Interacting Diffusion Processes for Event Sequence Forecasting

Mai Zeng [*] [1 2 3] **Florence Regol** [*] [1 2 3] **Mark Coates** [1 2 3]

Abstract

Neural Temporal Point Processes (TPPs) have emerged as the primary framework for predicting sequences of events that occur at irregular time intervals, but their sequential nature can hamper performance for long-horizon forecasts. To address this, we introduce a novel approach that incorporates a diffusion generative model. The model facilitates sequence-to-sequence prediction, allowing multi-step predictions based on historical event sequences. In contrast to previous approaches, our model directly learns the joint probability distribution of types and inter-arrival times for multiple events. The model is composed of two diffusion processes, one for the time intervals and one for the event types. These processes interact through their respective denoising functions, which can take as input intermediate representations from both processes, allowing the model to learn complex interactions. We demonstrate that our proposal outperforms state-of-the-art baselines for long-horizon forecasting of TPPs.

1. Introduction

Predicting sequences of events has many practical applications, including forecasting purchase times and modeling transaction patterns or social media activity. The problem requires a dedicated model because it involves the complex task of jointly modeling two challenging data types: strictly positive continuous data for inter-arrival times and categorical data representing event types.

Early works employed intensity-based models such as the Hawkes process (Liniger, 2009). This modelling choice has advantages, including interpretability — it specifies the dynamics between events in the sequence explicitly. Subsequent efforts targeted integrating deep learning methods within the intensity framework (Mei & Eisner, 2017; Zuo et al., 2020; Yang et al., 2022).

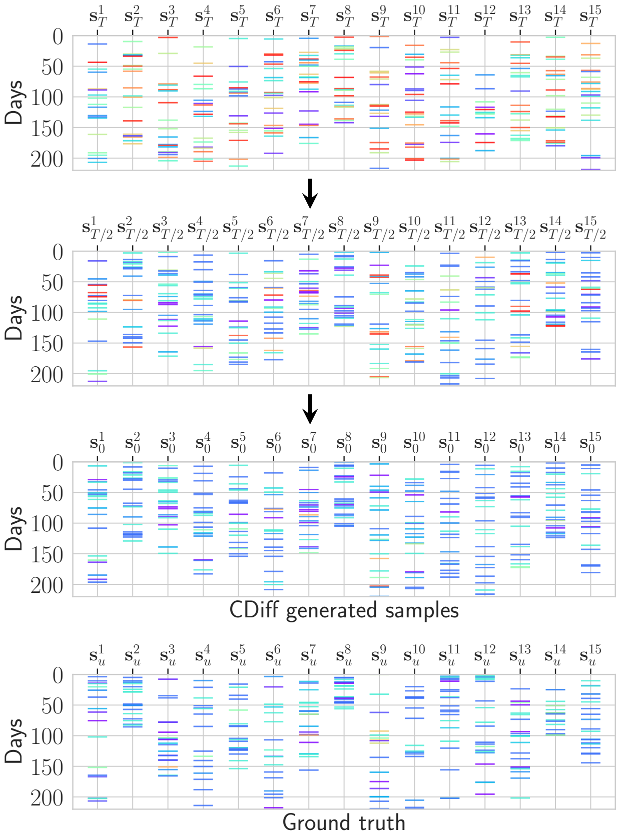


Figure 1. Visualization of the cross-diffusion generating process for 15 example Stackoverflow sequences. The colors indicates the different categories. We start by generating noisy sequences ($t = T$). Once we reach the end of the denoising process ($t = 0$), we recover sequences similar to ground truth sequences.

Although they can fit complex distributions, intensity-based formulations have drawbacks (Shchur et al., 2020a). As generative modeling research has developed, TPP models have moved away from the intensity parameterization, with more flexible specifications allowing them to use the full potential of recent generative models (Shchur et al., 2020a;

[*] Equal contribution [1] Department of Electrical and Computer Engineering, McGill University, Montreal QC, Canada [2] International Laboratory on Learning Systems (ILLS), Montreal, QC, Canada [3] Mila Québec AI Institute, Montreal, QC, Canada. Correspondence to: Mai Zeng <mai.zeng@mail.mcgill.ca>, Florence Regol <florence.robert-regol@mail.mcgill.ca>, Mark Coates <mark.coates@mcgill.ca>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Gupta et al., 2021; Lin et al., 2022).

Until recently, research has focused on next event forecasting. In (Xue et al., 2022; Deshpande et al., 2021), attention has turned to longer horizons, with the goal being forecasting multiple events. Recently proposed methods remain autoregressive, which can lead to a faster accumulation of error, as we illustrate in our experiments, but they are paired with additional modules that strive to mitigate this.

Our proposal goes a step further by directly generating a sequence of events. Consequently, our model can capture intricate interactions within the sequence of events between arrival times and event types. The crux of our proposed architecture is depicted in Fig. 1. We introduce coupled denoising diffusion processes to learn the probability distribution of the event sequences. One is a categorical diffusion process; the other is real-valued. The interaction of the neural networks that model the reverse processes allows us to learn dependencies between event type and interarrival time. Fig. 1 provides a visualization of the generation process[1].

Our approach significantly outperforms existing baselines for long-term forecasting, while also improving efficiency. Our experimental analysis provides insights into how the model achieves this: it can capture more complex correlation structures and is better at predicting distant events.

2. Problem Statement

Consider a sequence of events denoted by $\mathbf{s}^+ = \{(x_i^+, e_i)\}_{1 \leq i \leq T}$, where $x_i^+ \in (0, \infty)$ corresponds to the time interval between the events e_i and e_{i-1}, and the event e_i belongs to one of K categories: $e_i \in \mathcal{C}, |\mathcal{C}| = K$. The $+$-superscript is used to emphasize that the time-intervals are strictly positive. Given that we observe the start of a sequence (the context) $\mathbf{s}_c^+ = \{(x_i^+, e_i)\}_{1 \leq i \leq I}$ (or $\mathbf{x}_c^+ = [x_1^+, ..., x_I^+]$ and $\mathbf{e}_c = [e_1, ..., e_I]$ in vector form) with $I < T$, the goal is to forecast the remaining events. The dataset consists of a set of sequences: $\mathcal{D} = \{\mathbf{s}^{+,j}\}_{j=1}^M$ of potentially varying length.

Next N events forecasting In this setting, the task is to predict the following N events in the sequence $\mathbf{s}_u^+$: $\mathbf{x}_u^+ = [x_{I+1}^+, ..., x_{I+N}^+]$ and $\mathbf{e}_u = [e_{I+1}, ..., e_{I+N}]$. We also consider a slightly different setting: **interval forecasting**, where the task is to predict the events in a given time interval. We include the description of that setting with the metrics and methodology in Appendix A.1.

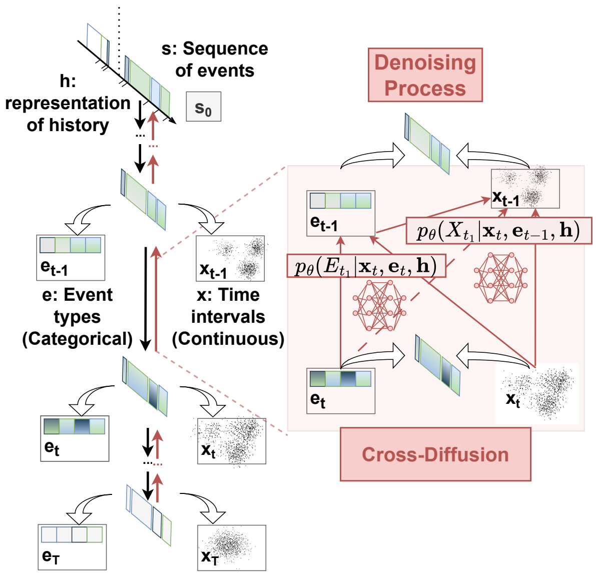

s: Sequence of events
h: representation of history
s_0
Denoising Process
e_{t-1}
x_{t-1}
$p_\theta(X_{t_1}|\mathbf{x}_t, \mathbf{e}_{t-1}, \mathbf{h})$
$p_\theta(E_{t_1}|\mathbf{x}_t, \mathbf{e}_t, \mathbf{h})$
e: Event types (Categorical)
x: Time intervals (Continuous)
e_t
x_t
Cross-Diffusion
e_T
x_T

Figure 2. Architectural overview of our model CDiff. We employ two interacting denoising diffusion processes, one categorical and one real-valued, to model the high-dimensional event sequences. The neural networks modeling the reverse diffusion steps interact, allowing them to learn dependencies between event types and interarrival times. Generating an entire sequence at once avoids the error propagation that can plague autoregressive models.

3. Related Work

We now briefly review and discuss relevant TPP modelling and forecasting literature. Bosser & Taieb (2023) and Shchur et al. (2021) provide more comprehensive reviews.

Hawkes-based methods. Early TPP forecasting approaches target single-event prediction (**Next** $N{=}1$ **event forecasting**) and adopt an intensity-based formulation (Rasmussen, 2011). The multivariate Hawkes Process (MHP) (Liniger, 2009) is the basis for many models (Du et al., 2016; Mei & Eisner, 2017; Zuo et al., 2020; Yang et al., 2022). Some approaches retain the intensity function but deviate from the MHP, incorporating graph learning (Zhang et al., 2021), non-parametric methods (Pan et al., 2021) or meta-learning (Bae et al., 2023). Other research addresses the efficiency (Shchur et al., 2020b; Nickel & Le, 2020) and expressiveness (Omi et al., 2019).

Non-Hawkes methods Striving to develop more effective models by departing from the intensity formulation, Shchur et al. (2020a) use a log-normal distribution paired with normalising flows. Lin et al. (2022) explore multiple conditional generative models for time forecasting including diffusion, variational inference, Generative Adversarial

[1] The code and implementation are available at our official repository

Networks (GANs), and normalizing flows. In all of these models, the types and interarrival times are modelled as conditionally independent given the history.

These works limit themselves to modelling a single upcoming event (N=1). As a result, they do not exploit the models' impressive ability to represent complex high dimensional data. In addition, modelling type and interarrival time independently is undesirable given that different event types can often be associated with very different arrival patterns.

Long horizon forecasting Xue et al. (2022) and Deshpande et al. (2021) consider long horizon forecasting. Xue et al. (2022) generate multiple candidate prediction sequences and introduce a selection module that aims to learn to select the best candidate. Deshpande et al. (2021) introduce a hierarchical architecture and a ranking objective to improve prediction of the number of events in each interval.

Although these works explicitly target long horizon forecasting, their generation mechanisms remain sequential. The techniques try to mitigate the error propagation in sequential models, but fundamentally they still only learn a model for $p(\{e_{h+1}, x^+_{h+1}\}|\{e_i, x^+_i\}_{i\leq h})$. As a result, the algorithms retain the core limitations of one-step ahead autoregressive forecasting. The approach of directly modeling a sequence of events has been explored in the non-marked setting (Lüdke et al., 2023). When there are no marks (which indicate different event types), the observed sequence of time intervals can be treated as (iid) samples from a conditional intensity distribution. Consequently, Lüdke et al. (2023) can directly parameterize the diffusion with a Poisson distribution in their Add-and-thin model. This differs from our approach – we model the interaction between event types and time intervals by learning the joint distribution. For completeness, we include a comparison in the Appendix with a modified version of Add-and-thin augmented with a naive event type predictor module.

4. Methodology

Model Overview Our proposal is to tackle the multi-event forecasting problem by directly modelling a complete sequence of N events. We therefore frame our problem as learning the conditional distribution $P(S^+_u = \mathbf{s}_u|\mathbf{s}_c)$, where $\mathbf{s}^+_u = (\mathbf{e}_u, \mathbf{x}^+_u)$ is the sequence of event types and interval to forecast and $\mathbf{s}^+_c = (\mathbf{e}_c, \mathbf{x}^+_c)$ is the historical (context) sequence. We introduce our Cross-Diffusion (CDiff) model, which comprises two interacting diffusion processes.

In a nutshell, we diffuse simultaneously both the time intervals and the event types of the target sequence: We first apply a Box-Cox transformation to the inter-arrival time values to transition from the strictly positive continuous domain ($X^+ \in (0, +\infty)$) to the more convenient unrestricted real space ($X \in (-\infty, +\infty)$). We use $S = (X, E)$ instead of $S^+ = (X^+, E)$ to indicate the event sequence with X in the unrestricted real space. We gradually add Gaussian noise to the transformed time intervals and uniform categorical noise to the types $S_0, S_1, \ldots, S_T$ until only noise remains in S_T. S_0 denotes the target sequence S_u. During training, we learn denoising distributions $p_\theta(S_{t-1}|S_t, \mathbf{s}_c)$ that can undo each of the noise-adding steps. Our denoising functions are split in two, but interact with each other, which is why we call our model "cross-diffusion." After training, we sample from $P(S_u|\mathbf{s}_c)$ by sampling noise S_T, then gradually reversing the chain by sampling from $p_\theta(S_{t-1}|S_t, \mathbf{s}_c)$ until we recover S_0. A high-level summary of our approach is illustrated in Fig. 2. The specifics of the model and its training are provided in the subsequent sections.

4.1. Model Details

A TPP model can be divided into two components (Lin et al., 2022): 1) the encoder of the variable length context $\mathbf{s}_c$; and 2) the generative model of the future events. We focus on the latter and adopt the transformer-based context encoder proposed by Xue et al. (2022) in order to generate a fixed-dimensional context representation denoted as $\boldsymbol{h} = f_\theta(\mathbf{s}^+_c)$.

Again, we first apply a Box-Cox transformation to the inter-arrival time values to transition from the strictly positive continuous domain ($X^+ \in (0, +\infty)$) to the more convenient unrestricted real space ($X \in (-\infty, +\infty)$). This allows us to model the variables with Gaussian distributions in the diffusion process. Appendix A.5 provides more detail.

Although the target distribution consists of a combination of categorical and continuous variables, we can define a single diffusion process for it. To achieve this, we begin by defining a forward/noisy process that introduces T new random variables, which are noisier versions of the sequence, represented by $S_0 = (X_0, E_0)$:

$$q(X_{1:T}, E_{1:T}|X_0, E_0) = \prod_{t=1}^{T} q(X_t, E_t|X_{t-1}, E_{t-1}). \quad (1)$$

For a diffusion model we strive to learn the *inverse denoising* process by learning the intermediate distributions $p_\theta(S_{t-1}|S_t, \mathbf{s}_c)$. The log likelihood of the target distribution $\log q(S_0|\mathbf{s}_c)$ is obtained by marginalizing over the denoising process. Following the diffusion model setup of (Ho et al., 2020), this marginalization can be approximated as:

$$\begin{aligned}\log q(S_0|\mathbf{s}_c) \geq \mathbb{E}_{q(S_0|\mathbf{s}_c)}\Big[&\log p_\theta(S_0|S_1, \mathbf{s}_c) \\ &- KL(q(S_T|S_0, \mathbf{s}_c)||q(S_T|\mathbf{s}_c)) \\ &- \sum_{t=2}^{T} KL\Big(q(S_{t-1}|S_t, S_0, \mathbf{s}_c)||p_\theta(S_{t-1}|S_t, \mathbf{s}_c)\Big)\Big]. \quad (2)\end{aligned}$$

Hence, we can summarize the generative diffusion model approach as follows: by minimizing the KL-divergences between the learned distributions $p_\theta(S_{t-1}|S_t, \mathbf{s}_c)$ and the noisy distributions $q(S_{t-1}|S_t, S_0, \mathbf{s}_c)$ at each t, we maximize the log likelihood of our target $\log q(S_0|\mathbf{s}_c)$.

Cross-diffusion for modeling event sequences As X_u and E_u are in different domains, we cannot apply a standard noise function to $q(X_t, E_t|X_{t-1}, E_{t-1})$. Instead, we factorize the noise-inducing distribution $q(S_t|S_{t-1}) = q(X_t|X_{t-1})q(E_t|E_{t-1})$. It is important to stress that this independence is only imposed on the forward (noise-adding) process. We do not assume independence in $q(S_0|\mathbf{s}_c)$ and our reverse diffusion process, described below, allows us to learn the dependencies. Denoting by $Cat(;p)$ a categorical distribution with parameter p, and given an increasing variance schedule $\{\beta_1, \ldots, \beta_T\}$, the forward process is:

$$q(S_t|S_{t-1}) = q(X_t|X_{t-1})q(E_t|E_{t-1}), \tag{3}$$

$$q(X_t|X_{t-1}) = \mathcal{N}(X_t; \sqrt{1-\beta_t}X_{t-1}, \beta_t\mathbf{I}), \tag{4}$$

$$q(E_t|E_{t-1}) = Cat(E_t; (1-\beta_t)E_{t-1} + \beta_t/K), \tag{5}$$

$$q(X_T) = \mathcal{N}(X_T; 0, \mathbf{I}), \tag{6}$$

$$q(E_T) = Cat(E_T; 1/K), \tag{7}$$

Next, we have to define the denoising process $p_\theta(S_{t-1}|S_t)$. We can express the joint distribution as:

$$p_\theta(S_{t-1}|S_t, \mathbf{s}_c) = p_\theta(X_{t-1}|S_t, E_{t-1}, \mathbf{s}_c)p_\theta(E_{t-1}|S_t, \mathbf{s}_c), \tag{8}$$

$$p_\theta(E_{t-1}|S_t, \mathbf{s}_c) = Cat(E_{t-1}|\pi_\theta(X_t, E_t, t, \mathbf{s}_c)), \tag{9}$$

$$p_\theta(X_{t-1}|S_t, E_{t-1}, \mathbf{s}_c) = \mathcal{N}(X_{t-1}; \mu_\theta(X_t, E_{t-1}, t, \mathbf{s}_c), \sigma_t). \tag{10}$$

Here we choose to fix $\sigma_t = \beta_t$ and μ_θ and π_θ are learnable. With the presented approach, during denoising, we first sample event types, and then conditioned on the sampled event types, we sample inter-arrival times. We can also choose to do the reverse. A sensitivity study in Appendix A.12 shows that this choice has a negligible effect on performance.

This can be viewed as two denoising processes that interact through the learnable functions μ_θ and π_θ. One models the inter-arrival times (Gaussian) and one models the event types (Categorical). The denoising processes are conditioned on the historical event sequence $\mathbf{s}_c$ and they are interacting with each other (through conditioning on $\mathbf{e}_{t-1}$ and $\mathbf{x}_t$). Therefore, we modify the standard parametrization of $\mu_\theta(\mathbf{x}_t, t)$ and $\pi_\theta(\mathbf{e}_t, t)$ from (Ho et al., 2020; Hoogeboom et al., 2021) to include these additional inputs. Rather than directly learning μ and π, we express them in terms of two other functions ϵ and ϕ to facilitate learning.

Introducing $\alpha_t \triangleq 1 - \beta_t$ and $\bar{\alpha}_t \triangleq \prod_{i \le t} \alpha_i$, the time denoising process is parameterized as:

$$\mu_\theta(\mathbf{x}_t, \mathbf{e}_{t-1}, t, \mathbf{s}_c) = \frac{\mathbf{x}_t}{\sqrt{\alpha_t}} - \frac{\beta_t \epsilon_\theta(\mathbf{x}_t, \mathbf{e}_{t-1}, t, \mathbf{s}_c)}{\sqrt{\alpha_t}\sqrt{1-\bar{\alpha}_t}}. \tag{11}$$

The denoising step of the event type is parameterized differently. Its learnable component is parameterized to directly predict, at step t, the targeted distribution of the data E_0, modeled as $\hat{\mathbf{e}}_0$, from the event type, e_t, the (transformed) time interval, x_t, the denoising step t, and the context s_c:

$$\hat{\mathbf{e}}_0 = \phi_\theta(\mathbf{e}_t, \mathbf{x}_t, t, \mathbf{s}_c).. \tag{12}$$

This prediction $\hat{\mathbf{e}}_0$ is then combined with the current $\mathbf{e}_t$ through a weighted sum, and subsequently normalized to obtain the parameters of the categorical distribution for E_{t-1} (parameterized as $\pi_\theta(\mathbf{x}_t, \mathbf{e}_t, t, \mathbf{s}_c)$ in Eqn 9):

$$\boldsymbol{\theta}(\mathbf{e}_t, \hat{\mathbf{e}}_0) = [\alpha_t \mathbf{e}_t + \frac{1-\alpha_t}{K}] \odot [\bar{\alpha}_{t-1}\hat{\mathbf{e}}_0 + \frac{1-\bar{\alpha}_{t-1}}{K}],$$
$$\tilde{\boldsymbol{\theta}} \triangleq \boldsymbol{\theta}(\mathbf{e}_t, \hat{\mathbf{e}}_0), \tag{13}$$

$$\pi_\theta(\mathbf{x}_t, \mathbf{e}_t, t, \mathbf{s}_c) = \tilde{\boldsymbol{\theta}} / \sum_{k=1}^{\mathrm{K}} \tilde{\boldsymbol{\theta}}_k. \tag{14}$$

Here $\odot$ denotes the Hadamard product. This concludes our description of the parameterization of the denoising process. The learnable components of CDiff are $\epsilon_\theta, \phi_\theta$ and f_θ. In our experiments, we use transformer-based networks that we describe in Section 5.4.

With a trained model $p_\theta(S^0|\mathbf{s}_c)$, given a context sequence $\mathbf{s}_c$, we can generate samples of the next N events, $\hat{\mathbf{s}}^0 \sim p_\theta(S^0|\mathbf{s}_c)$. To form the final predicted forecasting sequence $\hat{\mathbf{s}}_u$, we generate multiple samples, calculate the average time intervals, and set the event types to the majority types. With an abuse of notation, we denote this averaging of sequences as $\hat{\mathbf{s}}_u \triangleq \frac{1}{A}\sum_{a=1}^{A} \hat{\mathbf{s}}_a^0, \quad \hat{\mathbf{s}}_a^0 \sim p_\theta(S^0|\mathbf{s}_c)$.

4.2. Optimization

The log-likelihood objective is provided in Equation (2). We can separate the objective for the joint $q(S_0)$ into standard optimization terms of either continuous or categorical diffusion using Equation (8).

Starting with the first log term, we separate it as:

$$\mathbb{E}_{q(S_0|s_c)}\Big[\log p_\theta(S_0|S_1, \mathbf{s}_c)\Big]$$
$$\approx \sum_{j=1}^{M} \log p_\theta(\mathbf{x}_0^j|\mathbf{x}_1^j, \hat{\mathbf{e}}_0^j, \mathbf{s}_c^j) + \log p_\theta(\mathbf{e}_0^j|\mathbf{x}_1^j, \mathbf{e}_1^j, \mathbf{s}_c^j), \tag{15}$$

with $\hat{\mathbf{e}}_0^j \sim p_\theta(E_0|\mathbf{x}_1^j, \mathbf{e}_1^j, \mathbf{s}_c^j)$, $\mathbf{e}_1^j \sim q(E_1|\mathbf{e}_0^j, \mathbf{s}_c^j)$, and $\mathbf{x}_1^j \sim q(X_1|\mathbf{x}_0^j, \mathbf{s}_c^j)$. Next, we split the individual KL terms from (2) similarly:

$$\mathbb{E}_{q(S_0|s_c)}\Big[KL\Big(q(S_{t-1}|S_{t,0}, \mathbf{s}_c)||p_\theta(S_{t-1}|S_t, \mathbf{s}_c)\Big)\Big] =$$
$$\mathbb{E}_{q(S_0|\mathbf{s}_c)}\Big[KL\Big(q(X_{t-1}|X_{t,0}, \mathbf{s}_c)||p_\theta(X_{t-1}|S_t, E_{t-1}, \mathbf{s}_c)\Big)\Big]$$
$$+ \mathbb{E}_{q(S_0|\mathbf{s}_c)}\Big[KL\Big(q(E_{t-1}|E_{t,0}, \mathbf{s}_c)||p_\theta(E_{t-1}|S_t, \mathbf{s}_c)\Big)\Big]. \tag{16}$$

The target distribution of the event type can be expressed compactly by substituting the true $\mathbf{e}_0$ in Eqn (13):

$$\bar{\boldsymbol{\theta}}_{\text{post}}(\mathbf{e}_t, \mathbf{e}_0) \triangleq \boldsymbol{\theta}(\mathbf{e}_t, \mathbf{e}_0) / \sum_{k=1}^{\mathrm{K}} \boldsymbol{\theta}(\mathbf{e}_t, \mathbf{e}_0)_k, \tag{17}$$

$$q(\mathbf{e}_{t-1}|\mathbf{e}_t, \mathbf{e}_0) = Cat(\mathbf{e}_{t-1}|\bar{\boldsymbol{\theta}}_{\text{post}}(\mathbf{e}_t, \mathbf{e}_0)) \,. \tag{18}$$

We can therefore apply the typical optimization techniques of either continuous and categorical diffusion on each term:

$$\mathbb{E}_{q(S_0|s_c)}\Big[KL\Big(q(E_{t-1}|E_{t,0}, \mathbf{s}_c)||p_\theta(E_{t-1}|S_t, \mathbf{s}_c)\Big)\Big] \approx -\sum_{j=1}^{M}\sum_{k} \bar{\boldsymbol{\theta}}_{\text{post}}(\mathbf{e}_t^j, \mathbf{e}_0^j)_k \cdot \log \frac{\bar{\boldsymbol{\theta}}_{\text{post}}(\mathbf{e}_t^j, \mathbf{e}_0^j)_k}{\pi_\theta(\mathbf{x}_t^j, \mathbf{e}_t^j, t, \mathbf{s}_c^j)_k} \tag{19}$$

with $\mathbf{e}_t^j \sim q(E_t|\mathbf{e}_0^j, \mathbf{s}_c^j)$, $\mathbf{x}_t^j \sim q(X_t|\mathbf{x}_0^j, \mathbf{s}_c^j)$ for the event variables, and:

$$\mathbb{E}_{q(S_0|s_c)}\Big[KL\Big(q(X_{t-1}|X_{t,0}, s_c)||p_\theta(X_{t-1}|S_t, E_{t-1}, \mathbf{s}_c)\Big)\Big] \approx -\sum_{j=1}^{M} \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0^j + \sqrt{1-\bar{\alpha}_t}\epsilon, t, \hat{\mathbf{e}}_{t-1}^j, \mathbf{s}_c^j)\|^2 \tag{20}$$

with $\mathbf{e}_t^j \sim q(E_t|\mathbf{e}_0^j)$, $\mathbf{x}_t^j \sim q(X_t|\mathbf{x}_0^j)$, $\hat{\mathbf{e}}_{t-1}^j \sim p_\theta(E_{t-1}|\mathbf{x}_t^j, \mathbf{e}_t^j, \mathbf{s}_c^j)$ and $\epsilon \sim \mathcal{N}(0,1)$ for the continuous interarrival time variables.

Our final objective is hence given by:

$$\mathcal{L} = \sum_{j=1}^{M}\Bigg(\log p_\theta(\mathbf{x}_0^j|\mathbf{x}_1^j, \hat{\mathbf{e}}_0^j, \mathbf{s}_c^j) \log p_\theta(\mathbf{e}_0^j|\mathbf{x}_1^j, \mathbf{e}_1^j, \mathbf{s}_c^j) - \sum_{t=2}^{T}\Bigg(\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0^j + \sqrt{1-\bar{\alpha}_t}\epsilon, t, \hat{\mathbf{e}}_{t-1}^j, \mathbf{s}_c^j)\|^2 + \sum_{k=1}^{K} \bar{\boldsymbol{\theta}}_{\text{post}}(\mathbf{e}_t^j, \mathbf{e}_0^j)_k \cdot \log \frac{\bar{\boldsymbol{\theta}}_{\text{post}}(\mathbf{e}_t^j, \mathbf{e}_0^j)_k}{\pi_\theta(\mathbf{x}_t^j, \mathbf{e}_t^j, t, \mathbf{s}_c^j)_k}\Bigg)\Bigg). \tag{21}$$

Finally, we adhere to the common optimization approach used in diffusion models and optimize only one diffusion timestep term per sample instead of the entire sum. The timestep is selected by uniformly sampling $t \sim U(0, T)$. We employ the algorithm from (Song et al., 2021) to accelerate the sampling. Appendix A.11 provides further details.

5. Experiments

In our experiments, we set $N = 20$ (but include results for $N = 5, 10$). For each sequence in the dataset $\mathcal{D} = \{\mathbf{s}^j\}_{j=1}^M$, we set the last N events as $\mathbf{s}_u$ and set all earlier events as the context $\mathbf{s}_c$. Means and standard deviations are computed over 10 trials. We train for a maximum of 500 epochs and report the best trained model based on the validation set. Hyperparameter selection uses the Tree-Structured Parzen Estimator hyperparameter search algorithm from Bergstra et al. (2011). To avoid numerical error when applying the Box-Cox transformation to the x^+ values, we first add 1e-7 to all time values and then scale by 100. We transform back to x^+ after we estimate x using the inverse Box-Cox transformation, with the same parameter obtained from the train set, and downscale by 100. The detailed model description is in Appendix A.10, along with sensitivity studies for some of the hyperparameters.

5.1. Datasets

We use six real-world datasets. Taobao (Zhu et al., 2018) tracks user clicks made on a website; Taxi (Whong, 2014) contains trips to neighborhoods by taxi drivers; StackOverflow (Leskovec & Krevl, 2014) tracks the history of posts on stackoverflow; Retweet (Zhou et al., 2013) tracks user interactions on social media; MOOC (Kumar et al., 2019) tracks user interactions within an online course system; and Amazon (Ni et al., 2019) tracks the sequence of product categories reviewed by a group of users. We focus on datasets containing sequences with multiple events as our goal is multi-event prediction. Our synthetic dataset is generated from a Hawkes model. We follow Xue et al. (2022) for the train/val/test splits, which we report in Appendix A.4, together with additional dataset details.

5.2. Baselines

We compare our CDiff model with one naive and 6 state-of-the-art baselines for event sequence modeling. When available, we use reported hyperparameters, and otherwise we employ a tuning procedure (see Appendix A.6).

- **Homogeneous Poisson Process (naive)** is a constant intensity function. For the type prediction, we compute the marginal categorical distribution over the training set.
- **Neural Hawkes Process (NHP)** (Mei & Eisner, 2017) is a Hawkes-based model that uses a continuous LSTM.
- **Attentive Neural Hawkes Process (AttNHP)** (Yang et al., 2022) is a Hawkes-based model that integrates attention. It is the SOTA for single event forecasting.
- **Log-Normal Mixture Model (LNM)** (Shchur et al., 2020a) is an intensity-free temporal point process with the feature of fast sampling.
- **Temporal Conditional Diffusion Denoising Model (TCDDM)** (Lin et al., 2022) is a diffusion based generative model that relies on the assumption of conditional independence between inter-arrival time and event type.
- **Dual-TPP** (Deshpande et al., 2021): Dual-TPP targets

long horizon forecasting by jointly learning a distribution of the count of events in segmented time intervals.

- **HYPRO** (Xue et al., 2022) is the SOTA for multi-event/long horizon forecasting. It uses AttNHP as a base model, but includes a sequence selection module.

5.3. Evaluation Metrics

Assessing long-horizon performance is challenging as we must compare mixed-type vectors. There is no existing proper scoring rule. Therefore, we report multiple metrics.

Optimal Transport Distance (OTD): We use the OTD to compare event sequences, following Mei et al. (2019). $L(\hat{\mathbf{s}}_u, \mathbf{s}_u)$ is the minimum cost of editing a predicted event sequence $\hat{\mathbf{s}}_u$ into the ground truth $\mathbf{s}_u$. To accomplish this edit, we must identify the best *alignment* – a one-to-one partial matching $\mathbf{a}$ – of the events in the two sequences. We use the algorithm from (Mei et al., 2019) to find this alignment, and report the average **OTD** values when using various deletion/insertion cost constants $C = \{0.05, 0.5, 1, 1.5, 2, 3, 4\}$. Appendix A.3 presents more details about this metric.

RMSE$_e$ assesses how well the event type distribution in the predicted sequence matches ground truth. For each type k, we count the number of type-k events in $\mathbf{x}_u^+$, denoted C_k, as well as that in $\hat{\mathbf{x}}_u^+$, denoted $\hat{C}_k$. We report the root mean square error $\textbf{RMSE}_e = \sqrt{\frac{1}{M}\sum_{j=1}^{M}\frac{1}{K}\sum_{k=1}^{K}(C_k^j - \hat{C}_k^j)^2}$. We also report time-series forecasting metrics: $\textbf{RMSE}_{x^+}$, **MAPE**, and **sMAPE** (a normalized $MAPE$). Appendix A.3 provides metric details.

5.4. Implementation details

Since all methods are generative, we can generate several samples to form the final predictions $\hat{\mathbf{e}}_u$ and $\hat{\mathbf{x}}_u^+$. We generate 5 samples from $P(S_u^+|S_c^+ = \mathbf{s}_c^+)$ and average the time vectors to form $\hat{\mathbf{x}}_u^+$, and use majority voting over the event vectors to form $\hat{\mathbf{e}}_u$. For the history encoder f_θ, we adopt the architecture in AttNHP (Yang et al., 2022), which is a continuous-time Transformer module. For the two diffusion denoising functions $\epsilon_\theta(\cdot), \phi_\theta(\cdot)$, we use the PyTorch built-in transformer block (Paszke et al., 2019). We use the following positional encoding from (Zuo et al., 2020) for the sequence index i in the $f_\theta(\cdot)$ transformer:

$$[\mathbf{m}(y_j, D)]_i = \begin{cases} \cos(y_j/10000^{\frac{i-1}{D}}) & \text{if } i \text{ is odd}, \\ \sin(y_j/10000^{\frac{i}{D}}) & \text{if } i \text{ is even}. \end{cases} \quad (22)$$

Appendix A.13 provides more details about the positional encoding implementation and its use for the diffusion timestep t. For the diffusion process, we use a cosine β schedule, as proposed by Nichol & Dhariwal (2021). Appendix A.6 provides provides more detail concerning hyperparameters.

6. Results

Table 1 presents results of a subset of experiments for four selected metrics on real-world datasets. Complete results are in Appendix A.15. We test for significance using a paired Wilcoxon signed-rank test at the 5% significance level.

In alignment with previous findings, AttNHP consistently outperforms NHP, reaffirming its position as the SOTA single event forecasting method. HYPRO ranks as the second-best baseline since it leverages AttNHP as its base model and is designed for multi-event forecasting. Attention-based TCDDM and AttNHP show comparable results, while RNN models like NHP and others fall behind. The basic Homogeneous Poisson model ranks lowest. Our CDiff method consistently surpasses all baselines, often with a statistically significant margin, a trend that holds across various experiments, datasets, and metrics, as shown in Figure 3.

Figure 3(left) demonstrates CDiff's consistent top ranking. The middle and right panels show its outperformance for event type and time interval metrics. RNN-based models like LNM, NHP, and Dual-TPP fall short in long-term forecasting compared to attention-based models. TCDDM and LNM, while better at timing predictions, struggle with event type forecasting. This limitation is likely due to their assumption of conditional independence for event type prediction, which may impair their ability to capture complex relationships between event types and inter-arrival times.

6.1. CDiff can model complex inter-arrival times

We first examine the learned marginal distribution for time intervals. We use the Taobao dataset for our analysis because it is a relatively challenging dataset, with 17 event types and a marginal distribution of inter-arrival times that appears to be multi-modal. From the histograms of inter-arrival time prediction in Fig. 4, we see that CDiff is better at capturing the ground truth distribution. CDiff is effective at generating both longer intervals, falling within the range $(3h25, \infty]$, and shorter intervals, within the range $(0, 0.01h]$.

In contrast, HYPRO and AttNHP, the most competitive models, struggle to generate a sufficient number of values at the extremities of the marginal distribution. This also impacts the methods' ability to capture the joint relationship between time intervals and event types. To illustrate this, we consider two of the event categories for the Taobao dataset, and we plot the count histograms of the time intervals for categories 7 and 16 in Figure 5.

First, it is noticeable that HYPRO and AttNHP fail to generate an adequate number of events for these specific categories, resulting in counts lower than the ground truth. In contrast, CDiff generates the appropriate quantity. This implies that CDiff is better at capturing the marginal categorical distribution of events. For both event types, the

Table 1. **OTD**, **RMSE**$_e$, **RMSE**$_{x^+}$ and **sMAPE** of real-world datasets reported in mean ± s.d. Best are in bold, the next best is underlined. *indicates stat. significance w.r.t to the best method.

	Taxi				Taobao			
	OTD	**RMSE**$_e$	**RMSE**$_{x^+}$	**sMAPE**	**OTD**	**RMSE**$_e$	**RMSE**$_{x^+}$	**sMAPE**
HYPRO	21.653 ± 0.163	1.231 ± 0.015*	0.372 ± 0.004*	93.803 ± 0.454*	**44.336 ± 0.127**	2.710 ± 0.021*	0.594 ± 0.030*	134.922 ± 0.473*
Dual-TPP	24.483 ± 0.383*	1.353 ± 0.037*	0.402 ± 0.006*	95.211 ± 0.187*	47.324 ± 0.541*	3.237 ± 0.049*	0.871 ± 0.005*	141.687 ± 0.431*
Attnhp	24.762 ± 0.217*	1.276 ± 0.015*	0.430 ± 0.003*	97.388 ± 0.381*	45.555 ± 0.345*	2.737 ± 0.021	0.708 ± 0.010*	134.582 ± 0.920*
NHP	25.114 ± 0.268*	1.297 ± 0.019*	0.399 ± 0.040*	96.459 ± 0.521*	48.131 ± 0.297*	3.355 ± 0.030*	0.837 ± 0.009*	137.644 ± 0.764*
LNM	24.053 ± 0.609*	1.364 ± 0.032*	0.384 ± 0.005*	95.719 ± 0.779*	45.757 ± 0.287*	3.193 ± 0.043*	0.575 ± 0.012*	127.436 ± 0.606
TCDDM	22.148 ± 0.529	1.309 ± 0.030 *	0.382 ± 0.019	90.596 ± 0.574	45.563 ± 0.889 *	2.850 ± 0.058	0.569 ± 0.015	126.512 ± 0.491
CDiff	**21.013 ± 0.158**	**1.131 ± 0.017**	**0.351 ± 0.004**	**87.993 ± 0.178**	44.621 ± 0.139	**2.653 ± 0.022**	**0.551 ± 0.002**	**125.685 ± 0.151**
	StackOverflow				Retweet			
	OTD	**RMSE**$_e$	**RMSE**$_{x^+}$	**sMAPE**	**OTD**	**RMSE**$_e$	**RMSE**$_{x^+}$	**sMAPE**
HYPRO	42.359±0.170	1.140 ± 0.014	1.554 ± 0.010*	110.988 ± 0.559 *	61.031±0.092*	2.623 ± 0.036*	30.100 ± 0.413*	106.110± 1.505
Dual-TPP	41.752±0.200	**1.134 ± 0.019**	1.514 ± 0.017*	117.582 ± 0.420 *	61.095±0.101 *	2.679 ± 0.026*	28.914 ± 0.300	106.900± 1.293
AttNHP	42.591 ± 0.408*	1.142 ± 0.011	1.340 ± 0.006	108.542 ± 0.531	60.634 ± 0.097	2.561 ± 0.054	28.812 ± 0.272*	107.234± 1.293*
NHP	43.791 ± 0.147*	1.244 ± 0.030*	1.487 ± 0.004*	116.952 ± 0.404*	60.953 ± 0.079*	2.651 ± 0.045*	27.130 ± 0.224	107.075 ± 1.398*
LNM	46.280 ± 0.892*	1.447 ± 0.057 *	1.669 ± 0.005 *	115.122 ± 0.627 *	61.715 ± 0.152*	2.776 ± 0.043 *	27.582 ± 0.191	106.711 ± 1.615 *
TCDDM	42.128 ± 0.591	1.467 ± 0.014*	1.315 ± 0.004	107.659 ± 0.934	**60.501 ± 0.087**	2.387 ± 0.050	27.303 ± 0.152	**106.048 ± 0.610**
CDiff	**41.245 ± 1.400**	1.141 ± 0.007	**1.199 ± 0.006**	**106.175± 0.340**	60.661 ± 0.101	**2.293 ± 0.034**	**27.101 ± 0.113**	106.184 ± 1.121
	MOOC				Amazon			
	OTD	**RMSE**$_e$	**RMSE**$_{x^+}$	**sMAPE**	**OTD**	**RMSE**$_e$	**RMSE**$_{x^+}$	**sMAPE**
HYPRO	48.621 ± 0.352	1.169 ± 0.094	**0.410 ± 0.005**	**143.045 ± 7.992**	38.613 ± 0.536*	**2.007 ± 0.054**	0.477 ± 0.010*	82.506 ± 0.84
Dual-TPP	50.184 ± 1.127	1.312 ± 0.019*	0.435 ± 0.006*	147.003 ± 2.908*	42.646 ± 0.752*	2.562 ± 0.202	0.482 ± 0.012*	86.453 ± 2.044
AttNHP	49.121 ± 0.720*	1.297 ± 0.049	0.420 ± 0.009	147.756 ± 4.812	39.480 ± 0.326	2.166 ± 0.026*	0.476 ± 0.033	84.323 ± 1.815*
NHP	51.277 ± 1.768*	1.458 ± 0.063*	0.442 ± 0.007*	148.913 ± 11.628*	42.571 ± 0.293*	2.561 ± 0.060	0.519 ± 0.023*	92.053 ± 1.553*
LNM	52.890 ± 1.151*	1.428 ± 0.061*	0.454 ± 0.008*	149.987 ± 16.581*	43.820 ± 0.232*	3.050 ± 0.286*	0.481 ± 0.145*	90.910 ± 1.611*
TCDDM	50.739 ± 0.765*	1.407 ± 0.112*	0.429 ± 0.015	145.745 ± 11.835	42.245 ± 0.174*	2.998 ± 0.115*	0.476 ± 0.111	83.826 ± 1.508
CDiff	**47.214 ± 0.628**	**1.095 ± 0.048**	0.411 ± 0.009	146.361 ± 14.837*	**37.728 ± 0.199**	2.091 ± 0.163	**0.464 ± 0.086**	**81.987 ± 1.905**

ground truth exhibits many very short intervals (the first bin) and then a rapid drop. CDiff manages to follow this pattern, while also accurately capturing the number of events in the tail (the final bin). HYPRO and AttNHP struggle to match the rapid decay. In the bottom panel, they also fail to produce many large inter-arrival times. These observations may be attributed to the fact that HYPRO and AttNHP rely on exponential distributions to model time intervals and are autoregressive whereas our architecture does not rely on a parametric TPP model and jointly models the distribution of the N events in the sequence.

6.2. CDiff can forecast long horizon events

CDiff is explicitly designed to perform multi-event prediction, so we expect it to be better at predicting long horizon events, i.e., those near the end of the prediction horizon, such as events $N-1$ and N. To verify this, we perform the following experiment. We collect sequences of errors for time intervals. For each sequence to predict of length N, we compute a sequence of absolute errors made at each time interval: $[\delta_1, \ldots, \delta_N]$ where $\delta_i = |x_i^+ - \hat{x}_i^+|$. We can therefore construct a set of error sequences $\{[\delta_1^j, \ldots, \delta_N^j]\}_{j=1}^M$ for each baseline for a given dataset of M test sequences.

In general, a sequence of errors $[\delta_1, \ldots, \delta_N]$ is expected to increase, as it is harder to predict events further in the future. Our goal is to compare how fast this error is growing for the various forecasting approaches.

To do so, we use a one-tailed Wilcoxon signed-paired test to test our method, CDiff, against each baseline for each error step δ_i. We report the p-values for each different event index i. The tested null hypothesis is that the median of the population of differences between the paired data of CDiff error minus baseline error is equal or greater than zero. For later time intervals, rejection of the hypothesis implies that, with statistical significance, the median of the CDiff δ_i is smaller than the median of baseline δ_i. In Table 2, the p-values generally decrease as we move further into the future, showing an overall trend that the error of the competing baselines is increasing more rapidly than that of CDiff.

Table 2. The p-values obtained from the Wilcoxon signed-paired tests, comparing CDiff vs HYPRO, AttNHP, NHP and Dual-TPP at various future steps δ_i (step $i = 1, 5, 10, 20$).

Taobao	p-value δ_1	p-value δ_5	p-value δ_{10}	p-value δ_{20}
HYPRO	5.10e-3	1.704e-4	1.855e-06	2.099e-07
AttNHP	3.117e-1	6.157e-2	1.149e-3	9.440e-4
NHP	2.488e-09	2.777e-09	6.798e-11	1.061e-13
Dual-TPP	2.030e-05	1.511e-05	2.368e-13	3.725e-09
Stackoverflow	p-value δ_1	p-value δ_5	p-value δ_{10}	p-value δ_{20}
HYPRO	1.396e-07	1.913e-4	1.585e-10	1.427e-09
AttNHP	9.327e-4	3.192e-4	8.882e-06	4.146e-07
NHP	4.671e-3	8.490e-06	1.769e-3	3.127e-06
Dual-TPP	3.816e-05	8.887e-07	1.194e-08	3.542e-08

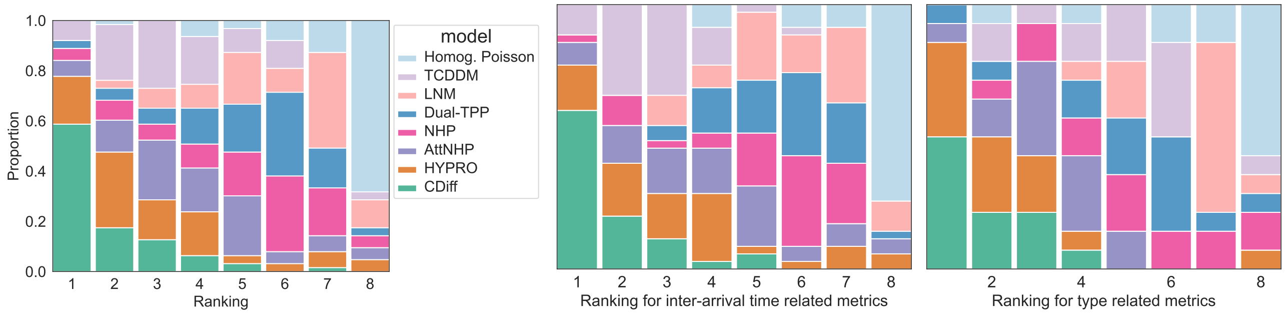


Figure 3. **Left)** Stacked column chart of ranks of the algorithms across the 5 datasets for all the metrics. We collect the rank for each metric (9 metrics in total, as we include additional metrics from the interval forecasting experiment described in the Appendix A.1). The x-axis is the rank, and the y-axis is the proportion adding up to 1. **Middle)** Stacked column chart of ranks only for time-related metrics (**RMSE**$_{x^+}$, **MAPE**, **sMAPE**, **RMSE**$_{|\mathbf{s}^+|}$, **MAE**$_{|\mathbf{s}^+|}$). **Right)** Stacked column chart of ranks only for type-related metric (**RMSE**$_e$).

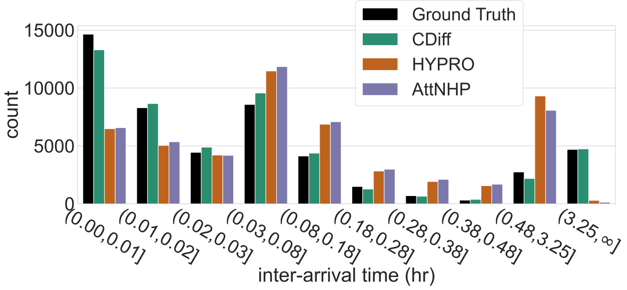


Figure 4. Histogram of true and predicted inter-arrival times for the Taobao dataset. Note that the bin widths gradually increase to make visual comparison easier.

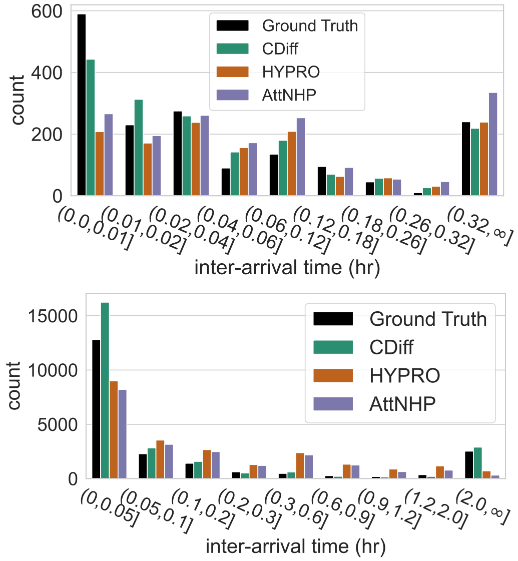


Figure 5. Histogram of true and predicted inter-arrival times for cases when the next event is type e=7 (top) and e=16 (bottom) for the Taobao dataset. Bin widths gradually increase so that counts are more comparable.

6.3. Forecasting shorter horizons

Figure 6 presents the results for shorter horizons: $N = 1, 5, 10$. See Appendix A.9 for more detailed results on the single event forecasting case, i.e., $N = 1$. All methods improve as we reduce the forecasting horizon. For **RMSE**$_\mathbf{e}$, all models perform similarly to $N = 20$. The performance difference grows as the prediction horizon increases. For **sMAPE**, CDiff outperforms the other models even for single event forecasting, and the outperformance increases rapidly with the prediction horizon. We attribute this to CDiff's ability to model more complex inter-arrival distributions.

6.4. Sampling and training efficiency

Table 4 summarizes the sampling time, number of trainable parameters, and training time for all methods across three datasets. Starting with sampling time, CDiff outperforms most models, except for LNM, due to its non-autoregressive nature allowing for simultaneous generation of all events in the sequence. Regarding space complexity, CDiff naturally has the largest number of parameters (except for Taxi, which is a simpler task) as the dimension of the predicted vectors is N times larger than all the other methods that generate one event at a time. To account for the complexity difference and ensure that CDiff's superior performance is not due to additional parameters, we conduct further experiments with a fixed number of parameters in Appendix A.7. In terms of training time, LNM is the most efficient, whereas CDiff, AttNHP, and NHP share similar training durations. Dual-TPP requires more time due to its count component, and HYPRO, which must also generate samples during training, demands the most training time.

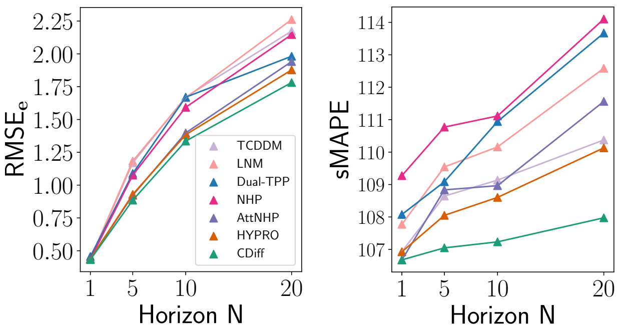


Figure 6. $RMSE_e$ and $sMAPE$, averaged over all datasets, for different horizons.

Table 3. Complexity analysis on three datasets for $N = 20$. The training time is for 500 epochs. The experiments were run on a GeForce RTX 2070 SUPER machine.

		HYPRO	DualTPP	Attnhp	LNM	TCDDM	CDiff
Taxi	**Sampling** (sec/s$_u$)	0.265	0.158	0.136	**0.079**	0.227	0.104
	num. param. (K)	40.5	40.1	19.3	19.1	20.3	**17.1**
	Training (mins)	95	45	35	**20**	45	35
Taobao	**Sampling** (sec/s$_u$)	0.325	0.240	0.209	**0.094**	0.285	0.129
	num. param. (K)	40.1	19.7	19.6	**17.3**	20.3	62.6
	Training (mins)	105	60	45	**30**	60	45
Stack.	**Sampling** (sec/s$_u$)	0.294	0.207	0.191	**0.111**	0.233	0.133
	num. param. (K)	41.0	40.3	20.1	**19.6**	20.3	63.9
	Training (mins)	105	60	45	**35**	60	45

6.5. Ablation – Joint vs Independent modeling of time and event type

In the ablation study, we verify whether it is necessary to model the joint distribution in order to achieve better performance. In order to demonstrate this, we conduct a new experiment by introducing an independent model. In this model, we model the future sequence $P(S_0|\mathbf{s}_c)$ using two independent processes (always conditioned on the same context $\mathbf{s}_c$) $P(S_0|\mathbf{s}_c) = P(X_0|\mathbf{s}_c)P(E_0|\mathbf{s}_c)$. In CDiff, we model the joint distribution by conditioning the time interval denoising distributions on the event type denoising distributions, and vice versa, as in Equations (9) and (10). In this ablation study we remove this interaction and strive to learn independent denoising distributions:

$$Cat(E_{t-1}|\pi_\theta(E_t, t, \mathbf{s}_c)) \tag{23}$$

$$\mathcal{N}(X_{t-1}|\mu_\theta(X_t, t, \mathbf{s}_c), \sigma_t) \tag{24}$$

The results are presented in Table 4. For each metric and dataset we present, CDiff is always better than CDiff-indep, confirming that modeling the joint distribution is necessary. Moreover, we can see that CDiff-indep is not even the second best baseline; HYPRO is the second best or best method for half of the metrics, while CDiff-indep is the second best for 3/8 of the metrics. The ablation study thus highlights that: i) using a diffusion model to predict a sequence of multiple events is an effective strategy, even if the dependencies between event type and time interval are ignored (CDiff-indep is the second or third-best method); ii) modelling the dependencies via cross-diffusion leads to a significant performance improvement.

Table 4. **OTD**, **RMSE**$_e$, **RMSE**$_{x^+}$ and **sMAPE** of Amazon and Stackoverflow reported in mean ± s.d. for comparison between conditional independent version of CDiff and the baselines. *indicates stat. significance w.r.t to the best method.

Amazon	**OTD**	**RMSE**$_e$	**RMSE**$_{x^+}$	**sMAPE**
HYPRO	$38.61 \pm 0.54^*$	$\mathbf{2.01 \pm 0.05}$	$0.48 \pm 0.01^*$	82.51 ± 0.84
Dual-TPP	$42.65 \pm 0.75^*$	2.56 ± 0.20	$0.48 \pm 0.01^*$	86.45 ± 2.04
AttNHP	39.48 ± 0.33	$2.17 \pm 0.03^*$	0.48 ± 0.03	$84.32 \pm 1.82^*$
NHP	$42.57 \pm 0.29^*$	2.56 ± 0.06	$0.52 \pm 0.02^*$	$92.05 \pm 1.55^*$
LNM	$43.82 \pm 0.23^*$	$3.05 \pm 0.29^*$	$0.48 \pm 0.15^*$	$90.91 \pm 1.61^*$
TCDDM	$42.25 \pm 0.17^*$	$3.00 \pm 0.12^*$	0.48 ± 0.11	83.83 ± 1.51
CDiff-indep	$40.49 \pm 0.60^*$	$2.70 \pm 0.24^*$	0.47 ± 0.04	$84.77\pm 1.32^*$
CDiff	$\mathbf{37.73 \pm 0.20}$	$\mathbf{2.09 \pm 0.16}$	$\mathbf{0.46 \pm 0.09}$	$\mathbf{81.99 \pm 1.91}$

Stackoverflow	**OTD**	**RMSE**$_e$	**RMSE**$_{x^+}$	**sMAPE**
HYPRO	42.36 ± 0.17	1.14 ± 0.01	$1.55 \pm 0.01^*$	$110.99 \pm 0.56\ ^*$
Dual-TPP	41.75 ± 0.20	$\mathbf{1.13 \pm 0.02}$	$1.51 \pm 0.02^*$	$117.58 \pm 0.42\ ^*$
AttNHP	$42.59 \pm 0.41^*$	1.14 ± 0.01	1.34 ± 0.01	108.54 ± 0.53
NHP	$43.79 \pm 0.15^*$	$1.24 \pm 0.03^*$	$1.49 \pm 0.00^*$	$116.95 \pm 0.40^*$
LNM	$46.28 \pm 0.89^*$	$1.45 \pm 0.06\ ^*$	$1.67 \pm 0.01\ ^*$	$115.12 \pm 0.63\ ^*$
TCDDM	42.13 ± 0.59	$1.47 \pm 0.01^*$	1.32 ± 0.00	107.66 ± 0.93
CDiff-indep	42.19 ± 0.14	$1.35 \pm 0.12^*$	1.26 ± 0.01	106.71 ± 0.44
CDiff	$\mathbf{41.25 \pm 1.40}$	1.14 ± 0.01	$\mathbf{1.20 \pm 0.01}$	$\mathbf{106.18 \pm 0.34}$

7. Limitations

Although offering impressive performance, there are limitations specific to our approach of modelling N events at once. Unlike previous autoregressive approaches, our method requires the practitioner to select a fixed number of events N to be modeled by the diffusion generative model. This can prove challenging when dealing with data that exhibits highly irregular time intervals (x_i^+). Essentially, if the length of time spanned by a fixed number of events varies significantly, then it will lead to a substantial variation in the nature and complexity of the forecasting task. This effect was not observed in the datasets we considered, as none displayed such high irregularities.

8. Conclusion

We have proposed a diffusion-based generative model, CDiff, for event sequence forecasting. Extensive experiments demonstrate the superiority of our approach over existing baselines for long horizons. The approach also offers improved sampling efficiency. Our analysis sheds light on the mechanics behind the improvements, revealing that our model excels at capturing intricate correlation structure and at predicting distant events.

Acknowledgement

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [funding reference number 260250] and of the Fonds de recherche du Québec.
Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), [numéro de référence 260250] et par les Fonds de recherche du Québec.

Impact Statement

Forecasting methods for temporal point processes are impactful as they have many applications. In general, accurate forecasting has the typical positive impact of optimizing resource usage efficiency but also can raise privacy concerns. In particular for TPP models, the specific applications of these algorithms and even the datasets used to benchmark those models include monitoring consumer behavior. This poses a potential risk of malicious exploitation.

References

Bacry, E., Bompaire, M., Deegan, P., Gaïffas, S., and Poulsen, S. V. Tick: a python library for statistical learning, with an emphasis on hawkes processes and time-dependent models. *J. Mach. Learn. Res.*, 18(214):1–5, 2018.

Bae, W., Ahmed, M. O., Tung, F., and Oliveira, G. L. Meta temporal point processes. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2023.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. Algorithms for hyper-parameter optimization. In *Adv. Neural Info. Process. Syst. (NeurIPS)*, 2011.

Bosser, T. and Taieb, S. B. On the predictive accuracy of neural temporal point process models for continuous-time event data. *Trans. on Mach. Learn. Res.*, June 2023.

Deshpande, P., Marathe, K., De, A., and Sarawagi, S. Long horizon forecasting with temporal point processes. In *Proc. Int. Conf. on Web Search and Data Mining (WSDM)*, 2021.

Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., and Song, L. Recurrent marked temporal point processes: Embedding event history to vector. In *Proc. Int. Conf. Data Min. Knowl. Discov. (SIGKDD)*, 2016.

Gupta, V., Bedathur, S. J., Bhattacharya, S., and De, A. Learning temporal point processes with intermittent observations. In *Proc. Int. Conf. on Artif. Intell. and Stats. (AISTAT)*, 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Adv. Neural Info. Process. Syst. (NeurIPS)*, 2020.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Adv. Neural Info. Process. Syst. (NeurIPS)*, 2021.

Kumar, S., Zhang, X., and Leskovec, J. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proc. Int. Conf. Data Min. Knowl. Discov. (SIGKDD)*, 2019.

Leskovec, J. and Krevl, A. SNAP Datasets: Stanford large network dataset collection, 2014.

Lin, H., Wu, L., Zhao, G., Liu, P., and Li, S. Z. Exploring generative neural temporal point process. *Trans. Mach. Learn. Res.*, August 2022.

Liniger, T. J. *Multivariate Hawkes Processes*. PhD thesis, ETH Zurich, 2009.

Lüdke, D., Biloš, M., Shchur, O., Lienen, M., and Günnemann, S. Add and thin: Diffusion for temporal point processes. In *Adv. Neural Info. Process. Syst. (NeurIPS)*, 2023.

Mei, H. and Eisner, J. The neural hawkes process: A neurally self-modulating multivariate point process. In *Adv. Neural Info. Process. Syst. (NeurIPS)*, 2017.

Mei, H., Qin, G., and Eisner, J. Imputing missing events in continuous-time event streams. In *Proc. Int. Conf. Machine Learning. (ICML)*, 2019.

Ni, J., Li, J., and McAuley, J. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proc. Conf. Empir. Methods Nat. Lang. Process. and the Int. Joint Conf. Nat. Lang. Process. (EMNLP-IJCNLP)*, 2019.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *Proc. Int. Conf. Machine Learning. (ICML)*, 2021.

Nickel, M. and Le, M. Learning multivariate hawkes processes at scale. arXiv preprint arXiv:2002.12501, 2020.

Omi, T., ueda, n., and Aihara, K. Fully neural network based model for general temporal point processes. In *Adv. Neural Info. Process. Syst. (NeurIPS)*, 2019.

Pan, Z., Wang, Z., Phillips, J. M., and Zhe, S. Self-adaptable point processes with nonparametric time decays. In *Adv. Neural Info. Process. Syst. (NeurIPS)*, 2021.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Adv. Neural Info. Process. Syst. (NeurIPS)*, volume 32, 2019.

Rasmussen, J. G. Lecture notes:Temporal point processes and the conditional intensity function. arXiv preprint arXiv:1806.00221, 2011.

Shchur, O., Biloš, M., and Günnemann, S. Intensity-free learning of temporal point processes. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2020a.

Shchur, O., Gao, N., Biloš, M., and Günnemann, S. Fast and flexible temporal point processes with triangular maps. In *Adv. Neural Info. Process. Syst. (NeurIPS)*, 2020b.

Shchur, O., Turkmen, A. C., Januschowski, T., and Günnemann, S. Neural temporal point processes: A review. In *Proc. Int. Joint Conf. on Artif. Intell. (IJCAI)*, 2021.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2021.

Tukey, J. W. On the comparative anatomy of transformations. *Ann. Math. Stat.*, 28(3):602–632, 1957.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020.

Whong, C. Foiling nyc's taxi trip data, 2014.

Xue, S., Shi, X., Zhang, J. Y., and Mei, H. HYPRO: A hybridly normalized probabilistic model for long-horizon prediction of event sequences. In *Adv. Neural Info. Process. Syst. (NeurIPS)*, 2022.

Yang, C., Mei, H., and Eisner, J. Transformer embeddings of irregularly spaced events and their participants. In *Proc. Int. Conf. Learn. Represent.*, 2022.

Zhang, Q., Lipani, A., and Yilmaz, E. Learning neural point processes with latent graphs. In *Proc. Web Conf. (WWW)*, 2021.

Zhou, K., Zha, H., and Song, L. Learning triggering kernels for multi-dimensional hawkes processes. In *Proc. Int. Conf. Machine Learning. (ICML)*, 2013.

Zhu, H., Li, X., Zhang, P., Li, G., He, J., Li, H., and Gai, K. Learning tree-based deep model for recommender systems. In *Proc. Int. Conf. Data Min. Knowl. Discov. (SIGKDD)*, 2018.

Zuo, S., Jiang, H., Li, Z., Zhao, T., and Zha, H. Transformer hawkes process. In *Proc. Int. Conf. Machine Learning. (ICML)*, 2020.

A. Appendix

A.1. Interval Forecasting

In this time-based setting, the task is to predict the events that occur within a given subsequent time interval t', i.e., $\mathbf{s}_u^+$: $\mathbf{x}_u^+ = [x_{I+1}^+, ...]$ and $\mathbf{e}_u = [e_{I+1}, ...]$ such that $||\mathbf{x}_u^+||_1 \leq t'$.

This different setting also calls for different metrics, and the predicted $\hat{\mathbf{s}}_u^+$ and ground truth $\mathbf{s}_u^+$ can have a different number of events. We report both **OTD** and the **RMSE**$_e$ metrics as they are robust to a varying number of events. We also report additional metrics that compare the number of events predicted:

1. **MAE**$_{|\mathbf{s}^+|} = \frac{1}{M}\sum_{j=1}^M \big||\mathbf{s}_u^{+,j}| - |\hat{\mathbf{s}}_u^{+,j}|\big|$;

2. **RMSE**$_{|\mathbf{s}^+|} = \sqrt{\frac{1}{M}\sum_{j=1}^M (|\mathbf{s}_u^{+,j}| - |\hat{\mathbf{s}}_u^{+,j}|)^2}$.

For our experiment, we retain the same context sequences $\mathbf{s}_c$ that were used for the **next N events forecasting setting.**. Table 5 details the time interval values t' of three experiments (long, medium and short horizon) for each dataset.

Table 5. Time interval for interval forecasting problem.

Dataset	t' long	t' medium	t' short	train/val/test	units
Synthetic	2	1	0.5	1500/400/500	second
Taxi	4.5	2.25	1.125	1300/200/400	hour
Taobao	19.5	9.25	5.25	1300/200/500	hour
Stack.	220	110	55	1400/400/400	day
Retweet	500	250	150	1400/600/800	second
MOOC	3.5	1.5	1	2400/717/1039	hour
Amazon	20	10	5	3500/1000/1500	hour

A.2. CDiff methodology for interval Forecasting

To adapt our CDiff model to this setting, we select a number of events, denoted as N, and repeatedly generate N-length sequences until we reach the end of the forecasting window t'. That is, while $||\mathbf{x}_u^+||_1 \leq t'$, we integrate the current $\mathbf{s}_u^+$ into the context $\mathbf{s}_c^+$ and regenerate N additional events that we attach at the end of $\mathbf{s}_u^+$. We set N to be the maximum number of events observed within the given time interval in the training data.

A.3. Metrics details and more OTD results

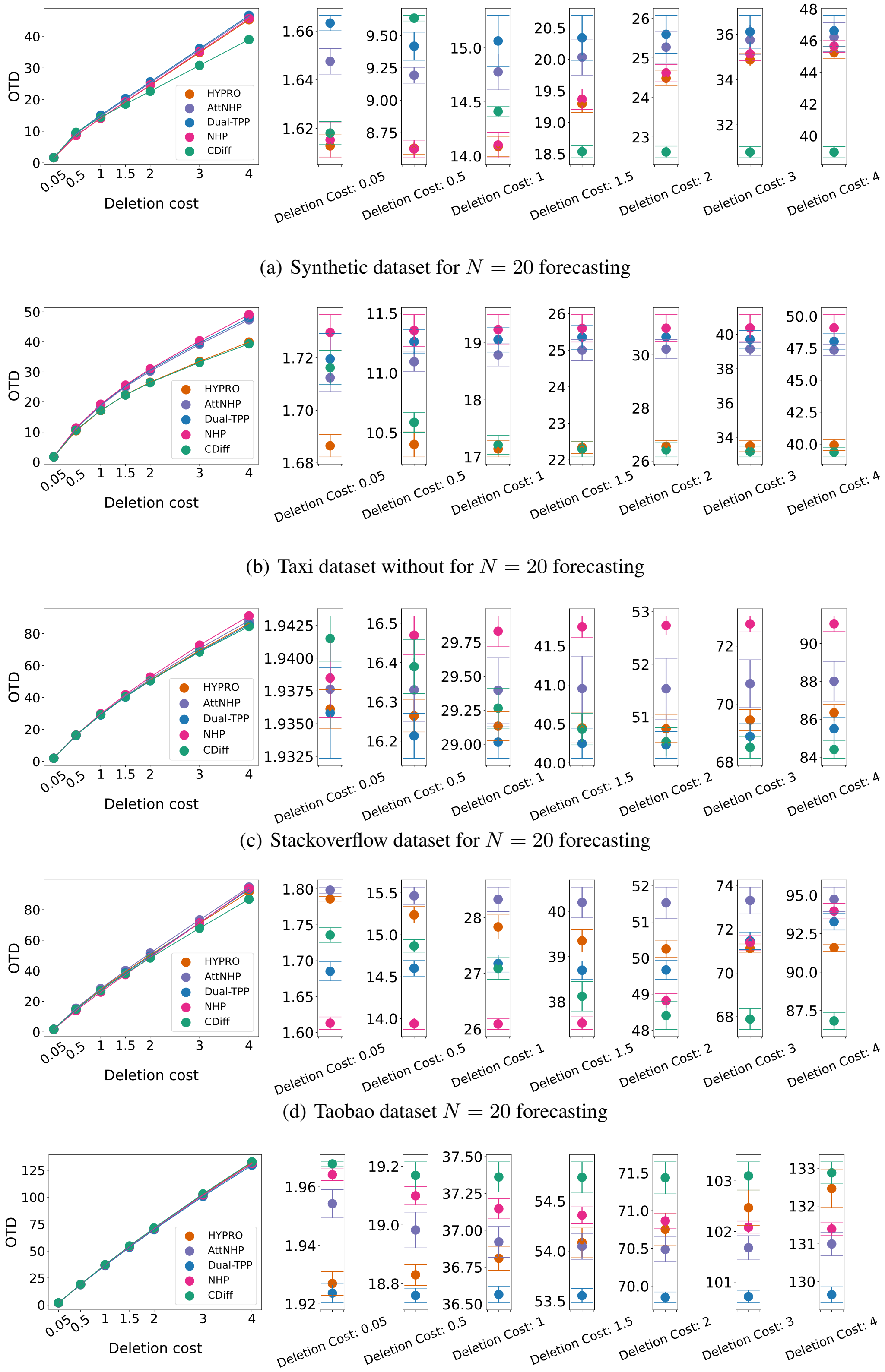
The time-interval metrics are given by;

$$RMSE_x = \sqrt{\frac{1}{M}\sum_{j=1}^{M} ||\mathbf{x}_u^{+,j} - \hat{\mathbf{x}}_u^{+,j}||_2^2}, \tag{25}$$

$$MAPE = \frac{1}{M}\sum_{j=1}^{M}\frac{100}{N}\sum_{i=1}^{N}\frac{|x_{u,i}^{+,j} - \hat{x}_{u,i}^{+,j}|}{|x_{u,i}^{+,j}|} \tag{26}$$

$$sMAPE = \frac{1}{M}\sum_{j=1}^{M}\frac{100}{N}\sum_{i=1}^{N}\delta_i^j, \delta_i^j = \frac{2|x_{u,i}^{+,j} - \hat{x}_{u,i}^{+,j}|}{|x_{u,i}^{+,j}| + |\hat{x}_{u,i}^{+,j}|}. \tag{27}$$

In the calculation of Optimal Transport Distance (OTD), the deletion cost hyperparameter, denoted by C_{del}, plays a pivotal role. Yang et al. (2022) provided a full description and pseudo code for the dynamic algirthm to calculate the OTD. This parameter quantifies the expense associated with the removal or addition of an event token, irrespective of its category. For our experimentation, we chose a variety of C_{del} values—$0.05, 0.5, 1, 1.5, 2, 3, 4$—based on the recommendations provided by (Xue et al., 2022). Subsequently, we calculated the mean OTD. In the following section, the OTD metrics are delineated for each individual C_{del} value. As evidenced by Fig. 7 and 8, our model outperforms across the board for the varying



(a) Synthetic dataset for $N = 20$ forecasting

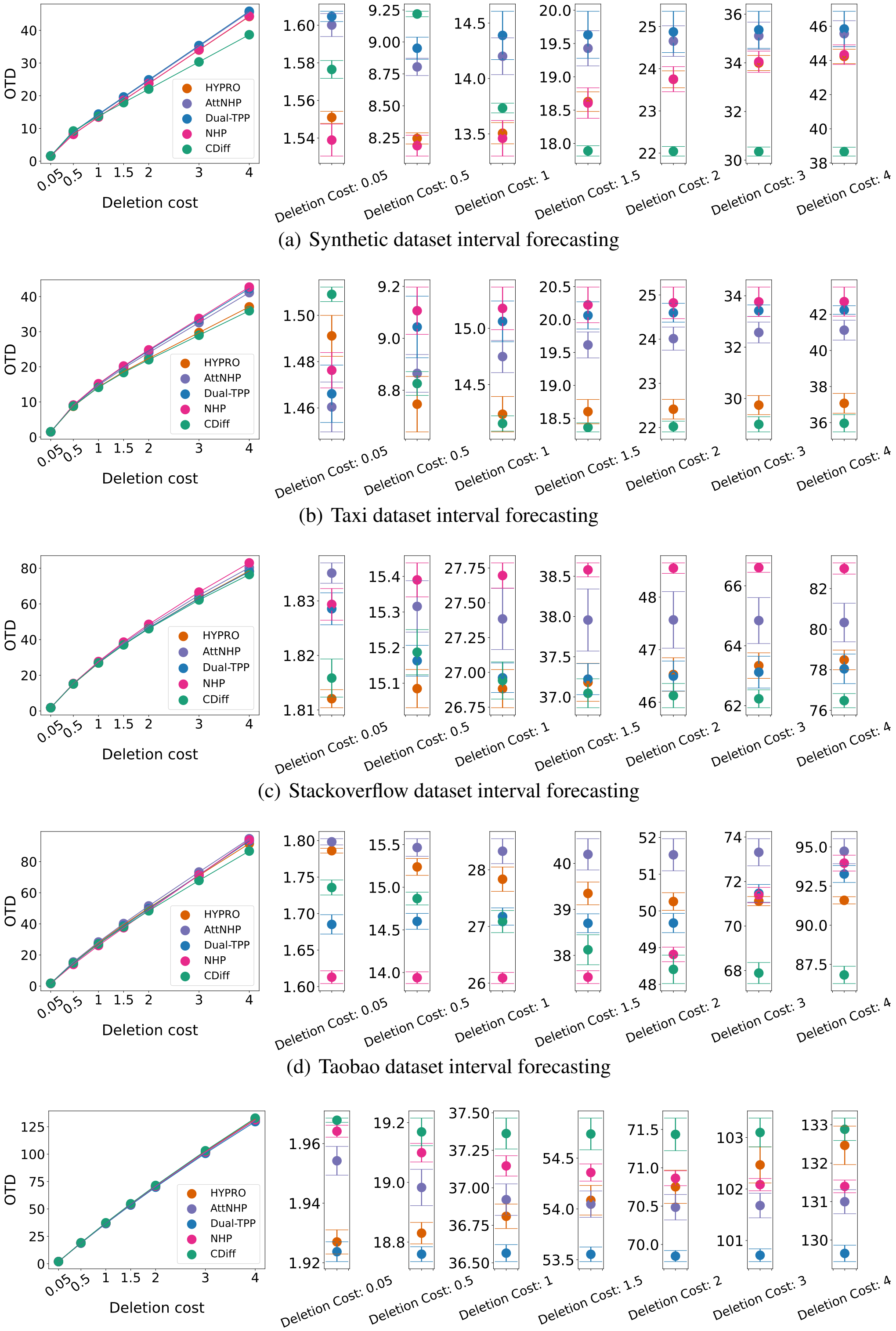
(b) Taxi dataset without for $N = 20$ forecasting

(c) Stackoverflow dataset for $N = 20$ forecasting

(d) Taobao dataset $N = 20$ forecasting

(e) Retweet dataset $N = 20$ forecasting

Figure 7. OTD for each specific deletion/addition cost for $N = 20$ forecasting, we chose a variety of C_{del} values—$0.05, 0.5, 1, 1.5, 2, 3, 4$—based on the recommendations in (Xue et al., 2022). Subsequently, we calculated the mean and s.d. of OTD across all the datasets.



(a) Synthetic dataset interval forecasting

(b) Taxi dataset interval forecasting

(c) Stackoverflow dataset interval forecasting

(d) Taobao dataset interval forecasting

(e) Retweet dataset interval forecasting

Figure 8. OTD for each specific deletion/addition cost for interval forecasting. We calculated the mean and s.d. of OTD across all the datasets for different C_{del} values.

C_{del} settings overall. We also see that the OTD steadily increases overall, and that different C can permute the ordering of the competing baselines. For low C_{del}, our method is outperformed by HYPRO and AttNHP sometimes, but this trend is reversed for larger C_{del} values for almost all datasets. This reflects the fact that the proposed CDiff method is better at predicting the number of events, so fewer deletions or additions are required.

A.4. Dataset details

- **Taobao** (Zhu et al., 2018) This dataset captures user click events on Taobao's shopping websites between November 25 and December 03, 2017. Each user's interactions are recorded as a sequence of item clicks, detailing both the timestamp and the item's category. All item categories were ranked by frequency, with only the top 16 retained; the remaining were grouped into a single category. Thus, we have $K = 17$ distinct event types, each corresponding to a category. The refined dataset features 2,000 of the most engaged users, with an average sequence length of 58. The disjoint train, validation and test sets consist of 1300, 200, and 500 sequences (users), respectively, randomly sampled from the dataset. The time unit is 3 hours; the average inter-arrival time is 0.06 (i.e., 0.18 hour).

- **Taxi** (Whong, 2014) This dataset contains time-stamped taxi pickup and drop off events with zone location ids in New York city in 2013. Following the processing procedure of (Mei et al., 2019), each event type is defined as a tuple of (location, action). The location is one of the 5 boroughs (Manhattan, Brooklyn, Queens, The Bronx, Staten Island). The action can be either pick-up or drop-off. Thus, there are $K = 5 \times 2 = 10$ event types in total. The values $k = 0, \dots, 4$ indicate pick-up events and $k = 5, \dots, 9$ indicate drop-off events. A subset of 2000 sequences of taxi pickup events with average length 39 are retained. The average inter-arrival time is 0.22 hour (time unit is 1 hour.) The disjoint train, validation and test sets are randomly sampled and are of sise 1400, 200, and 400 sequences, respectively.

- **StackOverflow** (Leskovec & Krevl, 2014) This dataset contains two years of user awards from a question-answer platform. Each user was awarded a sequence of badges, with a total of $K = 22$ unique badge types. The train, validation and test sets consist of 1400, 400 and 400 sequences, resepctively, and are randomly sampled from the dataset. The time unit is 11 days; the average inter-arrival time is 0.95.

- **Retweet** (Zhou et al., 2013) This dataset contains sequences of user retweet events, each annotated with a timestamp. These events are segregated into three categories ($K = 3$), denoted by: "small", "medium", and "large" users. Those with under 120 followers are labeled as small users; those with under 1363 followers are medium users, while the remaining users are designated as large users. Our studies focus on a subset of 9000 retweet event sequences. The disjoint train, validation and test sets consist of 6000, 1500, and 1500 sequences, respectively, randomly sampled from the dataset.

- **MOOC** (Kumar et al., 2019) This datasets contains sequences of records of student interactions within an online course platform. Each interaction represents an event and can manifest in different forms (97 distinct types), such as viewing a video, completing a quiz, and other activities. We utilized the pre-processing approach described by (Bosser & Taieb, 2023) in their extensive study on temporal point processes. This involved narrowing down the event types to a total of 50. Observing that a significant number of event sequences had less than or equal to 20 events, we chose to exclude these shorter sequences. Consequently, this process resulted in retaining 4,156 out of the initial 7,047 sequences, focusing on those with more than 20 events.

- **Amazon** (Ni et al., 2019) The dataset contains time-stamped user product review behavior from January 2008 to October 2018. It consists of sequences of product review events for individual users. Each event in these sequences includes the timestamp and the category of the product reviewed, with every category corresponding to a distinct event type. The study is conducted on a subset comprising the 5200 most active users, each having an average sequence length of 70 events. This led to a refinement of the event types to a total of $K = 16$.

- **Synthetic Multivariate Hawkes Dataset** The synthetic dataset is generated using the **tick**[2] package provided by Bacry et al. (2018), using the Hawkes process generator. Our study uses the same equations proposed by Lin et al. (2022). There are 5 event types. The impact function $g_{j,i}(y)$ measuring the relationship (impact) of type i on type j and is

[2] **tick** package can be found at `https://github.com/X-DataInitiative/tick`

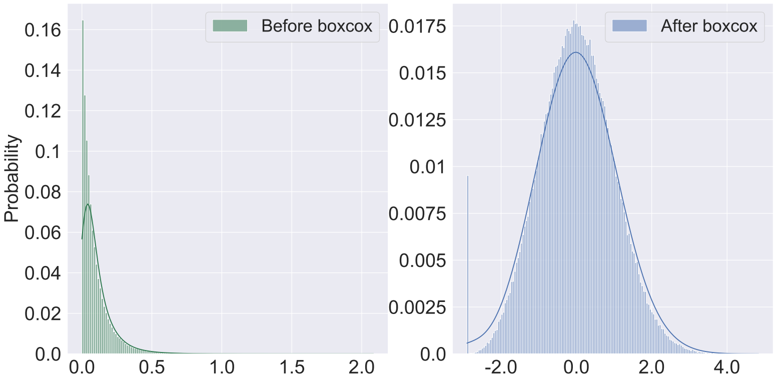


Figure 9. Inter-arrival time marginal histogram for synthetic dataset before (left) and after (right) boxcox transformation

uniformly-randomly chosen from the following four functions:

$$\begin{aligned} g_a(y) &= 0.99\exp(-0.4y) \\ g_b(y) &= 0.01\exp(-0.8y) + 0.03\exp(-0.6y) + 0.05\exp(-0.4y) \\ g_c(y) &= 0.25|\cos 3y|\exp(-0.1y) \\ g_d(y) &= 0.1(0.5+y)^2 \end{aligned} \tag{28}$$

A.5. Box-Cox Transformation

For our study, the inter-arrival time marginal distribution shown in Fig.9 (left) is clearly not a normal distribution. Since the diffusion probabilistic model we employ is a Gaussian-based generative model, we use the Box-Cox transformation to transform the inter-arrival time data, so that the transformed data approximately obeys a normal distribution.

The Box-Cox transformation (Tukey, 1957) is a family of power transformations that are used to stabilize variance and make data more closely follow a normal distribution. The transformation is defined as:

$$x(\lambda) = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \log(x) & \text{if } \lambda = 0. \end{cases} \tag{29}$$

Here:

- x is the original data;
- $x(\lambda)$ is the transformed data; and
- λ is the transformation parameter.

The inter-arrival time is strictly larger than 0 but it can be extremely small because of the scale of the dataset. Therefore, in order to prevent numerical errors in tbe Box-Cox transformation we add 1×10^{-7} time units to all inter-arrival times. We then scale all values by 100. We use the scaled inter-arrival time data from the train set to obtain the fitted λ shown in Eq.29 and apply the transformation with the fitted λ to the inter-arrival time data for both the validation dataset and test dataset. Fig.9 shows an example of marginal histogram of inter-arrival time for the Synthetic train set before (left) and after (right) the Box-cox transformation. We transform back the predicted sequence inter-arrival times with the same fitted λ obtained from the train set and undo the scaling by 100. We use the Box-cox transformation function from the **SciPy**[3] package provided by Virtanen et al. (2020).

[3] The SciPy package is available at `https://github.com/scipy/scipy`

Table 6. Sets of hyperparameters. Underlined values are those selected by the Tree-Structured Parzen Estimator (Bergstra et al., 2011)

Parameters	num. heads	num. layers	time embedding	Transformer feed-forward embedding	num. diffusion steps	LR
Synthetic	{1,2,4}	{1, 2, 4}	{4, 8, 16, 32, 64, 128}	{8, 16, 32, 64, 128, 256}	{50, 100, 200, 300, 500}	{0.001, 0.0025, 0.005}
Taxi	{1,2,4}	{1, 2, 4}	{4, 8, 16, 32, 64, 128}	{8, 16, 32, 64, 128, 256}	{50, 100, 200, 300, 500}	{0.001, 0.0025, 0.005}
Taobao	{1,2,4}	{1, 2, 4}	{4, 8, 16, 32, 64, 128}	{8, 16, 32, 64, 128, 256}	{50, 100, 200, 300, 500}	{0.001, 0.0025, 0.005}
Stackoverflow	{1,2,4}	{1, 2, 4}	{4, 8, 16, 32, 64, 128}	{8, 16, 32, 64, 128, 256}	{50, 100, 200, 300, 500}	{ 0.001, 0.0025, 0.005}
Retweet	{1,2,4}	{1, 2, 4}	{4, 8, 16, 32, 64, 128}	{8, 16, 32, 64, 128, 256}	{50, 100, 200, 300, 500}	{0.001, 0.0025, 0.005}
MOOC	{1,2,4}	{1, 2, 4}	{4, 8, 16, 32, 64, 128}	{8, 16, 32, 64, 128, 256}	{50, 100, 200, 300, 500}	{0.001, 0.0025, 0.005}
Amazon	{1,2,4}	{1, 2, 4}	{4, 8, 16, 32, 64, 128}	{8, 16, 32, 64, 128, 256}	{50, 100, 200, 300, 500}	{0.001, 0.0025, 0.005}

A.6. Hyper-parameters

Table 6 specifies the hyperparameters that we use for our experiments and the candidate values. We train for a maximum of 500 epochs and we select the best hyperparameters using the Tree-Structured Parzen Estimator (Bergstra et al., 2011). We have also performed a sensitivity study for the number of diffusion steps. As we can see in the following table, the method is not too sensitive to the number of diffusion steps. There is no sudden variation of performance as we gradually decreases the number of diffusion steps. A too low number of steps (25 and 50 steps in our case) is worst overall for all metrics and datasets. Once we use more steps (100, 200 or 500 steps) the performance becomes similar and the hyperparameter search sill select the best number of steps for each dataset.

Table 7. Ablation study on the number of diffusion steps. *indicates stat. significance w.r.t to the best method.

Taxi (Diffusion Step 100)	**OTD**	**RMSE$_e$**	**RMSE$_{x^+}$**	**sMAPE**
CDiff	**21.013** ± **0.158**	**1.131** ± **0.017**	**0.351** ± **0.004**	87.993 ± 0.178
CDiff-25	22.083 ± 0.410	1.135 ± 0.022	**0.351** ± **0.004**	87.963 ± 0.252
CDiff-50	21.045 ± 0.228	**1.131** ± **0.019**	0.352 ± 0.007	88.129 ± 0.193
CDiff-100	–	–	–	–
CDiff-200	21.545 ± 0.314	1.133 ± 0.015	**0.351** ± **0.010**	**87.839** ± **0.397**
CDiff-500	22.107 ± 0.244	1.138 ± 0.036	0.353 ± 0.009	88.053 ± 0.480

StackOverflow (Diffusion Step 200)	**OTD**	**RMSE$_e$**	**RMSE$_{x^+}$**	**sMAPE**
CDiff	**41.245** ± **1.400**	1.141 ± 0.007	**1.199** ± **0.006**	**106.175**± **0.340**
CDiff-25	42.742 ± 0.146*	1.169 ± 0.030*	1.331 ± 0.016*	109.941± 0.322*
CDiff-50	42.094 ± 0.444	1.172± 0.042*	1.306 ± 0.008	107.055± 0.400
CDiff-100	41.578 ± 0.261	**1.139** ± **0.017**	1.27 ± 0.022	105.365 ± 0.59
CDiff-200	–	–	–	–
CDiff-500	41.507 ± 0.16	1.153 ± 0.019	1.181 ± 0.23	107.842 ± 0.500

A.7. Comparison with fixed model size

To ensure a fair comparison, we conducted an experiment to compare CDiff and AttNHP with a similar number of parameters. We employed two methods to increase the number of parameters for AttNHP.

In the first way, we attach additional inference heads to the autoregressive baselines, until the number of parameters matches the size of our model. We denote the number of additional heads that are predicting future events in the name (for example, **method** with 2 heads is denoted as **method-2**). If the baseline has 2 additional heads, the first head is trained to predict the next event of the sequence (as usual), and the second head is trained to predict the second future event of the sequence. At inference, the model predicts the next 2 events, then integrates those 2 events into the context sequence s_c to predict the next 2 events (which would then be the 3rd and 4-th prediction) until N events have been predicted.

In the second, we increase the number of parameters for the baselines to match the number of parameters of our method. We denote it by **method-L**. Since we already used a hyperparameter search for each baseline, the increased number of parameters did not improve the results. We include it for completeness.

In the Table.8, we can see that neither the inclusion of additional heads nor the increase in the number of parameters is sufficient to reach CDiff's performance. Adding additional heads actually hurts the performance; for all datasets and all metrics, **AttNHP-3** is worse than the initial baseline **AttNHP**.

Although it is true that our model has more parameters than the baselines (approximately 1.5x more), it is better than the baselines in terms of the other complexity metrics that we report (namely training time and sampling time).

Table 8. Comparison between multi-head AttNHP, AttNHP with more parameters and CDiff. **OTD**, **RMSE**$_e$, **RMSE**$_{x+}$ and **sMAPE** of real-world datasets reported in mean ± s.d. Best are in bold, the next best is underlined.

Taxi	**OTD**	**RMSE**$_e$	**RMSE**$_{x+}$	**sMAPE**
AttNHP	$24.762 \pm 0.217^*$	$1.276 \pm 0.015^*$	$\underline{0.430 \pm 0.003}^*$	$\underline{97.388 \pm 0.381}^*$
AttNHP-3	$26.376\pm 0.229^*$	$1.554 \pm 0.022^*$	$0.452\pm 0.005^*$	$105.860\pm 0.504^*$
AttNHP-L	$\underline{24.174\pm 0.245}^*$	$\underline{1.274 \pm 0.022}^*$	$0.434 \pm 0.002^*$	$97.645\pm 0.693^*$
CDiff	$\mathbf{21.013 \pm 0.158}$	$\mathbf{1.131 \pm 0.017}$	$\mathbf{0.351 \pm 0.004}$	$\mathbf{87.993 \pm 0.178}$

Taobao	**OTD**	**RMSE**$_e$	**RMSE**$_{x+}$	**sMAPE**
AttNHP	$\underline{45.555 \pm 0.345}^*$	$\underline{2.737 \pm 0.021}$	$0.708 \pm 0.010^*$	$134.582 \pm 0.920^*$
AttNHP-3	$48.967\pm 0.072^*$	$3.877\pm 0.012^*$	$0.933\pm 0.005^*$	$136.130\pm 0.619^*$
AttNHP-L	$46.515\pm 0.191^*$	2.897 ± 0.019	$\underline{0.697\pm 0.005}$	$\underline{132.276\pm 0.993}^*$
CDiff	$\mathbf{44.621 \pm 0.139}$	$\mathbf{2.653 \pm 0.022}$	$\mathbf{0.551 \pm 0.002}$	$\mathbf{125.685 \pm 0.151}$

A.8. Add-and-thin comparison

Lüdke et al. (2023) proposed Add-and-thin model to perform multi-step forecasting for time intervals only. There is no consideration or modeling of event type. Since our work is focused on modeling the joint interaction between time intervals and event types, we cannot directly comparison with this method. However, for completeness, we can include a modified version by augmenting the add-thin-add model with a simple event type predictor module. This event type predictor model is based on the marginal probabilities of the training set. As we can see in the Table.9, this modified **Add-and-thin-augm.** is outperformed by CDiff for both the event type metrics and the time interval metrics, further demonstrating the importance of modeling the joint interaction of type and time. We would stress, however, that this modified version of Add-and-thin was not presented in the original paper.

Table 9. **OTD**, **RMSE**$_e$, **RMSE**$_{x+}$ and **sMAPE** of real-world datasets reported in mean ± s.d. Best are in bold, the next best is underlined. *indicates stat. significance w.r.t to the best method.

Taxi	**OTD**	**RMSE**$_e$	**RMSE**$_{x+}$	**sMAPE**
HYPRO	$\underline{21.653 \pm 0.163}$	$\underline{1.231 \pm 0.015}^*$	$\underline{0.372 \pm 0.004}^*$	$\underline{93.803 \pm 0.454}^*$
LNM	$24.053 \pm 0.609^*$	$1.364 \pm 0.032^*$	$0.384 \pm 0.005^*$	$95.719 \pm 0.779^*$
Add-and-Thin-augm.	$24.929 \pm 0.737^*$	–	$0.632 \pm 0.018\ ^*$	$107.070 \pm 0.590\ ^*$
CDiff	$\mathbf{21.013 \pm 0.158}$	$\mathbf{1.131 \pm 0.017}$	$\mathbf{0.351 \pm 0.004}$	$\mathbf{87.993 \pm 0.178}$
Taobao	**OTD**	**RMSE**$_e$	**RMSE**$_{x+}$	**sMAPE**
HYPRO	$\mathbf{44.336 \pm 0.127}$	$\underline{2.710 \pm 0.021}^*$	$\underline{0.594 \pm 0.030}^*$	$134.922 \pm 0.473^*$
LNM	$45.757 \pm 0.287^*$	$3.193 \pm 0.043^*$	$0.575 \pm 0.012^*$	127.436 ± 0.606
Add-and-Thin-augm.	$49.030 \pm 0.943^*$	–	$1.300 \pm 0.032\ ^*$	$144.597 \pm 0.699\ ^*$
CDiff	44.621 ± 0.139	2.653 ± 0.022	0.551 ± 0.002	125.685 ± 0.151
StackOverflow	**OTD**	**RMSE**$_e$	**RMSE**$_{x+}$	**sMAPE**
HYPRO	42.359 ± 0.170	$\underline{1.140 \pm 0.014}$	$1.554 \pm 0.010^*$	$110.988 \pm 0.559\ ^*$
LNM	$46.280 \pm 0.892^*$	$1.447 \pm 0.057\ ^*$	$1.669 \pm 0.005\ ^*$	$115.122 \pm 0.627\ ^*$
Add-and-Thin-augm.	$45.693 \pm 0.368^*$	–	$1.620\pm 0.090\ ^*$	$111.468 \pm 0.702^*$
CDiff	$\mathbf{41.245 \pm 1.400}$	1.141 ± 0.007	$\mathbf{1.199 \pm 0.006}$	$\mathbf{106.175\pm 0.340}$
Retweet	**OTD**	**RMSE**$_e$	**RMSE**$_{x+}$	**sMAPE**
HYPRO	$61.031\pm0.092^*$	$2.623 \pm 0.036^*$	$30.100 \pm 0.413^*$	$\underline{106.110\pm 1.505}$
LNM	$61.715 \pm 0.152^*$	$2.776 \pm 0.043\ ^*$	27.582 ± 0.191	$106.711 \pm 1.615\ ^*$
Add-and-Thin-augm.	$61.013 \pm 0.190^*$	–	$32.010 \pm 0.046\ ^*$	$116.895 \pm 0.607\ ^*$
CDiff	60.661 ± 0.101	$\mathbf{2.293 \pm 0.034}$	$\mathbf{27.101 \pm 0.113}$	106.184 ± 1.121

A.9. Next single event prediction

Figure 6 in our paper illustrates how the sequence length for prediction affects the overall ranking of the baselines. We include the specific results for $N = 1$ in Table 10 format here to improve readability for Figure 6.

A.10. Detailed Model details

Here we explain the learnable functions in Eq.11 and Eq.12: $\phi_\theta(e_t, x_t, s, t)$ and $\epsilon_\theta(e_t, x_{t-1}, s, t)$ serve as denoising functions for the type diffusion process and the time diffusion process, respectively.

Table 10. **RMSE$_{x^+}$**, **Accuracy**, **sMAPE** and **Error Rate** for $N = 1$ of real-world datasets reported in mean ± s.d. Since we only have one event, we can report the **Error Rate** of our single event type prediction. Best are in bold, the next best is underlined. HYPRO and Dual-TPP with single event forecasting will become AttNHP and RMTPP. *indicates stat. significance w.r.t to the best method.

Taxi	**RMSE$_{x^+}$** $\downarrow$	**Accuracy** $\uparrow$	**sMAPE** $\downarrow$
AttNHP	**0.321 ± 0.003**	0.905 ± 0.007	**85.132 ± 0.261**
RMTPP	0.335 ± 0.006	0.907 ± 0.010	89.115 ± 0.753
NHP	0.340 ± 0.007	**0.910 ± 0.007**	90.625 ± 0.608
LNM	0.377 ± 0.009*	0.904 ± 0.007	90.032 ± 0.470*
CDiff	0.337 ± 0.009	0.909 ± 0.004	87.124 ± 0.608

Taobao	**RMSE$_{x^+}$** $\downarrow$	**Accuracy** $\uparrow$	**sMAPE** $\downarrow$
AttNHP	0.527 ± 0.004	0.468 ± 0.011	129.133 ± 1.354
RMTPP	0.531 ± 0.007*	0.468 ± 0.021	131.432 ± 1.992*
NHP	0.531 ± 0.004*	0.458 ± 0.009	133.693 ± 2.246*
LNM	0.532 ± 0.007*	0.450 ± 0.007*	**126.009 ± 1.482**
CDiff	**0.516 ± 0.009**	**0.477 ± 0.003**	127.121 ± 1.356

- The history embedding is derived from a history encoder that processes $\mathbf{s}$, transforming it into an embedding with a dimension of $4 \times M$. This results in the overall sequence having dimensions of $R^{L\times(4\times M)}$.
- For time values t within the range of 0 to N_{step}, where N_{step} denotes the total number of diffusion steps, the time t undergoes a transformation into a positional encoding as described in Equation 21. The positional encoding $\mathbf{t}$ is then a vector in $\mathbb{R}^M$.
- $\mathbf{x}_t, \in \mathbb{R}^L_+$, has its dimension expanded to $\mathbb{R}^{L\times M}$ for the function $\phi_\theta(e_t, x_t, s, t)$, $\mathbb{R}^{L\times(2\times M)}$ for $\epsilon_\theta(e_t, x_{t-1}, s, t)$ with the same transformation specified in Equation 21.
- $\mathbf{e}_t$ denotes a sequence of one-hot vectors. Its dimension is augmented through a learnable event embedding matrix in $R^{M\times K}$, with the k-th column providing an M-dimensional embedding for the event type k. The resulting sequence embedding falls within $\mathbb{R}^{L\times M}$ for the function $\epsilon_\theta(e_t, x_{t-1}, s, t)$, $\mathbb{R}^{L\times(2\times M)}$ for $\phi_\theta(e_t, x_t, s, t)$
- To get the sequence order of the event tokens, an additional positional encoding specific to the order is computed. The event token order is converted into a positional encoding following Equation 21, resulting in dimensions of $\mathbb{R}^{L\times(4\times M)}$.
- By concatenating the embeddings for the diffusion time step, inter-arrival time, and event type, we obtain an embedding in $\mathbb{R}^{L\times(4\times M)}$. We incorporate the positional encoding by summing the positional encoding to the concatenated embedding.
- This sequence of embeddings is then processed by a transformer block, facilitating cross-attention between the history embedding (in $R^{L\times(4\times M)}$) and the embedding of the forecasted event token sequences (in $\mathbb{R}^{L\times(4\times M)}$), yielding an output dimension of $4M$.
- Finally, a linear projection is applied to the inter-arrival time embedding to convert it into $\mathbb{R}$, and for the event type embedding, a linear projection converts it into $\mathbb{R}^K$, followed by a softmax function to get the logits of $\mathbf{e}_t$.

A.11. Sampling Details

In order to achieve a faster sampling time, we leverage the work of Song et al. (2021). We can re-express Eq.11 as follows

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\mathbf{x}_t, t, \mathbf{e}_t, \mathbf{s}_c)}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\epsilon_\theta(\mathbf{x}_t, t, \mathbf{e}_t, \mathbf{s}_c) + \sigma_t\mathbf{z} \tag{30}$$

Given a trained DDPM model, we can specify $\{\sigma_t\}_{t=1}^{\tau}$ and specify $\tau \subset \{1, 2, .., T\}$ to accomplish the acceleration. In Eq.30, if we set $\sigma_t = 0$ then we are performing DDIM (Denoising Diffusion Implicit Model) acceleration as in (Song et al.,

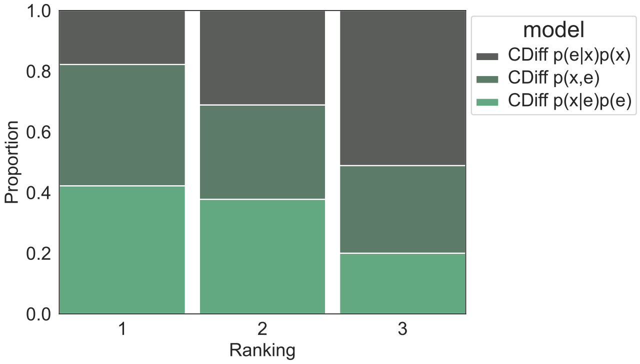


Figure 10. 100% Stacked column chart of ranks of different CDiff across the 5 datasets for all the metrics.

2021). For event type acceleration, we choose to directly jump steps, because for multinomial diffusion (Hoogeboom et al., 2021), instead of predicting noise, we predict $\mathbf{e}_0$. Therefore, our acceleration relies on decreasing the number of times we recalculate $\hat{\mathbf{e}}_0 = \phi_\theta(\mathbf{e}_t, \mathbf{x}_t, t, \mathbf{s}_c)$. That is, given a sub-set $\tau \subset \{1, 2, .., T\}$, we only recalculate $\hat{\mathbf{e}}_0$ $|\tau|$ times. In practice, we found it does not harm the prediction but it significantly accelerates the sampling due to $\phi_\theta(\cdot)$ requiring the majority of the computation effort.

A.12. Comparison with $p(\mathbf{x}, \mathbf{e})$ and $p(\mathbf{e}|\mathbf{x})p(\mathbf{x})$

Mathematically, $p(\mathbf{x}, \mathbf{e}) = p(\mathbf{e}|\mathbf{x})p(\mathbf{e}) = p(\mathbf{x}|\mathbf{e})p(\mathbf{e})$, so there should not be any theoretical difference between sampling the event type and interarrival time jointly or sampling one first and then the other, conditioned on the first. We conducted an experiment to check that this was also observed in the practical implementation. Fig.10 shows that the order of sampling does not have a major effect, although there is a minor advantage to either jointly sampling from $p(\mathbf{x}, \mathbf{e})$ or sampling the event type first (i.e., from $p(\mathbf{e}|\mathbf{x})p(\mathbf{e})$). This perhaps reflects that it is easier to learn the conditional inter-arrival time distributions, which may have slightly simpler structure.

A.13. Positional Encoding for CDiff

We use the transformer architecture as a denoising tool for reversing the diffusion processes. Therefore, we encode the position of both the diffusion step and the event token's order.

It is important that our choice of encoding can differentiate between these two different types of position information. To achieve this, we use as input $(i + y_N)$, where i is the order of the event token in the noisy event sequence, and y_N is the last timestamp of the historical event sequence.

into Eq. 22 (shown also below) for the order of the predicted sequence. This approach distinctly differentiates the positional information of the predicted event sequence from the diffusion time step's positional encoding. The positional encoding is then:

$$[\mathbf{m}(y_j, D)]_i = \begin{cases} \cos(y_j/10000^{\frac{i-1}{D}}) & \text{if } i \text{ is odd}, \\ \sin(y_j/10000^{\frac{i}{D}}) & \text{if } i \text{ is even}. \end{cases} \tag{31}$$

A.14. More Diffusion Visualization

Figure 11 shows the reverse process of CDiff for Taxi dataset (on the left) and Taobao dataset (on the right). Upon inspection, it is evident that the recovered sequences bear a strong resemblance to their respective ground truth sequences, both in terms of inter-arrival time patterns and event classifications.

In the Taxi dataset, the original sequences prominently feature events colored in Cyan and Orange. This indicates a high frequency of these two event categories, a pattern which is consistently replicated in the sequences derived from CDiff.

Conversely, for the Taobao dataset, the ground truth predominantly showcases shorter inter-arrival times, signifying closely clustered events. However, there are also occasional extended inter-arrival times introducing gaps in the sequences. Notably,

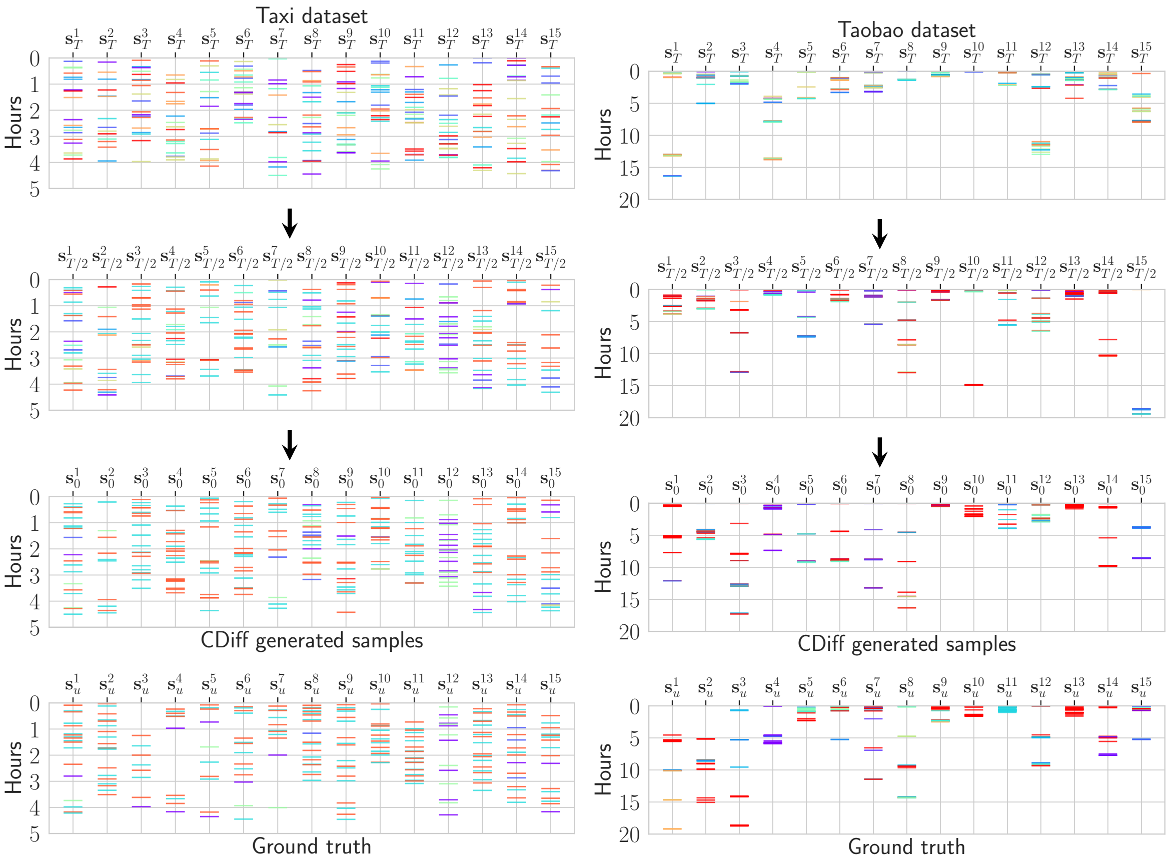


Figure 11. Visualization of the cross-diffusion generating process for 15 examples sequences of the Taxi dataset (left) and the Taobao dataset (right). The colors indicates the different categories. We start by generating noisy sequences ($t = T$). Once we reach the end of the denoising process ($t = 0$), we have recovered sequences similar to the ground truth sequences. We cut the sequence based on the time range so that every sequence can be aligned.

this dichotomy is accurately reflected in the reconstructed sequences.

A.15. Tables of results with different evaluation metrics for different horizon

Tables 11, 12 and 13 show the results of all metrics across all models for all datasets with different prediction horizons. We test for significance using a paired Wilcoxon signed-rank test at the 5% significance level.

Table 11. Results for all metrics across 7 different datasets for $N = 20$ **events forecasting** and **long interval forecasting**, bold case indicates the best, under line indicates the second best, * indicates stats. significance w.r.t. the method with the lowest value.

Synthetic dataset

	$N = 20$ **events forecasting**					**Interval forecasting t' long**			
	OTD	**RMSE$_e$**	**RMSE$_{x+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\|\mathbf{s}+\|}$**	**MAE$_{\|\mathbf{s}+\|}$**
HYPRO	20.609 ± 0.328	2.464 ± 0.039	0.104 ± 0.002*	717.417 ± 56.443	100.535 ± 0.084*	20.224 ± 0.236*	2.409 ± 0.082	1.608 ± 0.103	**0.573 ± 0.049**
Dual-TPP	22.117 ± 0.368*	2.506 ± 0.044*	0.108 ± 0.001*	724.681 ± 28.097*	100.857 ± 0.624*	21.521 ± 0.375*	2.511 ± 0.050*	2.297 ± 0.117*	0.952 ± 0.057*
Attnhp	21.843 ± 0.316*	2.509 ± 0.051*	0.104 ± 0.004*	682.086 ± 63.199	101.117 ± 0.295*	21.153 ± 0.206*	2.509 ± 0.048*	2.806 ± 0.073*	0.809 ± 0.033*
NHP	21.541 ± 0.203*	2.462 ± 0.018*	0.109 ± 0.001*	786.866 ± 31.782*	99.622 ± 0.426*	20.541 ± 0.203*	2.462 ± 0.021*	**1.411 ± 0.048**	0.588 ± 0.013*
LogNM	22.082 ± 0.225*	2.932 ± 0.028*	0.109 ± 0.005*	815.764 ± 32.480*	102.207 ± 0.472*	21.713 ± 0.198*	2.914 ± 0.019*	1.982 ± 0.078*	0.741 ± 0.054
TCDDM	21.270 ± 0.528	2.796 ± 0.027*	0.102 ± 0.002	700.630 ± 40.377	100.237 ± 0.275*	20.912 ± 0.310	2.735 ± 0.026	1.959 ± 0.03*	0.816 ± 0.011*
Homog. Poisson	22.595 ± 0.198*	2.946 ± 0.023*	0.129 ± 0.001*	1025.234 ± 139.141*	101.973 ± 0.380*	22.179 ± 0.298*	2.918 ± 0.037*	2.903 ± 0.065*	0.991 ± 0.067*
CDiff	**19.788 ± 0.343**	**2.375 ± 0.021**	**0.098 ± 0.02**	**668.287 ± 51.873**	**98.933 ± 0.573**	**19.674 ± 0.125**	**2.370 ± 0.061**	1.932 ± 0.094	0.812 ± 0.051*

Taxi dataset

	$N = 20$ **events forecasting**					**Interval forecasting t' long**			
	OTD	**RMSE$_e$**	**RMSE$_{x+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\|\mathbf{s}+\|}$**	**MAE$_{\|\mathbf{s}+\|}$**
HYPRO	21.653 ± 0.163	1.231 ± 0.015*	0.372 ± 0.004*	252.761 ± 6.827	93.803 ± 0.454*	19.632 ± 0.179	1.550 ± 0.026	4.326 ± 0.063*	2.781 ± 0.088*
Dual-TPP	24.483 ± 0.383*	1.353 ± 0.037*	0.402 ± 0.006*	285.590 ± 8.088*	95.211 ± 0.187*	20.952 ± 0.278*	1.627 ± 0.033*	4.995 ± 0.150*	3.795 ± 0.107*
Attnhp	24.762 ± 0.217*	1.276 ± 0.015*	0.430 ± 0.003*	286.869 ± 9.973*	97.388 ± 0.381*	20.588 ± 0.208*	1.590 ± 0.024	4.915 ± 0.116*	3.509 ± 0.112*
NHP	25.114 ± 0.268*	1.297 ± 0.019*	0.399 ± 0.040*	281.306 ± 8.271*	96.459 ± 0.521*	21.134 ± 0.148*	1.632 ± 0.030*	4.883 ± 0.119*	3.526 ± 0.135*
LogNM	24.053 ± 0.609*	1.364 ± 0.032*	0.384 ± 0.005*	282.173 ± 4.532*	95.719 ± 0.779*	20.422 ± 0.224	1.603 ± 0.033	5.072 ± 0.066*	3.796 ± 0.116*
TCDDM	22.148 ± 0.529	1.309 ± 0.030*	0.382 ± 0.019	259.944 ± 7.220	90.596 ± 0.574	20.191 ± 0.271	1.589 ± 0.064*	4.530 ± 0.118*	2.953 ± 0.237
Homog. Poisson	25.104 ± 0.083*	1.391 ± 0.032*	0.407 ± 0.002*	280.065 ± 7.541*	97.689 ± 0.613*	21.880 ± 0.175*	1.685 ± 0.019*	5.117 ± 0.151*	3.849 ± 0.105*
CDiff	**21.013 ± 0.158**	**1.131 ± 0.017**	**0.351 ± 0.004**	**243.2 ± 7.725**	**87.993 ± 0.178**	**19.028 ± 0.224**	**1.329 ± 0.029**	**3.690 ± 0.097**	**2.593 ± 0.124**

Taobao dataset

	$N = 20$ **events forecasting**					**Interval forecasting t' long**			
	OTD	**RMSE$_e$**	**RMSE$_{x+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\|\mathbf{s}+\|}$**	**MAE$_{\|\mathbf{s}+\|}$**
HYPRO	**44.336 ±0.127**	2.710 ±0.021*	0.594 ±0.030*	6397.66 ± 154.977	134.922 ± 0.473*	42.525 ± 0.151 *	**2.810 ± 0.028**	**4.022 ± 0.067**	3.019 ± 0.017*
Dual-TPP	47.324 ± 0.541*	3.237 ± 0.049*	0.871 ±0.014*	8325.564 ± 245.765*	141.687 ± 0.431*	38.530 ± 0.263*	4.439 ± 0.019*	5.893 ± 0.088*	3.832 ± 0.016*
Attnhp	45.555 ± 0.345*	2.737 ± 0.021	0.708 ± 0.011*	**6250.83 ± 265.440**	134.582 ± 0.920*	43.624 ± 0.282*	2.855 ± 0.020	4.097 ± 0.016	**2.892 ± 0.024**
NHP	48.131 ± 0.297*	3.355 ± 0.030*	0.837 ± 0.009*	7909.437 ± 149.274*	137.644 ± 0.764*	**38.204 ± 0.302**	3.515 ± 0.028*	5.41 ± 0.081*	3.998 ± 0.027*
LogNM	45.757 ± 0.287*	3.193 ± 0.043*	0.575 ± 0.012*	6558.437 ± 170.430	127.436 ± 0.606	39.769 ± 0.615	3.085 ± 0.076	4.914 ± 0.137*	3.814 ± 0.096*
TCDDM	45.563 ± 0.889*	2.850 ± 0.058	0.569 ± 0.015	6843.217 ± 278.296	126.512 ± 0.491	42.441 ± 0.434*	2.940 ± 0.094	4.231 ± 0.158	2.883 ± 0.057
Homog. Poisson	52.990 ± 0.234*	3.288 ± 0.022*	0.906 ± 0.012*	35474.601 ± 3495.078*	151.689 ± 0.615*	41.476 ± 0.811*	3.519 ± 0.036*	6.567 ± 0.083*	4.731 ± 0.075*
CDiff	44.621 ± 0.139	**2.653 ± 0.011**	**0.551 ± 0.013**	6850.359 ± 165.400	**125.685 ± 0.151**	40.783 ± 0.059*	2.831 ± 0.009	4.103 ± 0.034	2.947 ±0.019

Stackoverflow dataset

	$N = 20$ **events forecasting**					**Interval forecasting t' long**			
	OTD	**RMSE$_e$**	**RMSE$_{x+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\|\mathbf{s}+\|}$**	**MAE$_{\|\mathbf{s}+\|}$**
HYPRO	42.359 ± 0.170	1.140 ± 0.014	1.554 ± 0.010*	2013.055 ± 160.862*	110.988 ± 0.559*	38.460 ± 0.204	**1.294 ± 0.016**	2.672 ± 0.019*	1.496 ± 0.017
Dual-TPP	41.752 ± 0.200	**1.134 ± 0.019**	1.514 ± 0.017*	1729.83 ± 67.928*	117.582 ± 0.420*	38.474 ± 0.274	1.364 ± 0.019	3.332 ± 0.088*	1.753 ± 0.036*
Attnhp	42.591 ± 0.408*	1.145 ± 0.011	1.340 ± 0.006	**1519.740 ± 52.216**	108.542 ±0.531	39.76 ± 0.373*	1.385 ± 0.014*	3.424 ± 0.023*	1.813 ± 0.014*
NHP	43.791 ± 0.147*	1.244 ± 0.030*	1.487 ± 0.004*	1693.977 ± 113.300*	116.952 ± 0.404*	40.453 ± 0.188*	1.447 ± 0.012*	3.552 ± 0.051*	1.793 ± 0.057*
LogNM	46.280 ± 0.892*	1.447 ± 0.057*	1.669 ± 0.005*	2133.278 ± 163.516	115.122 ± 0.627*	42.594 ± 0.148*	1.507 ± 0.027	3.714 ± 0.078*	1.864 ± 0.076*
TCDDM	42.128 ± 0.591	1.467 ± 0.014*	1.315 ± 0.004	1762.121 ± 64.437	107.659 ± 0.934	38.697 ± 0.718	1.444 ± 0.019	2.623 ± 0.044	1.428 ± 0.070
Homog. Poisson	45.923 ± 0.286*	1.374 ± 0.022	1.359 ± 0.012*	2762.4786 ± 196.091*	116.447 ± 0.418*	43.288 ± 0.503*	1.539 ± 0.016*	3.459 ± 0.039*	1.778 ± 0.051*
CDiff	**41.245 ± 1.400**	1.141 ± 0.007	**1.199 ± 0.006**	1667.884 ± 32.220	**106.175 ± 0.340**	**37.659 ± 0.334**	1.421 ± 0.015*	**1.726 ± 0.043**	**1.239 ± 0.029**

Retweet dataset

	$N = 20$ **events forecasting**					**Interval forecasting t' long**			
	OTD	**RMSE$_e$**	**RMSE$_{x+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\|\mathbf{s}+\|}$**	**MAE$_{\|\mathbf{s}+\|}$**
HYPRO	61.031 ± 0.092*	2.623 ± 0.036*	30.100 ± 0.413*	19686.811 ± 966.339*	106.110 ± 1.505	59.292 ± 0.197	3.011 ± 0.029	3.109 ± 0.092	1.858 ± 0.067
Dual-TPP	61.095 ± 0.101*	2.679 ± 0.026*	28.914 ± 0.300	17619.400 ± 1003.001*	106.900 ± 1.293	**59.164 ± 0.069**	2.981 ± 0.041*	2.548 ± 0.133*	1.608 ± 0.028*
Attnhp	**60.634 ± 0.097**	2.561 ± 0.054	28.812 ± 0.272*	**15396.198 ± 1058.618**	107.234 ± 1.293*	59.302 ± 0.160	2.832 ± 0.057	2.736 ± 0.119	1.554 ± 0.084
NHP	60.953 ± 0.079	2.651 ± 0.045*	27.130 ± 0.224	15824.614 ± 1039.258	107.075 ± 1.398*	59.395 ± 0.098	2.780 ± 0.046	2.649 ± 0.104*	1.650 ± 0.044*
LogNM	61.715 ± 0.152*	2.776 ± 0.043*	27.582 ± 0.191	17914.114 ± 919.022	106.711 ± 1.615*	59.223 ± 0.247	2.815 ± 0.095	2.873 ± 0.118	1.847 ± 0.095*
TCDDM	60.501 ± 0.087	2.387 ± 0.050	27.303 ± 0.152	16070.5290 ± 540.227	**106.048 ± 0.610**	59.934 ± 0.122	2.762 ± 0.189	**2.131 ± 0.090**	1.129 ± 0.055
Homog. Poisson	61.224 ± 0.135*	3.179 ± 0.066*	35.125 ± 0.083	16800.047 ± 1793.164*	117.581 ± 0.500*	59.304 ± 0.194	2.920 ± 0.075*	3.076 ± 0.041*	1.901 ± 0.079*
CDiff	60.661 ± 0.101	**2.293 ± 0.034**	**27.101 ± 0.113**	16895.629 ± 741.331	106.184 ± 1.121	59.744 ± 0.574	**2.661 ± 0.030**	2.132 ± 0.131	**1.088 ± 0.031**

Mooc dataset

	$N = 20$ **events forecasting**					**Interval forecasting t' long**			
	OTD	**RMSE$_e$**	**RMSE$_{x+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\|\mathbf{s}+\|}$**	**MAE$_{\|\mathbf{s}+\|}$**
HYPRO	48.621 ± 0.352	1.169 ± 0.094	**0.410 ± 0.005**	12592.704 ± 235.279	**143.045 ± 7.992**	42.985 ± 0.113*	**1.037 ± 0.027**	5.769 ± 0.207	2.777 ± 0.119
Dual-TPP	50.184 ± 1.127	1.312 ± 0.019*	0.435 ± 0.006*	12511.299 ± 131.275	147.003 ± 2.908*	41.295 ± 0.074	1.272 ± 0.016*	6.121 ± 0.159*	3.255 ± 0.051
AttNHP	49.121 ± 0.720*	1.297 ± 0.049	0.420 ± 0.009	12838.668 ± 296.147	147.756 ± 4.812	43.001 ± 0.111*	1.038 ± 0.025	5.591 ± 0.083	2.597 ± 0.076
NHP	51.277 ± 1.768*	1.458 ± 0.063*	0.442 ± 0.007*	13082.583 ± 352.970	148.913 ± 11.628*	**40.933 ± 0.204**	1.298 ± 0.016*	6.160 ± 0.080	3.337 ± 0.047*
LogNM	52.890 ± 1.151*	1.428 ± 0.061*	0.454 ± 0.008*	14868.891 ± 315.812	149.987 ± 16.581*	41.003 ± 0.127	1.307 ± 0.039*	5.895 ± 0.057*	2.838 ± 0.063
TCDDM	50.739 ± 0.765*	1.407 ± 0.112	0.429 ± 0.015	12409.522 ± 267.312	145.745 ± 11.835	42.662 ± 0.200	1.199 ± 0.057*	5.634 ± 0.094	2.663 ± 0.091
Homog. Poisson	58.568 ± 0.147*	1.161 ± 0.004	0.536 ± 0.002*	235478.446 ± 353.632	175.587 ± 10.333*	43.442 ± 0.716*	1.088 ± 0.037*	6.943 ± 0.155*	3.741 ± 0.112*
CDiff	**47.214 ± 0.628**	**1.095 ± 0.048**	0.411 ± 0.009	**12243.367 ± 188.453**	146.361 ± 14.837*	42.118 ± 0.171*	1.041 ± 0.021	**5.584 ± 0.186**	**2.566 ± 0.092**

Amazon dataset

	$N = 20$ **events forecasting**					**Interval forecasting t' long**			
	OTD	**RMSE$_e$**	**RMSE$_{x+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\|\mathbf{s}+\|}$**	**MAE$_{\|\mathbf{s}+\|}$**
HYPRO	38.613 ± 0.536*	**2.007 ± 0.054**	0.477 ± 0.010*	1247.592 ± 96.544	82.506 ± 0.840	38.229 ± 0.052	**1.995 ± 0.005**	0.986 ± 0.011	**0.414 ± 0.004**
Dual-TPP	42.646 ± 0.752*	2.562 ± 0.202	0.482 ± 0.012*	1414.225 ± 70.306	86.453 ± 2.044	40.987 ± 0.490*	2.410 ± 0.034*	1.269 ± 0.011*	0.617 ± 0.003*
AttNHP	39.480 ± 0.326	2.166 ± 0.026*	0.476 ± 0.033	1372.409 ± 53.202	84.323 ± 1.815*	39.870 ± 0.641	2.042 ± 0.031	0.998 ± 0.008	0.417 ± 0.005
NHP	42.571 ± 0.293*	2.561 ± 0.060	0.519 ± 0.023*	1426.601 ± 16.437*	92.053 ± 1.553*	41.110 ± 0.272*	2.447 ± 0.053*	1.278 ± 0.005*	0.603 ± 0.005*
LNM	43.820 ± 0.232*	3.050 ± 0.286*	0.481 ± 0.145*	1523.064 ± 312.396*	90.910 ± 1.611*	41.953 ± 0.395	2.872 ± 0.015*	1.268 ± 0.007*	0.614 ± 0.009*
TCDDM	42.245 ± 0.174*	2.998 ± 0.115*	0.476 ± 0.111	**1086.146 ± 94.188**	83.826 ± 1.508	40.432 ± 0.307	2.797 ± 0.048	0.996 ± 0.004*	0.429 ± 0.003*
Homog. Poisson	43.940 ± 0.360*	4.870 ± 0.019*	0.691 ± 0.004*	1775.151 ± 37.202*	112.392 ± 0.464*	42.713 ± 0.474*	3.526 ± 0.037*	1.524 ± 0.009*	0.934 ± 0.006*
CDiff	**37.728 ± 0.199**	2.091 ± 0.163	**0.464 ± 0.086**	1189.691 ± 71.215	**81.987 ± 1.905**	**37.068 ± 0.038**	2.058 ± 0.009	**0.961 ± 0.018**	0.416 ± 0.006

Table 12. Results for all metrics across 7 different datasets for $N = 10$ **events forecasting** and **medium interval forecasting**, bold case indicates the best, under line indicates the second best, * indicates stats. significance w.r.t. the method with the lowest value

Synthetic dataset

	$N = 10$ **events forecasting**					**Interval forecasting t' medium**			
	OTD	**RMSE$_e$**	**RMSE$_{x^+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\lvert\mathbf{s}^+\rvert}$**	**MAE$_{\lvert\mathbf{s}^+\rvert}$**
HYPRO	**12.962 ± 0.128**	**1.747 ± 0.041**	0.104 ± 0.006	612.354 ± 21.017	99.473 ± 0.767	**13.263 ± 0.213**	**1.721 ± 0.011**	**1.404 ± 0.023**	**0.561 ± 0.029**
Dual-TPP	14.141 ± 0.125*	1.965 ± 0.053*	0.108 ± 0.008*	713.157 ± 30.615*	99.688 ± 0.672*	13.919 ±0.271*	1.777 ± 0.019*	2.109 ± 0.048*	0.667 ± 0.033*
Attnhp	13.916 ± 0.110	1.851 ± 0.039*	0.103 ± 0.009	587.161 ± 41.113	100.041 ± 0.551*	13.654 ± 0.163	1.799 ± 0.018*	1.517 ± 0.045*	0.741 ± 0.019*
NHP	13.588 ± 0.313	1.801 ± 0.016*	0.107 ± 0.005*	712.673 ± 66.121*	99.343 ± 0.721*	13.551 ± 0.197*	1.801 ± 0.030	1.408 ± 0.051	0.590 ± 0.027
LogNM	13.969 ± 0.266*	1.915 ± 0.029*	0.105 ± 0.004	667.876 ± 58.456*	99.552 ± 0.901	13.784 ± 0.192*	1.779 ± 0.029	1.493 ± 0.043*	0.571 ± 0.033
TCDDM	13.503 ± 0.160	1.863 ± 0.037	0.105 ± 0.001	632.431 ± 42.223	99.267 ± 0.576	13.559 ± 0.177*	1.761 ± 0.032	1.485 ± 0.025*	0.631 ± 0.011*
Homog. Poisson	15.532 ± 0.197*	2.057 ± 0.018*	0.143 ± 0.005*	1014.814 ± 72.140*	101.156 ± 0.601*	15.240 ± 0.232*	1.923 ± 0.087*	1.740 ± 0.049*	1.041 ± 0.019*
CDiff	13.792 ± 0.251	1.786 ± 0.019	**0.096 ± 0.005**	**419.982 ± 52.083**	**99.063 ± 0.523**	13.371 ± 0.572	1.773 ± 0.017*	1.473 ± 0.035*	0.632 ± 0.015*

Taxi dataset

	$N = 10$ **events forecasting**					**Interval forecasting t' medium**			
	OTD	**RMSE$_e$**	**RMSE$_{x^+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\lvert\mathbf{s}^+\rvert}$**	**MAE$_{\lvert\mathbf{s}^+\rvert}$**
HYPRO	11.875 ± 0.172	**0.764 ± 0.008**	0.363 ± 0.002	261.896 ± 33.712	89.524 ± 0.552	10.184 ± 0.191	**0.906 ± 0.019**	2.976 ± 0.093	2.216 ± 0.061
Dual-TPP	13.058 ± 0.220*	0.966 ± 0.011*	0.395 ± 0.003*	268.407 ± 41.313*	90.812 ± 0.497*	11.031 ± 0.227*	1.044 ± 0.027*	3.478 ± 0.147*	2.547 ± 0.127*
Attnhp	12.542 ± 0.336	0.823 ± 0.007	0.376 ± 0.003*	253.040 ± 37.710	92.812 ± 0.129	10.339 ± 0.194*	0.929 ± 0.031	3.249 ± 0.099*	2.341 ± 0.147*
NHP	13.377 ± 0.184*	0.922 ± 0.009*	0.397 ± 0.005*	269.204 ± 28.418*	92.182 ± 0.384*	11.115 ± 0.209*	1.044 ± 0.017*	3.523 ± 0.102*	2.548 ± 0.121*
LogNM	12.765 ± 0.106*	1.004 ± 0.013*	0.383 ± 0.015*	263.311 ± 26.418	93.120 ± 0.526*	10.527 ± 0.140*	0.958 ± 0.033*	3.398 ± 0.158*	2.431 ± 0.106*
TCDDM	11.885 ± 0.149	1.121 ± 0.072*	0.385 ± 0.009*	254.312 ± 33.659	90.703 ± 0.356	10.209 ± 0.337*	0.998 ± 0.035*	3.441 ± 0.201*	2.339 ± 0.154
Homog. Poisson	14.209 ± 0.097*	1.402 ± 0.033*	0.397 ± 0.004*	279.410 ± 19.417*	96.350± 0.513*	11.059 ± 0.172 *	1.112 ± 0.0315*	4.065 ± 0.197*	2.994 ± 0.251*
CDiff	**11.004 ± 0.191**	0.785 ± 0.007	**0.350 ± 0.002**	**236.572 ± 35.459**	**90.721 ± 0.291**	**9.335 ± 0.211**	0.926 ± 0.023	**2.972 ± 0.111**	**2.117 ± 0.090**

Taobao dataset

	$N = 10$ **events forecasting**					**Interval forecasting t' medium**			
	OTD	**RMSE$_e$**	**RMSE$_{x^+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\lvert\mathbf{s}^+\rvert}$**	**MAE$_{\lvert\mathbf{s}^+\rvert}$**
HYPRO	21.547 ± 0.138*	1.527 ± 0.035*	0.591 ± 0.019	**5968.317 ± 240.664**	133.147 ± 0.341	20.101 ± 0.127*	1.671 ± 0.012*	2.403 ± 0.042	1.391 ± 0.023
Dual-TPP	23.691 ± 0.203*	2.674 ± 0.032*	0.873 ± 0.010*	8413.261 ± 222.427*	139.271 ± 0.348*	**18.817 ± 0.215**	1.738 ± 0.010*	4.207 ± 0.076*	2.352 ± 0.021*
Attnhp	21.683 ± 0.215	1.514 ± 0.015*	0.608 ± 0.011*	6034.771 ± 170.267	135.271 ± 0.395	20.653 ± 0.162*	**1.342 ± 0.009**	**2.221 ± 0.045**	1.297 ± 0.011
NHP	24.068 ± 0.331*	2.769 ± 0.033*	0.855 ± 0.013*	7734.518 ± 276.670*	137.693 ± 0.225*	18.991 ± 0.278*	1.862 ± 0.014*	3.995 ± 0.077*	2.437 ± 0.017*
LogNM	23.195 ± 0.039*	2.429 ± 0.045*	0.602 ± 0.037*	6719.015 ± 163.868	127.411 ± 0.573	19.383 ± 0.402*	1.826 ± 0.005*	3.634 ± 0.058*	1.745 ± 0.014*
TCDDM	**21.012 ± 0.520**	2.598 ± 0.047	0.610 ± 0.022*	6630.487 ± 259.540	132.7112 ± 0.774	20.032 ± 0.691*	1.558 ± 0.015*	2.951 ± 0.069*	1.649 ± 0.0183
Homog. Poisson	27.353 ± 0.426*	2.772 ± 0.016*	0.887 ± 0.014*	19301.747 ± 349.301*	155.236 ± 0.729*	19.251 ± 0.221*	1.920 ± 0.009*	5.001 ±0.022 *	3.209 ± 0.011*
CDiff	21.221 ± 0.176	**1.416 ± 0.024**	**0.535 ± 0.016**	6718.144 ± 161.416	**126.824 ± 0.366**	19.677 ± 0.103*	1.438 ± 0.012	2.307 ± 0.059	**1.160 ± 0.019**

Stackoverflow dataset

	$N = 10$ **events forecasting**					**Interval forecasting t' medium**			
	OTD	**RMSE$_e$**	**RMSE$_{x^+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\lvert\mathbf{s}^+\rvert}$**	**MAE$_{\lvert\mathbf{s}^+\rvert}$**
HYPRO	21.062 ± 0.372	0.921 ± 0.019	1.235 ± 0.006	1925.362 ± 149.208*	107.566 ± 0.218*	18.523 ± 0.301	0.907 ± 0.013	2.327 ± 0.040	1.339 ± 0.033
Dual-TPP	21.229 ± 0.394*	0.936 ± 0.013	1.223 ± 0.010*	1845.469 ± 103.450*	107.274 ± 0.200*	19.155 ± 0.116*	0.923 ± 0.011*	2.344 ± 0.053*	1.478 ± 0.038*
Attnhp	22.019 ± 0.220*	0.978 ± 0.023	1.225 ± 0.007*	**1571.807 ± 99.921**	**100.137 ± 0.167**	19.487 ± 0.130*	0.973 ± 0.013*	2.415 ± 0.026*	1.455 ± 0.025*
NHP	21.655 ± 0.314*	0.970 ± 0.014*	1.266 ± 0.003*	1698.947 ± 123.208	108.867 ± 0.361*	19.314 ± 0.098*	0.959 ± 0.017*	2.481 ± 0.035*	1.419 ± 0.031*
LogNM	22.339 ± 0.322*	0.970 ± 0.011	1.251 ± 0.005	1841.119 ± 71.077*	105.674 ± 0.337	19.303 ± 0.137	0.955 ± 0.014	2.751 ± 0.028*	1.487 ± 0.046*
TCDDM	22.042 ± 0.193*	1.205 ± 0.014	1.228 ± 0.010*	1772.325 ± 221.358*	108.1113 ± 0.112*	18.920 ± 0.125	0.930 ± 0.015	2.472 ± 0.033	1.293 ± 0.050
Homog. Poisson	23.115 ± 0.318*	1.012 ± 0.027	1.327 ± 0.004*	2105.433 ± 88.409*	108.322 ± 0.315*	22.714 ± 0.300*	0.973 ± 0.023*	2.889 ± 0.020*	1.597 ± 0.021*
CDiff	**20.191 ± 0.455**	**0.916 ± 0.010**	**1.180 ± 0.003**	1880.59 ± 78.283	102.367 ± 0.267*	**18.268 ± 0.167**	**0.883 ± 0.009**	**2.107 ± 0.031**	**1.219 ± 0.023**

Retweet dataset

	$N = 10$ **events forecasting**					**Interval forecasting t' medium**			
	OTD	**RMSE$_e$**	**RMSE$_{x^+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\lvert\mathbf{s}^+\rvert}$**	**MAE$_{\lvert\mathbf{s}^+\rvert}$**
HYPRO	31.743 ± 0.068*	1.927 ± 0.027*	33.683 ± 0.245*	17696.498 ± 986.684*	**105.073 ± 0.958**	27.411 ± 0.190	2.013 ± 0.032*	2.741 ± 0.108*	1.971 ± 0.031*
Dual-TPP	31.652 ± 0.075*	1.963 ± 0.038*	28.104 ± 0.486*	17553.619 ± 731.120*	106.721 ± 0.774*	28.357 ± 0.176*	1.991 ± 0.050*	1.963 ± 0.094	1.615 ± 0.037
Attnhp	**30.337 ± 0.065**	1.823 ± 0.031*	**26.310 ± 0.333**	**14377.241± 1319.797**	106.021 ± 1.011	**26.787 ± 0.114**	**1.961 ± 0.029**	1.981 ± 0.115*	1.597 ± 0.058*
NHP	30.817 ± 0.090	**1.713 ± 0.024**	27.010 ± 0.429*	15214.175± 695.184*	107.053 ± 1.390*	27.617 ± 0.099*	1.997 ± 0.047*	1.959 ± 0.124*	1.562 ± 0.080*
LogNM	31.974 ± 0.032*	1.942 ± 0.062*	28.825 ± 0.221	17339.802 ± 765.475*	106.014 ± 0.633	27.283 ± 0.078*	1.995 ± 0.026*	2.327 ± 0.126	1.649 ± 0.069
TCDDM	32.006 ± 0.074	1.789 ± 0.094	29.124 ± 0.405	18874.939 ±828.544	106.738 ± 0.791	27.993 ± 0.230	2.035 ± 0.047*	1.997 ± 0.215	1.337 ± 0.080
Homog. Poisson	30.885 ± 0.017	1.987 ± 0.036*	33.241 ± 0.512*	17892.301 ± 355.213*	114.286 ± 0.753*	26.950 ± 0.306	1.987 ± 0.026*	2.774 ± 0.118*	2.023 ± 0.0355*
CDiff	31.237 ± 0.078*	1.745 ± 0.036	26.429 ± 0.201	15636.184 ± 713.516	105.767 ± 0.771	27.739 ± 0.105	1.973 ± 0.036	**1.907 ± 0.111**	**1.299 ± 0.043**

Mooc dataset

	$N = 10$ **events forecasting**					**Interval forecasting t' medium**			
	OTD	**RMSE$_e$**	**RMSE$_{x^+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\lvert\mathbf{s}^+\rvert}$**	**MAE$_{\lvert\mathbf{s}^+\rvert}$**
HYPRO	25.861 ± 0.352	1.032 ± 0.073	**0.391 ± 0.002**	**11931.797 ± 254.663**	**142.041 ± 5.730**	22.640 ± 0.171*	**0.921 ± 0.063**	**4.956 ± 0.277**	**2.214 ± 0.057**
Dual-TPP	28.785 ± 0.384*	1.087 ± 0.012*	0.421 ± 0.006*	12721.909 ± 126.31	146.841 ± 4.188*	22.359 ± 0.083*	1.028 ± 0.009	5.573 ± 0.173*	2.931 ± 0.029*
AttNHP	26.765 ± 0.221*	1.054 ± 0.009	0.421 ± 0.011	13138.381 ± 372.632*	144.641 ± 3.093*	23.185 ± 0.071*	0.958 ± 0.044	5.105 ± 0.040	2.491± 0.050
NHP	27.371 ± 0.632*	1.134 ± 0.064	0.429 ± 0.007*	13275.513 ± 262.612*	143.526± 9.509*	**21.275 ± 0.051**	1.038 ± 0.026*	5.349 ± 0.077*	3.163 ± 0.043*
LogNM	29.497 ± 0.325*	1.120 ± 0.037*	0.433 ± 0.013*	12692.049 ± 255.629	144.093 ± 5.077*	21.727 ± 0.183	1.121 ± 0.018*	5.297 ± 0.029	3.099 ± 0.060*
TCDDM	**24.515 ± 0.339**	1.218 ± 0.065*	0.425 ± 0.019	11958.023 ± 267.593	143.293 ± 12.089	23.020 ± 0.145*	1.126 ± 0.052*	5.224 ± 0.075*	2.476 ± 0.049
Homog. Poisson	33.349 ± 0.143*	1.269 ± 0.006*	0.443 ± 0.016*	232853.735 ± 71.130*	168.305 ± 3.126*	21.950 ± 0.043	1.183 ± 0.017	6.036 ± 0.261*	3.330 ± 0.026 *
CDiff	24.544 ± 0.305	**0.944 ± 0.032**	0.404 ± 0.003	12052.014 ± 213.141	144.313 ± 8.726*	22.768 ± 0.125*	0.935 ± 0.074	5.120 ± 0.116*	2.439 ± 0.034

Amazon dataset

	$N = 10$ **events forecasting**					**Interval forecasting t' medium**			
	OTD	**RMSE$_e$**	**RMSE$_{x^+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\lvert\mathbf{s}^+\rvert}$**	**MAE$_{\lvert\mathbf{s}^+\rvert}$**
HYPRO	24.956 ± 0.663	1.765 ± 0.039	**0.442 ± 0.015**	1211.590 ± 62.458	83.401 ± 1.033	24.096 ± 0.043*	1.678 ± 0.024	0.987 ± 0.009	**0.408 ± 0.010**
Dual-TPP	25.929 ± 0.280*	2.098 ± 0.101*	0.475 ± 0.008*	1376.448 ± 104.345*	82.352 ± 1.285	23.688 ± 0.411*	2.208 ± 0.094*	1.162 ± 0.031*	0.612 ± 0.009*
AttNHP	**24.116 ± 0.807**	**1.741 ± 0.039**	0.454 ± 0.014	1323.165 ± 62.289	84.323 ± 1.815	24.278 ± 0.218*	1.693± 0.067	0.998 ± 0.005	0.431 ± 0.010
NHP	25.730 ± 0.497*	1.843 ± 0.053*	0.491 ± 0.048*	1426.601 ± 16.437*	89.135 ± 1.092*	**22.506 ± 0.141**	1.884 ± 0.092*	1.218 ± 0.006	0.566 ± 0.010*
LNM	26.632 ± 0.519*	1.955 ± 0.112*	0.464 ± 0.066*	1555.852 ± 33.930*	89.305 ± 1.288*	23.049 ± 0.412	2.658 ± 0.030*	1.117 ± 0.009*	0.513 ± 0.008*
TCDDM	25.091 ± 0.227*	1.778 ± 0.090	0.448 ± 0.082	1274.340 ± 92.095	**82.105 ± 1.564**	24.007 ± 0.109*	2.103 ± 0.043*	**0.980 ± 0.004**	0.430 ± 0.011
Homog. Poisson	28.945 ± 0.441*	3.076 ± 0.021*	0.700 ± 0.009*	2103.582 ± 38.491*	109.143 ± 0.304*	23.745 ± 0.738	1.988 ± 0.057*	1.423 ± 0.005*	0.847 ± 0.003*
CDiff	24.230 ± 0.287	1.766± 0.079	0.450 ± 0.049	**1146.530 ± 43.595**	82.124 ± 2.094	23.994 ± 0.113*	**1.503 ± 0.034**	1.005 ± 0.010	0.409 ± 0.005

Table 13. Results for all metrics across 7 different datasets for $N = 5$ **events forecasting** and **small interval forecasting**, bold case indicates the best, under line indicates the second best, * indicates stats. significance w.r.t. the method with the lowest value

Synthetic dataset

	$N = 5$ events forecasting					**Interval forecasting t' small**			
	OTD	**RMSE$_e$**	**RMSE$_{x^+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\|\mathbf{s}^+\|}$**	**MAE$_{\|\mathbf{s}^+\|}$**
HYPRO	8.706 ± 0.138	1.216 ± 0.023	0.091 ± 0.003	510.171 ± 23.802*	98.857 ± 0.185	8.230 ± 0.210	1.184 ± 0.051*	1.281 ± 0.079*	0.724 ± 0.048*
Dual-TPP	8.644 ± 0.102*	1.280 ± 0.011*	0.093 ± 0.001*	453.129 ± 27.592*	98.683 ± 0.351	8.248 ± 0.235*	1.177 ± 0.047	1.161 ± 0.093*	0.560 ± 0.018*
Attnhp	8.687 ± 0.149	1.225 ± 0.031	0.089 ± 0.003	**415.593 ± 24.153**	100.762 ± 0.020	8.342 ± 0.078	1.192 ± 0.040*	1.131 ± 0.059	0.528 ± 0.034
NHP	8.565 ± 0.098	1.207 ± 0.017	0.094 ± 0.002*	431.286 ± 30.272	100.861 ± 0.183*	8.128 ± 0.274	**1.171 ± 0.053**	1.217 ± 0.073*	0.608 ±0.023*
LogNM	10.093 ± 0.145*	1.390 ± 0.019*	0.093 ± 0.005*	482.341 ± 29.601*	101.984 ± 0.147*	8.449 ± 0.093	1.244 ± 0.101*	1.239 ± 0.028*	0.552 ± 0.011
TCDDM	8.881 ± 0.112*	1.295 ± 0.008*	0.095 ± 0.001	472.54 ± 33.634	99.008 ± 0.251	8.593 ± 0.185	1.227 ± 0.061	1.221 ± 0.058*	0.524 ± 0.023
Homog. Poisson	10.23 ± 0.135*	1.268 ± 0.015*	0.101 ± 0.005*	486.35 ± 20.561*	101.357 ± 0.301*	10.587 ± 0.227*	1.265 ± 0.114*	1.475 ± 0.062*	0.679 ± 0.027*
CDiff	**8.459 ± 0.167**	**1.196 ± 0.015**	**0.088 ±0.002**	473.506 ± 15.600	**98.011 ± 0.197**	**8.095 ± 0.176**	1.175 ± 0.059	**1.068 ± 0.035**	**0.517 ± 0.039**

Taxi dataset

	$N = 5$ events forecasting					**Interval forecasting t' small**			
	OTD	**RMSE$_e$**	**RMSE$_{x^+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\|\mathbf{s}^+\|}$**	**MAE$_{\|\mathbf{s}^+\|}$**
HYPRO	5.952 ± 0.126	**0.500 ± 0.011**	0.322 ± 0.004	**221.745 ± 5.084**	**85.994 ± 0.227**	**4.780 ± 0.214**	**0.518 ± 0.010**	1.893 ± 0.052	1.405 ± 0.108
Dual-TPP	7.534 ± 0.111*	0.636 ± 0.009*	0.340 ± 0.003	252.822 ± 3.853*	89.727 ±0.320	6.225 ± 0.117*	0.647 ± 0.029*	1.910 ± 0.043*	1.417 ± 0.081*
Attnhp	6.441 ± 0.090	0.682 ± 0.010	0.347 ±0.002	259.480 ± 4.819*	89.070 ± 0.152	6.201 ± 0.111	0.642 ± 0.024	1.923 ± 0.062*	**1.362 ± 0.095**
NHP	7.405 ± 0.122*	0.641 ± 0.013*	0.351 ± 0.008*	231.504 ± 6.054*	91.625 ± 0.177*	6.244 ± 0.172*	0.653 ± 0.019*	1.927 ± 0.038*	1.387 ± 0.117*
LogNM	7.209 ± 0.184*	0.608 ± 0.008	0.335 ± 0.003	255.600 ± 4.601*	90.512 ± 0.169	6.664 ± 0.143*	0.721 ± 0.013*	1.897 ± 0.044*	1.401 ± 0.079
TCDDM	**5.877 ± 0.095**	0.648 ± 0.015*	0.327 ± 0.005	246.121 ± 5.512	88.051 ± 0.240	5.792 ± 0.110	0.683 ± 0.024	1.910 ± 0.037*	1.395 ± 0.100
Homog. Poisson	6.905 ± 0.094*	0.692 ± 0.007*	0.393 ± 0.006*	272.51 ± 3.049*	94.501 ± 0.192*	6.520 ± 0.133*	0.797 ± 0.019*	2.057 ± 0.012*	1.584 ± 0.078*
CDiff	5.966 ± 0.083	0.547 ± 0.007	**0.318 ± 0.003**	223.073 ± 6.221	89.535 ± 0.294	5.128 ± 0.148	0.603 ± 0.025	**1.889 ± 0.019**	1.363 ± 0.074

Taobao dataset

	$N = 5$ events forecasting					**Interval forecasting t' small**			
	OTD	**RMSE$_e$**	**RMSE$_{x^+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\|\mathbf{s}^+\|}$**	**MAE$_{\|\mathbf{s}^+\|}$**
HYPRO	11.317 ± 0.111	0.817± 0.037	0.573 ± 0.011*	4652.619 ±189.940	133.837 ± 0.524	11.546 ± 0.124*	0.866± 0.016	1.402± 0.062	0.654± 0.011
Dual-TPP	13.280 ± 0.092*	1.877±0.014*	0.691± 0.007*	6828.105 ± 235.303*	134.437± 0.458*	9.779± 0.194*	1.655± 0.028*	3.474± 0.037*	1.966±0.018*
Attnhp	11.223 ± 0.145	0.873± 0.023	0.550± 0.014	**4231.499± 155.699**	132.266± 0.532	11.498± 0.175*	**0.858± 0.020**	**1.312± 0.034**	**0.566± 0.024**
NHP	11.973 ± 0.176*	1.910 ± 0.031*	0.712± 0.017*	5961.627± 183.108*	134.693± 0.369*	**8.748± 0.294**	1.718± 0.035*	3.297± 0.051*	2.001± 0.015*
LogNM	11.052 ± 0.108*	1.941 ± 0.049*	0.601 ± 0.017	5006.301 ± 287.390	126.32 ± 0.591	10.395 ± 0.201*	1.304 ± 0.040*	1.932 ± 0.027*	0.994 ± 0.008
TCDDM	11.609± 0.184	1.690 ± 0.023*	0.675 ± 0.009	5042.501 ± 324.55*	129.009± 0.923*	11.203 ± 0.192*	1.209 ± 0.068*	2.003 ± 0.033	1.024 ± 0.020
Homog. Poisson	13.510 ± 0.203*	1.392 ± 0.034*	1.093 ± 0.047	5039.401 ± 442.580*	143.105 ± 0.699*	9.300 ± 0.225	1.527 ± 0.079*	3.342 ± 0.042*	2.401 ± 0.0028*
CDiff	**10.147 ± 0.140**	**0.730 ± 0.019**	**0.519 ± 0.008**	4736.039 ± 114.586	**124.339 ± 0.322**	9.122 ± 0.179	0.861 ± 0.022	1.628± 0.033	0.730 ± 0.013

Stackoverflow dataset

	$N = 5$ events forecasting					**Interval forecasting t' small**			
	OTD	**RMSE$_e$**	**RMSE$_{x^+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\|\mathbf{s}^+\|}$**	**MAE$_{\|\mathbf{s}^+\|}$**
HYPRO	11.590± 0.186	0.586 ± 0.019	1.227± 0.018	1413.759± 79.723	109.014± 0.422	9.677± 0.117	**0.530 ± 0.021**	1.689 ± 0.017*	1.007± 0.030
Dual-TPP	11.719 ± 0.109*	0.591 ± 0.026*	1.296 ± 0.010*	1319.909 ± 121.366	106.697 ± 0.381	9.963 ± 0.230*	0.563 ± 0.023	1.572 ± 0.036	0.987 ± 0.042
Attnhp	11.595 ± 0.197	0.575 ± 0.009	1.188± 0.014	1418.384± 48.412	105.799± 0.516	9.787± 0.321	0.552± 0.018	1.559 ± 0.031*	**0.963 ± 0.025**
NHP	11.807± 0.155*	0.596 ± 0.015*	1.261±0.013*	**1292.252± 133.873**	108.074± 0.661*	10.809± 0.182*	0.570± 0.026*	1.716± 0.037*	1.033± 0.027*
LogNM	13.124 ± 0.174*	0.702 ± 0.008*	1.182 ± 0.039	1335.23 ± 145.031	108.409 ± 0.692	11.015 ± 0.191*	0.629 ± 0.093*	1.664 ± 0.042*	1.032 ± 0.018*
TCDDM	11.41 ± 0.129	0.630 ± 0.015*	1.201 ± 0.028	1412.195 ± 135.312	107.893 ± 0.942	10.23 ± 0.096*	0.611 ± 0.024*	**1.532 ± 0.06**	1.021 ± 0.010*
Homog. Poisson	15.493 ± 0.144*	0.693 ± 0.013*	1.336 ± 0.059*	2034.235 ± 125.314	108.900 ± 0.074*	13.12 ± 0.073*	0.921 ± 0.045*	1.886± 0.008*	1.120 ± 0.039*
CDiff	**10.735 ± 0.183**	**0.571 ± 0.012**	**1.153 ± 0.011**	1386.314 ± 57.750	**100.586 ± 0.299**	**8.849 ± 0.187**	0.545 ± 0.015	1.564 ± 0.029*	0.991± 0.035

Retweet dataset

	$N = 5$ events forecasting					**Interval forecasting t' small**			
	OTD	**RMSE$_e$**	**RMSE$_{x^+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\|\mathbf{s}^+\|}$**	**MAE$_{\|\mathbf{s}^+\|}$**
HYPRO	16.145± 0.096	1.105± 0.026	27.236± 0.259	22428.809 ± 780.393	103.052± 1.206	**13.199 ± 0.089**	1.201± 0.053	1.602± 0.096*	1.103± 0.075*
Dual-TPP	16.050 ± 0.085	1.077± 0.027*	31.493±0.162*	**15403.772± 831.413**	**101.322± 1.127**	13.809± 0.048	1.197 ± 0.025*	1.478±0.082*	0.980 ± 0.038*
Attnhp	16.124 ± 0.089	1.058 ± 0.029	29.247 ± 0.145	18377.481 ± 878.880	105.93 ± 1.380	14.120 ± 0.127*	1.144 ± 0.034	1.315 ± 0.070	0.862 ± 0.051
NHP	15.945 ± 0.094	1.113 ± 0.040*	32.367 ± 0.104*	22611.646 ± 797.268*	107.022 ± 1.077*	14.201 ± 0.119*	1.161 ± 0.023*	1.369 ± 0.102*	0.894 ± 0.025*
LogNM	16.043 ± 0.222	1.313 ± 0.011*	30.853 ± 0.119	23084.93± 784.430	106.941 ± 2.031	13.937 ± 0.239	1.208 ± 0.029*	1.590 ± 0.113	0.874 ± 0.068
TCDDM	15.874 ± 0.053	1.194 ± 0.021*	28.530 ± 0.110	19093.229 ± 880.932	105.570 ± 0.94	14.771 ± 0.298*	1.340 ± 0.030*	1.275 ± 0.084	0.798 ± 0.028
Homog. Poisson	19.432 ± 0.033*	1.405 ± 0.008*	30.543 ± 0.083*	28094.854 ± 684.501*	108.591 ± 1.049*	15.039 ± 0.591*	1.347 ± 0.094*	1.898 ± 0.020*	1.091 ± 0.044*
CDiff	**15.858 ± 0.080**	**1.023± 0.036**	**26.078± 0.175**	21778.765± 689.206	106.62± 1.008	14.073 ± 0.065*	**1.127 ± 0.029**	**1.123 ±0.099**	**0.782 ± 0.063**

Mooc dataset

	$N = 5$ events forecasting					**Interval forecasting t' small**			
	OTD	**RMSE$_e$**	**RMSE$_{x^+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\|\mathbf{s}^+\|}$**	**MAE$_{\|\mathbf{s}^+\|}$**
HYPRO	11.718 ± 0.240	0.811 ± 0.045	0.308 ± 0.015	12949.391 ± 441.590	**142.735 ± 17.901**	10.657 ± 0.092	0.692 ± 0.018	1.772 ± 0.034	**0.890 ± 0.056**
Dual-TPP	14.503 ± 0.334*	0.950 ± 0.027	0.412 ± 0.006*	14492.350 ± 294.754	146.100 ± 31.051*	9.443 ± 0.130*	0.845 ± 0.021*	2.107 ± 0.045*	1.335 ± 0.030*
AttNHP	12.007 ± 0.214	0.854 ± 0.013*	**0.297 ± 0.009**	**11049.592 ± 509.283**	144.901 ± 24.093	10.201 ± 0.097*	0.775 ± 0.018	1.891 ± 0.029	1.084 ± 0.044
NHP	13.790 ± 0.327	0.983 ± 0.042*	0.394 ± 0.009	14092.491 ± 301.340	143.534 ± 15.324*	9.795 ± 0.204*	0.811 ± 0.021	1.960 ± 0.047	1.320 ± 0.084
LogNM	12.667 ± 0.255	0.818 ± 0.034	0.407 ± 0.016*	15030.593 ± 503.492	143.010 ± 25.029*	**8.763 ± 0.113**	0.796 ± 0.033	1.833 ± 0.042	1.230 ± 0.051*
TCDDM	10.491 ± 0.134	0.825 ± 0.019	0.355 ± 0.007	13059.245 ± 109.501	142.941 ± 20.302	9.506 ± 0.107	0.753 ± 0.023	**1.634 ± 0.094**	1.046 ± 0.064
Homog. Poisson	15.203 ± 0.075*	1.007 ± 0.004*	0.582 ± 0.009*	18930.407 ± 404.338*	175.587 ± 10.333*	9.104 ± 0.058	0.924± 0.018*	2.296 ± 0.106*	1.203 ± 0.049*
CDiff	**10.019 ± 0.429**	**0.792 ± 0.028**	0.310 ± 0.014	11304.592 ± 100.049	144.551 ± 25.537*	9.259 ± 0.212	**0.686 ± 0.021**	1.733 ± 0.104	0.923 ± 0.078

Amazon dataset

	$N = 5$ events forecasting					**Interval forecasting t' small**			
	OTD	**RMSE$_e$**	**RMSE$_{x^+}$**	**MAPE**	**sMAPE**	**OTD**	**RMSE$_e$**	**RMSE$_{\|\mathbf{s}^+\|}$**	**MAE$_{\|\mathbf{s}^+\|}$**
HYPRO	9.552 ± 0.172	1.397 ± 0.033	0.433 ± 0.008	1280.563 ± 45.347	82.847 ± 0.748	8.927 ± 0.052	1.284 ± 0.010	0.805 ± 0.004	0.391 ± 0.008
Dual-TPP	11.309 ± 0.093*	1.742 ± 0.302*	0.476 ± 0.010*	1420.118 ± 52.129	86.633 ± 0.573*	**8.201 ± 0.490**	1.408 ± 0.042*	1.007 ± 0.011*	0.517 ± 0.003*
AttNHP	**9.430 ± 0.131**	**1.117 ± 0.049**	0.427 ± 0.033	1335.591 ± 55.930	83.121 ± 0.415	9.072 ± 0.059*	1.053 ± 0.041	**0.763 ± 0.015**	**0.378 ± 0.007**
NHP	11.273 ± 0.198*	1.431 ± 0.024	0.501 ± 0.009*	1456.240 ± 35.557	90.591 ± 0.667*	9.113 ± 0.135*	1.288 ± 0.018	0.978 ± 0.012*	0.493 ± 0.012*
LNM	10.230 ± 0.224*	1.663 ± 0.168*	0.447 ± 0.015	1447.203 ± 112.480*	88.900 ± 0.610	9.042 ± 0.395	1.572 ± 0.031*	0.874 ± 0.007*	0.487 ± 0.009
TCDDM	10.557 ± 0.331*	1.409 ± 0.203	0.460 ± 0.032	1392.380 ± 84.213	82.401 ± 0.810	10.003 ± 0.120*	1.338 ± 0.014	0.793 ± 0.012	0.420 ± 0.005
Homog. Poisson	12.502 ± 0.155*	2.130 ± 0.028*	0.573 ± 0.007*	1839.291 ± 54.200*	105.831 ± 0.901*	8.923 ± 0.091	2.010 ± 0.014*	1.042 ± 0.010*	0.744 ± 0.011*
CDiff	9.478 ± 0.081	1.326 ± 0.082	**0.424 ± 0.018**	**1039.338 ± 43.030**	**81.287 ± 0.994**	9.093 ± 0.049*	**1.024 ± 0.016**	0.784 ± 0.009	0.390 ± 0.007