Hyper Hawkes Processes: Interpretable Models of Marked Temporal Point Processes

Alex James Boyd **Andrew Warrington**
Taha Kass-Hout **Parminder Bhatia** **Cao Xiao**
AI Innovation Lab, GE HealthCare
{alex.boyd,andrew.warrington}@gehealthcare.com

Abstract

Foundational marked temporal point process (MTPP) models, such as the Hawkes process, often use inexpressive model families in order to offer interpretable parameterizations of event data. On the other hand, neural MTPP models often trade interpretability for predictive performance. In this work, we present a new family of MTPP models: the *hyper Hawkes process* (HHP), which aims to be as flexible and performant as neural MTPPs, while retaining interpretable aspects. To achieve this, the HHP extends the classical Hawkes process to increase its expressivity by first expanding the dimension of the process into a latent space, and then introducing a hypernetwork to allow time- and data-dependent dynamics. These extensions define an MTPP family that is competitive with modern neural models across a range of benchmark tasks and metrics. Furthermore, by retaining the linearity of the recurrence, albeit now piecewise and conditionally linear, the HHP also retains much of the structure of the original Hawkes process, which we exploit to create direct probes into *how* the model creates predictions. HHP models therefore offer competitive predictions alongside an opportunity to "open the box" and inspect how predictions were generated.

1 Introduction

In modern machine learning, the pursuit of predictive accuracy often comes at the expense of interpretability. This trade-off is especially pronounced in marked temporal point processes (MTPPs): classical models such as the Hawkes process [Hawkes, 1971] offer transparent parameters but can underfit real-world data; while neural models such as the neural and transformer Hawkes processes [Mei and Eisner, 2017, Zuo et al., 2020] achieve excellent performance but do not generally expose explicit contributions from past events. General interpretability methods for neural networks [Räuker et al., 2023, Chefer et al., 2021, Rai et al., 2024, Maheswaranathan and Sussillo, 2020] can be indirect or ambiguous, motivating architectures with tractable event-level probes.

We introduce the *hyper Hawkes process* (HHP), a new family of intensity-based MTPP models designed to address this trade-off (Figure 1). The HHP extends the classical Hawkes process by (i) lifting recurrent dynamics into a latent space, decoupling them from mark dimensionality, and (ii) using a history-dependent hypernetwork [Ha et al., 2017] to adapt dynamics over time. These extensions enhance expressivity while preserving a conditionally linear recurrence. We exploit this for particle-removal probes inspired by linear regression diagnostics (e.g., DFBETA and DFFIT [Belsley et al., 2005]). On real-world benchmarks, HHP is competitive with modern neural MTPP models, while *also* providing an exact decomposition into event-level particles under the observed transitions.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: Six-metric ranking of the nine main models. Each metric averages per-dataset ranks of the reported five-seed means over the seven datasets; ties at the displayed precision receive midranks. The composite averages the six metric ranks. Lower is better. Bold and underlined entries indicate first and second under this convention. Full results appear in Tables 2, 5 and 6.

		Time Metrics			Mark Metrics			**Composite**
Model		Likelihood	RMSE	PCE	Likelihood	Accuracy	ECE	**Rank**
RMTPP	[Du et al., 2016]	7.57	5.64	6.14	7.57	7.50	6.86	6.88
NHP	[Mei and Eisner, 2017]	4.43	3.50	5.86	3.57	<u>2.50</u>	5.71	4.26
SAHP	[Zhang et al., 2020]	6.29	5.50	5.86	8.29	8.00	8.00	6.99
THP	[Zuo et al., 2020]	8.29	4.93	5.71	6.71	6.86	6.00	6.42
IFTPP	[Shchur et al., 2020]	3.00	6.00	**2.43**	5.29	5.64	**2.00**	4.06
AttNHP	[Yang et al., 2022]	4.43	7.57	4.86	3.57	3.93	4.07	4.74
S2P2	[Chang et al., 2025]	**1.71**	<u>3.21</u>	<u>3.71</u>	<u>2.43</u>	<u>2.50</u>	<u>3.79</u>	<u>2.89</u>
iTHP	[Meng et al., 2024]	6.71	6.29	6.29	5.57	6.14	4.50	5.92
HHP	**(Ours)**	<u>2.57</u>	**2.36**	4.14	**2.00**	**1.93**	4.07	**2.85**

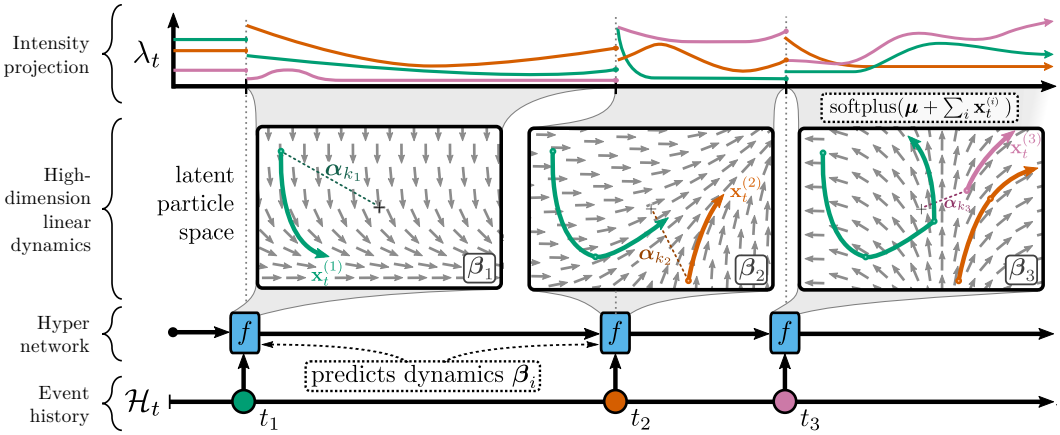


Figure 1: High-level schematic of the proposed *hyper Hawkes process* (HHP). The bottom row shows the sequence of marked events, which are input into a hypernetwork with transition function f. The hypernetwork outputs the dynamics of a generalized linear Hawkes process, the flow fields for which are shown. These dynamics are then projected into intensity space. This can be done efficiently in a forward pass if intensity computation is the sole objective. However, careful design of the HHP allows the latent state to be decomposed into individual additive *particles*, each attributed to a single event and shown as colored lines in the flow field. Their contributions to the pre-softplus readout are additive, while the final intensities need not be. This allows us to create a highly expressive and efficient model, while making each event's particle contribution directly tractable under the dynamics generated for the observed history.

Our main contributions are:

- We propose the *hyper Hawkes process* (HHP), combining interpretable structure with a flexible neural architecture.
- We demonstrate that HHP is competitive with modern neural MTPP models on real-world benchmarks.
- We develop particle-level probes exploiting HHP's structure, enabling direct insight into model mechanics, and explore these on synthetic tasks.

Paper Outline. Section 2 reviews MTPPs; Section 3 introduces HHP; and Section 5 evaluates predictive performance. Section 6 develops and demonstrates particle-level probes, and Section 7 discusses limitations and future directions.

2 Temporal Point Processes Background

2.1 Marked Temporal Point Processes

In this paper we consider a *marked temporal point process* as a discrete and finite sequence of time-mark pairs, $\mathcal{H}_t := \{(t_i, k_i) \mid t_i \leq t \text{ for } i \in \mathbb{N}\}$, where $t_i \in \mathbb{R}$, $t_{i-1} < t_i \forall i$, $k_i \in \mathcal{M}$, $\mathcal{M} := \{1, \ldots, K\}$, and $\mathcal{H}_t$ is referred to as an *event history*. We also define $\mathcal{H}_{t-}$ similarly to $\mathcal{H}_t$, except excluding events at *exactly* time t. Although we do not consider it here, note that $\mathcal{M}$ can be a more general space, such as countable or continuous.

One method for parameterizing an MTPP is using a *marked intensity process*. The intensity $\boldsymbol{\lambda}_t := [\lambda_t^1, \ldots, \lambda_t^K]^\top \in \mathbb{R}^K_{\geq 0}$ defines the rate of occurrence of events:

$$\lambda_t^k \mathrm{d}t := \mathbb{E}\left[k\text{-type event occurs in } [t, t + \mathrm{d}t] \mid \mathcal{H}_{t-}\right]. \tag{1}$$

The *total intensity* is then the rate that *any* event occurs, $\lambda_t := \sum_{k=1}^K \lambda_t^k$. It can then be shown that the log-likelihood for a sequence $\mathcal{H}_T$ is defined as [Daley and Vere-Jones, 2003, ch. 7.3]:

$$\mathcal{L}(\mathcal{H}_T) := \sum\nolimits_{i=1}^{N_T} \log \lambda_{t_i}^{k_i} - \int_0^T \lambda_s \mathrm{d}s. \tag{2}$$

To train an MTPP, we can optimize $\mathcal{L}(\mathcal{H}_T)$ over observed data (see, e.g., Mei and Eisner [2017]).

2.2 Hawkes Processes

The *Hawkes process* [Hawkes, 1971] is a widely studied temporal point process in which the occurrence of events can encourage further occurrences soon thereafter, a property often referred to as *self-excitation*. This family of processes is characterized by an intensity that takes the form:

$$\lambda_t = \sigma\left(\int_0^{t-} \phi(t - s)\mathrm{d}N_s\right) \equiv \sigma\left(\textstyle\sum_{i=1}^{N_{t-}} \phi(t - t_i)\right), \tag{3}$$

where $\sigma : \mathbb{R} \to \mathbb{R}_+$ ensures the intensity is non-negative, and excitation function $\phi : \mathbb{R}_+ \to \mathbb{R}$ encodes the influence prior events have on the rate of occurrence for future events.

The *linear Hawkes process* (LHP) is a common variant, employing an exponential kernel for the excitation function yielding: $\lambda_t = \mu + \sum_{i=1}^{N_{t-}} \alpha \exp(-\beta(t - t_i))$, where $\mu, \alpha, \beta \in \mathbb{R}_+$. The exponential kernel in particular makes the process Markovian, allowing us to describe the intensity function in the differential form of $\lambda_t = \mu + x_{t-}$ where $\mathrm{d}x_t = -\beta x_t + \alpha \mathrm{d}N_t$. This can easily be extended to accommodate K categorical marks by generalizing the impulse and rate parameters into positive matrices $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}_+^{K \times K}$ leading to the following marked intensity process:

$$\mathrm{d}\mathbf{x}_t = -\boldsymbol{\beta}\mathbf{x}_{t-}\mathrm{d}t + \boldsymbol{\alpha}\mathrm{d}\mathbf{N}_t, \tag{4}$$

$$\boldsymbol{\lambda}_t = \boldsymbol{\mu} + \mathbf{x}_{t-}, \tag{5}$$

where $\boldsymbol{\mu} \in \mathbb{R}_+^K$ is the vector of background intensities for different marks, $\mathrm{d}\mathbf{N}_t \in \{\mathbf{0}, \mathbf{e}_1, \ldots, \mathbf{e}_K\}$ is $\mathbf{0}$ when no event occurs and a one-hot vector corresponding to the mark k that occurs at time t, and $\mathbf{x}_0 = \mathbf{0}$. From this form, we can compute the left limit of the intensity at the next event, given a right limit $\mathbf{x}_{t_i}$ for time t_i, in closed form as: $\mathbf{x}_{t_{i+1}-} = e^{-\boldsymbol{\beta}(t_{i+1} - t_i)}\mathbf{x}_{t_i+}$, where $t_i < t_{i+1}$ and no event occurred between t_i and t_{i+1}. Hawkes processes such as this are widely used in statistical inference settings, due to $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ being directly interpretable by describing marginal event-to-event effects.

3 Hyper Hawkes Processes

We now introduce the core of our hyper Hawkes process (HHP). We start by identifying the mechanism that allows classical Hawkes processes to identify the effect a given event has on future intensities. We then extend the Hawkes process to be more flexible and performant while retaining a conditional decomposition into event-level particles. Lastly, we briefly discuss how the HHP is implemented in practice. Note we defer discussion on exactly *how* to interpret the model to Section 6.

3.1 Event-level Effects in Hawkes Processes

As shown in Equation (3), classical Hawkes processes are formally defined by their rectification function σ and their excitation function ϕ. While ϕ is often parameterized to isolate marginal direct effects of events of one type to another (e.g., through $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ for LHPs), they actually also provide event-level attribution. In particular, $\phi(t - t_i)$ directly encodes how much the i^{th} event is exciting or inhibiting an event at time t. $\phi(t - t_i)$ could even be generalized to conditional $\phi_i(t; \mathcal{H}_t)$ and this effect would still be identifiable if the final intensity is a monotonic transform of a linear combination of these effects: $\lambda_t = \sigma(\sum_{i=1}^{N_{t-}} \phi_i(t; \mathcal{H}_{t-}))$. For a history-conditioned kernel, this identifies the event's contribution under the observed history; deleting that event may also change later kernels. Note that the LHP maintains this property through the linearity of the recurrence relation, as for $t \in (t_i, t_{i+1}]$ it holds that:

$$\mathbf{x}_{t-} = e^{-\boldsymbol{\beta}(t-t_i)}\mathbf{x}_{t_i+} = \sum_{j=1}^{i} e^{-\boldsymbol{\beta}(t-t_j)}\boldsymbol{\alpha}\mathbf{e}_{k_j}.$$

3.2 Generalizing Hawkes Processes

While the LHP is widely used because of its interpretability, the model itself is not very expressive: the parameterization only allows excitation, the recurrence is directly coupled to the intensity, and the dynamics are time-invariant. We generalize the LHP by tackling these limits while ensuring that the recurrence remains linear conditional on the generated dynamics.

Increasing the Latent Dimension. The LHP has limited expressivity because each recurrent dimension ties one-to-one with an output intensity, i.e., changing dimension k in $\mathbf{x}_t$ only affects λ_t^k (akin to using an RNN's hidden state directly as output ($\hat{\mathbf{y}}_i := \mathbf{h}_i$) rather than $\hat{\mathbf{y}}_i := \sigma(\mathbf{W}\mathbf{h}_i + \mathbf{b})$). Additionally, $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ are parameterized to be strictly positive, thus only allowing for excitation and not inhibition. We address both of these issues by lifting the dynamics into a latent space:

$$\mathrm{d}\mathbf{x}_t = -\boldsymbol{\beta}\mathbf{x}_{t-}\mathrm{d}t + \boldsymbol{\alpha}\mathrm{d}\mathbf{N}_t, \tag{6}$$

$$\boldsymbol{\lambda}_t = \sigma(\boldsymbol{\mu} + \mathbf{W}\mathbf{x}_{t-}), \tag{7}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{d\times d}$, $\boldsymbol{\alpha} \in \mathbb{R}^{d\times K}$, $\mathbf{W} \in \mathbb{R}^{K\times d}$, $\boldsymbol{\mu} \in \mathbb{R}^K$, and $\sigma(z) = \log(1+e^z)$ acts elementwise. This decouples the recurrent dimension d from the number of marks K, while σ ensures inhibitory effects still result in a nonnegative intensity. This increases expressivity when $d > K$, or prevents $\mathcal{O}(K^2)$ parameter scaling when K is large by setting $d < K$. While the latent state is no longer directly in mark space, the fixed-dynamics pre-softplus contribution of event i is $\mathbf{W}\exp(-\boldsymbol{\beta}(t - t_i))\boldsymbol{\alpha}\mathbf{e}_{k_i}$.

Adding Time-Variation with Hypernetworks. Even in this expanded space, the model has limited expressivity because of the linearity of the recurrence relation and time-invariant dynamics. In general, RNNs (such as GRUs) do not suffer from this problem due to the nonlinearity applied when propagating the hidden state. Unfortunately, this is undesirable as it removes identifiable contributions, complicating the model interpretability (see the extensive literature on interpreting non-linear RNNs in even simple tasks, e.g. Miconi [2017], Maheswaranathan and Sussillo [2020]).

An alternative is to retain the conditional linearity of the recurrence but allow its generator $\mathbf{A}_t$ to vary with history. We could trivially make the dynamics mark-specific, only depending on the most recent event, but subsequent events will critically suppress the enduring influence of all preceding events. Therefore we instead introduce a *hypernetwork* [Ha et al., 2017], f_θ, that predicts dynamics as a function of the history. The resulting recurrence is:

$$\mathrm{d}\mathbf{x}_t = \mathbf{A}_t\mathbf{x}_{t-}\mathrm{d}t + \boldsymbol{\alpha}\mathrm{d}\mathbf{N}_t, \tag{8}$$

$$\mathbf{A}_t = f_\theta(\mathcal{H}_{t-}), \tag{9}$$

$$\boldsymbol{\lambda}_t = \sigma(\boldsymbol{\mu} + \mathbf{W}\mathbf{x}_{t-}). \tag{10}$$

We experimented with predicting history-dependent $\boldsymbol{\alpha}$, however, we found it only had a marginal impact on performance, and greatly detracts from the interpretability arguments presented in Section 6. We therefore do not create history-dependent impulses, and only utilize fixed impulses. Even with static impulses, Equation (8) now defines highly expressive non-linear latent dynamics as a function of individual event sequences. We use a standard multi-layer GRU as the hypernetwork throughout. This choice reflects the GRU's widely available, optimized implementation, linear work complexity, and good predictive performance. The "HHP" architecture is not fundamentally tied to this choice, however, and exploration of alternative hypernetwork architectures is interesting future work.

Efficient and Expressive Parameterization. The final component is selecting a parameterization for how f_θ generates $\mathbf{A}_t$ that is both efficient and expressive. For efficiency, we make $\mathbf{A}_t$ piecewise constant between events and use a diagonal spectral parameterization, $\mathbf{A}_i = \mathbf{V}_i \mathbf{D}_i \mathbf{V}_i^*$ on (t_i, t_{i+1}). Here $\mathbf{D}_i$ is diagonal, $\Re D_{i,aa} < 0$, and $\mathbf{V}_i$ is unitary [Jing et al., 2017]. The hypernetwork state after event i emits $\mathbf{V}_i$ and $\mathbf{D}_i$ for the following interval. Eigenvalues with negative real parts yield interval-wise decay, while unitarity avoids an ill-conditioned inverse. These restrictions exclude locally growing modes and non-normal transient amplification within an interval; changing bases across intervals can still produce richer history-dependent behavior. This leads to the final HHP update equation:

$$\mathbf{x}_{t_{i+1}-} = \mathbf{V}_i e^{\mathbf{D}_i \Delta_i} \mathbf{V}_i^* \mathbf{x}_{t_i+}, \quad \mathbf{x}_{t_{i+1}+} = \mathbf{x}_{t_{i+1}-} + \boldsymbol{\alpha} \mathbf{e}_{k_{i+1}}, \tag{11}$$

$$\boldsymbol{\lambda}_{t_{i+1}-} = \sigma(\boldsymbol{\mu} + \mathbf{W} \mathbf{x}_{t_{i+1}-}), \tag{12}$$

where $\Delta_i = t_{i+1} - t_i$; the exponential acts elementwise on the diagonal. For intermediate $t \in (t_i, t_{i+1})$, replace Δ_i by $t - t_i$. The optional complex-state implementation takes the real readout before softplus (see Section B). Extensive details on this and the implementation of the architecture can be found in Section B.

Allowing $\mathbf{V}_i$ to change after an event can yield transition matrices that are not simultaneously diagonalizable, unlike a fixed-basis model. This adds a mechanism for history-dependent state transport [Merrill et al., 2024]. It does not by itself establish a universal expressivity or state-tracking guarantee for the fitted HHP.

Computational considerations. For fixed dimensions and N events, the sequential GRU and aggregate-state recurrence use $\mathcal{O}(N)$ work and have an $\mathcal{O}(N)$ critical path. With M integration samples per interval, intensity integration adds $\mathcal{O}(NM)$ work; samples within an interval can be evaluated in parallel. Standard prediction propagates one aggregate state and does not materialize source particles.

Summary. We briefly summarize the HHP model we have proposed. A (nonlinear) hypernetwork consumes the event history, and outputs piecewise constant dynamics parameters for a high-dimensional linear recurrence with learned impulses at events. We parameterize the dynamics in a per-event eigenbasis also predicted by the hypernetwork. This allows for efficient closed-form computation of updates to the latent state in continuous time to time points of interest. We then decode the latent state by projecting it into the intensity space and applying a rectification function to ensure intensities are non-negative. This allows us to define a highly flexible intensity-based neural MTPP that we can efficiently evaluate at any time point, but that has a (conditionally) linear recurrence which will serve as a "bottleneck" that we can inspect, as we explore in Section 6.

Structural distinction from prior neural MTPPs. While HHP maintains a fixed-dimensional Markov state comprising its Hawkes and hypernetwork states, its defining property is a conditionally linear continuous-time recurrence whose solution admits an exact decomposition into event-level particles in the latent state and pre-softplus readout. NHPs use nonlinear state transitions, while THPs compute intensities via attention over the history. The HHP decomposition therefore exposes one exact pathway through its recurrence; output intensities after softplus are not additive so it is not afforded the same full interpretability as a LHP.

Overall Model. The complete architecture, including the hypernetwork recurrence, the latent Hawkes recurrence, and the intensity readout, is illustrated in Figure 3 and detailed in Section B.

4 Related Works

Neural MTPPs. Hawkes processes [Hawkes, 1971, Liniger, 2009] offer transparent parameterizations of the intensity function, but suffer from limited expressivity. Neural MTPPs improve flexibility using recurrent models [Du et al., 2016, Mei and Eisner, 2017], CNNs [Zhuzhel et al., 2023], transformers [Zhang et al., 2020, Zuo et al., 2020, Yang et al., 2022], and deep SSMs [Gao et al., 2024, Chang et al., 2025]. These approaches do not generally expose per-event conditional decompositions like HHP.

Interpretable MTPPs. Interpretability in MTPPs has traditionally been associated with Hawkes processes, where additive structure admits direct event-level effects [Hawkes, 1971]. Most neural MTPPs prioritize predictive performance, with interpretability typically provided post hoc (e.g., attention weights). Architectural constraints have been proposed to recover partial transparency, including

Interpretable Transformer Hawkes Processes (iTHP), which constrain attention to expose pair-wise event-type interactions while evaluating intensity throughout inter-event intervals [Meng et al., 2024] and event-type–parameterized neural ODEs [Song et al., 2024]. Rule-based approaches [Li et al., 2022, Yang et al., 2024, Li et al., 2020] provide explicit structure but do not scale gracefully. In contrast, HHP's conditionally linear latent recurrence yields an exact additive decomposition of its state and pre-softplus readout under the realized transitions.

Deep State Space Models and State Tracking. Connections between linear Hawkes processes and state space models (SSMs) have been established [Chang et al., 2025, Gu et al., 2022, Smith et al., 2022, Gu and Dao, 2023]. The S2P2 model [Chang et al., 2025] increases expressivity via deep stacks of LHP-like layers with fixed eigenvectors. In contrast, we use a *single* latent Hawkes process with both time-varying eigenvalues $\mathbf{D}$ and eigenvectors $\mathbf{V}$, a design motivated by results showing that non-simultaneously diagonalizable linear dynamics can support richer dynamics, such as state tracking [Merrill et al., 2024].

5 Experiments

We evaluate the HHP on common TPP benchmarks, finding that our approach matches modern neural MTPPs on predictive performance, even before consideration of interpretability. Please see Section C for experimental configurations and evaluation details.

Datasets. We evaluate baseline models and our HHP on seven widely used, real-world MTPP datasets, all distributed through the `EasyTPP` package. We provide more details in Section C.

Evaluation Metrics. We evaluate the per-event log-likelihood as our primary measure of performance. This is both what the models are trained to optimize and is a proper scoring metric (also known as a proper scoring rule [Heinrich-Mertsching et al., 2024]). We also compute the RMSE of the next event time prediction and the average accuracy of the next mark type prediction. Finally, we also evaluate *calibration*, which provides a measure of how well the model quantifies the uncertainty in its predictions [Bosser and Taieb, 2023], see Section C. In Table 1, following [Chang et al., 2025], we also provide a "composite metric", averaging ranks across six metrics and the seven datasets.

Ablations. As mentioned previously, the HHP generalizes the LHP in key ways. To assess each extension, we also measure the performance of three different ablations: (i) $\text{HHP}_{\neg\text{Stateful}}$, which disables the "statefulness" by setting the eigenvectors to a learned constant basis[1], $\mathbf{V}_i = \mathbf{V}$; (ii) $\text{HHP}_{\neg\text{Hyper}}$, which disables the hypernetwork, $\mathbf{A}_i = \mathbf{A}$; and (iii) $\text{HHP}_{\neg\text{Latent}}$, which both disables the hypernetwork *and* removes the latent space, setting $d = K$ and $\mathbf{W} = \mathbf{I}$.

Results. Results are presented in Table 2; likelihood components and calibration appear in Tables 5 and 6. HHP has higher total log-likelihood and mark accuracy and lower time RMSE than iTHP on every dataset; calibration is mixed. HHP and S2P2 have similar composite ranks (2.85 and 2.89, respectively) when calculated from the rounded reported means with midranks for ties.

Interestingly, we see that while the statefulness of the HHP does improve performance, even without it, the model $\text{HHP}_{\neg\text{Stateful}}$ is competitive with baselines. Even with static dynamics, $\text{HHP}_{\neg\text{Hyper}}$ outperforms several baselines. This suggests that a major bottleneck in the conventional Hawkes process was the tying of latent dimensions to the mark-space, as well as that the basic form of the Hawkes process provides a strong inductive bias for MTPPs.

6 On Interpretability

As discussed in Sections 2 and 3, a key feature of both the LHP and our proposed HHP is the linear recurrence structure. This structure enables us to view the model equivalently as a recurrence or through a particle perspective, where each event adds a distinct, trackable impulse to the latent state. We can use these particles to probe model outputs under the transitions generated for the observed history. We now introduce practical tools that exploit this structure and demonstrate their utility on synthetic data.

[1] Since the impulse matrix $\boldsymbol{\alpha}$ and readout matrix $\mathbf{W}$ are freely learned, a fixed basis can be absorbed into them as $\boldsymbol{\alpha}' = \mathbf{V}^*\boldsymbol{\alpha}$ and $\mathbf{W}' = \mathbf{W}\mathbf{V}$. Thus we can set $\mathbf{V}_i = \mathbf{I}$ and use a diagonal transition generator without changing the model family.

Table 2: Quantitative results for TPP models across datasets. Main-model entries report means and (standard deviations) across five randomly initialized models; ablation entries report point estimates. Aggregate ranks average each model's per-dataset ranks over the seven datasets, with midranks for ties at the displayed precision.

(a) Per event log-likelihood. Higher log-likelihood values indicate better performance.

Model		Per Event Log-Likelihood, $\mathcal{L}_{\text{Total}}$ (nats) (↑)							Average
		Amazon	Retweet	Taxi	Taobao	StackOverflow	Last.fm	MIMIC-II	Rank* (↓)
RMTPP	[Du et al., 2016]	-2.136 (0.003)	-7.098 (0.217)	0.346 (0.002)	1.003 (0.004)	-2.480 (0.019)	-1.780 (0.005)	-0.472 (0.026)	7.86
NHP	[Mei and Eisner, 2017]	0.129 (0.012)	-6.348 (0.000)	0.514 (0.004)	1.157 (0.004)	-2.241 (0.002)	-0.574 (0.011)	0.060 (0.017)	4.14
SAHP	[Zhang et al., 2020]	-2.074 (0.029)	-6.708 (0.029)	0.298 (0.057)	1.168 (0.028)	-2.341 (0.058)	-1.646 (0.083)	-0.677 (0.072)	7.14
THP	[Zuo et al., 2020]	-2.096 (0.002)	-6.659 (0.007)	0.372 (0.002)	0.790 (0.002)	-2.338 (0.014)	-1.712 (0.011)	-0.577 (0.011)	7.29
IFTPP	[Shchur et al., 2020]	0.496 (0.002)	-10.344 (0.016)	0.453 (0.002)	1.318 (0.017)	-2.233 (0.009)	-0.492 (0.017)	0.317 (0.052)	3.71
AttNHP	[Yang et al., 2022]	0.484 (0.077)	-6.499 (0.028)	0.493 (0.009)	1.259 (0.022)	-2.194 (0.016)	-0.592 (0.051)	-0.170 (0.077)	4.14
S2P2	[Chang et al., 2025]	0.781 (0.011)	-6.365 (0.003)	0.522 (0.004)	1.304 (0.039)	-2.163 (0.009)	-0.557 (0.046)	0.919 (0.069)	**1.71**
iTHP	[Meng et al., 2024]	-1.481 (0.008)	-6.4638 (0.0015)	0.285 (0.004)	0.841 (0.004)	-2.288 (0.002)	-2.085 (0.005)	0.194 (0.029)	6.57
HHP	**(Ours)**	0.616 (0.016)	-6.366 (0.007)	0.520 (0.003)	1.232 (0.014)	-2.209 (0.006)	-0.515 (0.006)	1.314 (0.048)	<u>2.43</u>
$\text{HHP}_{\neg\text{Stateful}}$		0.606	-6.370	0.508	1.249	-2.195	-0.572	1.114	3.43
$\text{HHP}_{\neg\text{Hyper}}$	(Ablations)	0.514	-6.796	0.469	1.224	-2.246	-1.028	0.305	5.29
$\text{HHP}_{\neg\text{Latent}}$		-0.170	-6.880	0.237	1.150	-2.374	-1.390	-0.533	7.71

(b) Prediction RMSE of the next event time prediction. Lower RMSE values indicate better performance.

Model		RMSE, time units (↓)							Average
		Amazon	Retweet	Taxi	Taobao	StackOverflow	Last.fm	MIMIC-II	Rank* (↓)
RMTPP	[Du et al., 2016]	0.338 (0.000)	16488 (070.5)	0.283 (0.001)	0.126 (0.000)	1.049 (0.003)	15.873 (0.000)	0.749 (0.010)	5.64
NHP	[Mei and Eisner, 2017]	0.339 (0.000)	15911 (004.0)	0.282 (0.001)	0.126 (0.000)	1.019 (0.001)	15.733 (0.008)	0.726 (0.001)	3.50
SAHP	[Zhang et al., 2020]	0.335 (0.001)	16102 (062.4)	0.290 (0.008)	0.126 (0.000)	1.031 (0.011)	15.757 (0.007)	1.142 (0.198)	5.50
THP	[Zuo et al., 2020]	0.332 (0.000)	16268 (018.7)	0.285 (0.001)	0.125 (0.000)	1.033 (0.005)	15.871 (0.000)	0.768 (0.005)	4.93
IFTPP	[Shchur et al., 2020]	0.327 (0.000)	16625 (000.2)	0.362 (0.178)	0.125 (0.000)	1.340 (0.724)	16.508 (0.555)	0.767 (0.029)	6.00
AttNHP	[Yang et al., 2022]	2.656 (1.950)	16171 (284.2)	1.739 (0.422)	0.130 (0.000)	1.256 (0.030)	15.865 (0.017)	0.860 (0.022)	7.57
S2P2	[Chang et al., 2025]	0.327 (0.000)	15987 (013.7)	0.281 (0.000)	0.126 (0.000)	1.014 (0.001)	15.720 (0.000)	0.894 (0.054)	<u>3.21</u>
iTHP	[Meng et al., 2024]	0.348 (0.001)	15823 (005.3)	0.292 (0.002)	0.1279 (0.0004)	1.029 (0.002)	16.821 (0.079)	0.781 (0.028)	6.29
HHP	**(Ours)**	0.324 (0.000)	15590 (011.3)	0.281 (0.001)	0.127 (0.001)	1.016 (0.001)	15.741 (0.033)	0.714 (0.013)	**2.36**
$\text{HHP}_{\neg\text{Stateful}}$		0.325	15559	0.283	0.125	1.017	15.793	0.720	2.79
$\text{HHP}_{\neg\text{Hyper}}$	(Ablations)	0.328	15516	0.283	0.125	1.025	15.831	0.772	3.79
$\text{HHP}_{\neg\text{Latent}}$		0.339	15672	0.294	0.126	1.038	15.888	0.804	6.36

(c) Mark prediction accuracy at the observed next-event time. Higher accuracy values indicate better performance.

Model		Accuracy, % (↑)							Average
		Amazon	Retweet	Taxi	Taobao	StackOverflow	Last.fm	MIMIC-II	Rank* (↓)
RMTPP	[Du et al., 2016]	30.8 (0.1)	53.4 (0.6)	91.4 (0.1)	60.9 (0.1)	45.6 (0.3)	52.5 (0.1)	92.3 (0.3)	7.50
NHP	[Mei and Eisner, 2017]	39.4 (0.1)	61.4 (0.0)	92.9 (0.1)	61.5 (0.2)	47.1 (0.1)	56.5 (0.1)	94.3 (0.0)	<u>2.50</u>
SAHP	[Zhang et al., 2020]	32.4 (1.0)	57.5 (2.2)	91.4 (0.7)	60.5 (0.2)	44.7 (2.0)	51.8 (0.7)	86.8 (0.9)	8.00
THP	[Zuo et al., 2020]	34.6 (0.1)	60.2 (0.1)	91.4 (0.0)	60.0 (0.0)	46.6 (0.2)	53.3 (0.1)	90.9 (0.2)	6.86
IFTPP	[Shchur et al., 2020]	35.9 (0.1)	50.4 (2.5)	91.8 (0.0)	61.0 (0.1)	45.6 (0.1)	56.4 (0.1)	93.4 (0.1)	5.64
AttNHP	[Yang et al., 2022]	38.9 (0.9)	60.7 (0.2)	92.6 (0.1)	61.3 (0.2)	48.2 (0.2)	55.8 (0.6)	92.9 (0.6)	3.93
S2P2	[Chang et al., 2025]	40.7 (0.0)	61.3 (0.0)	93.1 (0.1)	61.1 (0.1)	47.5 (0.3)	55.8 (0.4)	96.0 (0.4)	<u>2.50</u>
iTHP	[Meng et al., 2024]	35.7 (0.2)	60.9 (0.1)	90.7 (0.1)	56.6 (0.1)	46.7 (0.1)	53.0 (0.0)	94.5 (0.5)	6.14
HHP	**(Ours)**	40.8 (0.1)	61.2 (0.0)	93.0 (0.0)	61.4 (0.1)	47.1 (0.1)	56.6 (0.0)	96.9 (0.2)	**1.93**
$\text{HHP}_{\neg\text{Stateful}}$		40.9	61.1	92.9	61.7	47.3	56.4	96.8	2.57
$\text{HHP}_{\neg\text{Hyper}}$	(Ablations)	40.3	57.2	92.4	61.4	46.8	53.5	95.0	4.64
$\text{HHP}_{\neg\text{Latent}}$		34.0	57.6	91.2	60.6	46.6	54.3	90.5	7.50

* Each ablation is ranked separately against the nine main models; other ablations are excluded.

6.1 Practical Interpretability Tools for HHP

Using the linear recurrence of the HHP, we can directly probe how individual event particles affect the model's predictions under the observed dynamics. In this subsection, we introduce tools to quantify, visualize, and interpret the changes in predicted intensity from removing specific particles or groups of particles. These tools provide insight into the model's internal mechanism, going beyond basic aggregate parameter inspection.

Particle View. A central feature of both the LHP and the HHP is that the latent state can be decomposed into a sum of event-specific contributions, which we refer to as *particles*. In the LHP, event i contributes $\mathbf{p}_i(t) = e^{-\boldsymbol{\beta}(t-t_i)}\boldsymbol{\alpha}\mathbf{e}_{k_i}$ for $t > t_i$; the state is their sum. In HHP, the same decomposition holds conditional on the transition matrices generated by the observed history, with initial Hawkes state $\mathbf{x}_0 = \mathbf{0}$. We set $\mathbf{p}_i(t) = \mathbf{0}$ for $t \leq t_i$.

Our HHP preserves this structure, but with more expressive, history-dependent dynamics:

$$\mathbf{p}_i(t) := \left(\prod_{j=i}^{N_{t-}} \mathbf{V}_j e^{\mathbf{D}_j(\min\{t,t_{j+1}\}-t_j)} \mathbf{V}_j^* \right) \boldsymbol{\alpha} \mathbf{e}_{k_i}, \tag{13}$$

$$\mathbf{x}_{t-} = \sum_{i:t_i<t} \mathbf{p}_i(t), \tag{14}$$

$$\boldsymbol{\lambda}_t = \sigma\left(\boldsymbol{\mu} + \sum_{i:t_i<t} \mathbf{c}_i(t) \right), \ \mathbf{c}_i(t) := \mathbf{W}\mathbf{p}_i(t). \tag{15}$$

The product is ordered with later transitions on the left. Here $\mathbf{p}_i(t)$ is latent and $\mathbf{c}_i(t)$ is its pre-softplus readout contribution; the softplus output itself is not additive. For optional complex states, take the real part of the readout before softplus. Standard prediction maintains only the aggregate state, while particle reconstruction is an optional analysis. Edits, additions, or removals of events can also change later hypernetwork-generated transitions, which this decomposition holds fixed.

Reflection: A key part of HHP's design is that, after each new event, the updated dynamics apply uniformly to all existing particles. Because there is no skip connection from hypernetwork to output, the HHP cannot bypass the aggregation of particles to directly predict intensities. The hypernetwork sets history-dependent transitions that transport and decay these particles; its internal computation is not explained by the particle decomposition.

Leave-one-out. Due to nonlinear σ, a particle's output contrast depends on the other retained particles. Inspired by regression diagnostics [Belsley et al., 2005], we define a conditional leave-one-particle-out intensity contrast:

$$\mathrm{DF}\boldsymbol{\lambda}_t^{(i)} := \boldsymbol{\lambda}_t(\mathcal{H}_{t-}) - \sigma\left(\boldsymbol{\mu} + \mathbf{W}[\mathbf{x}_{t-} - \mathbf{p}_i(t)]\right). \tag{16}$$

For complex states, the second term uses the real readout, as above. Positive and negative components indicate particle-carried support and inhibition under the realized transitions. Summing over the K marks gives the contrast in total intensity. These nonlinear contrasts need not add to $\boldsymbol{\lambda}_t - \sigma(\boldsymbol{\mu})$. Removing an event and rerunning the model also changes later transitions, so this is not total event-deletion sensitivity. We visualize conditional particle contrasts in Figure 2.

Cumulative Effects. Integrating the conditional contrast gives $\mathrm{DF}\boldsymbol{\Lambda}_t^{(i)} := \int_0^t \mathrm{DF}\boldsymbol{\lambda}_s^{(i)} \mathrm{d}s$, with expected-count units. The integrated intensity $\boldsymbol{\Lambda}_t = \int_0^t \boldsymbol{\lambda}_s \mathrm{d}s$ is the pathwise compensator; under the model and suitable integrability, $\mathbb{E}[\mathbf{N}_t] = \mathbb{E}[\boldsymbol{\Lambda}_t]$. The integrated particle contrast is not a full counterfactual change in expected count. Integrating $|\mathrm{DF}\boldsymbol{\lambda}_t^{(i)}|$ summarizes unsigned particle use.

Group Influences. Finally, the linear structure of HHP enables us to extend these analyses to groups of events. Jointly removing a set A of events preceding t gives $\mathrm{DF}\boldsymbol{\lambda}_t^{(A)} := \boldsymbol{\lambda}_t - \sigma(\boldsymbol{\mu} + \mathbf{W}[\mathbf{x}_{t-} - \sum_{j\in A} \mathbf{p}_j(t)])$. This probes group effects under fixed transitions; the difference from a sum of individual contrasts reflects the nonlinear readout, as illustrated in Section D.

These tools probe how individual event particles enter the model's predictions under the realized dynamics. Cached observed-history transitions can be reused for requested contrasts. Materializing all source trajectories at all N event times can require $\mathcal{O}(N^2)$ values and work; all pairs at one target require $\mathcal{O}(N^2)$ contrasts, while enumerating arbitrary subsets is exponential. Full event deletion requires a separate sequence rerun per deletion.

6.2 Empirical Exploration

We now explore these probes using a synthetic memory task, shown in Figure 2. Marks are drawn sequentially from a homogeneous Poisson process, until a green "trigger" event is drawn, causing the previous mark to be repeated a predictable time later. In Figure 2 (top), the learned intensities show a delayed spike before returning toward background activity.

In the middle left panel, we show conditional particle contrasts for instantaneous *total* intensity. The green trigger's retained particle inhibits intensity during the delay and later provides strong support near the response under the realized transition path.

Small blue/orange particle contrasts do not imply that their source events are unimportant: an earlier mark can change subsequent generated transitions and thereby determine the preferred response. Figure 2 visualizes conditional particle use, not total event importance. Held-out deletion and fixed-time component analyses are given in Section D.3.

The bottom panel shows mark-specific conditional contrasts for two trigger particles (events 7 and 11) and two earlier non-trigger particles (events 6 and 10). In this trace, the trigger particles support the delayed response and inhibit competing marks. These panels describe the retained-particle pathway for one illustrated sequence. Additional synthetic examples are included in Section D.

6.3 Interpreting Interpretability

Our notion of interpretability aligns with mechanistic interpretability as defined by Bereska and Gavves [2024]: we expose part of the actual computation used to produce outputs. The linear recurrence actually acts as a convenient bottleneck, allowing these meaningful probes to be defined (also exploiting linearity and superposition as discussed by Bereska and Gavves [2024]). Our interpretation is partial, since the internal mechanisms of the hypernetwork remain opaque. Another perspective, following Shmueli [2010], is that the HHP is a *predictive model*, not an *explanatory model*: it forecasts future events, but does not replicate the true generative process. Our interpretability constructs therefore *describe* the model's solution strategy, and not causal relationships. In short, HHP is a predictive model that also offers a mechanistic description of its internal computation, and should not be confused with extracting true causal relationships. These probes expose the realized particle computation, while the hypernetwork pathway remains only partly characterized. Neither particle removal nor altered-history deletion is a causal explanation; fixed-time internal component swaps are diagnostic hybrids rather than realizable histories (Section D.3).

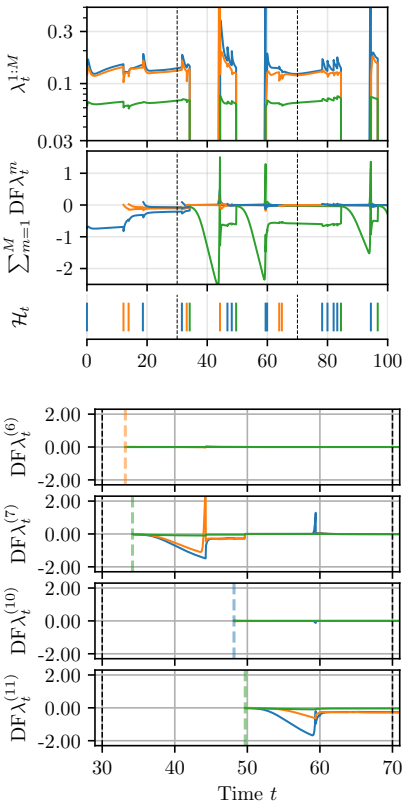


Figure 2: Conditional particle visualizations from Section 6.2. Top: predicted mark-specific intensities. Middle: leave-one-particle-out total-intensity contrasts, colored by source mark. Bottom: mark-specific particle contrasts for selected sources over $(30, 70)$. These contrasts hold the observed-history transition path fixed.

7 Conclusion

We introduce the Hyper Hawkes Process (HHP), an intensity-based MTPP model whose hypernetwork predicts the dynamics of a generalized Hawkes process. HHP is competitive with modern neural MTPPs while exposing a conditional particle decomposition that provides insight into its learned computation.

Interpreting flexible neural models remains challenging, and these probes require careful analysis. Future work will develop systematic methods for domain-specific analysis, integrate them into practical workflows, and explore how the exposed mechanisms can guide model design.

Limitations. Conditional particle contrasts describe retained-particle effects under observed transitions; they do not capture every effect of event deletion or identify causal relationships (Sections D and 6.3). The hypernetwork's internal computation remains opaque, so these probes provide a partial explanation. Finally, the sequential GRU limits parallelism over sequence length compared with architectures such as Chang et al. [2025]. Exploring parallelism in both the linear recurrence and hypernetwork is an interesting future extension.

References

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

Hongyuan Mei and Jason M Eisner. The neural Hawkes process: A neurally self-modulating multivariate point process. *Advances in Neural Information Processing Systems*, 30:6757–6767, 2017.

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.

Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. In *2023 ieee conference on secure and trustworthy machine learning (satml)*, pages 464–483. IEEE, 2023.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 782–791, 2021.

Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*, 2024.

Niru Maheswaranathan and David Sussillo. How recurrent networks implement contextual processing in sentiment analysis. In *International Conference on Machine Learning*, pages 6608–6619. PMLR, 2020.

David Ha, Andrew M Dai, and Quoc V Le. Hypernetworks. In *International Conference on Learning Representations*, 2017.

David A Belsley, Edwin Kuh, and Roy E Welsch. *Regression diagnostics: Identifying influential data and sources of collinearity*. John Wiley & Sons, 2005.

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 & Data Mining*, pages 1555–1564, 2016.

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

Oleksandr Shchur, Marin Biloš, and Stephan Günnemann. Intensity-free learning of temporal point processes. In *International Conference on Learning Representations*, 2020.

Chenghao Yang, Hongyuan Mei, and Jason Eisner. Transformer embeddings of irregularly spaced events and their participants. In *Proceedings of the tenth international conference on learning representations (ICLR)*, 2022.

Yuxin Chang, Alex James Boyd, Cao Xiao, Taha Kass-Hout, Parminder Bhatia, Padhraic Smyth, and Andrew Warrington. Deep continuous-time state-space models for marked event sequences. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Zizhuo Meng, Ke Wan, Yadong Huang, Zhidong Li, Yang Wang, and Feng Zhou. Interpretable transformer Hawkes processes: Unveiling complex interactions in social networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2200–2211, 2024.

Daryl J Daley and David Vere-Jones. *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*. Springer, 2003.

Thomas Miconi. Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. *Elife*, 6:e20899, 2017.

Li Jing, Yichen Shen, Tena Dubcek, John Peurifoy, Scott Skirlo, Yann LeCun, Max Tegmark, and Marin Soljačić. Tunable efficient unitary neural networks (eunn) and their application to rnns. In *International Conference on Machine Learning*, pages 1733–1741. PMLR, 2017.

William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models. *arXiv preprint arXiv:2404.08819*, 2024.

Thomas Josef Liniger. *Multivariate Hawkes processes*. PhD thesis, ETH Zurich, 2009.

Vladislav Zhuzhel, Vsevolod Grabar, Galina Boeva, Artem Zabolotnyi, Alexander Stepikin, Vladimir Zholobov, Maria Ivanova, Mikhail Orlov, Ivan Kireev, Evgeny Burnaev, et al. Continuous-time convolutions model of event sequences. *arXiv preprint arXiv:2302.06247*, 2023.

Anningzhe Gao, Shan Dai, and Yan Hu. Mamba Hawkes process. *arXiv preprint arXiv:2407.05302*, 2024.

Yujee Song, LEE Donghyun, Rui Meng, and Won Hwa Kim. Decoupled marked temporal point process using neural ordinary differential equations. In *The Twelfth International Conference on Learning Representations*, 2024.

Shuang Li, Mingqaun Feng, Lu Wang, Abdelmajid Essofi, Yufeng Cao, Junchi Yan, and Le Song. Explaining point processes by learning interpretable temporal logic rules. In *International Conference on Learning Representations*, 2022.

Yang Yang, Chao Yang, Boyang Li, Yinghao Fu, and Shuang Li. Neuro-symbolic temporal point processes. In *International Conference on Machine Learning*, pages 56665–56680. PMLR, 2024.

Shuang Li, Lu Wang, Ruizhi Zhang, Xiaofu Chang, Xuqin Liu, Yao Xie, Yuan Qi, and Le Song. Temporal logic point processes. In *International Conference on Machine Learning*, pages 5990–6000. PMLR, 2020.

Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

Jimmy TH Smith, Andrew Warrington, and Scott Linderman. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations*, 2022.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Claudio Heinrich-Mertsching, Thordis L Thorarinsdottir, Peter Guttorp, and Max Schneider. Validation of point process predictions with proper scoring rules. *Scandinavian Journal of Statistics*, 51 (4):1533–1566, 2024.

Tanguy Bosser and Souhaib Ben Taieb. On the predictive accuracy of neural temporal point process models for continuous-time event data. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. Survey Certification.

Leonard Bereska and Stratis Gavves. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. Survey Certification, Expert Certification.

Galit Shmueli. To explain or to predict? *Statistical science*, pages 289–310, 2010.

Siqiao Xue, Xiaoming Shi, Zhixuan Chu, Yan Wang, Hongyan Hao, Fan Zhou, Caigao Jiang, Chen Pan, James Y Zhang, Qingsong Wen, et al. EasyTPP: Towards open benchmarking temporal point processes. In *The Twelfth International Conference on Learning Representations*, 2024.

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197, 2019.

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

Chris Whong. FOILing NYC's taxi trip data. `https://chriswhong.com/open-data/foil_nyc_taxi/`, 2014. [Online; accessed Oct 15, 2024].

Hongyuan Mei, Guanghui Qin, and Jason Eisner. Imputing missing events in continuous-time event streams. In *International Conference on Machine Learning*, pages 4475–4485. PMLR, 2019.

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.

Mohammed Saeed, Christine Lieu, Greg Raber, and Roger G Mark. MIMIC II: a massive temporal icu patient database to support research in intelligent patient monitoring. In *Computers in cardiology*, pages 641–644. IEEE, 2002.

Òscar Celma Herrada et al. *Music recommendation and discovery in the long tail*. Universitat Pompeu Fabra, 2009.

Brian McFee, Thierry Bertin-Mahieux, Daniel PW Ellis, and Gert RG Lanckriet. The million song dataset challenge. In *Proceedings of the 21st International Conference on World Wide Web*, pages 909–916, 2012.

Table of Contents

- Section A: NeurIPS Reproducibility Checklist
- Section B: Full Model & Implementation Details
- Section C: Additional Quantitative Results & Experiment Details
- Section D: Additional Interpretability Explorations

A NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We evaluate prediction in Section 5 and conditional particle probes in Section 6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We extensively discuss in both Section 6.3 and Section 7 the bounds and limitations of the model and associated interpretability probes we provide, as well as the architectural limitations of the HHP model.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We do not include "pure" theoretical results in our submission, but note we do provide a thorough mathematical derivation of the HHP model and how it relates to existing models.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Experimental configurations and evaluation details are provided in Section C, with a reference for the previously reported baseline configurations.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The HHP implementation uses EasyTPP and publicly available datasets.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We use the field-standard EasyTPP dataset preparation and experiment runners. Details of hyperparameter selection are included in Section C.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Main-model benchmarks report standard deviations across five randomly initialized models. The held-out particle analysis reports 95% sequence-cluster bootstrap intervals; ablations are reported as point estimates.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include both the GPU used and dataset-specific training times.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We reviewed the Ethics Guidelines and believe there are no significant ethical issues posed by our submission. While the interpretability could be used to extract enhanced information about underlying data, this is not unique to our model in MTPPs and is inline with existing work in the field of interpretability. The particle probes describe model computation rather than causal effects, as discussed in Section 6.3.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

Justification: This work develops a general-purpose point-process architecture and diagnostic tools, without proposing a deployment or application-specific decision system.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: Data is already in the public domain. MTPP models are not a candidate for high risk of misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite EasyTPP and the original dataset sources in Section C.4.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our contribution is the HHP architecture and its EasyTPP-based implementation. The architecture, selected configurations and evaluation procedure are described in Sections B and C.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No crowdsourcing or human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: This research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

B Full Model & Implementation Details

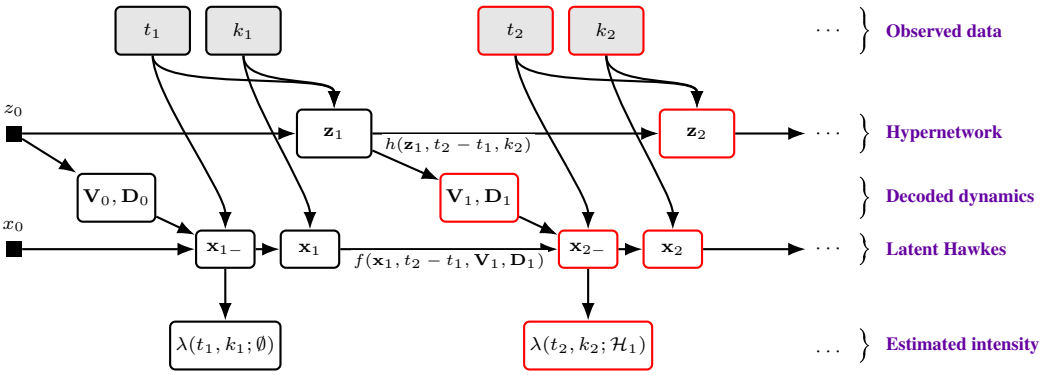


Figure 3: The full hyper Hawkes process architecture. We highlight data that is conditioned on with shaded boxes, and the variables that are updated/used in a single iteration, i.e., when the second observation becomes available. The top row represents the history $\mathcal{H}_t$, the second row represents the hypernetwork recurrence, the third row represents the latent Hawkes process, and the bottom row shows the intensities. Note we suppress the arrow from t_1 into $\mathbf{x}_{2-}$ for visual clarity.

We now summarize the model referred to as *the* Hyper Hawkes Process (HHP) throughout this paper. Its overall architecture is illustrated in Figure 3. The HHP is a recurrent neural marked temporal point process (MTPP) model, composed of a conditionally linear latent Hawkes recurrence and a hypernetwork h_ϕ that generates its interval dynamics.

B.1 Recurrent Update Mechanism

We begin by detailing how the HHP transitions from the i-th event to the $(i+1)$-th event. This step is highlighted in red in Figure 3, where $i = 1$ and $i + 1 = 2$. We write $\mathbf{x}_{t_i-}$ and $\mathbf{x}_{t_i+}$ for the states immediately before and after event i, and $\delta_i = t_i - t_{i-1}$ for its incoming gap.

After consuming event i, the hypernetwork takes its mark and *incoming* gap, δ_i, to produce $\mathbf{z}_i = h_\phi(\mathbf{z}_{i-1}, [\mathrm{emb}(k_i), \log(\delta_i + \epsilon)])$. It emits $\mathbf{V}_i, \mathbf{D}_i$ for the *following* interval of length $\Delta_i = t_{i+1} - t_i$; event $i+1$ has not yet been consumed. The three-step state and readout recurrence is

$$\begin{aligned}
\mathbf{x}_{t_{i+1}-} &= \mathbf{V}_i e^{\mathbf{D}_i \Delta_i} \mathbf{V}_i^* \mathbf{x}_{t_i+}, \\
\boldsymbol{\lambda}_{t_{i+1}-} &= \sigma(\boldsymbol{\mu} + \mathbf{W}\mathbf{x}_{t_{i+1}-}), \\
\mathbf{x}_{t_{i+1}+} &= \mathbf{x}_{t_{i+1}-} + \boldsymbol{\alpha}\mathbf{e}_{k_{i+1}}.
\end{aligned}$$

The left-limit readout scores event $i+1$ without using its mark as an input. Here $\sigma(a) = \log(1 + e^a)$ acts elementwise. The real-state configuration uses real $\mathbf{x}, \boldsymbol{\alpha}, \mathbf{W}$ and orthogonal $\mathbf{V}_i$. The implementation also permits complex states and unitary $\mathbf{V}_i$, with the real part of the readout taken before softplus. Benchmark-specific choices and learned output scales are noted in Section C.1.

Quantity	Shape in the real-state configuration
$\mathbf{x}_{t_i\pm}, \mathbf{p}_j(t)$	$\mathbb{R}^d$
$\mathrm{d}\mathbf{N}_t, \mathbf{e}_{k_i}$	$\mathbb{R}^K$
$\boldsymbol{\alpha}$ and $\mathbf{W}$	$\mathbb{R}^{d\times K}$ and $\mathbb{R}^{K\times d}$
$\boldsymbol{\mu}, \boldsymbol{\lambda}_t$	$\mathbb{R}^K$
$\mathbf{V}_i, \mathbf{D}_i$	$\mathbb{R}^{d\times d}$; $\mathbf{D}_i$ diagonal

The diagonal $\mathbf{D}_i$ combines learned negative-real base eigenvalues with positive scales emitted from $\mathbf{z}_i$, so its real entries are negative. In the optional complex-state variant, the base eigenvalues may also have imaginary parts. A separate transform of $\mathbf{z}_i$ produces the parameters for $\mathbf{V}_i$ using the structured orthogonal/unitary construction of Jing et al. [2017]; r sets its number of component blocks. This structure reduces matrix-vector products to component-wise operations and permutations.

This completes the iteration, with a new hypernetwork state and latent Hawkes state ready for the next iteration.

Computational Complexity. For fixed model dimensions, one streaming update uses constant state memory and work. Processing N events uses $\mathcal{O}(N)$ work and an $\mathcal{O}(N)$ critical path through the GRU. Retaining all interval outputs for training uses $\mathcal{O}(N)$ storage.

B.2 Architecture Hyperparameters and Learnable Parameters

The hypernetwork depends solely on the event history and emits the parameters $\mathbf{V}_i$ and $\mathbf{D}_i$ for the Hawkes recurrence. We use a GRU combined with deterministic orthonormal matrix construction [Jing et al., 2017]. The GRU state $\mathbf{z}_i$ takes as input the logarithm of the time since the last event and a mark embedding. This embedding dimension is a hyperparameter, set to one less than the GRU state dimension to maintain consistent input size. We collect the learned dynamics-generator parameters in $\phi \in \Phi$: the mark embedding, GRU weights and initial state, output heads generating the basis parameters and decay scales, and learned base eigenvalues.

The learnable parameters of the HHP are therefore:

- Dynamics-generator parameters $\phi \in \Phi$ as defined above,
- Mark-specific impulses $\boldsymbol{\alpha} \in \mathbb{R}^{d \times K}$,
- Emission layer parameters: projection matrix $\mathbf{W} \in \mathbb{R}^{K \times d}$ and background intensity $\boldsymbol{\mu} \in \mathbb{R}^K$.

For fixed softplus, the parameter set is:

$$\theta = \{\phi, \boldsymbol{\alpha}, \mathbf{W}, \boldsymbol{\mu}\} \in \Theta.$$

Key architectural hyperparameters include:

- Latent dimension of the Hawkes process (d),
- Dimension of the hypernetwork recurrence (we use a GRU and do not explore alternatives here).

B.3 Computing the Log-Likelihood

The log-likelihood for intensity-based MTPPs is defined in Equation (2), with further background in Daley and Vere-Jones [2003]. At a high level:

1. For a given event sequence, we compute the left-limit intensities for each observed mark type k_i, denoted $\left\{\lambda_{t_i}^{k_i}\right\}_{i=1}^{L}$, using the procedure in Section B.1.
2. These intensities form the first term of the log-likelihood.
3. To approximate the normalizing integral, we sample a fixed number of points $t' \in (t_i, t_{i+1})$ uniformly and compute the total intensity $\lambda_{t'} = \sum_{k=1}^{K} \lambda_{t'}^k$ at each sampled time.

The GRU recurrence is computed only once per event, not per sample point, since it is conditioned solely on events. This allows us to amortize its cost across all sampled points.

Computational Complexity. With M samples per interval, likelihood integration takes $\mathcal{O}(NM)$ work for fixed model dimensions. The M intensity evaluations within an interval are conditionally parallel after its right-limit state and parameters are available. A scan can parallelize the linear recurrence once all interval transitions are known, but the implemented GRU generates those transitions sequentially, so the full model retains an $\mathcal{O}(N)$ critical path. A Transformer or state-space hypernetwork could change this dependency, but those are untested architectural variants with their own work and memory costs. Materializing all N source-particle trajectories across N target times can use $\mathcal{O}(N^2)$ work and storage; exhaustive pairwise contrasts at one target also require $\mathcal{O}(N^2)$ evaluations.

C Additional Results & Experiment Details

This appendix provides experimental details and results for the Hyper Hawkes Process (HHP) and comparators. Experiments use EasyTPP [Xue et al., 2024], with dataset preparation and splits described below.

C.1 Training Details & Hyperparameter Configurations

For baselines other than iTHP, we use the hyperparameters and architectures in Chang et al. [2025]. HHP selection used validation log-likelihood over latent dimension $d \in \{8, 16, 32, 64, 128, 256\}$, GRU hidden size $h \in \{8, 16, 32, 64, 128\}$, GRU layers $l \in \{1, 2\}$, and rotation blocks $r \in \{2, 4, 8\}$. The selected configurations are listed in Table 3.

Table 3: Chosen hyperparameters for HHP across all seven benchmark datasets.

Dataset	d	h	l	r	**# Parameters**
Amazon	64	8	2	8	11257
Retweet	64	32	2	8	23940
Taxi	128	8	2	8	9667
Taobao	64	8	2	4	5122
StackOverflow	64	8	2	8	6959
Last.fm	64	32	2	8	42777
MIMIC-II	256	16	2	8	126412

HHP training used Adam with learning rate 0.01, a 300-epoch budget, and batch size 256 (128 for Last.fm), selecting checkpoints by validation log-likelihood. For iTHP, the selected width was $M = 64$ for Amazon and Retweet and $M = 16$ otherwise. Learning rates were 0.002 for Amazon, Taxi and MIMIC-II, 0.001 for Retweet, StackOverflow and Last.fm, and 0.0005 for Taobao. Batch size was 16 (4 for Last.fm), with 300 epochs (600 for MIMIC-II) and validation-likelihood checkpoint selection.

The selected Amazon and MIMIC-II HHP models use complex latent states; the other five use real states. Benchmark models additionally learn a positive output scale b_k for each mark, using $\sigma_k(a) = b_k^{-1} \log(1 + e^{b_k a})$ in place of fixed softplus. These scales extend the parameter inventory in Section B and are included in the counts above (including the implementation's padding channel). The implementation's factor of two in the real readout is absorbed into $\mathbf{W}$ in our notation.

The reported HHP runs used a single Nvidia H100 GPU (80GB). Among the archived completed runs, training took less than one hour for Taxi and Taobao, up to approximately 1.5 hours for StackOverflow and MIMIC-II, 5.7–9.2 hours for Amazon, 30.5–32.2 hours for Retweet, and 42.2–43.8 hours for Last.fm.

C.2 Metric Definitions and Evaluation Protocol

For an observed next event (t_i, k_i) following history $\mathcal{H}_{t_{i-1}}$, let $\lambda_i^k = \lambda_{t_i-}^k$ and $I_i = \int_{t_{i-1}}^{t_i} \sum_k \lambda_s^k \mathrm{d}s$. Its total log-likelihood contribution is $\ell_{\mathrm{total},i} = \log \lambda_i^{k_i} - I_i$. We split this into time and mark terms $\ell_{\mathrm{time},i} = \log(\sum_k \lambda_i^k) - I_i$ and $\ell_{\mathrm{mark},i} = \log(\lambda_i^{k_i} / \sum_k \lambda_i^k)$, so $\ell_{\mathrm{total},i} = \ell_{\mathrm{time},i} + \ell_{\mathrm{mark},i}$. Reported likelihoods average over valid scored events and are in nats per event. HHP and iTHP are scored with the same EasyTPP evaluator, using 100 Monte Carlo samples per inter-event interval for the integral and scoring every event after the first in each sequence.

Next-time RMSE is $\sqrt{n^{-1} \sum_i (\widehat{\delta}_i - \delta_i)^2}$ in each dataset's time units. The inherited EasyTPP one-step evaluator approximates the expected next gap by numerical integration; it predicts the next mark by maximizing normalized mark intensity at the *observed* next-event time. Accuracy is $n^{-1} \sum_i \mathbf{1}\{\widehat{k}_i = k_i\}$, printed as a percentage. Next-time integration uses 50,000 quadrature points. Main-model entries report means over five seeds, with standard deviations in parentheses; ablation entries report point estimates.

For time calibration, define $u_i = 1 - e^{-I_i}$ at the observed next gap. At probability grid points p_b, PCE is the mean of $|n^{-1} \sum_i \mathbf{1}\{u_i \le p_b\} - p_b|$. For mark calibration, let $q_i = \max_k \lambda_i^k / \sum_j \lambda_i^j$ and partition q_i into confidence bins B_b; ECE is $\sum_b |B_b|/n \,|\, \mathrm{acc}(B_b) - \mathrm{conf}(B_b)|$. Both are printed as percentages. Calibration uses 50 PCE grid points and 20 ECE bins, with 50 Monte Carlo samples per interval for HHP and iTHP.

Within each dataset and metric, the nine main models are ranked by their five-seed means, with higher better for likelihood and accuracy and lower better for RMSE, PCE, and ECE; ablations are excluded. Ties at the displayed precision receive midranks. Each ablation is ranked separately against the nine main models, excluding the other ablations. Each metric's average rank is the mean over the seven datasets. The composite is the mean of the six time/mark metric ranks; total likelihood is reported separately. The calibration tables' average PCE and ECE values are arithmetic means over the seven datasets. Ranks are calculated from the displayed values.

C.3 Dataset Statistics

We report the statistics of all seven datasets used in this work in Table 4. We used the `Hugging Face` versions of Amazon, Retweet, Taxi, Taobao and StackOverflow. For all datasets, we ensured that no two events occur at the same time (i.e., inter-arrival time is strictly positive), and event times do not lie on grid points that are effectively discrete-time events. Dataset descriptions and pre-processing details are provided in Section C.4.

C.4 Dataset Pre-processing

We used the default train/validation/test splits for Amazon, Retweet, Taxi, Taobao and StackOverflow. For MIMIC-II, we followed Du et al. [2016] and kept the 325 test sequences in the test split, further splitting the 2,935 training sequences into 2,600 for training and 325 for validation. For Last.fm, we randomly partitioned the data into 70%, 15%, and 15% splits for training, validation, and test, respectively. For all datasets, a small amount of jitter was added to event times if necessary to ensure no two events occurred at the same time and to avoid discrete-time artifacts.

Amazon [Ni et al., 2019] contains user product reviews, with product categories as marks. **Retweet** [Zhao et al., 2015] models retweet cascades, with event types based on user influence. **Taxi** [Whong, 2014, Mei et al., 2019] uses New York taxi pickup/dropoff data, with marks defined by location-action pairs. **Taobao** [Xue et al., 2022] consists of e-commerce viewing patterns, with item categories as marks. **StackOverflow** contains badges awarded to users on a Q&A website, with badge type as the mark. **MIMIC-II** [Saeed et al., 2002] records disease events during hospital visits, with disease type as the mark. For MIMIC-II and StackOverflow, we used the pre-processing from Du et al. [2016]. **Last.fm** [Celma Herrada et al., 2009, McFee et al., 2012] records music listening habits, with genres as marks. Each event is a play of a particular genre, and if a song had multiple genres, one was selected at random.

C.5 Full Results on Benchmark Datasets

Table 5 decomposes per-event likelihood into time and mark terms. Table 6 reports time PCE and mark ECE following Bosser and Taieb [2023]. HHP exceeds iTHP in time log-likelihood on every dataset and in mark log-likelihood on six of seven; iTHP is slightly higher on StackOverflow mark

Table 4: Statistics of the seven datasets we experiment with.

Dataset	K	**Number of Events**			**Sequence Length**			**Number of Sequences**		
		Train	Valid	Test	Min	Max	Mean	Train	Valid	Test
Amazon	16	288,377	40,995	84,048	14	94	44.8	6,454	922	1,851
Retweet	3	2,176,116	215,521	218,465	50	264	108.8	20,000	2,000	2,000
Taxi	10	51,584	7,404	14,820	36	38	37.0	1,400	200	400
Taobao	17	73,483	11,472	28,455	28	64	56.7	1,300	200	500
StackOverflow	22	90,497	25,762	26,518	41	101	64.8	1,401	401	401
Last.fm	120	1,534,738	344,542	336,676	6	501	207.2	7,488	1,604	1,604
MIMIC-II	75	9,619	1,253	1,223	2	33	3.7	2600	325	325

Table 5: Full breakdown of per-event log-likelihood metrics. Aggregate ranks average over the seven datasets.

Model	Per Event Log-Likelihood, $\mathcal{L}_{\text{Total}}$ (nats) (↑)							Avg. Ranking (↓)
	Amazon	Retweet	Taxi	Taobao	StackOverflow	Last.fm	MIMIC-II	
RMTPP	-2.136 (0.003)	-7.098 (0.217)	0.346 (0.002)	1.003 (0.004)	-2.480 (0.019)	-1.780 (0.005)	-0.472 (0.026)	7.86
NHP	0.129 (0.012)	-6.348 (0.000)	0.514 (0.004)	1.157 (0.004)	-2.241 (0.002)	-0.574 (0.011)	0.060 (0.017)	4.14
SAHP	-2.074 (0.029)	-6.708 (0.029)	0.298 (0.057)	1.168 (0.028)	-2.341 (0.058)	-1.646 (0.083)	-0.677 (0.072)	7.14
THP	-2.096 (0.002)	-6.659 (0.007)	0.372 (0.002)	0.790 (0.002)	-2.338 (0.014)	-1.712 (0.011)	-0.577 (0.011)	7.29
IFTPP	0.496 (0.002)	-10.344 (0.016)	0.453 (0.002)	1.318 (0.017)	-2.233 (0.009)	-0.492 (0.017)	0.317 (0.052)	3.71
AttNHP	0.484 (0.077)	-6.499 (0.028)	0.493 (0.009)	1.259 (0.022)	-2.194 (0.016)	-0.592 (0.051)	-0.170 (0.077)	4.14
S2P2	0.781 (0.011)	-6.365 (0.003)	0.522 (0.004)	1.304 (0.039)	-2.163 (0.009)	-0.557 (0.046)	0.919 (0.069)	**1.71**
iTHP	-1.481 (0.008)	-6.4638 (0.0015)	0.285 (0.004)	0.841 (0.004)	-2.288 (0.002)	-2.085 (0.005)	0.194 (0.029)	6.57
HHP (ours)	0.616 (0.016)	-6.366 (0.007)	0.520 (0.003)	1.232 (0.014)	-2.209 (0.006)	-0.515 (0.006)	1.314 (0.048)	<u>2.43</u>

Model	Per Event Next Event Time Log-Likelihood, $\mathcal{L}_{\text{Time}}$ (nats) (↑)							Avg. Ranking (↓)
	Amazon	Retweet	Taxi	Taobao	StackOverflow	Last.fm	MIMIC-II	
RMTPP	0.011 (0.001)	-6.191 (0.083)	0.622 (0.002)	2.428 (0.004)	-0.797 (0.005)	0.256 (0.007)	-0.188 (0.016)	7.57
NHP	2.116 (0.009)	-5.584 (0.001)	0.727 (0.003)	2.578 (0.006)	-0.699 (0.002)	1.198 (0.006)	0.225 (0.016)	4.43
SAHP	0.115 (0.049)	-5.872 (0.062)	0.645 (0.044)	2.604 (0.008)	-0.703 (0.031)	0.489 (0.078)	-0.244 (0.040)	6.29
THP	-0.068 (0.002)	-5.874 (0.007)	0.621 (0.002)	2.242 (0.002)	-0.772 (0.006)	0.220 (0.010)	-0.271 (0.004)	8.29
IFTPP	2.483 (0.001)	-9.500 (0.011)	0.735 (0.002)	2.708 (0.018)	-0.662 (0.007)	1.277 (0.016)	0.555 (0.050)	3.00
AttNHP	2.416 (0.092)	-5.726 (0.027)	0.714 (0.010)	2.654 (0.007)	-0.684 (0.005)	1.203 (0.015)	0.031 (0.055)	4.43
S2P2	2.652 (0.009)	-5.598 (0.002)	0.733 (0.003)	2.719 (0.038)	-0.641 (0.003)	1.257 (0.022)	1.050 (0.065)	**1.71**
iTHP	0.521 (0.008)	-5.6911 (0.0014)	0.591 (0.003)	2.320 (0.002)	-0.753 (0.002)	-0.224 (0.004)	0.342 (0.033)	6.71
HHP (ours)	2.492 (0.015)	-5.597 (0.005)	0.732 (0.002)	2.620 (0.013)	-0.670 (0.002)	1.252 (0.004)	1.394 (0.043)	<u>2.57</u>

Model	Per Event Next Mark Log-Likelihood, $\mathcal{L}_{\text{Mark}}$ (nats) (↑)							Avg. Ranking (↓)
	Amazon	Retweet	Taxi	Taobao	StackOverflow	Last.fm	MIMIC-II	
RMTPP	-2.147 (0.003)	-0.908 (0.141)	-0.276 (0.000)	-1.425 (0.002)	-1.683 (0.015)	-2.035 (0.004)	-0.284 (0.014)	7.57
NHP	-1.987 (0.003)	-0.764 (0.000)	-0.213 (0.002)	-1.421 (0.004)	-1.542 (0.001)	-1.772 (0.006)	-0.165 (0.002)	3.57
SAHP	-2.189 (0.030)	-0.836 (0.036)	-0.346 (0.024)	-1.436 (0.027)	-1.638 (0.032)	-2.136 (0.070)	-0.433 (0.031)	8.29
THP	-2.028 (0.002)	-0.785 (0.001)	-0.249 (0.001)	-1.451 (0.000)	-1.566 (0.008)	-1.932 (0.006)	-0.306 (0.009)	6.71
IFTPP	-1.988 (0.001)	-0.844 (0.007)	-0.282 (0.001)	-1.391 (0.005)	-1.571 (0.003)	-1.769 (0.004)	-0.239 (0.002)	5.29
AttNHP	-1.933 (0.024)	-0.773 (0.003)	-0.221 (0.002)	-1.395 (0.016)	-1.510 (0.013)	-1.795 (0.037)	-0.201 (0.025)	3.57
S2P2	-1.871 (0.002)	-0.767 (0.000)	-0.211 (0.002)	-1.415 (0.005)	-1.521 (0.008)	-1.814 (0.025)	-0.131 (0.014)	<u>2.43</u>
iTHP	-2.002 (0.004)	-0.7727 (0.0003)	-0.306 (0.003)	-1.479 (0.002)	-1.535 (0.001)	-1.862 (0.002)	-0.148 (0.008)	5.57
HHP (ours)	-1.877 (0.002)	-0.769 (0.002)	-0.212 (0.001)	-1.388 (0.003)	-1.539 (0.004)	-1.767 (0.003)	-0.079 (0.009)	**2.00**

log-likelihood. iTHP has lower time PCE on Amazon and MIMIC-II, and lower mark ECE on Retweet and StackOverflow. Taxi mark ECE ties at the displayed precision.

Table 6: Calibration results across the benchmark datasets. Aggregate columns average over the seven datasets.

(a) Probabilistic calibration error (PCE) for time calibration in percentage.

Model	**Probabilistic Calibration Error (PCE) (↓)**							**Average**	**Average**
	Amazon	Retweet	Taxi	Taobao	StackOverflow	Last.fm	MIMIC-II	**Ranking (↓)**	**PCE (↓)**
RMTPP	13.67 (0.03)	7.93 (0.62)	3.50 (0.03)	0.22 (0.16)	1.94 (0.10)	1.56 (0.01)	3.63 (0.37)	6.14	4.64
NHP	8.45 (0.28)	0.20 (0.19)	0.87 (0.50)	7.40 (0.68)	1.51 (0.11)	4.70 (0.13)	5.92 (0.14)	5.86	4.15
SAHP	12.04 (1.02)	8.51 (1.86)	2.52 (0.99)	3.18 (0.21)	1.50 (0.57)	2.53 (1.86)	2.28 (0.44)	5.86	4.65
THP	12.38 (0.05)	5.68 (0.08)	3.34 (0.02)	6.36 (0.04)	2.06 (0.11)	1.02 (0.08)	1.10 (0.06)	5.71	4.56
IFTPP	1.59 (0.09)	23.85 (0.26)	0.40 (0.10)	1.61 (0.74)	0.84 (0.34)	0.46 (0.44)	1.75 (0.33)	**2.43**	4.36
AttNHP	6.36 (0.63)	2.09 (0.85)	0.84 (0.27)	3.08 (0.16)	1.65 (0.24)	1.43 (0.14)	4.70 (0.33)	4.86	<u>2.88</u>
S2P2	5.88 (0.17)	0.44 (0.27)	0.55 (0.33)	2.07 (0.32)	1.03 (0.15)	1.38 (0.52)	11.70 (0.68)	<u>3.71</u>	3.29
iTHP	4.96 (0.56)	2.78 (0.19)	2.85 (0.19)	11.81 (0.59)	2.15 (0.16)	18.49 (0.26)	2.04 (0.78)	6.29	6.44
HHP (ours)	6.74 (0.54)	0.59 (0.47)	0.43 (0.18)	2.97 (0.62)	1.01 (0.24)	2.91 (0.38)	4.25 (1.45)	4.14	**2.70**

(b) Expected calibration error (ECE) for mark calibration in percentage.

Model	**Expected Calibration Error (ECE) (↓)**							**Average**	**Average**
	Amazon	Retweet	Taxi	Taobao	StackOverflow	Last.fm	MIMIC-II	**Ranking (↓)**	**ECE (↓)**
RMTPP	6.58 (0.15)	3.99 (4.28)	2.42 (0.16)	1.89 (0.24)	2.10 (0.27)	2.47 (0.45)	2.79 (0.43)	6.86	3.18
NHP	8.30 (0.21)	0.35 (0.06)	0.79 (0.10)	5.59 (0.69)	1.31 (0.16)	3.41 (0.41)	2.24 (0.32)	5.71	3.14
SAHP	8.17 (2.00)	6.27 (2.23)	6.77 (0.21)	2.68 (0.35)	1.71 (0.77)	6.26 (4.30)	5.41 (0.26)	8.00	5.32
THP	2.06 (0.17)	1.26 (0.11)	1.76 (0.07)	6.51 (0.03)	0.81 (0.14)	3.42 (0.70)	2.16 (0.39)	6.00	2.57
IFTPP	0.46 (0.10)	0.95 (1.12)	0.55 (0.19)	1.20 (0.20)	1.28 (0.54)	0.66 (0.05)	1.39 (0.23)	**2.00**	**0.93**
AttNHP	3.13 (0.61)	0.52 (0.16)	0.56 (0.10)	2.47 (0.12)	1.37 (0.42)	0.61 (0.16)	2.23 (0.50)	4.07	1.56
S2P2	0.88 (0.34)	0.52 (0.13)	0.58 (0.12)	1.96 (0.67)	1.98 (0.19)	1.01 (0.63)	1.62 (0.24)	<u>3.79</u>	<u>1.22</u>
iTHP	1.91 (0.23)	0.34 (0.09)	0.83 (0.23)	3.49 (0.46)	1.56 (0.24)	2.36 (0.18)	1.97 (0.40)	4.50	1.78
HHP (ours)	1.53 (0.22)	0.38 (0.38)	0.83 (0.09)	1.91 (0.29)	2.00 (0.64)	1.44 (0.53)	1.54 (0.33)	4.07	1.38

D Interpretability Exploration

In this appendix, we provide more concrete details on the interpretability scenario explored in the main paper, as well as introduce another scenario—along with an accompanying analysis using the proposed interpretability tools.

D.1 Delayed-Repeat Scenario

Data Generating Process. Sequences were sampled one event at a time, being drawn from a Poisson process with rate $\lambda = 1/3$. Marks are then randomly assigned to these events with probability 40% for blue, 40% for orange, and 20% for green. Should a green event be drawn at time t, we denote that as a "trigger" event. The next event has the exact same mark as the event that came before the trigger, and the time of the event will be drawn from $t + \mathcal{N}(10, 0.01)$. After this follow-up event is drawn, we return to drawing from the Poisson process as before. Sampling ends at $T = 100$. See Figure 4 for example sequences generated under this process.

The checkpoint was trained on 6,000 generated sequences, with 2,000 validation and 2,000 held-out test sequences, latent dimension $d = 32$, hidden dimension $h = 8$ in a single-layer GRU, and $r = 2$ component blocks for the orthogonal basis. It has 1,328 parameters and was trained for 300 epochs.

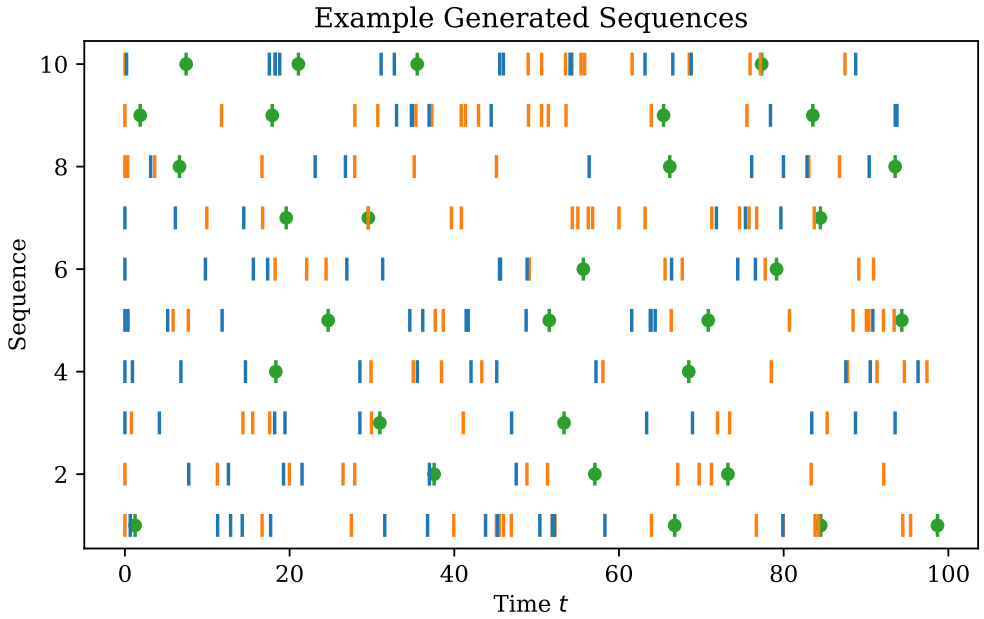


Figure 4: Visualization of ten example sequences drawn from the data generating process that was analyzed in Section 6.2. Trigger events are overlaid with dots for better readability.

D.2 Second Scenario

Data Generating Process. For this new scenario, we will simulate two processes separately and then treat the superposition of them as a single sequence to model. The first process is simple, green events are drawn from a Poisson process at rate $\lambda = 1/2$. For the second process, we will simulate a sequence of pairs of call and response events. We will label a "call" event as blue and draw it from an exponential distribution with rate $\lambda = 1/15$. After a call is drawn, the "response" event, which we will denote as orange, has its time equal to the call event plus a random offset drawn from $\mathcal{N}(10, 0.01)$. Afterwards, another call event is drawn offset from the previous response with the same exponential distribution as before, and so on. The superposition of the two produces a sequence with three possible marks, spanning $t \in [0, 100]$. See Figure 5 for example sequences generated under this process.

Aggregate Statistics. Aggregating conditional particle contrasts across events and sequences describes how the fitted model uses retained particles of each mark. For source i, a lifetime summary is $\sum_{m=1}^{K} \int_0^T |\mathrm{DF}\lambda_t^{(i),m}| \mathrm{d}t$. It has count units but is neither a full event-deletion effect nor a count of events caused by that source.

Figure 6 groups this lifetime particle-use summary by source mark. In this fitted model, green particles have smaller integrated contrasts than call (blue) particles; the call particles' contrasts can persist beyond a subsequent response. These observations concern the realized particle pathway and do not establish the total importance of the corresponding source events.

Output Intensities. We selected a held-out sequence at random from the data generating process. Figure 7 shows the resulting predicted intensities that the HHP model produces when conditioned on the sequence. We can see that when a call (blue) event occurs, the intensities for both call and response (orange) events drop to near zero (but never exactly zero due to $\sigma(z) \in (0, \infty)$). These values remain there until about 10 units of time later when the response intensity spikes. Then after a response event occurs, the intensities reset back to normal. All the while the green intensity is roughly stable and unresponsive to any ongoing events, which mirrors the true generation process. For the remainder of this section, this sequence will be used for subsequent analysis.

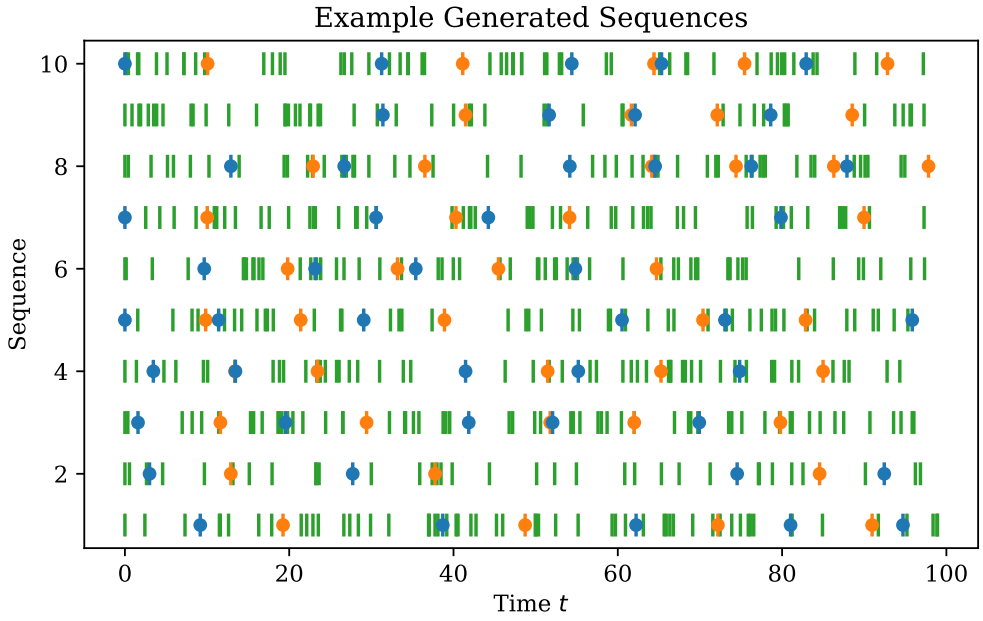


Figure 5: Visualization of ten example sequences drawn from the data generating process in the second synthetic scenario. Call (blue) and response (orange) events are overlaid with dots for better readability. Calls and responses alternate, beginning with a call, with any number of green events in the interim.

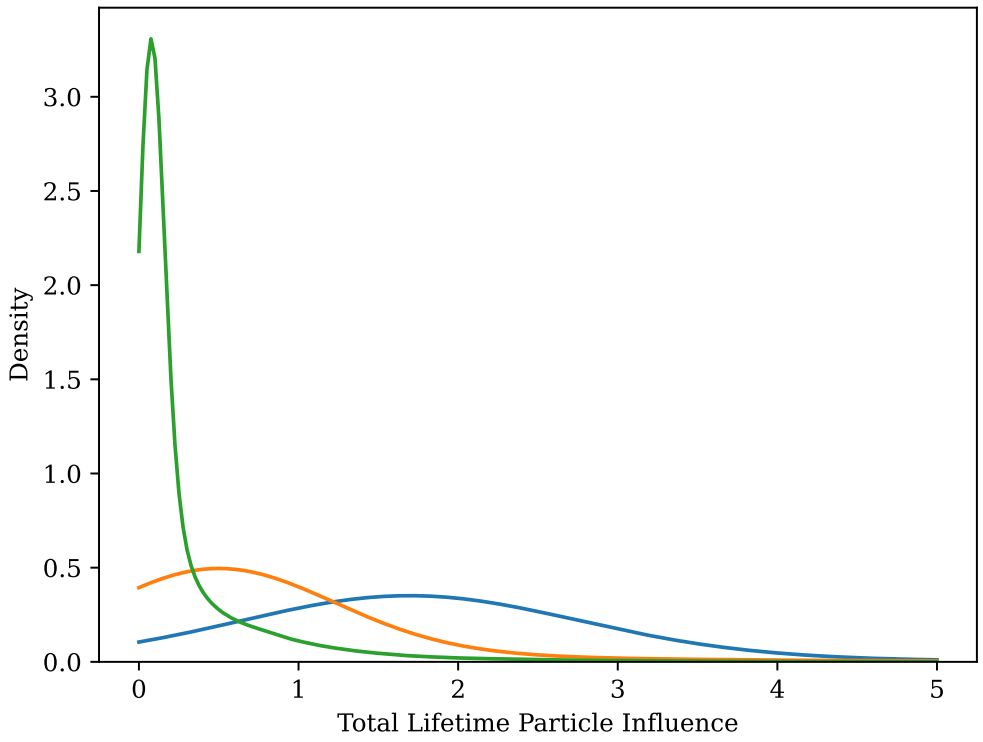


Figure 6: Distribution of integrated absolute conditional contrasts $\sum_{m=1}^{K} \int_0^T |\mathrm{DF}\lambda_t^{(i),m}| \mathrm{d}t$ for source particles, grouped by source mark.

Individual Effects. For the selected sequence, Figure 8 plots each particle's mark-specific conditional contrast $\mathrm{DF}\boldsymbol{\lambda}_t^{(i)}$ over $t \in [0, T]$. These traces show how the fitted model uses each retained particle under the realized transition path.

Before the first response (orange), several retained particles have positive conditional contrasts for response-mark intensity, while others have negative contrasts. This mixed support and inhibition is visible after the readout in Figure 7; the plot alone does not identify latent particle interactions or the effect of deleting those events.

After this first response event occurs, the earlier particles' plotted conditional contrasts become small. This does not establish that the latent particles have been erased. Such suppression is not uniform across the sequence: some particles contribute to spiking for both of the last two response events. When debugging a model, observations like these can motivate mechanisms such as explicit resetting.

Joint Effects. These signals describe conditional removal of one or more particles while the generated transition path is fixed. With that path fixed, latent particles propagate linearly. A joint-removal contrast can differ from the sum of individual contrasts because the output map is nonlinear and each contrast depends on the other retained particles.

For a set of particles A, let $L(A) = \sum_{m=1}^{K} \int_0^T |\mathrm{DF}\lambda_t^{(A),m}| \mathrm{d}t$. The pair diagnostic in Figure 9 is $|L(\{i,j\}) - L(\{i\}) - L(\{j\})|$. It measures nonadditivity after the output nonlinearity for the realized transition path, in integrated-intensity units; it is not a full event-deletion interaction. The heatmap shows larger fixed-dynamics output nonadditivity for pairs involving call (B) and response (O) particles, including a region around adjacent call-response pairs.

Retrospective Attribution. The same contrasts can be evaluated immediately before an observed event to inspect which retained source particles support or inhibit the fitted model's intensity for its mark. For observed event i with mark k_i, we plot $\mathrm{DF}\lambda_{t_i}^{(j),k_i}$ for prior particles $j < i$ in Figure 10. In the illustrated responses, recent call (blue) particles have large conditional contrasts on the intensity

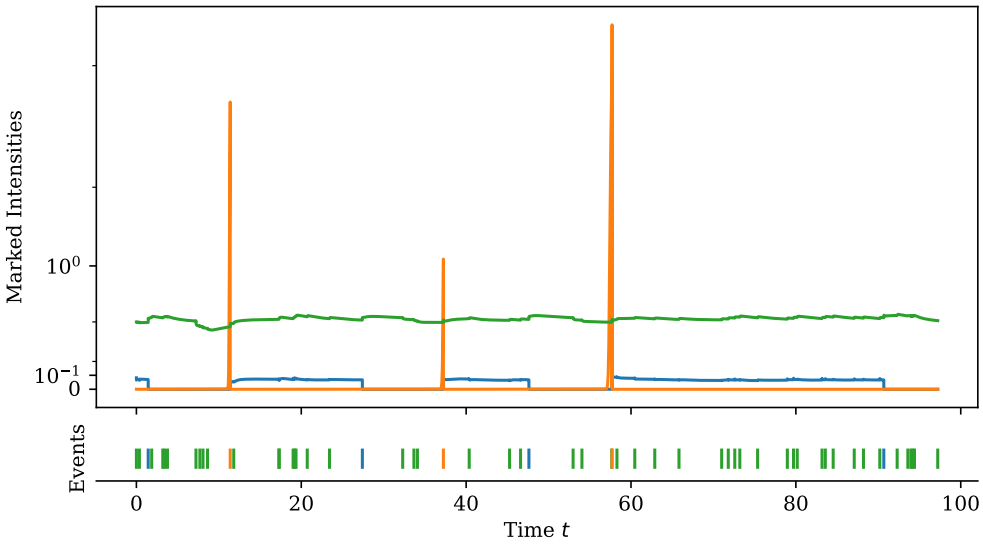


Figure 7: On top are the model's predicted intensities, each line corresponding to a mark matching in color. Below is the corresponding sequence the model is conditioning on to produce the above intensities. Note that when a blue event occurs, the orange and blue intensities drop to near zero, only to have an orange intensity spike around 10 units of time later.

scale. These are output changes under the original transitions, rather than attention probabilities or complete event-importance scores.

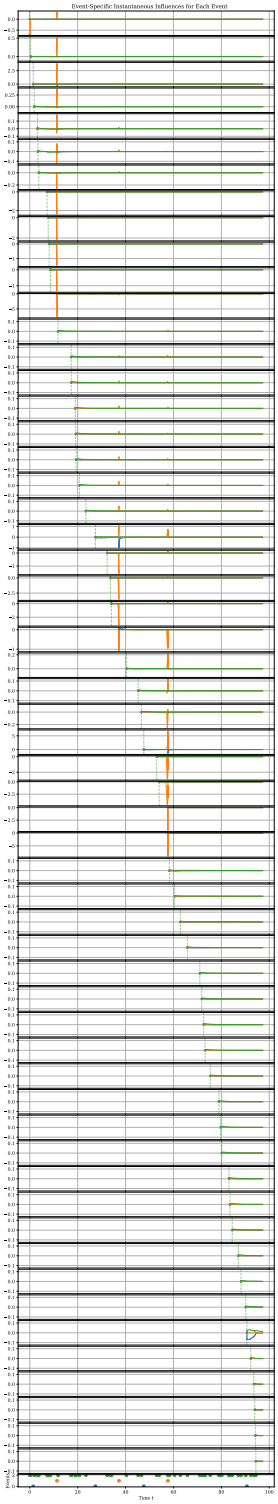


Figure 8: **Please view as a PDF to zoom in to details.** For the same sequence showcased in Figure 7, the individual $\mathrm{DF}\lambda_t^{(i)}$ values over time for a particle are displayed with the top-most plot showing the first event, $i = 1$, and second-to-last showing the last event, $i = N_T$. The color of the dashed line in each subplot indicates the mark of the particle being displayed. The colors of the solid lines indicate the instantaneous influence that particle has over future events of that color.

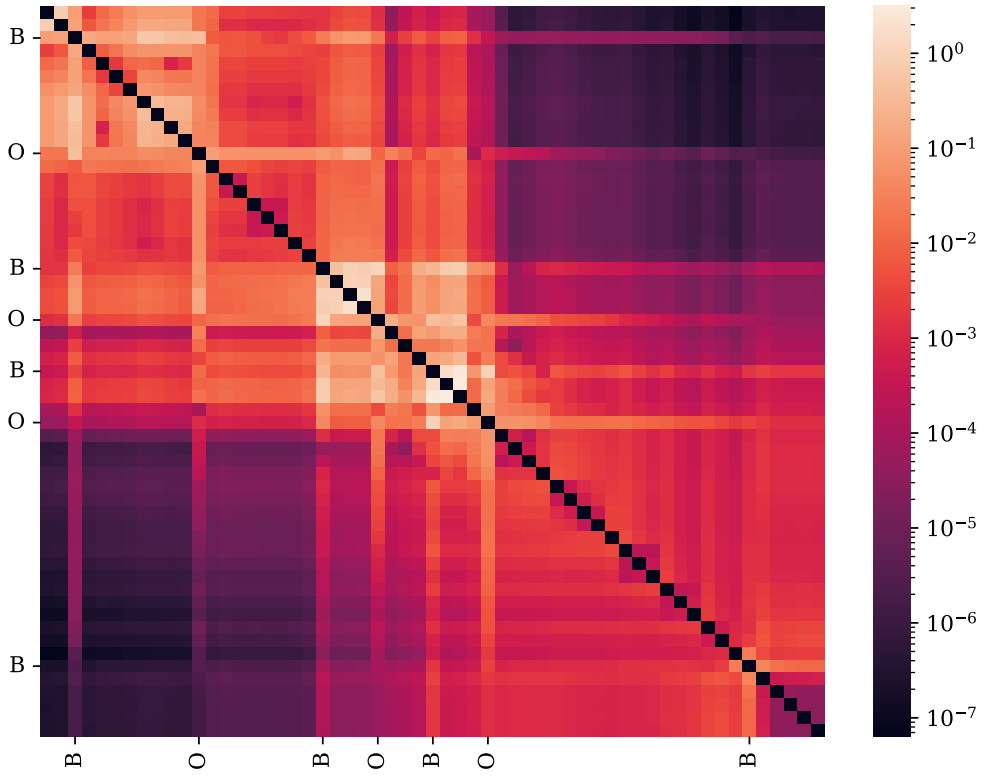


Figure 9: Fixed-dynamics joint-particle output nonadditivity for the sequence in Figure 7. Events are ordered chronologically on both axes; B and O denote call and response marks. Larger values indicate greater departure of the joint-removal contrast from the sum of single-removal contrasts after the nonlinear readout.

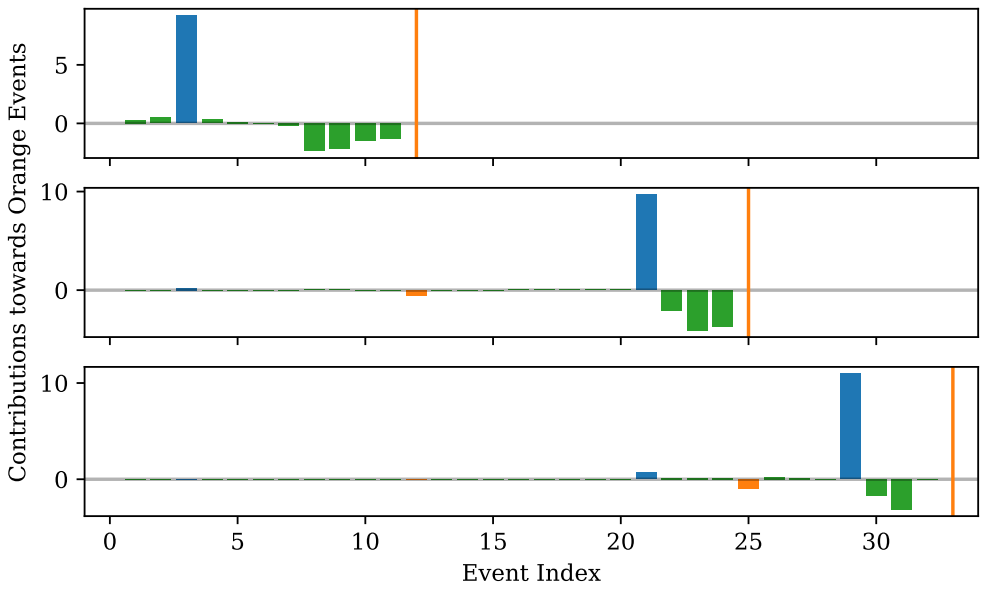


Figure 10: Conditional particle contrasts for three observed response (orange) marks in the sequence from Figure 7. Each bar is the change in response-mark intensity from removing a prior particle while retaining the observed-history transition path; colors denote source marks.

D.3 Conditional Particles, Event Deletion, and Generated Transitions

The component-swap experiment below separates the remembered event's impulse $\boldsymbol{\alpha}\mathbf{e}_{k_i}$ from the history-generated basis and decay parameters $\{\mathbf{V}_j, \mathbf{D}_j\}$ that determine the interval transitions $\mathbf{V}_j e^{\mathbf{D}_j \Delta_j} \mathbf{V}_j^*$. An impulse-only swap substitutes the impulse associated with the replacement mark while retaining the original transitions and all other event impulses. A transition-only swap uses transitions generated from the mark-swapped history while retaining the original impulses. Both swaps keep the event times, learned readout $\mathbf{W}$, and background $\boldsymbol{\mu}$ fixed.

We evaluated one fitted delayed-repeat checkpoint on every eligible held-out response: 5,489 responses in 1,985 of 2,000 test sequences. Eligibility required a green trigger, two preceding blue/orange events, a response repeating the mark immediately before the trigger, and a trigger-to-response delay in $[9.7, 10.3]$. At the observed response time, immediately before consuming the response, let F_0 be the intensity of its observed mark. Let F_{-p} remove a source particle under the original transitions, and let F_{-e} delete its event, preserve all other absolute times, reconstruct gaps, and rerun the model. Then $P = F_0 - F_{-p}$ is the conditional particle contrast, $T = F_0 - F_{-e}$ is deletion sensitivity, and $G = F_{-p} - F_{-e}$ gives the order-dependent identity $T = P + G$. The residual G includes changed later GRU gap inputs, transitions, and particle propagation; it is not a unique hypernetwork-pathway effect.

The trigger particle ranked among the top two prior particles by absolute target-mark contrast in 96.78% of eligible responses (95% sequence-cluster interval [96.31, 97.24]). This rank has a recency advantage because the trigger immediately precedes the response. When deleting the event holding the remembered mark exposed a different earlier mark, the preferred mark switched in 2,663/2,663 cases. When the exposed mark was the same, it remained preferred in 2,826/2,826 cases. Deletion merges adjacent gaps and thus changes later GRU inputs, in addition to removing the event impulse.

To hold event times and count fixed, we swapped only the remembered mark and independently substituted its impulse and the generated interval transitions for the 2,663 different-mark cases. The original prediction preferred the original mark in 2,663/2,663 cases. Swapping only the impulse switched none, whereas swapping only the transitions or swapping both switched all 2,663. The median replacement-minus-original intensity margins were -3.409 $[-3.456, -3.349]$ for the original prediction, -3.318 $[-3.374, -3.255]$ for the impulse-only swap, $+3.289$ $[+3.236, +3.331]$ for the transition-only swap, and $+3.373$ $[+3.316, +3.441]$ for both. Thus the impulse-only swap had a small but nonzero quantitative effect; changing the transitions sufficed to switch the preferred mark for this checkpoint without merging gaps. The crossed component predictions are internal hybrids, not realizable histories or causal interventions. These intervals resample held-out sequences for one fitted checkpoint, not training seeds. The endpoint is response-time target-mark intensity, not a demonstration of the trigger's timing mechanism or of effects on real-world data.

Table 7: Target-mark intensity remaining after each probe, as a median percentage of original response-time intensity, for one checkpoint. Brackets give 95% sequence-cluster bootstrap intervals (5,000 replicates).

Probe	Responses	Median remaining [95% interval]
Remove trigger particle	5,489	41.90% [41.36, 42.44]
Delete trigger and rerun	5,489	3.864% [3.818, 3.906]
Remove remembered-mark particle, different earlier mark	2,663	102.60% [102.42, 102.79]
Delete remembered-mark event, different earlier mark	2,663	0.00486% [0.00468, 0.00515]
Delete remembered-mark event, same earlier mark	2,826	100.19% [100.07, 100.34]