

IN-CONTEXT LEARNING OF TEMPORAL POINT PROCESSES WITH FOUNDATION INFERENCE MODELS

David Berghaus^{1,2}, Patrick Seifner^{1,3}, Kostadin Cvejovski⁴,

César Ojeda⁵ & Ramsés J. Sánchez^{1,2,3}

Lamarr Institute¹, Fraunhofer IAIS², University of Bonn³, JetBrains Research⁴

& University of Potsdam⁵

david.berghaus@iais.fraunhofer.de

ABSTRACT

Modeling multi-type event sequences with marked temporal point processes (MTPPs) provides a principled framework for uncovering governing dynamical rules and predicting future events. Current neural approaches to MTPP inference typically require training separate, specialized models for each target system. We pursue a fundamentally different strategy: leveraging amortized inference and in-context learning, we pretrain a deep neural network to infer, *in-context*, the conditional intensity functions of event histories from a context consisting of sets of event sequences. Pretraining is performed on a large synthetic dataset of MTPPs sampled from a broad distribution over point processes. Once pretrained, our Foundation Inference Model for Point Processes (FIM-PP) can estimate MTPPs from real-world data without additional training, or be rapidly finetuned to specific target systems. Experiments show that FIM-PP matches the performance of specialized models on multi-event prediction across common benchmark datasets.

Our pretrained model, repository, and tutorials are available online¹

1 INTRODUCTION

The mathematical modeling of asynchronous, irregular event sequences has long occupied a distinctive place in the machine learning community. Temporal point processes provide the canonical framework for modeling neural dynamics (Truccolo et al., 2005; Linderman & Adams, 2014), and they serve as the de facto tool for describing a wide range of internet phenomena, including retweeting, posting, and information cascades (Zhao et al., 2015; Cvejovski et al., 2020). Their ability to encode fine-grained temporal structure, together with their capacity to reveal causal interactions between events in an interpretable manner, has made them indispensable not only in neuroscience and social media, but also in finance (Ait-Sahalia et al., 2015) and epidemiology (Chiang et al., 2022). Despite this centrality, the development of foundation models has followed a different trajectory. Large-scale pretraining first emerged in natural language processing, enabled by massive internet corpora, and has only recently been extended to dynamical systems, with recent work addressing ODEs (d’Ascoli et al., 2024; Mauel et al., 2025; 2026), MJPs (Berghaus et al., 2024), SDEs (Seifner et al., 2025a), and even specific applications in pharmacology (Marin et al., 2025). It is therefore ironic that event data — the very modality underlying the internet activity that made text-based pretraining possible — has not yet given rise to a corresponding foundation model for point processes. The present work takes a first step toward filling this gap by developing a foundation model explicitly designed for temporal point processes.

Marked Temporal Point Processes (MTPPs) (Daley & Vere-Jones, 2007; Rasmussen, 2018) are stochastic processes consisting of ordered occurrence times, each accompanied by a categorical mark specifying its type. Formally, the objective is to specify the conditional distribution of the next event time and mark given the history of the process up to the current time. The extensive point process literature explores different ways of encoding event histories and specifying the stochastic mechanism that governs new arrivals and their marks (Lin et al., 2024). A common approach is

¹<https://fim4science.github.io/OpenFIM/intro.html>

to represent this distribution through a *conditional intensity function*, which describes the instantaneous rate at which events of different types occur given the history. Traditionally, models such as the Hawkes process (Hawkes, 1971) define this conditional intensity as a superposition of self-exciting effects from past events. Forecasting is then carried out recursively using Ogata’s thinning algorithm, applied to the conditional intensity. Building on this cornerstone, more recent work has extended Hawkes processes by parameterizing event histories with neural architectures (Mei & Eisner, 2017), including recurrent neural networks (Du et al., 2016), attention mechanisms (Zhang et al., 2020), and Transformers (Zuo et al., 2021). These models are typically trained either via maximum likelihood — often requiring expensive integral evaluations — or through generative approaches that bypass intensity modeling altogether and directly sample future events conditional on the past (Kerrigan et al., 2024; Zeng et al., 2024). However, a fundamental limitation of these approaches is their *lack of transferability*: each new dataset requires retraining from scratch, forcing the model to relearn representations for each distinct dynamical regime.

In contrast, modern approaches to dynamical systems increasingly prioritize *pretraining on synthetic data*, yielding general models that can learn dynamics *in-context*. This paradigm offers a crucial advantage: practitioners no longer need to train models *de novo* for every dataset, but can instead obtain accurate characterizations in a *zero-shot* manner. Within this family, two variants have emerged: Prior-fitted Networks (PFNs) and Foundation Inference Models (FIMs). PFNs train networks to approximate *predictive posterior distributions* in a sequence-to-sequence or context-to-sequence fashion (Müller et al., 2022; Hollmann et al., 2022; Müller et al., 2025). FIMs, by contrast, focus on directly *estimating the infinitesimal operators* of dynamical system (e.g., drift and diffusion functions for SDEs), thereby retaining a degree of interpretability (Berghaus et al., 2024; Seifner et al., 2025a;b; Mauel et al., 2026). Access to these operators enables the explicit study of physically relevant observables such as entropy production, stationary distributions, and attractors.

In the context of point processes, we adapt the FIM pretraining paradigm to MTPPs in three steps. First, we define a broad family of conditional intensity functions, thereby inducing a diverse prior over MTPPs. This prior captures assumptions about excitatory and inhibitory effects between events, as well as the interaction structure across event types. Second, we sample MTPPs from this prior, generate synthetic event sequences, and construct tuples consisting of context event sequences, event histories, and their corresponding intensities, thereby creating a meta-learning task that amortizes inference across heterogeneous dynamics. Third, we train a neural network to recover conditional intensities from observed context. We summarize our contributions as follows:

1. Introduce a synthetic data generation framework for sampling event sequences from a *broad prior distribution over MTPPs*, with randomized base intensities, kernels, and interaction types (i.e., excitatory, inhibitory, neutral). We empirically demonstrate that this construction encodes a strong prior, enabling models trained on it to *generalize* across both in-distribution processes and real-world event data.
2. Train the first transformer-based recognition model capable of estimating *in-context* the conditional intensity functions of MTPPs, where history representations serve as queries and the encoded sequence context provides the keys and values.
3. Show that the resulting model achieves strong *zero-shot* performance across synthetic benchmarks and multiple real-world datasets, and that it can be *rapidly finetuned on new event data*.

2 RELATED WORK

Here we provide a brief overview of temporal point processes. For detailed surveys and benchmarks of deep TPP models, including open challenges in history encoding, conditional intensity design, relational discovery, and learning strategies, see e.g., Lin et al. (2024) and Xue et al. (2024).

While the mathematical theory of point processes is extensive (Daley & Vere-Jones, 2007; Kingman, 1992), work on temporal point processes (TPPs) in machine learning has crystallized around two central questions: (i) how should representations of past events be constructed, and (ii) how should the future be modeled (Lin et al., 2024). Early approaches, epitomized by the Hawkes process, addressed both questions using linear self-exciting kernels. A natural extension is the neural Hawkes process (Mei & Eisner, 2017), along with related recurrent formulations (Du et al., 2016), which

rely on neural representations of past events, trained via likelihood maximization, and model future events using the thinning algorithm. Later work introduced more expressive architectures. Attention mechanisms (Zuo et al., 2021; Zhang et al., 2020; Yang et al., 2021) extend the memory horizon of TPPs, though at the cost of higher computational demand. Neural ODEs (Chen et al., 2018) have also been incorporated to better capture the irregular timing of events in latent representations (Song et al., 2024; Kidger et al., 2020).

To improve predictive accuracy over long horizons, different decompositions of the likelihood for future arrivals have been proposed (Rasmussen, 2018; Panos, 2024; Deshpande et al., 2021; Draxler et al., 2025). These works highlight the limitations of intensity-based inference, particularly when relying on thinning algorithms. Such limitations have motivated a shift toward generative models, which typically sample entire sequences. Approaches include optimal transport (Xiao et al., 2017), diffusion models (Zeng et al., 2024; Lüdke et al., 2023), and flow-matching methods (Kerrigan et al., 2024), often trading accuracy for interpretability. In contrast, traditional machine learning methods (Rasmussen, 2013; Malem-Shinitski et al., 2022) emphasize interpretability from the outset. A key advantage of the Hawkes process is that its excitation graph makes causal structure explicit (Xu et al., 2016; Wu et al., 2024), which has been especially relevant in neuroscience (Linderman & Adams, 2014; Truccolo et al., 2005) and in finance, where Hawkes models often serve as hidden drivers of observed activity (Ait-Sahalia et al., 2015). Applications extend more broadly, for instance to dynamics of text (Cvejoski et al., 2020; 2021), social online activity (Zhao et al., 2015) and operations research (Ojeda et al., 2021).

While Liu & Quan (2024) fine-tuned a pre-trained LLM for point-processes (with modest success), to the best of our knowledge, no prior work has presented a foundation model for point-processes that can operate zero-shot without further training.

3 PRELIMINARIES

In this section, we recall the definition and basic properties of *marked temporal point processes* (Daley & Vere-Jones, 2007) and *Hawkes processes* (Hawkes, 1971; Laub et al., 2015). Additionally, we define the inference problem our proposed approach tackles.

Marked Temporal Point Processes: We consider *marked temporal point processes* (marked TPPs, or MTPPs) as simple point processes on $\mathbb{R}_+ \times \mathcal{K}$, where \mathcal{K} is a discrete and finite set of *marks*. The density f of a sequence of events $\mathcal{S} = \{(t_i, \kappa_i)\}_{i=1}^n$ in the interval $[0, T]$, w.l.o.g. ordered by their *time component* $t_i \in \mathbb{R}_+$, factors into *conditional densities*

$$f(\{(t_i, \kappa_i)\}_{i=1}^n) = \prod_{i=1}^n f((t_i, \kappa_i) | \mathcal{H}_{t_i}) = \prod_{i=1}^n f(t_i | \mathcal{H}_{t_i}) f(\kappa_i | t_i, \mathcal{H}_{t_i}), \quad (1)$$

where $\mathcal{H}_t = \{(t_i, \kappa_i) | t_i < t\} \subset \mathcal{S}$ is the *history strictly preceding* t . By the last equality of equation 1, MTPPs may be characterized by dependent densities of the *next-event time* $f(t | \mathcal{H}_t)$ and its *event mark* $f(\kappa | t, \mathcal{H}_t)$. MTPPs are commonly represented by their piece-wise continuous *conditional intensity function*

$$\lambda(t, \kappa | \mathcal{H}_t) = \frac{f(t | \mathcal{H}_t)}{1 - \int_{t'}^t f(s | \mathcal{H}_s) ds} f(\kappa | t, \mathcal{H}_t) = \lambda(t | \mathcal{H}_t) f(\kappa | t, \mathcal{H}_t), \quad (2)$$

where t' is the last event time in \mathcal{H}_t , or $t' = 0$ if $\mathcal{H}_t = \emptyset$. The conditional intensity function may be interpreted of the *instantaneous rate* of mark κ occurring at t , conditioned of the history up to time t_i . Reversely, any such function λ , satisfying some mild conditions, defines the density of an MTPP on a set of events in an interval $[0, T]$ by

$$f(\{(t_i, \kappa_i)\}_{i=1}^n) = \left[\prod_{i=1}^n \lambda(t_i, \kappa_i | \mathcal{H}_{t_i}) \right] \exp\left(-\int_0^T \lambda(s | \mathcal{H}_s) ds\right). \quad (3)$$

Collection of TPPs: A TPP is just an MTPP with a single mark. An MTPP can be viewed as a *collection of TPPs* per mark, interdependent through a *joined history*. Indeed, given an MTPP as above, the conditional intensity $\lambda_\kappa(t | \mathcal{H}_t) = \lambda(t | \mathcal{H}_t) f(\kappa | t, \mathcal{H}_t)$ defines the *marginal TPP*

for mark $\kappa \in \mathcal{K}$, that may depend on other marks via \mathcal{H}_t . Conversely, a collection of TPPs with conditional intensity λ_κ per $\kappa \in \mathcal{K}$ can be *joined* to an MTPP. Using

$$\lambda(t | \mathcal{H}_t) = \sum_{\kappa \in \mathcal{K}} \lambda_\kappa(t | \mathcal{H}_t) \quad \text{and} \quad f(\kappa | t, \mathcal{H}_t) = \frac{\lambda_\kappa(t | \mathcal{H}_t)}{\lambda(t | \mathcal{H}_t)}, \quad (4)$$

in Eq. 2 defines the conditional intensity function $\lambda(t, \kappa | \mathcal{H}_t)$ of an MTPP. In fact, $\lambda(t, \kappa | \mathcal{H}_t) = \lambda_\kappa(t | \mathcal{H}_t)$. In contrast to some other neural methods (Du et al., 2016), which estimate $\lambda(t | \mathcal{H}_t)$ and $f(\kappa | t, \mathcal{H}_t)$, we design our model to parametrize TPPs per mark, conditioned on the joined history of all marks.

Hawkes Processes: A *Hawkes MTPP* with marks \mathcal{K} is defined by the conditional intensity

$$\lambda(t, \kappa | \mathcal{H}_t) = \max \left(0, \mu_\kappa(t) + \sum_{(t', \kappa') \in \mathcal{H}_t} \gamma_{\kappa\kappa'}(t - t') \right), \quad (5)$$

where $\{\mu_\kappa\}_{\kappa \in \mathcal{K}}$ is a set of *time-dependent base intensity functions*, and $\{\gamma_{\kappa\kappa'}\}_{\kappa, \kappa' \in \mathcal{K}}$ is a set of *interaction kernels*, specifying the influence of mark κ' on mark κ . If $\gamma_{\kappa\kappa'}$ is positive, the influence of κ' on κ is called *excitatory* or *exciting*, otherwise it is called *inhibitory* or *limiting*.

Simulation: We use *Ogata’s modified thinning algorithm* (Ogata, 1981) to generate synthetic training data from MTPP, and to simulate processes inferred by our model.

Inference Problem: Let $\mathcal{C} = \{\mathcal{S}^j\}_{j=1}^m$ be a collection of m event sequences $\mathcal{S}^j = \{(t_i^j, \kappa_i^j)\}_{i=1}^{n_j}$ observed from some system. Our objective is to *predict* or *simulate* the *next event* and estimate the *likelihood* of a (previously unseen) sequence \mathcal{S} , assuming an MTPP model. Previous neural methods *train an autoregressive encoding network on \mathcal{C}* that compresses the history \mathcal{H}_t of \mathcal{S} into some embedding \mathbf{h}_t for a neural estimate $\hat{\lambda}(t, \kappa | \mathbf{h}_t)$ of the conditional intensity. In contrast, we *pretrain* a deep neural network model to estimate *in-context* $\hat{\lambda}$, given a history of events, from a collection of context event sequences. Once trained, the model can be applied to *any* \mathcal{C} and \mathcal{H}_t , *without any further training*.

4 FOUNDATION INFERENCE MODELS FOR POINT PROCESSES

In this section, we present a novel *in-context learning method* for the MTPP intensity inference problem. In a two-step approach, we first generate a *large set of marked event sequences* from parametrized MTPPs, sampled from a *broad distribution* over MTPPs. This yields train data for a neural network recognition model, trained to estimate the *underlying known, ground-truth* intensity functions. Such *pretrained* inference model can be applied directly to real-world problems, or *swiftly finetuned* for improved performance.

4.1 SYNTHETIC DATASET GENERATION

To construct a synthetic dataset of MTPPs, we define a distribution over MTPPs via a distribution over conditional intensity functions of the form

$$\lambda(t, \kappa | \mathcal{H}_t) = \max \left(0, \mu_\kappa(t) + \sum_{(t', \kappa') \in \mathcal{H}_t} z_{\kappa\kappa'} \gamma_{\kappa\kappa'}(t - t') \right). \quad (6)$$

Given a sample from this distribution, we simulate a large set of marked event sequences and record the corresponding conditional intensities for training.

We use Eq. 6 to parameterize several classes of processes:

- The classical Hawkes process, for which μ_κ is constant in time and $\gamma_{\kappa\kappa'}(t)$ is an exponential function;
- The Poisson process, for which μ_κ is constant in time and $\gamma_{\kappa\kappa'} = 0$;
- Periodic processes, for which $\mu_\kappa(t)$ is a positive sinusoidal function;

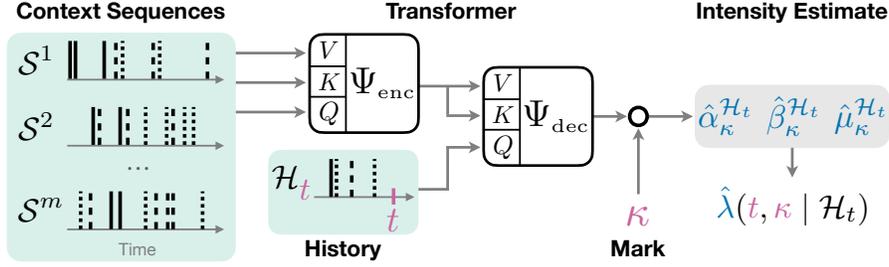


Figure 1: Schematic representation of FIM-PP. A *context* of marked event sequences \mathcal{S}^j is encoded by a self-attentive *transformer encoder*. The result is further processed by a *transformer decoder*, using a *history* \mathcal{H}_t of marked events before time t as queries. The final embedding is joined with an encoding of *mark* κ . The results is projected to a set of parameters that determine the value of the *conditional intensity function* $\hat{\lambda}$ evaluated at (t, κ) .

- Processes with high initial excitation, for which we choose $\mu_k(t)$ to follow a Gamma distribution;
- Processes with non-monotonic, shifted kernels, for which we choose $\gamma_{\kappa\kappa'}(t)$ to follow a Rayleigh distribution.

A complete overview of the process families and their hyperparameters is provided in Table 6 in the Appendix. Furthermore, Figure 7 in the Appendix reports summary statistics of our simulated pretraining distribution, and confirms that it is broad enough to cover the real-world datasets depicted in Figures 8 to 12, also in the Appendix.

For each process, we randomly sample $z_{\kappa\kappa'} \in \{-1, 0, 1\}$ for $\kappa, \kappa' \in \mathcal{K}$ to cover *excitatory* ($z_{\kappa\kappa'} = 1$), *inhibitory* ($z_{\kappa\kappa'} = -1$) and *non-influencing* ($z_{\kappa\kappa'} = 0$) interactions.

4.2 FOUNDATION INFERENCE MODEL ARCHITECTURE

We now present the architecture of FIM-PP, a pretrained deep neural network for inference of MTPPs from sets $\mathcal{C} = \{\mathcal{S}^j\}_{j=1}^m$ of marked event sequences $\mathcal{S}^j = \{(t_i^j, \kappa_i^j)\}_{i=1}^{n_j}$. FIM-PP processes the *context sequences* \mathcal{C} and estimates the conditional intensity function $\hat{\lambda}$ of an MTPP that describes the observed dynamics. Following previous intensity-based methods (Zhang et al., 2020; Zuo et al., 2021), $\hat{\lambda}$ is implemented by a flexible parametrized function family $\hat{\lambda}(\cdot, \kappa | \mathcal{H}_\cdot)$ for all $\kappa \in \mathcal{K}$. FIM-PP estimates its parameters by encoding the history $\mathcal{H}_t = \{(t_i^{\text{hist}}, \kappa_i^{\text{hist}})\}_{i=1}^{n_{\text{hist}}}$ before time $t > t_{n_{\text{hist}}}^{\text{hist}}$, *subject to the processed context sequences*. Figure 1 depicts a schematic representation of this approach.

To cover applications in different time scales, FIM-PP instance normalizes its inputs and renormalizes $\hat{\lambda}$ accordingly. Appendix C provides the details. Once trained, FIM-PP can be applied for all counts of marks $|\mathcal{K}|$ up to some fixed upper bound, similar to in-context methods in other domains (d’Ascoli et al., 2024; Berghaus et al., 2024).

We denote linear projections by ϕ , feed-forward neural networks by Φ , attention layers with residual connections by ψ , transformer encoders by Ψ_{enc} and decoders by Ψ_{dec} . Let $E \in \mathbb{N}$ denote the model’s embedding dimension.

Context Encoding: To encode \mathcal{C} , we combine encodings of individual sequences \mathcal{S}^j . Recognizing the importance of inter-observation times for the inference problem, we consider $\Delta t_i^j = t_i^j - t_{i-1}^j$ as an additional feature, identifying $t_0^j = 0$. To encode \mathcal{S}^j , we first embed the features $(t_i^j, \kappa_i^j, \Delta t_i^j)$ of the i -th event in sequence j into embeddings

$$\mathbf{u}_i^j = \phi_t(t_i^j) + \phi_\kappa(\kappa_i^j) + \phi_{\Delta t}(\Delta t_i^j) \in \mathbb{R}^E. \quad (7)$$

Sinusoidal output activations from Shukla & Marlin (2020) enhance the networks ϕ_t and $\phi_{\Delta t}$. Let $\mathbf{S}^j = [\mathbf{u}_1^j, \dots, \mathbf{u}_{n_j}^j] \in \mathbb{R}^{n_j \times E}$ denote the matrix of embedding of sequence \mathcal{S}^j . We extract a

context sequence embedding $\mathbf{c}_j \in \mathbb{R}^E$ by applying a transformer encoder $\tilde{\mathbf{S}}^j = \Psi_{\text{enc}}^{\text{cont}}(\mathbf{S}^j) \in \mathbb{R}^{n_j \times E}$, followed by fixed-query attention

$$\mathbf{c}_j = \psi^{\text{cont}}(\mathbf{q}^{\text{cont}}, \tilde{\mathbf{S}}^j, \tilde{\mathbf{S}}^j) \in \mathbb{R}^E, \quad (8)$$

where $\mathbf{q}^{\text{cont}} \in \mathbb{R}^E$ is a learnable query.

We emphasize that we use an hierarchical approach where every sequence gets processed independently by $\Psi_{\text{enc}}^{\text{cont}}$ and encoded into a single embedding \mathbf{c}_j . This is much more scalable compared to combining all events into a single long sequence. The embeddings of all sequences $\mathbf{C} = [\mathbf{c}_1, \dots, \mathbf{c}_m] \in \mathbb{R}^{m \times E}$ are finally combined by another transformer encoder $\tilde{\mathbf{C}} = \Psi_{\text{enc}}^{\text{comb}}(\mathbf{C}) \in \mathbb{R}^{m \times E}$.

Context-aware History Encoding: To encode the history \mathcal{H}_t of events prior to time $t > t_{n_{\text{hist}}}^{\text{hist}}$, we embed each tuple $(t_i^{\text{hist}}, \kappa_i^{\text{hist}}, \Delta t_i^{\text{hist}})$ into feature vectors $\mathbf{H} = [\mathbf{u}_1^{\text{hist}}, \dots, \mathbf{u}_{n_{\text{hist}}}^{\text{hist}}] \in \mathbb{R}^{n_{\text{hist}} \times E}$, reusing the networks from Eq. 7. These history embeddings serve as the queries of a transformer decoder $\Psi_{\text{dec}}^{\text{hist}}$, which attends to the context representation $\tilde{\mathbf{C}}$ (used as keys and values), yielding a unified encoding $\mathbf{h}_t^{\text{hist}}$ that integrates both history and context:

$$\mathbf{h}_t^{\text{hist}} = \Psi_{\text{dec}}^{\text{hist}}(\mathbf{H}, \tilde{\mathbf{C}}) \in \mathbb{R}^E. \quad (9)$$

Intensity Parametrization: To extract intensity functions for all marks from $\mathbf{h}_t^{\text{hist}}$, we concatenate $\mathbf{h}_t^{\text{hist}}$ with a (linear) encoding of κ' to form $\mathbf{v}_{\kappa'}^{\mathcal{H}_t} = [\mathbf{h}_t^{\text{hist}}, \phi_{\kappa'}^{\text{hist}}(\kappa')] \in \mathbb{R}^{2E}$, and project it to non-negative parameter estimates

$$\hat{\alpha}_{\kappa'}^{\mathcal{H}_t} = \Phi_{\alpha}(\mathbf{v}_{\kappa'}^{\mathcal{H}_t}) \in \mathbb{R}_+, \quad \hat{\beta}_{\kappa'}^{\mathcal{H}_t} = \Phi_{\beta}(\mathbf{v}_{\kappa'}^{\mathcal{H}_t}) \in \mathbb{R}_+ \quad \text{and} \quad \hat{\mu}_{\kappa'}^{\mathcal{H}_t} = \Phi_{\mu}(\mathbf{v}_{\kappa'}^{\mathcal{H}_t}) \in \mathbb{R}_+. \quad (10)$$

We enforce non-negativity via softplus output activations. These parameters define our neural conditional intensity estimate:²

$$\hat{\lambda}(t, \kappa' | \mathcal{H}_t) = \hat{\mu}_{\kappa'}^{\mathcal{H}_t} + (\hat{\alpha}_{\kappa'}^{\mathcal{H}_t} - \hat{\mu}_{\kappa'}^{\mathcal{H}_t}) \exp\left(-\hat{\beta}_{\kappa'}^{\mathcal{H}_t}(t - t_{n_{\text{hist}}}^{\text{hist}})\right). \quad (11)$$

This parametrization is both flexible and interpretable. Indeed, immediately after incorporating a new event into the history, the intensity jumps to $\hat{\alpha}_{\kappa'}^{\mathcal{H}_t}$. Over long inter-event intervals, the intensity relaxes toward $\hat{\mu}_{\kappa'}^{\mathcal{H}_t}$. The relaxation rate is governed by $\hat{\beta}_{\kappa'}^{\mathcal{H}_t}$.

Although this parametrization may appear tailored to Hawkes-type dynamics, its parameters $\hat{\mu}_{\kappa'}^{\mathcal{H}_t}$, $\hat{\alpha}_{\kappa'}^{\mathcal{H}_t}$, and $\hat{\beta}_{\kappa'}^{\mathcal{H}_t}$ are themselves history- and mark-dependent. As a result, the model can represent rich local intensity behaviors, including localized triggering patterns (e.g., Rayleigh- or power-law-like kernels) and time-dependent baseline intensities. We highlight this empirically in Section 5.

Training: To train FIM-PP on a set of sequences \mathcal{C}_{λ} from our synthetic pretraining data, we select a *target sequence* $\mathcal{T} \in \mathcal{C}_{\lambda}$ to provide a history of events and use remaining sequences $\mathcal{C}_{\lambda} \setminus \{\mathcal{T}\}$ as context. We subsample \mathcal{C}_{λ} , truncate sequences and vary the number of marks throughout training, which enables us to apply a pretrained FIM-PP in a wide range of (real-world) settings. Our train objective is the next-event negative log likelihood of the target sequence:

$$\mathcal{L}_{\text{NLL}} = \sum_{\kappa \in \mathcal{K}} \int_0^T \hat{\lambda}(s, \kappa | \mathcal{H}_s) ds - \sum_{(t, \kappa) \in \mathcal{T}} \hat{\lambda}(t, \kappa | \mathcal{H}_t). \quad (12)$$

We remark that we also experimented with a supervised sMAPE loss, the results of which can be found in the Appendix Tables, starting at Table 3. We found that this approach performs similarly well but is more computationally expensive and less flexible, which is why we use the NLL model in the main text. Appendix D discusses the training of FIM-PP in greater detail.

Finetuning: FIM-PP can be finetuned on the train split \mathcal{C} of an evaluation dataset, minimizing \mathcal{L}_{NLL} . For each iteration, a random sequence $\mathcal{T} \in \mathcal{C}$ in the train split is selected as the target sequence. The remaining sequences $\mathcal{C} \setminus \{\mathcal{T}\}$ serve as context. Finetuning progress is monitored by processing target sequences from the validation split, given the train split context.

²Note that the functional form of $\hat{\lambda}$ is similar to the conditional intensity in Zhang et al. (2020).

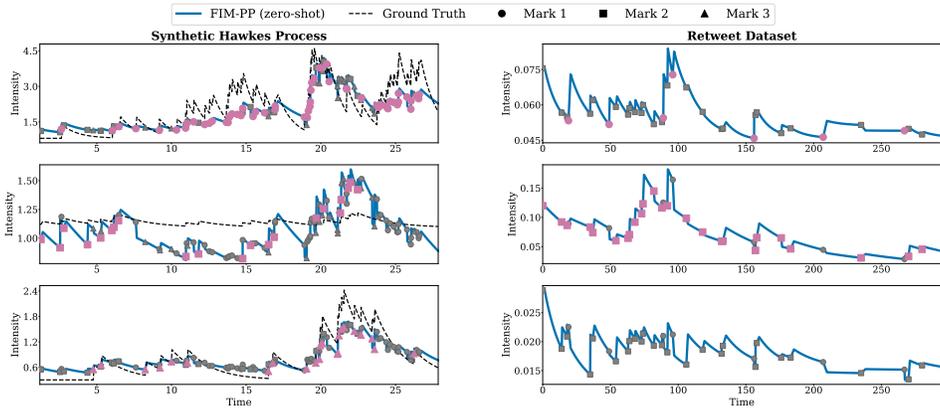


Figure 2: Example intensity estimates of FIM-PP on a synthetic Hawkes process with three marks, constant base intensity and exponential decaying kernels (left) and a real-world RETWEET dataset (right). Each row contains the intensity for one mark. Events of the same mark are colored magenta, while events for other marks are gray. For the Hawkes process, the model (blue line) estimate matches the ground-truth intensity level (black dashed line) closely. For the RETWEET data, FIM-PP estimates a mixture of many excitatory and a few inhibitory interactions.

5 EXPERIMENTS

In this section, we repeat the experiments by Zeng et al. (2024), who introduced CDiff, a recent state-of-the-art diffusion-based marked event sequence forecasting model. They compare their method against a range of intensity-based and intensity-free baselines, on common benchmark datasets, evaluated on standard metrics. Moreover, they made their code and exact evaluation dataset splits available³, which allows us to directly compare against their results. In the following, we recall their experimental setup, describe the pretraining and application of FIM-PP, before presenting and analyzing our results.

5.1 EXPERIMENTAL SETUP

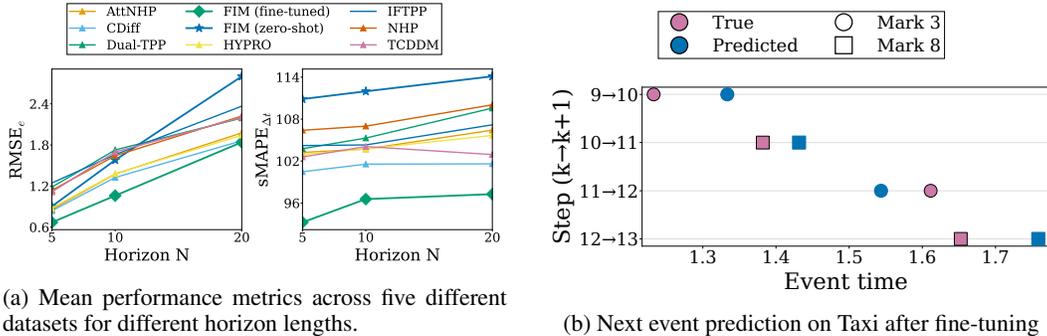
Prediction Task: Given a sequence of events $\mathcal{S} = \{(t_i, \kappa_i)\}_{i=1}^n$, the task is to predict the *next* $N \in \mathbb{N}$ events following \mathcal{S} , where N is the *prediction horizon length*. We denote the ground-truth continuation by $\mathcal{S}_{\text{final}}^N$ and the model prediction by $\hat{\mathcal{S}}_{\text{final}}^N$. We refer to the case $N = 1$ as *next-event prediction* and to $N > 1$ as *multi-event prediction*.

Evaluation Metrics: We evaluate predictions by comparing $\hat{\mathcal{S}}_{\text{final}}^N$ with $\mathcal{S}_{\text{final}}^N$ across five metrics. For $N > 1$, we report the Optimal Transport Distance (OTD) (Mei et al., 2019); event count error (RMSE_e), comparing the number of predicted and true events per mark; and two standard regression metrics on waiting times: RMSE_{Δt} and sMAPE_{Δt}. For the special case $N = 1$, OTD and RMSE_e are not applicable, and we instead report next-event mark prediction accuracy (Acc). Formal definitions of all metrics are provided in Appendix F.

Evaluation Data: We benchmark on five widely used real-world datasets: TAXI, TAobao, STACK-OVERFLOW, AMAZON, and RETWEET. These datasets vary in the number of marks, sequence lengths, and event counts, making them a strong testbed for evaluating the broad applicability of FIM-PP. We use the preprocessing and train/test/validation splits of Zeng et al. (2024). Appendix E contains further details, including dataset statistics and original sources.

Baselines: We compare FIM-PP against methods falling into two categories: models that learn joint distributions over multiple events, and autoregressive approaches such as FIM-PP. The first category includes Dual-TPP (Deshpande et al., 2021), HYPRO (Xue et al., 2022), and the Cross-diffusion Model (CDiff) (Zeng et al., 2024). The second category further splits into intensity-based and intensity-free approaches. Intensity-based baselines are the Neural Hawkes Process (NHP) (Mei

³<https://github.com/networkslab/cdiff>



(a) Mean performance metrics across five different datasets for different horizon lengths.

(b) Next event prediction on Taxi after fine-tuning

Figure 3: (a) shows that FIM-PP (zs) is competitive but slightly worse than the baseline models. FIM-PP (f) however performs best among all horizon lengths. (b) shows that FIM-PP (f) also reliably captures patterns in the Taxi dataset.

& Eisner, 2017), and the Attentive Neural Hawkes Process (A-NHP) (Mei et al., 2022). Intensity-free baselines are the Intensity-Free Temporal Point Process (IFTTP) (Shchur et al., 2020), and the Temporal Conditional Diffusion Denoising Model (TCDDM) (Lin et al., 2022).

5.2 PRETRAINING, FINE-TUNING AND INTENSITY PREDICTIONS

Pretraining. We pretrain a single FIM-PP on a synthetic dataset containing 14.4M events, simulated from 72K point processes of diverse kernels, and sparsity levels, and varying number of marks, sequences, and events. Appendix B contains the details. The model has 16M parameters and supports up to $|\mathcal{K}| = 22$ marks to cover all evaluation datasets. Further pretraining details are provided in Appendix D.

Finetuning. In *zero-shot mode*, we apply the pretrained model directly to all evaluation datasets, and label the results by FIM-PP (zs). We also experiment with *finetuning* FIM-PP on the train split of all evaluation datasets, and label these results by FIM-PP (f). FIM-PP utilizes up to 2000 sequences from the train split of an evaluation dataset as context, limited only by the maximum number of sequences seen during training. The ablations in Figure 6 also indicate that typically much less than 2000 context paths would be sufficient. The sequences in the test split are given as context to FIM-PP. Finally, for multi-event prediction, FIM-PP simulates events autoregressively, similar to other (intensity-based) baselines.

Due to the relatively small parameter size of FIM-PP and the pre-trained prior, fine-tuning can be performed quickly. For all datasets, fine-tuning was achieved in a few minutes, requiring not more than 11 GB of memory. This means that fine-tuning FIM-PP does not take longer than training the baseline models from scratch, which reportedly takes up to 4 hours (Xue et al., 2024, Sec. D.1). Fine-tuning is therefore feasible for users with small computational resources. Further speed-ups might be possible with LoRA (Hu et al., 2022).

Intensity Predictions. Figure 2 shows intensities inferred by FIM-PP in zero-shot mode, both from an unseen synthetic Hawkes process (in-distribution generalization), and from the RETWEET dataset (out-of-distribution generalization). Moreover, Figure 4 in the Appendix illustrates the performance of FIM-PP (zs) on various synthetic datasets and emphasizes that FIM-PP (zs) also generalizes well for processes with powerlaw kernels, a kernel type that was not present in the pre-training dataset. This confirms that our parameterization is general enough. In what follows, we quantitatively evaluate FIM-PP.

5.3 MULTI-EVENT PREDICTION

Table 1 reports OTD and sMAPE_{Δt} results for $N = 20$ and four datasets. Remarkably, FIM-PP achieves competitive performance in *zero-shot mode*, matching or surpassing specialized baselines on TAXI and RETWEET data. This shows that, solely from pretraining on synthetic data, the model

Table 1: Performance on four real-world datasets, predicting $N = 20$ events. Results for baseline methods were extracted from Zeng et al. (2024). We report mean and standard deviation over 10 trials for two metrics. Best results are bold.

| Method | TAXI | | STACKOVERFLOW | | AMAZON | | RETWEET | |
|-------------|-------------------------|-----------------------|-------------------------|-------------------------|-----------------------|-----------------------|-------------------------|-------------------------|
| | OTD | sMAPE $_{\Delta t}$ | OTD | sMAPE $_{\Delta t}$ | OTD | sMAPE $_{\Delta t}$ | OTD | sMAPE $_{\Delta t}$ |
| HYPRO | 21.60 \pm 0.20 | 93.8 \pm 0.4 | 42.40 \pm 0.20 | 111.00 \pm 0.60 | 38.6 \pm 0.5 | 82.5 \pm 0.8 | 61.03 \pm 0.09 | 106.11 \pm 1.51 |
| Dual-TPP | 24.48 \pm 0.38 | 95.2 \pm 0.2 | 41.75 \pm 0.20 | 117.58 \pm 0.42 | 42.6 \pm 0.7 | 86.5 \pm 2.0 | 61.10 \pm 0.10 | 106.90 \pm 1.29 |
| A-NHP | 24.76 \pm 0.22 | 97.4 \pm 0.4 | 42.59 \pm 0.41 | 108.54 \pm 0.53 | 39.5 \pm 0.3 | 84.3 \pm 1.8 | 60.63 \pm 0.10 | 107.23 \pm 1.29 |
| NHP | 25.11 \pm 0.27 | 96.5 \pm 0.5 | 43.79 \pm 0.15 | 116.95 \pm 0.40 | 42.6 \pm 0.3 | 92.1 \pm 1.6 | 60.95 \pm 0.08 | 107.08 \pm 1.40 |
| IFTPP | 24.05 \pm 0.61 | 95.7 \pm 0.8 | 46.28 \pm 0.89 | 115.12 \pm 0.63 | 43.8 \pm 0.2 | 90.9 \pm 1.6 | 61.72 \pm 0.15 | 106.71 \pm 1.62 |
| TCDDM | 22.15 \pm 0.53 | 90.6 \pm 0.6 | 42.13 \pm 0.59 | 107.66 \pm 0.93 | 42.2 \pm 0.2 | 83.8 \pm 1.5 | 60.50 \pm 0.09 | 106.05 \pm 0.61 |
| CDiff | 21.01 \pm 0.16 | 88.0 \pm 0.2 | 41.25 \pm 1.40 | 106.18 \pm 0.34 | 37.7 \pm 0.2 | 82.0 \pm 1.9 | 60.66 \pm 0.10 | 106.18 \pm 1.12 |
| FIM-PP (zs) | 23.15 \pm 0.07 | 76.8 \pm 0.4 | 49.26 \pm 0.06 | 96.36 \pm 0.05 | 46.2 \pm 0.1 | 128.6 \pm 0.4 | 60.24 \pm 0.16 | 99.07 \pm 0.39 |
| FIM-PP (f) | 17.91 \pm 0.12 | 76.8 \pm 0.5 | 39.80 \pm 0.04 | 88.25 \pm 0.19 | 37.2 \pm 0.1 | 81.2 \pm 0.1 | 59.44 \pm 0.08 | 87.59 \pm 0.17 |

can translate contextual patterns into accurate multi-event predictions, without any further training or supervision.

The same table also demonstrates the effectiveness of finetuning. The finetuned FIM-PP (f) consistently outperforms both FIM-PP (zs) and all baselines, across the four datasets and nearly all metrics. Additional experiments with shorter horizons ($N = 10, 5$) and alternative metrics (RMSE $_e$, RMSE $_{\Delta t}$) are reported in Appendix A, providing a complementary view.

Figure 3a summarizes performance across horizon lengths by averaging results over all datasets. In aggregate, FIM-PP (zs) performs on par with the baselines, whereas FIM-PP (f) consistently outperforms them, in agreement with our previous analysis. We attribute the effectiveness of finetuning to two factors: (i) the strong prior encoded in the model weights through pretraining on our synthetic distribution, which provides a favorable initialization for finetuning; and (ii) the flexibility of the foundation model architecture, which enables *direct access* to patterns in the training split *at evaluation time*, since these patterns can be retrieved from the context. The first point is supported by Figure 5 in the Appendix, which shows that the pretrained model converges faster and achieves better final performance than the same architecture trained from scratch.

5.4 NEXT-EVENT PREDICTION

The *next-event prediction* task is a special case of multi-event prediction, but it differs in nature. Whereas multi-event prediction requires estimating the *distribution* over a set of future events, next-event prediction focuses on accurately forecasting a *single* event. This distinction is also reflected in the evaluation metrics (see Appendix F)⁴.

Table 2 reports next-event prediction results ($N = 1$) on two real-world datasets. FIM-PP (zs) performs well on event-time prediction for TAXI, but struggles with mark accuracy on TAXI and with both event-time and mark accuracy on TAOBAO. These difficulties can be explained by dataset-specific patterns: sequences in TAXI often alternate consistently between two marks, a pattern unlikely to appear in our point process prior. In contrast, TAOBAO is heavily dominated by a single mark and occasionally exhibits long waiting times. Again, patterns not covered by our pretraining distribution.

Compared to the baselines, which easily incorporate such patterns during training, FIM-PP (zs) cannot reproduce them accurately from the context alone. Appendix G further analyzes these out-of-distribution patterns.

When finetuned, however, FIM-PP can adapt to (some) of these characteristics. Table 2 shows substantial improvements in mark prediction accuracy (Acc) after finetuning, and Figure 3 illustrates that FIM-PP (f) successfully recovers the alternating pattern in the TAXI dataset. Nevertheless,

⁴For instance, mark accuracy (Acc) targets the correctness of one event, while RMSE $_e$ compares histograms over multiple events.

Table 2: Next-event prediction performance on two real-world datasets, displayed with mean and standard deviation over 10 trials. Results for baseline methods were extracted from Zeng et al. (2024) (we remark that they did not report next-event prediction results for the other datasets). Best results are bold.

| Method | TAXI | | | TAOBAO | | |
|-------------|------------------------|------------------------|-------------------------|------------------------|------------------------|--------------------------|
| | RMSE $_{\Delta t}$ | Acc | sMAPE $_{\Delta t}$ | RMSE $_{\Delta t}$ | Acc | sMAPE $_{\Delta t}$ |
| A-NHP | 0.32 ± 0.00 | 0.91 ± 0.01 | 85.13 ± 0.26 | 0.53 ± 0.00 | 0.47 ± 0.01 | 129.13 ± 1.35 |
| Dual-TPP | 0.34 ± 0.01 | 0.91 ± 0.01 | 89.12 ± 0.75 | 0.53 ± 0.01 | 0.47 ± 0.02 | 131.43 ± 1.99 |
| NHP | 0.34 ± 0.01 | 0.91 ± 0.01 | 90.63 ± 0.61 | 0.53 ± 0.00 | 0.46 ± 0.01 | 133.69 ± 2.25 |
| IFTPP | 0.38 ± 0.01 | 0.90 ± 0.01 | 90.03 ± 0.47 | 0.53 ± 0.01 | 0.45 ± 0.01 | 126.01 ± 1.48 |
| CDiff | 0.34 ± 0.01 | 0.91 ± 0.00 | 87.12 ± 0.61 | 0.52 ± 0.01 | 0.48 ± 0.00 | 127.12 ± 1.36 |
| FIM-PP (zs) | 0.34 ± 0.01 | 0.47 ± 0.03 | 98.03 ± 0.92 | 0.13 ± 0.00 | 0.52 ± 0.01 | 112.89 ± 0.66 |
| FIM-PP (f) | 0.45 ± 0.03 | 0.91 ± 0.00 | 85.91 ± 0.86 | 0.16 ± 0.00 | 0.60 ± 0.01 | 115.98 ± 0.72 |

We remark that the released source code of CDIFF Zeng et al. (2024) only predicts the last event per sequence for next event prediction. We believe that this is a bug because it also produced very unreasonable results. We hence report the performance for all events, not just the last.

FIM-PP (f) still lags behind the baselines in mark prediction accuracy (Acc). In Appendix G, we suggest broadening the synthetic pretraining distribution to better capture such distinctive patterns upfront.

6 CONCLUSIONS

In this work, we introduced FIM-PP, the first *Foundation Inference Model* for inferring marked temporal point processes (MTPPs) from real-world data. Our experiments show that a *single* FIM-PP, pretrained only on synthetic MTPP data, can already match the predictive performance of existing intensity-based MTPP methods in *zero-shot mode*, i.e., *without any further training*. *Finetuning* on target data further improves results within only a few iterations, enabling FIM-PP to *outperform competing methods on the majority of evaluated tasks*.

Limitations: Although our pretraining distribution is broad, it is not universal and therefore cannot capture many real-world patterns. As a result, zero-shot performance may degrade under distribution shift. FIM-PP is further constrained by the maximum number of marks $|\mathcal{K}|$ and the maximum sequence length used during training. When these limits are exceeded, the model may not fully exploit the available context.

Future Work: An important direction for future work is to *broaden the pretraining distribution* beyond the parametrization in Eq. 6, with the goal of capturing a wider range of data patterns in zero-shot mode, and providing an even stronger initialization for finetuning. In addition, *intensity-free* methods have recently shown strong predictive performance (Panos, 2024). We plan to investigate how such methods can be incorporated into our amortized in-context learning framework.

7 REPRODUCIBILITY STATEMENT

Our core methodology consists of two parts: *synthetically generated training data* and a *foundation inference model* for marked temporal point process. *Data generation* is described extensively in Section 4.1 and complemented by Appendix B, which covers the exact hyperparameters and design choices required to reproduce our training dataset. Section 4.2 describes the architecture of FIM-PP. *Training details*, including hyperparameter choices and submodule sizes, are described in Appendix D. Our pretrained model, repository, and tutorials are available online⁵.

The *real-world datasets* used in our experiments are described in Appendix E, including dataset sizes and numbers of marks. For data sourcing and preprocessing, we follow Zeng et al. (2024), as discussed in Appendix E. Finally, the *evaluation metrics* used in all experiments are described in Appendix F.

⁵<https://fim4science.github.io/OpenFIM/intro.html>

ACKNOWLEDGMENTS

This research has been funded by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the Lamarr Institute for Machine Learning and Artificial Intelligence. Additionally, César Ojeda was supported by Deutsche Forschungsgemeinschaft (DFG) – Project-ID 318763901 – SFB1294.

REFERENCES

- Yacine Aït-Sahalia, Julio Cacho-Diaz, and Roger JA Laeven. Modeling financial contagion using mutually exciting jump processes. *Journal of Financial Economics*, 117(3):585–606, 2015.
- David Berghaus, Kostadin Cvejoski, Patrick Seifner, César Ojeda, and Ramsés J. Sanchez. Foundation inference models for markov jump processes. *Proceedings of the 38th International Conference on Neural Information Processing Systems*, 2024.
- Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.
- Wen-Hao Chiang, Xueying Liu, and George Mohler. Hawkes process modeling of covid-19 with mobility leading indicators and spatial covariates. *International journal of forecasting*, 38(2): 505–520, 2022.
- Kostadin Cvejoski, Ramsés J Sánchez, Bogdan Georgiev, Christian Bauckhage, and César Ojeda. Recurrent point review models. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2020.
- Kostadin Cvejoski, Ramses J. Sanchez, Christian Bauckhage, and Cesar Ojeda. Dynamic review-based recommenders. In *International Data Science Conference*, 2021. arXiv:2110.14747.
- D. J. Daley and David Vere-Jones. *An Introduction to the Theory of Point Processes: Volume II: General Theory and Structure*. Probability and Its Applications. Springer Science+Business Media, 2007. ISBN 978-0-387-21337-8. URL <https://link.springer.com/book/10.1007/978-1-4757-2001-3>.
- Stéphane d’Ascoli, Sören Becker, Philippe Schwallier, Alexander Mathis, and Niki Kilbertus. ODEFormer: Symbolic regression of dynamical systems with transformers. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=TzoHLiGVMo>.
- Prathamesh Deshpande, Kamlesh Marathe, Abir De, and Sunita Sarawagi. Long horizon forecasting with temporal point processes. In *Proceedings of the 14th ACM international conference on web search and data mining*, pp. 571–579, 2021.
- Felix Draxler, Yang Meng, Kai Nelson, Lukas Laskowski, Yibo Yang, Theofanis Karaletsos, and Stephan Mandt. Transformers for mixed-type event sequences. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL <https://openreview.net/forum?id=MtwsRjPZhf>.
- Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1555–1564, 2016.
- Alan G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 04 1971. ISSN 0006-3444. doi: 10.1093/biomet/58.1.83. URL <https://doi.org/10.1093/biomet/58.1.83>.
- Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- Gavin Kerrigan, Kai Nelson, and Padhraic Smyth. Eventflow: Forecasting temporal point processes with flow matching. *arXiv preprint arXiv:2410.07430*, 2024.
- Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33:6696–6707, 2020.
- John Frank Charles Kingman. *Poisson processes*, volume 3. Clarendon Press, 1992.
- Patrick J. Laub, Thomas Taimre, and Philip K. Pollett. Hawkes processes, 2015. URL <https://arxiv.org/abs/1507.02822>.
- Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>, June 2014.
- Haitao Lin, Lirong Wu, Guojiang Zhao, Liu Pai, and Stan Z. Li. Exploring generative neural temporal point process. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL <https://openreview.net/forum?id=NPfS5N3jbL>.
- Haitao Lin, Cheng Tan, Lirong Wu, Zhangyang Gao, Zicheng Liu, and Stan Z. Li. An extensive survey with empirical studies on deep temporal point process. *Journal of \LaTeX Class Files*, ??: 1–22, 2024. arXiv:2110.09823.
- Scott Linderman and Ryan Adams. Discovering latent network structure in point process data. In *International conference on machine learning*, pp. 1413–1421. PMLR, 2014.
- Zefang Liu and Yinzhu Quan. Tpp-llm: Modeling temporal point processes by efficiently fine-tuning large language models. *arXiv preprint arXiv:2410.02062*, 2024.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL <https://arxiv.org/abs/1711.05101>.
- David Lüdke, Marin Biloš, Oleksandr Shchur, Marten Lienen, and Stephan Günnemann. Add and thin: Diffusion for temporal point processes. *Advances in Neural Information Processing Systems*, 36:56784–56801, 2023.
- Noa Malem-Shinitzki, César Ojeda, and Manfred Opper. Variational bayesian inference for nonlinear hawkes process with gaussian process self-effects. *Entropy*, 24(3):356, 2022.
- César Ali Ojeda Marin, Wilhelm Huisinga, Purity Kavwele, and Niklas Hartung. Amortized in-context mixed effect transformer models: A zero-shot approach for pharmacokinetics. *arXiv preprint arXiv:2508.15659*, 2025.
- Maximilian Mael, Manuel Hinz, Patrick Seifner, David Berghaus, and Ramses J Sanchez. Towards foundation inference models that learn odes in-context. *arXiv preprint arXiv:2510.12650*, 2025.
- Maximilian Mael, Johannes R Hübers, David Berghaus, Patrick Seifner, and Ramses J Sanchez. Foundation inference models for ordinary differential equations. *arXiv preprint arXiv:2602.08733*, 2026.
- Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6463c88460bd63bbe256e495c63aa40b-Paper.pdf.
- Hongyuan Mei, Guanghui Qin, and Jason Eisner. Imputing missing events in continuous-time event streams. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4475–4485. PMLR, 09–15 Jun 2019. URL <https://proceedings.mlr.press/v97/mei19a.html>.

- Hongyuan Mei, Chenghao Yang, and Jason Eisner. Transformer embeddings of irregularly spaced events and their participants. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=Rty5g9imm7H>.
- Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=KSugKcbNf9>.
- Samuel Müller, Arik Reuter, Noah Hollmann, David Rügamer, and Frank Hutter. Position: The future of bayesian prediction is prior-fitted. *arXiv preprint arXiv:2505.23947*, 2025.
- Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 188–197, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1018. URL <https://aclanthology.org/D19-1018/>.
- Y. Ogata. On lewis’ simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31, 1981.
- César Ojeda, Kostadin Cvejoski, Bodgan Georgiev, Christian Bauckhage, Jannis Schuecker, and Ramsés J Sánchez. Learning deep generative models for queuing systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9214–9222, 2021.
- Aristeidis Panos. Decomposable transformer point processes. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=OesteJF01s>.
- Jakob Gulddahl Rasmussen. Bayesian inference for hawkes processes. *Methodology and Computing in Applied Probability*, 15(3):623–642, 2013.
- Jakob Gulddahl Rasmussen. Temporal point processes and the conditional intensity function. *arXiv preprint arXiv:1806.00221*, 2018.
- Patrick Seifner, Kostadin Cvejoski, David Berghaus, Cesar Ojeda, and Ramses J Sanchez. In-context learning of stochastic differential equations with foundation inference models. In *The Thirtieth Annual Conference on Neural Information Processing Systems*, 2025a. URL <https://openreview.net/forum?id=ceCJPoZOKJ>.
- Patrick Seifner, Kostadin Cvejoski, Antonia Körner, and Ramses J Sanchez. Zero-shot imputation with foundation inference models for dynamical systems. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL <https://openreview.net/forum?id=NPSZ7V1CCY>.
- Oleksandr Shchur, Marin Biloš, and Stephan Günnemann. Intensity-free learning of temporal point processes. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=HygOjhEYDH>.
- Satya Narayan Shukla and Benjamin Marlin. Multi-time attention networks for irregularly sampled time series. In *ICML Workshop on the Art of Learning with Missing Values (Artemiss)*, 2020. URL <https://openreview.net/forum?id=mXbhcalKnYM>.
- Yujee Song, Donghyun Lee, Rui Meng, and Won Hwa Kim. Decoupled marked temporal point process using neural ordinary differential equations. *arXiv preprint arXiv:2406.06149*, 2024.
- Wilson Truccolo, Uri T Eden, Matthew R Fellows, John P Donoghue, and Emery N Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of neurophysiology*, 93(2):1074–1089, 2005.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

- Dongxia Wu, Tsuyoshi Idé, Georgios Kollias, Jiri Navratil, Aurelie Lozano, Naoki Abe, Yian Ma, and Rose Yu. Learning granger causality from instance-wise self-attentive hawkes processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 415–423. PMLR, 2024.
- Shuai Xiao, Mehrdad Farajtabar, Xiaojing Ye, Junchi Yan, Le Song, and Hongyuan Zha. Wasserstein learning of deep generative point process models. In *Advances in Neural Information Processing Systems*, 2017. arXiv:1705.08051.
- Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning granger causality for hawkes processes. In *International conference on machine learning*, pp. 1717–1726. PMLR, 2016.
- Siqiao Xue, Xiaoming Shi, James Y. Zhang, and Hongyuan Mei. HYPRO: A hybridly normalized probabilistic model for long-horizon prediction of event sequences. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL <https://openreview.net/forum?id=n6QYLj1YhkG>.
- Siqiao Xue, Xiaoming Shi, Zhixuan Chu, Yan Wang, Hongyan Hao, Fan Zhou, Caigao JIANG, Chen Pan, James Y. Zhang, Qingsong Wen, JUN ZHOU, and Hongyuan Mei. EasyTPP: Towards open benchmarking temporal point processes. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=PJwAkg0z7h>.
- Chenghao Yang, Hongyuan Mei, and Jason Eisner. Transformer embeddings of irregularly spaced events and their participants. *arXiv preprint arXiv:2201.00044*, 2021.
- Mai Zeng, Florence Regol, and Mark Coates. Interacting diffusion processes for event sequence forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, ICML’24. JMLR.org, 2024.
- Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive Hawkes process. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11183–11193. PMLR, 13–18 Jul 2020. URL <https://proceedings.mlr.press/v119/zhang20q.html>.
- 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*, pp. 1513–1522, 2015.
- Ke Zhou, Hongyuan Zha, and Le Song. Learning triggering kernels for multi-dimensional hawkes processes. In Sanjoy Dasgupta and David McAllester (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1301–1309, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL <https://proceedings.mlr.press/v28/zhoul3.html>.
- Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. Learning tree-based deep model for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’18, pp. 1079–1088. ACM, July 2018. doi: 10.1145/3219819.3219826. URL <http://dx.doi.org/10.1145/3219819.3219826>.
- Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International Conference on Machine Learning*, 2021. arXiv:2002.09291.

A ADDITIONAL RESULTS

A.1 PERFORMANCE ON VARIOUS SYNTHETIC DATASETS

In this section we highlight the performance of FIM-PP (zs) on various synthetic datasets coming from different processes. Figure 4 compares the estimated intensity to the ground-truth on some synthetic processes, including the powerlaw kernel. The model captures the essence of all processes, including the out-of-distribution powerlaw kernel.

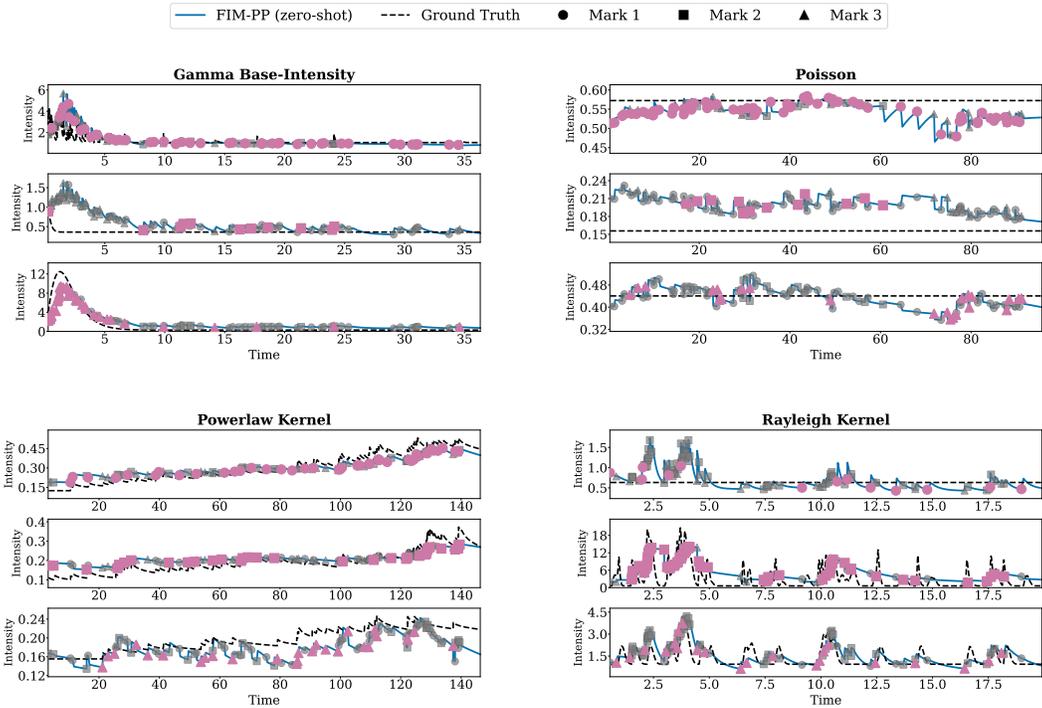


Figure 4: Intensity predictions of $FIM-PP$ (zs) on synthetic datasets of four different process types. We remark that the model has not been trained on powerlaw kernels but still predicts them with decent accuracy.

A.2 LONG-HORIZON PREDICTION

The experimental setup defined by Zeng et al. (2024) covers four metrics (OTD, $RMSE_c$, $RMSE_{\Delta t}$, $sMAPE_{\Delta t}$) and five real-world datasets (TAXI, TAobao, STACKOVERFLOW, AMAZON and RETWEET). Table 3, Table 4 and Table 5 contain the long horizon results for all these datasets and metrics for horizon sizes $N = 20$, $N = 10$ and $N = 5$, respectively.

A.3 FINE-TUNING

Due to the relatively small parameter count of 16.1M, finetuning takes just a few minutes and less than 11GB of GPU memory for all datasets. In Figure 5 we compare the finetuning speed against training a $FIM-PP$ from scratch, i.e. with randomly initialized parameters. The results indicate that pre-training yields a good initialization for finetuning, which converges much faster than training from randomly initialized set of parameters. Moreover, the finetuned model generally archives lower errors on the test split.

A.4 PERFORMANCE WITH VARYING CONTEXT SIZE

We also investigated the sensitivity of $FIM-PP$ to the number of context paths passed to the model. We studied this behavior on data from three synthetic processes and evaluated the results based on the $sMAPE$ metric. Figure 6 contains the results of this experiment. Initially, providing more context paths to the model improves the accuracy of the estimated process. After a few hundred context paths, the performance saturates; more paths only improve marginally improve the estimate. Crucially, the performance does not degrade with more paths, highlighting the robustness of our model.

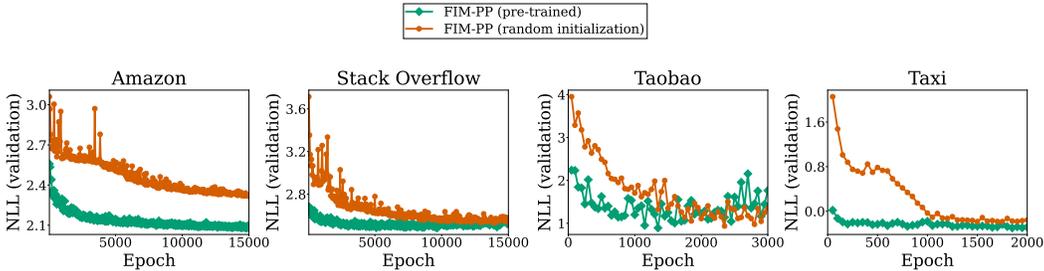


Figure 5: Comparison of the fine-tuning loss curves of a pre-trained FIM-PP model versus random initialization. Note that one epoch corresponds to just one inference-path prediction and is therefore very fast. Our results indicate that the pre-training achieves faster convergence as well as a higher loglikelihood when converged.

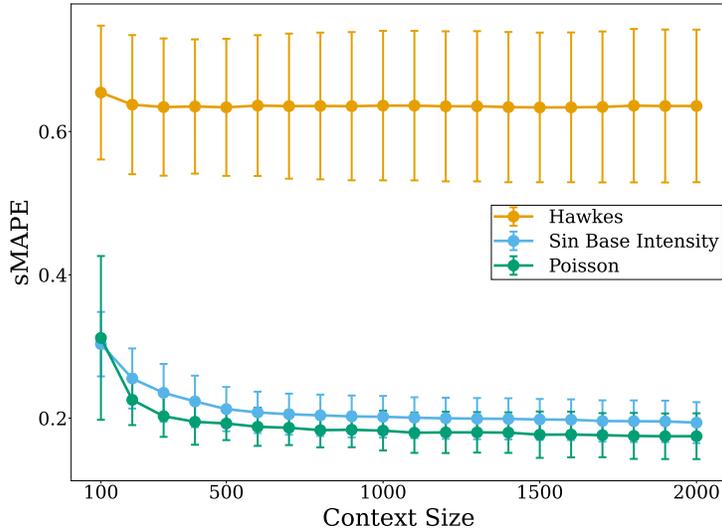


Figure 6: sMAPE error against the ground truth intensity for varying number of context paths. We used 100 points per path. Our results indicate that the performance of the model does not noticeably improve for more than 500 paths, at least for the synthetic datasets tested.

B DATA GENERATION

To train our Foundation Inference Model, we generate a comprehensive synthetic dataset of marked temporal point processes. Each process is an instance of a multivariate Hawkes process. The conditional intensity function $\lambda(t, \kappa | \mathcal{H}_t)$ at time $t \in \mathbb{R}_+$ for a mark $\kappa \in \mathcal{K}$ given a history of marked events $\mathcal{H}_t = \{(t_i, \kappa_i) | t_i < t\}$ is defined as

$$\lambda(t, \kappa | \mathcal{H}_t) = \max \left(0, \mu_\kappa(t) + \sum_{(t', \kappa') \in \mathcal{H}_t} z_{\kappa\kappa'} \gamma_{\kappa\kappa'}(t - t') \right), \quad (13)$$

where μ_κ is the time-dependent base intensity for mark κ , $\gamma_{\kappa\kappa'}$ is the interaction kernel between κ and $\kappa' \in \mathcal{K}$ and $z_{\kappa\kappa'} \in \mathbb{R}$ are sampled pre-factors of the interaction, varying the interaction behavior further.

To the best of our knowledge, no open-source solution for sampling such Hawkes processes with time-dependent base intensity functions exists. Hence, we implemented an efficient custom sam-

pling library for such processes in C++. We will release the source code of this library in the supplementary material of our work.

B.1 DATASET CONFIGURATIONS

We sample Hawkes processes instances over a set of marks \mathcal{K} in equation 13 in two stages.

At first, the functional forms for the base intensities μ_κ and interaction kernels $\gamma_{\kappa\kappa'}$ are drawn from a library of parametric functions. The parameters for these functions are then sampled from specified prior distributions. Our used functional forms and their parameters are summarized in Table 6. The parameter ranges were chosen more or less arbitrarily so that the paths look *realistic*. We keep these choices fixed and did not modify them based on the performance on our evaluation sets in order to prevent overfitting to those, which would be contrary to the concept of a foundation model.

The pre-factors further diversify the sampled processes by introducing sparse connectivity and inhibitory effects. For each process with interactions, we choose one of two pre-factor distributions Z_{strong} and Z_{sparse} on $\{-1, 0, 1\}$, which differ by their induced connectivity:

$$Z_{\text{strong}} = \text{Categorical}(-1 : 0.06, 0 : 0.4, 1 : 0.54) \quad (14)$$

$$Z_{\text{sparse}} = \text{Categorical}(-1 : 0.01, 0 : 0.9, 1 : 0.09) \quad (15)$$

In other words, for Z_{strong} , only 40% of interactions will be non-influencing, while for Z_{sparse} , 90% of interactions will be non-influencing. For influencing interactions, 90% will be excitatory, while 10% will be inhibitory.

Once the full intensity function for a process is defined, event sequences are generated using Ogata’s modified thinning algorithm.

Figure 7 contains summarizing statistics of our train distribution. Aggregated, this prior is very broad. Importantly, it covers distributions of the real-world datasets in our experiments. The corresponding statistics for these datasets are depicted in Figures 8 to 12.

B.2 DATASET SIZE

We sample instances of every Hawkes process configuration in Table 6, and simulate them for different number of marks, sequences and events, detailed in Table 7. In total, our training data consisted of 72k processes and 14.4M events.

C INSTANCE NORMALIZATION

To ensure that FIM-PP can generalize across datasets with vastly different time scales, we introduce an instance normalization scheme that makes the model agnostic to the absolute units of time. Let $\mathcal{C} = \{\mathcal{S}^j\}_{j=1}^m$ denote a context of FIM-PP, i.e. a set of marked event sequences $\mathcal{S}^j = \{(t_i^j, \kappa_i^j)\}_{i=1}^{n_j}$. Identifying $t_0^j = 0$, we define the inter-event times as $\Delta t_i^j = t_i^j - t_{i-1}^j$ and the maximum inter-event time in the context as

$$\Delta t_{\max}^{\text{cont}} = \max_{j=1, \dots, m} \max_{i=1, \dots, n_j} \Delta t_i^j. \quad (16)$$

All time-related inputs to the model, including context event times t_i^j , inter-event times features Δt_i^j , and history event times (e.g. from a target sequence during training) t , are scaled by the maximum inter-event time:

$$t' = \frac{t}{\Delta t_{\max}^{\text{cont}}}. \quad (17)$$

This transformation maps all temporal information to a canonical scale where the largest inter-event gap becomes 1.

This change of time variable also transforms the intensity function. To preserve the number of expected events within a differential interval, the intensities must be related by $\lambda(t)dt = \lambda'(t')dt'$. Since $dt = \Delta t_{\max}^{\text{cont}} dt'$, it follows that the intensity in the normalized time domain, $\lambda'(t')$, is a scaled version of the original:

$$\lambda'(t') = \Delta t_{\max}^{\text{cont}} \cdot \lambda(t). \quad (18)$$

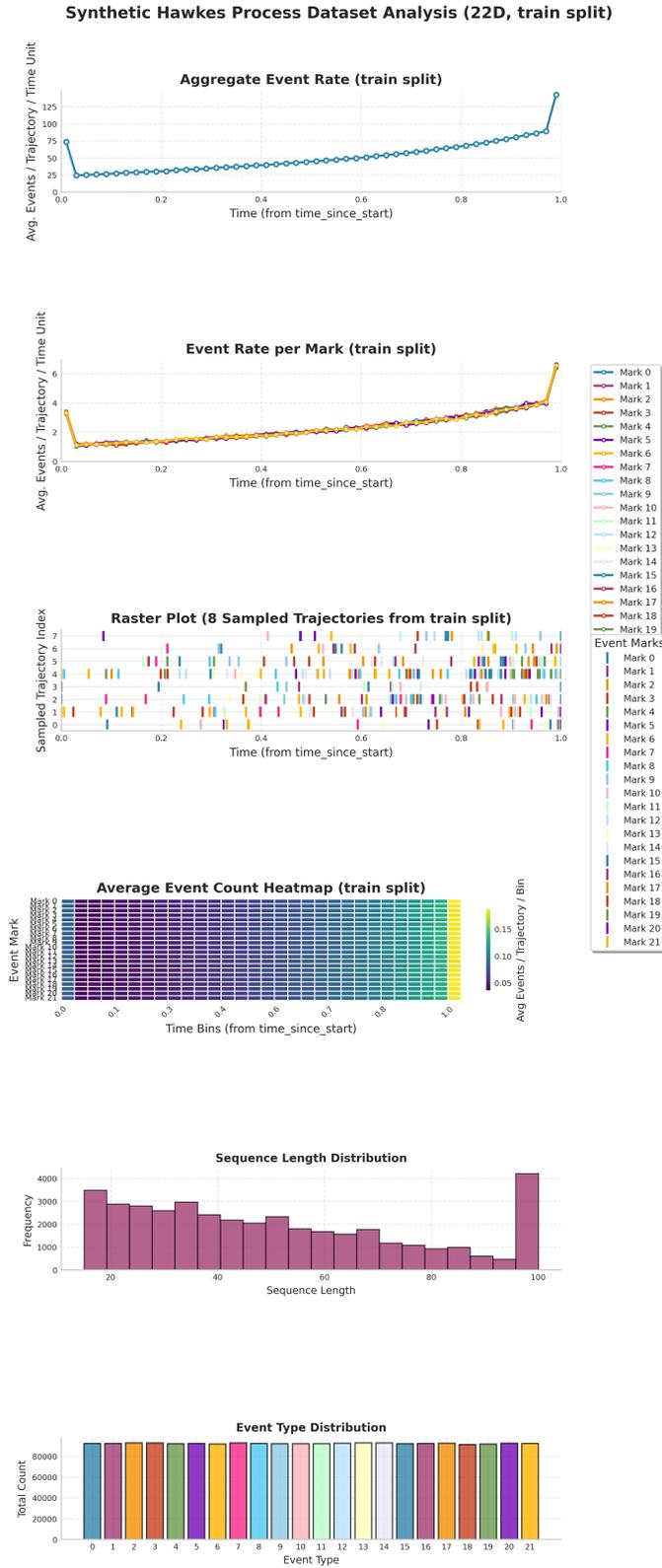


Figure 7: Distribution of our synthetic training data for Hawkes processes with 22 marks. The distribution is almost uniform and very broad and therefore also captures real world phenomena.

EasyTPP Dataset Analysis - Amazon (16D, train split)

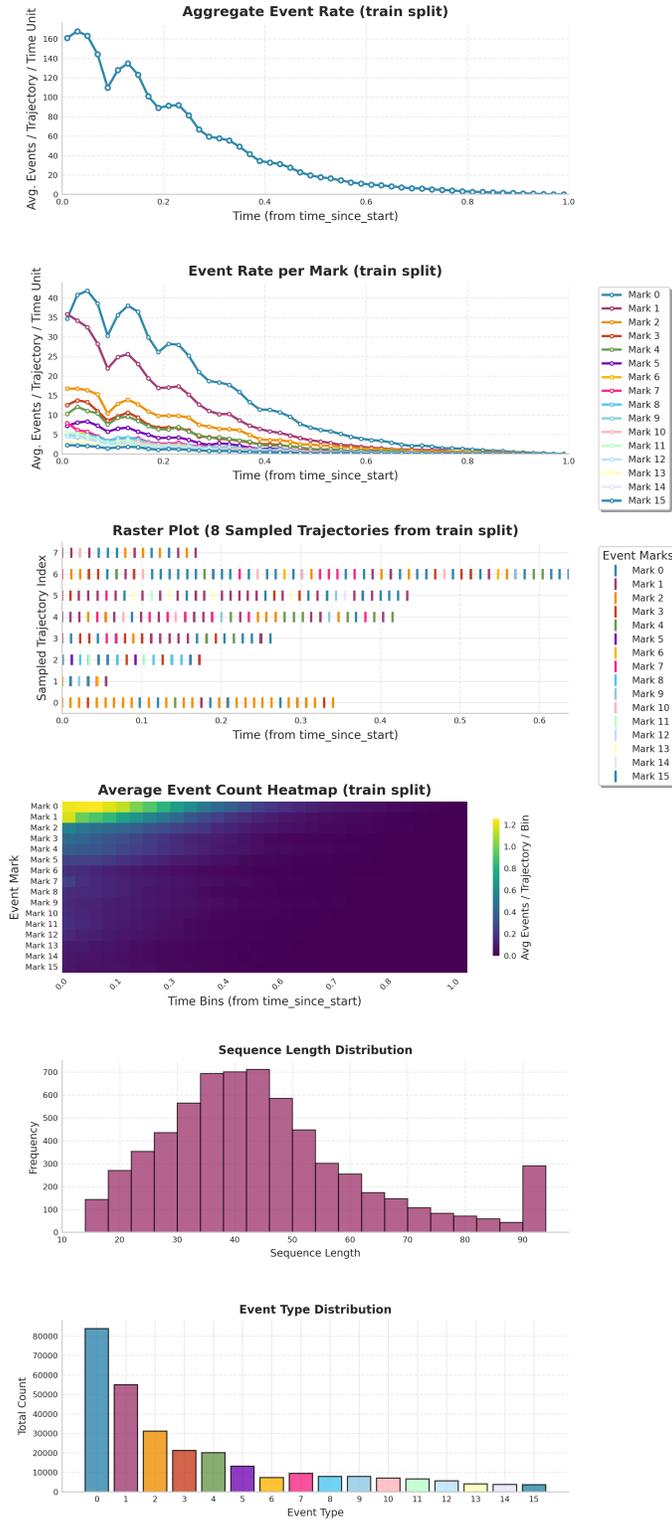


Figure 8: Amazon dataset statistics.

EasyTPP Dataset Analysis - Taxi (10D, train split)

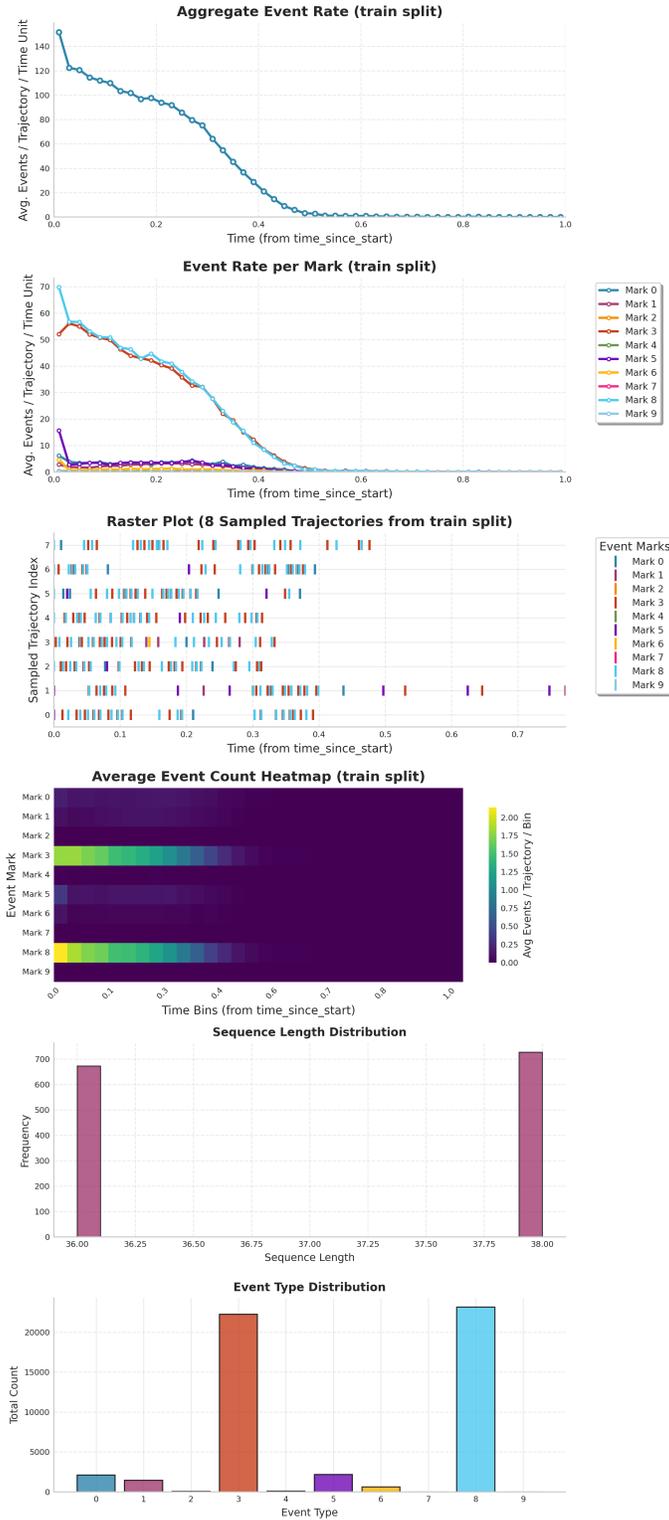


Figure 9: Taxi dataset statistics.

EasyTPP Dataset Analysis - Taobao (17D, train split)

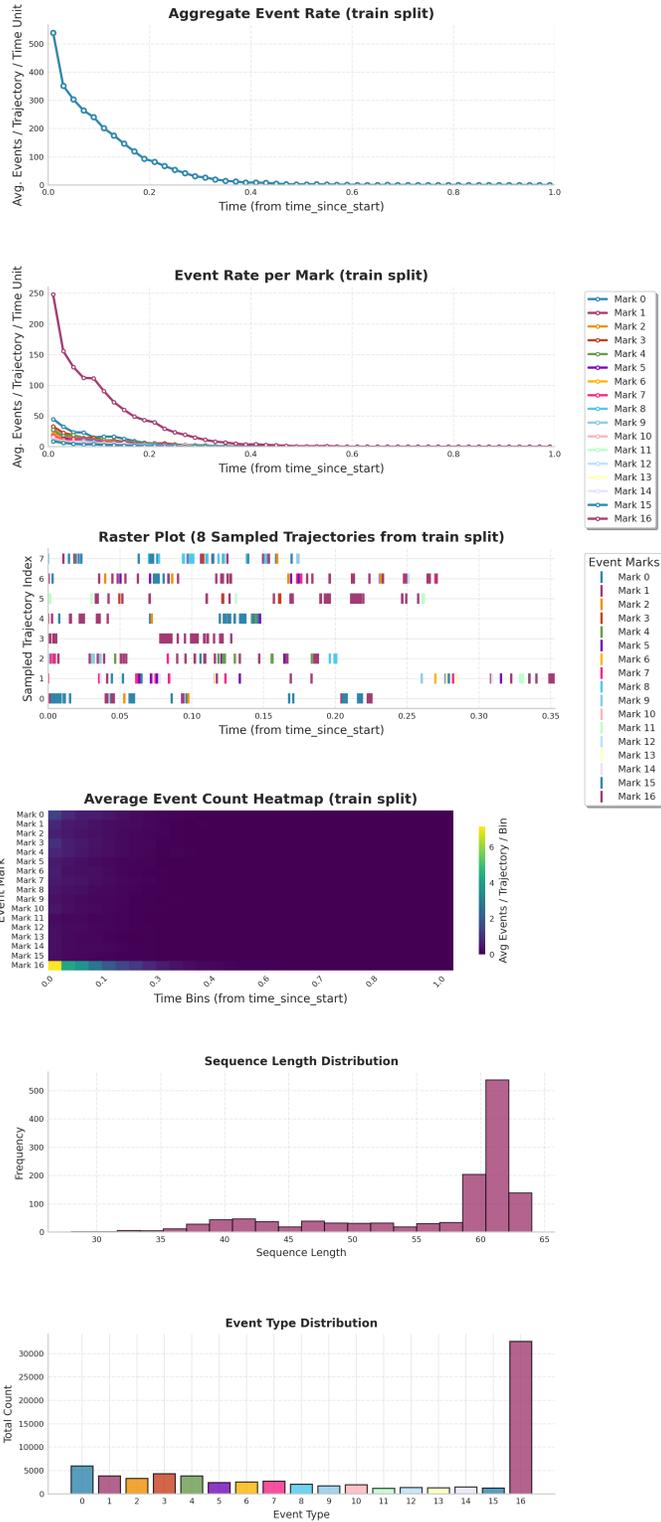


Figure 10: Taobao dataset statistics.

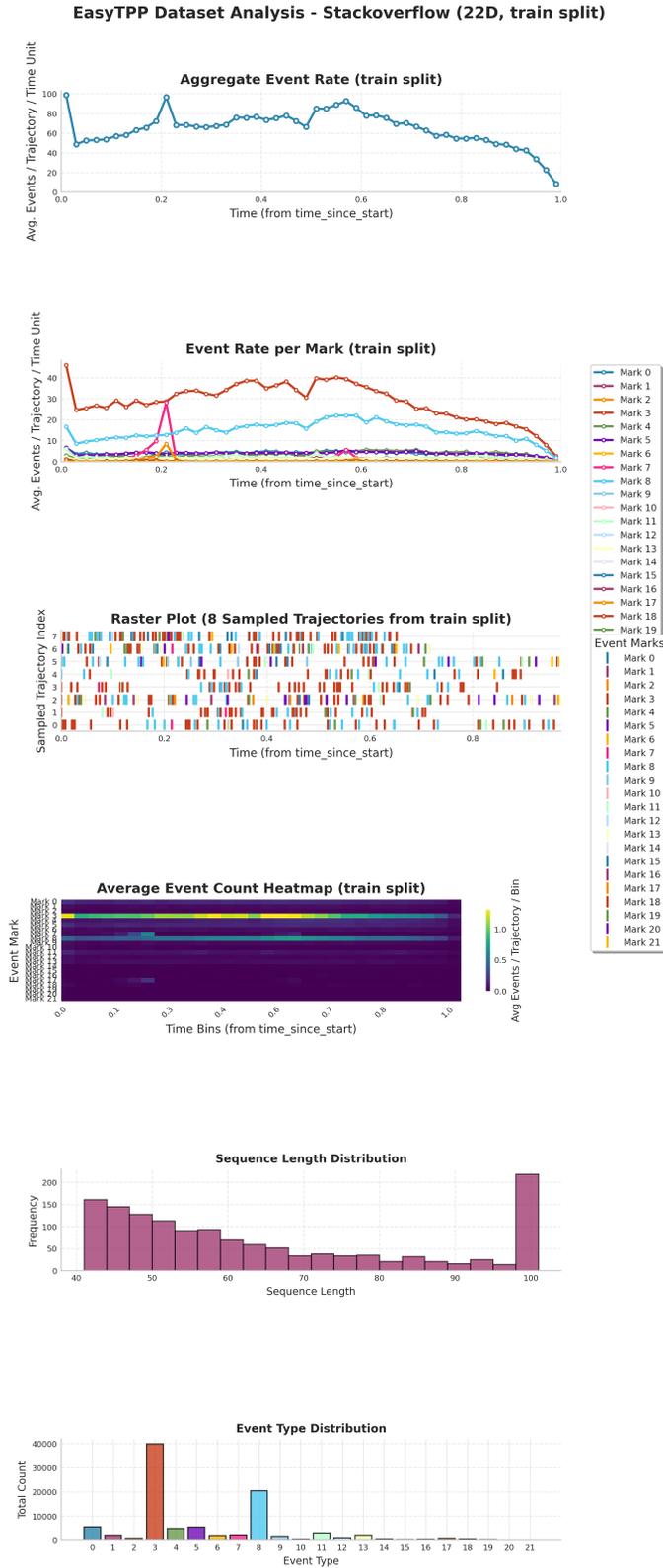


Figure 11: Stackoverflow dataset statistics.

EasyTPP Dataset Analysis - Retweet (3D, train split)

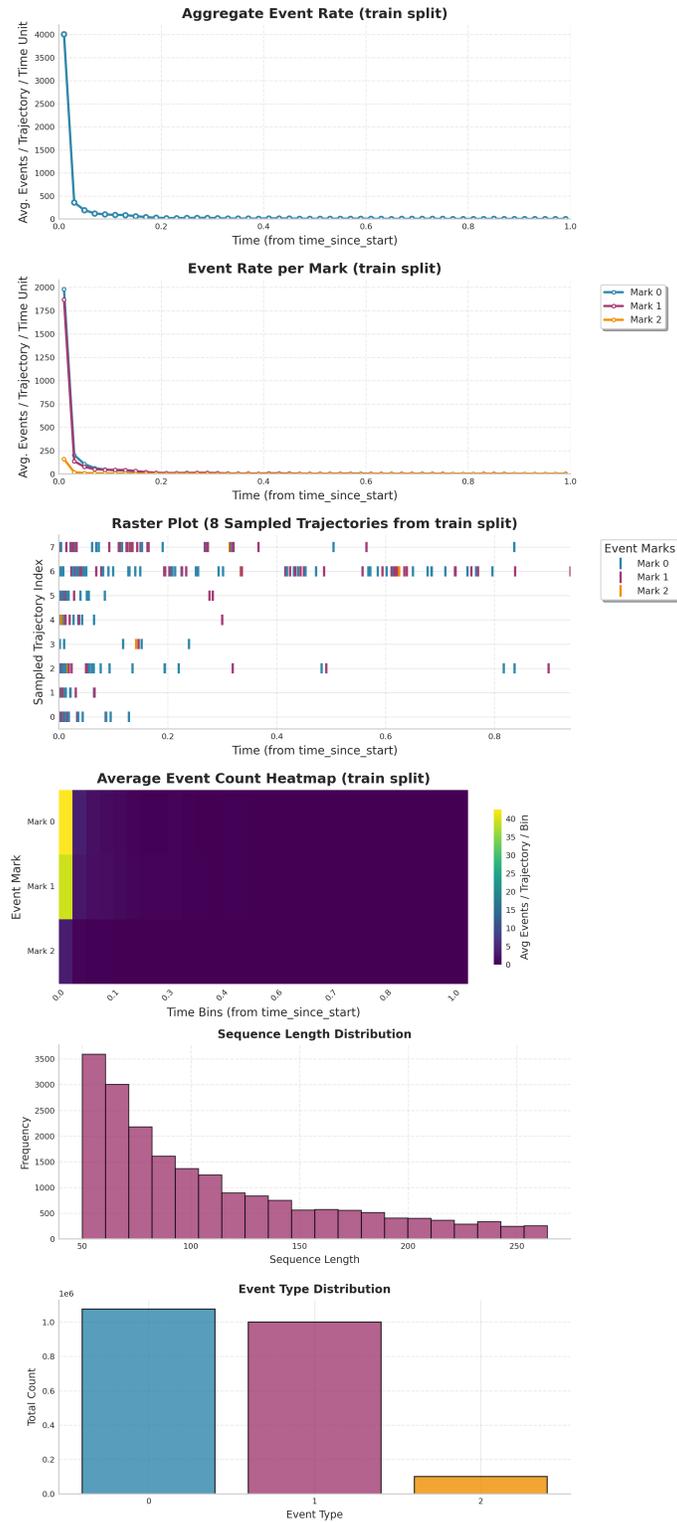


Figure 12: Retweet dataset statistics.

Consequently, the model is trained to predict this normalized intensity $\lambda'(t')$. During inference, to obtain the intensity in the original, real-world time scale, the model’s output is simply denormalized by dividing by the same constant $\Delta t_{\max}^{\text{cont}}$. This entire process allows the FIM to learn scale-invariant temporal dynamics, a key requirement for effective zero-shot inference on unseen data.

D TRAINING DETAILS

Our Foundation Inference Model was trained on the comprehensive synthetic datasets described in Appendix B. The training took about 5 days on a single NVIDIA A100-80GB.

D.1 DATA HANDLING AND BATCHING

Each sample in our dataset represents a single underlying process, comprising up to 2000 distinct time series paths. During training, we dynamically partition these paths into context and inference sets for each batch.

On-the-fly Path Selection For each sample, we randomly select a single path ($P_{\text{inference}} = 1$) to serve as the inference target. The remaining paths are designated as the context set. To train a model that is robust to varying amounts of contextual information, the number of context paths presented in each training step is randomized. Specifically, for each sample in a batch, we uniformly sample a number of context paths between a minimum of 400 and a maximum of 2000.

Variable Sequence Lengths As a form of data augmentation, we also vary the length of the historical sequences. For 90% of the training batches, all sequences (both context and inference) are truncated to a random length chosen uniformly from the interval $[15, 100]$. For the remaining 10% of batches, the full sequence length of 100 events is used. This strategy encourages the model to make reliable predictions from both short and long historical contexts. For validation, we use fixed, full-length sequences to ensure consistent and comparable evaluation metrics.

D.2 HYPERPARAMETERS AND OPTIMIZATION

The model architecture is based on the Transformer Vaswani et al. (2017). Context sequences are processed by a 4-layer Transformer encoder, and the resulting path summaries are further refined by a 2-layer Transformer encoder. The history of the target sequence is processed by a 4-layer Transformer decoder, which attends to the context summary as memory. Both encoders and the decoder use 4 attention heads and a hidden dimension of 256. The final intensity parameters (μ, α, β) are predicted by three separate Multi-Layer Perceptrons (MLPs), each with two hidden layers of 256 units.

In total, our model has 16.1 million trainable parameters.

We trained the model using the AdamW optimizer Loshchilov & Hutter (2019) with a learning rate of 5×10^{-5} and a weight decay of 10^{-4} . To accelerate computation, we utilized bfloat16 mixed-precision training.

D.3 TRAIN OBJECTIVE

We use the standard negative log-likelihood (NLL) for a marked temporal point process as the train objective for FIM-PP. By Section 3, the MTPP density at a sequence of events $\mathcal{S} = \{(t_i, \kappa_i)\}_{i=1}^n$ in the interval $[0, T]$ is

$$f(\{(t_i, \kappa_i)\}_{i=1}^n) = \left[\prod_{i=1}^n \lambda(t_i, \kappa_i | \mathcal{H}_{t_i}) \right] \exp \left(- \int_0^T \lambda(s | \mathcal{H}_s) ds \right). \quad (19)$$

Thus, the NLL of a target sequence under the distribution induced the model’s predicted intensity function $\hat{\lambda}$ is

$$\mathcal{L}_{\text{NLL}} = \sum_{\kappa \in \mathcal{K}} \hat{\Lambda}(T, \kappa) - \sum_{(t, \kappa) \in \mathcal{T}} \hat{\lambda}(t, \kappa | \mathcal{H}_t). \quad (20)$$

where $\hat{\Lambda}(T, \kappa) = \int_0^T \hat{\lambda}(s, \kappa | \mathcal{H}_s) ds$ is the predicted integrated intensity. We approximate the integral using Monte Carlo integration

$$\hat{\Lambda}(T, \kappa) \approx \frac{T}{N_{\text{MC}}} \sum_{i=1}^{N_{\text{MC}}} \hat{\lambda}(s_i, \kappa | \mathcal{H}_{s_i}), \quad (21)$$

with $N_{\text{MC}} = 100$ samples and $s_i \sim \mathcal{U}(0, T)$.

E EVALUATION DATASETS

To evaluate the inference capabilities of FIM-PP, we use five widely-recognized real-world datasets that were not seen during training:

AMAZON This dataset comprises sequences of product reviews from users on the Amazon platform, collected over a ten-year period from 2008 to 2018 Ni et al. (2019). Each sequence represents the review history of a single user. An event is defined by the timestamp of a review, and its mark corresponds to one of 16 distinct product categories. The analysis is performed on a subset of 5,200 of the most active users to ensure sequences are sufficiently long for meaningful analysis.

TAXI Derived from New York City’s public taxi trip records, this dataset captures the operational patterns of taxi drivers. Each sequence corresponds to the activity log of an individual driver. Events are either pick-ups or drop-offs, and the event marks are defined by the combination of the event type (pick-up/drop-off) and the borough where it occurred, resulting in 10 unique marks. The dataset consists of sequences from a random sample of 2,000 drivers.

TAOBAO This dataset originates from the 2018 Tianchi Big Data Competition and contains logs of user interactions on the Taobao e-commerce platform over a period in late 2017 Zhu et al. (2018). The sequences track the behavior of anonymous users, including actions like browsing and purchasing. The 17 event types correspond to different product category groups. For the evaluation, sequences from the 2,000 most active users are utilized.

STACKOVERFLOW Sourced from the popular question-and-answering website StackOverflow, this dataset tracks the awarding of achievement badges to users over a two-year span Leskovec & Krevl (2014). Each sequence represents a user’s history of earned badges. The events are the timestamps when badges were awarded, and the marks are the 22 different types of badges available on the platform. The evaluation subset includes 2,200 active users.

RETWEET This dataset tracks the dynamics of information spread through time-stamped user retweet sequences Zhou et al. (2013). Each sequence corresponds to the retweet history of an individual user. An event is defined by the timestamp of a retweet, and its mark is categorized into one of three types based on the influence of the original poster: "small" (fewer than 120 followers), "medium" (fewer than 1,363 followers), and "large" (all other users). The analysis is performed on a subset of 5,200 active users.

For all real-world datasets, we use the pre-processing and splits from Zeng et al. (2024).

To compare against the other models, FIM-PP uses the sequences which the other models used for training as context and used the same inference sequences for evaluation.

F EVALUATION METRICS

Following Zeng et al. (2024), we adopt a comprehensive set of metrics to evaluate both the temporal and categorical aspects of the predicted sequences. Let $\mathcal{S}_{\text{future}} = \{(t_i, \kappa_i)\}_{i=1}^N$ be a ground truth sequence of N future events, and let $\hat{\mathcal{S}}_{\text{future}}$ be the corresponding predicted sequence. The metrics are defined based on the sequence of inter-arrival times $\Delta \mathbf{t} = [\Delta t_1, \dots, \Delta t_N]$ (where $\Delta t_i = t_i - t_{i-1}$) and the sequence of marks.

Optimal Transport Distance (OTD) We use the Optimal Transport Distance (OTD) to provide a holistic measure of similarity between the predicted and ground truth event sequences (Mei et al., 2019). OTD calculates the minimum cost required to transform the predicted sequence $\hat{\mathcal{S}}_{\text{future}}$ into the ground truth sequence $\mathcal{S}_{\text{future}}$ through a series of operations (insertions, deletions, and substitutions), each associated with a cost. This metric effectively captures discrepancies in timing, marks, and the total number of events.

RMSE on Event Counts (RMSE_e) This metric evaluates how well the model captures the distribution of event types in the predicted sequence. For each event type $\kappa \in \mathcal{K}$, we count its occurrences in the ground truth sequence (C_{κ}) and the predicted sequence (\hat{C}_{κ}). The RMSE_e is the root mean squared error over the vector of these counts, averaged across all m test sequences:

$$\text{RMSE}_e = \sqrt{\frac{1}{m} \sum_{j=1}^m \sum_{\kappa \in \mathcal{K}} (C_{j,\kappa} - \hat{C}_{j,\kappa})^2} \quad (22)$$

Event Type Accuracy (Acc) This metric directly measures the model’s ability to predict the correct event type at each position in the sequence. It is calculated as the fraction of events for which the predicted mark $\hat{\kappa}_i$ matches the ground truth mark κ_i , averaged over all test sequences. This provides a strict, position-wise evaluation of the categorical predictions.

$$\text{Acc} = \frac{1}{m} \sum_{j=1}^m \frac{1}{N} \sum_{i=1}^N \mathbb{I}(\kappa_{j,i} = \hat{\kappa}_{j,i}) \quad (23)$$

where $\mathbb{I}(\cdot)$ is the indicator function. Unlike RMSE_e, which assesses the overall distribution of event types, accuracy penalizes mispredictions at specific positions, making it a more challenging metric for sequential order. A higher accuracy indicates better performance.

Time-series Forecasting Metrics To specifically assess the accuracy of the predicted inter-arrival times Δt , we report two standard time-series forecasting metrics.

- **RMSE on Inter-arrival Times (RMSE_{Δt}):** The standard root mean squared error between the predicted and true vectors of inter-arrival times.

$$\text{RMSE}_{\Delta t} = \sqrt{\frac{1}{m} \sum_{j=1}^m \frac{1}{N} \sum_{i=1}^N (\Delta t_{j,i} - \hat{\Delta t}_{j,i})^2} \quad (24)$$

- **Symmetric Mean Absolute Percentage Error (sMAPE_{Δt}):** A normalized version of MAPE that is less sensitive to outliers and zero values.

$$\text{sMAPE}_{\Delta t} = \frac{100}{m} \sum_{j=1}^m \frac{1}{N} \sum_{i=1}^N \frac{2|\Delta t_{j,i} - \hat{\Delta t}_{j,i}|}{|\Delta t_{j,i}| + |\hat{\Delta t}_{j,i}|} \quad (25)$$

G CHALLENGES IN NEXT EVENT PREDICTION

Our evaluations in table 2 reveal that FIM-PP in zero-shot mode already performs well for next event time prediction. It however gets a noticeably worse error in the next event type prediction. Upon investigating this, we found that many of the real-world datasets have specific patterns (such as oscillations between two marks) that FIM-PP (zs) struggles to capture (see fig 13). After fine-tuning, it is however able to spot these patterns well (see fig 14). This might also explain why FIM-PP performs better on long-horizon tasks: The specific order of the events does not matter here.

We hypothesize that the underlying reason for this shortcoming is that our synthetic dataset distribution does not capture these patterns well. We are planning to investigate this further and to update our synthetic distribution to include such patterns and provide an updated version of FIM-PP.

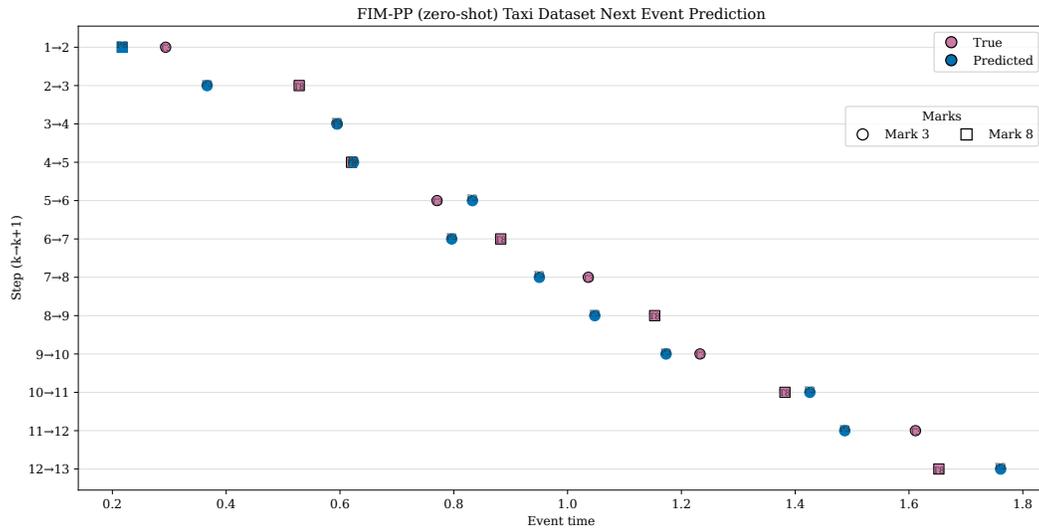


Figure 13: FIM-PP in zero-shot mode struggles to predict the next event type right if the dataset has alternating patterns such as here for the Taxi dataset.

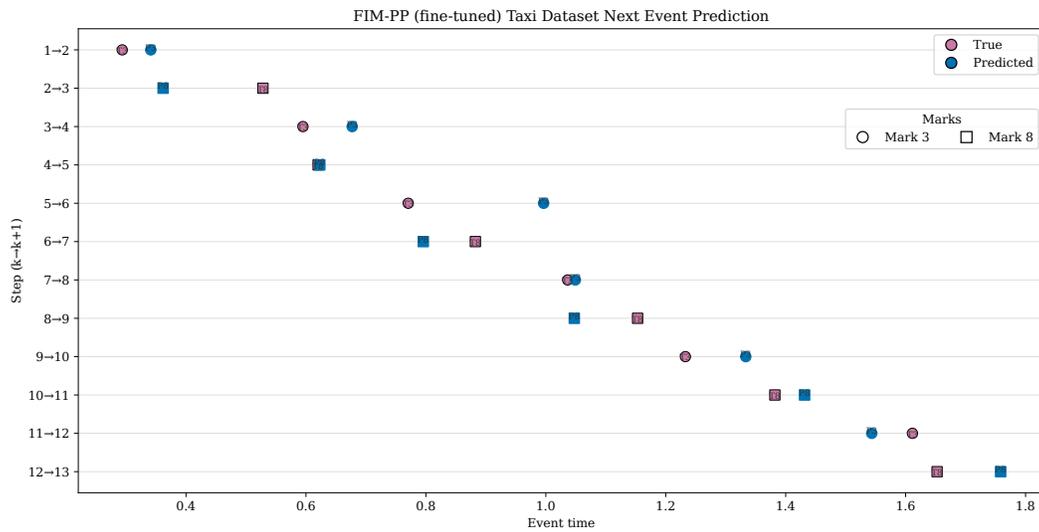


Figure 14: After fine-tuning, FIM-PP is able to spot the alternating pattern between mark 3 and mark 8 in the Taxi dataset.

Table 3: Prediction of $N = 20$ events in test sequences of five real-world datasets. Error-bars indicate the standard deviation over 10 trials. Results for the baseline methods were extracted from Zeng et al. (2024). Best results are bold.

| Dataset | Method | OTD | RMSE _e | RMSE _{Δt} | sMAPE _{Δt} |
|----------------------------|-----------------------------|----------------------|----------------------|---------------------------------------|----------------------------------------|
| TAXI | HYPRO | 21.653±0.163 | 1.231±0.015 | 0.372±0.004 | 93.803±0.454 |
| | Dual-TPP | 24.483±0.383 | 1.353±0.037 | 0.402±0.006 | 95.211±0.187 |
| | A-NHP | 24.762±0.217 | 1.276±0.015 | 0.430±0.003 | 97.388±0.381 |
| | NHP | 25.114±0.268 | 1.297±0.019 | 0.399±0.040 | 96.459±0.521 |
| | IFTPP | 24.053±0.609 | 1.364±0.032 | 0.384±0.005 | 95.719±0.779 |
| | TCDDM | 22.148±0.529 | 1.309±0.030 | 0.382±0.019 | 90.596±0.574 |
| | CDiff | 21.013±0.158 | 1.131±0.017 | 0.351±0.004 | 87.993±0.178 |
| | FIM-PP (zs) (NLL) | 23.145 ±0.073 | 1.421 ±0.014 | 0.277 ±0.000 | 76.765 ±0.386 |
| | FIM-PP (zs) (S-MAPE) | 41.543 ±0.026 | 3.986 ±0.009 | 0.297 ±0.000 | 90.149 ±0.366 |
| | FIM-PP (f) (NLL) | 17.914 ±0.117 | 0.705 ±0.006 | 0.314 ±0.004 | 76.828 ±0.549 |
| FIM-PP (f) (S-MAPE) | 16.493 ±0.034 | 0.707 ±0.004 | 0.278 ±0.001 | 76.538 ±0.214 | |
| TAOBAO | HYPRO | 44.336±0.127 | 2.710±0.021 | 0.594±0.030 | 134.922±0.473 |
| | Dual-TPP | 47.324±0.541 | 3.237±0.049 | 0.871±0.005 | 141.687±0.431 |
| | A-NHP | 45.555±0.345 | 2.737±0.021 | 0.708±0.010 | 134.582±0.920 |
| | NHP | 48.131±0.297 | 3.355±0.030 | 0.837±0.009 | 137.644±0.764 |
| | IFTPP | 45.757±0.287 | 3.193±0.043 | 0.575±0.012 | 127.436±0.606 |
| | TCDDM | 45.563±0.889 | 2.850±0.058 | 0.569±0.015 | 126.512±0.491 |
| | CDiff | 44.621±0.139 | 2.653±0.022 | 0.551 ±0.002 | 125.685 ±0.151 |
| | FIM-PP (zs) (NLL) | 64.281 ±0.077 | 3.949 ±0.010 | 1.988 ±0.006 | 169.687 ±0.089 |
| | FIM-PP (zs) (S-MAPE) | 65.088 ±0.037 | 3.251 ±0.007 | 4.042 ±0.025 | 174.814 ±0.102 |
| | FIM-PP (f) (NLL) | 60.106 ±0.464 | 2.428 ±0.005 | 16.068 ±0.109 | 152.528 ±0.377 |
| FIM-PP (f) (S-MAPE) | 58.900 ±0.436 | 2.347 ±0.014 | 14.734 ±0.120 | 152.547 ±0.406 | |
| STACKOVERFLOW | HYPRO | 42.359±0.170 | 1.140±0.014 | 1.554±0.010 | 110.988±0.559 |
| | Dual-TPP | 41.752±0.200 | 1.134 ±0.019 | 1.514±0.017 | 117.582±0.420 |
| | A-NHP | 42.591±0.408 | 1.142±0.011 | 1.340±0.006 | 108.542±0.531 |
| | NHP | 43.791±0.147 | 1.244±0.030 | 1.487±0.004 | 116.952±0.404 |
| | IFTPP | 46.280±0.892 | 1.447±0.057 | 1.669±0.005 | 115.122±0.627 |
| | TCDDM | 42.128±0.591 | 1.467±0.014 | 1.315±0.004 | 107.659±0.934 |
| | CDiff | 41.245±1.400 | 1.141±0.007 | 1.199±0.006 | 106.175±0.340 |
| | FIM-PP (zs) (NLL) | 49.259 ±0.056 | 2.393 ±0.015 | 1.068 ±0.002 | 96.364 ±0.048 |
| | FIM-PP (zs) (S-MAPE) | 43.630 ±0.242 | 1.488 ±0.010 | 1.050 ±0.002 | 93.544 ±0.485 |
| | FIM-PP (f) (NLL) | 39.792 ±0.042 | 1.336 ±0.030 | 1.018 ±0.003 | 88.248 ±0.189 |
| FIM-PP (f) (S-MAPE) | 39.245 ±0.056 | 1.407 ±0.022 | 1.008 ±0.003 | 88.277 ±0.231 | |
| AMAZON | HYPRO | 38.613±0.536 | 2.007 ±0.054 | 0.477±0.010 | 82.506±0.840 |
| | Dual-TPP | 42.646±0.752 | 2.562±0.202 | 0.482±0.012 | 86.453±2.044 |
| | A-NHP | 39.480±0.326 | 2.166±0.026 | 0.476±0.033 | 84.323±1.815 |
| | NHP | 42.571±0.293 | 2.561±0.060 | 0.519±0.023 | 92.053±1.553 |
| | IFTPP | 43.820±0.232 | 3.050±0.286 | 0.481±0.145 | 90.910±1.611 |
| | TCDDM | 42.245±0.174 | 2.998±0.115 | 0.476±0.111 | 83.826±1.508 |
| | CDiff | 37.728±0.199 | 2.091±0.163 | 0.464±0.086 | 81.987±1.905 |
| | FIM-PP (zs) (NLL) | 46.219 ±0.108 | 2.073 ±0.012 | 0.464 ±0.001 | 128.635 ±0.398 |
| | FIM-PP (zs) (S-MAPE) | 45.153 ±0.092 | 2.106 ±0.009 | 0.453 ±0.002 | 124.476 ±0.494 |
| | FIM-PP (f) (NLL) | 37.208 ±0.098 | 2.030 ±0.019 | 0.366 ±0.001 | 81.188 ±0.142 |
| FIM-PP (f) (S-MAPE) | 37.454 ±0.060 | 2.116 ±0.002 | 0.361 ±0.002 | 87.257 ±0.198 | |
| RETWEET | HYPRO | 61.031±0.092 | 2.623±0.036 | 30.100±0.413 | 106.110±1.505 |
| | Dual-TPP | 61.095±0.101 | 2.679±0.026 | 28.914±0.300 | 106.900±1.293 |
| | A-NHP | 60.634±0.097 | 2.561±0.054 | 28.812±0.272 | 107.234±1.293 |
| | NHP | 60.953±0.079 | 2.651±0.045 | 27.130±0.224 | 107.075±1.398 |
| | IFTPP | 61.715±0.152 | 2.776±0.043 | 27.582±0.191 | 106.711±1.615 |
| | TCDDM | 60.501±0.087 | 2.387±0.050 | 27.303±0.152 | 106.048±0.610 |
| | CDiff | 60.661±0.101 | 2.293 ±0.034 | 27.101±0.113 | 106.184±1.121 |
| | FIM-PP (zs) (NLL) | 60.238 ±0.161 | 4.172 ±0.064 | 24.057 ±0.050 | 99.069 ±0.390 |
| | FIM-PP (zs) (S-MAPE) | 59.392 ±0.149 | 4.323 ±0.042 | 24.804 ±0.014 | 106.317 ±0.187 |
| | FIM-PP (f) (NLL) | 59.437 ±0.082 | 2.703 ±0.012 | 21.985 ±0.014 | 87.585 ±0.171 |
| FIM-PP (f) (S-MAPE) | 59.150 ±0.061 | 3.081 ±0.020 | 21.800 ±0.025 | 87.754 ±0.109 | |

Table 4: Prediction of 10 events in test sequences of five real-world datasets. Error-bars indicate the standard deviation over 10 trials. Results for the baseline methods were extracted from Zeng et al. (2024). Best results are bold.

| Dataset | Method | OTD | RMSE _e | RMSE _{Δt} | sMAPE _{Δt} |
|----------------------------|-----------------------------|----------------------|----------------------|----------------------|-----------------------|
| TAXI | HYPRO | 11.875±0.172 | 0.764±0.008 | 0.363±0.002 | 89.524±0.552 |
| | Dual-TPP | 13.058±0.220 | 0.966±0.011 | 0.395±0.003 | 90.812±0.497 |
| | A-NHP | 12.542±0.336 | 0.823±0.007 | 0.376±0.003 | 92.812±0.129 |
| | NHP | 13.377±0.184 | 0.922±0.009 | 0.397±0.005 | 92.182±0.384 |
| | IFTPP | 12.765±0.106 | 1.004±0.013 | 0.383±0.015 | 93.120±0.526 |
| | TCDDM | 11.885±0.149 | 1.121±0.072 | 0.385±0.009 | 90.703±0.356 |
| | CDiff | 11.004±0.191 | 0.785±0.007 | 0.350±0.002 | 90.721±0.291 |
| | FIM-PP (zs) (NLL) | 13.820 ±0.124 | 1.190 ±0.013 | 0.281 ±0.001 | 78.141 ±0.414 |
| | FIM-PP (zs) (S-MAPE) | 19.425 ±0.127 | 1.959 ±0.015 | 0.299 ±0.001 | 90.510 ±0.267 |
| | FIM-PP (f) (NLL) | 8.336 ±0.071 | 0.451 ±0.006 | 0.291 ±0.004 | 75.366 ±0.160 |
| FIM-PP (f) (S-MAPE) | 8.175 ±0.052 | 0.465 ±0.004 | 0.275 ±0.001 | 73.529 ±0.138 | |
| TAOBAO | HYPRO | 21.547±0.138 | 1.527±0.035 | 0.591±0.019 | 133.147±0.341 |
| | Dual-TPP | 23.691±0.203 | 2.674±0.032 | 0.873±0.010 | 139.271±0.348 |
| | A-NHP | 21.683±0.215 | 1.514±0.015 | 0.608±0.011 | 135.271±0.395 |
| | NHP | 24.068±0.331 | 2.769±0.033 | 0.855±0.013 | 137.693±0.225 |
| | IFTPP | 23.195±0.039 | 2.429±0.045 | 0.602±0.037 | 127.411±0.573 |
| | TCDDM | 21.012 ±0.520 | 2.598±0.047 | 0.610±0.022 | 132.711±0.774 |
| | CDiff | 21.221 ±0.176 | 1.416±0.024 | 0.535 ±0.016 | 126.824 ±0.366 |
| | FIM-PP (zs) (NLL) | 31.880 ±0.040 | 2.024 ±0.004 | 1.955 ±0.011 | 170.278 ±0.029 |
| | FIM-PP (zs) (S-MAPE) | 32.249 ±0.041 | 1.822 ±0.004 | 3.538 ±0.039 | 172.683 ±0.194 |
| | FIM-PP (f) (NLL) | 27.974 ±0.162 | 1.325 ±0.010 | 14.954 ±0.253 | 145.821 ±1.120 |
| FIM-PP (f) (S-MAPE) | 27.940 ±0.075 | 1.358 ±0.009 | 14.844 ±0.055 | 153.499 ±0.553 | |
| STACKOVERFLOW | HYPRO | 21.062±0.372 | 0.921±0.019 | 1.235±0.006 | 107.566±0.218 |
| | Dual-TPP | 21.229±0.394 | 0.936±0.013 | 1.223±0.010 | 107.274±0.200 |
| | A-NHP | 22.019±0.220 | 0.978±0.023 | 1.225±0.007 | 100.137±0.167 |
| | NHP | 21.655±0.314 | 0.970±0.014 | 1.266±0.003 | 108.867±0.361 |
| | IFTPP | 22.339±0.322 | 0.970±0.011 | 1.251±0.005 | 105.674±0.337 |
| | TCDDM | 22.042±0.193 | 1.205±0.014 | 1.228±0.010 | 108.111±0.112 |
| | CDiff | 20.191±0.455 | 0.916±0.010 | 1.180±0.003 | 102.367±0.267 |
| | FIM-PP (zs) (NLL) | 23.527 ±0.033 | 1.188 ±0.005 | 1.039 ±0.003 | 92.919 ±0.556 |
| | FIM-PP (zs) (S-MAPE) | 21.182 ±0.146 | 0.857 ±0.010 | 1.047 ±0.002 | 93.715 ±0.388 |
| | FIM-PP (f) (NLL) | 19.938 ±0.093 | 0.823 ±0.010 | 1.012 ±0.004 | 87.503 ±0.402 |
| FIM-PP (f) (S-MAPE) | 19.846 ±0.152 | 0.832 ±0.010 | 1.004 ±0.003 | 87.110 ±0.393 | |
| AMAZON | HYPRO | 24.956±0.663 | 1.765±0.039 | 0.442±0.015 | 83.401±1.033 |
| | Dual-TPP | 25.929±0.280 | 2.098±0.101 | 0.475±0.008 | 82.352±1.285 |
| | A-NHP | 24.116±0.807 | 1.741±0.039 | 0.454±0.014 | 84.323±1.815 |
| | NHP | 25.730±0.497 | 1.843±0.053 | 0.491±0.048 | 89.135±1.092 |
| | IFTPP | 26.632±0.519 | 1.955±0.112 | 0.464±0.066 | 89.305±1.288 |
| | TCDDM | 25.091±0.227 | 1.778±0.090 | 0.448±0.082 | 82.105 ±1.564 |
| | CDiff | 24.230±0.287 | 1.766±0.079 | 0.450±0.049 | 82.124 ±2.094 |
| | FIM-PP (zs) (NLL) | 21.736 ±0.115 | 1.141 ±0.010 | 0.449 ±0.002 | 120.894 ±0.393 |
| | FIM-PP (zs) (S-MAPE) | 21.418 ±0.142 | 1.198 ±0.008 | 0.445 ±0.002 | 118.944 ±0.487 |
| | FIM-PP (f) (NLL) | 18.428 ±0.124 | 1.091 ±0.016 | 0.361 ±0.001 | 87.264 ±0.323 |
| FIM-PP (f) (S-MAPE) | 18.566 ±0.072 | 1.153 ±0.007 | 0.352 ±0.002 | 82.555 ±0.278 | |
| RETWEET | HYPRO | 31.743±0.068 | 1.927±0.027 | 33.683±0.245 | 105.073±0.958 |
| | Dual-TPP | 31.652±0.075 | 1.963±0.038 | 28.104±0.486 | 106.721±0.774 |
| | A-NHP | 30.337 ±0.065 | 1.823±0.031 | 26.310±0.333 | 106.021±1.011 |
| | NHP | 30.817±0.090 | 1.713±0.024 | 27.010±0.429 | 107.053±1.390 |
| | IFTPP | 31.974±0.032 | 1.942±0.062 | 28.825±0.221 | 106.014±0.633 |
| | TCDDM | 32.006±0.074 | 1.789±0.094 | 29.124±0.405 | 106.738±0.791 |
| | CDiff | 31.237±0.078 | 1.745±0.036 | 26.429±0.201 | 105.767±0.771 |
| | FIM-PP (zs) (NLL) | 31.027 ±0.031 | 2.355 ±0.032 | 27.085 ±0.002 | 97.590 ±0.152 |
| | FIM-PP (zs) (S-MAPE) | 30.986 ±0.104 | 2.396 ±0.012 | 28.305 ±0.056 | 105.397 ±0.496 |
| | FIM-PP (f) (NLL) | 30.592 ±0.037 | 1.611 ±0.031 | 25.021 ±0.034 | 86.875 ±0.108 |
| FIM-PP (f) (S-MAPE) | 30.788 ±0.051 | 1.652 ±0.018 | 24.836 ±0.042 | 85.222 ±0.086 | |

Table 5: Prediction of 5 events in test sequences of five real-world datasets. Error-bars indicate the standard deviation over 10 trials. Results for the baseline methods were extracted from Zeng et al. (2024). Best results are bold.

| Dataset | Method | OTD | RMSE _e | RMSE _{Δt} | sMAPE _{Δt} |
|---------------------------|----------------------------|----------------------|----------------------|----------------------|-----------------------|
| TAXI | HYPRO | 5.952±0.126 | 0.500±0.011 | 0.322±0.004 | 85.994±0.227 |
| | Dual-TPP | 7.534±0.111 | 0.636±0.009 | 0.340±0.003 | 89.727±0.320 |
| | A-NHP | 6.441±0.090 | 0.682±0.010 | 0.347±0.002 | 89.070±0.152 |
| | NHP | 7.405±0.122 | 0.641±0.013 | 0.351±0.008 | 91.625±0.177 |
| | IFTPP | 7.209±0.184 | 0.608±0.008 | 0.335±0.003 | 90.512±0.169 |
| | TCDDM | 5.877±0.095 | 0.648±0.015 | 0.327±0.005 | 88.051±0.240 |
| | CDiff | 5.966±0.083 | 0.547±0.007 | 0.318±0.003 | 89.535±0.294 |
| | FIM-PP (zs) (NLL) | 6.773 ±0.064 | 0.655 ±0.013 | 0.246 ±0.001 | 74.912 ±0.793 |
| | FIM-PP (zs) (SMAPE) | 6.763 ±0.018 | 0.654 ±0.005 | 0.248 ±0.001 | 76.402 ±0.802 |
| | FIM-PP (f) (NLL) | 4.083 ±0.032 | 0.311 ±0.007 | 0.250 ±0.002 | 71.108 ±0.902 |
| FIM-PP (f) (SMAPE) | 4.103 ±0.029 | 0.315 ±0.005 | 0.248 ±0.001 | 73.181 ±0.409 | |
| TAOBAO | HYPRO | 11.317±0.111 | 0.817±0.037 | 0.573±0.011 | 133.837±0.524 |
| | Dual-TPP | 13.280±0.092 | 1.877±0.014 | 0.691±0.007 | 134.437±0.458 |
| | A-NHP | 11.223±0.145 | 0.873±0.023 | 0.550±0.014 | 132.266±0.532 |
| | NHP | 11.973±0.176 | 1.910±0.031 | 0.712±0.017 | 134.693±0.369 |
| | IFTPP | 11.052±0.108 | 1.941±0.049 | 0.601±0.017 | 126.320±0.591 |
| | TCDDM | 11.609±0.184 | 1.690±0.023 | 0.675±0.009 | 129.009±0.923 |
| | CDiff | 10.147 ±0.140 | 0.730 ±0.019 | 0.519 ±0.008 | 124.339 ±0.322 |
| | FIM-PP (zs) (NLL) | 15.951 ±0.042 | 1.129 ±0.007 | 1.761 ±0.013 | 168.299 ±0.249 |
| | FIM-PP (zs) (SMAPE) | 15.955 ±0.034 | 1.106 ±0.007 | 1.918 ±0.026 | 167.486 ±0.199 |
| | FIM-PP (f) (NLL) | 13.173 ±0.261 | 0.745 ±0.010 | 14.892 ±0.370 | 146.921 ±0.858 |
| FIM-PP (f) (SMAPE) | 13.572 ±0.166 | 0.759 ±0.005 | 17.384 ±0.416 | 150.562 ±0.830 | |
| STACKOVERFLOW | HYPRO | 11.590±0.186 | 0.586±0.019 | 1.227±0.018 | 109.014±0.422 |
| | Dual-TPP | 11.719±0.109 | 0.591±0.026 | 1.296±0.010 | 106.697±0.381 |
| | A-NHP | 11.595±0.197 | 0.575±0.009 | 1.188±0.014 | 105.799±0.516 |
| | NHP | 11.807±0.155 | 0.596±0.015 | 1.261±0.013 | 108.074±0.661 |
| | IFTPP | 13.124±0.174 | 0.702±0.008 | 1.182±0.039 | 108.409±0.692 |
| | TCDDM | 11.410±0.129 | 0.630±0.015 | 1.201±0.028 | 107.893±0.942 |
| | CDiff | 10.735±0.183 | 0.571±0.012 | 1.153±0.011 | 100.586±0.299 |
| | FIM-PP (zs) (NLL) | 11.520 ±0.057 | 0.657 ±0.003 | 1.030 ±0.001 | 93.296 ±0.506 |
| | FIM-PP (zs) (SMAPE) | 11.334 ±0.055 | 0.631 ±0.004 | 1.020 ±0.000 | 91.864 ±0.443 |
| | FIM-PP (f) (NLL) | 10.353 ±0.051 | 0.527 ±0.004 | 0.990 ±0.003 | 86.443 ±0.128 |
| FIM-PP (f) (SMAPE) | 10.341 ±0.013 | 0.525 ±0.007 | 0.984 ±0.003 | 86.133 ±0.212 | |
| AMAZON | HYPRO | 9.552±0.172 | 1.397±0.033 | 0.433±0.008 | 82.847±0.748 |
| | Dual-TPP | 11.309±0.093 | 1.742±0.302 | 0.476±0.010 | 86.633±0.573 |
| | A-NHP | 9.430 ±0.131 | 1.117±0.049 | 0.427±0.033 | 83.121±0.415 |
| | NHP | 11.273±0.198 | 1.431±0.024 | 0.501±0.009 | 90.591±0.667 |
| | IFTPP | 10.230±0.224 | 1.663±0.168 | 0.447±0.015 | 88.900±0.610 |
| | TCDDM | 10.557±0.331 | 1.409±0.203 | 0.460±0.032 | 82.401±0.810 |
| | CDiff | 9.478 ±0.081 | 1.326±0.082 | 0.424±0.018 | 81.287±0.994 |
| | FIM-PP (zs) (NLL) | 11.124 ±0.059 | 0.736 ±0.004 | 0.449 ±0.004 | 119.129 ±0.746 |
| | FIM-PP (zs) (SMAPE) | 11.029 ±0.033 | 0.735 ±0.005 | 0.445 ±0.003 | 117.793 ±0.635 |
| | FIM-PP (f) (NLL) | 10.034 ±0.060 | 0.737 ±0.006 | 0.341 ±0.004 | 78.738 ±0.339 |
| FIM-PP (f) (SMAPE) | 10.004 ±0.019 | 0.732 ±0.003 | 0.343 ±0.004 | 79.623 ±0.524 | |
| RETWEET | HYPRO | 16.145±0.096 | 1.105±0.026 | 27.236±0.259 | 103.052±1.206 |
| | Dual-TPP | 16.050±0.085 | 1.077±0.027 | 31.493±0.162 | 101.322±1.127 |
| | A-NHP | 16.124±0.089 | 1.058±0.029 | 29.247±0.145 | 105.930±1.380 |
| | NHP | 15.945±0.094 | 1.113±0.040 | 32.367±0.104 | 107.022±1.077 |
| | IFTPP | 16.043±0.222 | 1.313±0.011 | 30.853±0.119 | 106.941±2.031 |
| | TCDDM | 15.874±0.053 | 1.194±0.021 | 28.530±0.110 | 105.570±0.940 |
| | CDiff | 15.858±0.080 | 1.023 ±0.036 | 26.078±0.175 | 106.620±1.008 |
| | FIM-PP (zs) (NLL) | 15.747 ±0.032 | 1.342 ±0.027 | 28.138 ±0.068 | 98.668 ±0.794 |
| | FIM-PP (zs) (SMAPE) | 15.780 ±0.017 | 1.331 ±0.030 | 28.472 ±0.071 | 100.765 ±0.803 |
| | FIM-PP (f) (NLL) | 15.645 ±0.020 | 1.033 ±0.034 | 25.308 ±0.135 | 83.010 ±0.278 |
| FIM-PP (f) (SMAPE) | 15.585 ±0.040 | 1.012 ±0.040 | 25.413 ±0.086 | 84.945 ±0.343 | |

Table 6: Summary of the parametrized base intensities and interaction kernels of Hawkes processes used in our synthetic data generation. The parameters of each configuration are sampled from uniform distributions, covering a wide range of processes.

| Dataset Configuration | Base Intensity $\mu(t)$ | Interaction Kernel $\gamma(t)$ | Parameter Distributions |
|------------------------------------------------------|----------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Constant Base & Exponential Kernel (no interactions) | Constant $\mu(t) = c_0$ | Exponential Decay $\gamma(t) = \alpha e^{-\beta t}$ ($\gamma_{ij, i \neq j} = 0$) | $c_0 \sim \mathcal{U}(0.01, 1.3)$ $\alpha \sim \mathcal{U}(0.005, 1.0)$ $\beta \sim \mathcal{U}(0.001, 10.0)$ |
| Constant Base & Exponential Kernel | Constant $\mu(t) = c_0$ | Exponential Decay $\gamma(t) = \alpha e^{-\beta t}$ | $c_0 \sim \mathcal{U}(0.01, 1.3)$ $\alpha \sim \mathcal{U}(0.005, 1.0)$ $\beta \sim \mathcal{U}(0.001, 10.0)$ |
| Sinusoidal Base & Exponential Kernel | Sinusoidal $\mu(t) = A \sin(\omega(t - \gamma)) + c_0$ | Exponential Decay $\gamma(t) = \alpha e^{-\beta t}$ | $c_0 \sim \mathcal{U}(0.05, 0.15)$ $A \sim \mathcal{U}(0.0, 10.0)$ $\omega \sim \mathcal{U}(0.1, 15.0)$ $\gamma \sim \mathcal{U}(0.0, 5.0)$ $\alpha \sim \mathcal{U}(0.1, 0.6)$ $\beta \sim \mathcal{U}(0.8, 2.0)$ |
| Gamma Base & Exponential Kernel | Gamma Shape + Constant $\mu(t) = At^p e^{-\beta_0 t} + c_0$ | Exponential Decay $\gamma(t) = \alpha e^{-\beta_1 t}$ | $c_0 \sim \mathcal{U}(0.1, 1.3)$ $A \sim \mathcal{U}(10.0, 50.0)$ $p \sim \mathcal{U}(1.0, 2.0)$ $\beta_0 \sim \mathcal{U}(1.0, 10.1)$ $\alpha \sim \mathcal{U}(0.005, 1.0)$ $\beta_1 \sim \mathcal{U}(0.001, 10.0)$ |
| Poisson Process | Constant $\mu(t) = c_0$ | Zero Kernel $\gamma(t) = 0$ | $c_0 \sim \mathcal{U}(0.01, 1.3)$ |
| Constant Base & Rayleigh Kernel | Constant $\mu(t) = c_0$ | Rayleigh $\gamma(t) = a_0 \frac{(t - t_{\text{shift}})}{a_1^2} \exp\left(-\frac{(t - t_{\text{shift}})^2}{2a_1^2}\right)$ | $c_0 \sim \mathcal{U}(0.01, 1.3)$ $a_0 \sim \mathcal{U}(0.001, 1.0)$ $a_1 \sim \mathcal{U}(0.05, 0.25)$ $t_{\text{shift}} \sim \mathcal{U}(0.0, 0.1)$ |

Table 7: For each dataset configuration, we sample Hawkes processes with varying numbers (#) of marks, sequences and events per sequence.

| # Marks | # Samples | # Sequences | # Events per Sequence |
|---------|-----------|-------------|-----------------------|
| 1 | 1000 | 2000 | 100 |
| 5 | 1000 | 2000 | 100 |
| 10 | 1000 | 2000 | 100 |
| 15 | 1000 | 2000 | 100 |
| 22 | 5000 | 2000 | 100 |