COUNTERFACTUAL HISTORY DISTILLATION ON CONTINUOUS-TIME EVENT SEQUENCES

Anonymous authors
Paper under double-blind review

ABSTRACT

This study aims to distill history events that have essential information for predicting subsequent events with counterfactual analysis. The problem is named Counterfactual History Distillation (CHD). CHD distills a minimum set of events from history, based on which the distribution provided by a trained MTPP model fits the events observed later, and the distribution based on the remaining events in history cannot. It can help understand what event marks may have more influence on the occurrence of future events and what events in history may have a causal relationship with the events observed later. This study proposes a robust solution for CHD, called MTPP-based Counterfactual History Distiller (MTPP-CHD). MTPP-CHD learns to select the optimal event combination from history for the events observed later. Experiment results demonstrate the superiority of MTPP-CHD by outperforming baselines in terms of distillation quality and processing speed.

1 INTRODUCTION

The Marked Temporal Point Process (MTPP) (Daley & Vere-Jones, 2003) is a well-defined stochastic process that maps historical event sequences to a probability distribution which can be used to predict future events. Learning MTPP by neural networks has been well investigated (Du et al., 2016; Mei & Eisner, 2017; Omi et al., 2019; Zhang et al., 2020a; Zuo et al., 2020; Shchur et al., 2020; Mei et al., 2022; Zhang et al., 2023b; Zhou & Yu, 2023; Lüdke et al., 2023). These algorithms, belonging to the Neural Marked Temporal Point Process (NMTPP) family, enable people to train and use MTPP in high-stake real-world applications like the fake news mitigation (Farajtabar et al., 2017; Zhang et al., 2021b; 2022b) and recommendation systems (Hosseini et al., 2017; Cai et al., 2018).

Counterfactual analysis, also known as counterfactual reasoning, is one of the basic cognitive reasoning approaches. Counterfactual analysis reveals casual relations by searching for the smallest modification to the input that could completely change the output (Tan et al., 2021). For example, to investigate why one piece of disinformation becomes viral on Twitter by counterfactual analysis, we search for the answer by removing retweets of some accounts from the retweet history and then feeding the modified history to an existing model to emulate whether the disinformation still goes viral. If it stopped going viral after we removed retweets of multiple accounts and became viral again when we added them back, we would conclude that these accounts might be the culprits.

Recently, Noorbakhsh & Rodriguez (2022), Zhang et al. (2022b), and Hizli et al. (2023) investigate how the prediction of an MTPP model changes with handcrafted modifications of history with counterfactual analysis. Unlike these studies, we aim to distill a minimum subset of history events with the essential information for predicting the following events using an MTPP model with counterfactual analysis. If the history is modified by removing the minimum subset of events, the accuracy of MTPP model will drop significantly. The problem is named Counterfactual History Distillation (CHD). It can help understand what event marks may have more influence on the occurrence of subsequent events and what events in history may have a causal relationship with the events observed later.

While CHD with conventional counterfactual analysis works in concept, the distilled events are not always satisfactory. It is expected that distilled events have the essential information and the events left in history have trivial information for predicting the subsequent events. However, our study shows this is not true in many scenarios as the prediction accuracy of the MTPP model based on the distilled events is worse than that based on the events left in history. This means the result of CHD with conventional counterfactual analysis sometimes can be faulty (See Section 2.2 for more information). To address this issue, we refine CHD by adding one more constraint to enforce that distilled events are informative. Without loss of generality, perplexity (Moore & Lewis, 2010) is applied to evaluate prediction accuracy, *i.e*., how well the distribution of the next events produced by the MTPP model fits the subsequent events observed. CHD is a combinatorial problem. Inspired by the rationalization (Lei et al., 2016), we propose a solution for CHD, named MTPP-based Counterfactual History Distiller (MTPP-CHD), which probes various combinations of events in history with the support of Gumbel-softmax trick (Bengio et al., 2013; Maddison et al., 2017). We show that MTPP-CHD outperforms baseline models in terms of efficiency and distillation quality. We also demonstrate that distilled events can help understand the influence of different marks on the occurrence of future events. In summary, the contributions of this study are threefold:

1. To the best of our knowledge, this study is the first to distill a minimum subset of history events with the essential information for predicting the subsequent events using an MTPP model. We name the problem Counterfactual History Distillation (CHD).
2. This study demonstrates the issues when solving Counterfactual History Distillation (CHD) by conventional counterfactual analysis and refines it with one more constraint to ensure that the distilled events are desirable.
3. This study proposes a robust solution MTPP-CHD for CHD, which learns to select the optimal event combination from history by leveraging Gumbel-softmax trick. Experiment results demonstrate the superiority of MTPP-CHD by outperforming baselines in terms of distillation optimization and processing speed. We also demonstrate that distilled events can help understand the influence of different marks on the occurrence of future events.

2 Problem Definition

2.1 Marked Temporal Point Process

The Marked Temporal Point Process (MTPP) describes a random process of an event sequence $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$. Each event $x_i = (m_i, t_i)$ comprises a categorical mark $m_i \in \mathbb{M} = \{k_1, k_2, \cdots, k_M\}$ and its occurrence time t_i. This paper considers the simple MTPP, which only allows at most one event at every time, thus $t_i < t_j$ if $i < j$. Let $\boldsymbol{\mathcal{H}}_{t_l}$ denote the history up to(include) the time t_l when the most recent event happened and $\boldsymbol{\mathcal{H}}_{t-}$ denote the history up to(exclude) the current time t. Given $\boldsymbol{\mathcal{H}}_{t-}$, the conditional intensity function $\lambda^*(m,t)$ is the probability that an event with mark m will happen at time t (Daley & Vere-Jones, 2003)[1]:

$$\lambda^*(m,t) = \lim_{\Delta t \to 0} \frac{P\left(m, t \in (t, t+\Delta t] | \boldsymbol{\mathcal{H}}_{t-}\right)}{\Delta t}. \tag{1}$$

With $\lambda^*(m,t)$, we can define the joint probability distribution $p^*(m,t)$ of the next event whose mark is m and the time to occur is t.

$$p^*(m,t) = \lambda^*(m,t) \exp\left(-\sum_{k\in\mathbb{M}} \int_{t_l}^{t} \lambda^*(k,\tau)\mathrm{d}\tau\right). \tag{2}$$

The Negative Log-Likelihood (NLL) loss on $\boldsymbol{x}$ observed in a time interval $[t_0, T]$ is:

$$L = -\log p(\boldsymbol{x}) = -\sum_{i=1}^{n} \log \lambda^*(m_i, t_i) + \sum_{k\in\mathbb{M}} \int_{t_0}^{T} \lambda^*(k,\tau)\mathrm{d}\tau. \tag{3}$$

Equation (3) is the training loss of many MTPP models (Du et al., 2016; Mei & Eisner, 2017; Omi et al., 2019; Zhang et al., 2020a; Zuo et al., 2020; Shchur et al., 2020; Mei et al., 2022).

2.2 Problem Statement and Formulation

A dataset $\boldsymbol{D}$ contains event sequences. Suppose an MTPP model has been trained on $\boldsymbol{D}$. For any subsequence $(x_1, \cdots, x_j, x_{j+1}, \cdots, x_{n-1}, x_n)$ of an event sequence in $\boldsymbol{D}$, the first part $(x_1, \cdots, x_j)$,

[1]The asterisk reminds that this function conditions on history.

Table 1: When solving CHD with conventional counterfactual analysis, the percentage of $\mathcal{H}_d$s that have less information than the corresponding $\mathcal{H}_l$s.

StackOverflow ($|\mathbf{x}_o| = 15, |\mathcal{H}_f| = 40$)	Retweet ($|\mathbf{x}_o| = 10, |\mathcal{H}_f| = 25$)	Yelp ($|\mathbf{x}_o| = 10, |\mathcal{H}_f| = 25$)
19.068 %	0.4627 %	1.0114 %

denoted as $\mathcal{H}_f$, is the history relative to the second part $(x_{j+1}, \cdots, x_{n-1}, x_n)$, denoted as $\boldsymbol{x}_o$. The second part consists of events observed after the first part.

For each $x_i \in \boldsymbol{x}_o$, the MTPP model can be used to produce the distribution of the next event $p(x|\mathcal{H})$ where $\mathcal{H}$ includes the previous events before x_i in the subsequence. We can evaluate how well the distribution fits x_i by using $p(x_i|\mathcal{H})$. The high $p(x_i|\mathcal{H})$ indicates fitting well. $\mathcal{H}$ consists of events in $\mathcal{H}_f$ and events in $\boldsymbol{x}_o$ before x_i. We aim to search for a subset of the events in $\mathcal{H}_f$. For different subsets, the events in $\boldsymbol{x}_o$ before x_i are same. To make the presentation simple in the rest of the paper, $\mathcal{H}$ represents the events in $\mathcal{H}_f$ and ignores the events in $\boldsymbol{x}_o$. To judge if $\boldsymbol{x}_o$ fits $p(\boldsymbol{x}_o|\mathcal{H})$, we use *perplexity*, denoted as $\mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}))$. Its definition is:

$$\mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H})) = \exp\left(-\frac{1}{|\boldsymbol{x}_o|}\log\prod_{x_i\in\boldsymbol{x}_o} p(x_i|\mathcal{H})\right). \tag{4}$$

A lower perplexity indicates $p(\boldsymbol{x}_o|\mathcal{H})$ better fitting $\boldsymbol{x}_o$. Perplexity has been widely used to select in-domain data from non-specific-domain datasets (Moore & Lewis, 2010; Toral et al., 2015; Feng et al., 2022) and evaluation of Large Language Models (LLMs) (Brown et al., 2020; Du et al., 2022; Zhang et al., 2022a; Zeng et al., 2022).

Counterfactual History Distillation (CHD) aims to distill essential events in $\mathcal{H}_f$ that enable the MTPP model to generate $p(\boldsymbol{x}_o|\mathcal{H}_f)$ fitting $\boldsymbol{x}_o$. Following conventional counterfactual analysis, the problem is to identify the minimum subset $\mathcal{H}_d \subseteq \mathcal{H}_f$ so that $p(\boldsymbol{x}_o|\mathcal{H}_f)$ fits $\boldsymbol{x}_o$, but $p(\boldsymbol{x}_o|\mathcal{H}_l = \mathcal{H}_f - \mathcal{H}_d)$ does not. The formal definition is:

$$\begin{aligned} &\min_{\mathcal{H}_d\subseteq\mathcal{H}_f} |\mathcal{H}_d| \\ &\text{s.t.} \quad \frac{\mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_f))}{\mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_l))} \leqslant \epsilon_l \end{aligned} \tag{5}$$

where $\epsilon_l \in (0, 1)$ is a threshold to ensure the information in $\mathcal{H}_l$ about $\boldsymbol{x}_o$ is trivial.

However, our study shows that the result of the conventional counterfactual analysis is problematic in many scenarios. Table 1 shows the percentage of subsequences in three real-world datasets where $\mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_d))$ is greater than $\mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_l))$ by solving the optimization problem in Equation (5)[2]. This means $\mathcal{H}_l$ sometimes contains more information about $\boldsymbol{x}_o$ than $\mathcal{H}_d$, which is undesirable. To address this issue, we refine CHD by adding one more constraint to enforce that the information in $\mathcal{H}_d$ is significantly more than $\mathcal{H}_l$ for predicting $\boldsymbol{x}_o$.

$$\begin{aligned} &\min_{\mathcal{H}_d\subseteq\mathcal{H}_f} |\mathcal{H}_d| \\ &\text{s.t.} \quad \frac{\mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_f))}{\mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_l))} \leqslant \epsilon_l, \\ &\qquad \frac{\mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_f))}{\mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_d))} \geqslant \epsilon_d. \end{aligned} \tag{6}$$

where $\epsilon_d \in (0, 1)$ is another threshold to ensure the information in $\mathcal{H}_d$ about $\boldsymbol{x}_o$ is sufficient. $\epsilon_d > \epsilon_l$. For the ease of computation, we apply logarithm to the constraints in Equation (6):

$$\begin{aligned} &\min_{\mathcal{H}_d\subseteq\mathcal{H}_f} |\mathcal{H}_d| \\ &\text{s.t.} \quad \log\mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_f)) - \log\mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_l)) \leqslant \log\epsilon_l, \\ &\qquad \log\mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_f)) - \log\mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_d)) \geqslant \log\epsilon_d. \end{aligned} \tag{7}$$

[2]Here, we solve the optimization problem by training a MTPP-CHD with L_n and L_l. See Section 3 for definitions of MTPP-CHD, L_n, and L_l.

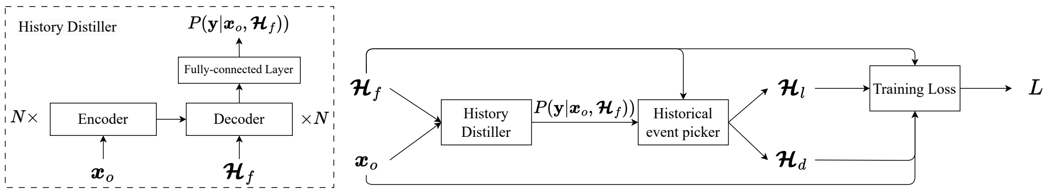


Figure 1: Architecture of MTPP-CHD.

The perplexity of $p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}})$ is tricky if $\boldsymbol{\mathcal{H}}$ is an empty set $\varnothing$. In this case, we have $p(\boldsymbol{x}_o|\varnothing) = p(\boldsymbol{x}_o, \varnothing)/p(\varnothing)$ where $p(\varnothing) = 0$. Due to division by 0, $p(\boldsymbol{x}_o|\varnothing)$ is undefined. Intuitively, when we continuously remove events from $\boldsymbol{\mathcal{H}}$, the information in $\boldsymbol{\mathcal{H}}$ decreases so that $p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}})$ approaches zero. So, we define $p(\boldsymbol{x}_o|\varnothing)$ an infinitesimal number, which induces $\text{ppl}(p(\boldsymbol{x}_o|\varnothing)) \to +\infty$. We have the following proposition (proven in Appendix A.1):

Proposition 1. *Counterfactual History Distillation (CHD) defined in Equation (7) always has a solution for any $\epsilon_l \in (0, 1)$, $\epsilon_d \in (0, 1)$, and $\epsilon_d > \epsilon_l$.*

3 MTPP-BASED COUNTERFACTUAL HISTORY DISTILLER (MTPP-CHD)

The proposed CHD solution, MTPP-based Counterfactual History Distiller (MTPP-CHD), is sketched in Figure 1. MTPP-CHD consists of three components. The first component, *history distiller*, processes $\boldsymbol{\mathcal{H}}_f$ and $\boldsymbol{x}_o$ using an encoder-decoder transformer, then pushes the resultant representations into a fully connected layer. The output is $p(\mathbf{y}|\boldsymbol{\mathcal{H}}_f, \boldsymbol{x}_o)$. Here, $\mathbf{y}$ is a mask vector of size $|\boldsymbol{\mathcal{H}}_f|$, each for one event in $\boldsymbol{\mathcal{H}}_f$. For $y_i \in \mathbf{y}$, if $y_i = 0$, the corresponding element $x_i \in \boldsymbol{\mathcal{H}}_f$ goes to $\boldsymbol{\mathcal{H}}_l$. If $y_i = 1$, the corresponding element $x_i \in \boldsymbol{\mathcal{H}}_f$ goes to $\boldsymbol{\mathcal{H}}_d$. All trainable parameters are in the first component. The second component, *historical event picker*, derives $\boldsymbol{\mathcal{H}}_l$ and $\boldsymbol{\mathcal{H}}_d$ based on $p(\mathbf{y}|\boldsymbol{\mathcal{H}}_f, \boldsymbol{x}_o)$. The third component, *training loss*, employs a trained MTPP model to evaluate the derived $\boldsymbol{\mathcal{H}}_l$ and $\boldsymbol{\mathcal{H}}_d$ for training MTPP-CHD. The third component only exists during training.

3.1 TRAINING OF MTPP-CHD

Training MTPP-CHD begins by initializing the parameters of the history distiller. Given history $\boldsymbol{\mathcal{H}}_f$ and $\boldsymbol{x}_o$, the history distiller processes them using an encoder-decoder transformer to represent each event in $\boldsymbol{\mathcal{H}}_f$ so that it is aware of other events in $\boldsymbol{\mathcal{H}}_f$ and events in $\boldsymbol{x}_o$. Then, the representations of events in $\boldsymbol{\mathcal{H}}_f$ are fed to a fully connected layer and the output is $p(\mathbf{y}|\boldsymbol{\mathcal{H}}_f, \boldsymbol{x}_o)$, the distribution of mask vector $\mathbf{y}$. For $y_i \in \mathbf{y}$, $p(y_i|\boldsymbol{\mathcal{H}}_f, \boldsymbol{x}_o)$ is a categorical distribution of two categories, *i.e.*, $\{0, 1\}$. If $p(y_i = 0|\boldsymbol{\mathcal{H}}_f, \boldsymbol{x}_o)$ is larger, the corresponding event is more likely to go to $\boldsymbol{\mathcal{H}}_l$. If $p(y_i = 1|\boldsymbol{\mathcal{H}}_f, \boldsymbol{x}_o)$ is larger, the corresponding event is more likely to go to $\boldsymbol{\mathcal{H}}_d$.

Algorithm 1 Historical event picker during training.

Input: $\boldsymbol{\mathcal{H}}_f$ and $p(\mathbf{y}|\boldsymbol{x}_o, \boldsymbol{\mathcal{H}}_f)$;
Output: $\boldsymbol{\mathcal{H}}_l$ and $\boldsymbol{\mathcal{H}}_d$;
$\boldsymbol{\mathcal{H}}_d = \varnothing, \boldsymbol{\mathcal{H}}_l = \varnothing$;
$\hat{\mathbf{y}} \leftarrow$ sampling $p(\mathbf{y}|\boldsymbol{x}_o, \boldsymbol{\mathcal{H}}_f)$ with Gumbel-softmax trick;
for $x_i \in \boldsymbol{\mathcal{H}}_f$ **do**
  if $\hat{y}_i == 1$ **then**
    $\boldsymbol{\mathcal{H}}_d \leftarrow \boldsymbol{\mathcal{H}}_d \cup x_i$;
  else
    $\boldsymbol{\mathcal{H}}_l \leftarrow \boldsymbol{\mathcal{H}}_l \cup x_i$;
  end if
end for
return $\boldsymbol{\mathcal{H}}_l, \boldsymbol{\mathcal{H}}_d$;

Next, the historical event picker samples masks from $p(\mathbf{y}|\boldsymbol{\mathcal{H}}_f, \boldsymbol{x}_o)$ and obtains the corresponding $\boldsymbol{\mathcal{H}}_d$ and $\boldsymbol{\mathcal{H}}_l$ for multiple times. Algorithm 1 shows how a sample, denoted as $\hat{\mathbf{y}}$, is drawn and processed to return $\boldsymbol{\mathcal{H}}_d$ and $\boldsymbol{\mathcal{H}}_l$ during training. Specially, a sample is drawn by sampling categorical distribution $p(y_i|\boldsymbol{\mathcal{H}}_f, \boldsymbol{x}_o)$ for each element $y_i \in \mathbf{y}$. If $\hat{y}_i = 0$, the corresponding element in $\boldsymbol{\mathcal{H}}_f$ goes to $\boldsymbol{\mathcal{H}}_l$. If $\hat{y}_i = 1$, the corresponding element in $\boldsymbol{\mathcal{H}}_f$ goes to $\boldsymbol{\mathcal{H}}_d$. To draw sample from $p(y_i|\boldsymbol{\mathcal{H}}_f, \boldsymbol{x}_o)$ in a differentiable way, we use the Gumbel-softmax trick (Bengio et al., 2013; Maddison et al., 2017). After the sample $\hat{\mathbf{y}}$ is drawn, the distilled events form $\boldsymbol{\mathcal{H}}_d$ and the remaining events constitute $\boldsymbol{\mathcal{H}}_l$. In natural language processing, a similar method has been used for rationalization (Lei et al., 2016) to search a document for an optimal combination of sentences related to a claim.

The third component evaluates $\boldsymbol{\mathcal{H}}_d$ and $\boldsymbol{\mathcal{H}}_l$ for the loss. According to Equation (7), the loss function of MTPP-CHD comprises two aspects: L_e for enforcing perplexity-based constraints and L_n for minimizing the length of $\mathcal{H}_d$. With $\mathcal{H}_d$ and $\mathcal{H}_l$ derived from sample $\hat{\mathbf{y}}$, we enforce perplexity-based constraints in a differentiable way by using two surrogate hinge losses, inspired by (Mothilal et al., 2020; Tan et al., 2021):

$$\begin{aligned} L_l(\hat{\mathbf{y}}) &= \max(\log \mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_f)) - \log \mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_l)) - \log \epsilon_l, 0). \\ L_d(\hat{\mathbf{y}}) &= \max(\log \mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_d)) - \log \mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_f)) + \log \epsilon_d, 0). \end{aligned} \tag{8}$$

Reducing loss L_l will increase $\log \mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_l))$ until its gap to $\log \mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_f))$ is larger than ϵ_l. Reducing loss L_d will decrease $\log \mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_d))$ until its gap to $\log \mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_f))$ is smaller than $\log \epsilon_d$. The loss L_e is based on N samples from $p(\mathbf{y}|\mathcal{H}_f, \boldsymbol{x}_o)$, where $\hat{\mathbf{y}}_i$ refers to the i-th sample:

$$\begin{aligned} L_e &= \mathbb{E}_{\hat{\mathbf{y}} \sim p(\mathbf{y}|\mathcal{H}_f, \boldsymbol{x}_o)}(L_l(\hat{\mathbf{y}}) + L_d(\hat{\mathbf{y}})) \\ &\approx \frac{1}{N}\sum_{i=1}^{N}(L_l(\hat{\mathbf{y}}_i) + L_d(\hat{\mathbf{y}}_i). \end{aligned} \tag{9}$$

We use a trained MTPP model to estimate the conditional probability distribution $p(\boldsymbol{x}_o|\mathcal{H})$ in Equation (8). Any MTPP models outputting $p^*(m, t)$ defined in Equation (2) should work. The only requirement is that the MTPP model is differentiable, so MTPP-CHD obtains the gradient $\nabla_{\hat{\mathbf{y}}} L_e$ to enable training. In this paper, the trained MTPP model is FullyNN (Omi et al., 2019). Details about FullyNN and how we train FullyNN on $\boldsymbol{D}$ are available in Appendix B.2.

The loss L_n aims to minimize the length of $\mathcal{H}_d$, *i.e.*, the number of distilled events. Because $\mathcal{H}_d$ is derived from $\hat{\mathbf{y}}$, minimizing the length of $\mathcal{H}_d$ equals to maximizing ℓ^0-norm of $\hat{\mathbf{y}}$, *i.e.*, the number of nonzero elements in $\hat{\mathbf{y}}$. However, the ℓ^0-norm is not differentiable. As a workaround, some studies optimize the differentiable ℓ^1-norm (Tan et al., 2021). However, optimizing ℓ^1-norm of a vector $\mathbf{a} \in \mathbb{R}^d$ has limited effects on optimizing ℓ^0-norm because there is no consistent relation between them. ℓ^0-norm can decrease, stay unchanged, or even increase when ℓ^1-norm decreases. Interestingly, ℓ^1-norm of $\hat{\mathbf{y}}$ has a consistent relation with ℓ^0-norm of $\hat{\mathbf{y}}$ because $\hat{\mathbf{y}}$ only contains 0 and 1. This means that ℓ^0-norm is always equal to ℓ^1-norm for $\hat{\mathbf{y}}$. This means optimizing $\hat{\mathbf{y}}$'s ℓ^1-norm is equivalent to optimizing $\hat{\mathbf{y}}$'s ℓ^0-norm. We define L_n as the normalized ℓ^1-norm by dividing the length of $\hat{\mathbf{y}}$:

$$L_n = \frac{\|\hat{\mathbf{y}}\|_1}{|\hat{\mathbf{y}}|}. \tag{10}$$

With L_e and L_n properly defined, the training loss L of MTPP-CHD is the sum of L_n and L_e. We use two hyperparameters α and β to balance the number of distilled events and the perplexity gap.

$$L = \alpha L_n + \beta L_e. \tag{11}$$

3.2 Inference of MTPP-CHD

Counterfactual history distillation with the learned MTPP-CHD consists of the trained history distiller and an inference-specific historical event picker. The inference process is presented in Algorithm 2. The history distiller takes in history $\mathcal{H}_f$ and $\boldsymbol{x}_o$ for $p(\mathbf{y}|\mathcal{H}_f, \boldsymbol{x}_o)$. During inference, the historical event picker returns the optimal $\mathcal{H}_d$ based on $p(\mathbf{y}|\mathcal{H}_f, \boldsymbol{x}_o)$. To do that, elements $y_i \in \mathbf{y}$ are sorted in descending order based on $p(y_i = 1|\mathcal{H}_f, \boldsymbol{x}_o)$. Initially, $\mathcal{H}_d$ is empty and $k = 1$. The top-k element is moved to $\mathcal{H}_d$ and $\mathcal{H}_l$ includes the remaining elements. With the trained MTPP model, $\mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_l))$ and $\mathrm{ppl}(p(\boldsymbol{x}_o|\mathcal{H}_d))$ are calculated. If the constraints in Equation (7) are satisfied, $\mathcal{H}_d$ is returned. If not, $k = k + 1$ and the same process is taken until the constraints in Equation (7) are satisfied and $\mathcal{H}_d$ is returned.

Algorithm 2 Historical event picker during inference.

Input: $\mathcal{H}_f$ and $p(\mathbf{y}|\boldsymbol{x}_o, \mathcal{H}_f)$;
Output: $\mathcal{H}_d$;
$\mathcal{H}_f^* \leftarrow$ sort $\mathcal{H}_f$ in descending order of $p(y_i|\boldsymbol{x}_o, \mathcal{H}_f)$;
$\mathcal{H}_d = \varnothing, \mathcal{H}_l = \mathcal{H}_f$;
for x_i in $\mathcal{H}_f^*$ **do**
  if $\mathcal{H}_d$, $\mathcal{H}_l$ satisfy the constraints in Equation (7) **then**
    break;
  end if
  $\mathcal{H}_d \leftarrow \mathcal{H}_d \cup x_i$;
  $\mathcal{H}_l \leftarrow \mathcal{H}_l - x_i$;
end for
return $\mathcal{H}_d$;

4 Experiments

This section evaluates the effectiveness of MTPP-CHD by answering following questions: (i) Does solving CHD in Equation (7) lead to better distillation compared with Equation (5)? (ii) Does the proposed MTPP-CHD solve CHD with L_n and L_e with good distillation quality and efficiency? , and (iii) What statistical features or knowledge can be exploited from the $\boldsymbol{\mathcal{H}}_d$?

4.1 EXPERIMENT SETTINGS

The same experiments are run 3 times with different random seeds, and their mean and standard deviation (1-sigma) are reported. More details are available in Appendix B.2 including the hardware and software for the experiments, the hyperparameters of MTPP-CHD, the setting of ϵ_l and ϵ_d, and a brief introduction of FullyNN.

Baseline Models To our knowledge, no previous studies investigated CHD in the context of MTPP. This means we do not have baselines from existing studies to compare with. Brute force is infeasible because solving a combinatorial problem like CHD is NP-hard (Karp, 1972). We notice some studies applying counterfactual analysis in recommender systems. They greedily search for the smallest subset of history that the recommendation would change with the subset removed (Ghazimatin et al., 2020; Tran et al., 2021; Zhong & Negre, 2022). This motivates us to adopt a Greedy Search (GS) baseline. It solves CHD by incrementally selecting from $\boldsymbol{\mathcal{H}}_f$ the event that increases the gap between $\log \mathrm{ppl}(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_l)$ and $\log \mathrm{ppl}(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_d)$ the most and inserting it to $\boldsymbol{\mathcal{H}}_d$ until the two constraints are satisfied. We also take a Random Distillation (RD) baseline to show the difficulty of CHD. RD randomly moves Q events from $\boldsymbol{\mathcal{H}}_f$ to $\boldsymbol{\mathcal{H}}_d$ and calculates the gap between $\log \mathrm{ppl}(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_f)$ and $\log \mathrm{ppl}(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_l)$. This is repeated multiple times and the average of these gaps is recorded. Q starts from 0. RD stops and returns Q when the average gaps satisfy the two constraints in Equation (7); otherwise, increase Q by 1 and repeat the previous process.

Evaluation Metrics We are concerned to which extent the optimization objective of CHD is achieved, *i.e.*, minimizing $|\boldsymbol{\mathcal{H}}_d|$ while two constraints are satisfied. For all ($\boldsymbol{\mathcal{H}}_f$,$\boldsymbol{x}_o$) pairs in the test dataset $\boldsymbol{T}$, we calculate the average length of $\boldsymbol{\mathcal{H}}_d$ provided by a CHD approach.

$$|\bar{\boldsymbol{\mathcal{H}}}_d| = \frac{1}{|\boldsymbol{T}|} \sum_{(\boldsymbol{\mathcal{H}}_f,\boldsymbol{x}_o)\in \boldsymbol{T}} |\boldsymbol{\mathcal{H}}_d|. \tag{12}$$

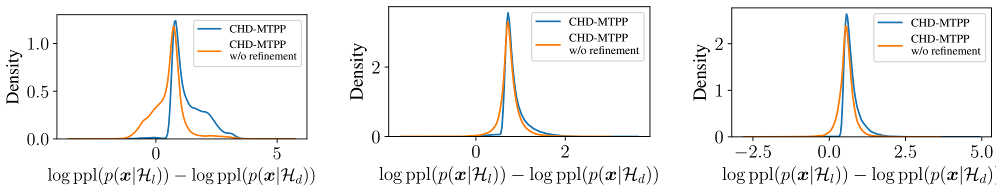
Lower $|\bar{\boldsymbol{\mathcal{H}}}_d|$ indicates the CHD approach obtains shorter $\boldsymbol{\mathcal{H}}_d$ that meets the constraints in Equation (7), thus better.

Datasets We test MTPP-CHD and baselines on three real-world datasets: Retweet (Zhao et al., 2015), StackOverflow (Leskovec & Krevl, 2014) and Yelp. Retweet contains 2.6 million events, StackOverflow 480K events, and Yelp 400K events. All subsequences with $n = |\boldsymbol{\mathcal{H}}_f| + |\boldsymbol{x}_o|$ events are extracted from these datasets. Further, each subsequence is split into $\boldsymbol{\mathcal{H}}_f$ and $\boldsymbol{x}_o$. Each dataset has 5 different $|\boldsymbol{\mathcal{H}}_f|$ and $|\boldsymbol{x}_o|$ settings. More details are presented in Appendix B.

4.2 EXPERIMENT RESULTS

4.2.1 EFFECTIVENESS OF COUNTERFACTUAL ANALYSIS REFINEMENT

CHD can be tackled by working out the optimization problem defined in Equation (5). This method is based on counterfactual analysis but is problematic as pointed out in Section 2.2. To prevent such undesirable results, we refine the counterfactual analysis with a new constraint on $\boldsymbol{\mathcal{H}}_d$ as defined in Equation (7). To investigate the impact of the new constraint, we compare MTPP-CHD, our solution of CHD based on the counterfactual analysis with $\boldsymbol{\mathcal{H}}_d$ constraint, against MTPP-CHD without refinement, based on counterfactual analysis without $\boldsymbol{\mathcal{H}}_d$ constraint. Figure 2 presents the distribution of $\log \mathrm{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_l)) - \log \mathrm{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_d))$ of our MTPP-CHD and the MTPP-CHD without refinement. If $\boldsymbol{\mathcal{H}}_d$ has less information than $\boldsymbol{\mathcal{H}}_l$, the value of $\log \mathrm{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_l)) - \log \mathrm{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_d))$ is less than zero; otherwise greater than 0. Our MTPP-CHD demonstrates the resultant $\boldsymbol{\mathcal{H}}_d$ always has more information than the corresponding $\boldsymbol{\mathcal{H}}_l$. In contrast, MTPP-CHD without refinement may lead to some resultant $\boldsymbol{\mathcal{H}}_d$s having less information than corresponding $\boldsymbol{\mathcal{H}}_l$s. Such an undesirable situation is significant on StackOverflow.



(a) StackOverflow ($|\mathbf{x}_o| = 15$, $|\mathcal{H}_f| = 40$)

(b) Retweet ($|\mathbf{x}_o| = 10$, $|\mathcal{H}_f| = 25$)

(c) Yelp ($|\mathbf{x}_o| = 10$, $|\mathcal{H}_f| = 25$)

Figure 2: The distribution of $\log \mathrm{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_l)) - \log \mathrm{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_d))$ of our MTPP-CHD and the MTPP-CHD without refinement.

4.2.2 Distillation Quality

We solve CHD by working out the optimization problem defined in Equation (7), where the optimization objective is to identify $\boldsymbol{\mathcal{H}}_d$ with the minimum number of events under two constraints. The resultant $\boldsymbol{\mathcal{H}}_d$ with fewer events indicates a better solution. Table 2 reports $|\bar{\boldsymbol{\mathcal{H}}}_d|$ using our MTPP-CHD and baselines. First, GS outperforms RD by a consistent and noticeable margin on all datasets. It demonstrates that CHD is a difficult task that cannot be properly solved with a simple solution like RD. Second, our MTPP-CHD demonstrates the performance better than both baselines. With GS, it repeatedly identifies the individual event that affects L_e the most and moves it from $\boldsymbol{\mathcal{H}}_f$ to $\boldsymbol{\mathcal{H}}_d$. However, this method cannot capture the effect of event combinations in $\boldsymbol{\mathcal{H}}_f$ and may lead to suboptimal solutions. In contrast, our MTPP-CHD overcomes the weakness of GS by searching for optimal event combinations and therefore demonstrates better performance.

Table 2: The average length of $\boldsymbol{\mathcal{H}}_d$ returned by MTPP-CHD and baselines (the standard deviation of GS is 0 because GS is deterministic).

	$\lvert\mathbf{x}_o\rvert$	$\lvert\mathcal{H}_f\rvert$	MTPP-CHD	GS	RD
StackOverflow	15	40	**21.484**$\pm$**0.0073**	23.681$\pm$0.0000	36.424$\pm$0.0033
	15	45	**23.700**$\pm$**0.0802**	25.700$\pm$0.0000	40.582$\pm$0.0042
	15	50	**26.115**$\pm$**0.5226**	27.699$\pm$0.0000	44.693$\pm$0.0015
	20	50	**27.416**$\pm$**0.0974**	28.927$\pm$0.0000	44.898$\pm$0.0046
	25	50	**27.811**$\pm$**0.2973**	29.636$\pm$0.0000	45.159$\pm$0.0011
Retweet	10	25	**12.281**$\pm$**0.2001**	14.722$\pm$0.0000	24.004$\pm$0.0004
	10	30	**13.297**$\pm$**0.2264**	16.511$\pm$0.0000	28.620$\pm$0.0003
	10	35	**14.390**$\pm$**0.0899**	18.053$\pm$0.0000	33.207$\pm$0.0018
	15	35	**20.632**$\pm$**0.5377**	24.875$\pm$0.0000	34.532$\pm$0.0008
	20	35	**28.140**$\pm$**1.4211**	29.990$\pm$0.0000	34.894$\pm$0.0006
Yelp	10	25	**9.6412**$\pm$**0.0148**	11.640$\pm$0.0000	23.112$\pm$0.5788
	10	30	**9.8174**$\pm$**0.0898**	12.587$\pm$0.0000	27.396$\pm$0.7311
	10	35	**10.008**$\pm$**0.2310**	13.508$\pm$0.0000	31.600$\pm$0.9164
	15	35	**13.422**$\pm$**0.0436**	18.237$\pm$0.0000	33.257$\pm$0.6701
	20	35	**18.160**$\pm$**0.4387**	22.562$\pm$0.0000	34.114$\pm$0.4259

4.2.3 Effectiveness of L_e and L_n

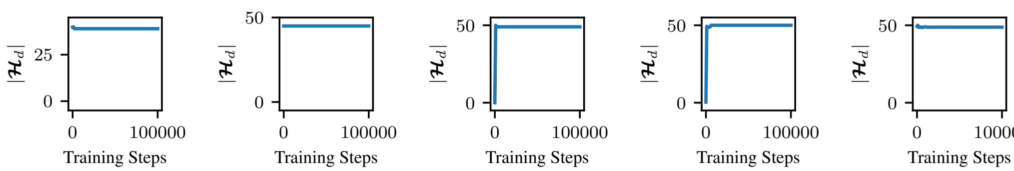
Training MTPP-CHD is achieved by minimizing loss L_e and L_n. Minimizing L_e is applied to force MTPP-CHD to move more events from $\boldsymbol{\mathcal{H}}_f$ to $\boldsymbol{\mathcal{H}}_d$ so that the two constraints in MTPP-CHD are satisfied. On the other hand, minimizing L_n is applied to encourage MTPP-CHD to move fewer events from $\boldsymbol{\mathcal{H}}_f$ to $\boldsymbol{\mathcal{H}}_d$ so that $|\boldsymbol{\mathcal{H}}_d|$ is minimized. To verify that, Figure 3 (a) report the number of events in $\boldsymbol{\mathcal{H}}_d$ returned by the MTPP-CHD trained by minimizing L_e only on dataset StackOverflow, and Figure 3 (b) report the number of events in $\boldsymbol{\mathcal{H}}_d$ returned by the MTPP-CHD trained by minimizing L_n only on dataset StackOverflow. As expected, all events in $|\boldsymbol{\mathcal{H}}_f|$ are moved to $|\boldsymbol{\mathcal{H}}_d|$ in the former while no events in $|\boldsymbol{\mathcal{H}}_f|$ are moved to $|\boldsymbol{\mathcal{H}}_d|$ in the latter. The same results can be observed on other datasets in Appendix C.2).

Table 3: Total time used to solve all CHD tasks in test data (first three columns) and time used for MTPP-CHD training (last column).

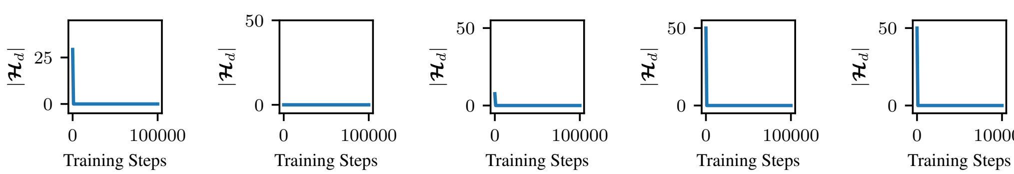
	MTPP-CHD	GS	RD	MTPP-CHD(Training)
StackOverflow ($\lvert\mathbf{x}_o\rvert = 15$,$\lvert\mathcal{H}_f\rvert = 40$)	**2.86h**	33.4h	27.2h	24.3h
Retweet ($\lvert\mathbf{x}_o\rvert = 10$, $\lvert\mathcal{H}_f\rvert = 25$)	**9.09h**	67.0h	83.8h	29.7h
Yelp ($\lvert\mathbf{x}_o\rvert = 10$, $\lvert\mathcal{H}_f\rvert = 25$)	**1.81h**	13.5h	17.0h	14.3h

4.2.4 Model Efficiency

This section reports the performance of MTPP-CHD and baselines regarding time efficiency. For MTPP-CHD, it must be trained first to learn model parameters using training data and then solve CHD. For GS and RD, they are directly applied to solve CHD because they have no parameter to train. In Table 3, the first three columns report the total time of the trained MTPP-CHD and baselines to solve CHD on all $(\boldsymbol{\mathcal{H}}_f, \boldsymbol{x}_o)$ pairs in three test datasets. More results are available in Appendix Table 9. The results tell that MTPP-CHD is significantly faster than baselines. The reason is that GS and RD have to interact with the MTPP model multiple times for one $\boldsymbol{\mathcal{H}}_d$. On the other hand, the trained MTPP-CHD does not need to interact with MTPP model because it already learned which event should be distilled from MTPP during training. To have a better understanding of the time efficiency for MTPP-CHD, the last column of Table 3 reports the time used by MTPP-CHD for training (see Table 6 for training data size). It is comparable with the time consumed by GS and RD. Since MTPP-CHD only needs to be trained once, it is much more efficient compared with GS and RD.



(a) The number of event in $\mathcal{H}_d$ returned by MTPP-CHD trained by minimizing L_e only.



(b) The number of event in $\mathcal{H}_d$ returned by MTPP-CHD trained by minimizing L_n only.

Figure 3: Effectiveness of L_e and L_n (from left to right: $(|\boldsymbol{x}_o|, |\boldsymbol{\mathcal{H}}_f|) = (15, 40)$, $(15, 45)$, $(15, 50)$, $(20, 50)$, $(25, 50)$).

4.3 Analysis of Distilled Events

The resultant $\boldsymbol{\mathcal{H}}_d$ is a minimum subset of events in $\boldsymbol{\mathcal{H}}_f$ that represents the essential information in history from the perspective of the underlying MTPP model. Specifically, the accuracy of MTPP model based on $\boldsymbol{\mathcal{H}}_d$ is close to $\boldsymbol{\mathcal{H}}_f$, and the accuracy of MTPP based on $\boldsymbol{\mathcal{H}}_l$ is significantly lower than $\boldsymbol{\mathcal{H}}_f$. Investigating the events in $\boldsymbol{\mathcal{H}}_d$ may disclose interesting insights.

Given a dataset, the events with particular marks may influence the occurrence of the subsequent events more, for example, a retweet by famous users in Retweet. To verify it, we compare $\boldsymbol{\mathcal{H}}_d$ returned using MTPP-CHD against $\boldsymbol{\mathcal{H}}_d$ using RD on the test data of Retweet in terms of mark percentage. The mark percentage is calculated as the ratio of the number of events for that mark in $\boldsymbol{\mathcal{H}}_d$s to the number of events for the same mark in $\boldsymbol{\mathcal{H}}_f$s within the test data. RD randomly selects events from $\boldsymbol{\mathcal{H}}_f$ to constitute $\boldsymbol{\mathcal{H}}_d$. In contrast, $\boldsymbol{\mathcal{H}}_d$ returned using MTPP-CHD has the essential information for predicting the next events. If a mark has more influence on the occurrence of the subsequent events, the mark is expected to be more frequent in $\boldsymbol{\mathcal{H}}_d$ returned using MTPP-CHD than using RD. From Figure 4, Mark 2 refers to famous users. We can observe that Mark 2 is consistently more frequent in $\mathcal{H}_d$ returned using MTPP-CHD while other marks are not. The result tells that the retweets by famous users have more influence on the occurrence of the subsequent retweets.

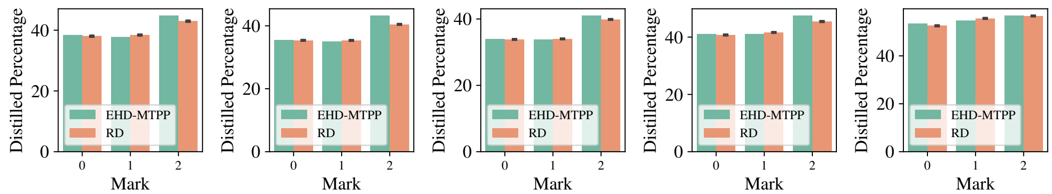


Figure 4: The percentage of events for different marks in $\mathcal{H}_d$ returned by MTPP-CHD and Random Distillation (RD) on test date of Retweet (from left to right: $(|\boldsymbol{x}_o|, |\mathcal{H}_f|) = (10, 25)$, $(10, 30)$, $(10, 35)$, $(15, 35)$, $(20, 35)$). All results pass the significance test with p-value 0.

5 Related Works

5.1 Counterfactual analysis

Counterfactual analysis on MTPP models Recently, Noorbakhsh & Rodriguez (2022), Zhang et al. (2022b) and Hizli et al. (2023) used counterfactual analysis to investigate how the prediction of an MTPP model changes with handcrafted modifications of history. Noorbakhsh & Rodriguez (2022) successfully perform counterfactual analysis on the Hawkes process, an instance of MTPP, by deterministically accepting or rejecting the future events generated by the thinning algorithm (Ogata, 1981). Zhang et al. (2022b) use counterfactual analysis to estimate the influence of fake news engagements. By comparing the intensity function with manually modified history, they discover that users tend to behave differently if they recently engaged in misinformation. Hizli et al. (2023) use counterfactual analysis to evaluate the effect of medical treatments by checking how the blood glucose dynamics changes with and without a specific treatment.

CHD differs from existing counterfactual analysis related to MTPP models (Noorbakhsh & Rodriguez, 2022; Zhang et al., 2022b; Hizli et al., 2023). They investigate how a predefined modification to history would change the prediction of MTPP models. In contrast, CHD aims to find a minimal modification $\mathcal{H}_d$ so that the MTPP model can generate a distribution fitting $\boldsymbol{x}_o$ based on $\mathcal{H}_d$ but cannot based on $\mathcal{H}_l$. In summary, the methods in these studies cannot solve CHD.

Counterfactual analysis on Classifiers Some researchers use counterfactual analysis to analyze how binary and multi-class classifiers make decisions and name the task Counterfactual Explanations (CFE) (Verma et al., 2020). The definition of CFE involves a classifier f, an input feature $\mathbf{x}$, and an expected output y. We expect a counterfactual input $\mathbf{x}'$ by solving the following optimization problem:

$$\begin{aligned} \arg\min_{\mathbf{x}'} \quad & d(\mathbf{x}, \mathbf{x}') \\ \text{s.t.} \quad & f(\mathbf{x}') = y' \end{aligned} \tag{13}$$

where $d(\mathbf{x}, \mathbf{x}')$ refers to the distance between $\mathbf{x}$ and $\mathbf{x}'$. Equation (13) means the expected $\mathbf{x}'$ should be similar to $\mathbf{x}$ while still changes the classification result from y to y'. Usually, the similarity between $\mathbf{x}$ and $\mathbf{x}'$ means we should change as few features as possible, but sometimes it means the overall modification to $\mathbf{x}$ should be as small as possible (Verma et al., 2020). CFE generation is a well-investigated task with many existing works (Wachter et al., 2017; Dhurandhar et al., 2018; 2019; Joshi et al., 2019; Kanamori et al., 2020; Mothilal et al., 2020; Ramakrishnan et al., 2020; Parmentier & Vidal, 2021; Chen et al., 2022).

CHD is fundamentally different from CFE. CFE modifies the continuous input that would change the discrete output of a classifier (Verma et al., 2020). However, CHD manipulates the discrete input sequence that would change the continuous output of the MTPP model, *i.e.*, the accuracy for predicting the events observed later.

Counterfactual analysis on Recommendation Systems The recommendation system community has used counterfactual analysis to investigate how user behaviors and item features affect recommendation results (Mehrotra et al., 2018; Wang et al., 2020; Ghazimatin et al., 2020; Yang et al., 2021; Tran et al., 2021; Wang et al., 2021; Xu et al., 2021; Wang et al., 2022b; Zhong & Negre, 2022; Zhang et al., 2022b; Mu et al., 2022; Zhang et al., 2023a). Ghazimatin et al. (2020) proposed PRINCE, the first approach explaining recommendations concerning users' activities in Heterogeneous Information Networks(HIN). By greedily removing as few events as possible from the historical user event sequence that could replace the current recommendation with a different item, PRINCE identifies which interactions are responsible for model decisions. PRINCE heavily relies on the structure of HIN to efficiently find the solution, which limits its general use. To solve this, Tran et al. (2021) proposed ACCENT. It greedily searches for the smallest subset of history that the recommendation would change after training a new system with the subset removed. Zhong & Negre (2022) discuss applying SHAP(SHapley Additive exPlanations) (Lundberg & Lee, 2017) to greedily select features as the recommendation explanation. Zhang et al. (2023a) proposed PaGE-LINK. This graph-based explanation algorithm exploits the complete graph information from a learned GNN recommender to explain the recommendation results.

Besides, counterfactual analysis has been applied to understand how the reinforcement learning agent behaves in different environment states (Atrey et al., 2020; Wang et al., 2019; Li et al., 2021a; Zhou et al., 2022; Ji et al., 2023). Some researchers realize that they can detect and mitigate the bias in pretrained computer vision and language models by counterfactual analysis (Huang et al., 2020; Abbasnejad et al., 2020; Zhang et al., 2020c; Niu et al., 2021; Qian et al., 2021; Wang et al., 2022a).

5.2 Logic Point Processes

Besides counterfactual analysis, researchers have developed other ways to find causal relations between events on continuous time. One of the favorites is the Granger causality (Xu et al., 2016; Zhang et al., 2020b; Marcinkevics & Vogt, 2021; Zhu et al., 2022; Jalaldoust et al., 2022). Granger causality explores mutual relations across different marks, checking which mark helps the event forecast on other marks. Other works exploit logic rules from the temporal relation between different events, *e.g.*, one event happens before another event, then construct the conditional intensity function based on these relations (Li et al., 2021b; Yang et al., 2024; Song et al., 2024). Shi et al. (2023) use logic rules extracted by LLMs to improve the accuracy of next-event prediction. Zhang et al. (2021a) report an unsupervised approach to pick out exogenous events from a given sequence, called TPP-Select. TPP-Select separates all observed events into two types: endogenous events and exogenous events. Endogenous events occur because of historical influence, while exogenous events exist because of unknown external factors. By removing exogenous events from the dataset, TPP-Select can improve MTPP model training performance.

CHD differs from these works. CHD discloses causal relations between history and events observed later, while Granger causality (Idé et al., 2021; Wu et al., 2024) explores mutual relations across different marks to find which mark helps the event forecast on other marks. Other works (Li et al., 2020; Song et al., 2024) exploit logic rules between different events, *e.g.*, one event happens before another event, then construct the conditional intensity function based on these rules. In summary, the methods in these studies cannot solve CHD.

6 Conclusions

This study investigates Counterfactual History Distillation (CHD) to distill the essential events in history that can influence the occurrence of the subsequent events. This study demonstrates the issue of solving Counterfactual History Distillation (CHD) by conventional counterfactual analysis and refines the definition to ensure the distilled events are informative. With deliberate methods including Gumbel-softmax trick, the proposed solution MTPP-based Counterfactual History Distiller (MTPP-CHD) learns by effectively probing various event combinations. Its superiority has been observed in distillation optimization and processing speed in tests on real-world datasets. This study demonstrates analyzing the distilled events may disclose insights into the causal relation between events and event marks in continuous-time event sequences.

REFERENCES

Ehsan Abbasnejad, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. Counterfactual vision and language learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 10041–10051. IEEE, 2020. doi: 10.1109/CVPR42600.2020.01006.

Akanksha Atrey, Kaleigh Clary, and David D. Jensen. Exploratory not explanatory: Counterfactual analysis of saliency maps for deep reinforcement learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Yoshua Bengio, Nicholas LÃl'onard, and Aaron Courville. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation, 2013. arXiv:1308.3432 [cs].

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Renqin Cai, Xueying Bai, Zhenrui Wang, Yuling Shi, Parikshit Sondhi, and Hongning Wang. Modeling sequential online interactive behaviors with temporal point process. In Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (eds.), *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pp. 873–882. ACM, 2018. doi: 10.1145/3269206.3271782.

Ziheng Chen, Fabrizio Silvestri, Jia Wang, He Zhu, Hongshik Ahn, and Gabriele Tolomei. Relax: Reinforcement learning agent explainer for arbitrary predictive models. In Mohammad Al Hasan and Li Xiong (eds.), *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, volume abs/2110.11960, pp. 252–261. ACM, 2022. doi: 10.1145/3511808.3557429.

D. J. Daley and D. Vere-Jones (eds.). *An Introduction to the Theory of Point Processes Volume I: Elementary Theory and Methods*. Probability and its Applications. Springer, 2 edition, 2003. ISBN 978-0-387-21564-8.

Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Pai-Shun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 590–601, 2018.

Amit Dhurandhar, Tejaswini Pedapati, Avinash Balakrishnan, Pin-Yu Chen, Karthikeyan Shanmugam, and Ruchir Puri. Model Agnostic Contrastive Explanations for Structured Data, 2019.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (eds.), *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 1555–1564. ACM, 2016. doi: 10.1145/2939672.2939875.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. GLM: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 320–335. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.26.

Mehrdad Farajtabar, Jiachen Yang, Xiaojing Ye, Huan Xu, Rakshit Trivedi, Elias B. Khalil, Shuang Li, Le Song, and Hongyuan Zha. Fake news mitigation via point process based intervention. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1097–1106. PMLR, 2017.

Yukun Feng, Patrick Xia, Benjamin Van Durme, and João Sedoc. Automatic document selection for efficient encoder pretraining. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9522–9530. Association for Computational Linguistics, 2022.

Azin Ghazimatin, Oana Balalau, Rishiraj Saha Roy, and Gerhard Weikum. PRINCE: provider-side interpretability with counterfactual explanations in recommender systems. In James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang (eds.), *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pp. 196–204. ACM, 2020. doi: 10.1145/3336191.3371824.

Çaglar Hizli, St John, Anne Juuti, Tuure Saarinen, Kirsi Pietiläinen, and Pekka Marttinen. Temporal causal mediation through a point process: Direct and indirect effects of healthcare interventions. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/b7d9b1d4a9464d5d1ece82198e351349-Abstract-Conference.html.

Seyyed Abbas Hosseini, Keivan Alizadeh, Ali Khodadadi, Ali Arabzadeh, Mehrdad Farajtabar, Hongyuan Zha, and Hamid R. Rabiee. Recurrent poisson factorization for temporal recommendation. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pp. 847–855. ACM, 2017. doi: 10.1145/3097983.3098197.

Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. Reducing sentiment bias in language models via counterfactual evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 65–83. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.7.

Tsuyoshi Idé, Georgios Kollias, Dzung T. Phan, and Naoki Abe. Cardinality-regularized hawkes-granger model. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 2682–2694, 2021.

Amirkasra Jalaldoust, Katerina Hlavácková-Schindler, and Claudia Plant. Causal discovery in hawkes processes by minimum description length. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 6978–6987. AAAI Press, 2022.

Jianchao Ji, Zelong Li, Shuyuan Xu, Max Xiong, Juntao Tan, Yingqiang Ge, Hao Wang, and Yongfeng Zhang. Counterfactual Collaborative Reasoning. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, WSDM '23, pp. 249–257. Association for Computing Machinery, 2023. ISBN 978-1-4503-9407-9. doi: 10.1145/3539597.3570464.

Shalmali Joshi, Oluwasanmi Koyejo, Warut Vijitbenjaronk, Been Kim, and Joydeep Ghosh. Towards Realistic Individual Recourse and Actionable Explanations in Black-Box Decision Making Systems, 2019.

Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Hiroki Arimura. DACE: distribution-aware counterfactual explanation by mixed-integer linear optimization. In Christian Bessiere (ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pp. 2855–2862. ijcai.org, 2020. doi: 10.24963/ijcai.2020/395.

Richard M. Karp. Reducibility among Combinatorial Problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger (eds.), *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20âĂŞ22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*, The IBM Research Symposia Series, pp. 85–103. Springer US, Boston, MA, 1972. ISBN 978-1-4684-2001-2. doi: 10.1007/978-1-4684-2001-2_9.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 107–117. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1011.

Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

Jiahui Li, Kun Kuang, Baoxiang Wang, Furui Liu, Long Chen, Fei Wu, and Jun Xiao. Shapley counterfactual credits for multi-agent reinforcement learning. In Feida Zhu, Beng Chin Ooi, and Chunyan Miao (eds.), *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pp. 934–942. ACM, 2021a. doi: 10.1145/3447548.3467420.

Shuang Li, Lu Wang, Ruizhi Zhang, Xiaofu Chang, Xuqin Liu, Yao Xie, Yuan Qi, and Le Song. Temporal Logic Point Processes. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5990–6000. PMLR, 2020.

Shuang Li, Mingquan Feng, Lu Wang, Abdelmajid Essofi, Yufeng Cao, Junchi Yan, and Le Song. Explaining Point Processes by Learning Interpretable Temporal Logic Rules. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, September 2021b.

David Lüdke, Marin Bilos, Oleksandr Shchur, Marten Lienen, and Stephan Günnemann. Add and Thin: Diffusion for Temporal Point Processes. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, volume abs/2311.01139, 2023.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4765–4774, 2017.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Ricards Marcinkevics and Julia E. Vogt. Interpretable models for granger causality using self-explaining neural networks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (eds.), *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pp. 2243–2251. ACM, 2018. doi: 10.1145/3269206.3272027.

Hongyuan Mei and Jason Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6754–6764, 2017.

Hongyuan Mei, Chenghao Yang, and Jason Eisner. Transformer embeddings of irregularly spaced events and their participants. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Robert C. Moore and William Lewis. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pp. 220–224. Association for Computational Linguistics, 2010.

Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, pp. 607–617. Association for Computing Machinery, 2020. ISBN 978-1-4503-6936-7. doi: 10.1145/3351095.3372850.

Shanlei Mu, Yaliang Li, Wayne Xin Zhao, Jingyuan Wang, Bolin Ding, and Ji-Rong Wen. Alleviating Spurious Correlations in Knowledge-aware Recommendations through Counterfactual Generator. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, pp. 1401–1411. Association for Computing Machinery, 2022. ISBN 978-1-4503-8732-3. doi: 10.1145/3477495.3531934.

Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual VQA: A cause-effect look at language bias. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 12700–12710. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.01251.

Kimia Noorbakhsh and Manuel Gomez Rodriguez. Counterfactual Temporal Point Processes. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, Nov. 28-Dec. 9th, 2022, New Orleans, USA*, 2022.

Y. Ogata. On Lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31, January 1981. ISSN 1557-9654. doi: 10.1109/TIT.1981.1056305. Conference Name: IEEE Transactions on Information Theory.

Takahiro Omi, Naonori Ueda, and Kazuyuki Aihara. Fully neural network based model for general temporal point processes. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 2120–2129, 2019.

Axel Parmentier and Thibaut Vidal. Optimal counterfactual explanations in tree ensembles. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8422–8431. PMLR, 2021.

Chen Qian, Fuli Feng, Lijie Wen, Chunping Ma, and Pengjun Xie. Counterfactual inference for text classification debiasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5434–5445. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.422.

Goutham Ramakrishnan, Yun Chan Lee, and Aws Albarghouthi. Synthesizing action sequences for modifying model decisions. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 5462–5469. AAAI Press, 2020.

Oleksandr Shchur, Marin Bilos, and Stephan Günnemann. Intensity-free learning of temporal point processes. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Xiaoming Shi, Siqiao Xue, Kangrui Wang, Fan Zhou, James Y. Zhang, Jun Zhou, Chenhao Tan, and Hongyuan Mei. Language models can improve event prediction by few-shot abductive reasoning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/5e5fd18f863cbe6d8ae392a93fd271c9-Abstract-Conference.html.

Zitao Song, Chao Yang, Chaojie Wang, Bo An, and Shuang Li. Latent Logic Tree Extraction for Event Sequence Explanation from LLMs, 2024.

Juntao Tan, Shuyuan Xu, Yingqiang Ge, Yunqi Li, Xu Chen, and Yongfeng Zhang. Counterfactual Explainable Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pp. 1784–1793. Association for Computing Machinery, 2021. ISBN 978-1-4503-8446-9. doi: 10.1145/3459637.3482420.

Antonio Toral, Pavel Pecina, Longyue Wang, and Josef van Genabith. Linguistically-augmented perplexity-based data selection for language models. *Computer Speech & Language*, 32(1):11–26, 2015. ISSN 0885-2308. doi: 10.1016/j.csl.2014.10.002.

Khanh Hiep Tran, Azin Ghazimatin, and Rishiraj Saha Roy. Counterfactual Explanations for Neural Recommenders. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pp. 1627–1631. Association for Computing Machinery, 2021. ISBN 978-1-4503-8037-9. doi: 10.1145/3404835.3463005.

Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan Hines, John Dickerson, and Chirag Shah. Counterfactual explanations and algorithmic recourses for machine learning: A review. *ACM Computing Surveys*, abs/2010.10596, 2020. ISSN 0360-0300. doi: 10.1145/3677119.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR, 2017.

Hanqing Wang, Wei Liang, Jianbing Shen, Luc Van Gool, and Wenguan Wang. Counterfactual Cycle-Consistent Learning for Instruction Following and Generation in Vision-Language Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15471–15481, 2022a.

Wenjie Wang, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. User-controllable Recommendation Against Filter Bubbles. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, pp. 1251–1261. Association for Computing Machinery, 2022b. ISBN 978-1-4503-8732-3. doi: 10.1145/3477495.3532075.

Yue Wang, Yao Wan, Chenwei Zhang, Lu Bai, Lixin Cui, and Philip Yu. Competitive Multi-agent Deep Reinforcement Learning with Counterfactual Thinking. In *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 1366–1371, 2019. doi: 10.1109/ICDM.2019.00175. ISSN: 2374-8486.

Zhenlei Wang, Jingsen Zhang, Hongteng Xu, Xu Chen, Yongfeng Zhang, Wayne Xin Zhao, and Ji-Rong Wen. Counterfactual Data-Augmented Sequential Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pp. 347–356. Association for Computing Machinery, 2021. ISBN 978-1-4503-8037-9. doi: 10.1145/3404835.3462855.

Zifeng Wang, Xi Chen, Rui Wen, Shao-Lun Huang, Ercan E. Kuruoglu, and Yefeng Zheng. Information theoretic counterfactual learning from missing-not-at-random feedback. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Dongxia Wu, Tsuyoshi Ide, Georgios Kollias, Jiri Navratil, Aurelie Lozano, Naoki Abe, Yian Ma, and Rose Yu. Learning Granger Causality from Instance-wise Self-attentive Hawkes Processes. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, pp. 415–423. PMLR, 2024. ISSN: 2640-3498.

Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning granger causality for hawkes processes. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1717–1726. JMLR.org, 2016.

Shuyuan Xu, Yunqi Li, Shuchang Liu, Zuohui Fu, Yingqiang Ge, Xu Chen, and Yongfeng Zhang. Learning causal explanations for recommendation. In *The 1st International Workshop on Causality in Search and Recommendation*, 2021.

Mengyue Yang, Quanyu Dai, Zhenhua Dong, Xu Chen, Xiuqiang He, and Jun Wang. Top-N Recommendation with Counterfactual User Preference Simulation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pp. 2342–2351. Association for Computing Machinery, 2021. ISBN 978-1-4503-8446-9. doi: 10.1145/3459637.3482305.

Yang Yang, Chao Yang, Boyang Li, Yinghao Fu, and Shuang Li. Neuro-Symbolic Temporal Point Processes, 2024.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. GLM-130B: An Open Bilingual Pre-trained Model. In *11th International Conference on Learning Representations, ICLR 2023*, September 2022.

Ping Zhang, Rishabh K. Iyer, Ashish Tendulkar, Gaurav Aggarwal, and Abir De. Learning to select exogenous events for marked temporal point process. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 347–361, 2021a.

Qiang Zhang, Aldo Lipani, Ömer Kirnap, and Emine Yilmaz. Self-attentive hawkes process. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11183–11193. PMLR, 2020a.

Shichang Zhang, Jiani Zhang, Xiang Song, Soji Adeshina, Da Zheng, Christos Faloutsos, and Yizhou Sun. PaGE-Link: Path-based Graph Neural Network Explanation for Heterogeneous Link Prediction. In *Proceedings of the ACM Web Conference 2023*, WWW '23, pp. 3784–3793. Association for Computing Machinery, 2023a. ISBN 978-1-4503-9416-1. doi: 10.1145/3543507.3583511.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open Pre-trained Transformer Language Models. *CoRR*, abs/2205.01068, June 2022a. doi: 10.48550/ARXIV.2205.01068. arXiv:2205.01068 [cs].

Wei Zhang, Thomas Kobber Panum, Somesh Jha, Prasad Chalasani, and David Page. CAUSE: learning granger causality from event sequences using attribution methods. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11235–11245. PMLR, 2020b.

Yixuan Zhang, Quyu Kong, and Feng Zhou. Integration-free training for spatio-temporal multimodal covariate deep kernel point processes. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, volume abs/2310.05485, 2023b.

Yizhou Zhang, Karishma Sharma, and Yan Liu. Vigdet: Knowledge informed neural temporal point process for coordination detection on social media. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 3218–3231, 2021b.

Yizhou Zhang, Defu Cao, and Yan Liu. Counterfactual Neural Temporal Point Process for Estimating Causal Influence of Misinformation on Social Media. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, Nov. 28-Dec. 9th, 2022, New Orleans, USA*, 2022b.

Zhu Zhang, Zhou Zhao, Zhijie Lin, Jieming Zhu, and Xiuqiang He. Counterfactual contrastive learning for weakly-supervised vision-language grounding. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020c.

Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. SEISMIC: A Self-Exciting Point Process Model for Predicting Tweet Popularity. In Longbing Cao, Chengqi Zhang, Thorsten Joachims, Geoffrey I. Webb, Dragos D. Margineantu, and Graham Williams (eds.), *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pp. 1513–1522. ACM, 2015. doi: 10.1145/2783258.2783401.

Jinfeng Zhong and Elsa Negre. Shap-enhanced counterfactual explanations for recommendations. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, SAC '22, pp. 1365–1372. Association for Computing Machinery, 2022. ISBN 978-1-4503-8713-2. doi: 10.1145/3477314.3507029.

Hanhan Zhou, Tian Lan, and Vaneet Aggarwal. PAC: Assisted Value Factorization with Counterfactual Predictions in Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15757–15769, 2022.

Zihao Zhou and Rose Yu. Automatic integration for spatiotemporal neural point processes. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, volume abs/2310.06179, 2023.

Sujia Zhu, Yue Shen, Zihao Zhu, Wang Xia, Baofeng Chang, Ronghua Liang, and Guodao Sun. VAC2: Visual Analysis of Combined Causality in Event Sequences, 2022.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11692–11702. PMLR, 2020.

A Proofs

A.1 Proof of Proposition 1

Proof. CHD defined in Equation (7) has two constraints. For the first constraint, we have:

$$\log \mathrm{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_f)) \leqslant \log \mathrm{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_l)) + \log \epsilon_l \tag{14}$$

For the second constraint, we have:

$$\log \mathrm{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_f)) \geqslant \log \mathrm{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_d)) + \log \epsilon_d \tag{15}$$

By connecting Equation (14) and Equation (15), we get:

$$\frac{\mathrm{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_l))}{\mathrm{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_d))} \geqslant \frac{\epsilon_d}{\epsilon_l} \tag{16}$$

For any $\epsilon_l \in (0, 1)$ and $\epsilon_d \in (0, 1)$ where $\epsilon_d > \epsilon_l$, we can always move more events from $\boldsymbol{\mathcal{H}}_f$ to $\boldsymbol{\mathcal{H}}_d$ so that Equation (16) is satisfied. In the extreme case that $\frac{\epsilon_d}{\epsilon_l}$ is an any large number, all events in $\boldsymbol{\mathcal{H}}_f$ can be moved to $\boldsymbol{\mathcal{H}}_d$ so that $\boldsymbol{\mathcal{H}}_l = \varnothing$; then we have $\mathrm{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_l)) \to +\infty$ that can always guarantee the inequation in Equation (16) held. □

B Experiment Details

B.1 Datasets

Table 4 reports the basic information of three real-world datasets, Retweet, StackOverflow, and Yelp. Table 5 shows different settings of $|\boldsymbol{\mathcal{H}}_f|$ and $|\boldsymbol{x}_o|$ for the subsequences $(\boldsymbol{\mathcal{H}}_f, \boldsymbol{x}_o)$ in experiments. Table 6 reports the number of events in training, validation, and test datasets for different settings of $|\boldsymbol{\mathcal{H}}_f|$ and $|\boldsymbol{x}_o|$. Table 7 presents the hyperparameters used for training the MTPP-CHD model on Retweet, StackOverflow, and Yelp. Because generating $\boldsymbol{\mathcal{H}}_d$ and $\boldsymbol{\mathcal{H}}_l$ from $\boldsymbol{\mathcal{H}}_f$ runs faster on the CPU, we train and evaluate all CHD approaches on Xeon Gold 6132 CPUs instead of GPUs.

Retweet (Zhao et al., 2015) records when users retweet a particular message on Twitter. The mark of this dataset distinguishes all users into 3 different types. Mark 0 refers to the normal user, whose follower count is lower than the overall median. Mark 1 refers to the influential user, whose follower count is higher than the median but lower than the top-5% of the entire user base. Mark 2 refers to the famous user, whose follower count is in the top-5% of the entire user base.

StackOverflow (Leskovec & Krevl, 2014) was collected from Stackoverflow[3], a popular question-answering website about various topics. Users providing decent answers will receive different badges as rewards. We have 22 marks in this dataset, representing 22 different badges that users can receive for their answers.

Yelp[4] contains the reviews of restaurants, shopping centers, and stores in the US on Yelp. We categorize these reviews into three groups based on the reviewers. Mark 0 refers to the normal reviewer. The number of reviews a normal reviewer has is lower than the overall median, which is 5 reviews in our case. Mark 1 refers to the influential reviewers. These reviewers write more reviews than normal reviewers but less than the top-5% reviewers. Mark 2 refers to the famous reviewers, the top-5% reviewers who write more than 92 reviews.

B.2 MTPP Model

MTPP-CHD can work with any MTPP models that provide $p^*(m, t)$. Without loss of generality, this study uses FullyNN (Omi et al., 2019). Table 8 presents the hyperparameters used for training the FullyNN on Retweet, StackOverflow, and Yelp.

[3] https://stackoverflow.com
[4] https://www.yelp.com

Table 4: The basic information of datasets where the number of sequences, events, and marks are in the first three columns, $\bar{\tau}$ and $\sigma(\tau)$ are the mean and standard deviation of the time intervals between adjacent events, t_0 and T are the earliest start time and the latest end time of all sequences.

	Sequences	Events	Marks	$\bar{\tau}$	$\sigma(\tau)$	t_0	T
Retweet	24 000	2 610 102	3	2574	16 302	0	604 799
StackOverflow	6633	480 414	22	0.8747	1.2091	1324	1390
Yelp	4022	409 946	3	7.2644	13.410	0	751

Table 5: Settings of $|\mathcal{H}_f|$ and $|\boldsymbol{x}_o|$ in experiments for each dataset.

	(# of events in $\boldsymbol{x}_o$, # of events in $\mathcal{H}_f$)
Retweet	(10, 25), (10, 30), (10, 35), (15, 35), (20, 35)
StackOverflow	(15, 40), (15, 45), (15, 50), (20, 50), (25, 50)
Yelp	(10, 25), (10, 30), (10, 35), (15, 35), (20, 35)

Table 6: The number of events in training, validation, and test dataset for different setting of $|\mathcal{H}_f|$ and $|\boldsymbol{x}_o|$.

	$(\boldsymbol{x}_o,\mathcal{H}_f)$	training	validation	test
Retweet	(10, 25)	1 476 116	145 521	148 465
	(10, 30)	1 376 116	135 521	135 521
	(10, 35)	1 276 116	125 521	128 465
	(15, 35)	1 176 383	115 551	118 497
	(20, 35)	1 081 289	106 047	108 970
StackOverflow	(15, 40)	99 791	10 826	29 232
	(15, 45)	87 623	9451	25 824
	(15, 50)	77 341	8307	22 951
	(20, 50)	68 635	7350	20 504
	(25, 50)	61 254	6512	18 385
Yelp	(10, 25)	213 677	25 937	29 562
	(10, 30)	197 622	23 952	27 492
	(10, 35)	181 567	21 967	25 422
	(15, 35)	165 587	19 996	23 359
	(20, 35)	150 640	18 157	21 406

Table 7: Hyperparamters settings for training MTPP-CHD.

	Retweet	StackOverflow	Yelp
Training Steps	100 000	100 000	100 000
Warmup Steps	5000	5000	5000
Batch Size	256	128	128
Hidden Vector	64	64	64
Input Vector	32	32	32
Q, K, V	32	32	32
Head	4	4	4
N	4	4	4
M	4	4	4
Learning Rate	0.001	0.001	0.001
ϵ_l	0.5	0.5	0.6
ϵ_d	0.9	0.9	0.9
α	1.0	1.0	1.0
β	1.0	1.0	1.0

FullyNN estimates the integral of conditional intensity functions $\Lambda^*(m,t) = \int_{t_l}^{t} \lambda^*(m,\tau)d\tau$ and calculates the value of the intensity function at time t from the gradient of $\Lambda^*(m,t)$:

$$\Lambda^*(m,t) = \int_{t_l}^{t} \lambda^*(m,\tau)d\tau = \text{FullyNN}(m,t) \tag{17}$$

$$\lambda^*(m,t) = \frac{\partial \Lambda^*(m,t)}{\partial t} = \frac{\partial \text{FullyNN}(m,t)}{\partial t} \tag{18}$$

$$p^*(m,t) = \lambda^*(m,t)\exp\left(-\Lambda^*(t)\right) \tag{19}$$

$$= \frac{\partial \text{FullyNN}(m,t)}{\partial t}\exp\left(-\sum_{n\in\mathbb{M}} \text{FullyNN}(n,t)\right) \tag{20}$$

This helps FullyNN elude calculating $\Lambda^*(m,t)$ by numerical integration methods, such as Monte Carlo integration, to predict MTPP faster and more accurately. The FullyNN is trained on NVIDIA A100 GPUs.

Table 8: Hyperparamters settings for training MTPP Models.

	Retweet	StackOverflow	Yelp
Training Steps	400 000	200 000	200 000
Warmup Steps	80 000	40 000	40 000
Batch Size	32	32	32
History Embedding	32	32	32
Optimizer	AdamW	AdamW	AdamW
Intensity Vector	16	32	16
Learning Rate	0.002	0.002	0.002
Layers	4	2	4

C Additional Experiment Results

Additional experiment results in Section 4.2 are reported here.

C.1 Effectiveness of Counterfactual Analysis Refinement

Figure 5 demonstrates the distribution of $\log \text{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_l)) - \log \text{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_d))$ on StackOverflow, Retweet, and Yelp at various settings of $|\boldsymbol{\mathcal{H}}_f|$ and $|\boldsymbol{x}_o|$ using MTPP-CHD with and without refinement, respectively. The results further support the conclusion in Section 4.2.1 that the resultant $\boldsymbol{\mathcal{H}}_d$s have more information than the corresponding $\boldsymbol{\mathcal{H}}_l$s for predicting the following events $|\boldsymbol{x}_o|$ using MTPP-CHD with refinement. In contrast, MTPP-CHD without refinement may lead to the resultant $\boldsymbol{\mathcal{H}}_d$s having less information than the corresponding $\boldsymbol{\mathcal{H}}_l$s.

C.2 Effectiveness of L_e and L_n

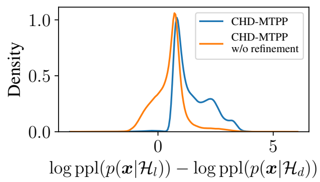
Section 4.2.3 demonstrate that minimizing loss L_e leads to $\boldsymbol{\mathcal{H}}_d$ with fewer events and minimizing loss L_n leads to $\boldsymbol{\mathcal{H}}_d$ with more events, respectively, on StackOverflow. The results on Retweet and Yelp are reported in Figure 6 and Figure 7, respectively. They are consistent with the results in Section 4.2.3.

C.3 Model Efficiency

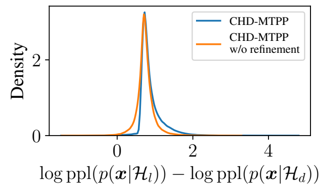
Table 9 presents the total time of the trained MTPP-CHD and baselines to solve CHD on three real-world datasets at more settings of $|\boldsymbol{\mathcal{H}}_f|$ and $|\boldsymbol{x}_o|$ in the first three columns, and the time used by MTPP-CHD for training on these datasets in the last column. The results futher demonstrate that MTPP-CHD solves CHD more efficiently than baselines.

C.4 Analysis Of Distilled Events

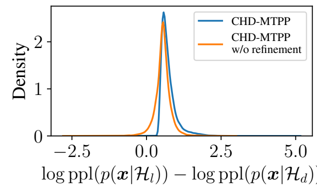
In Figure 8 and Figure 9, we present the percentage of different marks in $\boldsymbol{\mathcal{H}}_d$ returned by MTPP-CHD and RD on the test data of StackOverflow and Yelp. For StackOverflow, the results demonstrate some marks have more information but others have less information for predicting the following events. For Yelp, all marks seemingly have the similar information about $\boldsymbol{x}_o$.



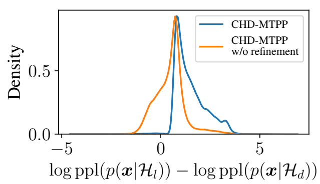
(a) StackOverflow ($|\mathbf{x}_o| = 15$, $|\mathcal{H}_f| = 45$)



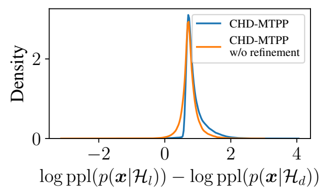
(b) Retweet ($|\mathbf{x}_o| = 10$, $|\mathcal{H}_f| = 30$)



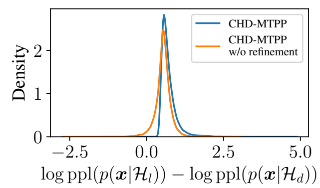
(c) Yelp ($|\mathbf{x}_o| = 10$, $|\mathcal{H}_f| = 30$)



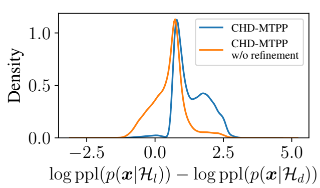
(d) StackOverflow ($|\mathbf{x}_o| = 15$, $|\mathcal{H}_f| = 50$)



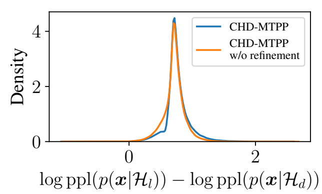
(e) Retweet ($|\mathbf{x}_o| = 10$, $|\mathcal{H}_f| = 35$)



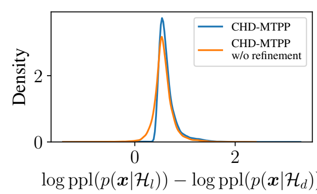
(f) Yelp ($|\mathbf{x}_o| = 10$, $|\mathcal{H}_f| = 35$)



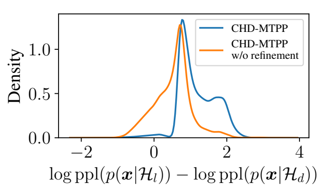
(g) StackOverflow ($|\mathbf{x}_o| = 20$, $|\mathcal{H}_f| = 50$)



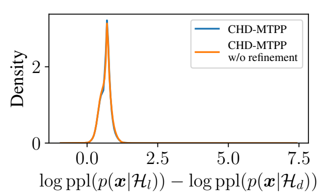
(h) Retweet ($|\mathbf{x}_o| = 15$, $|\mathcal{H}_f| = 35$)



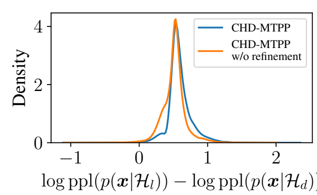
(i) Yelp ($|\mathbf{x}_o| = 15$, $|\mathcal{H}_f| = 35$)



(j) StackOverflow ($|\mathbf{x}_o| = 15$, $|\mathcal{H}_f| = 45$)

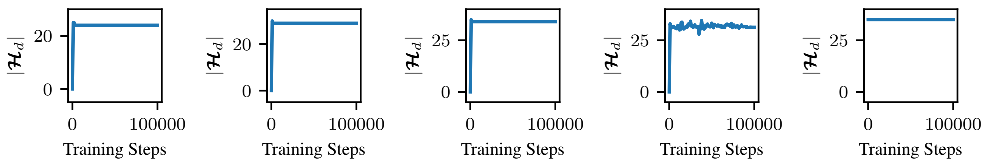


(k) Retweet ($|\mathbf{x}_o| = 20$, $|\mathcal{H}_f| = 35$)

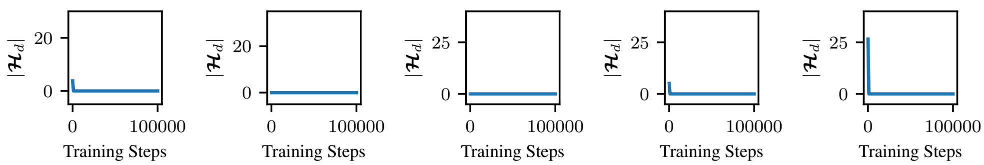


(l) Yelp ($|\mathbf{x}_o| = 20$, $|\mathcal{H}_f| = 35$)

Figure 5: The distribution of $\log \mathrm{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_l)) - \log \mathrm{ppl}(p(\boldsymbol{x}_o|\boldsymbol{\mathcal{H}}_d))$ of our MTPP-CHD and the MTPP-CHD without refinement.

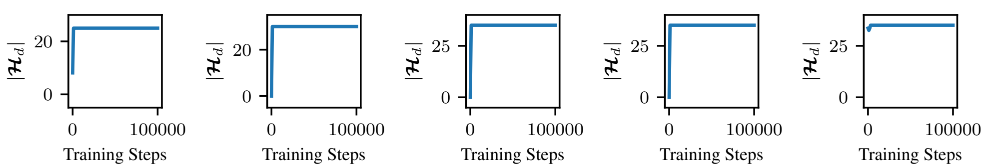


(a) The number of event in $\mathcal{H}_d$ returned by MTPP-CHD trained by minimizing L_e only.

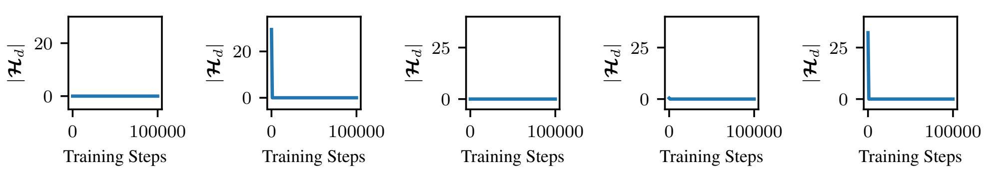


(b) The number of event in $\mathcal{H}_d$ returned by MTPP-CHD trained by minimizing L_n only.

Figure 6: Effectiveness of L_e and L_n on Retweet (from left to right: $(|\boldsymbol{x}_o|, |\boldsymbol{\mathcal{H}}_f|) = (15, 40)$, $(15, 45)$, $(15, 50)$, $(20, 50)$, $(25, 50)$).



(a) The number of event in $\mathcal{H}_d$ returned by MTPP-CHD trained by minimizing L_e only.



(b) The number of event in $\mathcal{H}_d$ returned by MTPP-CHD trained by minimizing L_n only.

Figure 7: Effectiveness of L_e and L_n on Yelp (from left to right: $(|\boldsymbol{x}_o|, |\boldsymbol{\mathcal{H}}_f|) = (15, 40)$, $(15, 45)$, $(15, 50)$, $(20, 50)$, $(25, 50)$).

Table 9: Total time used to solve all CHD tasks in test data (first three columns) and time used for MTPP-CHD training (last column).

	MTPP-CHD	GS	RD	MTPP-CHD(Training)
StackOverflow ($\lvert\mathbf{x}_o\rvert = 15$,$\lvert\mathcal{H}_f\rvert = 45$)	**2.83h**	26.3h	36.6h	24.0h
StackOverflow ($\lvert\mathbf{x}_o\rvert = 15$,$\lvert\mathcal{H}_f\rvert = 50$)	**2.86h**	26.7h	41.1h	25.3h
StackOverflow ($\lvert\mathbf{x}_o\rvert = 20$,$\lvert\mathcal{H}_f\rvert = 50$)	**2.58h**	24.0h	36.9h	29.5h
StackOverflow ($\lvert\mathbf{x}_o\rvert = 25$,$\lvert\mathcal{H}_f\rvert = 50$)	**2.35h**	21.7h	33.4h	31.8h
Retweet ($\lvert\mathbf{x}_o\rvert = 10$, $\lvert\mathcal{H}_f\rvert = 30$)	**9.93h**	93.5h	89.0h	29.0h
Retweet ($\lvert\mathbf{x}_o\rvert = 10$, $\lvert\mathcal{H}_f\rvert = 35$)	**10.8h**	102h	111h	33.4h
Retweet ($\lvert\mathbf{x}_o\rvert = 15$, $\lvert\mathcal{H}_f\rvert = 35$)	**10.1h**	93.8h	103h	37.5h
Retweet ($\lvert\mathbf{x}_o\rvert = 20$, $\lvert\mathcal{H}_f\rvert = 35$)	**9.38h**	87.5h	94.9h	36.8h
Yelp ($\lvert\mathbf{x}_o\rvert = 10$, $\lvert\mathcal{H}_f\rvert = 30$)	**2.01h**	18.8h	17.6h	14.8h
Yelp ($\lvert\mathbf{x}_o\rvert = 10$, $\lvert\mathcal{H}_f\rvert = 35$)	**2.16h**	20.3h	22.1h	15.4h
Yelp ($\lvert\mathbf{x}_o\rvert = 15$, $\lvert\mathcal{H}_f\rvert = 35$)	**2.00h**	18.7h	20.5h	15.5h
Yelp ($\lvert\mathbf{x}_o\rvert = 20$, $\lvert\mathcal{H}_f\rvert = 35$)	**1.87h**	17.2h	18.9h	15.2h

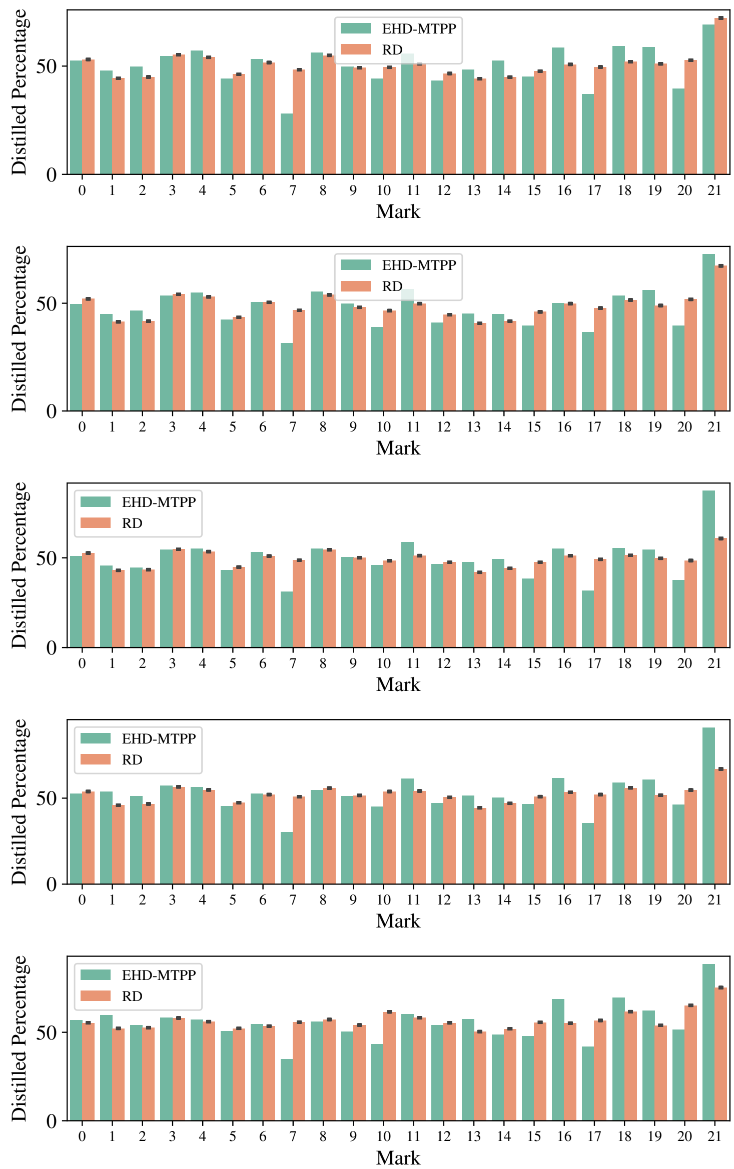


Figure 8: The percentage of events for different marks in $\mathcal{H}_d$ returned by MTPP-CHD and Random Distillation (RD) on test date of StackOverflow (from left to right: $(|\boldsymbol{x}_o|, |\mathcal{H}_f|) = (10, 25)$, $(10, 30)$, $(10, 35)$, $(15, 35)$, $(20, 35)$). The results pass the significance test with p-values smaller than $\alpha = 0.005$ for most marks.

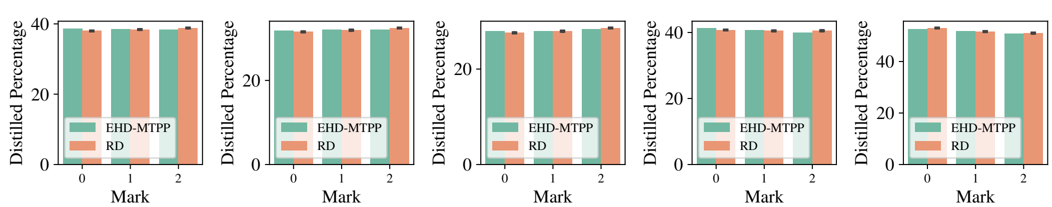


Figure 9: The percentage of events for different marks in $\mathcal{H}_d$ returned by MTPP-CHD and Random Distillation (RD) on test date of Yelp (from left to right: $(|\boldsymbol{x}_o|, |\mathcal{H}_f|) = (10, 25)$, $(10, 30)$, $(10, 35)$, $(15, 35)$, $(20, 35)$). The results pass the significance test with p-values smaller than $\alpha = 0.005$.