Unveiling Latent Causal Rules: A Temporal Point Process Approach for Abnormal Event Explanation

Yiling Kuang* **Chao Yang*** **Yang Yang** **Shuang Li†**
The Chinese University of Hong Kong, Shenzhen

Abstract

In high-stakes systems such as healthcare, it is critical to understand the causal reasons behind unusual events, such as sudden changes in patient's health. Unveiling the causal reasons helps with quick diagnoses and precise treatment planning. In this paper, we propose an automated method for uncovering "if-then" logic rules to explain observational events. We introduce *temporal point processes* to model the events of interest, and discover the set of latent rules to explain the occurrence of events. To achieve this, we employ an Expectation-Maximization (EM) algorithm. In the E-step, we calculate the likelihood of each event being explained by each discovered rule. In the M-step, we update both the rule set and model parameters to enhance the likelihood function's lower bound. Notably, we optimize the rule set in a *differential* manner. Our approach demonstrates accurate performance in both discovering rules and identifying root causes. We showcase its promising results using synthetic and real healthcare datasets.

1 Introduction

Detecting and understanding abnormal events is crucial in various fields. In healthcare, quickly spotting and diagnosing sudden patient deterioration can save lives. In e-commerce, detecting odd user behavior may stop fraud and ensure safe shopping experiences. In manufacturing, finding irregularities in production processes can make operations more efficient and reduce downtime. These examples highlight the importance of abnormal event detection for enhancing safety, security, and efficiency in various areas.

However, effectively implementing abnormal event detection comes with challenges. A major obstacle is the lack of labeled abnormal event data during training. Labeling events as normal or abnormal is often expensive and labor-intensive. Consider the context of credit card transactions. When the system encounters an unusually large transaction, determining its underlying rationale can be intricate. It may indeed be a legitimate and routine transaction, or it could be indicative of fraudulent activity. Identifying the true reasons require thorough investigation, which is both time and resource consuming. In our paper, we address this issue by introducing a *rule-based probabilistic* approach for predicting and explaining events. We treat event's explanatory reasons as hidden variables and use an EM algorithm to learn rules that explain events without explicit label information regarding anomaly. For each observed event, the triggered rules will provide insight into the most probable causal factors and determine whether the event should be flagged as abnormal or normal.

We introduce *temporal point process* to model events of interest in continuous time. The *time-to-event* is treated as a random variable characterized by the *intensity* function, indicating the occurrence rate of the event in real time. We formulate the intensity function as a *mixture model*, where each component is defined by an "if-then" rule factor. We assume that the "if" condition should occur before the "then" part, and the "if" condition also specifies the temporal order constraints that the relevant logic variables must meet, such as "if X_1 and X_2 are true and X_1 happens before X_2, then Y is true." These explanatory temporal logic rules will be discovered from data, which *compete* to influence when an event occurs. When an event happens, the specific rule that is triggered based on preceding events will provide an explanation for its occurrence.

As mentioned above, we formulate an EM algorithm to simultaneously learn the rule set and temporal point

*Equal Contribution. †Corresponding Author. Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

processes model parameters. In the E-step, we calculate the assignment probability of each event to its causal rules based on our current estimations of the model parameters and the rule set. In the M-step, we update the rule set and model parameters to maximize the expected likelihood calculated in E-step.

Notably, *learning the rule set* poses a formidable challenge due to its inherently *combinatorial* nature. It requires selecting the proper subsets of logic variables and then considering their temporal order relations to create a logic rule. This variable selection process is both time-consuming and non-differentiable. In our paper, we propose to solve a relaxed continuous optimization problem by directly modeling the logic variable selection probability in forming a rule. Specifically, each rule is generated by *a reparameterized K-subset sampling with the Gumbel noise injected to make the rule sampling process differentiable* [Xie and Ermon, 2019]. We further use a padding trick to make the generated rules with various lengths not exceeding K. In the M-step, both the rule set and the model parameters can be learned end-to-end in a differentiable way.

Our contributions can be summarized as follows.

(*i*) We leverage the continuous-time temporal point process modeling framework and formulate the intensity function as a rule-informed mixture model. Our rule-based probabilistic model is inherently interpretable, ensuring safe and reliable event prediction and explanation.

(*ii*) We design an EM learning algorithm that can accommodate the latent logical reasons for each event. Our overall learning algorithm is efficient and differentiable, which can automatically discover the set of explanatory logic rules from the population data and determine the most likely logical reason for each event.

(*iii*) We conduct empirical evaluations of our algorithm using both synthetic and real datasets, demonstrating its strong performance in rule discovery and root cause determination.

2 Related Work

We will compare our method with some existing works from the following aspects.

Temporal Point Process (TPP) Models TPP models provide an elegant tool for modeling the time intervals or time-to-event in continuous time. Current research in this area primarily focuses on enhancing the flexibility of intensity functions. For example, Du et al. [2016] introduced the RMTPP, a neural point process model that employs a Recurrent Neural Network to model the intensity function. Mei and Eisner [2017] further improved RMTPP by developing a continuous-time RNN. Zuo et al. [2020] and Zhang et al. [2020] leveraged the self-attention mechanism to capture the long-term dependencies of events. However, relying on flexible black-box models, which can excel at event prediction with sufficient training data, presents challenges. These models lack interpretability, making it difficult to understand the occurrence of events. In scenarios like abnormal event detection, the objective goes beyond merely predicting unusual events; it also involves provding causal explanations for these occurrences. The absence of interpretability renders black-box models unsuitable for high-stakes systems where transparency is critical.

The contemporary AI perspective underscores the value of crafting intrinsically interpretable models over relying solely on explaining black-box models [Rudin, 2019]. Along this line of research, recently, Li et al. [2020, 2021] proposed to design a rule-informed intensity function, trying to add the interoperability. However, their approaches lack the capability to provide detailed logical explanations for individual events, in contrast to our method. Additionally, their rule learning process is non-differentiable, relying on a branch and bound algorithm.

Rule Set Mining Rule set discovery in an unsupervised fashion has long been a classic topic. Traditional approaches include Itemset Mining Methods like Apriori [Agrawal et al., 1994] and NEclatcloesed [Aryabarzan and Minaei-Bidgoli, 2021], which target frequent itemset identification without accounting for event order. Sequential Pattern Mining methods, typified by CM-SPADE [Fournier-Viger et al., 2014a] and VGEN [Fournier-Viger et al., 2014b], aim to unveil temporal relationships but they lack precision and cannot handle timestamp information. In addition to the unsupervised methods, Inductive Logic Programming (ILP) [Srinivasan, 2001] provides a powerful supervised rule learning method, yet requiring balanced positive and negative examples to have a good performance. In contrast, our method leverages fine-grained temporal information to learn temporal logic rules from data and in an unsupervised manner.

3 Problem Setup

We define a set of logic variables (or predicates) as $\mathcal{X}$, where each variable $X_u \in \mathcal{X}$ is a boolean variable. Our target event, such as a sudden change in patient's health or an unusual large transaction, is denoted as Y and $Y \in \mathcal{X}$. We add a temporal dimension to the logic variables and each grounded logic variable becomes a sequence of spiked events, denoted as $\{X_u(t)\}_{t\geq 0}$, where each $X_u(t) \in \{0, 1\}$, $\forall t$. Specifically,

$X_u(t)$ jumps to 1 (i.e., true) at the time step when the event happens. In our problem, each data sample is a $|\mathcal{X}|$-dimensional multivariate temporal processes, denoted as $\mathcal{H}_t = \{X_u(t)\}_{u=1,\dots,|\mathcal{X}|}$.

We are interested in modeling and providing insight into the occurrence of $\{Y(t)\}_{t\geq 0}$. We treat the time intervals as random variables and the duration until event Y happens is characterized by the *conditional intensity function*, denoted as $\lambda(t \mid \mathcal{H}_{t-})$ where $t-$ means up to t but not include t. By definition, $\lambda(t \mid \mathcal{H}_{t-})dt = \mathbb{E}[N([t, t+dt]) \mid \mathcal{H}_{t-}]$, where $N([t, t+dt])$ denotes the number of events occurring in the interval $[t, t+dt)$. Given the occurrence time of event Y, i.e., $(t_1, \dots, t_n)$, the joint likelihood function is given by

$$p(t_1, \dots, t_n) = \prod_{i=1}^{n} p^*(t_i), \quad \text{with}$$
$$p^*(t_i) = \lambda^*(t_i) \exp\left(-\int_{t_{i-1}}^{t_i} \lambda^*(s)ds\right). \tag{1}$$

Here, to simplify the notation, we denote $p^*(t) := p(t \mid \mathcal{H}_{t-})$ and $\lambda^*(t) := \lambda(t \mid \mathcal{H}_{t-})$. In this paper, we will model $\lambda^*(t)$ using the rule-based features, through which we aim to understand the underlying logical reasons to explain the occurrence of target events. Specifically, our goals include:

(*i*) Discover the rule set $\mathcal{F} := \{f_1, f_2, ..., f_H\}$ from the entire events $\mathcal{H}_{t_n}$ up to t_n, with each rule having a general form

$$f: \quad Y \leftarrow \left(\bigwedge_{X_u \in \mathcal{X}_f} X_u\right) \bigwedge \left(\bigwedge_{X_u, X_v \in \mathcal{X}_f} r_j(X_u, X_v)\right) \tag{2}$$

where $\mathcal{X}_f$ is the set of body predicates of rule f. $r_j(X_u, X_v)$ is a set of temporal relations between pairs of predicates and can take any of the three types, i.e., "Before", "Equal", and "After", and can also be none, meaning there are no temporal order constants existing for X_u and X_v.

(*ii*) Infer the assignment of each occurred target event $\{Y(t_1), \dots, Y(t_n)\}$ to the following two groups:

- spontaneous events (i.e., cannot be explained by any logic rule or it's meaningless to be explained);
- can be explained by one logic rule from the discovered rule set $\mathcal{F} = \{f_1, f_2, ..., f_H\}$.

4 Model

We first describe how to construct the intensity function for Y based on the logic-informed features. For a general rule as defined in Eq. (2), we can construct the rule-informed feature as

$$\phi_f(\mathcal{H}_t) = \prod_{X_u \in \mathcal{X}_f} X_u(t_u) \cdot \prod_{X_u, X_v \in \mathcal{X}_f} r_j(X_u, X_v) \in \{0, 1\} \tag{3}$$

where we need to check from $\mathcal{H}_t$ whether each $X_u \in \mathcal{X}_f$ has been once grounded as true and whether the temporal relation grounded by their times holds (for multiple events we choose the last one). The temporal relation $r_j(X_u, X_v)$ is grounded according to its type,"Before", "Equal", or "After" (denoted as R_b, R_e, R_a respectively) by the event times as follows,

$$R_b(t_u, t_v) = \mathbb{1}\{t_u - t_v < -\delta\}$$
$$R_e(t_u, t_v) = \mathbb{1}\{|t_u - t_v| < \delta\}$$
$$R_a(t_u, t_v) = \mathbb{1}\{t_u - t_v > \delta\}$$

where δ (can be 0) is specified as the time tolerance. If the body condition of f is grounded to be true given history, then the feature $\phi_f(\mathcal{H}_t)$ is 1, otherwise it is 0. A better illustration can be found in Fig. 1.

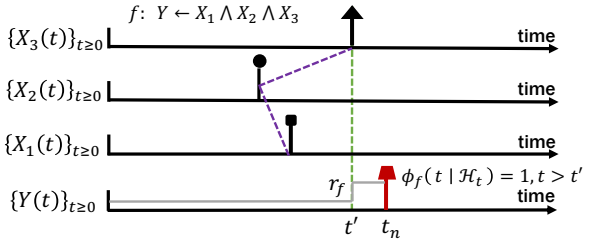

Figure 1: Illustration of the feature construction. For the logic rule: $f: Y \leftarrow X_1 \bigwedge X_2 \bigwedge X_3$, whenever the body condition becomes trues, the rule gets fired, as a result the feature ϕ_f becomes 1, and the intensity function of the head predicate is boosted by r_f.

Given the rule-informed feature defined in Eq. (3), we can build our intensity model for $\{Y(t)\}_{t\geq 0}$ as follows. For the i-th event in the dataset, by considering all the rules $\mathcal{F} = \{f_1, f_2, ..., f_H\}$, we model the intensity function as a *mixture-of-components* form,

$$\lambda^*(t_i \mid \boldsymbol{z}_i) = b_0 + \sum_{h=1,\dots,H} \gamma_h z_{ih} \phi_{f_h}(\mathcal{H}_{t_i}) \tag{4}$$

where b_0 is the base term, h is the index of the rule, and $\gamma_h > 0$ denotes the impact weight of each rule f_h. The *latent* variable $\boldsymbol{z}_i = [z_{ih}]_{h=1,\dots,H}$ is a *one-hot* vector. When $z_{ih} = 1$, it means event i is due to the h-th rule factor, which provides logical reasons to explain its occurrence. We assume the categorical $\boldsymbol{z}_i$ has a distribution $\boldsymbol{\pi} = [\pi_h]$, which indicates the probability of a rule appearing in the population and is our learnable parameter.

In the next section, we design an EM algorithm, which aims to jointly learn the rule set and point process model parameters from data, and simultaneously perform the inference of each $\boldsymbol{z}_i$.

5 Learning: EM Algorithm

To accommodate the *latent* rule assignment $\boldsymbol{z}_i$ for each event, it is natural to use an EM learning algorithm.

(*i*) In the E-step, the *posterior probability* over $z_i = [z_{ih}]_{h=0,\dots,H}$ for each event Y^i is computed. We use z_{i0} to indicate that *the event cannot be explained by any discovered rule.*

(*ii*) In the M-step, both the rule set and the point process model parameters are updated by maximizing the *expected likelihood* found in E-step. Denote all the learnable model parameters as

$$\theta = [b_0, [\gamma_h]_{h\in[1,\dots,H]}, \boldsymbol{\pi}, \mathcal{F}].$$

Notably, we devise a series of techniques in EM to enhance the stability of the optimization process. The E-step has been successfully resolved in a closed-form solution. The M-step is more involved, for which we employ three-fold approximations and reparameterization to facilitate the end-to-end differentiable training.

Complete Data Likelihood

To simplify the derivation, let's start with the i-th event $Y(t_i)$ and consider its complete data likelihood. The following derivations can be easily extended to the entire event sequence.

For an occurred event $Y(t_i)$ at time t_i, given the rule index $\boldsymbol{z}_i = [z_{ih}]_{h=0,\dots,H}$, we can write the *likelihood* for the *complete data* $\{(t_i, \boldsymbol{z}_i)\}$ as

$$\begin{aligned}
&p^*_\theta(t_i, \boldsymbol{z}_i) = p^*_\theta(t_i \mid \boldsymbol{z}_i)p(\boldsymbol{z}_i), \quad \text{where} \\
&p(z_{ih} = 1) = \pi_h, \quad \text{with} \quad 0 \le \pi_h \le 1, \sum_{h=0}^{H} \pi_h = 1 \\
&p^*_\theta(t_i \mid z_{ih} = 1) \\
&= \lambda^*_\theta(t_i \mid z_{ih} = 1) \cdot \exp\left(-\int_{t_{i-1}}^{t_i} \lambda^*_\theta(s \mid z_{ih} = 1)ds\right). \quad (5)
\end{aligned}$$

The complete data log-likelihood $\log p^*_\theta(t_i, \boldsymbol{z}_i)$ is easier to optimize compared to the log-likelihood of data, computed as

$$\ell(\theta) = \log\left[\sum_{\boldsymbol{z}_i} p^*_\theta(t_i, \boldsymbol{z}_i)\right] \quad (6)$$

which requires the marginalization of the latent variable $\boldsymbol{z}_i$.

5.1 E-step: Compute Posterior $Q^{new}(\boldsymbol{z}_i)$

To derive the lower bound of the log-likelihood, let's denote the a distribution for $\boldsymbol{z}_i = [z_{ih}]_{h=1,\dots,H}$ as $Q(\boldsymbol{z}_i)$ and $Q(\boldsymbol{z}_i) \ge 0$ and $\sum_{\boldsymbol{z}_i} Q(\boldsymbol{z}_i) = 1$. The above constraints make $Q(\boldsymbol{z}_i)$ a valid probability mass function.

Now we can create a lower bound for log-likelihood function

$$\ell(\theta) = \log \mathbb{E}_{Q(\boldsymbol{z}_i)}\left[\frac{p^*_\theta(t_i, \boldsymbol{z}_i)}{Q(\boldsymbol{z}_i)}\right] \ge \mathbb{E}_{Q(\boldsymbol{z}_i)}\left[\log \frac{p^*_\theta(t_i, \boldsymbol{z}_i)}{Q(\boldsymbol{z}_i)}\right]$$

where the inequality is due to the Jensen's inequality.

In the current iteration of the EM algorithm, suppose we aim to find $Q^{new}(\boldsymbol{z}_i)$, so that the lower bound equals to log-likelihood $\ell\left(\theta^{old}\right)$. According to Jensen's inequality, the equality between the lower bound and $\ell\left(\theta^{old}\right)$ holds if $\frac{p^*_{\theta^{old}}(t_i, \boldsymbol{z}_i)}{Q^{new}(\boldsymbol{z}_i)}$ is a constant. In other words, $Q^{new}(\boldsymbol{z}_i)$ needs to take the posterior of $\boldsymbol{z}_i$ to make the lower bound tight, i.e.,

$$\begin{aligned}
\textbf{E-step:} \quad Q^{new}(z_{ih} = 1) &= \frac{p^*_{\theta^{old}}(t_i, z_{ih} = 1)}{\sum_{h'=0}^{H} p^*_{\theta^{old}}(t_i, z_{ih'} = 1)} \\
&= \frac{\pi_h^{old} p^*_{\theta^{old}}(t_i \mid z_{ih} = 1)}{\sum_{h'=0}^{H} \pi_{h'}^{old} p^*_{\theta^{old}}(t_i \mid z_{ih'} = 1)} \quad (7)
\end{aligned}$$

with a more detailed calculation can be found in Eq. (5).

5.2 M-step: Optimize θ^{new}

For the M-step, we aim to update the model parameters θ (including the rule set $\mathcal{F}$) to maximize the expected log-likelihood computed in the E-step. which is a lower bound of the log-likelihood, i.e.,

$$\theta^{new} := \arg\max_\theta \underbrace{\mathbb{E}_{Q^{new}(\boldsymbol{z}_i)}\left[\log \frac{p^*_\theta(t_i, \boldsymbol{z}_i)}{Q^{new}(\boldsymbol{z}_i)}\right]}_{\text{“energy”}}$$

which is equal to solve (since $Q^{new}(\boldsymbol{z}_i)$ does not depend on θ)

$$\theta^{new} := \arg\max_\theta \mathbb{E}_{Q^{new}(\boldsymbol{z}_i)}\left[\log p^*_\theta(t_i, \boldsymbol{z}_i)\right] \quad (8)$$

where $\log p^*_\theta(t_i, \boldsymbol{z}_i)$ is the complete data log-likelihood. We further partition θ as $\theta := [\theta_0, \mathcal{F}]$ where θ_0 contains the model parameters excluding $\mathcal{F}$. Then we rewrite the above formulation as

$$\max_{\theta_0} \max_{\mathcal{F}} \mathbb{E}_{Q^{new}(\boldsymbol{z}_i)}\left[\log p^*_{\theta_0, \mathcal{F}}(t_i, \boldsymbol{z}_i)\right] \quad (9)$$

and we will alternate the optimization of θ_0 and $\mathcal{F}$.

5.2.1 Optimization wrt $\mathcal{F}$

We first discuss how to update $\mathcal{F}$ given current θ_0. We aim to propose a differentiable method to learn the rule set. To achieve this goal, we will employ three-fold approximations or reparameterization:

(*i*) **Continuous Relaxation of the Rule Learning:** We formulate the rule learning problem as a logic variable selection problem and encode the selection results into a binary matrix $A \in \{0, 1\}^{H\times(|\mathcal{X}|+M)}$. We then perform the continuous relaxations and relax A to the selection probability $\tilde{A} \in [0, 1]^{H\times(|\mathcal{X}|+M)}$.

(*ii*) **Continuous Approximation of the Boolean Logic Features:** We employ a logic-informed boolean feature approximation to ensure that the log-likelihood becomes a deterministic function of parameter $\tilde{A}$.

(*iii*) **Differentiable Top-K Subset Sampling:** We perform a differentiable top-K subset sampling with Gumbel noise injected to learn each rule. The learnable parameters are now becoming an unbounded weight matrix $W \in (R^+)^{H\times(|\mathcal{X}|+M)}$. We leverage simulated annealing by tuning the temperature, which balances the approximation errors and issues caused by gradient vanishing in training.

In summary, we have performed the following series of approximations:

$$\underbrace{\mathcal{F}}_{\text{rule set}} \rightarrow \underbrace{A}_{\text{binary}} \rightarrow \underbrace{\tilde{A}}_{\text{selection probability}} \rightarrow \underbrace{W}_{\text{unbounded weight}} \tag{10}$$

and in the end we optimize W end-to-end in a differentiable way. We provide a more detailed explanation as follows.

I: Continuous Relaxation of the Rule Learning

We first formulate the rule learning as a variable selection problem and introduce a binary matrix A to encode the rule content (with columns indicating the logic variables and rows indicating the rules):

$$A = \begin{array}{c} \\ f_1 \\ \vdots \\ f_H \end{array} \begin{array}{c} \begin{array}{ccccccc} X_1 & X_2 & \dots & X_{|\mathcal{X}|} & \xi_1 & \dots & \xi_M \end{array} \\ \begin{pmatrix} 0 & 1 & \dots & 1 & 0 & \dots & 1 \\ \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ 1 & 0 & \dots & 1 & 1 & \dots & 0 \end{pmatrix} \end{array}. \tag{11}$$

To accommodate for various rule lengths and meanwhile ensure that the rule length dones not exceed K, we append M (a hyperparameter) dummy columns after $X_{|\mathcal{X}|}$, indicating *empty-predicates* to adjust for various rule lengths. We let $\sum_{j=1}^{|\mathcal{X}|+M} a_{hj} = K$, and whenever $a_{hj} = 1$ for $j > |\mathcal{X}|$, it means that the rule length, determined by counting the valid predicates, will be less than K. A better illustration can be found in Fig. 2. Learning such A involves challenging discrete optimization. Instead, we will relax the problem to learning the selection probability matrix $\tilde{A}$ with $0 \le \tilde{a}_{hj} \le 1$. Given the selected predicates encoded in A, we can further identify their temporal relations. For each paired X_u and X_v, we introduce a learnable parameter $\boldsymbol{\alpha}$ to denote their corresponding probabilities

$$\boldsymbol{\alpha} = [\alpha_b, \alpha_e, \alpha_a, \alpha_n]$$

where $\alpha_b+\alpha_e+\alpha_a+\alpha_n = 1$ and $0 \le \alpha_b, \alpha_e, \alpha_a, \alpha_n \le 1$. That is, $\boldsymbol{\alpha}$ belongs to a 3-d simplex. We soften the

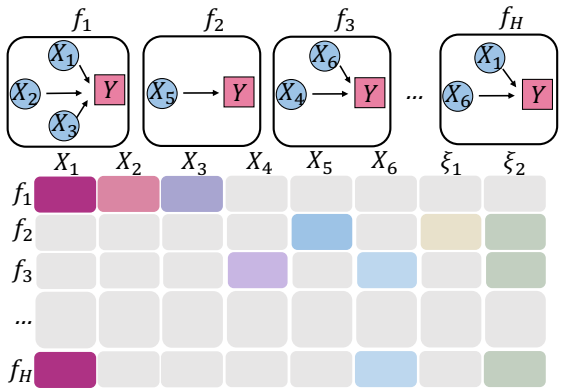

Figure 2: Illustration of rule content. Top diagrams indicate H learned rules given the bottom binary matrix A.

relation $r_i(X_u, X_v)$ by replacing it with

$$\begin{aligned}&\text{softmax}\left\{\alpha_b R_b\left(t_u, t_v\right), \alpha_e R_e\left(t_u, t_v\right), \alpha_a R_a\left(t_u, t_v\right),\right.\\ &\left.\alpha_n\left(1-\alpha_b R_b\left(t_u, t_v\right)-\alpha_e R_e\left(t_u, t_v\right)-\alpha_a R_a\left(t_u, t_v\right)\right)\right\}.\end{aligned} \tag{12}$$

For each rule, we consider the temporal relations between all possible pairs of selected predicates. If a rule involves multiple combinations of two distinct predicates, we employ the softmin operation on these r_i values. Using softmin helps prevent excessive value reduction resulting from multiple multiplications. The softmin value for a rule h is denoted as $\mathcal{T}_h, h = 1, \dots, H$. The combination of softmax and softmin operations ensures that the probability learning process remains differentiable. We represent the selection probabilities for all temporal relations in a rule as $\boldsymbol{\alpha_h}$, and these learned values determine the types of temporal relations.

II: Continuous Approximation of the Boolean Logic Features

To learn the rule set in a differentiable way, in the M-step we propose an approximation to the boolean logic feature as shown in Eq. (3) and rewrite the intensity function as

$$\lambda^*_{\text{soft}}(t_i \mid \boldsymbol{z}_i) = b_0 + \sum_{h=1,\dots,H} \gamma_h z_{ih} \mathcal{T}_h \mathcal{K}_{Lap}\left\{\sum_{j=1}^{|\mathcal{X}|} X_j(t_j)\tilde{a}_{hj}, K\right\}. \tag{13}$$

We use Laplace kernel denoted as $\mathcal{K}_{Lap}$ with center K to reparameterize the boolean feature and make it differentiable with $\tilde{A}$. As previously mentioned, $\mathcal{T}_h$ provides a softened approximation for temporal relations. This approach transforms the complete data log-likelihood into a deterministic function of $\tilde{A}$, enabling the gradient computation through automatic differentiation.

III: Differentiable Top-K Subset Sampling

Given the above description, now the rule learning process can be regarded as a top-K subset sampling process, which is governed by the selection probability $\tilde{A}$. We further use the reparameterization trick [Xie and Ermon, 2019] for this top-K subset sampling.

Assume the top-K subset sampling can be parameterized by an unbounded weight matrix $W = [w_1; w_2; \ldots; w_H] \in (R^+)^{H\times(|\mathcal{X}|+M)}$, with each element non-negative. The top-K relaxed subset sampling (using Gumbel noise) is summarized in Algorithm 1 (Appendix Section 1). By injecting the Gumbel noise, the introduced key $\hat{r}_j$ is differentiable with respect to the input weight w_h. We can use a top-K relaxation based on successive applications of the softmax function [Plötz and Roth, 2018] to obtain the relaxed top-K vector $\tilde{a}_h$.

To optimize sampling matrix W, we have

$$\textbf{M-step:} \quad \frac{\partial \ell(\theta_0, W)}{\partial w_h} = \frac{\ell(\theta_0, W)}{\partial \tilde{a}_h} \frac{\partial \tilde{a}_h}{\partial \hat{\gamma}_h} \frac{\partial \hat{\gamma}_h}{\partial w_h}. \tag{14}$$

The parameter W is updated with gradient descent.

With this reparameterization trick, the level of discreteness of A can be controlled by the temperature parameter τ. Thus, in our algorithm, an almost discrete matrix A is used to select rules, and weight matrix W is updated in every iteration to sample A. During the learning process, we employed cycled simulated annealing to fine-tune the temperature parameter τ. The choice of τ balances the trade-off between approximation errors and gradient vanishing issues. Each time when we iterate all data, τ anneals from a large value to a small non-zero one. Rule content A is learned via unsupervised learning without labels, and the number of rules H and rule length range can be tuned via cross-validation or predefined with domain knowledge.

5.2.2 Optimization wrt θ_0

M-step: We optimize other model parameters by taking gradient of $\ell(\theta_0, W)$ wrt $[b_0, [r_h]_{h\in[1,\ldots,H]}, [\boldsymbol{\alpha_h}]_{h\in[1,\ldots,H]}, \boldsymbol{\pi}]$, and set these gradients to 0. We need to make sure that these model parameters are non-negative. This can be guaranteed by projected gradient descent or using the change-of-variable trick (e.g., parameterize b_0 as $\exp(b_0)$).

Specifically, solving $\boldsymbol{\pi}$ in the M-step has a closed-form solution. Given the constraints $\sum_{h=0}^{H} \pi_h = 1$ we have

$$\textbf{M-step:} \quad \pi_h = \frac{n_h}{n}, \quad n_h = \sum_{i=1}^{n} Q^{new}(z_{ih} = 1). \tag{15}$$

6 Experiments

6.1 Synthetic Data Experiments

We assess our model's learning accuracy in terms of rule discovery, rule weight estimation, and rule prior distribution learning across various scenarios.

I: Experiment Setup In our empirical analysis, we systematically explore various aspects to evaluate the scalability and performance of our method: (*i*) **Diverse Ground Truth Rules:** We vary the number of ground truth rules, ranging from 1 to 4, to examine the model's performance across different complexity levels. (*ii*) **Variation in Predicate Set:** We consider five scenarios with different size of predicate set, including 10, 15, 20, 25, and 30, to assess the scalability of our method. A larger logic variable set represents a more challenging problem. (*iii*) **Variation in Data Size:** We conduct experiments with varying data sizes, including sequences of 5000, 10,000, 20,000, 30,000, and 40,000 instances, to investigate the EM algorithm's scalability under different data scales.

It is important to note that only a small subset of the predicates is included in the ground truth rules. In other words, the majority of the candidate logic variables are spurious variables. We aim to demonstrate that our method excels in pinpointing the most critical logic variables during the rule learning process. We present the primary results based on experiments with 20,000 sequences for each scenario. For additional details and comprehensive findings, please refer to the Appendix (Section 2).

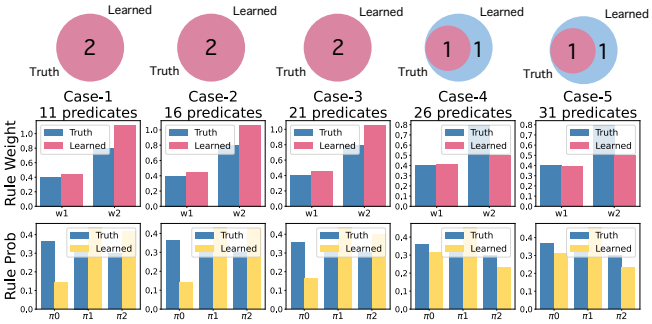

Figure 3: Our proposed model's performance is evaluated across all five scenarios for group-2, with two ground truth rules. We evaluate our model's performance in terms of rule discovery, rule weight learning, and rule prior distribution learning. The color "blue" indicates ground truth rules, weights, and prior distributions, whereas the colors "red" and "yellow" indicate the learning results.

II: Accuracy We evaluate the accuracy of our method in terms of rule set discovery and point process

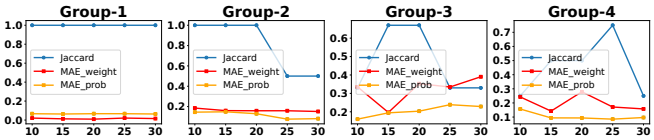

Figure 4: Jaccard similarity score and MAE of rule weights and rule preferential probabilities for all 4 groups. X-axis indicates the predicate library size and Y-axis indicates the value of Jaccard similarity and MAE.

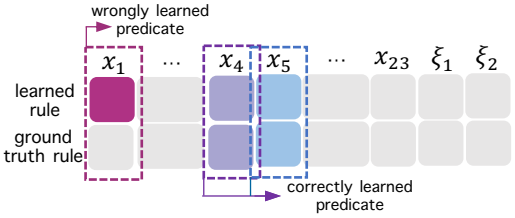

Figure 5: One example of erroneously uncovered rule for Group-3 (3 ground truth rules) Case-4 (23 to-be-searched predicates). Ground truth rule: $Y \leftarrow x_4 \wedge x_5 \wedge (x_4 \text{ Before } x_5)$. Learned rule: $Y \leftarrow x_1 \wedge x_4 \wedge x_5 \wedge (\ x_4 \text{ Before } x_5)$. We see that only one more predicate was wrongly excluded.

model parameter learning. In particular, we assess its accuracy in increasingly challenging scenarios by gradually enhancing the number of spurious logic variables in the predicate set.

In Fig. 8, we present the results for group-2, while the complete results are available in the Appendix (Section 2). The top row of plots employs Venn diagrams to visually depict the agreement between the actual rule set and the learned rule set, quantified through the Jaccard similarity score (the intersection area divided by the union area). A rule is deemed correct when both its content and temporal relation are accurately learned. Our proposed model consistently identifies almost all the ground truth rules across all groups.

The middle row diagrams compare the true rule weights with the learned rule weights, and the bottom row diagrams compare the true rule prior distributions with the learned rule distributions. Once the ground truth rules are accurately learned, the learned prior rule distributions closely align with the true values, resulting in low mean absolute errors (MAEs). The MAE is calculated treating weights and priors as vectors for each case.

In addition, we also compared our method with several classic unsupervised rule mining methods, including Apriori [Agrawal et al., 1994], NEclat-cloesed [Aryabarzan and Minaei-Bidgoli, 2021], CM-SPADE [Fournier-Viger et al., 2014a] and VGEN [Fournier-Viger et al., 2014b]. These baseline methods identify rules using the frequency thresholds, resulting in coarse rule discovery. From the experiment results, these approaches lead to the extraction of a vast number of noisy and erroneous rules, while the few accurate rules tend to remain hidden within this extensive rule set. A complete results and discussion can be found in Appendix (Section 4 and 6).

III: Scalability Fig. 4 illustrates the Jaccard similarity scores and MAE for all groups as the number of redundant predicates increases. Across all four groups, we observe a decrease in Jaccard similarity score and a slight increase in MAE with the growing number of predicates. However, it's important to note that our model maintains stable and reliable when the size of the predicate set is appropriate (i.e., around 20-30) and an adequate event sequences are available.

Furthermore, our approach exhibits promising results in identifying the causal rules for each event, inferred by the posterior of the latent rule index. Tab. 1 presents the accuracy of the posterior probabilities for each rule. Notably, in scenarios where almost all ground truth rules were successfully learned, the posterior accuracy is exceptionally high. However, in more challenging tasks, such as those in Group-3 and Group-4, where a larger number of ground truth rules are involved, the accuracy of the rule index posterior experiences a slight decrease.

In fact, our proposed model didn't deviate significantly in the case of erroneously uncovered rules. Fig. 5 provides an example of a learned wrong rule from Group-3, Case-4. In this instance, the discovered inaccurate rule includes only one additional redundant predicate. Similar patterns are observed in other cases as well.

Case \ Group	Group-1	Group-2	Group-3	Group-4
1 (10preds)	0.8036	0.7794	0.4011	0.3807
2 (15preds)	0.7962	0.7831	0.3925	0.3941
3 (20preds)	0.7979	0.7717	0.3783	0.3700
4 (25preds)	0.7986	0.6821	0.3130	0.3647
5 (30preds)	0.8039	0.6789	0.3261	0.2781

Table 1: Rule index inference accuracy on synthetic data for all 4 groups.

IV: Event Prediction We compared our model with several state-of-the-art baselines in terms of event predictions, using MAE as the evaluation metric for event time prediction. A complete descriptions of these baseline methods can be found in Appendix Section 4. As shown in Tab. 2, our model outperforms all baselines, particularly in complex scenarios like case-5 with 30 predicates across all groups. In other cases, the MAE varies depending on rule learning outcomes, but it consistently matches or surpasses the baselines.

Method \ Group	Group-1	Group-2	Group-3	Group-4
THP[Zuo et al., 2020]	2.2155	2.0089	2.2364	2.4121
RMTPP[Du et al., 2016]	2.2354	2.1843	2.3147	2.4523
ERPP[Xiao et al., 2017]	2.3532	2.0175	2.3564	2.4264
GCH[Xu et al., 2016]	2.5233	2.3754	2.4655	2.6427
LG-NPP[Zhang et al., 2021]	2.4145	2.4732	2.6564	2.5612
GM-NLF[Eichler et al., 2017]	2.3147	2.3622	2.4522	2.4362
TELLER[Li et al., 2021]	2.2281	2.4812	2.5121	2.4030
CLNN[Yan et al., 2023]	2.3858	2.5165	2.4972	2.5096
OURS*	**2.2075**	**1.8586**	**2.1957**	**2.3559**

Table 2: Event time prediction MAE on synthetic data for case-5 (30 preds) of all 4 groups.

6.2 Healthcare Data Experiments

MIMIC-IV[1] is a dataset of electronic health records for patients who have been admitted to the intensive care unit (ICU) [Johnson et al., 2023]. In our study, we focused on 4074 patients diagnosed with sepsis [Saria, 2018], as sepsis is a leading cause of mortality in the ICU, particularly when it progresses to septic shock. Septic shocks are critical medical emergencies, and timely recognition and treatment are crucial for improving survival rates. The objective of this experiment is to identify logic rules associated with septic shocks for both the general population and specific patients. These rules could serve as potential early alarms when abnormal indicators are detected, aiding in timely intervention.

Weight	Rule
0.566	Rule 1: LowUrine ← Arterial Blood Pressure Diastolic
0.467	Rule 2: LowUrine ← Inoized Calcium
0.527	Rule 3: LowUrine ← Arterial CO2 Pressure
0.466	Rule 4: LowUrine ← Venous 02 Pressure
0.464	Rule 5: LowUrine ← Respiratory Rate ∧ Hemoglobin
0.664	Rule 6: LowUrine ← Heart Rate ∧ BUN ∧ WBC ∧ (Heart Rate Before BUN) ∧ (BUN Before WBC)

Table 3: Some examples of our discovered temporal logic rules in MIMIC-IV.

Routine Vital Signs and Lab values A total of 29 variables associated with sepsis were extracted, including routine vital signs and laboratory values, as suggested and utilized by Komorowski et al. [2018]. We recorded the time points at which these variables first exhibited abnormal values within the 48-hour period preceding the onset of abnormal urine output (i.e., our target event).

Discovered Logic Rules Real-time urine output was utilized as the health state indicator, as low urine output directly indicates a deteriorate circulatory system and serves as a warning sign for septic shock. Due to the frequent fluctuations in urine output within the ICU setting, only instances where urine output becomes abnormal after maintaining a normal level for at least 48 hours were considered valid target events (i.e., meaningful to predict and explain).

In Tab. 3, we present a portion of the discovered explanatory logic rules and their corresponding learned weights based on our method. These displayed rules have been justified by doctors.

Rule 1 highlights the significance of abnormal arterial blood pressure diastolic as a crucial indicator for septic shock, with a substantial rule weight of 0.566. Therefore, patients should closely pay high attention to their arterial blood pressure diastolic and promptly seek clinical support if it deviates from the normal range.

Rule 2, on the other hand, noted by doctors, may be less relevant as fluctuations in calcium homeostasis can occur in specific patients during sepsis. Rule 3 and Rule 4 highlight the individual significance of abnormal arterial CO_2 pressure and venous O_2 pressure as additional early indicators of a compromised circulatory system. These rules bear relatively significant weights of 0.527 and 0.466, respectively.

Rule 5 and Rule 6 demonstrate the combined effects on low urine output. Rule 5, with a weight of 0.464, suggests that when the respiratory rate of patients exceeds 20 breaths per minute and there is a sharp decline in hemoglobin levels, they should be highly vigilant for the possibility of septic shock. Furthermore, Rule 6, which has the highest weight of 0.664, indicates that if the heart rate of patients surpasses 90 beats per minute, blood urea nitrogen (BUN) and white blood cell count (WBC) fall outside the normal ranges, and all three signals occur successively, immediate emergency medical treatment should be sought.

Our methodology incorporated invaluable input from medical experts who thoroughly reviewed our discovered rule results. Their feedback consistently validated the soundness and relevance of our findings. Additionally, we bolstered our discovered rules by referencing and citing supporting evidence from reputable medical sources, with discussions found in Appendix Section 8. We also compared our method with the post-hoc method [Crabbé and Van Der Schaar, 2021] in terms of interpretability, with the corresponding results in

[1] https://mimic.mit.edu/

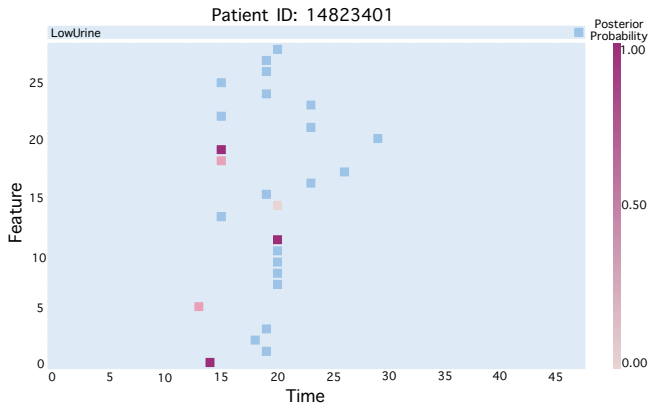

Figure 6: Rule identification results for patient 14823401. X-axis: time ranges from 0 to 48 h. Y-axis: feature id (29 in total). Low urine output occurred at 48h. Deep blue blocks record the occurrence time of a routine vital sign or abnormal lab value. Learned rule with largest posterior probability (colored in purple): LowUrine ← `Heart Rate` ∧ `BUN` ∧ `WBC`, `Heart Rate Before BUN`, `BUN Before WBC`. Other rules with relatively large posterior probability are colored in pink and off white. The darker the color, the greater the posterior probability.

Appendix Section 5.

Logic Rule Identification for Each Event Our method can identify the most likely causal rule specific to individual patients. For instance, let's consider patient with ID:14823401 as an example and the results are visualized in Fig. 6. Based on the analysis, the rule with the highest posterior probability that the patient is likely to adhere to is Rule 6: LowUrine ← `Heart Rate` ∧ `BUN` ∧ `WBC`. The identified rule will help with the individual healthcare delivery and treatment design.

Event Prediction We used the same baselines as in the synthetic data experiments, with the MAE serving as the evaluation metric for predicting the occurrence time of "Low Urine" events. Our model's performance is presented in Tab. 4, where it outperforms all the baselines.

Method	LowUrine
THP	2.4234
RMTPP	2.4643
ERPP	2.6122
GCH	2.5367
LG-NPP	2.5672
GM-NLF	2.6925
TELLER	2.4401
CLNN	2.4371
OURS*	**2.3675**

Table 4: Event time prediction MAE (unit/hour) on healthcare data.

7 Conclusion

In this paper, we leverage the temporal point process models to effectively discover latent logic rules to predict and explain abnormal events. We design an EM algorithm that can jointly learn model parameters, discover rule set, and infer the most likely rule factor for each event. We especially can learn the rule set in a differentiable way. Our EM-based algorithm may be adapted to survival analysis by modifying the complete likelihood accordingly and modeling the hazard function rather than the hazard ratio. The framework readily accommodates repeated events, as joint likelihood of a long sequence with repeated events aligns with multiple survival processes. Our method has exhibited strong performance, as demonstrated through experiments with synthetic and real healthcare datasets.

Acknowledgements

References

Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499. Santiago, Chile, 1994.

Djillali Annane, Eric Bellissant, and Jean-Marc Cavaillon. Septic shock. *The Lancet*, 365(9453):63–78, 2005.

Nader Aryabarzan and Behrouz Minaei-Bidgoli. Neclat-closed: A vertical algorithm for mining frequent closed itemsets. *Expert Systems with Applications*, 174:114738, 2021.

Sean M Bagshaw, Derek R Townsend, and Robert C McDermid. Disorders of sodium and water balance in hospitalized patients. *Canadian Journal of Anesthesia/Journal canadien d'anesthésie*, 56(2):151–167, 2009.

Samir Benchekroune, Peter CJ Karpati, Christine Berton, Cédric Nathan, Joaquim Mateo, Mansour Chaara, Florence Riché, Marie-Josèphe Laisné, Didier Payen, and Alexandre Mebazaa. Diastolic arterial blood pressure: a reliable early predictor of survival in human septic shock. *Journal of Trauma and Acute Care Surgery*, 64(5):1188–1195, 2008.

Jonathan Crabbé and Mihaela Van Der Schaar. Explaining time series predictions with dynamic masks. In *International Conference on Machine Learning*, pages 2166–2177. PMLR, 2021.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD international conference on knowl-*

edge discovery and data mining, pages 1555–1564, 2016.

Raymond Durkin, Mary Ann Gergits, James F Reed III, and John Fitzgibbons. The relationship between the arteriovenous carbon dioxide gradient and cardiac index. *Journal of critical care*, 8(4):217–221, 1993.

Michael Eichler, Rainer Dahlhaus, and Johannes Dueck. Graphical modeling for multivariate hawkes processes with nonparametric link functions. *Journal of Time Series Analysis*, 38(2):225–242, 2017.

Philippe Fournier-Viger, Antonio Gomariz, Manuel Campos, and Rincy Thomas. Fast vertical mining of sequential patterns using co-occurrence information. In *Advances in Knowledge Discovery and Data Mining: 18th Pacific-Asia Conference, PAKDD 2014, Tainan, Taiwan, May 13-16, 2014. Proceedings, Part I 18*, pages 40–52. Springer, 2014a.

Philippe Fournier-Viger, Antonio Gomariz, Michal Šebek, and Martin Hlosta. Vgen: fast vertical mining of sequential generator patterns. In *Data Warehousing and Knowledge Discovery: 16th International Conference, DaWaK 2014, Munich, Germany, September 2-4, 2014. Proceedings 16*, pages 476–488. Springer, 2014b.

Philippe Fournier-Viger, Ted Gueniche, Souleymane Zida, and Vincent S Tseng. Erminer: sequential rule mining using equivalence classes. In *Advances in Intelligent Data Analysis XIII: 13th International Symposium, IDA 2014, Leuven, Belgium, October 30–November 1, 2014. Proceedings 13*, pages 108–119. Springer, 2014c.

G Gernardin, C Pradier, F Tiger, P Deloffre, and M Mattei. Blood pressure and arterial lactate level are early indicators of short-term survival in human septic shock. *Intensive care medicine*, 22:17–25, 1996.

Hernando Gomez, Can Ince, Daniel De Backer, Peter Pickkers, Didier Payen, John Hotchkiss, and John A Kellum. A unified theory of sepsis-induced acute kidney injury: inflammation, microcirculatory dysfunction, bioenergetics and the tubular cell adaptation to injury. *Shock (Augusta, Ga.)*, 41(1):3, 2014.

Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Benjamin Moody, Brian Gow, Li-wei H Lehman, et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific Data*, 10(1):1–9, 2023.

Matthieu Komorowski, Leo A Celi, Omar Badawi, Anthony C Gordon, and A Aldo Faisal. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature medicine*, 24 (11):1716–1720, 2018.

C Langenberg, L Wan, M Egi, CN May, and R Bellomo. Renal blood flow in experimental septic acute renal failure. *Kidney international*, 69(11):1996–2002, 2006.

Mitchell M Levy, Mitchell P Fink, John C Marshall, Edward Abraham, Derek Angus, Deborah Cook, Jonathan Cohen, Steven M Opal, Jean-Louis Vincent, and Graham Ramsay. International sepsis definitions conference, 2003. 2001 SCCM/ESICM/ACCP/ATS/SIS International Sepsis Definitions Conference . . . , 2001.

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

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 *International Conference on Learning Representations*, 2021.

Jihad Mallat, Florent Pepy, Malcolm Lemyze, Gaelle Gasan, Nicolas Vangrunderbeeck, Laurent Tronchon, Benoit Vallet, and Didier Thevenin. Central venous-to-arterial carbon dioxide partial pressure difference in early resuscitation from septic shock: a prospective observational study. *European Journal of Anaesthesiology— EJA*, 31(7):371–380, 2014.

Carter E Mecher, Eric C Rackow, Mark E Astiz, and Max Harry Weil. Venous hypercarbia associated with severe sepsis and systemic hypoperfusion. *Critical care medicine*, 18(6):585–589, 1990.

Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems*, 30, 2017.

Barry A Mizock. Alterations in carbohydrate metabolism during stress: a review of the literature. *The American journal of medicine*, 98(1):75–84, 1995.

Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. *Advances in Neural information processing systems*, 31, 2018.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

Suchi Saria. Individualized sepsis treatment using reinforcement learning. *Nature medicine*, 24(11):1641–1642, 2018.

Mervyn Singer, Clifford S Deutschman, Christopher Warren Seymour, Manu Shankar-Hari, Djillali Annane, Michael Bauer, Rinaldo Bellomo, Gordon R Bernard, Jean-Daniel Chiche, Craig M Coopersmith, et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). *Jama*, 315(8): 801–810, 2016.

Ashwin Srinivasan. The aleph manual, 2001.

Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen Chu. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Sang Michael Xie and Stefano Ermon. Reparameterizable subset sampling via continuous relaxations. *arXiv preprint arXiv:1901.10517*, 2019.

Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning granger causality for hawkes processes. In *International conference on machine learning*, pages 1717–1726. PMLR, 2016.

Ruixuan Yan, Yunshi Wen, Debarun Bhattacharjya, Ronny Luss, Tengfei Ma, Achille Fokoue, and Anak Agung Julius. Weighted clock logic point process. In *The Eleventh International Conference on Learning Representations*, 2023.

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

Qiang Zhang, Aldo Lipani, and Emine Yilmaz. Learning neural point processes with latent graphs. In *Proceedings of the Web Conference 2021*, pages 1495–1505, 2021.

Julie R Zivin, Theodore Gooley, Richard A Zager, and Michael J Ryan. Hypocalcemia: a pervasive metabolic abnormality in the critically ill. *American journal of kidney diseases*, 37(4):689–698, 2001.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International conference on machine learning*, pages 11692–11702. PMLR, 2020.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [No]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [No]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

8 Appendix

Appendix Overview

In the following, we will provide supplementary materials to better illustrate our methods and experiments.

- Section 8.1 presents the algorithm box of Top-K Subset Sampling.
- Section 8.2 provides detailed analysis for synthetic data experiments, including experiment setup, implementation details, results analysis, and model scalability analysis.
- Section 8.3 provides the information of computing infrastructure for both synthetic data experiments and real-world data experiments.
- Section 8.4 introduces the baseline methods we considered in this paper.
- Section 8.5 provides the comparison of our method with a SOTA post-hoc method.
- Section 8.6 provides the detailed comparison results compared with other rule mining baselines.
- Section 8.7 describes the details of MIMIC-IV dataset preprocessing and risk factors extracting.
- Section 8.8 provides doctor verification and medical references towards our experiment results using MIMIC-IV datasets.

8.1 Algorithm of Top-K Subset Sampling

Algorithm 1: Relaxed Subset Sampling

Input: Weight vector $w_h = [w_{h1}, w_{h2}, \ldots, w_{hn}]$, subset size K, temperature $\tau > 0$
Output: Relaxed K-hot vector $a_h = [a_{h1}, \ldots, a_{hn}]$, where $\sum_{j=1}^{n} a_{hj} = K, 0 \leq a_{hj} \leq 1$

$\hat{\mathbf{r}} \leftarrow [\ \ ]$
for $j \leftarrow 1$ to n **do**
  $u_j \leftarrow \text{Uniform}(0, 1)$
  $\hat{r}_j \leftarrow -\log(-\log(u_j)) + \log(w_{hj})$
  $\hat{r}.\text{append}(\hat{r}_j)$
end for
$a_h \leftarrow \text{RelaxedTopK}(\hat{\mathbf{r}}, K, \tau)$

8.2 Detailed Analysis for Synthetic Data Experiments

Setup In our empirical analysis, we systematically explore various aspects to evaluate the scalability and performance of our method: **(i) Diverse Ground Truth Rules:** We vary the number of ground truth rules, ranging from 1 to 4 (4 groups in total), to examine the model's performance across different complexity levels. **(ii) Variation in Predicate Set:** We consider five cases with different size of predicate set, including 10, 15, 20, 25, and 30, to assess the scalability of our method. A larger logic variable set represents a more challenging problem. **(iii) Variation in Data Size:** We conduct experiments with varying data sizes, including sequences of 5000, 10,000, 20,000, 30,000, and 40,000 instances, to investigate our proposed EM algorithm's scalability under different data scales.

We listed all the ground truth rules and discovered rules for all groups in Tab. 5 using 20000 sequences for each case. In Fig. 7-10, we report the discovery accuracy of the rule set , the learning accuracy of the rule weight parameters and rule prior (i.e., the appearance prior probability of each rule in population) distribution parameters, using 20000 sequences for each case. In Fig. 11, we report how the rule learning accuracy and the computation time change over different predicate sets and sample size.

Implementation Details The E-step, which refers to inference, has a closed-form.The M-step is more involved. In M-step, we learn model parameters, including the predicate assignment to rules, the temporal relationships among the selected predicates, the impact weight of each rule, and the prior distribution of the rules, using coordinate descent. In other words, we optimize one component of the model parameters by holding the others.

The most critical aspect of this process involves selecting the predicates and determining the temporal relationships within a rule. To achieve a stable optimization, we set a small learning rate, specifically 0.0001, to prevent rapid changes in predicate selection. Additionally, to ensure comprehensive exploration of the rule set and fine-tuning of predicate weights, we cyclically anneal the temperature parameter, denoted as τ in Alg. 1, during each iteration. In other words, as we iterate through the data, τ transitions from a large value to a small, non-zero temperature. As for the learning rate pertaining to temporal relations, we set it slightly higher at 0.0035 to facilitate the rapid convergence of these relations.

Results Analysis Across all four groups of 20000 sequences, we observe that an increase in the number of redundant predicates within the same group leads to a slight decrease in rule learning accuracy, as well as reduced accuracy in rule weights and probabilities. In different groups, the presence of more underlying ground truth logic rules poses a greater challenge in accurately learning these rules in synthetic datasets. Therefore, when the number of ground truth logic rules increases while the number of redundant predicates remains constant, both rule learning accuracy and the accuracy of rule weights and probabilities decrease. However, despite the decline in experimental performance that occurs when the task is challenging, our model remains stable and reliable when the size of the predicate set is appropriate (i.e., around 20-30) and there are sufficient event sequences available.

It is crucial to emphasize that our reported final model parameter accuracy relies on the rule set discovery accuracy. If our algorithm learns an incorrect logic rule, we will compute the accuracy of the corresponding rule prior distribution and rule weight parameter as 0, which is very harsh.

In fact, our mined imprecise rules didn't deviate significantly from the ground truth rules. Illustrated in Tab. 5, the discovered inaccurate rules only add or miss a small number of redundant predicates, with the majority parts of the true rule content successfully learned.

Scalability of Our Method The scalability results are illustrated in Fig. 11, where we report how the rule learning accuracy and the computation time change over different predicate sets and sample size. As the sample sizes increase, the accuracy of learned temporal logic rules improves, but at the same time, the required training time also increases. When the number of ground truth temporal logic rules increases, there is a slight decrease in learning accuracy, but it remains within an acceptable range. From this result, we conclude that our method scales fairly well with the growing number of predicates and sample sizes.

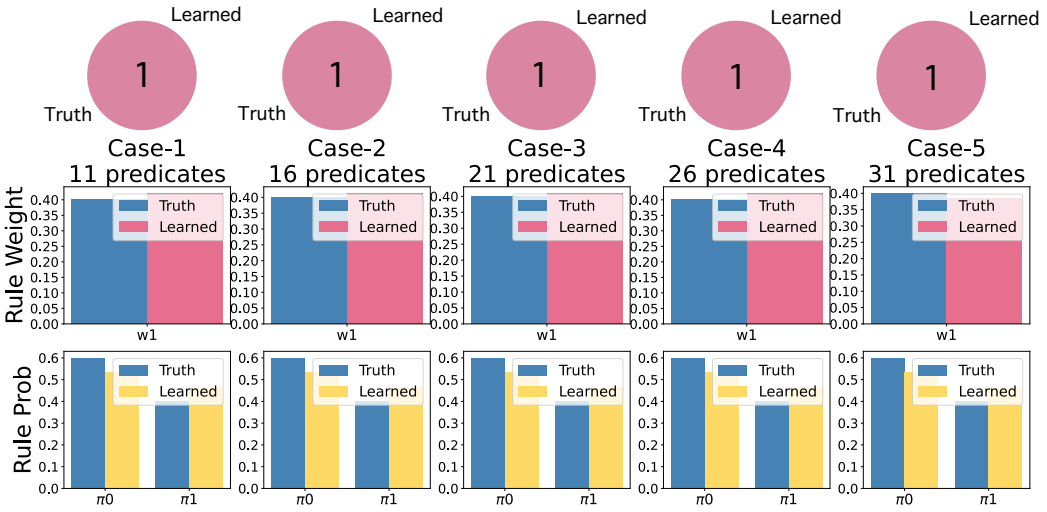

Figure 7: Rule discovery ability, rule weight learning accuracy, and the accuracy of learning the probability of a rule appearing in the population of our proposed model on all 5 cases for group-1 (1 ground truth rules) using 20000 sequences. Blue one indicates ground truth rule/ground truth rule weight/ground truth probability of a rule appearing in the population, and red/yellow one indicates learned rule weight/learned probability of a rule appearing in the population.

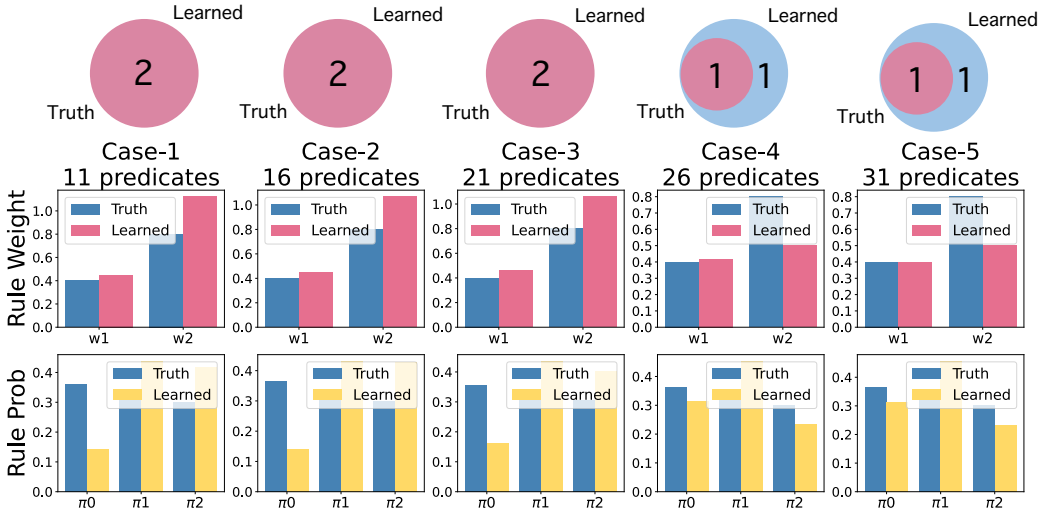

Figure 8: Learning results for group-2 (2 ground truth rules).

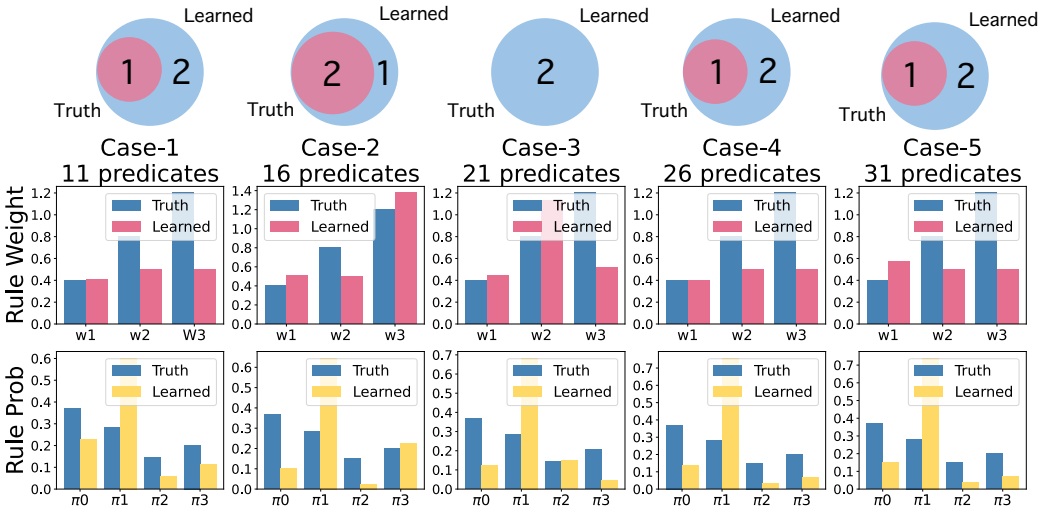

Figure 9: Learning results for group-3 (3 ground truth rules).

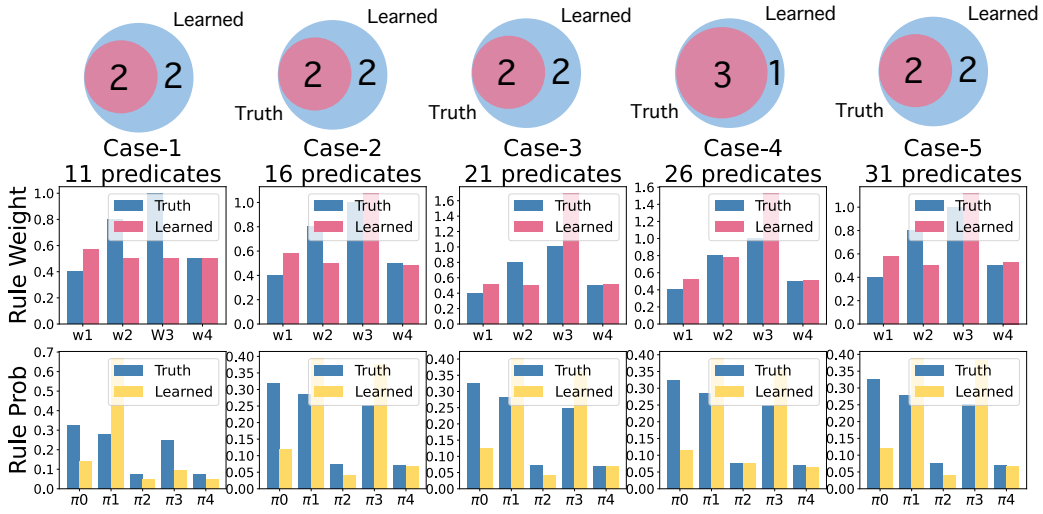

Figure 10: Learning results for group-4 (4 ground truth rules).

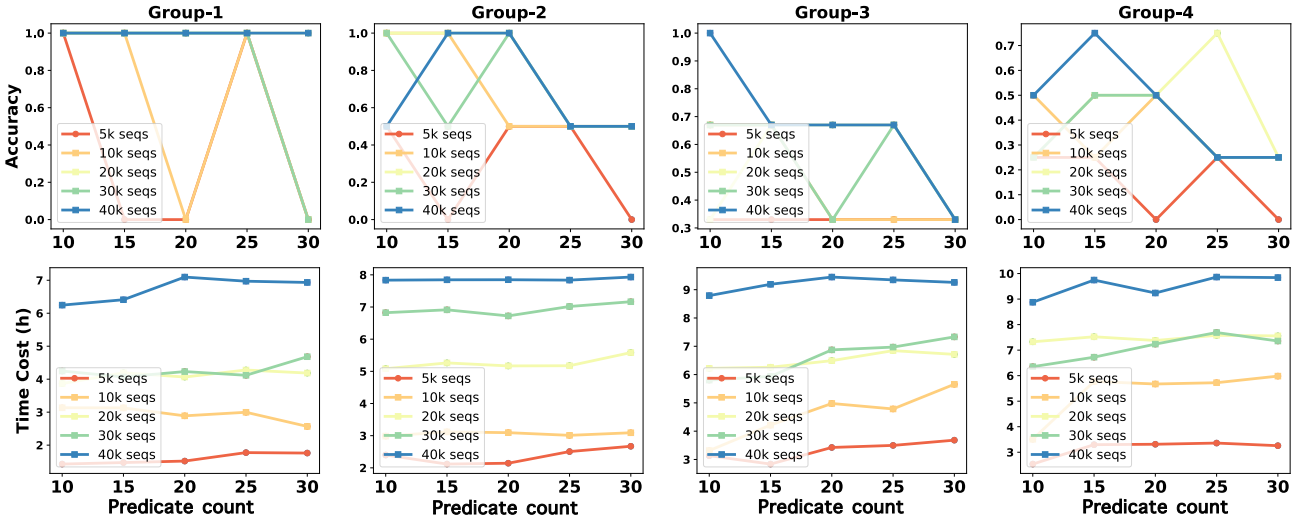

Figure 11: Top: rule learning accuracy for each case. Bottom: time consumed for each case.

	Group-1	Group-2	Group-3	Group-4
Ground Truth	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3 \wedge X_1$ Before X_2	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3 \wedge X_1$ Before X_2 * $Y \leftarrow X_4 \wedge X_5 \wedge X_4$ After X_5	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3$ * $Y \leftarrow X_4 \wedge X_5 \wedge X_4$ Before X_5 * $Y \leftarrow X_6 \wedge X_7 \wedge X_6$ After X_7	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3$ * $Y \leftarrow X_4 \wedge X_5$ * $Y \leftarrow X_6 \wedge X_8$ * $Y \leftarrow X_7 \wedge X_9 \wedge X_{10}$
Case-1	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3 \wedge X_1$ Before X_2	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3 \wedge X_1$ Before X_2 * $Y \leftarrow X_4 \wedge X_5 \wedge X_4$ After X_5	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3$ * $Y \leftarrow X_4 \wedge X_5 \wedge X_6 \wedge X_4$ After $X_6 \wedge X_5$ After X_6 * $Y \leftarrow X_5 \wedge X_6 \wedge X_7 \wedge X_5$ After $X_6 \wedge X_5$ After $X_7 \wedge X_6$ After X_7	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3$ * $Y \leftarrow X_4 \wedge X_5 \wedge X_{10}$ * $Y \leftarrow X_6 \wedge X_8 \wedge X_9$ * $Y \leftarrow X_8 \wedge X_9 \wedge X_{10}$
Case-2	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3 \wedge X_1$ Before X_2	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3$ * $Y \leftarrow X_4 \wedge X_5 \wedge X_4$ Before X_5 * $Y \leftarrow X_6 \wedge X_7 \wedge X_6$ After X_7	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3$ * $Y \leftarrow X_4 \wedge X_5 \wedge X_7 \wedge X_4$ After $X_7 \wedge X_5$ After X_7 * $Y \leftarrow X_6 \wedge X_7 \wedge X_6$ After X_7	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3$ * $Y \leftarrow X_4 \wedge X_5 \wedge X_8$ * $Y \leftarrow X_6 \wedge X_8$ * $Y \leftarrow X_7 \wedge X_9 \wedge X_{10}$
Case-3	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3 \wedge X_1$ Before X_2	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3 \wedge X_1$ Before X_2 * $Y \leftarrow X_4 \wedge X_5 \wedge X_4$ After X_5	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3$ * $Y \leftarrow X_4 \wedge X_5 \wedge X_4$ Before X_5 * $Y \leftarrow X_9 \wedge X_{10} \wedge X_{11} \wedge X_9$ Equal X_{11}	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3$ * $Y \leftarrow X_4 \wedge X_5 \wedge X_8$ * $Y \leftarrow X_6 \wedge X_8$ * $Y \leftarrow X_7 \wedge X_9 \wedge X_{10} \wedge X_7$ After X_9
Case-4	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3 \wedge X_1$ Before X_2	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3 \wedge X_1$ Before X_2 * $Y \leftarrow X_2 \wedge X_4 \wedge X_5 \wedge X_2$ Equal $X_5 \wedge X_2$ Equal $X_6 \wedge X_5$ After X_6	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3$ * $Y \leftarrow X_1 \wedge X_4 \wedge X_5 \wedge X_4$ Before X_5 * $Y \leftarrow X_1 \wedge X_6 \wedge X_7 \wedge X_6$ After X_7	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3$ * $Y \leftarrow X_4 \wedge X_5$ * $Y \leftarrow X_6 \wedge X_8 \wedge X_9$ * $Y \leftarrow X_8 \wedge X_9 \wedge X_{10} \wedge X_9$ After X_{10}
Case-5	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3 \wedge X_1$ Before X_2	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3 \wedge X_1$ Before X_2 * $Y \leftarrow X_2 \wedge X_4 \wedge X_5 \wedge X_2$ Equal $X_5 \wedge X_2$ Equal $X_6 \wedge X_5$ After X_6	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3$ * $Y \leftarrow X_1 \wedge X_4 \wedge X_5 \wedge X_4$ Before X_5 * $Y \leftarrow X_1 \wedge X_6 \wedge X_7 \wedge X_6$ After X_7	* $Y \leftarrow X_1 \wedge X_2 \wedge X_3$ * $Y \leftarrow X_4 \wedge X_5 \wedge X_{10}$ * $Y \leftarrow X_6 \wedge X_8 \wedge X_9$ * $Y \leftarrow X_8 \wedge X_9 \wedge X_{10}$

Table 5: Results of Group-1, Group-2, Group-3 and Group-4. *Ground truth rules / learned grounded rules. * Rules that are wrongly learned. * Rules that are not learned.

8.3 Computing Infrastructure

All synthetic data experiments, as well as the real-world data experiments, including the comparison experiments with baselines, are performed on Ubuntu 20.04.3 LTS system with Intel(R) Xeon(R) Gold 6248R CPU @ 3.00GHz, 227 Gigabyte memory.

8.4 About Baselines

We consider the following baselines through synthetic data experiments and healthcare data experiments to compare the rule learning ability and event prediction with our proposed model:

Neural-based (black-box) models for irregular event data

- Transformer Hawkes Process (THP) [Zuo et al., 2020]: It is a concurrent self-attention based point process model with additional structural knowledge.
- Recurrent Marked Temporal Point Processes (RMTPP) [Du et al., 2016]: It uses RNN to learn a representation of the past events and time intervals.
- ERPP [Xiao et al., 2017]: It uses two RNN models for the background of intensity function and the history-dependent intensity part.
- LG-NPP algorithm [Zhang et al., 2021]: It applies self-attention to embed the event sequence data. It uses a latent random graph to model the relationship between different event sequences and proposes a bilevel programming algorithm to uncover the latent graph and embedding parameters.

Simple parametric/nonparametric models for irregular event data

- Granger Causal Hawkes (GCH) [Xu et al., 2016]: A multivariate Hawkes model with sparse-group-lasso and pairwise similarity constriants. It uses an EM learning algorithm with the basis function of Hawkes process selected adaptively.
- GM-NLF algorithm [Eichler et al., 2017]: It is a multivariate Hawkes Processes with nonparameteric intensity function.

Logical models for irregular event data

- Clock Logic Neural Networks (CLNN) [Yan et al., 2023]: It learns weighted clock logic (wCL) formulas as interpretable temporal rules by which some events promote or inhibit other events. Moreover, CLNN models temporal relations between events using conditional intensity rates informed by a set of wCL formulas.
- TELLER [Li et al., 2021]: A non-differentiable algorithm for uncovering temporal logic rules.

We compared our model with these baselines in event prediction accuracy. Our model achieves promising results and meanwhile is interpretable as CLNN.

Itemset mining methods

- Apriori [Agrawal et al., 1994]: An itemset mining method, which identifies frequent individual items and iteratively extends them to larger itemsets.
- NEclatcloesed [Aryabarzan and Minaei-Bidgoli, 2021]: An itemset mining method, which enhances Apriori's speed and memory efficiency, by employing a depth-first traversal of an itemset search tree, constructing partial closures, and using a hashmap for subsumption checking.

Sequential pattern mining

- CM-spade [Fournier-Viger et al., 2014a]: A sequential pattern mining method, which employs co-occurrence data via a compact Co-occurrence Map (CMAP) for pruning infrequent candidates. CM-SPADE integrates CMAP into the SPADE algorithm, enhancing performance by eliminating costly database scans.
- VGEN [Fournier-Viger et al., 2014b]: A sequential pattern mining method, extracts fewer yet more representative sequential patterns using a depth-first search approach and leverages co-occurrence data for candidate pruning.

Sequential rule mining

- ERMiner [Fournier-Viger et al., 2014c]: A sequential rule mining method, which extracts sequential rules from sequences, utilizing equivalence classes, merging strategies, and a Sparse Count Matrix for efficient search and candidate pruning.

Inductive logic programming (ILP)

- Aleph [Srinivasan, 2001]: Utilizes a general-to-specific hill-climbing strategy to derive logical theories from positive and negative examples. It starts with a broad clause encompassing all positives, iteratively refining it by adding literals to maintain positive coverage and exclude negatives. Note that ILR is a supervised rule learning method. To implement ILR, we need to release the event label information to the algorithm, which poses an unfair comparision with our method.

We compared our algorithm with these baselines in (temporal) rule mining accuracy with the results summarized and compared in Fig. 12. We also explained the performance metric that can be used to evaluate the rule learning accuracy.

Post-hoc method

- Dynamask [Crabbé and Van Der Schaar, 2021]: A SOTA post-hoc method "Dynamask" that aims to explain the black-box time series models by learning dynamic masks.

We compared the interpretability performance of this post-hoc method with ours on the MIMIC dataset and found medical references to justify the results. Details can be found in Sec. 8.5

8.5 Comparison with post-hoc methods

Post-hoc methods are techniques that try to explain the predictions of a deep learning model after the model is well trained. It is easy to implement. However, the reliability and applicability of post-hoc method have always been questioned [Rudin, 2019]. One key concern is that the post-hoc methods explain the models not the world.

In contrast, logic-based models possess inherent interpretability and function as transparent white-box solutions, although the computation is more challenging.

To further support our arguments, we adopted a SOTA post-hoc method Dynamask [Crabbé and Van Der Schaar, 2021] that aims to explain the black-box time series models by learning dynamic masks. We compare this post-hoc method with our method in terms of knowledge discovery.

We apply Dynamask to our MIMIC-IV datasets. The method assumption is that we have trained a flexible black-box model (such as LSTM) that can accurately capture the complex dynamics of multivariate time series. To explain the black-box model in a post-hoc way, we will construct dynamic perturbation operators, with the mask coefficients indicating the saliency of each feature at each time step.

We considered a single layer LSTM with 128 hidden states as our black-box model and fed sufficient time series sequences for training. These sequences included lab values in numerical form, queried from the MIMIC-IV dataset. In the subsequent post-hoc explanation phase, we employed the Dynamask approach to acquire interpretable

sparse binary dynamic masks. These masks were trained to provide explanations, focusing on segments of time series spanning a 48-hour time window containing instances of abnormal low urine output.

The goal here is to provide counterfactual explanations, such as determining how historical attributes could have been altered to avert occurrences of low urine output events.

In the below table, we list several representative discovered patterns in the learned masks. For each pattern, we report some important (event_i, time_j), observational status, and corresponding counterfactual status, that is, when the observation of event i at time j is perturbed by using the observation of this event at adjacent times, it will cause the original abnormal "Low Urine Output" in the 48th hour to change to the "Normal Higher Value".

Patient-ID	Time	Event	Observational Status	Counterfactual Status
10606611	17	Arterial Blood Pressure systolic	abnormal	normal
	17	Arterial Blood Pressure mean	abnormal	normal
	17	Respiratory Rate	abnormal	normal
	11	Venous O2 Pressure	abnormal	normal
	22	Venous O2 Pressure	abnormal	normal
	28	Venous O2 Pressure	abnormal	normal
10623984	42	Sodium	normal	abnormal
	42	Chloride	normal	abnormal
	42	Hemoglobin	abnormal	normal
	42	Platelet Count	abnormal	normal
	39	Lactic Acid	abnormal	normal
	46	Lactic Acid	abnormal	normal
11281568	47	SpO2	abnormal	normal
	3	Sodium	normal	abnormal
	30	Sodium	normal	abnormal
	34	Sodium	normal	abnormal
	32	WBC	abnormal	normal
	40	WBC	abnormal	normal
11395953	37	Sodium	normal	abnormal
	37	BUN	abnormal	normal
	37	IRN	normal	abnormal
	37	Venous O2 Pressure	abnormal	normal
	37	Venous O2 Pressure	abnormal	normal
	45	Lactic Acid	abnormal	normal
	47	Lactic Acid	abnormal	normal

Table 6: Representative Patterns derived using Dynamask.

The results may imply that abnormal status of variables such as Arterial Blood Pressure, Arterial CO2 Pressure, Venous O2 Pressure, Respiratory Rate, and others play a crucial role in influencing changes in patients' urine output. These findings align with the discoveries outlined in our paper. Additionally, sudden changes in Lactic Acid and Platelet Count should also alert patients.

Some of Dynamask's conclusions are consistent with our findings. However, unlike our method, which has been strictly reviewed and affirmed by medical experts, Dynamask's findings have not been confirmed by experts. Furthermore, we verify and compare some inaccuracies in counterfactual interpretation with reference to the medical literature. Experiment results have indicated that Dynamask might overlook critical clinical features.

For instance, it omitted tachycardia (abnormal heart rate), a significant omission. Tachycardia is a well-established early symptom of sepsis, observed in over 80% of patients [Komorowski et al., 2018]. Tachycardia reflects compensatory mechanisms responding to infection-induced hypovolemia and myocardial depression, both influencing urine output [Singer et al., 2016]. In contrast, our method effectively incorporates abnormalities in heart rate within learned rules, such as

Rule 6: LowUrine $\leftarrow$ abnormal Heart Rate $\wedge$ abnormal BUN $\wedge$ abnormal WBC, abnormal Heart Rate Before abnormal BUN, abnormal BUN Before abnormal WBC

Experiment results have also shown that Dynamask may discover patterns that contradict clinical knowledge. For instance, clinical understanding suggests that some sepsis patients may experience extracellular fluid volume depletion, leading to hyponatremia [Bagshaw et al., 2009]. Sepsis-induced inflammatory response and fluid shifts can decrease blood sodium concentration, potentially causing concentrated urine and reduced urine output. Only when blood sodium concentration returns to normal from an abnormal state might urine output normalize. However, Dynamask's patterns for specific patients exhibit perplexing opposite scenarios, diminishing the algorithm's reliability.

The above examples reflect that our method better aligns with established clinical understanding of sepsis, whereas Dynamask occasionally fails to capture the intricate physiological interplay between sepsis, fluid/electrolyte status, and urine output.

In summary, our logic-based has the following advantages over Dynamask (and maybe other post-hoc methods):

(1) Post-hoc methods rely on accurate black-box models, which are hard to obtain when the sample size is small.

(2) The instance-wise explanations generated by Dynamask are not enough to provide a global understanding of the learned model. Our method can easily generalize through learning logical rules and be used in inductive settings.

(3) Dynamask's explanations can only reflect the importance of each feature independently and cannot express the relationships between features, requiring further human interpretation. Our method can directly learn interpretable rules and their corresponding weights.

8.6 Compare with other rule mining methods

	Ground Truth	Ours*	Apriori	NEclatclosed	CM-SPADE	VGEN	ERMiner	Aleph	TELLER
Group-1 (1 rules)	Y ← X1 ∧ X2 ∧ X3 ∧ X1 Before X2	Y ← X1 ∧ X2 ∧ X3 ∧ X1 Before X2	{X1, X2, X3, Y} #sup: 0.37 #rank: 7	{X1, X2, X3, Y} #sup: 0.37 #rank: 5	X3 -> X1 -> X2 -> Y #sup: 0.12 #rank: 146	X1 -> X3 -> X2 -> Y #sup: 0.13 #rank: 97	{X1, X2, X3} -> {Y} #sup: 0.37 #conf: 0.46 #rank: 6	Y ← X1 ∧ X2 ∧ X1 Before X2	Y ← X1 ∧ X2 ∧ X3 ∧ X1 Before X2
Group-2 (2 rules)	Y ← X1 ∧ X2 ∧ X3 ∧ X1 Before X2	Y ← X1 ∧ X2 ∧ X3 ∧ X1 Before X2	{X1, X2, X3, Y} #sup: 0.26 #rank: 7	{X1, X2, X3, Y} #sup: 0.26 #rank: 6	X1 -> X2 -> X3 -> Y #sup: 0.06 #rank: 156	X1 -> X3 -> X2 -> Y #sup: 0.10 #rank: 85	{X1, X2, X3} -> {Y} #sup: 0.26 #conf: 0.38 #rank: 6	Y ← X1 ∧ X2 ∧ X1 Before X2	Y ← X1 ∧ X2 ∧ X3 ∧ X1 Before X2
	Y ← X4 ∧ X5 ∧ X4 After X5	Y ← X2 ∧ X4 ∧ X5 ∧ X2 Equal X5 ∧ X2 Equal X6 ∧ X5 After X6	{X4, X5, Y} #sup: 0.11 #rank: 485	{X4, X5, Y} #sup: 0.11 #rank: 466	X5 -> X4 -> Y #sup: 0.11 #rank: 77	X5 -> X4 -> Y #sup: 0.11 #rank: 77	{X4, X5} -> {Y} #sup: 0.11 #conf: 0.37 #rank: 9	Y ← X4 ∧ X5 ∧ X4 After X5	Y ← X4 ∧ X5 ∧ X4 After X5
Group-3 (3 rules)	Y ← X1 ∧ X2 ∧ X3	Y ← X1 ∧ X2 ∧ X3	{X1, X2, X3, Y} #sup: 0.27 #rank: 7	{X1, X2, X3, Y} #sup: 0.27 #rank: 7	X3 -> X1 -> X2 -> Y #sup: 0.04 #rank: 10752	X3 -> X1 -> X2 -> Y #sup: 0.04 #rank: 617	{X1, X2, X3} -> {Y} #sup: 0.27 #conf: 1.0 #rank: 4	Y ← X2	Y ← X1 ∧ X2 ∧ X3
	Y ← X4 ∧ X5 ∧ X4 Before X5	Y ← X1 ∧ X4 ∧ X5 ∧ X4 Before X5	{X4, X5, Y} #sup: 0.11 #rank: 427	{X4, X5, Y} #sup: 0.11 #rank: 447	X4 -> X5 -> Y #sup: 0.11 #rank: 107	X4 -> X5 -> Y #sup: 0.11 #rank: 65	{X4, X5} -> {Y} #sup: 0.11 #conf: 0.38 #rank: 197	Y ← X4 ∧ X5 ∧ X4 Before X5	Y ← X4 ∧ X5 ∧ X4 Before X5
	Y ← X6 ∧ X7 ∧ X6 After X7	Y ← X1 ∧ X6 ∧ X7 ∧ X6 After X7	{X6, X7, Y} #sup: 0.16 #rank: 39	{X6, X7, Y} #sup: 0.15 #rank: 40	X7 -> X6 -> Y #sup: 0.16 #rank: 29	X7 -> X6 -> Y #sup: 0.16 #rank: 29	{X6, X7} -> {Y} #sup: 0.16 #conf: 0.38 #rank: 193	Y ← X6 ∧ X7 ∧X6 After X7	Y ← X7 ∧ X9
Group-4 (4 rules)	Y ← X1 ∧ X2 ∧ X3	Y ← X1 ∧ X2 ∧ X3	{X1, X2, X3, Y} #sup: 0.28 #rank: 27	{X1, X2, X3, Y} #sup: 0.28 #rank: 27	X3 -> X1 -> X2 -> Y #sup: 0.05 #rank: 642	X3 -> X1 -> X2 -> Y #sup: 0.05 #rank: 586	{X1, X2, X3} -> {Y} #sup: 0.28 #conf: 1.0 #rank: 5	Y ← X2	Y ← X1 ∧ X2 ∧ X3
	Y ← X4 ∧ X5 ∧ X4 After X5	Y ← X4 ∧ X5 ∧ X10	{X4, X5, Y} #sup: 0.14 #rank: 361	{X4, X5, Y} #sup: 0.14 #rank: 361	X5 -> X4 -> Y #sup: 0.06 #rank: 519	X5 -> X4 -> Y #sup: 0.06 #rank: 450	{X4, X5} -> {Y} #sup: 0.06 #conf: 0.38 #rank: 157226	Y ← X4 ∧ X5 ∧ X4 After X5	Y ← X1 ∧ X4
	Y ← X6 ∧ X8	Y ← X6 ∧ X8 ∧ X9	{X6, X8, Y} #sup: 0.25 #rank: 30	{X6, X8, Y} #sup: 0.25 #rank: 30	X6 -> X8 -> Y #sup: 0.12 #rank: 32	X6 -> X8 -> Y #sup: 0.12 #rank: 32	{X6, X8} -> {Y} #sup: 0.25 #conf: 1.0 #rank: 6	Y ← X6	Y ← X6 ∧ X9
	Y ← X7∧X9∧X10 ∧X7 Equal X9	Y ← X8 ∧ X9 ∧ X10	{X7, X9, X10, Y} #sup: 0.17 #rank: 124	{X7, X9, X10, Y} #sup: 0.17 #rank: 124	X7 -> X9 -> X10 -> Y #sup: 0.02 #rank: 2014	X7 -> X9 -> X10 -> Y #sup: 0.02 #rank: 757	{X7, X9, X10} -> {Y} #sup: 0.07 #conf: 0.23 #rank: 212951	Y ← X7 ∧ X9 ∧ X7 Equal X9	Y ← X1 ∧ X7

Figure 12: Learned rules using newly added baselines. $\#sup$ is support, the number of samples that contain corresponding pattern divided by the number of all samples. $\#conf$ is confidence, the number of samples that contain all items of X before all items of Y, divided by the number of samples that contains items in X. $\#rank$ is the descending order of $\#conf$ for a pattern among all Y-related patterns.

Association Rule Mining (ARM) is an unsupervised learning algorithm aimed at identifying frequent itemsets using metrics like support and confidence that are then utilized to deduce association rules.

Group	Method	#support	#rank	Jaccard Index
Group-1	Apriori	0.37	7	0
	CM-SPADE	0.12	146	0
	ERMiner	0.37	6	0
	Ours*	–	–	1
Group-2	Apriori	0.26, 0.11	7, 485	0
	CM-SPADE	0.11, 0.06	77, 156	0
	ERMiner	0.26, 0.11	6, 9	0
	Ours*	–	–	0.5
Group-3	Apriori	0.27, 0.16, 0.11	7, 39, 427	0
	CM-SPADE	0.16, 0.11, 0.04	29, 107, 10752	0
	ERMiner	0.27, 0.16, 0.11	4, 193, 197	0
	Ours*	–	–	0.33
Group-4	Apriori	0.28, 0.25, 0.17, 0.14	27, 30, 124, 361	0
	CM-SPADE	0.12, 0.06, 0.05, 0.02	32, 519, 642, 2014	0
	ERMiner	0.28, 0.25, 0.07, 0.06	5, 6, 157226, 212951	0
	Ours*	–	–	0.25

Table 7: Compare our method with other rule mining baselines using #support, #rank, and jaccard similarity metircs.

As shown in Fig. 12, we compare three types of ARM: **itemset mining methods**, which ignore event ordering information, including Apriori [Agrawal et al., 1994] and NEclatcloesed [Aryabarzan and Minaei-Bidgoli, 2021]; **sequential pattern mining**, which cannot handle precise timestamps, including CM-spade [Fournier-Viger et al., 2014a] and VGEN [Fournier-Viger et al., 2014b]; and **sequential rule mining**, which ignore internal temporal relations, including ERMiner [Fournier-Viger et al., 2014c]. The learned rules by itemset mining methods like $\{X, Y\}$ only capture event co-occurrence, neglecting temporal relationships. The acquired rules by sequential pattern mining method take the form of $X_i \to ... \to X_j \to Y$, which represents sequences where events strictly adhere to a defined temporal relationship. Learned rules by sequential rule mining method are like $\{X\} \to \{Y\}$, where events in the rule body precede those in the rule head, but internal temporal relations within the body and head events are disregarded. Unlike ARM, our method considers a range of temporal relations (Before, After, Equal, None) between pairwise body predicates. Our rule forms encompass ARM, enabling us to manage more intricate temporal relationships.

Further, we consider **inductive logic programming (ILP) method**, which is a supervised technique that learns logical rules from factual data, positive and negative examples. It prioritizes positive coverage while minimizing negative coverage and can incorporate background knowledge through extended logic programs. ILP requires labeled positive and negative samples to learn logic rules, which can be expensive or hard to acquire in practice. To implement ILR in our setting, we release the true event label information to ILR, which poses an unfair comparison to our method. In contrast, our algorithm only utilizes the event items' timestamps or ordering information for rule learning, treating label information as a latent variable inferred in the E-step and thereby does not require prior knowledge of the sample labels.

A rigorous comparison between these baselines and our method is shown in Tab. 7, where we selected three representative ARM methods to summarize the results for Group-1 to Group-4 datasets below for a clear comparison. The metrics we used are as follows:

- **#Support**: is the threshold hyperparameter for baseline methods, determining the rule appearance frequency threshold of the discovered rules in the final set.
- **#Rank**: represents the occurrence frequency ranking of true rules in the discovered rule set. A high #Rank indicates practical unfeasibility
- **Jaccard index**: (between 0 and 1) measures overlap between ground truth rule set and the discovered rule set, counting shared and distinct members.

Tab. 7 confirms our method's better rule discovery accuracy compared to these rule mining baselines. Significantly,

despite the increased problem difficulty from Group-1 to Group-4, our method's Jaccard index drops, our discovered rules still consistently capture the majority or partial rule content. While baseline methods may include true rules in their discovered sets, the high #Rank suggests practical challenges in extracting them from such extensive rule sets.

8.7 MIMIC-IV dataset preprocessing and risk factors extracting

MIMIC-IV[2] is a publicly available database sourced from the electronic health record of the Beth Israel Deaconess Medical Center [Johnson et al., 2023]. Information available includes patient measurements, orders, diagnoses, procedures, treatments, and deidentified free-text clinical notes. Sepsis is a leading cause of mortality in the ICU, particularly when it progresses to septic shock. Septic shocks are critical medical emergencies, and timely recognition and treatment are crucial for improving survival rates. In the real-world experiments on MIMIC-IV dataset, we aim to find logic rules related to septic shocks for the whole patient samples and infer the most likely rule reasons for specific patients, which would be potential early alarm when some abnormal indicators occur.

Patients We select 4074 patients that satisfied the following criteria from the dataset: (1) The patients are diagnosed with sepsis [Saria, 2018]. (2) Patients, if diagonized with sepsis, the timestamps of any clinical testing, specific lab values, timestamps of medication administration and corresponding dosage were not missing.

Outcome Real time urine output was treated as the outcome indicator since low urine output signals directly indicate a poor circulatory system and is a warning sign of septic shock.

Risk Factors Suggested by [Komorowski et al., 2018], we extract 28 risk factors associated with sepsis which are consistent with expert consensus. Based on the distinct clinical characteristics of these risk factors, they can be categorized into the following five groups:

- Vital Signs:
  - Heart Rate: The number of times the heart beats per minute. An elevated or abnormal heart rate may indicate physiological stress or an underlying condition.
  - Arterial Blood Pressure (systolic, mean, diastolic): Measures the force exerted by the blood against the arterial walls during different phases of the cardiac cycle. Abnormal blood pressure values may indicate cardiovascular dysfunction or organ perfusion issues.
  - Temperature (Celsius): Body temperature is a measure of the body's internal heat. Abnormal temperatures may indicate infection, inflammation, or other systemic disorders.
  - Respiratory Rate: The number of breaths taken per minute. Abnormal respiratory rates may suggest respiratory distress or dysfunction.
  - SpO_2: Oxygen saturation level in the blood. Decreased SpO_2 levels may indicate inadequate oxygenation.
- Biochemical Parameters:
  - Potassium, Sodium, Chloride, Glucose: Electrolytes and blood sugar levels that help maintain essential bodily functions. Abnormal levels may indicate electrolyte imbalances, metabolic disorders, or organ dysfunction.
  - Blood Urea Nitrogen (BUN), Creatinine: Indicators of renal function. Elevated levels may suggest impaired kidney function.
  - Magnesium, Ionized Calcium: Important minerals involved in various physiological processes. Abnormal levels may indicate electrolyte imbalances or organ dysfunction.
  - Total Bilirubin: A byproduct of red blood cell breakdown. Elevated levels may indicate liver dysfunction.
  - Albumin: A protein produced by the liver. Abnormal levels may indicate malnutrition, liver disease, or kidney dysfunction.
- Hematological Parameters
  - Hemoglobin: A protein in red blood cells that carries oxygen. Abnormal levels may indicate anemia or oxygen-carrying capacity issues.
  - White Blood Cell (WBC): Cells of the immune system involved in fighting infections. Abnormal levels may indicate infection or inflammation.
  - Platelet Count: Blood cells responsible for clotting. Abnormal levels may suggest bleeding disorders or impaired clotting ability.

[2] https://mimic.mit.edu/

  - Partial Thromboplastin Time (PTT), Prothrombin time (PT), INR: Tests that assess blood clotting function. Abnormal results may indicate bleeding disorders or coagulation abnormalities.

- Blood Gas Analysis
  - pH (Arterial): A measure of blood acidity or alkalinity. Abnormal pH values may indicate acid-base imbalances or respiratory/metabolic disorders.
  - Arterial Base Excess: Measures the amount of excess or deficit of base in arterial blood. Abnormal levels may indicate acid-base imbalances or metabolic disturbances.
  - Arterial CO2 Pressure, Venous O2 Pressure: Parameters that assess respiratory and metabolic function. Abnormal values may indicate respiratory failure or metabolic disturbances.

- Metabolic Parameter
  - Lactic Acid: An indicator of tissue perfusion and oxygenation. Elevated levels may suggest tissue hypoxia or impaired cellular metabolism.

These risk factors are commonly assessed in sepsis patients to monitor their clinical status and guide appropriate interventions. The interpretation of these factors requires clinical judgment and consideration of the patient's overall condition. Tab. 8 shows the categories of the 28 variables extracted from MIMIC-IV dataset, and their reference range.

Data Preprocessing Due to the frequent fluctuations in urine output within the ICU setting, we considered only those instances in which urine output became abnormal after maintaining a normal level for at least 48 hours. These instances were regarded as valid target events that hold significance for prediction and explanation. Additionally, for each patient, we selectively extracted the initial time period that met the criteria. Regarding the risk factors, we documented the time points at which these variables first became abnormal from normal within the 48-hour period preceding the transition of urine output from normal to abnormal.

Category	Risk Factor	Ref Range Lower	Ref Range Higher
Vital Signs	Heart Rate	60/min	100/min
	Arterial Blood Pressure systolic	90 mmHg	120 mmHg
	Arterial Blood Pressure mean	70 mmHg	100 mmHg
	Arterial Blood Pressure diastolic	60 mmHg	80 mmHg
	Temperature Celsius	36.5	37.5
	Respiratory Rate	12/min	20/min
	SpO_2	95%	-
Biochemical Parameters	Potassium	3.5 mmol/L	5.0 mmol/L
	Sodium	135 mmol/L	145 mmol/L
	Chloride	98 mmol/L	106 mmol/L
	Glucose	3.9 mmol/L	5.6 mmol/L
	BUN	7 mg/dL	20 mg/dL
	Creatinine	0.5 mg/dL	1.2 mg/dL
	Magnesium	1.7 mg/dL	2.3 mg/dL
	Ionized Calcium	4.5 mg/dL	5.5 mg/dL
	Total Bilirubin	-	1.2 mg/dL
	Albumin	3.4 g/dL	5.4 g/dL
Hematological Parameters	Hemoglobin	12.0 g/dL	17.5 g/dL
	WBC	4000 cells/mm^3	11000 cells/mm^3
	Platelet Count	150000 platelets/mm^3	450000 platelets/mm^3
	PTT	25s	35s
	PT	11s	13s
	INR	0.9	1.1
Blood Gas Analysis	pH (Arterial)	7.35	7.45
	Arterial CO_2 Pressure	35 mmHg	45 mmHg
	Venous O_2 Pressure	40 mmHg	-
	Arterial Base Excess	-2 mmol/L	2 mmol/L
Metabolic Parameter	Lactic Acid	-	2 mmol/L

Table 8: Description of the risk factors extracted from MIMIC-IV dataset and their corresponding reference range. Variables like SpO_2, Total Bilirubin, Venous O_2 Pressure, and Lactic Acid only have an upper or lower limit for the reference value.

8.8 Doctor Verification and Medical References

Evaluating interpretability of learned temporal logic rules in real-world problems require human expert inspection. In our clinical experiments using the MIMIC-IV dataset, our methodology included the invaluable input of medical experts who reviewed our discovered rule results. Their feedback consistently affirmed the sensibility of our findings. Furthermore, we reinforced our discovered rules with evidence from medical references, as exemplified below:

- **Rule 1: LowUrine ← Arterial Blood Pressure Diastolic**: Arterial diastolic blood pressure serves as an indicator of renal perfusion pressure. A decrease in arterial diastolic blood pressure can diminish vascular perfusion to the kidneys, impacting the glomerular filtration rate and urine output for sepsis patients [Gernardin et al., 1996, Benchekroune et al., 2008].
- **Rule 2: LowUrine ← Ionized Calcium**: Decreased ionized calcium levels are common in critically ill sepsis patients due to an inflammatory response that can inhibit calcium absorption [Mizock, 1995]. Sepsis-related damage to the gastrointestinal mucosa further reduces dietary calcium absorption [Zivin et al., 2001].
- **Rules 3: LowUrine ← Arterial CO2 Pressure, Rule 4: LowUrine ← Venous O2 Pressure**: These rules involve arterial CO2 pressure and venous O2 pressure, both of which are linked to cardiac output and tissue hypoxia in septic shock [Mallat et al., 2014, Mecher et al., 1990, Durkin et al., 1993, Mecher et al., 1990].
- **Rule 5: LowUrine ← Respiratory Rate ∧ Hemoglobin, Respiratory Rate None Hemoglobin**: In early sepsis stages, peripheral vasodilation can lead to hypovolemic shock, decreasing renal perfusion pressure and renal function [Levy et al., 2001, Langenberg et al., 2006]. Sepsis-induced anemia exacerbates tissue hypoxia [Gomez et al., 2014].
- **Rule 6: LowUrine ← Heart Rate ∧ BUN ∧ WBC, Heart Rate Before BUN, BUN Before WBC**: Sequential abnormalities in heart rate, blood urea nitrogen (BUN), and white blood cell (WBC) levels suggest developing renal dysfunction and reduced urine output in sepsis [Annane et al., 2005]. These markers indicate sepsis-induced renal hypoperfusion and glomerular filtration decline.

*Note that the predicates in the body of the above-mentioned learned rules are all in an abnormal state.

References

Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499. Santiago, Chile, 1994.

Djillali Annane, Eric Bellissant, and Jean-Marc Cavaillon. Septic shock. *The Lancet*, 365(9453):63–78, 2005.

Nader Aryabarzan and Behrouz Minaei-Bidgoli. Neclatclosed: A vertical algorithm for mining frequent closed itemsets. *Expert Systems with Applications*, 174:114738, 2021.

Sean M Bagshaw, Derek R Townsend, and Robert C McDermid. Disorders of sodium and water balance in hospitalized patients. *Canadian Journal of Anesthesia/Journal canadien d'anesthésie*, 56(2):151–167, 2009.

Samir Benchekroune, Peter CJ Karpati, Christine Berton, Cédric Nathan, Joaquim Mateo, Mansour Chaara, Florence Riché, Marie-Josèphe Laisné, Didier Payen, and Alexandre Mebazaa. Diastolic arterial blood pressure: a reliable early predictor of survival in human septic shock. *Journal of Trauma and Acute Care Surgery*, 64(5): 1188–1195, 2008.

Jonathan Crabbé and Mihaela Van Der Schaar. Explaining time series predictions with dynamic masks. In *International Conference on Machine Learning*, pages 2166–2177. PMLR, 2021.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1555–1564, 2016.

Raymond Durkin, Mary Ann Gergits, James F Reed III, and John Fitzgibbons. The relationship between the arteriovenous carbon dioxide gradient and cardiac index. *Journal of critical care*, 8(4):217–221, 1993.

Michael Eichler, Rainer Dahlhaus, and Johannes Dueck. Graphical modeling for multivariate hawkes processes with nonparametric link functions. *Journal of Time Series Analysis*, 38(2):225–242, 2017.

Philippe Fournier-Viger, Antonio Gomariz, Manuel Campos, and Rincy Thomas. Fast vertical mining of sequential patterns using co-occurrence information. In *Advances in Knowledge Discovery and Data Mining: 18th Pacific-Asia Conference, PAKDD 2014, Tainan, Taiwan, May 13-16, 2014. Proceedings, Part I 18*, pages 40–52. Springer, 2014a.

Philippe Fournier-Viger, Antonio Gomariz, Michal Šebek, and Martin Hlosta. Vgen: fast vertical mining of sequential generator patterns. In *Data Warehousing and Knowledge Discovery: 16th International Conference, DaWaK 2014, Munich, Germany, September 2-4, 2014. Proceedings 16*, pages 476–488. Springer, 2014b.

Philippe Fournier-Viger, Ted Gueniche, Souleymane Zida, and Vincent S Tseng. Erminer: sequential rule mining using equivalence classes. In *Advances in Intelligent Data Analysis XIII: 13th International Symposium, IDA 2014, Leuven, Belgium, October 30–November 1, 2014. Proceedings 13*, pages 108–119. Springer, 2014c.

G Gernardin, C Pradier, F Tiger, P Deloffre, and M Mattei. Blood pressure and arterial lactate level are early indicators of short-term survival in human septic shock. *Intensive care medicine*, 22:17–25, 1996.

Hernando Gomez, Can Ince, Daniel De Backer, Peter Pickkers, Didier Payen, John Hotchkiss, and John A Kellum. A unified theory of sepsis-induced acute kidney injury: inflammation, microcirculatory dysfunction, bioenergetics and the tubular cell adaptation to injury. *Shock (Augusta, Ga.)*, 41(1):3, 2014.

Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Benjamin Moody, Brian Gow, Li-wei H Lehman, et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific Data*, 10(1):1–9, 2023.

Matthieu Komorowski, Leo A Celi, Omar Badawi, Anthony C Gordon, and A Aldo Faisal. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature medicine*, 24(11): 1716–1720, 2018.

C Langenberg, L Wan, M Egi, CN May, and R Bellomo. Renal blood flow in experimental septic acute renal failure. *Kidney international*, 69(11):1996–2002, 2006.

Mitchell M Levy, Mitchell P Fink, John C Marshall, Edward Abraham, Derek Angus, Deborah Cook, Jonathan Cohen, Steven M Opal, Jean-Louis Vincent, and Graham Ramsay. International sepsis definitions conference, 2003. 2001 SCCM/ESICM/ACCP/ATS/SIS International Sepsis Definitions Conference . . . , 2001.

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

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 *International Conference on Learning Representations*, 2021.

Jihad Mallat, Florent Pepy, Malcolm Lemyze, Gaelle Gasan, Nicolas Vangrunderbeeck, Laurent Tronchon, Benoit Vallet, and Didier Thevenin. Central venous-to-arterial carbon dioxide partial pressure difference in early resuscitation from septic shock: a prospective observational study. *European Journal of Anaesthesiology—EJA*, 31(7):371–380, 2014.

Carter E Mecher, Eric C Rackow, Mark E Astiz, and Max Harry Weil. Venous hypercarbia associated with severe sepsis and systemic hypoperfusion. *Critical care medicine*, 18(6):585–589, 1990.

Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems*, 30, 2017.

Barry A Mizock. Alterations in carbohydrate metabolism during stress: a review of the literature. *The American journal of medicine*, 98(1):75–84, 1995.

Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. *Advances in Neural information processing systems*, 31, 2018.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

Suchi Saria. Individualized sepsis treatment using reinforcement learning. *Nature medicine*, 24(11):1641–1642, 2018.

Mervyn Singer, Clifford S Deutschman, Christopher Warren Seymour, Manu Shankar-Hari, Djillali Annane, Michael Bauer, Rinaldo Bellomo, Gordon R Bernard, Jean-Daniel Chiche, Craig M Coopersmith, et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). *Jama*, 315(8):801–810, 2016.

Ashwin Srinivasan. The aleph manual, 2001.

Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen Chu. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Sang Michael Xie and Stefano Ermon. Reparameterizable subset sampling via continuous relaxations. *arXiv preprint arXiv:1901.10517*, 2019.

Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning granger causality for hawkes processes. In *International conference on machine learning*, pages 1717–1726. PMLR, 2016.

Ruixuan Yan, Yunshi Wen, Debarun Bhattacharjya, Ronny Luss, Tengfei Ma, Achille Fokoue, and Anak Agung Julius. Weighted clock logic point process. In *The Eleventh International Conference on Learning Representations*, 2023.

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

Qiang Zhang, Aldo Lipani, and Emine Yilmaz. Learning neural point processes with latent graphs. In *Proceedings of the Web Conference 2021*, pages 1495–1505, 2021.

Julie R Zivin, Theodore Gooley, Richard A Zager, and Michael J Ryan. Hypocalcemia: a pervasive metabolic abnormality in the critically ill. *American journal of kidney diseases*, 37(4):689–698, 2001.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International conference on machine learning*, pages 11692–11702. PMLR, 2020.