Probabilistic Querying of Continuous-Time Event Sequences

Alex Boyd[1] **Yuxin Chang**[2] **Stephan Mandt**[1,2] **Padhraic Smyth**[1,2]
[1]Department of Statistics [2]Department of Computer Science
University of California, Irvine

Abstract

Continuous-time event sequences, i.e., sequences consisting of continuous time stamps and associated event types ("marks"), are an important type of sequential data with many applications, e.g., in clinical medicine or user behavior modeling. Since these data are typically modeled in an autoregressive manner (e.g., using neural Hawkes processes or their classical counterparts), it is natural to ask questions about future scenarios such as "what kind of event will occur next" or "will an event of type A occur before one of type B." Addressing such queries with direct methods such as naive simulation can be highly inefficient from a computational perspective. This paper introduces a new typology of query types and a framework for addressing them using importance sampling. Example queries include predicting the n^{th} event type in a sequence and the hitting time distribution of one or more event types. We also leverage these findings further to be applicable for estimating general "A before B" type of queries. We prove theoretically that our estimation method is effectively always better than naive simulation and demonstrate empirically based on three real-world datasets that our approach can produce orders of magnitude improvements in sampling efficiency compared to naive methods.

1 Introduction

Continuous-time event data occurs across a wide range of applications and areas such as user behavior modeling (Mishra et al., 2016; Kumar et al., 2019), finance (Bacry et al., 2012; Hawkes, 2018), and healthcare (Nagpal et al., 2021; Chiang et al., 2022). The data typically consists of sets of variable-length sequences where each sequence is a set of ordered events, and each event is associated with a continuous time-stamp and a categorical event type. Such data are often modeled as marked temporal point processes (MTPPs), and a broad variety of modeling frameworks have been successfully developed both in the statistical literature (e.g., Hawkes processes (Hawkes, 1971)) and in the machine learning literature (e.g., neural MTPP models (Mei and Eisner, 2017)). These MTPP modeling frameworks provide a general and flexible setup for making one-step-ahead predictions such as the timing and/or type of the next event time, conditioned on a partial history of sequence.

In this paper we look beyond one-step ahead predictions and instead investigate how to efficiently answer queries that involve more complex statements about future events and their timing. Such queries include hitting time queries ("what is the probability that at least one event of type A will occur before time t"), queries of the form "what is the probability that A will occur before B," as well as computing the marginal distribution of event types for the n^{th} next event (irrespective of time). These types of queries are useful across a variety of applications, such as making predictions conditioned on a patient's medical and treatment history, or conditioned on a customer's page view and purchase history.

However, exact computation of such queries is intractable in general except in the case of simple parametric models, such as Poisson processes. For a standard MTPP model to directly answer such queries requires that all intervening events (from current time to the event(s) of interest in the query) are marginalized over. In particular, this involves marginalizing over both the combinatorially-large space of possible event types as well as the uncountably infinite space of possible event timings. While direct simulation of future trajectories from a model provides one avenue for answering such queries (e.g., see Daley and Vere-Jones (2003)) these "naive" methods can be very inefficient (both statistically and computationally), as we will demonstrate later in the paper. More efficient alternative approaches (to the naive simulation method) appear to be completely unexplored (to our knowledge), for both neural and non-neural MTPP models.

We develop a general query framework based on impor-

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

tance sampling that enables efficient estimates of various types of queries. In our approach, we first transform each query into unified forms and then derive the distribution of interest as functions of type-specific intensities (expected instantaneous rates of occurrence). Our proposed novel marginalization scheme empowers real-time computation of probabilistic queries, with proven higher efficiency compared to naive estimates. Furthermore, experiments on three real-world datasets in different domains demonstrate that our proposed estimation method is significantly more efficient than the naive estimate in practice. For example, for hitting time queries with neural Hawkes processes, we show an average magnitude of 10^3 reduction in estimator variance.

Our approach for answering probabilistic queries is general-purpose in the sense that it can be integrated with any intensity-based black-box MTPP model, either parametric or neural. To summarize, our main contributions are:

- We identify and formalize a general class of probabilistic queries that cover a wide range of queries of interest, such as the distribution of the first occurrence of certain event types (hitting times), the n^{th} occurring event type irrespective of time (marginal mark queries), or queries addressing the order of event types ("A before B" queries).
- Within this class of queries, we develop a novel proposal distribution for importance sampling. This distribution is easy to sample from, simple to evaluate likelihoods with, and results in guaranteed increases in efficiency when compared to existing estimation techniques.
- We evaluate our proposed estimation technique across three real-world user behavior datasets, as well as on simulated data. In all cases, we find dramatic reductions in estimator variance compared to existing methods—often times by several orders of magnitude.

2 Related Work

A large variety of MTPP models have been developed over recent decades, aimed at modeling sequences of marked event data with varying sorts of behaviors. This behavior has been both explicitly modeled with parametric MTPP models (Isham and Westcott, 1979; Daley and Vere-Jones, 2003), and implicitly modeled using neural network-based methods (Du et al., 2016; Biloš et al., 2019; Shchur et al., 2020; Enguehard et al., 2020; Zuo et al., 2020). Of particular note in these categories are the self-exciting Hawkes process (Hawkes, 1971; Liniger, 2009) and the neural Hawkes process (Mei and Eisner, 2017). The majority of neural MTPP models utilize some form or extension of recurrent neural networks to model conditional intensity functions (or equivalent transformations thereof). MTPP models have been broadly applied to next event prediction across a number of different application areas: seismology (Ogata, 1998), finance (Bacry et al., 2012; Hawkes, 2018), social media behavior (Mishra et al., 2016; Rizoiu et al., 2017), and medical outcomes (Cox, 1972; Andersen et al., 2012).[1] Neural-based methods have also been successful at additional tasks such as imputing missing data (Shchur et al., 2020; Mei et al., 2019; Gupta et al., 2021), sequential representation learning (Shchur et al., 2020; Boyd et al., 2020), and long-term forecasting (Deshpande et al., 2021).

Answering probabilistic queries in some capacity has been previously explored at a model-specific level. Primary examples include continuous-time Markov processes (Shelton and Ciardo, 2014), continuous-time Bayesian networks (Nodelman et al., 2002; Fan et al., 2010), and Markovian self-exciting processes (Oakes, 1975). In this prior work, the assumed parametric form of the model allows for analytic forms of specific queries under certain conditions. For instance, the Markovian self-exciting process provides a representation that makes estimating hitting time queries directly tractable.

However, to the best of our knowledge, apart from the naive sampling approach (e.g., Daley and Vere-Jones (2003)), there is no existing work on answering general probabilistic queries (such as hitting time of a collection of event types) for black-box MTPP models, which is the focus of this paper. For discrete-time models, estimating these queries has been investigated in our prior work (Boyd et al., 2022), and while there does not exist a direct mapping of those techniques to continuous time, this previous work will serve as a large source of inspiration for what we propose in this paper.

3 Preliminaries

3.1 Notation for Event Sequences

Let $\tau_1, \tau_2, \cdots \in \mathbb{R}_{\geq 0}$ be a sequence of continuous random variables with the constraint that $\forall_i : \tau_i < \tau_{i+1}$. These represent the time of occurrence for events of interest. Each event has an associated categorical value, such as a label or a location, that is referred to as a *mark*. An event is jointly represented as (i) a time of occurrence τ_i and (ii) an associated mark random variable $\kappa_i \in \mathbb{M}$. In this work we will focus on the finite discrete setting of a fixed vocabulary for marks: $\mathbb{M} = \{1, 2, \ldots, K\}$, although more generally the mark space $\mathbb{M}$ can be defined on a variety of different domains.

Let the *sequence* of events over a specified time range $[a, b] \subset \mathbb{R}_{\geq 0}$ be denoted as

$$\mathcal{S}[a, b] = \{(\tau_i, \kappa_i) \,|\, \tau_i \in [a, b] \text{ for } i \in \mathbb{N}\}.$$

[1] Survival analysis is a special case of temporal point processes where the event of interest can only occur once.

with similar definitions for $\mathcal{S}(a, b]$ and $\mathcal{S}[a, b)$. For simplicity, we will let $\mathcal{S}(t)$ be shorthand for $\mathcal{S}[0, t)$ such that $\mathcal{S}(\tau_i) = \{(\tau_1, \kappa_1), \ldots, (\tau_{i-1}, \kappa_{i-1})\}$.[2] We will use $\mathcal{S}_k$ to refer to *mark-specific* sequences, i.e., $\mathcal{S}_k(t) = \{(\tau_i, \kappa_i) \in \mathcal{S}(t) \,|\, \kappa_i = k\}$.

3.2 Marked Temporal Point Processes

The generative mechanism for these point patterns are generally referred to as *marked temporal point processes* (MTPPs). MTPP models are capable of approximating the distribution of a given sequence of N events, $p(\mathcal{S}[0, \tau_N])$.[3] These models are typically constructed in an autoregressive fashion,

$$p(\mathcal{S}[0, \tau_N]) = \prod_{i=1}^{N} p(\tau_i, \kappa_i \,|\, \mathcal{S}[0, \tau_{i-1}]),$$

where the distribution for the next event (τ_i, κ_i) conditioned on the preceding terms is modeled with the expected instantaneous rate of change for each mark. This is referred to as the *marked intensity function* and is defined formally as

$$\lambda_k(t \,|\, \mathcal{S}(t))dt := \mathbb{E}_p \left[\mathbb{1}(|\mathcal{S}_k[t, t + dt)| = 1) \,|\, \mathcal{S}(t)\right]$$

where $\mathbb{1}(\cdot)$ is the indicator function and $\mathbb{E}_p$ is the expected value with respect to distribution p. For brevity, we typically use the following $*$ convention to suppress the conditional: $\lambda_k^*(t) := \lambda_k(t \,|\, \mathcal{S}(t))$. Note that these functions not only condition on the preceding events, but also on the fact that no events have occurred since the last event up until time t, i.e., $p(\cdot \,|\, \mathcal{S}[0, t)) \neq p(\cdot \,|\, \mathcal{S}[0, \tau_{i-1}])$.

The total intensity function, $\lambda^*(t) := \sum_{k \in \mathbb{M}} \lambda_k^*(t)$, is sufficient to describe the timing of the next event τ_i. The distribution of the mark conditioned on the timing of the next event is naturally described as $p(\kappa_i = k \,|\, \tau_i = t) \equiv \frac{\lambda_k^*(t)}{\lambda^*(t)}$. We will be assuming that the native output of any model we are working with will produce a vector of marked intensity functions over the mark space $\mathbb{M}$ evaluated at time t. Any MTPP with a defined set of marked intensity functions can be easily sampled from by utilizing a thinning procedure (Ogata, 1981), if not directly.

Lastly, the likelihood of a given sequence $\mathcal{S}$ of length N over an observation window $[0, T]$ can be computed in terms of intensity values:

$$p(\mathcal{S}[0, T]) = \left(\prod_{i=1}^{N} \lambda_{\kappa_i}^*(\tau_i)\right) \exp\left(-\int_0^T \lambda^*(s)ds\right).$$

[2] Note that the majority of the point process literature refers to this sequence as a *history* of events and is represented via $\mathcal{H}$. We forgo this traditional terminology and notation to emphasize that our work is primarily about estimating queries for *future* events.

[3] For brevity and consistent notation, we will be using $p(\cdot)$ in reference to both probability densities and masses when appropriate.

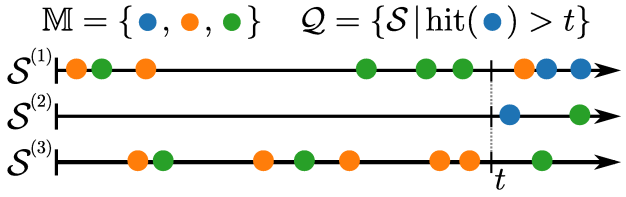

Figure 1: Example query space $\mathcal{Q}$ for the hitting time (first occurrence) of blue marks being greater than some time t. Sequences shown, $\mathcal{S}^{(i)}$, all belong to the query space as they each do not contain a blue event occurring before time t.

4 Querying MTPPs

We are interested in evaluating probabilistic statements, or rather *queries*, on any MTPP model p, e.g., a model trained from data. Furthermore, we are interested in evaluating queries that are conditioned on a partially observed sequence (e.g., "what is the likelihood that at least one event of type A will occur in the next year given a patient's medical history?").

Formally, we define a probabilistic query as a probability statement of the form

$$p(\mathcal{S} \in \mathcal{Q}) \text{ where } \mathcal{Q} \subset \Omega_{\mathcal{S}} \equiv \text{Sample Space of } \mathcal{S},$$

where p is the model's distribution over future event sequences. We refer to $\mathcal{Q}$ as the *query space*. The contents of the query space naturally will vary depending on the query at hand. An example query space and a subset of associated valid sequences can be seen in Fig. 1. It is worth noting that in most contexts, the cardinality of $\mathcal{Q}$ will be uncountably infinite.

This section will begin by discussing what probabilistic queries are readily available and tractable for a given model. Following this, we will present a novel class of queries, of which include hitting time and marginal mark queries, as well as an importance sampling estimation procedure. Finally, we will discuss "A before B" queries and how to efficiently estimate them under our novel framework. *Without loss of generality, we suppress the notation for conditioning on partially observed sequences and present all derivations and notations for unconditional queries.*

4.1 Directly Tractable Queries

Due to the model's autoregressive nature, queries about the immediate next event of a sequence are the only types of queries that can be directly evaluated without marginalization. We will now present the two main types for MTPPs.

Marginal Distribution of Next Event Time In general, it can be shown that $\lambda^*(t) = \frac{f_{\tau_i}(t \,|\, \mathcal{S}[0, \tau_{i-1}])}{1 - F_{\tau_i}(t \,|\, \mathcal{S}[0, \tau_{i-1}])}$ where $t \in (\tau_{i-1}, \tau_i]$, f_{τ_i} is the probability density function (PDF)

of τ_i, and F_{τ_i} is the and cumulative density function (CDF) of τ_i. By recognizing that $\lambda^*(t) = -\frac{d}{dt}\log(1 - F_{\tau_i}(t \,|\, \mathcal{S}[0, \tau_{i-1}]))$, we find that the CDF of the next event timing τ_1 is

$$p(\tau_1 \leq t) := F_{\tau_1}(t) = 1 - \exp\left(-\int_0^t \lambda^*(s)ds\right).$$

Differentiating this result with respect to t yields the PDF: $f_{\tau_1}(t) = \lambda^*(t)\exp\left(-\int_0^t \lambda^*(s)ds\right)$. Note that we only immediately have access to the analytical form of the first future event timing τ_1. To achieve the same results for τ_i in general would require marginalizing over all $i-1$ events which is rather cumbersome to do exactly.

Marginal Distribution of Next Mark Let $A \subset \mathbb{M}$. It follows then that the probability of the first event having a mark in A is computed as follows:

$$\begin{aligned} p(\kappa_1 \in A) &= \int_0^\infty p(\kappa_1 \in A \,|\, \tau_1 = t) f_{\tau_1}(t)dt \\ &= \int_0^\infty \frac{\lambda_A^*(t)}{\lambda^*(t)} \lambda^*(t) \exp\left(-\int_0^t \lambda^*(s)ds\right) dt \\ &= \int_0^\infty \lambda_A(t) \exp\left(-\int_0^t \lambda^*(s)ds\right) dt, \end{aligned}$$

where $\lambda_A^*(t) = \sum_{k \in A} \lambda_k^*(t)$. Replacing the outer integration bounds of $[0, \infty)$ with $[a, b]$ gives the joint query $p(\tau_1 \in [a, b], \kappa_1 \in A)$.

Both of these different queries can potentially be computed analytically if the form of λ^* permits, otherwise they can be estimated using approximate integration techniques.

4.2 Naive Estimation of Queries

When considering more complex queries, for example those that deal with sequences of events or those far in the future, it becomes necessary to rely on simulating potential trajectories in order to estimate their values. This is due to the fact that exactly representing a probabilistic query in terms of intensity values involves many nested integrals (for each potential interim event), potentially an infinite amount of them depending on the query.

The de facto method for approximating arbitrary probabilistic queries involves generating sequences and computing the relative frequency for which the query condition is met in the sampled sequences (Daley and Vere-Jones, 2003). This can be seen as a Monte Carlo estimate with the following formulation:

$$p(\mathcal{S}(T) \in \mathcal{Q}) = \mathbb{E}_p\left[\mathbb{1}(\mathcal{S}(T) \in \mathcal{Q})\right],$$

where $\mathbb{1}(\cdot)$ is the indicator function. We refer to this procedure as "naive" estimation because this does not take into account any information about the query when sampling.

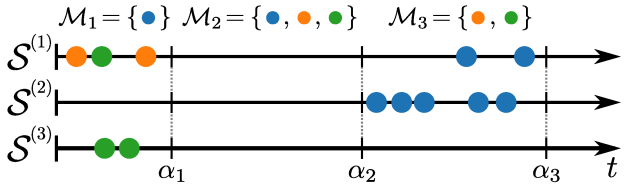

Figure 2: Three potential sequences $\mathcal{S}^{(i)}$ that satisfies the condition of an example restricted-mark query. The mark space $\mathbb{M}$ in this context is equivalent to that in Fig. 1.

4.3 General Restricted-Mark Queries

One way to improve upon the naive procedure is to leverage information about the query in a proposal distribution in conjunction with importance sampling. To do so though, we must first constrain ourselves to a specific class of query being considered. Additionally, this class of interest should take into account different aspects of sampling sequences using MTPPs for our proposal distribution. Namely, these models can easily be forced to *not* sample events of specific types over a period of time (e.g., set $\lambda_A^*(t) := 0$ for some time interval). Conversely, it is not immediately obvious how to *encourage* or *force* an event to occur within a specified time range.

As such, a natural class of queries can be seen in which over one or more specified spans of time we restrict what types of events are allowed and not allowed to occur. We term this class as "general restricted-mark queries."

We will now more formally define this class of queries. Consider positive real values $\alpha_1, \ldots, \alpha_n$ such that $\alpha_i < \alpha_{i+1}$. These values naturally split the timeline $\mathbb{R}_{\geq 0}$ into $n+1$ spans: $[0, \alpha_1], (\alpha_1, \alpha_2], \ldots, (\alpha_{n-1}, \alpha_n], (\alpha_n, \infty)$. Furthermore, let the subsets $\mathcal{M}_i \subseteq \mathbb{M}$ for $i = 1, \ldots, n$ represent restricted mark spaces for the first n spans. The class of queries is concerned with how likely sequences spanning $[0, \alpha_n]$ respect the restricted mark spaces in each associated interval:

$$\begin{aligned} &p\left(\cup_{i=1}^n \{\text{No events with types } \mathcal{M}_i \text{ in } t \in (\alpha_{i-1}, \alpha_i]\}\right) \\ &= p\left(\wedge_{i=1}^n \forall_{(\tau,\kappa) \in \mathcal{S}(\alpha_{i-1}, \alpha_i]} \kappa \notin \mathcal{M}_i\right) \text{ with } \alpha_0 = 0. \quad (1) \end{aligned}$$

See Fig. 2 for an illustrated example query. This is a very flexible class of queries that includes many meaningful individual queries, which will be further discussed in Section 4.4.

Importance Sampling and Proposal Distribution Let q be a proposal distribution with support over at least the intersection of the support of p and the query space $\mathcal{Q}$ (i.e., $\text{supp}(q) \supseteq \text{supp}(p) \cap \mathcal{Q}$). It then follows that

$$\mathbb{E}_p\left[\mathbb{1}(\mathcal{S}(T) \in \mathcal{Q})\right] = \mathbb{E}_q\left[\mathbb{1}(\mathcal{S}(T) \in \mathcal{Q}) \frac{p(\mathcal{S}(T))}{q(\mathcal{S}(T))}\right]. \quad (2)$$

It can be shown that the optimal proposal distribution (i.e., lowest estimator variance) takes the form (Robert and Casella, 2004):

$$q_{\text{optimal}}(\mathcal{S}(T)) := \frac{|\mathbb{1}(\mathcal{S}(T) \in \mathcal{Q})|p(\mathcal{S}(T))}{\mathbb{E}_p[|\mathbb{1}(\mathcal{S}(T) \in \mathcal{Q})|]} = p(\mathcal{S}(T) \,|\, \mathcal{S}(T) \in \mathcal{Q}),$$

however, this is not immediately usable since it involves computing the exact query that we are trying to estimate in the first place.

The more our actual proposal distribution q resembles q_{optimal}, the more efficient our estimation procedure will be. Since conditioning on future events is difficult for neural autoregressive models, we can instead only apply immediate "local" restrictions on the trajectory such that a sequence will remain within $\mathcal{Q}$. This can be accomplished by letting q be a MTPP with intensity

$$\mu_k^*(t) = \mathbb{1}(k \notin \mathcal{M}_i)\lambda_k^*(t)$$

for $k \in \mathcal{M}$ and $t \in (\alpha_{i-1}, \alpha_i]$. Note that this can be seen as the natural extension of the proposal distribution in Boyd et al. (2022) to continuous time. This naturally leads to the likelihood of any sequence generated under q as being

$$q(\mathcal{S}[0,T]) = \left(\prod_{i=1}^{N} \mu^*_{\kappa_i}(\tau_i)\right) \exp\left(-\int_0^T \mu^*(s)ds\right) = \left(\prod_{i=1}^{N} \lambda^*_{\kappa_i}(\tau_i)\right) \exp\left(-\sum_{i=1}^{n} \int_{\alpha_{i-1}}^{\alpha_i} \lambda^*_{\mathbb{M}\setminus\mathcal{M}_i}(s)ds\right)$$

where $N = |\mathcal{S}[0,T]|$. This proposal distribution was constructed so that every sample generated will always belong to the query space. Applying this to Eq. (2) yields

$$p(\mathcal{S}(T) \in \mathcal{Q}) = \mathbb{E}_q\left[\exp\left(-\sum_{i=1}^{n} \int_{\alpha_{i-1}}^{\alpha_i} \lambda^*_{\mathcal{M}_i}(s)ds\right)\right]. \quad (3)$$

Any query in this class can now be estimated in an unbiased fashion by using Monte Carlo estimation on Eq. (3).

Estimator Efficiency Since both the naive and importance sampled estimators are unbiased, whichever has lower variance can be seen as the more *efficient* estimator.

Assume that $\mathcal{Q}$ belongs to a general restricted-mark query and that $\pi = p(\mathcal{S}(T) \in \mathcal{Q})$. Let

$$\hat{\pi}_{\text{Naive}}(\mathcal{S}(T)) = \mathbb{1}(\mathcal{S}(T) \in \mathcal{Q}),$$
$$\hat{\pi}_{\text{Imp.}}(\mathcal{S}(T)) = \exp\left(-\sum_{i=1}^{n} \int_{\alpha_{i-1}}^{\alpha_i} \lambda^*_{\mathcal{M}_i}(s)ds\right),$$

where both are unbiased estimators of π under p and q respectively. Note that $\hat{\pi}_{\text{Imp.}}(\cdot) \in [0,1]$ as $\lambda_k^*(\cdot) \geq 0$. Finally, let relative efficiency of the two estimators be defined as

$$\text{eff}(\hat{\pi}_{\text{Imp.}}, \hat{\pi}_{\text{Naive}}) := \frac{\text{Var}_p\left[\hat{\pi}_{\text{Naive}}(\mathcal{S}(T))\right]}{\text{Var}_q\left[\hat{\pi}_{\text{Imp.}}(\mathcal{S}(T))\right]}.$$

Theorem 1. *If $\pi \in (0,1)$ and $\lambda^*(t) < \infty$ for all $t \in [0,T]$, then eff$(\hat{\pi}_{Imp.}, \hat{\pi}_{Naive}) > 1$. In other words, under these conditions $\hat{\pi}_{Imp.}$ is always more efficient than $\hat{\pi}_{Naive}$.*

Proof. Since the naive estimator is unbiased and binary, then it follows that $\hat{\pi}_{\text{Naive}}(\mathcal{S}(T)) \sim \text{Bern}(\pi)$. Thus, $\text{Var}_p\left[\hat{\pi}_{\text{Naive}}(\mathcal{S}(T))\right] = \pi - \pi^2$.

To approach the variance of the importance sampling estimator, we note that

$$\begin{aligned}\text{Var}_p\left[\hat{\pi}_{\text{Imp.}}(\mathcal{S}(T))\right] &= \mathbb{E}_q\left[\hat{\pi}^2_{\text{Imp.}}\right] - \mathbb{E}_q\left[\hat{\pi}_{\text{Imp.}}\right]^2 \\ &= \mathbb{E}_q\left[\hat{\pi}^2_{\text{Imp.}}\right] - \pi^2 \\ &\leq \mathbb{E}_q\left[\hat{\pi}_{\text{Imp.}}\right] - \pi^2 \text{ since } \hat{\pi}_{\text{Imp.}} \in [0,1] \\ &= \pi - \pi^2\end{aligned}$$

The equality only holds if $\pi \in \{0,1\}$ or $\hat{\pi}_{\text{Imp.}} \sim \text{Bern}(\pi)$. The latter condition is due to the fact that for $[0,1]$ bounded random variables with mean π, if the variance is equal to $\pi - \pi^2$ then this implies it is Bernoulli (see Appendix for proof). However, when $\pi \in (0,1)$ then unless $\lambda^*(t) = \infty$ for some subset of $[0,T]$ it is impossible for $\hat{\pi}_{\text{Imp.}}(\mathcal{S}(T))$ to equal 0. Thus, outside of those circumstances the inequality is strict and $\text{eff}(\hat{\pi}_{\text{Imp.}}, \hat{\pi}_{\text{Naive}}) > 1$. □

4.4 Practical Estimation of Complex Queries

We will now apply our findings from Section 4.3 to produce estimators for three different complex, probabilistic queries.

Marginal Distribution of Hitting Time Let $A \subset \mathbb{M}$ and $A \neq \emptyset$. The first occurrence of an event with type $k \in A$, regardless of events of other types, is referred to as the *hitting time* of A or $\text{hit}(A)$. The probabilistic query of the CDF of the hitting time of A at a specific time t can be seen as a query under the general restricted-mark class:

$$\begin{aligned}p(\text{hit}(A) \leq t) &= 1 - p(\text{hit}(A) > t) \\ &= 1 - p(\{\text{No events of types } A \text{ in } [0,t]\}) \\ &= 1 - p(\forall_{(\tau,\kappa)\in\mathcal{S}[0,t]} \kappa \notin A) \\ &= 1 - \mathbb{E}_q\left[\exp\left(-\int_0^t \lambda^*_A(s)ds\right)\right].\end{aligned}$$

Note that this derivation relies on this query being a special case of the general framework outlined in Eq. (1) where $n = 1$, $\alpha_0 = 0$, $\alpha_1 = t$, and $\mathcal{M}_1 = A$. Interestingly, the importance sampled result of this query greatly resembles the CDF of the general first event timing: $F_{\tau_1}(t) = 1 - \exp\left(-\int_0^t \lambda^*(s)ds\right)$.[4] Furthermore, should $A = \mathbb{M}$ then we recover $F_{\tau_1}(t)$ as the estimator becomes deterministic (due to $\mu^*(t) = 0 \implies q(\mathcal{S}) \propto \mathbb{1}(\mathcal{S} = \emptyset)$).

[4] It is important to remember that in the general case, we must marginalize over possible trajectories for other types of events A' as these can all either potentially influence the intensity of events of type A.

Marginal Distribution of n^{th} Mark Let $A \subset \mathbb{M}$ and $n \geq 1$. The distribution of the marginal n^{th} mark describes how likely it is that the n^{th} event has a mark $k \in A$, irrespective of the timing of itself or of any of the $n-1$ events that occurred prior. In contrast to hitting time queries, we do not fix the integration bounds but rather sample them to be the timings of the τ_{n-1} and τ_n. In doing so, this query falls under the general mark-restricted framework:

$$\begin{aligned} p(\kappa_n \in A) &= p(\{\text{No events of types } A' \text{ in } (\tau_{n-1}, \tau_n]\}) \\ &= p(\forall_{(\tau,\kappa)\in\mathcal{S}(\tau_{n-1},\tau_n]} \kappa \notin A') \\ &= \mathbb{E}_q \left[\exp\left(- \int_{\tau_{n-1}}^{\tau_n} \lambda^*_{A'}(s) ds \right) \right] \end{aligned}$$

where $A' = \mathbb{M} \setminus A$. This can be seen as a special case under Eq. (1) where $\alpha_0 = 0$, $\alpha_i = \tau_i$ for $i = 1, \ldots, n$, $\mathcal{M}_1, \ldots, \mathcal{M}_{n-1} = \emptyset$, and $\mathcal{M}_n = A'$. Tying the values of the boundaries α_i to the random event times τ_i effectively ensures that each span with a restricted vocabulary $\mathcal{M}_i$ only pertains to the occurrence of one event. In doing so, we actually recover the ability to estimate queries purely concerning the marks, similar to the discrete sequence setting (Boyd et al., 2022).

It is worth noting that, interestingly, we can also compute the complement under the same framework as $p(\kappa_n \in A) = 1 - \mathbb{E}_q \left[\exp\left(- \int_{\tau_{n-1}}^{\tau_n} \lambda^*_A(s) ds \right) \right]$.

"A before B" Queries The last class of queries we will discuss are what we refer to as "A before B" queries. To be precise, we are interested in the probability of an event with some type $k \in A$ occurring before an event with some type $k \in B$ where $A \cap B = \emptyset$ and non-empty $A, B \subset \mathbb{M}$. In math, this is formally represented as $p(\text{hit}(A) < \text{hit}(B))$.

Surprisingly, with our previous developments we can actually estimate this query using importance sampling in conjunction with proposal distribution q. For the proposal distribution, let $\mu^*_k(t) = \mathbb{1}(k \notin A \cup B)\lambda^*_k(t)$. It then can be shown that

$$\begin{aligned} &p(\text{hit}(A) < \text{hit}(B)) \\ &\quad = 1 - \mathbb{E}_q \left[\int_0^\infty \lambda^*_B(t) \exp\left(- \int_0^t \lambda^*_{A\cup B}(s) ds \right) dt \right] \\ &\quad = \mathbb{E}_q \left[\int_0^\infty \lambda^*_A(t) \exp\left(- \int_0^t \lambda^*_{A\cup B}(s) ds \right) dt \right] \end{aligned} \tag{4}$$

with both expressions being equal due to the complement $1 - p(\text{hit}(A) > \text{hit}(B))$ also being estimable under this derivation. See the Appendix for derivations.

Interestingly, just like the parallels between the hitting time CDF and the first event time CDF, there exist similar comparisons for Eq. (4) and the analytical form of the marginal distribution for the first mark $p(\kappa_1 \in A) = \int_0^\infty \lambda^*_A(t) \exp\left(- \int_0^t \lambda^*(s) \right) dt$. Additionally, should $B = A'$ then the estimator becomes deterministic and we recover the form of $p(\kappa_1 \in A)$.

Note that the expectations in Eq. (4) are with respect to $\mathcal{S}(\infty) \sim q$, which is naturally not possible to evaluate; however, since the integrands are non-negative we can compute natural lower and upper bounds by sampling $\mathcal{S}(T) \sim q$ and integrating over $[0, T]$ instead of $[0, \infty)$. Lastly, since these bounds utilize the same proposal distribution, we can actually compute both at the same time for a little extra computation. It then follows that a good estimate for $p(\text{hit}(A) < \text{hit}(B))$ would be an average of the upper and lower bounds:

$$\begin{aligned} &p(\text{hit}(A) < \text{hit}(B)) \approx \\ &\frac{1}{2} + \mathbb{E}_q \left[\int_0^T \frac{\lambda^*_A(t) - \lambda^*_B(t)}{2} \exp\left(- \int_0^t \lambda^*_{A\cup B}(s) ds \right) dt \right], \end{aligned} \tag{5}$$

where $T > 0$ can either be set as a constant or could be dynamically determined on a per sequence basis based on some precision threshold. Since T is truncated, this estimate is no longer unbiased.

5 Experiments

We investigate the effectiveness of our novel importance sampling regime in the context of estimating hitting time, "A before B," and marginal mark distribution queries, while conditioning on partially observed sequences. We find that across both synthetic and real settings as well as parametric and neural-network-based models that our importance sampling estimator dramatically reduces variance compared to naive sampling and results in a much lower error on average. Furthermore, we demonstrate that, on average, these gains in performance outweigh any potential increases in computation time.

Ground Truth Computation of any arbitrary query $p(\mathcal{S}(T) \in \mathcal{Q})$ to arbitrary precision is intractable in the general case. Given this, in our experiments we compute our queries with an unbiased estimator to high precision using a large amount of computation, with much higher precision than any of the methods and scenarios evaluated for a given experiment. We refer to the result of this high-precision computation as "ground truth" below.

Metrics of Interest There are two primary metrics with which we judge query estimation procedures: mean relative absolute error and relative efficiency (or variance reduction should one of the estimators be biased). The former is defined as the mean of $|\pi - \hat{\pi}|/\pi$, where $\pi = p(\mathcal{S}(T) \in \mathcal{Q})$ and $\hat{\pi}$ is some estimator of π, over different queries (and potentially models). This particular form of error is chosen to offset the fact that $\pi \in [0, 1]$, which can lead to naturally closer estimates should π be close to 0 or 1. The latter metric

Table 1: Real-world Dataset Summary Statistics

Dataset	# Sequences	T_{max}	# Marks
MovieLens	34,935	43,000	182
MOOC	6,863	715	97
Taobao	17,777	192	1,000

of interest is the relative efficiency (or variance reduction) of importance sampling compared to naive sampling. This is calculated by dividing the variance of the naive estimator (calculated using ground truth: $\pi(1-\pi)$) by the variance of the importance sampled estimator (calculated empirically). As an example, a value of 5 for this metric indicates that, on average, 5 times as many samples are needed for naive estimation to achieve an estimator variance as low as that of importance sampling.

5.1 Real-world Experiments

Datasets We conduct our real-world experiments on three sequential user-behavior datasets. In all three, a sequence is defined as the records generated by a single individual. The **MovieLens 25M** dataset (Harper and Konstan, 2015) contains records of user-generated movie reviews alongside a rating. Marks represent the categories under which a reviewed movie can be classified as. The **MOOC** dataset (Kumar et al., 2019) is a collection of online user-behaviors for students taking an online course. Marks represent the type interaction a student has performed. Lastly, the **Taobao** user behavior dataset (Zhu et al., 2018) contains page-viewing records from users on an e-commerce platform. Marks are defined as the category of the item being viewed, with categories outside of the top 1,000 most frequent being discarded. All datasets were split into 75% training, 10% validation, and 15% test splits for model fitting and experiments. Summary statistics for these datasets can be found in Table 1. All preprocessing details for these three datasets can be found in the Appendix.

Models All real-world experiments use neural Hawkes models (Mei and Eisner, 2017), one trained for each dataset. Each model was trained to convergence on the training split with stability/generality ensured via the validation split. All training and model details can be found in the Appendix.

Hitting Time Queries: For each dataset, we randomly sample 1,000 different sequences $\mathcal{S}(T)$. For each sequence, we condition on the first five events, $\mathcal{S}[0, \tau_5]$, and evaluate a hitting time query for the remaining future.[5] The specific hitting time query asked is $p(\text{hit}(k) \leq t \mid \mathcal{S}[0, \tau_5])$ where $k := \kappa_6$ and $t := 10 \times \tau_6$ for $(\tau_6, \kappa_6) \in \mathcal{S}(T)$.

[5]All experiments evaluate necessary integrals with the trapezoidal rule. For more details, see Appendix.

We compared estimating this query with naive sampling and importance sampling using varying amounts of samples: $\{2, 4, 10, 25, 50, 250, 1000\}$. Mean RAE compared to ground truth (estimated using importance sampling with 5000 samples) can be seen in Fig. 3a. We witness roughly an order of magnitude of improvement in performance for the same amount of samples. Primarily, we attribute this improvement to the fact that naive sampling only collects binary values, whereas our proposed procedure collects much more dense information over the entire span $[\tau_5, t]$.

We also analyze the relative efficiency of our estimator compared to naive sampling. For each query asked, the efficiency was estimated using 5000 importance samples. The results can be seen in Fig. 3b. We achieve a dramatic decrease in variance by several orders of magnitude, in the majority of contexts, across all datasets. Interestingly, it appears that the efficiency is correlated with the underlying ground truth value π. We believe this may be due to the form of the importance sampling estimator: $1 - \exp\left(-\int_0^t \lambda_k^*(s)ds\right)$. Since the intensity function is non-negative, it is simple for the model to produce estimates close to 0; however, to producing values close to 1 requires the integral to tend towards infinity.

"A before B" Queries: Similar to the hitting time experiments, for "A before B" queries we similarly sample 1000 random test sequences and condition on the first five events $\mathcal{S}[0, \tau_5]$. Then, we estimate the query $p(\text{hit}(A) < \text{hit}(B) \mid \mathcal{S}[0, \tau_5])$ where A and B are randomly chosen to contain one third of the mark space $\mathbb{M}$.

We compared estimating this query with naive sampling and importance sampling using varying amounts of samples: $\{2, 4, 10, 25, 50, 250\}$. We utilized the truncated importance sampled estimator, Eq. (5), where T is chosen dynamically for each sequence such that a maximum difference of 0.01 is allowed between the upper and lower bounds. Mean RAE compared to ground truth (estimated using naive sampling with 5,000 samples) can be seen in Fig. 4a.[6] Like the hitting time results, we can see roughly an order of magnitude improvement in performance. Some results indicate that the limiting factor is the precision threshold for choosing T (e.g., see MovieLens results). We also see a similar variance reduction relative to previous experiments, shown in Fig. 4b. Here, the runtime cost is much greater as we have to accumulate an integral over an indefinite amount of time; however, we can see that on average it is still very much "worth it" to utilize this framework over naive sampling as evidenced by all of the blue dots above the red line.

Marginal Mark Distribution Queries: We additionally performed n^{th} marginal mark distribution queries in much

[6]Importance sampling would have been used for ground truth here; however, it is more sound to use an unbiased estimator for ground truth.

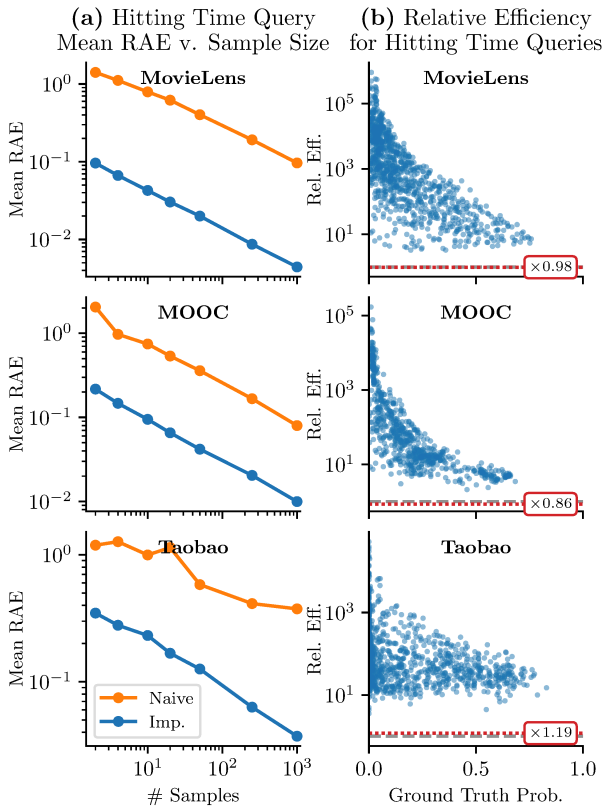

Figure 3: Results from 1000 different hitting time queries evaluated on models trained on three different datasets. (a) Average relative absolute error for naive and importance sampling shown in comparison to number of sampled sequences used. (b) Estimated relative efficiency values for importance sampling compared to naive sampling plotted against ground truth hitting time query values. Gray dashed lines indicate an efficiency of 1. Red lines with associated text box indicate the average multiplicative increase in computation time for importance sampling.

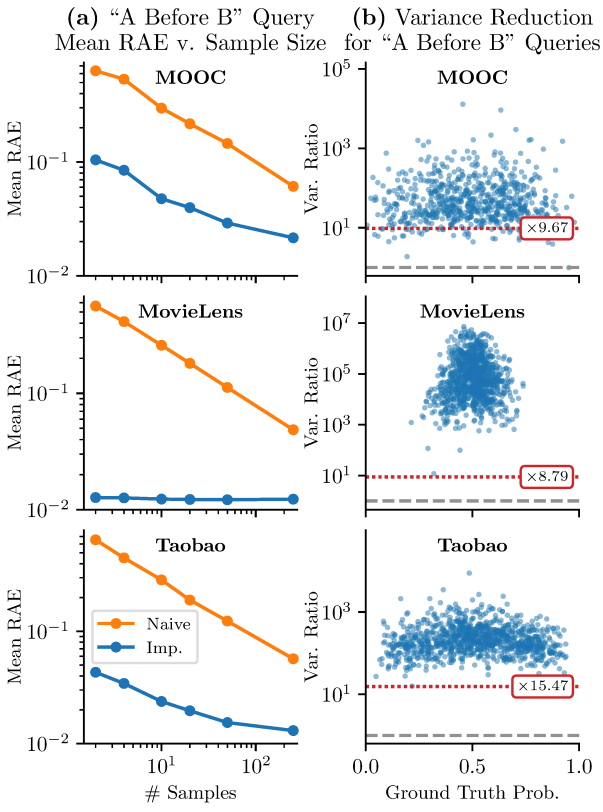

Figure 4: Same setup as seen in Fig. 3 with the same models and datasets only applied to "A before B" queries. Results for (b) are presented as "variance reduction" instead of "relative efficiency" since our derived estimator for importance sampling is biased due to truncating the integral in Eq. (5).

the same vein as the hitting time queries. Due to space limitations, the majority of the details and results can be found in the Appendix. That being said, we found that the resulting relative efficiencies for these queries to be much less than those of the other queries, but still more efficient than naive as Theorem 1 suggests. Across the datasets, the median relative efficiency ranged from 1.9 to 2.7. We speculate this to be due to the fact that the bounds of integration in the estimator are tied to sampled event times rather than being static values, inducing quite a lot of potential variance in the estimator.

5.2 Synthetic Experiments

For artificial experiments, we wanted to investigate the trends of our estimation procedures for a variety of queries over *many different* models—something that is difficult to do with real-world data as we typically only have access to one model trained on a given dataset. As such, we primarily focus on randomly instantiated parametric Hawkes processes with both exponential kernels and Gamma decay kernels (see Appendix for details). Under this setting, we were able to recreate similar findings in terms of estimation error and efficiency for the types of queries evaluated with real-world data. Due to space limitations these results can be seen in the Appendix.

In addition to these expected results, we also sought to investigate how different aspects of the underlying model affect the estimation procedure. In particular, we measured the average wall-clock time taken to generate samples for naive estimation and importance sampling as a function of how much cross-mark interaction is present. We modulate the *interaction strength* in these generated models by changing the scale of the randomly generated mark-to-mark intensity parameters. The results can be seen in Fig. 5 where we evaluated both estimation procedures on random hitting time queries for 1,000 different generated models with each across a span of interaction strengths. As more cross-mark

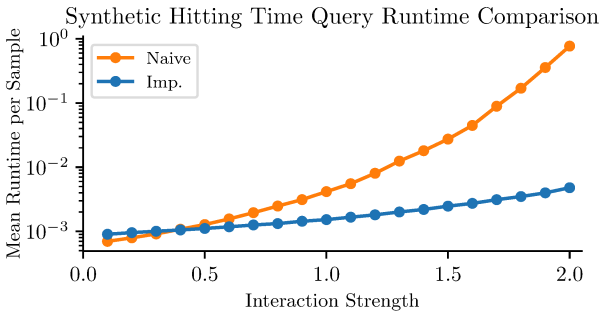

Figure 5: Average wall-clock time taken to generate a single sample under naive and importance sampling for hitting time queries with 1,000 randomly instantiated parametric exponential-kernel Hawkes models with different scalings of "interaction strength" (i.e., amount of modeled cross-mark interaction).

excitement is encouraged by a model, the runtime it takes to sample a sequence over the same observation window becomes much longer in general; however, importance sampling counters this trend due to zeroing out (potentially several) marked intensities in q, thus barring events from happening. From these results, we can see that our importance sampling procedure is more robust to the underlying dynamics of a model over sampling windows fixed in time.

6 Conclusion

In this work, we proposed a general restricted-mark framework that enables us to efficiently answer a range of intractable probabilistic queries for continuous-time sequential event data. Experimental results show that we gain a significant improvement in sampling efficiency, by several orders of magnitude, from the use of importance sampling compared to naive estimates. These improvements were consistent across three real-world datasets in different application domains with varying sequence lengths and numbers of marks. The results can in principle be further improved by producing more computationally efficient sampling procedures to use in conjunction with our proposed importance sampling estimators.

Acknowledgements

We thank Sam Showalter and our reviewers for their invaluable feedback in the writing of this paper. This work was supported by National Science Foundation Graduate Research Fellowship grant DGE-1839285, by an NSF CAREER Award, by the National Science Foundation under award numbers 1900644, 2003237, and 2007719, by the National Institute of Health under awards R01-AG065330-02S1 and R01-LM013344, by the Department of Energy under grant DE-SC0022331, by the HPI Research Center in Machine Learning and Data Science at UC Irvine, and by Qualcomm Faculty awards.

References

Andersen, P. K., Borgan, O., Gill, R. D., and Keiding, N. (2012). *Statistical Models based on Counting Processes*. Springer Science & Business Media.

Bacry, E., Dayri, K., and Muzy, J.-F. (2012). Non-parametric kernel estimation for symmetric Hawkes processes. Application to high frequency financial data. *The European Physical Journal B*, 85(5):1–12.

Biloš, M., Charpentier, B., and Günnemann, S. (2019). Uncertainty on asynchronous time event prediction. *Advances in Neural Information Processing Systems*, 32.

Boyd, A., Bamler, R., Mandt, S., and Smyth, P. (2020). User-dependent neural sequence models for continuous-time event data. In *Advances in Neural Information Processing Systems*, volume 33, pages 21488–21499. Curran Associates, Inc.

Boyd, A., Showalter, S., Mandt, S., and Smyth, P. (2022). Predictive querying for autoregressive neural sequence models. In *Advances in Neural Information Processing Systems*, volume 35.

Chiang, W.-H., Liu, X., and Mohler, G. (2022). Hawkes process modeling of COVID-19 with mobility leading indicators and spatial covariates. *International Journal of Forecasting*, 38(2):505–520.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220.

Daley, D. J. and Vere-Jones, D. (2003). *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*, page 274. Springer.

Deshpande, P., Marathe, K., De, A., and Sarawagi, S. (2021). Long horizon forecasting with temporal point processes. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 571–579.

Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., and Song, L. (2016). Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1555–1564.

Enguehard, J., Busbridge, D., Bozson, A., Woodcock, C., and Hammerla, N. (2020). Neural temporal point processes for modelling electronic health records. In *Machine Learning for Health*, pages 85–113. PMLR.

Fan, Y., Xu, J., and Shelton, C. R. (2010). Importance sampling for continuous time Bayesian networks. *Journal of Machine Learning Research*, 11(72):2115–2140.

Gupta, V., Bedathur, S., Bhattacharya, S., and De, A. (2021). Learning temporal point processes with intermittent observations. In *International Conference on Artificial Intelligence and Statistics*, pages 3790–3798. PMLR.

Harper, F. M. and Konstan, J. A. (2015). The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TIIS)*, 5(4):1–19.

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

Hawkes, A. G. (2018). Hawkes processes and their applications to finance: a review. *Quantitative Finance*, 18(2):193–198.

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

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

Kumar, S., Zhang, X., and Leskovec, J. (2019). Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1269–1278.

Liniger, T. J. (2009). *Multivariate Hawkes processes*. PhD thesis, ETH Zurich.

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

Mei, H., Qin, G., and Eisner, J. (2019). Imputing missing events in continuous-time event streams. In *International Conference on Machine Learning*, pages 4475–4485. PMLR.

Mishra, S., Rizoiu, M.-A., and Xie, L. (2016). Feature driven and point process approaches for popularity prediction. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1069–1078.

Nagpal, C., Jeanselme, V., and Dubrawski, A. (2021). Deep parametric time-to-event regression with time-varying covariates. In *Survival Prediction-Algorithms, Challenges and Applications*, pages 184–193. PMLR.

Nodelman, U., Shelton, C., and Koller, D. (2002). Continuous time Bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 378–387.

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

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

Ogata, Y. (1998). Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, 50(2):379–402.

Rizoiu, M.-A., Xie, L., Sanner, S., Cebrian, M., Yu, H., and Van Hentenryck, P. (2017). Expecting to be hip: Hawkes intensity processes for social media popularity. In *Proceedings of the 26th International Conference on World Wide Web*, pages 735–744.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*, volume 2. Springer.

Shchur, O., Biloš, M., and Günnemann, S. (2020). Intensity-free learning of temporal point processes. *International Conference on Learning Representations (ICLR)*.

Shelton, C. R. and Ciardo, G. (2014). Tutorial on structured continuous-time Markov processes. *Journal of Artificial Intelligence Research*, 51:725–778.

Zhu, H., Li, X., Zhang, P., Li, G., He, J., Li, H., and Gai, K. (2018). Learning tree-based deep model for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1079–1088.

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

A Efficiency Proof Lemma

Lemma 2. *If a bounded random variable $X \in [0,1]$, with mean π and CDF F, has Var$[X] = \pi(1-\pi)$, then $X \sim$ Bern(π).*

Proof. Let X be a random variable with support $[0,1]$, mean π, and variance $\pi(1-\pi)$. It then follows that:

$$\begin{aligned}
\text{Var}[X] &= \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2 \\
\implies \pi(1-\pi) &= \mathbb{E}\left[X^2\right] - \pi^2 \\
\implies \pi &= \mathbb{E}\left[X^2\right] \\
&= \int_{[0,1]} x^2 dF(x) \\
&= \int_{\{0,1\}} x^2 dF(x) + \int_{(0,1)} x^2 dF(x) \\
&= p(X=1) + \int_{(0,1)} x^2 dF(x)
\end{aligned}$$

$\int_{(0,1)} x^2 dF(x) > 0$ if and only if $p(X \in (0,1)) > 0$. If we assume that $p(X \in (0,1)) > 0$, then it follows that:

$$\begin{aligned}
\pi &= p(X=1) + \int_{(0,1)} x^2 dF(x) \\
&< p(X=1) + \int_{(0,1)} x dF(x) \\
&= p(X=1) + (\pi - p(X=1)) \\
&= \pi,
\end{aligned}$$

however, $\pi \not< \pi$. Hence, by contradiction $p(X \in (0,1)) = 0$ which implies that $p(X=1) = \pi$ and $p(X=0) = 1-\pi$ since $\mathbb{E}[X] = \pi$. Thus, it can be concluded that $X \sim$ Bern(π). □

B Deriving "A before B" Estimator

Let $A, B \subset \mathbb{M}$ and $A \cap B = \varnothing$. Recall that $\mathcal{S}_A[0,t]$ is the sequence of events over times $[0,t]$ with the restriction that the marks must all belong to A. Finally, let q describe a proposal distribution with $\mu_k^*(t) = \mathbb{1}(k \notin A \cup B)\lambda_k^*(t)$. With this in mind, we derive the expected value expression for the "A before B" queries:

$$\begin{aligned}
&p(\text{hit}(A) < \text{hit}(B)) = \int_0^\infty p(\text{hit}(A) < \text{hit}(B), \text{hit}(A) = t)\, dt \\
&= \int_0^\infty \sum_{k \in A} p(\mathcal{S}[t,t] = \{(t,k)\}, \mathcal{S}_A(t) = \varnothing, \mathcal{S}_B(t) = \varnothing)\, dt \\
&= \int_0^\infty \sum_{k \in A} p(\mathcal{S}[t,t] = \{(t,k)\}, \mathcal{S}_{A \cup B}(t) = \varnothing)\, dt \\
&= \int_0^\infty \sum_{k \in A} \mathbb{E}_p\left[p(\mathcal{S}[t,t] = \{(t,k)\}, \mathcal{S}_{A \cup B}(t) = \varnothing \,|\, \mathcal{S}(t))\right] dt
\end{aligned}$$

$$
\begin{aligned}
&= \int_0^\infty \sum_{k \in A} \mathbb{E}_{\mathcal{S}(t) \sim p} \left[p\left(\mathcal{S}[t,t] = \{(t,k)\} \,|\, \mathcal{S}_{A \cup B}(t) = \varnothing, \mathcal{S}(t)\right) p\left(\mathcal{S}_{A \cup B}(t) = \varnothing \,|\, \mathcal{S}(t)\right) \right] dt \\
&= \int_0^\infty \sum_{k \in A} \mathbb{E}_{\mathcal{S}(t) \sim p} \left[p\left(\mathcal{S}[t,t] = \{(t,k)\} \,|\, \mathcal{S}(t)\right) \mathbb{1}\left(\mathcal{S}_{A \cup B}(t) = \varnothing\right) \right] dt \\
&= \int_0^\infty \sum_{k \in A} \mathbb{E}_{\mathcal{S}(t) \sim p} \left[p\left(\mathcal{S}[t,t] = \{(t,k)\} \,|\, \mathcal{S}(t)\right) \mathbb{1}\left(\mathcal{S}_{A \cup B}(t) = \varnothing\right) \right] dt \\
&= \int_0^\infty \sum_{k \in A} \mathbb{E}_{\mathcal{S}(t) \sim p} \left[\lambda_k^*(t) \mathbb{1}\left(\mathcal{S}_{A \cup B}(t) = \varnothing\right) \right] dt \\
&= \int_0^\infty \mathbb{E}_{\mathcal{S}(t) \sim p} \left[\lambda_A^*(t) \mathbb{1}\left(\mathcal{S}_{A \cup B}(t) = \varnothing\right) \right] dt \\
&= \int_0^\infty \mathbb{E}_{\mathcal{S}(t) \sim q} \left[\lambda_A^*(t) \mathbb{1}\left(\mathcal{S}_{A \cup B}(t) = \varnothing\right) \frac{p\left(\mathcal{S}(t)\right)}{q\left(\mathcal{S}(t)\right)} \right] dt \\
&= \int_0^\infty \mathbb{E}_{\mathcal{S}(t) \sim q} \left[\lambda_A^*(t) \exp\left(- \int_0^t \lambda_{A \cup B}^*(s) ds \right) \right] dt \\
&= \int_0^\infty \mathbb{E}_{\mathcal{S}(\infty) \sim q} \left[\lambda_A^*(t) \exp\left(- \int_0^t \lambda_{A \cup B}^*(s) ds \right) \right] dt \\
&= \mathbb{E}_{\mathcal{S}(\infty) \sim q} \left[\int_0^\infty \lambda_A^*(t) \exp\left(- \int_0^t \lambda_{A \cup B}^*(s) ds \right) \right] dt
\end{aligned}
$$

where the last line is justified due to the Dominated Convergence Theorem. The prerequisites for this theorem are satisfied by noting that:

$$
\begin{aligned}
\int_0^\infty \lambda_A^*(t) \exp\left(- \int_0^t \lambda_{A \cup B}^*(s) ds \right) dt &\leq \int_0^\infty \lambda_{A \cup B}^*(t) \exp\left(- \int_0^t \lambda_{A \cup B}^*(s) ds \right) dt \\
&= - \int_0^\infty \frac{d}{dt} \exp\left(- \int_0^t \lambda_{A \cup B}^*(s) ds \right) dt \\
&= \exp\left(- \int_0^0 \lambda_{A \cup B}^*(s) ds \right) - \exp\left(- \int_0^\infty \lambda_{A \cup B}^*(s) ds \right) \\
&= 1 - \exp\left(- \int_0^\infty \lambda_{A \cup B}^*(s) ds \right) \\
&\leq 1.
\end{aligned}
$$

C Further Experimental Details and Results

C.1 Dataset Preprocessing

We evaluate our methods for probabilistic querying on three real-world user-behavior datasets in different application domains that are publicly available. All datasets do not include personally identifiable information, where users are identified by unique integer IDs. For all our experiments, sequences are defined as the event histories of each user, where events have timestamps in seconds. We changed the time resolution from seconds to hours for better interpretability of our query implications. Additionally, we only consider sequences with at least 5 events and at most 200 events. We use 75% of the sequences for training, 10% for validation, and 15% for testing.

MovieLens The MovieLens 25M dataset (Harper and Konstan, 2015) contains 25 million movie ratings by 162,000 users. The movie category (genre) associated with each rating is modeled as marks, and the exact rating value is ignored.[7] For

[7] A single movie in this dataset can possibly have multiple categories associated with it. To accommodate this, if a movie has multiple categories we randomly select a subset of two categories to represent the movie. Note this highlights the benefits of formulating queries as sets of marks instead of just singular marks. To evaluate the hitting time of the next "comedy" movie reviewed, then we would need to evaluate the hitting time of the set of all pairs of categories where one element is the comedy genre. This is essentially describing marginalizing over a hierarchical structure for the marks.

Table 2: Model Hyperparameters for Real-World Datasets

Hyperparameter	MovieLens	MOOC	Taobao
# Training Epochs	100	100	300
Mark Embedding Size	32	32	64
Recurrent Hidden State Size	64	64	128

each sequence, the start and the end time are defined as the first and the last event time of each user respectively, because the time span for different users ranges from seconds to years. The first event is discarded in the sequence of history and is only used to indicate $t = 0$. For consistent dynamics across the dataset, we filter the data to only contain reviews at or after the year 2015. This leaves 34,935 remaining sequences, each from a unique user.

MOOC The MOOC user action dataset (Kumar et al., 2019) represents user activities on a massive open online course (MOOC) platform. It consists of 411,749 course activities in 97 different types modeled as marks for 7,047 users, out of which 4,066 users dropped out after an activity. Timestamps are standardized to start from timestamp 0. We use the last event time for drop-out users as the end of their sequences, and the maximum timestamp for the other users.

Taobao The Taobao user behavior dataset (Zhu et al., 2018) was originally intended for recommendations for online shopping, which includes four behaviors: page viewing, purchasing, adding items to the chart, and to wishlist. We focus on page viewing of users as events, and model the item category as the event mark, which has marketing implications such as click through rate of recommending some types of items. Due to the large scale of the dataset, we use a subset of 2,000,000 events on 8 consecutive calendar days inclusive (November 25th, 2017 - December 2nd, 2017), as well as the most frequent 1,000 marks (item categories) to demonstrate query answering. All user sequences have the same length.

C.2 Modeling Details

For each of the real-world datasets, a neural Hawkes process model (Mei and Eisner, 2017) was trained with a batch size of 128, a learning rate of 0.001, a linear warm-up learning rate schedule over the first 1% of training iterations, a max allowed gradient norm of 10^4 for training stability, and the Adam stochastic gradient optimization algorithm (Kingma and Ba, 2015) with default hyperparameters. Specific datasets had specific model hyperparameters due to differences in the amount of data and total possible marks. The details for these can be found in Table 2. All models were trained for a fixed amount of epochs; however, each one was confirmed to have converged based on average held-out validation log-likelihood.

C.3 Integration Approximation

For the real-world experiments, many integrals need to be evaluated in order to produce estimates for various queries. Since we use essentially black-box MTPP models, we do not have access to an analytical form for integration. As such, we must estimate every integral at play.

To do this, we utilize the trapezoidal rule. For reference, this involves estimating integrals with the following summation:

$$\int_a^b f(x)dx \approx \sum_{i=1}^{N} \left(f(x_i) + f(x_{i-1})\right) \frac{x_i - x_{i-1}}{2}$$

where the points $x_{i-1} < x_i$ span the interval $[a, b]$ with $x_0 = a$ and $x_N = b$. For hitting time queries and marginal mark queries, we utilize $N = 1000$ integration points with equal spacing. It is likely that we could get by with much less for these queries, however, for the sake of high precision for experimental results we utilized a large amount of sample points.

For the "A before B" queries, we found that the resolution at which the estimator is evaluated at is of much more importance than the other queries. As such, for this query we estimate integrals in an online fashion during the sampling procedure for each proposal distribution sample sequence in conjunction with a very high proposal dominating rate (see Ogata (1981) for details). This allowed for a much more efficient procedure (in both computation and memory consumption) compared to integrating results after sampling.

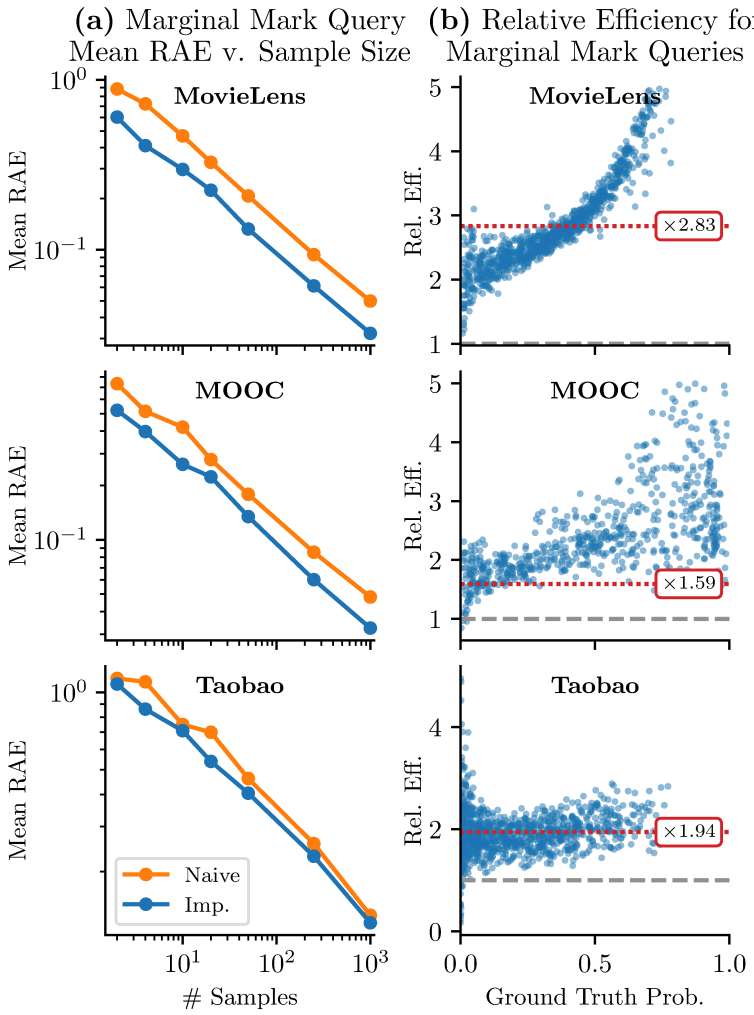

Figure 6: Results from 1,000 different marginal mark queries evaluated on models trained on three different datasets. (a) Average relative absolute error for naive and importance sampling shown in comparison to number of sampled sequences used. (b) Estimated relative efficiency values for importance sampling compared to naive sampling plotted against ground truth marginal mark query values. Gray dashed lines indicate an efficiency of 1. Red lines with associated text box indicate the average multiplicative increase in computation time for importance sampling.

C.4 Marginal Mark Query Experiments

Similar to the hitting time experiments, for the marginal mark queries we similarly sample 1000 random test sequences and condition on the first five events $\mathcal{S}[0, \tau_5]$. Then, we estimate the query $p(\kappa_8 \in A \,|\, \mathcal{S}[0, \tau_5])$ where A is a randomly selected subset of all of the unique marks that appear in the entire sequence $\mathcal{S}$. This is done to ensure that A contains relevant marks for the given sequence.

We compared estimating this query with naive sampling and importance sampling using varying amounts of samples: $\{2, 4, 10, 25, 50, 250, 1000\}$. Mean RAE compared to ground truth (estimated using importance sampling with 5,000 samples) can be seen in Fig. 6a. We witness roughly 1.5 to 3 times improvement in performance for the same amount of samples. Similar to hitting time query results, we attribute this improvement to the fact that naive sampling only collects binary values, whereas our proposed procedure collects much more dense information over the entire span from τ_5 to $\tau_8 \sim q$.

We also analyze the relative efficiency of our estimator compared to naive sampling. For each query asked, the efficiency was estimated using 5,000 importance samples. The results can be seen in Fig. 6b. We achieve a decent decrease in variance, in the majority of contexts, across all datasets. Like the hitting time query results, we also note a pretty strong correlation

between underlying ground truth values and the relative efficiency of this estimator.

Notably, these results do not appear to be as drastic as the hitting time query results. We believe this is due to the fact that the estimator's bounds of integration are sampled from the proposal distribution to be between τ_{N-1} and τ_N for each sequence (whereas the bounds for the hitting time query $p(\text{hit}(k) \leq t)$ is always the span of $[0, t]$). This added variability seems to dampen the impact of the integration in the first place.

C.5 Synthetic Data Experiments

We also perform experiments on hitting time queries and "A before B" queries using self-exciting parametric Hawkes processes (Hawkes, 1971). The intensity for Hawkes processes with exponential kernels has the explicit form:

$$\begin{aligned}\lambda_k^*(t) &= \mu_k + \sum_{\kappa=1}^{K} \int_0^t \phi_{\kappa k}(t-u) dN_\kappa(u) \\ &= \mu_k + \sum_{\kappa=1}^{K} \sum_{\tau_{\kappa,i}<t} \phi_{\kappa k}(t - \tau_{\kappa,i}),\end{aligned} \tag{6}$$

where $\tau_{\kappa,i}$ refers to the time when the i^{th} event of type κ occurs, $\phi(\boldsymbol{x}) = \boldsymbol{\alpha} e^{-\boldsymbol{\beta}\boldsymbol{x}}$ with $\boldsymbol{\alpha}, \boldsymbol{\beta} > \mathbf{0}$, and Equation 6 can be expressed in matrix form. The first term $\boldsymbol{\mu}$ is referred to as the *base intensity* or *background intensity* in literature. Each event instantaneously increases the intensity by the corresponding value of α and its influence decays exponentially with β and over time. Under this parametric form, the integrals for query estimates can be computed in closed forms.

We also conduct both experiments on hitting time and "A before B" queries using Hawkes processes with Gamma kernels. The Gamma kernel has the form of $\phi(\boldsymbol{x}) = \boldsymbol{x} e^{-\boldsymbol{x}}$, and the corresponding Hawkes processes do not have closed-form solutions to these queries.

We evaluate our methods on (i) hitting time queries $p(\text{hit}(k) \leq t)$ and (ii) "A before B" queries $p(\text{hit}(A) < \text{hit}(B))$. All results are averaged over 1,000 different randomly initiated parametric self-exciting Hawkes models that are not feasible for real-world datasets. These random models have different total amounts of marks ranging from $K = 3$ to $K = 10$, and have different inter-event effects as well as exponential rates of decay. We use 10 integration points for hitting time queries and 1,000 integration points for "A before B" queries. [8]

For each hitting time query, we fix $t = 1$ and $k = 0$, because the model is randomly generated. For the "A before B" queries, like the real-world experiments we let them be randomly sampled subsets of the vocabulary such $|A| = |B| \approx K/3$. We evaluate the hitting time queries using varying amounts of samples: $\{2, 4, 10, 25, 50, 250, 1000\}$. For "A before B" queries, we only use $\{2, 4, 10, 25, 50, 250\}$ number of samples because the query estimates take longer. Ground truth probabilities are calculated using 5,000 samples with importance sampling for hitting time queries and with naive method for "A before B" queries respectively.

The plots in Figs. 7 and 8 reveal similar patterns and illustrate that our method is more efficient than the naive estimates averaged over a range of different model settings.

[8] For the "A before B" queries, using 1,000 integration points after sampled provided sufficient precision and we did not need to employ the online integration approach used with the real-world experiments. This is most likely attributable to the well-behaved dynamics exhibited by the parametric Hawkes intensity. This is also why we used a reduced amount of integration points for the synthetic hitting time queries as well compared to the real-world experiments.

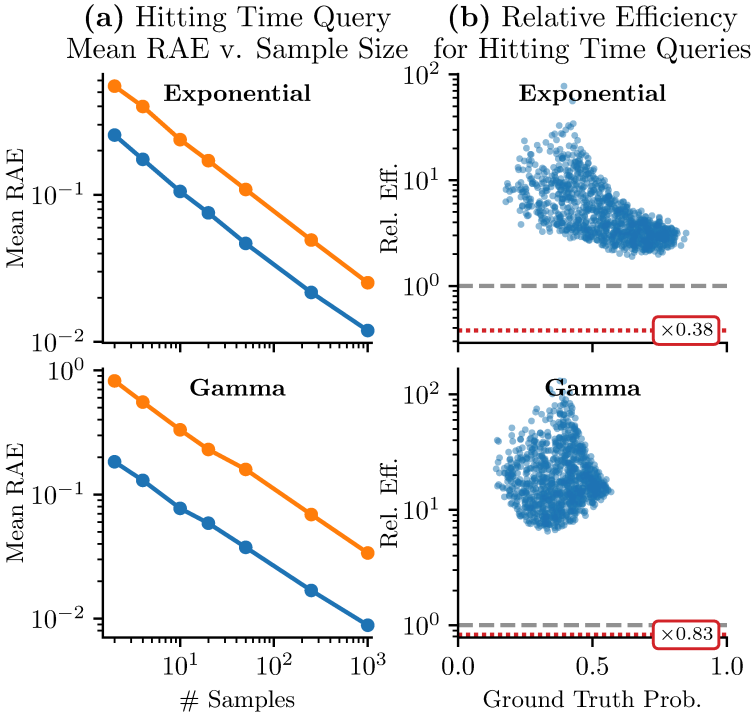

Figure 7: Synthetic experiments for hitting time queries evaluated on parametric self-exciting Hawkes processes with both exponential and Gamma kernels. (a) Average relative absolute error for naive and importance sampling shown in comparison to number of sampled sequences used. (b) Estimated relative efficiency values for importance sampling compared to naive sampling plotted against ground truth hitting time query values. Gray dashed lines indicate an efficiency of 1. Red lines with associated text box indicate the average multiplicative increase in computation time for importance sampling.

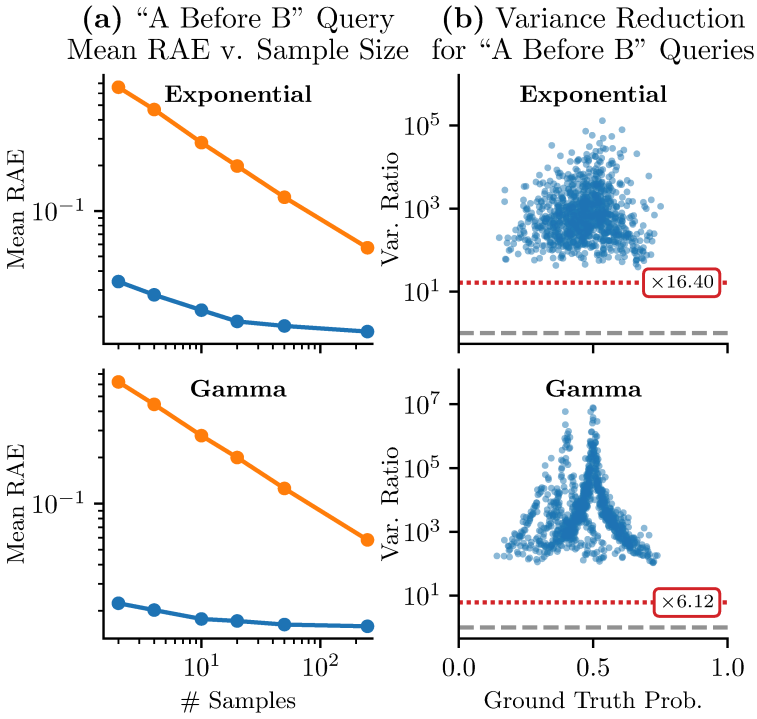

Figure 8: Synthetic experimental results evaluated on 1,000 different random models and "A before B" queries for parametric self-exciting Hawkes processes with both exponential and Gamma kernels. (a) Average relative absolute error for naive and importance sampling shown in comparison to number of sampled sequences used. (b) Estimated relative efficiency values for importance sampling compared to naive sampling plotted against ground truth "A before B" query values. Gray dashed lines indicate an efficiency of 1. Red lines with associated text box indicate the average multiplicative increase in computation time for importance sampling.