_0^T \lambda_u^*\left(t\right) dt\right] \tag{5}$$

and $\lambda_u^*\left(t\right) := \lambda_u\left(t_n^u \mid \mathcal{H}_{t_n^u}; p_{\boldsymbol{\phi},\boldsymbol{\pi}}(\boldsymbol{\theta})\right)$ is the marginal intensity as defined in Eq. (3) and $T$ is the time horizon.

To approximate the marginal intensity, we first draw samples from each NF component. For each component $m$, we generate samples

$$\boldsymbol{\theta}^{(s)} \sim p_m(\boldsymbol{\theta})$$

using the reparameterization $\boldsymbol{\theta}^{(s)} = F_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(s)})$ with $\boldsymbol{\epsilon}^{(s)} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}_d\right)$. Since the model parameters $\boldsymbol{\mu}, \boldsymbol{\alpha}, \beta$ and $\boldsymbol{\delta}$ are constrained to be nonnegative, we modify the output of the NF by applying a softplus activation to the last layer so that the generated $\boldsymbol{\theta}$ always satisfies this nonnegativity condition. The expectation is then approximated via Monte Carlo, yielding the marginal intensity:

$$\lambda_u\left(t \mid \mathcal{H}_t; p(\boldsymbol{\theta})\right) \approx \sum_{m=1}^{M} \pi_m \left[\frac{1}{S}\sum_{s=1}^{S} f_u\left(t; \boldsymbol{\theta}^{(s)}\right)\right]. \tag{6}$$

**Gradient Computation with Respect to $\boldsymbol{\phi}_m$** To compute the gradient of the log-likelihood with respect to the parameters $\boldsymbol{\phi}_m$ of the $m$-th NF component, we apply the reparameterization trick. For each sample $\boldsymbol{\theta}^{(s)} = F_{\boldsymbol{\phi}_m}\left(\boldsymbol{\epsilon}^{(s)}\right)$, where $\boldsymbol{\epsilon}^{(s)} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}_d\right)$, we approximate the gradient as:

$$\nabla_{\boldsymbol{\phi}_m} \mathbb{E}_{\boldsymbol{\theta}\sim p_m(\boldsymbol{\theta})}\left[f_u(t;\boldsymbol{\theta})\right] \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_{\boldsymbol{\phi}_m} f_u\left(t; F_{\boldsymbol{\phi}_m}\left(\boldsymbol{\epsilon}^{(s)}\right)\right).$$

This enables efficient end-to-end training by backpropagating through NF generator using automatic differentiation.

**Handling the Non-differentiability of $\boldsymbol{\delta}$** The intensity function is inherently non-differentiable with respect to the delay parameter $\boldsymbol{\delta}$ due to the indicator function in the triggering kernel (as shown in Eq. (1)). To obtain gradient estimates for $\delta_{uu'}$, we approximate the indicator function with a smooth sigmoid function $\sigma(z) = \frac{1}{1+\exp(-\tau z)}$ where $\tau > 0$ is a temperature parameter that controls the steepness of the sigmoid. In other words, we approximate

$$\mathbb{1}\left\{t - t_n^{u'} \geq \delta_{uu'}\right\} \approx \sigma\left(t - t_n^{u'} - \delta_{uu'}\right). \tag{7}$$

This approximation makes intensity function differentiable with respect to delay parameters $\delta_{uu'}$, enabling gradient-based optimization. Details can be found in Appendix B.1.

**Gradient Computation with Respect to $\boldsymbol{\pi}$** For the mixture weights $\boldsymbol{\pi} \in \Delta^M$, we eliminate the probability simplex constraint by reparameterizing them via a softmax function. Define $\pi_m = \frac{\exp(w_m)}{\sum_{j=1}^{M}\exp(w_j)}$, where $w_m \in \mathbb{R}$ are unconstrained parameters. This reparameterization allows us to compute gradients with respect to $w_m$ (and hence $\boldsymbol{\pi}$) via standard backpropagation through the softmax, simplifying optimization over the simplex.

*Discussion: Frequentist v.s. Bayesian Approach?* While our method models distributions over Hawkes process parameters, it follows a **frequentist approach**, not a Bayesian one. Instead of specifying priors and performing posterior inference, we **learn the parameter distributions directly** by optimizing the log-likelihood of observed event sequences. Specifically, we first compute the marginal intensity function

by averaging the Hawkes intensity over sampled parameters from a flow-based generator, and then use this marginal intensity to evaluate the log-likelihood. This formulation enables us to flexibly capture heterogeneity in temporal dynamics without relying on approximate Bayesian inference or prior assumptions.

# 5 THEORETICAL ANALYSIS

We begin by establishing the identifiability of the fixed parameter $\boldsymbol{\theta}$ in the delayed Hawkes process (Theorem 1) and extend this to show that the distribution over parameters $p(\boldsymbol{\theta})$ is also identifiable (Theorem 2). Unlike traditional approaches that assume fixed parameters, our method learns a distribution over parameters to capture population-level heterogeneity. Therefore, establishing the identifiability of $p(\boldsymbol{\theta})$ is critical to ensuring meaningful and interpretable inferences. Building on this, we prove the consistency of the maximum likelihood estimator (MLE) for $p(\boldsymbol{\theta})$ under standard regularity conditions (Theorem 3). Together, these results form the theoretical foundation of our delayed Hawkes process framework—demonstrating why it is identifiable and statistically reliable in practice.

## 5.1 IDENTIFIABILITY

We first establish the identifiability of the fixed parameters $\boldsymbol{\theta}$ in the delayed Hawkes process. That is, the model parameters can be uniquely recovered from the conditional intensity functions under mild and practically reasonable assumptions.

**Theorem 1 (Identifiability of fixed $\boldsymbol{\theta}$)** *Let $\mathcal{H}_t$ be a realization of the delayed multivariate Hawkes process as defined in Eq.* (1)*. Suppose the conditional intensity functions satisfy*

$$f_u(t \mid \mathcal{H}_t; \boldsymbol{\theta}) = f_u(t \mid \mathcal{H}_t; \tilde{\boldsymbol{\theta}}), \quad \forall u \in [U] \tag{8}$$

*almost everywhere. Then, under the conditions listed below, it follows that $\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}$.*

**Proof** The proof proceeds in four steps:

(1) **Baseline rate** $\mu_u$: Integrating both sides of Eq. (8) over $[0, t_{(1)}]$, where $t_{(1)}$ is the first event time in $\mathcal{H}_t$, and using $t_{(1)} > 0$ almost surely, we obtain $\mu_u = \tilde{\mu}_u$.

(2) **Delays** $\delta_{uu'}$: Due to the exponential triggering kernel, each past event contributes a peak to the intensity at $t = t_n^{u'} + \delta_{uu'}$. If $\delta_{uu'} \neq \tilde{\delta}_{uu'}$, the peak locations differ, violating Eq. (8).

(3) **Decay rate** $\beta$: Differentiating both sides of Eq. (8) yields $\beta(f_u(t) - \mu_u) = \tilde{\beta}(f_u(t) - \mu_u)$. Since $f_u(t) - \mu_u > 0$ with nonzero probability, we conclude $\beta = \tilde{\beta}$.

(4) **Triggering strengths** $\alpha_{uu'}$: With known $\mu_u$, $\delta_{uu'}$, and $\beta$, the equality of $f_u(t)$ implies $\alpha_{uu'} = \tilde{\alpha}_{uu'}$. ■

**Mild Conditions for Identifiability.** The theorem holds under the following assumptions, which are easily satisfied in practice:

- **Model assumptions:**
  - (i) $\beta$ is shared across all $(u, u')$.
  - (ii) Each $u$ has at least one $u'$ such that $\alpha_{uu'} > 0$.
  - (iii) Delays $\delta_{uu'}$ are fixed and non-negative.
  - (iv) $\mu_u > 0$ for all $u$.
- **Data assumptions:**
  - (i) Event times are continuous and distinct.
  - (ii) For every $(u, u')$ with $\alpha_{uu'} > 0$, at least one event in $u'$ triggers an event in $u$ after delay $\delta_{uu'}$.
  - (iii) $t_{(1)} > 0$ almost surely.
  - (iv) Observation window $[0, T]$ is long enough to observe delayed interactions.

These conditions ensure identifiability while being mild and verifiable in real-world applications. Violations (e.g., simultaneous events, zero delays, or degenerate parameters) may lead to non-identifiability.

In our method, rather than estimating a fixed parameter $\boldsymbol{\theta}$, we aim to learn a distribution $p(\boldsymbol{\theta})$ to capture heterogeneous triggering patterns across event sequences. Therefore, establishing the identifiability of $p(\boldsymbol{\theta})$ is critical to ensure that our model learns a meaningful and unique distribution consistent with observed data.

**Theorem 2 (Identifiability of $p(\boldsymbol{\theta})$)** *Let $\Theta \subset \mathbb{R}^d$ be the parameter space, and let $f_u(t \mid \mathcal{H}_t; \boldsymbol{\theta})$ denote the conditional intensity function. Suppose the following conditions hold:*

*(i) The mapping $\boldsymbol{\theta} \mapsto f_u(t \mid \mathcal{H}_t; \boldsymbol{\theta})$ is injective for almost every $t \in \mathbb{R}^+$.*

*(ii) The function class $\mathcal{F} = \{f_u(t \mid \mathcal{H}_t; \boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta\}$ is complete, meaning that if a measurable function $g : \Theta \to \mathbb{R}$ satisfies*

$$\int_\Theta f_u(t \mid \mathcal{H}_t; \boldsymbol{\theta}) g(\boldsymbol{\theta}) d\boldsymbol{\theta} = 0 \quad \text{for all } t,$$

*then $g(\boldsymbol{\theta}) = 0$ almost everywhere on $\Theta$.*

*Then, if two distributions $p(\boldsymbol{\theta})$ and $q(\boldsymbol{\theta})$ induce the same marginal intensities (as defined in Eq. (2)):*

$$\lambda_u\left(t \mid \mathcal{H}_t; p(\boldsymbol{\theta})\right) = \lambda_u\left(t \mid \mathcal{H}_t; q(\boldsymbol{\theta})\right), \quad \forall u \in [U],$$

*for almost every t, it follows that $p(\boldsymbol{\theta}) = q(\boldsymbol{\theta})$ almost everywhere on $\Theta$.*

We have already established the injectiveness of the mapping $\boldsymbol{\theta} \mapsto f_u(t \mid \mathcal{H}_t; \boldsymbol{\theta})$ in Theorem 1. For the smoothed intensity function used in our model (Eq. (7)), we also prove the completeness of the function class $\mathcal{F}$ (see Appendix A.1 and A.2). Together, these results satisfy the mild and practically realistic conditions required by Theorem 2, thereby ensuring the identifiability of $p(\boldsymbol{\theta})$ in our delayed Hawkes framework.

## 5.2 CONSISTENCY

In this paper, we learn the parameter distribution $p(\boldsymbol{\theta})$ by maximizing the log-likelihood $\mathcal{L}(p_{\boldsymbol{\phi},\boldsymbol{\pi}})$ defined in Eq. (5).

Therefore, establishing consistency of the MLE $\hat{p}(\boldsymbol{\theta})$ is critical to guarantee that as the observation window $T$ grows, our learned distribution converges to the true underlying distribution $p^*(\boldsymbol{\theta})$.

**Theorem 3 (Consistency of MLE $\hat{p}(\boldsymbol{\theta})$)** *Assume the true parameter distribution is $p^*(\boldsymbol{\theta})$ and the Hawkes process model is correctly specified. Suppose the following conditions hold:*

- *The parameter space for $p(\boldsymbol{\theta})$ is compact (or satisfies appropriate regularity conditions).*
- *The mapping $\boldsymbol{\theta} \mapsto f_u(t \mid \mathcal{H}_t; \boldsymbol{\theta})$ is injective for almost every $t$ (by Theorem 1).*
- *The function class $\mathcal{F} = \{f_u(t \mid \mathcal{H}_t; \boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta\}$ is complete, ensuring identifiability of $p(\boldsymbol{\theta})$ (by Theorem 2).*
- *The empirical log-likelihood converges uniformly to its expectation as $T \to \infty$ (via standard point process law of large numbers arguments).*

*Formally, the MLE $\hat{p}(\boldsymbol{\theta})$ satisfies*

$$\hat{p}(\boldsymbol{\theta}) \xrightarrow{p} p^*(\boldsymbol{\theta}) \quad as \quad T \to \infty.$$

*That is, the MLE is consistent.*

Theorem 3 ensures that with sufficient data ($T \to \infty$), the MLE $\hat{p}(\boldsymbol{\theta})$ converges in probability to the true distribution $p^*(\boldsymbol{\theta})$. A proof sketch is provided in Appendix A.3.

# 6 EXPERIMENT

## 6.1 EXPERIMENTAL SETUP

**Baselines** We select several state-of-the-art baselines grouped by their evaluation focus:

*(i) Parameter Distribution Learning Tasks*: This task aims to accurately learn the underlying distribution of model parameters $p(\boldsymbol{\theta})$. Hypernet [Ha et al., 2016, Chauhan et al., 2023] approaches this by training a hypernetwork to produce samples from the parameter distribution. The $\beta$-VAE [Higgins et al., 2017] frames this as a Bayesian inference problem, inferring a posterior over parameters $\boldsymbol{\theta}$ given a prior and data. Both aim to capture uncertainty and variability in parameters beyond point estimates.

*(ii) Comparison of Different Flow Models*: This group evaluates various NF architectures for flexible and expressive parameter distribution modeling, including Planar flows [Rezende and Mohamed, 2015], RealNVP [Dinh et al., 2016], Glow [Kingma and Dhariwal, 2018], RQ-NSF (Rational-Quadratic Neural Spline Flow) [Durkan et al., 2019], and ResFlow (Residual Flow) [Chen et al., 2019].

*(iii) Prediction Tasks*: Here, we compare established TPP models for event prediction. Non-parametric baselines include GM-NLF [Eichler et al., 2017], MMEL [Zhou et al., 2013], and Gibbs-Hawkes [Zhang et al., 2018]. Other flexible TPP models include RMTPP [Du et al., 2016], THP [Zuo et al., 2020], PromptTPP [Xue et al., 2023], HYPRO [Xue et al., 2022], MLE-SGL [Xu et al., 2016], and GC-CGD [Wei et al., 2022]. AttNHP [Yang et al., 2021] serves as the base model for PromptTPP and HYPRO.

**Evaluation Metrics** In multivariate TPPs, parameter learning can be decomposed over event types. We *fix a target type $u$ and evaluate how well the model captures how other types $u'$ influence it*. Specifically, we consider:

*(i) Parameter Distribution Accuracy*: We evaluate the quality of learned parameter distributions (e.g., $[\alpha_{uu'}]_{u' \in U}$, $[\delta_{uu'}]_{u' \in U}$) using the average of marginal KL divergence:

$$\text{aKL} = \frac{1}{U} \sum_{u'=1}^{U} \frac{1}{N} \sum_{n=1}^{N} p(x_{(uu'),n}) \log \left( \frac{p(x_{(uu'),n})}{\hat{p}(x_{(uu'),n})} \right) \tag{9}$$

Here, $p$ is the true density, $\hat{p}$ is the estimated one, and $x_{(uu'),n}$ denotes sampled parameters. Since joint distribution estimation suffers from the curse of dimensionality, we report marginal KL as a tractable proxy. We report the detailed computation process in Appendix B.2.

*(ii) Prediction Accuracy*: We use Root Mean Squared Error (RMSE) to evaluate prediction of the target event $u$'s event times, following prior work [Du et al., 2016, Zuo et al., 2020]:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{t}_i - t_i)^2} \tag{10}$$

## 6.2 SYNTHETIC DATA EXPERIMENTS

**Preprocessing** We consider both *uni-modal* (Gaussian) and *multi-modal* (Gaussian mixture) marginal distributions for each parameter. To evaluate scalability, we vary the dimension of synthetic Hawkes process datasets in $\{2, 3, 5, 7, 9\}$ and the number of training samples in $\{2500, 5000, 7500, 10000, 12500, 15000\}$. Leveraging the decomposability of the Hawkes likelihood, we focus on a single target dimension $u$ for each case. We analyze the impact parameters $[\alpha_{uu'}]_{u' \in U}$ and delays $[\delta_{uu'}]_{u' \in U}$, assuming known base intensity $\mu_u$ and decay rate $\beta$ for synthetic data, while learning all parameter distributions for real-world datasets. We further test the model's robustness under varying decay rate $\beta$ distributions across event types (Appendix D.4).

**Parameter Distribution Learning Performance** Figure 3 compares the marginal parameter distributions learned by our model, Hypernet, and $\beta$-VAE. Hypernet fails to capture multi-modal patterns and lacks an explicit density form, limiting its ability to perform accurate distribution estimation. $\beta$-VAE performs competitively on uni-modal distributions but struggles with multi-modal cases and requires prior knowledge of the underlying distribution, limiting its generalizability. In contrast, our model accurately recovers both uni-modal and multi-modal distributions without any prior assumptions. To quantitatively evaluate performance, we adopt a consistent KL divergence computation (Appendix B.2). The numerical results in Table 4 (Appendix D.1) further confirm that our method consistently

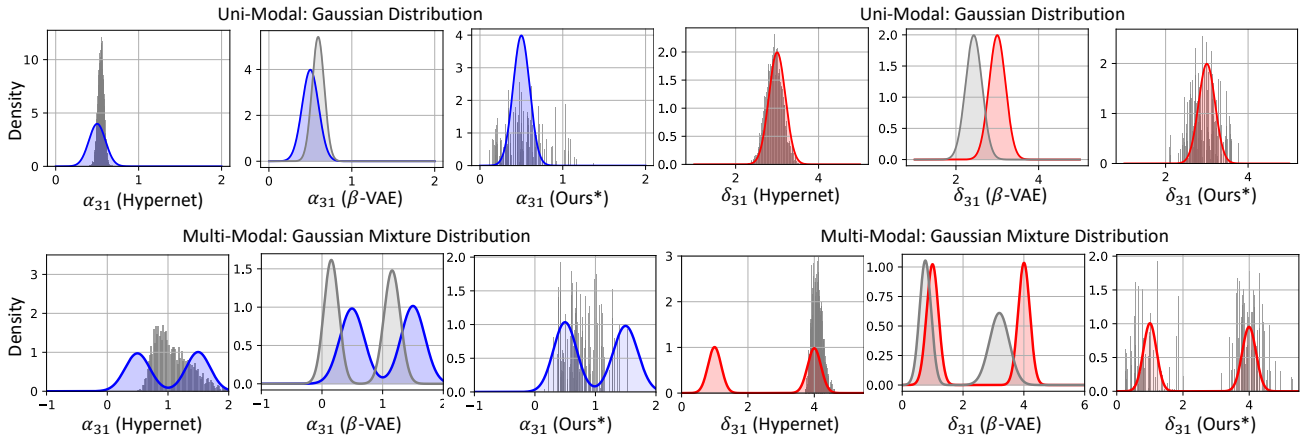


Figure 3: Visualization examples comparing the performance of various models on parameter distribution learning tasks with 3-dimensional datasets and 7500 samples. We report the learned marginal distribution for $\alpha_{31}$ and $\delta_{31}$ in these figures. Complete results can be found in Appendix D.1.

outperforms both Hypernet and $\beta$-VAE across various sample sizes.

Our flow-based model effectively captures the joint distribution and dependencies within the parameter set $\boldsymbol{\theta}$. Depicted in Figure 4, the samples of $\boldsymbol{\alpha}$ and $\boldsymbol{\delta}$ from our well-trained model basically match ground truth joint densities and the underlying density patterns also be uncovered.

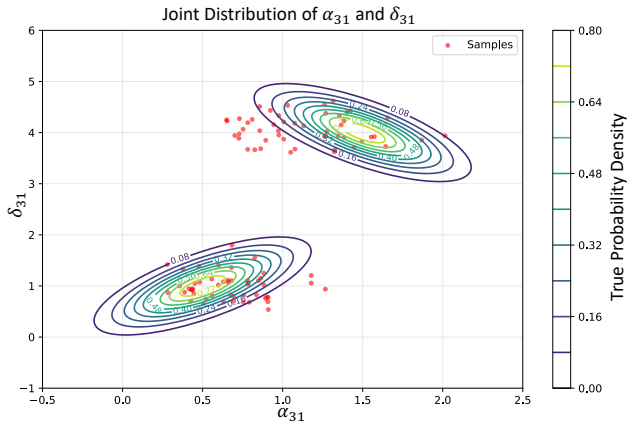


Figure 4: True joint distribution (**contours**) of $\alpha_{31}$ and $\delta_{31}$ and samples (**red circles**) from our well-trained model using multi-modal dataset with 3-dimensional and 7500 samples.

**Scalability and Ablation Study** To evaluate the scalability of our proposed model, we vary the dimensionality and sample sizes. Shown in Appendix D.5, Figure 8, as the training sample size increases, the training time increases while the converged negative log-likelihood decreases, and distribution learning accuracy increases accordingly. Our model demonstrates high efficiency, converging within 1.2 hours even in the most complex scenarios, utilizing 15000 samples of training data with 9 dimensions. As the dimensionality of Hawkes processes increases, the distribution learning accuracy of our model may slightly decrease but remains satisfactory. Encouragingly, as the training sample size grows, the learning performance becomes stable.

To assess the importance of different components of our model, we ablate several modules as in Table 1. Violating all the modules would cause a significant degrade. Furthermore, removing ensemble modules would lead to a decrease in the model's performance, especially for multi-modal distributions. Under current modules combination, our model almost achieves the lowest training converged negative log-likelihood, highest learning accuracy, while maintaining relatively high time efficiency.

Moreover, we investigate the performance when using different normalizing flow models in Table 5, Appendix D.2. Taking into account factors such as data volume and dimensionality, our model strikes a balance between model effectiveness and training efficiency and can select suitable normalizing flow models for different datasets. Detailed selections of flow models are reported in Appendix C.3.

**Prediction** The learned parameter distributions will facilitate prediction of upcoming events. The prediction results on synthetic datasets are presented in Table 2, from which one can observe that our model outperforms all baselines.

### 6.3 MIMIC-IV DATASET EXPERIMENTS

**Preprocessing** MIMIC-IV[1] is an electronic health record dataset of patients admitted to the intensive care unit (ICU) [Johnson et al., 2023]. We focused on patients diagnosed with sepsis [Saria, 2018], a leading cause of mortality in the ICU. Following the approach suggested by Komorowski et al. [2018], we selected 21 treatments categorized as vasopressors, antibiotics, and auxiliary treatment (details shown in Appendix E.1) from which a total of 7377 samples were extracted. Since normal urine reflects the

[1] https://mimic.mit.edu/

Table 1: Ablation study on synthetic and real-world datasets. Our current selection of modules are highlighted in blue. For synthetic datasets, we use 7500 samples with 3 dimensions cases. We ablate the following modules: *i) Delay*: whether assume that time lag (delay effect) presents in the data, *ii) Dist*: whether assume the parameters (impact $\boldsymbol{\alpha}$ and delay $\boldsymbol{\delta}$) of grounded Hawkes process follow certain distributions, *iii) Ensem*: whether ensemble multiple normalizing flows, and *iv) DiffBase*: whether vary the input base distributions.

**Synthetic Dataset**

| Delay. | Dist. | Ensem. | DiffBase. | Uni-Modal | | | | Multi-Modal | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | NLL ↓ | aKL ($\alpha$) ↓ | aKL ($\delta$) ↓ | Time ↓ | NLL ↓ | aKL ($\alpha$) ↓ | aKL ($\delta$) ↓ | Time ↓ |
| ✗ | ✗ | ✗ | ✗ | 32.62 | – | – | **0.15h** | 38.43 | – | – | **0.36h** |
| ✓ | ✗ | ✗ | ✗ | 28.64 | – | – | 0.16h | 37.35 | – | – | 0.40h |
| ✓ | ✓ | ✗ | ✗ | **25.08** | 1.26 | 2.20 | 0.18h | 36.52 | 4.33 | 3.67 | 0.42h |
| ✓ | ✓ | ✓ | ✗ | 25.26 | **1.22** | **2.16** | 0.21h | **30.42** | **3.16** | **2.25** | 0.56h |
| ✓ | ✓ | ✗ | ✓ | 26.52 | 1.37 | 2.32 | 0.20h | 33.97 | 3.86 | 3.11 | 0.52h |
| ✓ | ✓ | ✓ | ✓ | 25.71 | 1.32 | 2.28 | 0.38h | 33.68 | 3.82 | 2.95 | 0.67h |

**Real-World Dataset**

| Delay. | Dist. | Ensem. | DiffBase. | MIMIC-IV | | | Covid Policy | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | NLL ↓ | RMSE ↓ | Time ↓ | NLL ↓ | RMSE ↓ | Time ↓ |
| ✗ | ✗ | ✗ | ✗ | 26.15 | 3.52 | **0.18h** | 42.80 | 4.25 | **0.13h** |
| ✓ | ✗ | ✗ | ✗ | 24.52 | 3.20 | 0.24h | 39.46 | 4.08 | 0.17h |
| ✓ | ✓ | ✗ | ✗ | 22.55 | 2.92 | 0.31h | 37.80 | 3.93 | 0.22h |
| ✓ | ✓ | ✓ | ✗ | **21.32** | **2.86** | 0.34h | 36.94 | **3.35** | 0.25h |
| ✓ | ✓ | ✗ | ✓ | 22.08 | 2.95 | 0.38h | 37.56 | 3.72 | 0.30h |
| ✓ | ✓ | ✓ | ✓ | 21.67 | 2.90 | 0.53h | **36.25** | 3.54 | 0.42h |

impact of drugs and treatments on improving the physical condition of a patient, our objective is to uncover impact and delay effect of treatments on patients' physical well-being, as observed through normal urine events.

**Ablation Study** Since we are uncertain about the presence of delay effects and whether parameters adhere to specific distributions in real datasets, we need to validate our assumptions through ablation studies first. In Table 1, for the MIMIC-IV dataset, assuming no delay effect and fixed parameters results in a higher converged negative log-likelihood and decreased accuracy in prediction tasks, validating the rationale behind our current configuration.

**Case Study and Prediction** Shown in Figure 5, the positive impact of vasoconstrictors, antibiotics, and auxiliary treatment are comparable. The delay effect distribution of vasoconstrictors exhibits a right-skewed pattern, with a mean around 0.5 hours, indicating that vasoconstrictors show clearly short-term response to yield a positive impact on human circulatory systems. Antibiotics typically require longer time to take effect, with a time lag distribution in the population generally following a normal distribution centered around a mean of 1.2 hours. Due to auxiliary treatment encompassing various therapies such as Furosemide and Invasive Ventilation, its delay effect displays a multi-modal pattern with the first local peak around 0.3 hours and the second local peak appearing near 0.8 hours.

As in Table 2, our model accurately predicts the next normal urine event with the lowest RMSE than all other baselines.

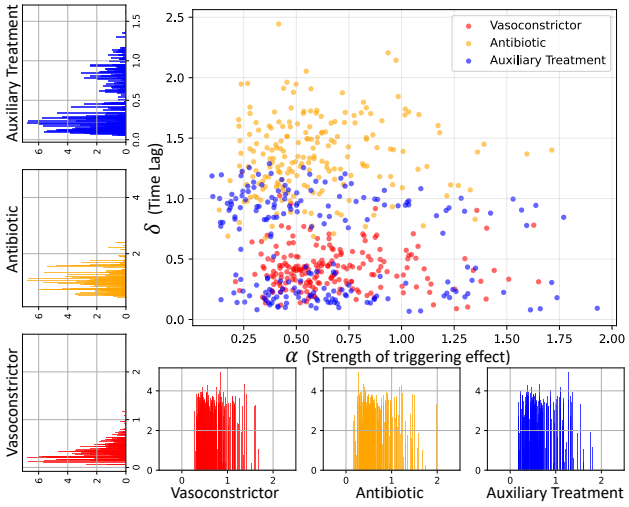


Figure 5: Learned impact $\boldsymbol{\alpha}$ and delay effect $\boldsymbol{\delta}$ distributions for MIMIC-IV dataset (**left and bottom bar plots**) and 2D scatter plot for samples of these two parameters (**top-right**).

### 6.4 COVID POLICY DATASET EXPERIMENTS

**Preprocessing** The Covid-19 Policy dataset [2] collects data on governments' implementation of specific measures and their timing to control COVID-19 pandemic [Hale et al., 2021, 2020]. Epidemic prevention policies are organized into 4 categories including Containment Closure, Healthcare System, Vaccination, and Economic policies, which could be referred to Appendix E.2. We conducted experiments

[2] https://github.com/OxCGRT/covid-policy-dataset

Table 2: Event time prediction RMSE ↓ on two synthetic datasets using 7500 samples with 3 dimensions case (denoted as "Uni-Modal" and "Multi-Modal" to indicate different underlying parameter distributions), MIMIC-IV data (evaluating prediction of occurrence time of normal urine event of patients) and Covid Policy data (evaluating prediction of occurrence time of dropping confirmed cases/infections). Results from our model are shaded in red.

| Category | Method | Uni-Modal | Multi-Modal | MIMIC-IV | Covid Policy |
|---|---|---|---|---|---|
| Non-Param. | GM-NLF [Eichler et al., 2017] | 2.36 | 2.72 | 4.29 | 6.72 |
| | MMEL [Zhou et al., 2013] | 2.41 | 2.85 | 4.47 | 6.45 |
| | Gibbs-Hawkes [Zhang et al., 2018] | 1.98 | 2.64 | 3.87 | 6.12 |
| Param. | RMTPP [Du et al., 2016] | 2.15 | 2.77 | 3.82 | 5.24 |
| | THP [Zuo et al., 2020] | 1.92 | 2.46 | 3.26 | 5.08 |
| | PromptTPP [Xue et al., 2023] | 1.85 | 2.40 | 3.13 | **3.18** |
| | HYPRO [Xue et al., 2022] | 1.89 | 2.37 | 3.05 | 3.42 |
| | MLE-SGL [Xu et al., 2016] | 1.96 | 2.57 | 3.63 | 5.81 |
| | GC-CGD [Wei et al., 2022] | 1.90 | 2.45 | 3.18 | 5.26 |
| | **Ours*** | **1.79** | **2.25** | **2.86** | 3.35 |

on data from Australia and France for the years 2021-2022 based on the most severe COVID-19 situations and effective governmental policies, aiming to investigate the impact of the policies from different categories on event of dropping daily average number of confirmed cases.

**Ablation Study** Like in MIMIC-IV, the ablation study in Table 1 also shows that assuming the existence of delay effects in the data and parameters following specific distributions indeed enhances model performance, validating that the data align with our assumptions.

**Case Study and Prediction** In Figure 9 and 10, Appendix E.2, overall, the lag for the effectiveness of government policies in Australia seems shorter than in France. Here, we take Containment Closure (CC) policies as examples, whose positive impact is normally distributed and is almost larger than all other policies in these two countries. In Australia, when government enforces CC policies, the population generally divides into two groups (exhibiting two peaks in the distribution). One group promptly complies with isolation measures, leading to a decrease in new cases of COVID-19 around 5 days after policy implementation. The other group responds more slowly, requiring approximately 6.5 days for the policies to take effect. The pattern of CC policies in France is different, roughly following a normal distribution with a mean of 7.5 days. In terms of Healthcare System policies, delay effects in Australia also exhibit a bimodal distribution, with peaks at around 7 and 9 days, whereas in France with longer onset times, approximately 7.5 and close to 11 days, respectively, but the variance is smaller. In both countries, the impact of Vaccination and Economic policies is smaller than that of the two policies mentioned above. In Australia, the time lag for Vaccination to take effect typically peaks at 10 and 11 days, while for Economic policies, the mean time lag is 14.5 days. In France, the mean time lags are approximately 11.5 and 15 days, respectively, for these two policies.

Our model also has demonstrated competitive prediction performance on the COVID policy dataset. As in Table 2, the RMSE of predicting the time of next infection dropping event is the second lowest among all baselines, closely approaching the lowest.

### 6.5 OTHER REAL-WORLD DATASET EXPERIMENTS

In addition to the healthcare datasets, we also considered the StackOverflow [Leskovec and Krevl, 2014], Taobao [Xue et al., 2022], and Taxi [Whong, 2014] datasets for prediction tasks (predicting both the next event type and its time, while incorporating per-event negative log-likelihood and event type prediction error rate as extra evaluation metrics) to ensure broader applicability and validate our approach across diverse domains. As shown in Table 13, Appendix E.3, our model maintains strong performance across these baselines on almost all datasets. This also suggests that StackOverflow and Taobao datasets may inherently exhibit delayed event-triggering effects (while Taxi dataset may not) among different event types, further validating our approach.

## 7 CONCLUSION

In this paper, we propose the Flow-based Delayed Hawkes Process, an extension of multivariate Hawkes models that uses normalizing flows to flexibly model the distribution of parameters, capturing heterogeneous event dynamics while preserving interpretability. We provide theoretical guarantees on parameter identifiability and MLE consistency under mild conditions. Experiments on synthetic and real-world data demonstrate consistent superiority over state-of-the-art baselines in modeling diverse temporal event patterns. This work advances accurate and interpretable analysis of event data with delay effects and complex triggering behaviors.

**Acknowledgements**

Shuang Li's research was in part supported by the Key Program of the NSFC under grant No. 72495131, NSFC

under grant No. 62206236, Shenzhen Stability Science Program 2023, Shenzhen Science and Technology Program ZDSYS20230626091302006, Longgang District Key Laboratory of Intelligent Digital Economy Security and SRIBD Innovation Fund SIF20240010.

## References

Lucas Berry and David Meger. Normalizing flow ensembles for rich aleatoric and epistemic uncertainty modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6806–6814, 2023.

Vinod Kumar Chauhan, Jiandong Zhou, Ping Lu, Soheila Molaei, and David A Clifton. A brief review of hypernetworks in deep learning. *arXiv preprint arXiv:2306.06955*, 2023.

Ricky TQ Chen, Jens Behrmann, David K Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. *Advances in Neural Information Processing Systems*, 32, 2019.

Rainer Dahlhaus and Michael Eichler. Causality and graphical models in time series analysis. *Oxford Statistical Science Series*, pages 115–137, 2003.

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

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

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.

Manisha Dubey, PK Srijith, and Maunendra Sankar Desarkar. Hyperhawkes: Hypernetwork based neural temporal point process. *arXiv preprint arXiv:2210.00213*, 2022.

Manisha Dubey, PK Srijith, and Maunendra Sankar Desarkar. Time-to-event modeling with hypernetwork based hawkes process. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3956–3965, 2023.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.

Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. Generative multi-adversarial networks. *arXiv preprint arXiv:1611.01673*, 2016.

Michael Eichler, Rainer Dahlhaus, and Johannes Dueck. Graphical modeling for multivariate hawkes processes with nonparametric link functions. *Journal of Time Series Analysis*, 38(2):225–242, 2017.

Tian Gao, Dharmashankar Subramanian, Debarun Bhattacharjya, Xiao Shou, Nicholas Mattei, and Kristin P Bennett. Causal inference for event pairs in multivariate point processes. *Advances in Neural Information Processing Systems*, 34:17311–17324, 2021.

Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

Thomas Hale, S Webster, A Petherick, T Phillips, and B Kira. Oxford covid-19 government response tracker (oxcgrt). *Last updated*, 8:30, 2020.

Thomas Hale, Noam Angrist, Rafael Goldszmidt, Beatriz Kira, Anna Petherick, Toby Phillips, Samuel Webster, Emily Cameron-Blake, Laura Hallas, Saptarshi Majumdar, et al. A global panel database of pandemic policies (oxford covid-19 government response tracker). *Nature human behaviour*, 5(4):529–538, 2021.

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

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. Mgan: Training generative adversarial nets with multiple generators. In *International conference on learning representations*, 2018.

Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Benjamin Moody, Brian Gow, Li-wei H Lehman, et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific Data*, 10(1):1–9, 2023.

Sobin Joseph, Lekhapriya Dheeraj Kashyap, and Shashi Jain. Shallow neural hawkes: Non-parametric kernel estimation for hawkes processes. *arXiv preprint arXiv:2006.02460*, 2020.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

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

Matthieu Komorowski, Leo A Celi, Omar Badawi, Anthony C Gordon, and A Aldo Faisal. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature medicine*, 24(11): 1716–1720, 2018.

Shinsuke Koyama, Taiki Horie, and Shigeru Shinomoto. Estimating the time-varying reproduction number of covid-19 with a state-space method. *PLoS computational biology*, 17(1):e1008679, 2021.

Laurent Lesage, Madalina Deaconu, Antoine Lejay, Jorge Augusto Meira, Geoffrey Nichil, and Radu State. Hawkes processes framework with a gamma density as excitation function: application to natural disasters for insurance. *Methodology and Computing in Applied Probability*, 24(4):2509–2537, 2022.

Jure Leskovec and Andrej Krevl. Snap datasets: Stanford large network dataset collection, 2014.

Erik Lewis and George Mohler. A nonparametric em algorithm for multiscale hawkes processes. *Journal of nonparametric statistics*, 1(1):1–20, 2011.

Erik Lewis, George Mohler, P Jeffrey Brantingham, and Andrea L Bertozzi. Self-exciting point process models of civilian deaths in iraq. *Security Journal*, 25:244–264, 2012.

Nazanin Mehrasa, Ruizhi Deng, Mohamed Osama Ahmed, Bo Chang, Jiawei He, Thibaut Durand, Marcus Brubaker, and Greg Mori. Point process flows. *arXiv preprint arXiv:1910.08281*, 2019a.

Nazanin Mehrasa, Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. A variational auto-encoder model for stochastic point processes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3165–3174, 2019b.

Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems*, 30, 2017.

Hongyuan Mei, Guanghui Qin, Minjie Xu, and Jason Eisner. Neural datalog through time: Informed temporal modeling via logical specification. In *International Conference on Machine Learning*, pages 6808–6819. PMLR, 2020.

Gonçalo Mordido, Haojin Yang, and Christoph Meinel. microbatchgan: Stimulating diversity with multi-adversarial discrimination. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3061–3070, 2020.

Tu Nguyen, Trung Le, Hung Vu, and Dinh Phung. Dual discriminator generative adversarial nets. *Advances in neural information processing systems*, 30, 2017.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

JA Quesada, A López-Pineda, VF Gil-Guillén, JM Arriero-Marín, F Gutiérrez, and C Carratala-Munuera. Incubation period of covid-19: A systematic review and meta-analysis. *Revista Clínica Española (English Edition)*, 221 (2):109–117, 2021.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

Tiago Santos, Florian Lemmerich, and Denis Helic. Bayesian estimation of decay parameters in hawkes processes. *Intelligent Data Analysis*, 27(1):223–240, 2023.

Suchi Saria. Individualized sepsis treatment using reinforcement learning. *Nature medicine*, 24(11):1641–1642, 2018.

Oleksandr Shchur, Marin Biloš, and Stephan Günnemann. Intensity-free learning of temporal point processes. *arXiv preprint arXiv:1909.12127*, 2019.

Song Wei, Yao Xie, Christopher S Josef, and Rishikesan Kamaleswaran. Granger causal chain discovery for sepsis-associated derangements via multivariate hawkes processes. *arXiv preprint arXiv:2209.04480*, 2022.

Spencer Wheatley, Vladimir Filimonov, and Didier Sornette. Estimation of the hawkes process with renewal immigration using the em algorithm. *arXiv preprint arXiv:1407.7118*, 2014.

Chris Whong. Foiling nyc's taxi trip data. *FOILing NYCs Taxi Trip Data. Np*, 18:14, 2014.

Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen Chu. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning granger causality for hawkes processes. In *International conference on machine learning*, pages 1717–1726. PMLR, 2016.

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

Siqiao Xue, Yan Wang, Zhixuan Chu, Xiaoming Shi, Caigao Jiang, Hongyan Hao, Gangwei Jiang, Xiaoyun Feng, James Y Zhang, and Jun Zhou. Prompt-augmented temporal point process for streaming event sequence. *arXiv preprint arXiv:2310.04993*, 2023.

Chenghao Yang, Hongyuan Mei, and Jason Eisner. Transformer embeddings of irregularly spaced events and their participants. *arXiv preprint arXiv:2201.00044*, 2021.

Changyong Zhang. Modeling high frequency data using hawkes processes with power-law kernels. *Procedia Computer Science*, 80:762–771, 2016.

Lu-ning Zhang, Jian-wei Liu, and Xin Zuo. Survival analysis of failures based on hawkes process with weibull base intensity. *Engineering Applications of Artificial Intelligence*, 93:103709, 2020a.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes process. In *International conference on machine learning*, pages 11183–11193. PMLR, 2020b.

Rui Zhang, Christian Walder, Marian-Andrei Rizoiu, and Lexing Xie. Efficient non-parametric bayesian hawkes processes. *arXiv preprint arXiv:1810.03730*, 2018.

Ke Zhou, Hongyuan Zha, and Le Song. Learning triggering kernels for multi-dimensional hawkes processes. In *International conference on machine learning*, pages 1301–1309. PMLR, 2013.

Shixiang Zhu, Minghe Zhang, Ruyi Ding, and Yao Xie. Deep fourier kernel for self-attentive point processes. In *International Conference on Artificial Intelligence and Statistics*, pages 856–864. PMLR, 2021.

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 OVERVIEW

In the following, we will provide supplementary materials to better illustrate our methods and experiments.

- Section A provides theoretical guarantees.
- Section B presents more details of our model and implementation.
- Section C reports the reproducibility analysis.
- Section D provides more synthetic dataset experiments and corresponding analysis.
- Section E provides more real-world dataset experiments and corresponding analysis.
- Section F states the limitations and broader impacts of our proposed model.

# A THEORETICAL PROOFS

## A.1 PROOF OF THEOREM 2

**Proof** [Theorem 2] Define the operator $\mathcal{T}$ that maps a latent distribution $p(\boldsymbol{\theta})$ to its marginal intensity:

$$\mathcal{T}(p)(t) = \int_{\Theta} f_u(t \mid \mathcal{H}_t; \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}.$$

By assumption, $\mathcal{T}(p) = \mathcal{T}(q)$ for almost every $t$, so defining $g(\boldsymbol{\theta}) = p(\boldsymbol{\theta}) - q(\boldsymbol{\theta})$, we obtain:

$$\int_{\Theta} f_u(t \mid \mathcal{H}_t; \boldsymbol{\theta}) g(\boldsymbol{\theta}) d\boldsymbol{\theta} = 0 \quad \text{for almost every } t.$$

By the completeness assumption, this implies $g(\boldsymbol{\theta}) = 0$ almost everywhere, concluding that $p(\boldsymbol{\theta}) = q(\boldsymbol{\theta})$ almost everywhere. ■

## A.2 PROOF OF COMPLETENESS FOR THE SMOOTH TRIGGERING FUNCTION

**Proof** [Completeness for the Smooth Triggering Function] Consider the intensity function:

$$f_u\left(t \mid \mathcal{H}_t; \boldsymbol{\theta}\right) = \mu_u + \sum_{u'=1}^{U} \sum_{n=1}^{N_u(t)} \alpha_{uu'} h\left(-\beta\left(t - t_n^{u'} - \delta_{uu'}\right)\right)$$

where $\mu_u \geq 0$, $\alpha_{uu'} \geq 0$, $\beta > 0$, $\delta_{uu'} \geq 0$, and $h(\cdot)$ is a smooth function (i.e., sigmoid-exponential product). We prove the family $f_u\left(t \mid \mathcal{H}_t; \boldsymbol{\theta}\right)$ is complete, i.e., if

$$\int_{\Theta} f_u\left(t \mid \mathcal{H}_t; \boldsymbol{\theta}\right) p(\boldsymbol{\theta}) d\boldsymbol{\theta} = 0 \quad \forall t$$

then $p(\boldsymbol{\theta}) = 0$ almost everywhere.

Substitute $f_u\left(t \mid \mathcal{H}_t; \boldsymbol{\theta}\right)$ into the integral equation:

$$\int_{\Theta} \mu_u p(\boldsymbol{\theta}) d\boldsymbol{\theta} + \sum_{u'=1}^{U} \sum_{n=1}^{N_u(t)} \int_{\Theta} \alpha_{uu'} h\left(-\beta\left(t - t_n^{u'} - \delta_{uu'}\right)\right) p(\boldsymbol{\theta}) d\boldsymbol{\theta} = 0 \quad \forall t$$

The first term is $t$-independent, while the second term depends on $t$ through $h(\cdot)$. For equality to hold globally, both terms must vanish individually.

The $t$-independence of the first term implies:

$$\int_{\Theta} \mu_u p(\boldsymbol{\theta}) d\boldsymbol{\theta} = 0$$

Since $\mu_u \geq 0$, this forces $\int_{\Theta} p(\boldsymbol{\theta}) d\boldsymbol{\theta} = 0$. For the second term, smoothness and the parametric structure of $h(\cdot)$ ensure the family $\left\{h\left(-\beta\left(t - t_n^{u'} - \delta_{uu'}\right)\right)\right\}$ is linearly independent for distinct $\left(\beta, \delta_{uu'}, t_n^{u'}\right)$. By the Haar condition (for Chebyshev systems), a nontrivial linear combination of these functions cannot vanish identically unless all coefficients are zero.

The integral equation reduces to a moment problem:

$$\sum_{u'=1}^{U} \sum_{n=1}^{N_u(t)} \int_{\Theta} \alpha_{uu'} h\left(-\beta\left(t - t_n^{u'} - \delta_{uu'}\right)\right) p(\boldsymbol{\theta}) d\boldsymbol{\theta} = 0 \quad \forall t$$

Because $h(\cdot)$ generates a complete basis (via exponential/sigmoid properties), the only solution is $p(\boldsymbol{\theta}) = 0$ almost everywhere.

As a conclusion, the linear independence of $\left\{h\left(-\beta\left(t-t_n^{u'}-\delta_{uu'}\right)\right)\right\}$ and the Haar condition ensure uniqueness. Hence, the family $\{f_u\left(t \mid \mathcal{H}_t;\boldsymbol{\theta}\right)\}$ is complete, and:

$$\lambda_u\left(t \mid \mathcal{H}_t; p(\boldsymbol{\theta})\right) = \lambda_u\left(t \mid \mathcal{H}_t; q(\boldsymbol{\theta})\right) \Longrightarrow p(\boldsymbol{\theta}) = q(\boldsymbol{\theta})$$

■

### A.3 PROOF OF THEOREM 3

**Proof** [Proof Sketch of Theorem 3] First, we show the **uniform convergence**: Under standard regularity conditions for point processes, the empirical log-likelihood function converges uniformly (by the law of large numbers) to its expected value as the observation window $T$ grows. That is, for all candidate distributions $p$,

$$\frac{1}{T}\mathcal{L}\left(p\right) \to \mathbb{E}\left[\frac{1}{T}\mathcal{L}\left(p\right)\right] \quad \text{almost surely}$$

Second, we have proved the **identifiability**. By Theorems 1 and 2, the mapping from the latent parameters $\boldsymbol{\theta}$ to the intensity function $f_u\left(t \mid \mathcal{H}_t;\boldsymbol{\theta}\right)$ is injective, and the family $\mathcal{F}$ is complete. Therefore, the expected $\log$ likelihood has a unique maximum at the true distribution $p^*(\boldsymbol{\theta})$.

Then, we can use the standard MLE consistency results: The standard Wald's consistency theorem implies that the our maximizer of the empirical log likelihood, $\hat{p}(\boldsymbol{\theta})$, converges in probability to $p^*(\boldsymbol{\theta})$ as $T \to \infty$. ■

## B IMPLEMENTATION DETAILS

### B.1 DYNAMIC SIGMOID MASK MODULE

We want to emphasize that the exponential kernel of Hawkes process contains an indicator function, which would result in the interruption of gradients during backpropagation. To address this issue, we propose a dynamic sigmoid mask module,

$$\mathbb{I}(t-t_n^{u'}-\delta_{uu'} \geq 0) := \text{sigmoid}(\frac{C}{\gamma_t}\cdot(t-t_n^{u'}-\delta_{uu'})) \tag{11}$$

where $C$ is a large constant and $\gamma_t$ is the cyclical annealing temperature, which is given by

$$\gamma_t = \begin{cases} h(\tau), & \tau \leq R \\ c, & \tau > R \end{cases} \quad \text{with} \quad \tau = \frac{\text{mod}(t-1, \lceil B/M \rceil)}{B/M} \tag{12}$$

where $t$ is the iteration number, $B$ is the total number of iteration, $c < C$ is a fixed constant, $h(\cdot)$ is a monotonically increasing function with value start with 1, $M$ is the number of cycles, and $R$ represents the proportion used to increase $\gamma$ within a cycle. In other words, we split the training process into $M$ cycles, each starting with $\gamma = 1$ and ending with $\gamma = C$. Within one cycle, there are two consecutive stages (divided by $R$), one is the annealing stage and the other is the fixing stage. This ensures that the output of the dynamic sigmoid mask module approximates the binary output (0 or 1) of the original indicator function while preserving gradient flow and preventing gradient stagnation.

In our implementation, we can confirm that our model strictly enforces temporal causality by: (i) Explicit temporal masking: we apply strict masking to ensure that only events where $t_n^{u'} < t$ can contribute to the intensity function at time $t$. This masking is applied immediately after computing the sigmoid values but before they contribute to the intensity calculation. (ii) Batched computation structure: while we do parallelize kernel computations for efficiency, the implementation enforces a strict time-ordering constraint. The code includes explicit conditional filtering that zeros out any influence from event times $t_n^{u'}$ that occur after the evaluation time $t$. (iii) Training evaluation consistency: this masking remains consistent between training and evaluation phases, regardless of the annealing schedule of the sigmoid temperature parameters.

### B.2 COMPUTATION OF KL DIVERGENCE

Due to the inherent characteristics of different deep generative models, we must standardize the KL divergence computation metric to ensure a fair comparison of their performance. For our flow-based model, we can obtain samples and corresponding learned densities from well-trained model. The average KL divergence can be directly computed according to Eq. (9). Hypernet can only generate samples but cannot yield corresponding densities. To align with current computation approach of KL divergence, after obtaining the samples from well-trained Hypernet model, we fit the learned distributions based on samples and then get the corresponding learned densities. Note that in this process, we assume the distribution format is known for Hypernet samples. For $\beta$-VAE, we directly obtain the learned parameter distribution from the latent representation. Therefore, we can sample from the latent distributions and know the corresponding learned densities.

For our flow-based model, Hypernet model, and $\beta$-VAE, we plug the samples from well-trained models into the ground truth distributions to get the corresponding ground truth densities for these samples so that we can compute the KL divergence

according to Eq. (9).

# C REPRODUCIBILITY ANALYSIS

## C.1 BASELINES

**Parameter Distribution Learning Tasks**

- Hypernet [Ha et al., 2016, Chauhan et al., 2023]: We utilize hypernets to obtain the samples and use these samples to compute likelihood of Hawkes process and therefore backward training the hypernet.
- $\beta$-VAE [Higgins et al., 2017]: We utilize the latent representation of $\beta$-VAE to estimate the parameter distributions of Hawkes processes.

**Comparsion of Different Flow Models**

- Planer [Rezende and Mohamed, 2015]: For this model, the approximations of distributions are through a normalizing flow, whereby transforming a simple initial density into a more complex one by applying a sequence of invertible transformations until a desired level of complexity is attained.
- RealNVP [Dinh et al., 2016]: It uses real-valued non-volume preserving (Real NVP) transformations, which are stably invertible and learnable transformations.
- Glow [Kingma and Dhariwal, 2018]: It is a simple type of generative flow using an invertible $1 \times 1$ convolution.
- RQ-NSF (Rational-Quadratic Neural Spline Flow) [Durkan et al., 2019]: A fully-differentiable module based on monotonic rational-quadratic splines, which enhances the flexibility of both coupling and autoregressive transforms while retaining analytic invertibility.
- ResFlow (Residual Flow) [Chen et al., 2019]: A flow-based generative model that produces an unbiased estimate of the log density and has memory-efficient backpropagation through the log density computation, which allows us to use expressive architectures and train via maximum likelihood.

**Prediction Tasks**

- **Non-parametric Models**
  - GM-NLF [Eichler et al., 2017]: It shows that the Granger causality structure of the process is fully encoded in the corresponding Hawkes kernels. It introduces a new nonparametric estimator of the Hawkes kernels based on a time-discretized version of the point process by using an infinite order autoregression. And it derived the consistency and asymptotic normality of the estimator.
  - MMEL [Zhou et al., 2013]: The proposed model focuses on the nonparametric learning of the triggering kernels for multi-dimensional Hawkes processes, and the proposed algorithm combines the idea of decoupling the parameters through constructing a tight upper-bound of the objective function and application of Euler Lagrange equations for optimization in infinite dimensional functional space.
  - Gibbs-Hawkes [Zhang et al., 2018]: An efficient nonparametric Bayesian estimation method of the kernel function of Hawkes processes. This method is based on the cluster representation of Hawkes processes. Utilizing the finite support assumption of the Hawkes process, it efficiently samples random branching structures, and thus, splits the Hawkes process into clusters of Poisson processes. By using the a block Gibbs sampler, the samples building the estimation can converge to the desired posterior.
- **Parametric Models**
  - RMTPP [Du et al., 2016]: The approach considers the intensity function of a temporal point process as a nonlinear function that depends on the history. It utilizes a recurrent neural network to automatically learn a representation of the influences from the event history, which includes past events and time intervals, thereby fitting the intensity function of the temporal point process.
  - THP [Zuo et al., 2020]: The model employs a concurrent self-attention module to embed historical events and generate hidden representations for discrete time stamps. These hidden representations are then used to model the interpolated continuous time intensity function. THP can also incorporate additional structural knowledge. Importantly, THP surpasses RNN-based approaches in terms of computational efficiency and the ability to capture long-term dependencies.
  - PromptTPP [Xue et al., 2023]: The model incorporates a continuous-time retrieval prompt pool into the base TPP, enabling sequential learning of event streams without the need for buffering past examples or task-specific attributes. Specifically, this approach consists of a base TPP model, a pool of continuous-time retrieval prompts, and a prompt-event interaction layer. By addressing the challenges associated with modeling streaming event sequences, this mode enhances the model's performance.

- HYPRO [Xue et al., 2022]: The hybridly normalized probabilistic (HYPRO) model is capable of making long-horizon predictions for event sequences. This model consists of two modules: the first module is an auto-regressive base TPP model that generates prediction proposals, while the second module is an energy function that assigns weights to the proposals, prioritizing more realistic predictions with higher probabilities. This design effectively mitigates the cascading errors commonly experienced by auto-regressive TPP models in prediction tasks, thereby improving the model's accuracy in long-term forecasting.
- MLE-SGL [Xu et al., 2016]: It proposes an effective method to learn the Granger causality for Hawkes process. The model represents impact functions using a series of basis functions and recovers the Granger causality graph via group sparsity of the impact functions' coefficients. The proposed learning algorithm combines a maximum likelihood estimator (MLE) with a sparse group-lasso (SGL) regularizer. Additionally, the flexibility of the model allows to incorporate the clustering structure event types into learning framework.
- GC-CGD [Wei et al., 2022]: This work proposes a linear Hawkes process model, coupled with ReLU link function to recover a Granger Causal graph with both exciting and inhibiting effects. The method is a scalable two-phase gradient-based method to obtain a maximum surrogate-likelihood estimator. In the first phase, it constrains all parameters to be non-negative and perform projected gradient descent with fixed step length. In the second phase, it performs batch coordinate gradient descent on those variables whose corresponding rows (in the trigerring effect matrix) could have negative values.

### C.2 COMPUTING INFRASTRUCTURE

All the experiments for both synthetic dataset experiments and real-world dataset experiments, including the comparison experiments with baselines, are performed on Ubuntu 20.04.3 LTS system with Intel(R) Xeon(R) Gold 6248R CPU @ 3.00GHz, 227 Gigabyte memory.

### C.3 HYPER-PARAMETERS SELECTION

Our model is easy to implement and reproduce the results. We present the selected hyper-parameters on synthetic and real-world datasets in Table 3. The hyper-parameter selection metric is a trade-off between training converged log-likelihood, prediction performance, and time efficiency.

Table 3: Descriptions and values of hyper-parameters used for models trained on the synthetic and real-world datasets.

| Hyper-parameters | Value Used | | | |
|---|---|---|---|---|
| | Syn-Data (Uni-Modal) | Syn-Data (Multi-Modal) | MIMIC-IV | Covid Policy Tracker |
| Max Epochs | 128 | 128 | 256 | 256 |
| Batch Size | 64 | 64 | 64 | 64 |
| Hidden Size | 32 | 32 | 32 | 32 |
| # NFs Ensembled | 2 | 2 | 3 | 3 |
| # Layers for single NF | 6 | 6 | 8 | 8 |
| # Samples for single NF | 100 | 100 | 100 | 100 |
| Base Dist. | $\mathcal{N}(0,1)$ | $\mathcal{N}(0,1)$ | $\mathcal{N}(0,1)$ | $\mathcal{N}(0,1)$ |
| Learning Rate | 1e-3 | 1e-3 | 5e-4 | 5e-4 |
| Optimizer | Adam | Adam | Adam | Adam |
| Flow Model | RealNVP | RealNVP | ResFlow | RealNVP |

## D MORE SYNTHETIC DATASET EXPERIMENTS

### D.1 COMPLETE VISUALIZATION EXAMPLES COMPARING THE PERFORMANCE ON PARAMETER DISTRIBUTION LEARNING TASKS

In Figure 6, we report the complete visualization results of the learned marginal distribution for target dimension ($u = 3$), e.g., $\alpha_{31}, \alpha_{32}, \alpha_{33}$ and $\delta_{31}, \delta_{32}, \delta_{33}$, using 3-dimensional datasets with 7500 samples. The results demonstrate that our model not only accurately captures uni-modal distributions but also performs well on multi-modal distributions, significantly outperforming Hypernet and $\beta$-VAE.

To test the scalability and fairly compare different deep generative models for learning parameter distributions in our problem setting, we vary the size of training samples within $\{2500, 5000, 7500, 10000, 12500, 15000\}$. The results are shown in Table 4, from which one can observe that our model consistently outperforms Hypernet and $\beta$-VAE in almost all cases.

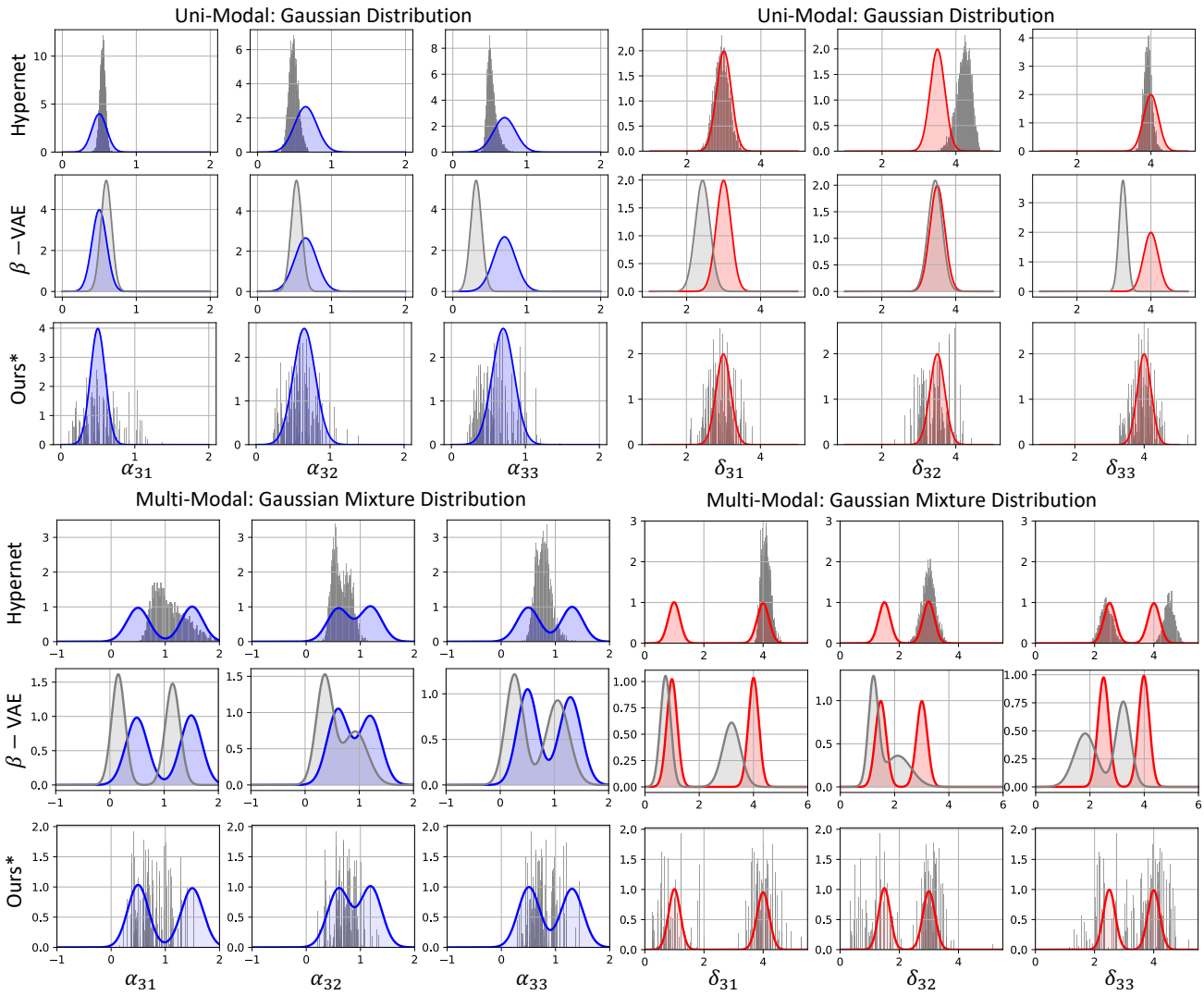


Figure 6: Visualization examples comparing different models on parameter distribution learning tasks with 3-dimensional datasets and 7500 samples. We report the learned marginal distribution for target dimension ($u = 3$) in these figures.

As we build joint distributions of the parameters in the generating process of synthetic datasets, our flow-based model inherently can effectively capture their dependencies. Depicted in Figure 7, the samples of $\boldsymbol{\alpha}$ and $\boldsymbol{\delta}$ from our well-trained model basically match ground truth joint densities.

### D.2 COMPARE WITH OTHER FLOW-BASED MODELS

In Table 5, we compare the model performance produced by different flow-based models. In our setting, simple normalizing flow models like RealNVP are capable of uncovering ground truth parameter distributions. When employing dense flows, although there is an enhancement in model performance, it requires excessive computational resources. In practical applications, we must strike a balance between model effectiveness and training efficiency. Taking into account factors such as data volume and dimensionality, our model can select suitable normalizing flow models. Detailed selections of flow models for synthetic datasets and real-world datasets experiments can be found in Appendix C.3.

### D.3 COMPARE WITH TRADITIONAL PARAMETRIC MODELS

Our use of normalizing flows as complex priors offers two key advantages. The first one is flexible modeling of any irregular distributions (e.g., skewed patterns) that capture population variance complexity. The second one is stronger exploration and expressive power compared to traditional parametric models – though requiring larger datasets. To further illustrate this, we have added more experiments: we consider conventional probabilistic models with simple priors, including mixture of uniform and mixture of gaussian models (abbreviated as “MoU” and “MoG” respectively). As shown in Table 6 and Table 7, our flow-based model demonstrates consistent superiority over traditional approaches across synthetic dataset

Table 4: Compare the accuracy of learned parameter distributions across different models using **KL divergence** as metric with varying sample sizes. Bold signifies the best result, while underlined text indicates the second-best result.

| Model | Uni-Modal | | | | | | Multi-Modal | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 2500 | 5000 | 7500 | 10000 | 12500 | 15000 | 2500 | 5000 | 7500 | 10000 | 12500 | 15000 |
| Hypernet ($\alpha$) | 5.51 | 8.22 | 6.70 | 5.08 | 5.09 | 5.30 | 39.68 | 37.92 | 33.95 | 33.53 | 33.30 | 32.61 |
| $\beta$-VAE ($\alpha$) | 1.83 | **1.29** | 1.53 | 1.26 | 1.24 | 1.19 | 5.12 | 4.55 | 4.42 | **3.23** | 2.94 | 2.87 |
| **Ours*** ($\alpha$) | **1.48** | 1.42 | **1.22** | **1.17** | **1.05** | **0.91** | **4.62** | **4.18** | **3.16** | 3.38 | **2.79** | **2.52** |
| Hypernet ($\delta$) | 6.22 | 3.54 | 3.99 | 1.57 | 1.47 | **1.38** | 14.79 | 11.74 | 7.86 | 5.95 | 11.24 | 8.60 |
| $\beta$-VAE ($\delta$) | **3.62** | 4.17 | 3.22 | 1.69 | **1.37** | 1.52 | **2.88** | **2.74** | 2.43 | 2.37 | 2.19 | 2.10 |
| **Ours*** ($\delta$) | 3.83 | **2.57** | **2.16** | **1.56** | 1.43 | 1.45 | 2.92 | 2.85 | **2.25** | **2.29** | **1.86** | **1.63** |

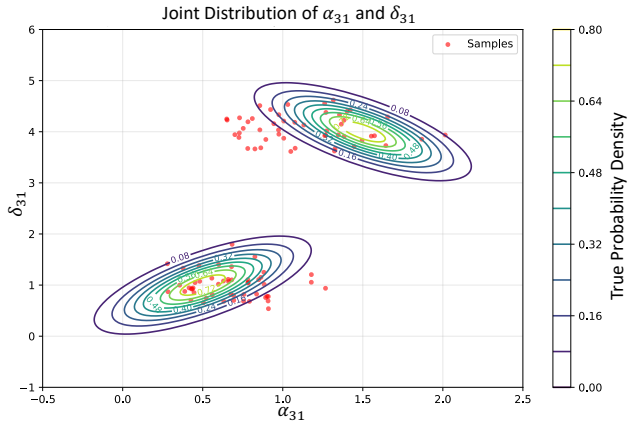


Figure 7: True joint distribution (**contours**) of $\alpha_{31}$ and $\delta_{31}$ and samples (**red circles**) from our well-trained model using multi-modal dataset with 3-dimensional and 7500 samples.

distribution learning and real-world prediction tasks, compared with mixture of uniform and mixture of gaussian models. Furthermore, our model exhibits improved performance with increased training data, as quantitatively verified by the lower KL divergence.

### D.4 EXPERIMENTS ON SYNTHETIC DATASETS WITH VARYING DECAY

While estimating extra distributions add computational complexity, it is still achievable. We have extended our synthetic datasets with varying $\beta$ for different event types. While we observe marginal accuracy declines in learning the distributions of $\alpha$, $\beta$, and $\delta$ on datasets with varying $\beta$ distributions, the overall performance remains satisfactory. Importantly, our flow-based model consistently demonstrates superior performance compared to Hypernet and $\beta$-VAE across all experimental settings in parameter distribution learning tasks, and our flow-based model also outperforms other baseline models in prediction tasks, as shown in Table 8 and Table 9 respectively.

### D.5 SCALABILITY EXPERIMENTS

To evaluate the scalability of our proposed model, we vary the dimensionality within $\{2, 3, 5, 7, 9\}$ and sample sizes within $\{2500, 5000, 7500, 10000, 12500, 15000\}$. Our model demonstrates high efficiency and good scalability, converging within 1.2 hours even in the most complex scenarios, utilizing 15000 samples of training data with 9 dimensions. Shown in Figure 8, as the training sample size increases, the training time increases while the converged negative log-likelihood decreases, and distribution learning accuracy increases accordingly. As the dimensionality of Hawkes processes increases, the distribution learning accuracy of our model may slightly decrease but remains satisfactory. Encouragingly, as the training sample size grows, the learning performance becomes stable.

## E MORE REAL-WORLD DATASET EXPERIMENTS

### E.1 HEALTHCARE DATA EXPERIMENTS – MIMIC-IV

MIMIC-IV[3] is a publicly available database sourced from the electronic health record of the Beth Israel Deaconess Medical Center [Johnson et al., 2023]. Available information includes patient measurements, orders, diagnoses, procedures,

[3]https://mimic.mit.edu/

Table 5: Compare different normalizing flow models. We take uni-modal distribution dataset with 3 dimensions and 7500 samples as an example.

| Model | NLL ↓ | aKL ($\alpha$) | aKL ($\delta$) | Time ↓ |
|---|---|---|---|---|
| Planerr [Rezende and Mohamed, 2015] | 29.52 | 1.56 | 2.83 | **0.18h** |
| Glow [Kingma and Dhariwal, 2018] | 25.70 | 1.32 | 2.28 | 0.38h |
| RealNVP [Dinh et al., 2016] | <u>25.26</u> | <u>1.22</u> | **2.16** | <u>0.21h</u> |
| RQ-NSF [Durkan et al., 2019] | 25.54 | 1.24 | 2.23 | 0.27h |
| ResFlow [Chen et al., 2019] | **24.93** | **1.18** | <u>2.20</u> | 0.56h |

Table 6: Compare the distribution learning ability using varying training sample size (on multi-modal synthetic datasets) between our flow-based model and other mixture models. The comparison metric is the average KL divergence between learned distributions and ground truth distributions.

| Metric | aKL ($\alpha$) | | | aKL ($\delta$) | | |
|---|---|---|---|---|---|---|
| | 2500 | 7500 | 12500 | 2500 | 7500 | 12500 |
| MoU | 38.27 | 35.10 | 34.33 | 21.82 | 15.40 | 13.36 |
| MoG | <u>5.45</u> | <u>5.08</u> | <u>4.93</u> | <u>3.79</u> | <u>3.21</u> | <u>2.85</u> |
| **Ours*** | **4.62** | **3.16** | **2.79** | **2.92** | **2.25** | **1.86** |

treatments, and deidentified free-text clinical notes. Sepsis is a leading cause of mortality in the ICU, particularly when it progresses to septic shock. Septic shocks are critical medical emergencies, and timely recognition and treatment are crucial for improving survival rates. In the real-world healthcare data experiments on MIMIC-IV dataset, we aim to uncover the delay effect of the treatments related to septic shocks for the whole patient samples.

**Patients** We select 1943 patients that satisfied the following criteria from the dataset: (i) The patients are diagnosed with sepsis [Saria, 2018]. (ii) Patients, if diagnosed with sepsis, the timestamps of any clinical testing and timestamps of medication administration and corresponding dosage were not missing.

**Treatment** Suggested by Komorowski et al. [2018], we extracted 21 treatment associated with sepsis which are consistent with expert consensus. Based on the distinct clinical characteristics of these treatments, they can be categorized into the following three groups, which are shown in Table 10. Vasopressor therapy is a fundamental treatment of septic-shock-induced hypotension as it aims at correcting the vascular tone depression and then improving organ perfusion pressure; Antibiotics also should be given within a few hours of the diagnosis of sepsis; Some auxiliary treatments such as packed red blood cells and invasive ventilation are also necessary in ICU.

**Outcome** We treated real-time urine as the outcome indicator since low urine is the direct indicator of bad circulatory systems and the signal for septic shock. In contrast, normal urine reflects the effect of the drugs and treatments and the improvement of the patients' physical condition. Some treatments will have a rapid effect on the urine while others might take longer to exert an effect.

**Preprocessing** Due to the frequent fluctuations in urine output within the ICU setting, we considered only those instances in which urine output became normal after maintaining an abnormal level for at least 24 hours. These instances were regarded as valid target events that hold significance for prediction and explanation. For each patient, we extracted all the periods that met the criteria. We also documented all the intake time points within the 24 hours leading up to the transition of urine output from abnormal to normal during clinical treatment. The processed data set has 7377 records in total, and we split them by 80%, 10%, and 10% as the training, evaluation, and testing set.

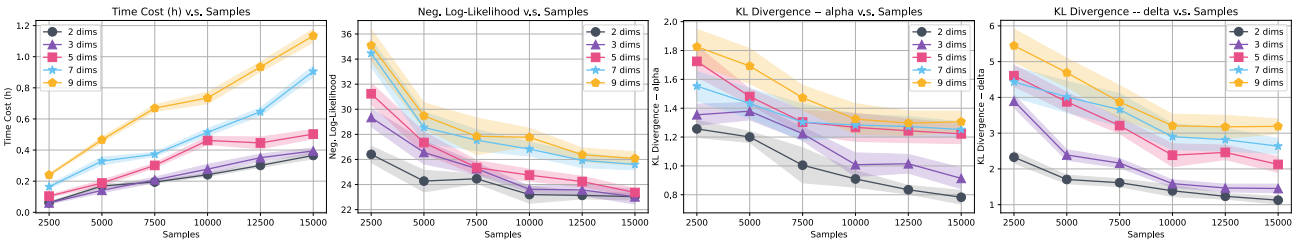


Figure 8: Scalability experiments with varying training samples and dimensions. We take uni-modal distribution datasets as examples. All the experiments are conducted over 5 random runs and the standard errors are reflected in the shaded areas.

Table 7: Compare the prediction performance between our flow-based model and other mixture models using two real-world datasets.

| Metric | MIMIC-IV | | Covid Polcy | |
|---|---|---|---|---|
| | NLL ↓ | RMSE ↓ | NLL ↓ | RMSE ↓ |
| MoU | 28.90 | 4.13 | 43.67 | 4.28 |
| MoG | 24.25 | 3.54 | 39.02 | 3.80 |
| **Ours*** | **21.32** | **2.86** | **36.94** | **3.35** |

Table 8: Compare the accuracy of learned parameter distributions across different models using KL divergence as metric with varying sample sizes for different setting of decay ($\beta$) parameter. Here we focus on multi-modal distributions.

| Synthetic | Shared $\beta$ | | | Varied $\beta$ | | |
|---|---|---|---|---|---|---|
| | 2500 | 7500 | 12500 | 2500 | 7500 | 12500 |
| Hypernet ($\alpha$) | 39.68 | 33.95 | 33.30 | 40.12 | 39.61 | 39.30 |
| $\beta$-VAE ($\alpha$) | 5.12 | 4.42 | 2.94 | 5.60 | 5.29 | 4.72 |
| **Ours*** ($\alpha$) | **4.62** | **3.16** | **2.79** | **4.93** | **3.75** | **3.38** |
| Hypernet ($\beta$) | 39.82 | 29.82 | 27.41 | 65.92 | 62.56 | 59.63 |
| $\beta$-VAE ($\beta$) | 5.94 | 5.52 | 5.17 | 6.30 | 5.89 | 5.22 |
| **Ours*** ($\beta$) | **4.82** | **4.55** | **3.76** | **5.13** | **4.79** | **4.21** |
| Hypernet ($\delta$) | 14.79 | 7.89 | 11.24 | 39.47 | 39.20 | 38.67 |
| $\beta$-VAE ($\delta$) | 2.88 | 2.43 | 2.19 | 3.61 | 3.42 | 3.10 |
| **Ours*** ($\delta$) | **2.92** | **2.25** | **1.86** | **3.18** | **2.75** | **2.34** |

### E.2 HEALTHCARE DATA EXPERIMENTS – COVID POLICY

**Policy Information** The descriptions of the policies of the two countries considered in our experiments (Australia and France) are summarized in Table 11. In Table 12, we tick the policy for these two countries if it appears in the datasets.

**Preprocessing** We aim to investigate the impact of the policies of different categories on the daily average number of confirmed cases. We tallied the cumulative confirmed cases over 7 consecutive days to capture the epidemic spread trend to avoid daily noise. To understand the waiting time for each policy to work, we marked the date when confirmed cases started decreasing as a "dropping infection event". We conducted experiments on data from Australia and France for the years 2021-2022 based on the most severe COVID-19 situations and effective governmental policies. For each dataset of country, we split them by $80\%$, $10\%$, and $10\%$ as the training, evaluation, and testing set.

**Experiment Results** In Figure 9 and Figure 10, we visualize the learned distributions and samples from well-trained models for Australia and France. Overall, the lag for the effectiveness of government policies in Australia seems shorter than in France. The positive impact of containment and closure policies is normally distributed and is almost larger than all other policies in these two countries. In Australia, when government enforces Containment Closure policies, the population generally splits into two groups (exhibiting two peaks in the distribution). One group promptly complies with isolation measures, leading to a decrease in new cases of COVID-19 around 5 days after policy implementation. The other group responds more slowly, requiring approximately 6.5 days for the policies to take effect. The pattern of Containment Closure policies in France is different, roughly following a normal distribution with a mean of 7.5 days. In terms of Healthcare System policies, delay effects in Australia also exhibit a bimodal distribution, with peaks at around 7 and 9 days, whereas in France with longer onset times, approximately 7.5 and close to 11 days, respectively, but the variance is smaller. In both countries, the impact of Vaccination and Economic policies is smaller than that of the two policies mentioned above. In Australia, the time lag for Vaccination to take effect typically peaks at 10 and 11 days, while for Economic policies, the mean time lag is 14.5 days. In France, the mean time lags are approximately 11.5 and 15 days, respectively, for these two policies.

### E.3 OTHER REAL-WORLD DATASET EXPERIMENTS

we further extended our evaluation to additional datasets (StackOverflow, Taobao, and Taxi) by predicting both the next event type and its time, while incorporating per-event Negative Log-Likelihood (NLL) (lower is better) and event type prediction error rate (lower is better) as extra evaluation metrics, as well as RMSE. As shown in Table 13 below, our model maintains strong performance across these baselines on almost all datasets, except for Taxi dataset. This also suggests that StackOverlow and Taobao datasets may inherently exhibit delayed event-triggering effects (while Taxi dataset may not), further validating our approach.

Table 9: Prediction tasks on synthetic datasets for varied and shared decay parameter. Here we focus on multi-modal distributions.

| Synthetic Datasets | | Shared $\beta$ | | Varied $\beta$ | |
|---|---|---|---|---|---|
| Category | Method | NLL ↓ | RMSE ↓ | NLL ↓ | RMSE ↓ |
| Non-Param. | GM-NLF | 34.27 | 2.72 | 34.47 | 3.15 |
| | MMEL | 34.55 | 2.85 | 34.98 | 3.27 |
| | Gibbs-Hawkes | 33.86 | 2.64 | 34.50 | 3.04 |
| Param. | RMTPP | 32.67 | 2.77 | 34.19 | 2.90 |
| | THP | 32.10 | 2.46 | 33.80 | 2.86 |
| | PromptTPP | 32.52 | 2.40 | 33.67 | 2.79 |
| | HYPRO | – | 2.37 | – | 2.83 |
| | MLE-SGL | 33.78 | 2.57 | 34.52 | 3.22 |
| | GC-CGD | 33.54 | 2.45 | 34.73 | 3.05 |
| | **Ours*** | **30.42** | **2.25** | **32.25** | **2.68** |

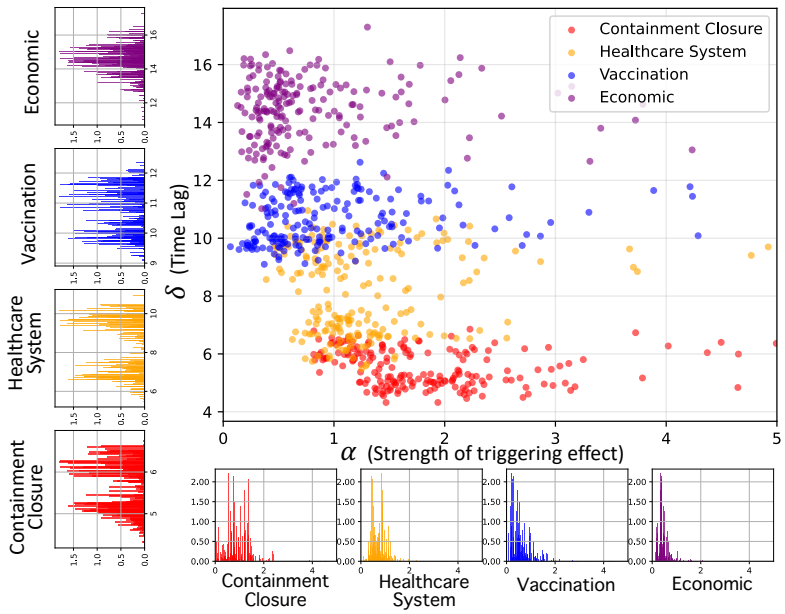


Figure 9: Learned impact $\alpha$ and delay effect $\delta$ distributions for Covid Policy dataset in Australia (**left and bottom bar plots**) and 2D scatter plot for samples of these two parameters (**top-right**).

## F LIMITATIONS AND BROADER IMPACTS

We assume that $\beta$ is shared across all event types in our paper. In fact, estimating the decay parameter of the Hawkes process is inherently challenging; only a limited number of studies [Santos et al., 2023] addressing this task, demonstrate that the estimation difficulties relate to the noisy, non-convex shape of the log-likelihood of Hawkes process as a function of the decay. Yet, our proposed model can easily extend to the estimation of the decay parameter distributions, but stability needs to be improved. We can also attempt to perform our method on more forms of triggering kernels for Hawkes process to validate the performance, such as Gamma kernel, Weibull-based kernel, power-law kernel, and so on.

Our proposed model can infer the time-lag distributions which are of scientific meaning and help trace the original causal time that supports the root cause analysis. In healthcare, inferring the distributions of time lags and other parameters that affect drug efficacy can assist clinicians in identifying the timing of drug effects. This information enables them to develop more effective treatment strategies for patients. In a pandemic, our model can help decision-makers and citizens understand governmental responses consistently, aiding efforts to fight the pandemic. However, this requires our algorithm to provide high accuracy. In clinical applications, our method can serve as a reference for inexperienced novice doctors, providing them with valuable guidance.

Table 10: Description of the treatment extracted from MIMIC-IV dataset.

| Category | Treatment |
|---|---|
| **Vasoconstrictor** | Epinephrine |
| | Phenylephrine |
| | Norepinephrine |
| | Dobutamine |
| | Dopamine |
| | Vasopressin |
| | Angiotensin II (Giapreza) |
| **Antibiotic** | Vancomycin |
| | Caspofungin |
| | Cefepime |
| | Ceftriaxone |
| | Gentamicin |
| | Micafungin |
| | Tobramycin |
| | Piperacillin/Tazobactam |
| **Auxiliary Treatment** | Furosemide (Lasix) |
| | Heparin Sodium |
| | Invasive Ventilation |
| | Packed Red Blood Cells |
| | IV Immune Globulin (IVIG) |
| | Acetaminophen-IV |

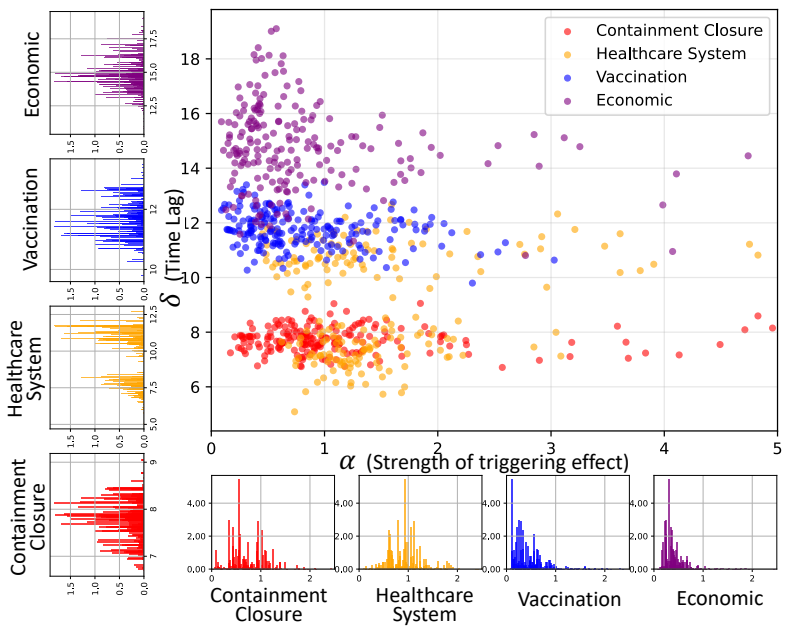


Figure 10: Learned impact $\alpha$ and delay effect $\delta$ distributions for Covid Policy dataset in France (**left and bottom bar plots**) and 2D scatter plot for samples of these two parameters (**top-right**).

Table 11: Policies description of each code.

| Category | Code | Explain |
|---|---|---|
| **Containment & closure policies** | C1 | School closing. |
| | C2 | Workplace closing. |
| | C3 | Cancel public events. |
| | C4 | Restrictions on gatherings. |
| | C5 | Close public transport. |
| | C6 | Stay at home requirements. |
| | C7 | Restrictions on internal movement. |
| | C8 | International travel controls. |
| **Health system policies** | H1 | Public information campaigns. |
| | H2 | Testing policy. |
| | H3 | Contact tracing. |
| | H4 | Emergency investment in healthcare. |
| **Vaccination policies** | V1 | Vaccine prioritisation. |
| | V2 | Vaccine eligibility/availability. |
| | V3 | Vaccine financial support. |
| | V4 | Mandatory Vaccination. |
| **Economic policies** | E1 | Income support. |
| | E2 | Debt/contract relief. |
| | E3 | Fiscal measures. |
| | E4 | International support. |

Table 12: The implemented policies for Australia (AUS) and France in 2021-2022.

| **Nations** | **Policies** | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Containment & closure | | | | | | | | Health system | | | | Vaccination | | | | Economic | | | |
| | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | H1 | H2 | H3 | H4 | V1 | V2 | V3 | V4 | E1 | E2 | E3 | E4 |
| **AUS** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | | |
| **France** | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | |

Table 13: Experiments on three newly introduced real-world dataset. Note that we report per-event negative log-likelihood (NLL ↓), event type prediction error rate (ER% ↓), and event time prediction root mean square error (RMSE ↓) for these new real-world dataset.

| Real-World Datasets | | StackOverflow | | | Taobao | | | Taxi | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Category | Method | NLL ↓ | ER % ↓ | RMSE ↓ | NLL ↓ | ER % ↓ | RMSE ↓ | NLL ↓ | ER % ↓ | RMSE ↓ |
| Non-Param. | GM-NLF | 2.88 | 59.20 | 1.39 | 1.68 | 57.81 | 0.84 | 0.50 | 23.62 | 0.68 |
| | MMEL | 2.95 | 58.74 | 1.66 | 1.52 | 59.23 | 0.82 | 0.55 | 19.92 | 0.70 |
| | Gibbs-Hawkes | 2.90 | 58.81 | 1.62 | 1.40 | 60.08 | 0.84 | 0.53 | 21.14 | 0.73 |
| Param. | RMTPP | 2.83 | 56.85 | 1.38 | 1.64 | 57.20 | 0.76 | 0.35 | 16.43 | 0.53 |
| | THP | 2.68 | 52.73 | 1.38 | 1.22 | 53.38 | 0.73 | 0.48 | 13.28 | 0.46 |
| | PromptTPP | 2.71 | 51.53 | 1.37 | 1.25 | 54.26 | 0.67 | 0.44 | **13.15** | **0.43** |
| | HYPRO | – | 51.70 | 1.35 | – | 52.37 | 0.69 | – | 13.26 | 0.47 |
| | MLE-SGL | 3.12 | 58.34 | 1.43 | 1.26 | 58.40 | 0.79 | 0.38 | 17.95 | 0.68 |
| | GC-CGD | 3.04 | 57.36 | 1.40 | 1.38 | 55.89 | 0.73 | **0.32** | 17.22 | 0.64 |
| | **Ours*** | **2.64** | **51.22** | **1.33** | **1.18** | **52.06** | **0.64** | 0.52 | 16.08 | 0.56 |