Incorporating Causal Effects into Deep Learning Predictions on EHR Data

Jia Li*, Haoyu Yang*, Xiaowei Jia†, Vipin Kumar*, Michael Steinbach*, Gyorgy Simon‡

Abstract—Electronic Health Records (EHR) data analysis plays a crucial role in healthcare system quality. Because of its highly complex underlying causality and limited observable nature, causal inference on EHR is quite challenging. Deep Learning (DL) achieved great success among the advanced machine learning methodologies. Nevertheless, it is still obstructed by the inappropriately assumed causal conditions. This work proposed a novel method to quantify clinically well-defined causal effects as a generalized estimation vector that is simply utilisable for causal models. We incorporated it into DL models to achieve better predictive performance and result interpretation. Furthermore, we also proved the existence of causal information blink spots that regular DL models cannot reach.

Keywords— Causal Effect, EHR, Deep Learning

I. INTRODUCTION

Given the wide adoption of Electronic Health Records (EHR) systems in the US, there is a surge of interest in building EHR data-driven models for predicting clinical events, and it is still ongoing. They are used to estimate disease progress or identify the risky population, enabling better diagnosis, prognosis, and treatment decisions. Hence their predictive accuracy, as well as the result interpretations, are both crucial for healthcare quality, especially in the context of real-time applications at the bedside, like precision medicine [1].

It is known that clinical processes are motivated by the underlying causal mechanism and lead to observable events along the timeline, e.g., with some medication through organs, the effects sequentially come to observation. But events will cause subsequent events and produce much-complicated relations, such as mutually determined disease progress and medical upgrading. Therefore, people have widely adopted causality modeling techniques in EHR analysis in the past decade, which has been well developed as an essential branch of Machine Learning (ML) [2].

However, challenges still exist. In practice, lots of clinical events can only be limited to observable. Thus, information in EHR may not be sufficient to reflect the underlying causation fully. The significant application that aims to discover causal information as much as possible from observations is known as causal inference. This area has attracted various applications of advanced learning techniques, such as DL methods [6], [26]. Based on Neural Networks (NNs) architecture, DL models have shown great success in harnessing the power of extensive EHR data and coming up with better performance. This success is thanks to their advanced ability to extract deep hierarchical features and non-linear dependencies.

Nevertheless, modeling graphical causality relies on certain causal assumptions, and so do the DL-based models. They may lead to inevitable deviations in causal model results and their interpretations.

On the other hand, health statistics have provided extensive knowledge about the hidden causation, but they are not intuitively utilisable for DL models. We notice that among the applications on EHR, DL models are bare to be considered as short of the capability to discover causal effects. In other words, questions are still unclear: In which scope the causal knowledge is learnable by integral modeling? And does there exist DL-irreplaceable knowledge in domain statistics?

To answer these questions, we propose a novel method to incorporate clinically well-defined causal effects into DL models and check their predictive performance improvement. We hypothesize that some causal information exists that regular DL models cannot automatically discover. And they need external knowledge infused to reinforce their causal learning ability.

We summarize our contributions as:

1) Proved existence of the causal-info blink spots for DL.
2) Raised a novel method to quantify the domain-defined causal effects as a generalized estimation vector.
3) Realized a generally usable and configurable Causal EHR data Generator and successfully used it to simulate a complex comorbidities progress trajectory.

II. RELATED WORK

EHR data attributes usually include diagnosis, lab results, medical notes, prescriptions, along with demographics (age, gender, race) and other clinically relevant features. The various data types make it richly informative and challenging to build up the causal model.

In this section, we will introduce structural causal models with DL-based techniques and their limitations on EHR data; and then briefly summarize the current DL applications in EHR studies.

A. Graphical Causal Models with DL

In the last decade, the interest in combining DL models with causality learning has increased in both the DL and the causality communities [18]. These tasks can be briefly classified as causal structure learning, causal effects estimation, or on a mixed purpose of both.

The methods that learn causality as an integral structural model are known as "Structural Equation models" (SEMs), or...
"Functional Causal Models” (FCMs) [19], [20]. The structure is commonly described as conditionally-dependent relationships among variables using Directed Acyclic Graphs (DAGs). Usually, the parametric methods assume deterministic functions for all direct causal relations and apply constraints globally on the joint distribution [3].

On the other hand, the advantage of NNs-based methods is on generalized non-parametric modeling and being able to avoid prior assumptions. But due to the short of structural consideration, they are commonly applied on causal effects estimations within a determined or non-required causal structure [21]–[23]. In recent years, a methodology arose to transform the discrete graphical structure learning problems into continuous optimization ones that make NNs applicable [4]. The key idea is to quantify the acyclic-ness that is optimizable by a constrained adjacency matrix.

However, the learned causal relations do not necessarily have a causal interpretation. Mostly, they rely on or imply the existence of Causal Sufficiency Assumption or Causal Markov Assumption that are possibly unverifiable [24].

(1) Causal Sufficiency.

It assumes that the DAG of modeling has included all causal relations among variables. In some scenarios, it can also be referred to as Causality Faithfulness.

Indeed, the conditional dependency of DAG can only provide necessary but not sufficient conditions for identifying causalities. There may exist other relations of interest, not entailed by the structural modeling but assumed to be nonexistent.

(2) Causal Markov Condition.

It means that every node in the DAG is conditionally independent of its nondescendents, given its parents, which has been generally acknowledged short of the capability to distinguish confounding.

In practical applications, deep learning models are much more potent than the conventional methods of discovering hidden causal relations. But since they still have applied inappropriate assumptions and black-box nature, they can hardly guarantee to produce interpretable results.

\[ X_t = \text{Patient Features Variable} \]
\[ M_t = \text{Treatments (Including Medications)} \]
\[ Y_t = \text{Diagnosis} \]

\[ S_t = \text{Disease Severity (Unobservable)} \]

Fig. 1: The EHR Causal Structure. Left: observed view. Right: knowledge-based inferred view. The solid circle means element observable, and the dashed one tells not.

B. DL Applications on EHR

In the recent five years, DL-based methods turned out to be favorable for EHR data applications. They are primarily used in two categories: prediction tasks and representation learning. Here we mainly concentrate on the predictions.

DL models are commonly adopted to enhance predictive performance effectively. For example, Deep Patient [14] predicts multi-outcomes of patients, Deeper [8] targets on hospital re-admissions, and Doctor AI [28] was proposed for heart failure prediction.

For temporal EHR data, RNN is mainly considered as an advanced time-series model [5], [10], [11]. There are two popular mechanisms of RNNs, Gated Recurrent Units (GRU) [28], [29] and Long Short-term Memory (LSTM), [15], [27] that employ different network architectures but are similarly efficient to leveraging temporal information.

However, EHR data has particular characteristics that may mislead DL models. Considering their applications on treatment effects learning, we raise a hypothesis that RNNs cannot automatically discover the underlying causal information of interest and may need external knowledge infusion in some ways.

III. CAUSALITY ON EHR DATA

This section mainly discusses the mechanism of underlying causality in EHR. And then, from the aspects of causal assumptions, we use living examples to illustrate the reason for their violations.

A. Underlying Causal Structure

Suppose an EHR dataset spans \( T \) time steps. At any step \( t \), the observed data elements include: recorded patient attributes \( X \) (e.g., lab results, vital signs, and demographics), subscribed medications \( M \) (or some other clinical treatments), and disease diagnosis \( Y \).

Fig. 1 displays the typical causal structure from two views: the one directly observable from data and the perspective based on background knowledge.

Here the hidden variable \( S_t \) represents patient’s Disease Severity at time \( t \). Without loss of generalization, we define it as aggregating all helpful information to infer the causal relations at \( t \). As \( X_t \) only indicates the visible part, we generally consider \( S_t \) more informative than \( X_t \).

Commonly, the medical plans of patients are decided by physicians’ observation, including the historical records denoted as \( X_t \) here. Then the medication effects reflect on the subsequent records \( X_{t+1} \), which form a typical confounding relationship. The scientists from traditional statistics have adequately explored solutions to perform de-confounding that involve extensive clinical knowledge.

However, a more significant challenge for predictions on EHR is the unobservable critical information. The disease severity does not have a specific clinical definition but only acts as a conceptual element in our models. Among the machine learning literature, we have ever reviewed, mostly \( X_t \) is simply considered the roll of \( S_t \) without other choices.
But that essentially leads to some problems, like the violation of regularly applied causal assumptions.

On the other hand, the experts from health stats have revealed that although the disease severity is not directly observable, they can be inferred, or at least partially inferred from the observations by leveraging domain knowledge. Therefore, the motivation of this work is to combine the advantages of generalizable DL methods and interpretable statistical models.

B. Invalid Causal Assumptions on EHR

Because of the omitted underlying information, the causality learned of EHR may be distorted to some extent and possibly leads to violations of standard causal assumptions.

1) Unreliable Criterion - Causal Sufficiency violated.
Usually, disease criterion is set up as a metric of certain test results. But it may be triggered by other causes and lead to false positives.

[Example A]. For diabetes patients, high Blood Pressure (BP) is a sign of danger. But in cardiovascular surgery (which may be caused by severe diabetes), increasing BP by anticoagulant drugs is mandatory to prevent blood clotting.

2) Mislead Medicine - Causal Sufficiency violated.
Adding more intense medication is usually considered worse ill, but the reality may be counter-intuitive.

[Example B]. Increasing insulin use is a significant sign of worse diabetes. But after a kidney transplant surgery (which may be caused by severe diabetes), this increasing conversely indicates a more functional body.

3) Non-negligible History - Causal Markov Condition violated.
Similar records may be developed as different disease progresses.

[Example C]. Pregnant women who just developed gestational Hypertension (HTN) has very similar lab results with an HTN patient who just got pregnant but differentiate on personal effects to anti-hypertension drugs.

IV. PROPOSED METHODOLOGY

To bring usable knowledge into the DL modeling process, we propose a causal effect estimator, calculated with the traditional method but produce new information that can hardly be discovered by regular DL models, and eventually increase the predictive performance. We must emphasize that our calculation does not rely on any extra data input.

We will first introduce the definition, present the calculation procedure, and finally, analyze the fundamental principle.

A. Treatment Effect Estimator \( \Delta \)

We use \( X_t \) to denote the patients observation records at time \( t \) that are predictive to the next time step outcome \( Y_{t+1} \) or records \( X_{t+1} \). Practically, \( X_t \) consists of lab results and vital signs, such as blood pressure and BMI index. Then, we use \( M_t \) to denote the treatments of interest at time \( t \). In our context, \( M_t \) is the vector of all possible medications, but it can also include surgeries, therapies, or any other clinical treatments in practice.

Let vector \( M_t = (m_1, m_2, \ldots, m_n) \), where \( m_i \in [0, 1] \). \( m_i \) indicates whether the \( i \)th medication (out of \( n \) in total) is applied to patient at time \( t \) (\( m_i = 1 \)) or not (\( m_i = 0 \)). We use \( M_t^{+} \) and \( M_t^{-} \) to represent the pair of the two possibilities at time \( t \) that \( M_t^{+} \) has \( m_i = 1 \) while \( M_t^{-} \) has \( m_i = 0 \), and all the other medications keep identical between them. To be specific, \( M_t^{+} = (m_1, m_2, \ldots, m_i = 1, \ldots, m_n) \) and \( M_t^{-} = (m_1, m_2, \ldots, m_i = 0, \ldots, m_n) \).

The considered scenario is that given known \( M_t^{+} \) at time \( t - 1 \), the patient will possibly be under ongoing treatment as \( M_t^{-} \) or additionally medicated as \( M_t^{+} \). Accordingly, we use \( (X_{t+1}, M_t^{+}) \) and \( (X_{t+1}, M_t^{-}) \) to represent two possible values of the observable records at \( t + 1 \), under the two different medical plans at \( t \), respectively.

\textbf{Definition 1.} The Treatment Effect Estimator \( \Delta_{t+1} \) about time \( t \), is defined as

\[
\Delta_{t+1} = \begin{cases} 
(X_{t+1}^{+} - X_{t+1}^{-}), & \text{if} \ I(M_t^{+}) = 1, \text{given} \ I(M_t^{-}) = 1 \\
0, & \text{otherwise}
\end{cases}
\]

Here we only discussed the situation with a single medication \( m_i \) possibly changed. However, it can stand for any medications subset of interest, as the user-determined.

The proposed estimator \( \Delta_{t+1} \) can be described as “how much \( X \) would be changed if the additional medications were applied, compared with not applied.” Apparently, \( \Delta_{t+1} \) can only have non-zero meaningful value when \( M_t^{+} \) is known to be happened. In other words, at time \( t \) the truly observed \textbf{Factual Value} is \( X_{t+1}^{+} \), while \( X_{t+1}^{-} \) is called as \textbf{Counterfactual Value}.

The idea about \( \Delta \) is inspired by a well-grounded conception from clinical studies, the \textbf{Individual Treatment Effect (ITE)} \cite{31}, which is defined as the individual-level difference between the factual and counterfactual values of the outcome for each patient, such as some diagnosis or mortality risk. Instead of outcomes, we creatively use it upon data attributes to form a “difference vector” to describe the medical effect.

B. Estimation of \( \Delta \) Value

By \( \Delta \)'s definition, the challenge to calculate it is estimating the counterfactual values, which are not explicitly observable from data. Since \( \Delta \) can have a non-zero value only when \( M_t^{+} \) is true, the estimation task can be defined as: finding the counterfactual value for each patient who certainly has \( M_t^{+} \) at \( t - 1 \) and \( M_t^{-} \) at \( t \).

Intuitively, the so-called counterfactual value is the outcome value of a “counterfactual case” if it exists. Thus we can search through the entire dataset and find a record “matched” with the target record but turns out to choose the alternative medical plan at time \( t \), and then we take its value at time \( t + 1 \) to be the desired counterfactual \( X_{t+1}^{+} \). Ideally, the matched records pair should have identical values at \( t - 1 \), which are the conditions
for making medical decisions at \( t \). However, in practice, we need to figure out how to find a record as matched as possible.

In clinical studies, a conventional method, Propensity Score Matching (PSM), is widely used to isolate the unbiased treatment effects. Here “isolate” means selecting a subpopulation from the cohort, in which the different medical plans are decided in a uniform probability. In other words, the “randomized clinical trial” is strategically picked out from the actual patients’ data. To be specific, the selection strategy is to find pairs of matching records, both of which have the same chance to be assigned \( M^+_t \) given their similar \( X^+_{t-1} \), but only one turns out to be \( M^+_t \) while the other gets \( M^-_{t} \) at time \( t \).

**Definition 2.** Given record \( A \) with \( M^A_t \neq M^A_{t-1} \), record \( B \) is Counterfactual Record of \( A \), if and only if \( M^B_t = M^B_{t-1} = M^A_{t-1} \) and \( \mathbb{P}(M^A_t|X^A_{t-1}) \approx \mathbb{P}(M^A_t|X^B_{t-1}) \).

To simplify the notation, here we use \( M^A_t \) and \( X^A_t \) to denote medications vector and the record value of \( A \) at time \( t \), and as the same way, \( M^B_t \) and \( X^B_t \) are \( B \) notations. \( \mathbb{P} \) stands for probability; and the approximate equation means that \( A \) and \( B \) have their \( X_{t-1} \) values to be approximately same predictive about \( M^A_t \).

In this work, the conditional probability about \( M^A_t \) is estimated as a predictive logistic regression. The desired \( \Delta \) is calculated as \( \Delta_{t+1} = X^A_{t+1} \equiv X^B_{t+1} \). In practice, it is unnecessary to get an estimation for every possible \( \Delta_{t+1} \), which could be exponentially numerous. A meaningful \( \Delta \) is as to represent a treatment effect of interest. For the example in Definition 1, \( \Delta_{t+1} \) is estimated as the conditional effect of medical plan \( M^+_{t} \), given the ongoing medical plan \( M^-_{t} \).

### C. Functional Theory Analysis

This section will theoretically analyze the functional principle of the proposed method. At first, \( \Delta \) contains information that could be discoverable in sequentially learning. And then, \( \Delta \) has the nature to provide attention guidance to NNs, and also be intuitively configurable.

1) \( \Delta \) reveals new information: \( M^A_t \) is only discovered if \( M^A_t \neq M^A_{t-1} \), \( X^A_t \) is decomposed into two components \( S^N_t \) that models the natural progression and \( S^M_t \) that adjusts for medications. The right side of Fig.2 shows the disease progress over time with unobserved \( S^N_t \) and \( S^M_t \) involved. The observed data elements \( X_t \) and \( S_{t+1} \) represent the patients observation records at time \( t \) and \( t+1 \) respectively; and \( M_t \) is the medications vector, also referred as clinical treatments; The outcomes \( Y \) of interest are downstream from \( S^M_t \).

In the full model, the disease severity \( S_t \) is decomposed into two components \( S^N_t \) that models the natural progression and \( S^M_t \) that adjusts for medications. The right side of Fig.2 shows the correspondingly changed causation. This decomposition aims to isolate the severity adjustment \( S^M_t \) that medications \( M_{t-1} \) can only decided, and leaves \( S^N_t \) independent with \( M_{t-1} \), which is the definition of pure effect from treatment \( M_{t-1} \) upon the unobservable severity \( S_t \). In Fig.2 the element \( S^N_{t+1} \) is independent with \( M^+_t \) for any time \( t \).

**Fig. 2: Disease Progress Causal Graph.** The solid outline means element observable, and the dashed one means not. Here the node \( M^+_{t} \) stands for its binary indicator \( \mathbb{I}(M^+_{t}) \).

The observations \( X_t \) at the same time \( t \) are also composed of two parts \( X^M_t \) and \( X^N_t \), separated on the similar principle that \( X^N_t \) stands for the observed values naturally progressed without medication, and \( X^M_t \) adjusts the values for medicine applied.

Notably, the severity component \( S^N_t \) and \( S^M_t \) cannot be assertedly assumed independent of the observed values on the other side, i.e., \( X^M_t \) and \( X^N_t \), respectively. The decomposed causation has four links at most, as presented in Fig.2. For a straightforward view, we differentially color the \( S^N_t \)-started and the \( S^M_t \)-started ones as green and blue respectively. Moreover, we name the causal functions toward \( X^M_t \) as \( f \), while \( X^N_t \) as \( g \), and each one comprises two individual terms that \( f = (f_1, f_2) \) and \( g = (g_1, g_2) \). \( f_1, f_2, g_1, g_2 \) can be arbitrary functions, thus they represent the generalized model that does not require any prior assumption about distribution. However, not all of the four are necessary for the model.

**Theorem 1.** Function \( f_2 \) is constant, and \( X^M_t \perp \perp S^N_t \).

**Proof:** It has been known that \( X^M = f(S^A_t) = f_1(S^M_t) + f_2(S^N_t) \). And then, by definition we knows \( X^M = 0 \) iff \( S^M = 0 \). Therefore, we have \( f_2(S^N_t) = -f_1(0) \) to be constant, i.e. \( X^M \perp \perp S^N_t \).

According to Theorem 1, function \( f_2 \) is dashed in Fig.2 to indicate its unnecessary. We attempt to estimate observations at time \( t+1 \), including \( X^N_{t+1} \) and \( Y \), by only given the observable elements at time \( t \); at the same time, \( M^+_{t} \) is supposed to be happened, i.e. \( \mathbb{I}(M^+_{t}) = 1 \).

To simplify the estimation of \( X^N_{t+1} \), let us firstly convert \( X^N_{t+1} \) as an expression of \( X^N_{t+1} \), and then concentrate to estimate \( X^M_{t+1} \).

By definitions and Theorem 1, we know that \( X^M = f_1(S^M_t) \), and \( X^N = g_1(S^M_t) + g_2(S^N_t) = g(S_t) \). Thus we have the expression for \( X^N_t \) and also its isomorphic expression about \( X^N_{t+1} \):

\[
X^N_t = g(f_1^{-1}(X^M), S^N) \tag{1}
\]
\[
X^N_{t+1} = g(f_1^{-1}(X^M_{t+1}), S^N_{t+1}) \tag{2}
\]

Given \( \mathbb{I}(M^+_{t}) = 1 \), the non-zero valued \( S^M_{t+1} \) has decomposition as:

\[
S^M_{t+1} = \alpha S^M_t + \beta S^N_t \tag{3}
\]
where $\alpha$ and $\beta$ are the linear transformation matrices for describing two components of the severity temporal progression: one is from $S^M_t$ to $S^M_{t+1}$ and the other is from $S^N_t$ to $S^N_{t+1}$. Then we have:

$$X^M_{t+1} = f_1(\alpha S^M_t + \beta S^N_t) = \alpha X^M_t + f_1(\beta S^N_t)$$

that can be converted as the expression of $S^N_t$:

$$S^N_t = \frac{1}{\beta}f^{-1}_1(X^M_{t+1} - \alpha X^M_t) \quad (4)$$

By substituting Eq.(4) into Eq.(1) we have:

$$X^M_{t+1} = (\alpha - \beta)X^M_t + \beta f_1(g^{-1}(X^N_t)) \quad (5)$$

From Eq.(5) we know that $X^M_{t+1}$ is estimable given only the observable variables $X^M_t$ and $X^N_t$ without requiring the hidden values of severity. On the other hand, from Eq.(4) we find that the estimation of $X^N_t$ still needs value of $S^N_{t+1}$, whose isomorphic expression about $S^N_t$ is shown as Eq.(4). From Eq.(4) we can deduce that $S^N_{t+1}$ will become estimable when the following $X^M_{t+2}$ is observed, but not yet at the current time $t+1$.

In short, $X^M_{t+1}$ can be estimable if $X^M_t$ and $X^N_t$ are known, while $X^N_{t+1}$ is not directly estimable thus can only be deduced by $X^N_{t+1} = (X^M_{t+1} + X^M_{t+1})$ with known $X^M_{t+1}$ and an estimated $X^M_{t+1}$ at time $t+2$.

In practical prediction tasks, values of $X_t$, $X_{t+1}$, $X_{t+2}$, ... will successively become observed along the timeline. Therefore, for any time $t$ the value of $X^N_t$ will become estimable if and only if the splitting between $X^M_t$ and $X^N_t$ is known.

**Theorem 2.** $\Delta_t$ reveals necessary information for estimating the observable treatment effect $X^M_{t+1}$.

**Proof:** By Definition 1, we have

$$\Delta_{t+1} = (X^M_{t+1}\mathcal{M}^+ - X^M_{t+1}\mathcal{M}^-) = [X^M_{t+1} + X^N_{t+1}] - x^N_{t+1} = X^M_{t+1}$$

and subsequently have $\Delta_t = X^M_t$ due to the isomorphism.

This equation indicates $\Delta_t$ to be an effective estimation of $X^M_t$, and $X^N_t$ is equivalently estimated. Thus, $\Delta_t$ reveals the critical information for estimating the follow-up $X^M_{t+1}$ and $X^N_{t+1}$.

With the value of $\Delta_t$ estimated by our method, the performance of predicting outcome $Y$ depends on how capable the model is to learn the general causal functions $f_1, g, \alpha, \beta$, and also the ones started from severity $S$ toward $Y$. Notably, in both theorems we use $g$ as an integral function instead of comprising two separable components like $g = (g_1, g_2)$. That implies $(S^M, S^N)$ to be possibly associated and jointly predictive of $X^N$, without assuming independently with each other.

Compared to parametric regressions, DL methods are more potentially powerful to be improved performances since, without causality distribution assumptions, they are much better at modeling generalization.

2) $\Delta$ can guide NNs attention: .

Inherently, the proposed method faces the challenge of interesting causal effects selection. Per definition, $\Delta$ is estimable between a pair of medical plans that have an arbitrary subset of the medications to be different. Thus the theoretically possible $\Delta$ could be exponentially numerous if the considered medications vector is within a comparatively large length or dimension.

But in practical problems, causal effects of interest are more likely to be predefined and in a reasonable total number for intuitively understandable. In EHR data analysis, the primarily selected modeling objective is one specific or a few relative diseases, like a group of comorbidities; and the treatment effects of interest are often about adding the medications of interest, given the ongoing medical plan. So the number of meaningful $\Delta$ is practically limited.

Consequently, the estimated $\Delta$s will highly possible to be a sparse matrix, while the empty elements are left as zeros. The non-zero elements with larger absolute values indicate their corresponding treatment effects to be more comparatively significant, including both positively and negatively.

In the input layer of NNs, the sparse $\Delta$s attributes can very suitably work as the attention vector to guide the optimization processing toward more efficient direction and zero out the misleading or ignorable factors.

Furthermore, in case of inappropriate causal assumptions, $\Delta$ can be manually customized to avoid misled modeling. As in the examples mentioned above:

**[Example A].** At surgery time, zero out the BP relative digits in $\Delta$ to avoid wrong attention.

**[Example B].** The doctor can set the effects of adding insulin dose as positive or negative in $\Delta$.

**[Example C].** $\Delta$ can bring historical information since it is reflecting the changes that happened at $t-1$ .

V. EXPERIMENTAL DESIGN

The experiments aim to confirm $\Delta$ to be significantly informative by comparing the predictive performances with and without $\Delta$.

For convincing conclusions, we employed two type datasets to perform experiments: (1) a real-world EHR dataset of type-II diabetes patients; (2) multiple synthetic datasets generated with underlying causation, determined by random DAGs.

Due to the practical limitation of actual data (e.g., missing values), the advantage of $\Delta$ cannot be fully revealed on it. That’s why synthetic data is necessary.

This section will introduce the two types of datasets and the experimental tasks on them, respectively. The synthetic data generation method will be illustrated, and we have published the source code for free downloading.
TABLE I: Study Cohort Description

| Variable                  | Median | Interquartile Range |
|---------------------------|--------|---------------------|
| Age [years]               | 45     | 31, 59              |
| Male [%]                  | 43.4   |                     |
| LDL [mg/dL]               | 105    | 83, 130             |
| TG [mg/dL]                | 130    | 91, 187             |
| HDL [mg/dL]               | 50     | 41, 61              |
| SBP [Hg mm]               | 120    | 108, 132            |
| DBP [Hg mm]               | 70     | 60, 78              |
| FPG [mg/dL]               | 101    | 92, 117             |
| Follow-up [years]         | 10     | 6, 12               |
| Antihypertensive medication | 42.4  | 30968               |
| Antilipemic medication    | 32.8   | 23925               |
| Progressed to DM          | 16.9   | 12367               |
| Progressed to CAD         | 17.9   | 13065               |
| Progressed to CKD         | 7.2    | 5267                |
| Progressed to Stroke      | 1.1    | 779                 |

A. Real EHR Data

Mayo Clinic (MC) provides primary care to residents of Olmsted County, Minnesota, and has an integrated electronic health record system including diagnoses, medications, laboratory results, and clinical notes.

With research consent, we used a retrospective cohort of de-identified data from 73,045 primary care patients at Mayo Clinic, Rochester, MN. The cohort consists of patients aged ≥18 and ≤ 89 at baseline on Jan. 1st, 2005, having at least one visit before and after baseline. These patients were followed until 2017 (median follow-up time is ten years). We extracted patient demographics and diagnoses (ICD-9), laboratory results, vital signs, and medications longitudinally for six non-overlapping time windows: before-2004, 2005-2006, 2007-2008, 2009-2010, 2011-2012, 2013-after. For the outcomes that happened in time window t, we make predictions based on observations from previous time windows, i.e., t-1, t-2, etc. The latest observed values are taken for the predictors in each time window.

The patient cohort description is given in Table I.

1) Medications: All referred medications for T2DM and related comorbidities have been rolled up to National Drug File Reference Terminology NDF-RT pharmaceutical subclasses. There exist three medication classes: T2DM (DM) drugs, Hyperlipidemia (HL) drugs, and Hypertension (HTN) drugs. The table displays all three classes in increasing levels, where a higher level is commonly for higher severity. Each class is encoded as a binary vector with one digit representing one level, and we finally concatenate all three vectors to comprise the treatment indicator M.

2) Prediction Tasks: We comprehensively set up two types of tasks. One type is binary classification, and another one is continuous outcome regression, as below:

Task 1: Disease diagnosis predictions. The outcomes in this task include the disease diabetes (DM) and its three complications (CAD, CKD, Stroke). We aim to predict whether the disease developed in the subsequent time window, using predictors from the immediately preceding time windows.

For a brief, we refer to the number of preceding time windows as Time Step or Step. e.g., Step=1 means the predictors from the preceding time window (2011-2012); And Step=2 means from the preceding two time windows (2009-2010 and 2011-2012).

We perform four groups of experiments, with Time Step being 1, 2, 3, 4 respectively.

Task 2: Lab results forecasting. The outcomes in this task are continuous values, including three critical lab results in the immediately subsequent time window. Same as Task 1, four groups of experiments have Time Step = 1, 2, 3, 4 respectively.

3) Compared Models: We implemented two RNN architectures in a different way of using Δ: one is simply augmenting Δ as input, and another is pre-training with Δ, where the latter is expected to be more efficient to utilize Δ’s information. RNN-GRU and RNN-LSTM mechanisms are used, respectively. Besides, compared with RNNs, two linear models are implemented as baseline methods.

We obtain a pair of performances for each model: with and without Δ input, whose difference reflects Δ’s effect.

Model 1: Augmentation Architecture RNN. For each time window t, we concatenate X_t with Δ_t as new predictors, then use such augmented data as input to the RNN model.

Model 2: Unsupervised Pre-training RNN. An alternative approach to use Δ to guide the learning process of the RNN model: We pre-train the RNN such that the model predicts Δ_t+1 from X_t at each t. Since the pre-training does not require labels, i.e., unsupervised, the test data can also be used.

Model 3: Logistic Regression Model. Δ_t variables are simply augmented with X_t as new predictors.

Model 4: Linear Mixed-Effects Model. A standard method in EHR data analysis, as patients are usually categorized as subgroups, and a mixed-effect model is convenient to capture the group-level effect. Δ is also augmented as new predictors.

B. Synthetic EHR Data

1) Causal data Generation Method: The synthetic EHR data generation comprises three stages: I) Initialization, II) Disease Assignment, III) Treatment Assignment.

The detailed algorithms of the three stages are provided in Appendix A of this paper as supplementary material.

The three stages are for generating three different status of data, X_init, X_disc and X_trea sequentially, where everyone is based on the preceding one. But only the last matrix X_trea will be exported as the synthetic EHR data, and the front two are hidden ground truth. Δ matrix generated along with X_trea.
Stage I: Initialization. Randomly set up the healthy values for each lab result variable by a normal distribution, then initialize all patients’ healthy data $X_{\text{init}}$ accordingly.

Stage II: Disease Assignment. Randomly set up diseases and the DAG causal graph, then build up causality function from diseases to lab results, i.e., diseases influence labs value. Generate all patients’ diseased data $X_{\text{dise}}$ accordingly.

Stage III: Treatment Assignment. Randomly set up multiple medicine lines for each disease, and build up the causality function from medicines to the responded disease, i.e., treat-medicines lines for each disease, and build up the causality function. Randomly set up multiple and heart disease. Notably, the disease severity $d$ as a group of comorbidities, such as diabetes, hypertension, and heart disease. Typically, these diseases form a group of comorbidities, and with a pre-determined causal graph, $G$ exists at most 20 possible disease-developing paths (i.e., disease trajectories).

We set up 10 classes of medicines corresponding to the 10 diseases. Each class contains $1 \sim 3$ medicine levels, ordered with increasing impacts on lab value, including both positive typical and adverse side effects.

3) Prediction Tasks: The experiments on synthetic data aim to exhibit the predictive performance changing by varying hyper-parameter settings. The parameters include: 1) Time Step (i.e., the number of observed preceding years); 2) Training data sample size (i.e., the number of synthetic patients).

From these results, we can confirm the conditions in which the proposed $\Delta$ can be most effective, i.e., the most helpful to enhance the predictive performance. Additionally, we can observe whether involving $\Delta$ would influence the computing efficiency, i.e., running time of the training process.

For well concentrating, unlike in the actual data experiments, we only perform the binary classification tasks: To predict the 5-years-risk of the synthetic disease diagnosis, i.e., whether the patient will develop the disease in future 5 years.

We adopt AUC (Area under the ROC Curve) as the performance evaluation. For each model, we compare the two AUCs with and without $\Delta$ input. Additionally, the training time is also recorded and the comparison is reached at the end.

4) Compared Models:
- GLM, i.e. Logistic Linear Regression Model.
- RNN-GRU in Unsupervised Pre-training Architecture.
- RNN-LSTM in Unsupervised Pre-training Architecture.

Here we dropped the Linear Mixed-Effects Model and the $\Delta$-Augmentation RNNs, because they did not show outperformance compared to their brother methods.

We expect: 1) RNN-GRU and RNN-LSTM can outperform the linear model GLM in most of the tasks, especially with large sample size; 2) adding $\Delta$ input can significantly enhance predictive performance for both linear model and RNN model; 3) The training time is expected to be longer with $\Delta$ added, but not dramatically.

VI. Experimental Results

For real EHR data, all experiments are conducted for 20 times, with 20 randomly splitting of training and test. For synthetic EHR data, we independently generate 5 different data, and perform all experiments for 4 times on each one, with 4 randomly splitting.

All shown results are on averaged values over these multiple independent runs, and the corresponding conference intervals (CI) are also displayed, as the shadowed areas in figures.

The splitting rate is 0.7 : 0.3 for training and test.
Table III: Disease Diagnosis Prediction Performance of 4 Different Outcomes on Real EHR Dataset

| Methods | AUC Performance |
|---------|-----------------|
| GLM     |                 |
| Step=1  | No ∆ 0.783 0.7806** No ∆ 0.777 0.7883** No ∆ 0.7829 0.7005 0.8067*** No ∆ 0.7846 0.7019 0.8059*** |
| Step=2  | 0.8000 0.8041*** 0.7488 0.7529*** 0.8325 0.7387 0.8448*** 0.8437 0.7459 0.8558*** |
| Step=3  | 0.6867 0.6942*** 0.6483 0.6533*** 0.6887 0.6872 0.7083*** 0.6778 0.6665 0.7054*** |
| Step=4  | 0.6722 0.6756*** 0.6712 0.6758*** 0.6559 0.6845*** 0.6601*** 0.8564 0.6722* 0.6893*** |
| Step=5  | 0.7005 0.7102** 0.6998 0.7156*** 0.6060 0.6557 0.7061*** 0.6074 0.6600 0.7103*** |
| LSTM    |                 |
| Step=1  | No ∆ 0.7127 0.6418*** No ∆ 0.6259 0.6280*** No ∆ 0.6121 0.6310*** 0.6337*** No ∆ 0.6049 0.6320*** 0.6372*** |
| Step=2  | 0.6475 0.6527*** 0.6487 0.6528*** 0.6079 0.6418*** 0.6454*** 0.6059 0.6494*** 0.6486*** |
| Step=3  | 0.8154 0.8159 0.8154 0.8159 0.7685 0.6470 0.7877*** 0.7684 0.6469 0.7942*** |
| Step=4  | 0.8182 0.8200*** 0.7922 0.7814*** 0.7990 0.6652 0.8099*** 0.7899 0.6603 0.8147*** |
| Step=5  | 0.7816 0.7829* 0.7456 0.7469* 0.7352 0.6359 0.7590*** 0.7327 0.6661 0.7600*** |
| PreTrain |                 |
| Step=1  | 0.7187 0.7233*** 0.6707 0.6756*** 0.6548 0.6514 0.6707*** 0.6011 0.6113* 0.6064*** |
| Step=2  | 0.6409 0.6395 0.6409 0.6394 0.6165 0.6143 0.6754*** 0.6115 0.6242 0.6675*** |
| Step=3  | 0.6395 0.6388 0.6458 0.6482 0.6053 0.6118 0.6648*** 0.6011 0.6113* 0.6064*** |
| Step=4  | 0.6363 0.6443*** 0.6519 0.6412 0.6024 0.6233* 0.6614*** 0.6059 0.6104 0.6528*** |
| Step=5  | 0.6591 0.6565** 0.6593 0.6550*** 0.6012 0.6324* 0.7013*** 0.6044 0.6088*** 0.6882*** |

A. Results on Real Diabetes EHR Data

Table III and IV display the performances of binary diagnostic predictions (in AUC) and the continuous labs value forecasts (in MSE) respectively. Each row in the Table represents an individual task with a specified outcome and time step. The columns indicate applied methods, and the bold text in each row indicates the best performer. The star symbol signifies significance level of the improvement with ∆ participation, compared to the no ∆ version. The improvements are verified by one-sided paired t-tests, and significance level definitions are as p-value intervals: * * * for (−∞, 0.001], ** for (0.001, 0.01], * for (0.01, 0.05].

In Table III and IV, the RNN methods with ∆ pre-train architecture, including both RNN-GRU and RNN-LSTM (but especially LSTM), outperformed the others mostly. Specifically, they won 10 times out of 16 binary classification tasks and 10 times out of 12 continuous labs value forecasting tasks. On the other hand, RNN methods without ∆ input never outperformed the pre-training ones.

Except for RNNs, ∆ can also benefit linear models. Most of the time, with ∆ augmented, linear models performed better but not as significantly as RNNs.

B. Results on Synthetic EHR Data

Fig. 4: Comparison of the predictive performances with increasing numbers of samples for different Time Steps.
Fig. 4 shows AUC performances of the three compared models. The predicted outcomes are five randomly selected synthetic diseases, whose performances are equally weighted. For each model, with and without ∆ performances are in the same color but distinguished by solid and dashed line style. With varying Time Step=5, 10, 15, the sample size increases from 250 to 5000 gradually. Shadows in figure represent CI. As we expected, the performance with ∆ is always better than the one without ∆ input. And both RNN-GRU and RNN-LSTM consistently perform better than GLM when the sample size is large enough (in our case >= 500).

The hyper-parameters (Time Step and sample size) obviously influence the performance. With larger Time Step, all models tend to perform better, due to more historical information observed; With increasing sample size, RNN models present an upward trend on performance, but level off when sample size has been large enough. On the other hand, GLM model slightly performs the upward trend but much weaker than RNNs, especially with small time Step = 5.

Fig. 5: Performance gains by adding ∆ for three methods: GLM, GRU, and LSTM.

Figure 5 shows the effect of adding ∆. It includes the AUC gain of involving ∆, i.e. the difference of with-∆-AUC minus without-∆-AUC, and keeps consistent layout with Figure 4. With shrinking sample size (from 5000 to 250), RNN-LSTM clearly presents the trend that ∆ becomes more helpful on improving predictions; RNN-GRU trends the same but less obviously; And this pattern is almost vanished on GLM model.

On the other hand, increasing Time Step from 5 to 15 did not result in any monotonic tendency.

For more convincing conclusions, we performed paired t-test on each pair of AUC vectors. Each vector consists of the performances of all synthetic disease prediction tasks. The vector here has a length 80. The p-values are listed in Table IV and all turn out to be significant. Particularly, RNN-LSTM shows a clear pattern that when the sample size is comparatively small, ∆ can be more certainly to be helpful, which is consistent with the conclusion from Figure 5.

Among the three models, RNN-LSTM acts the most stably to present ∆’s effect changing. Because LSTM has the most complex architecture, and the largest number of parameters, it is more able to capture enough information from data than the other two. In contrast, the simplest model GLM failed to fully utilize the new information brought by ∆.

On the side of training efficiency, adding ∆ does not dramatically influence it. For GLM, augmentation of ∆ increased training time by around 80%, e.g. with sample size=5000 and Time Step=15, training without ∆ runs 10.8 secs by average and increased to 18.82 secs with ∆. RNN models spend hundreds of secs on the training process, and after pre-training using ∆, their training time reduced by around 1/3 because of prior knowledge. However, the entire time cost is almost doubled compared to the version without pre-training.

VII. CONCLUSION

In this paper, we proposed a novel method to quantitatively estimate the domain-defined treatment effects from EHR data as a generalized feature vector, which is simply utilisable for any properly selected machine learning methodology.

We demonstrated the theoretical analysis about the principle of our method and proved that the proposed estimator comes with new causal information, which can hardly be automatically discovered by regular RNN models. Furthermore, with comprehensive experiments we have displayed that our method can effectively help Deep Learning models to leverage clinical knowledge that is originally hidden in EHR data; and consequentially improved their predictive performance, especially when the sample size is comparatively small.

We believe that our works have taken the Deep Learning applications a step closer to a form of interpretable Artificial Intelligence integrated Healthcare.

REFERENCES

[1] Deigner, Hans-Peter, and Matthias Kohl, eds. Precision medicine: tools and quantitative approaches. Academic Press, 2018.
[2] Wu, Jionglin, Jason Roy, and Walter F. Stewart. Prediction modeling using EHR data: challenges, strategies, and a comparison of machine learning approaches. Medical care (2010): S106-S113.
[3] Glymour, Clark, Kun Zhang, and Peter Spirtes. "Review of causal discovery methods based on graphical models." Frontiers in genetics 10 (2019): 524.
[4] Luo, Yunan, Jian Peng, and Jianzhu Ma. "When causal inference meets deep learning." Nature Machine Intelligence 2.8 (2020): 426-427.
[5] Aczon, Melissa, et al. "Dynamic mortality risk predictions in pediatric critical care using recurrent neural networks." arXiv preprint arXiv:1701.06695 (2017).
[6] Shickel, et al. "Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis." IEEE journal of biomedical and health informatics 22.5 (2017): 1589-1604.
[7] Miotto, Riccardo, et al. "Deep learning for EHR-based disease risk prediction." Machine learning for healthcare conference. PMLR, 2016.
[8] Wickramasinghe, Nilmini. "DeepR: a convolutional net for medical records." (2017).
[9] Choi, et al. "Doctor ai: Predicting clinical events via recurrent neural networks." Machine learning for healthcare conference. PMLR, 2016.
[10] Esteban, Cristóbal, et al. "Predicting clinical events by combining static and dynamic information using recurrent neural networks." 2016 IEEE International Conference on Healthcare Informatics (ICHI). IEEE, 2016.
[11] Che, Zhengqing, et al. "Recurrent neural networks for multivariate time series with missing values." Scientific reports 8.1 (2018): 1-12.
[12] Choi, Edward, et al. "Multi-layer representation learning for medical concepts." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016.
APPENDIX A
SYNTHETIC EHR DATA GENERATION ALGORITHMS

The algorithms below are referred to in Section 5.2.1. And all used notations are listed in Table VI. Synthetic EHR data generation comprises three stages: I) Initialization, II) Disease Assignment, III) Treatment Assignment.

TABLE VI: Notations in Causal Data Generation Algorithm

| Notation | Description |
|----------|-------------|
| T | time windows |
| \( p \) | patient |
| \( L \) | lab result |
| \( D \) | disease |
| \( M \) | medication |
| \( e \) | event |
| \( G \) | causal graph |
| \( \Delta \) | causation mapping |

Algorithm 1: (Stage I) Initialization

Result: \((N_p \times T \times N_l)\) data matrix \(X_{init}\)

\[
\text{for } p = 1 \ldots N_p \text{ do}
\]
\[
\text{for } l = 1 \ldots N_l \text{ do}
\]
\[
\text{\quad Random } \mu, \sigma; \\
\text{\quad Random } x \sim N(\mu, \sigma), t = 1 \ldots T; \\
\text{\quad } X_{init}[p,l,t] = x;
\]

Algorithm 2: (Stage II) Disease Assignment

Result: \((N_p \times T \times N_l)\) data matrix \(X_{dise}\)

Initialize \(X_{dise} = X_{init}\);

Random graph \(G = (D, \{e\})\); // Diseases causal graph

Random \(L = F_d(D)\); // Causation from \(D\) to \(L\)

\[
\text{for } p = 1 \ldots N_p \text{ do}
\]
\[
\text{\quad Random path } g \text{ from } G; \\
\text{\quad for } d \in V(g) \text{ do}
\]
\[
\text{\quad Update } L \text{ by } L = F_d(d); \\
\text{\quad for } l \in D \text{ do}
\]
\[
\text{\quad Update } X_{dise}[p,l] \text{; }
\]

Algorithm 3: (Stage III) Treatment Assignment

Result: \((N_p \times T \times N_l)\) data matrix \(X_{trea}\) and \(\Delta\)

Initialize \(X_{trea} = X_{dise}\);

Initialize \(\Delta\) as zeros;

Random \(D = F_m(M)\); // Causation from \(M\) to \(D\)

\[
\text{for } m \in M \text{ do}
\]
\[
\text{Update } D \text{ by } D = F_m(m); \\
\text{for } d \in D \text{ do}
\]
\[
\text{Update } L \text{ by } L = F_d(d); \\
\text{\quad for } l \in D \text{ do}
\]
\[
\text{\quad Update } X_{trea}[p,l]; \\
\text{\quad Update } \Delta[p,l] = X_{trea}[p,l] - x_0;
\]

Stage I: Initialization. Randomly set up the healthy values for each lab result variable by a normal distribution, then initialize all patients’ healthy data \(X_{init}\) accordingly.

Stage II: Disease Assignment. Randomly set up diseases and the DAG causal graph of them, then build up causality function from diseases to lab results, i.e. diseases influence labs value. Generate all patients’ diseased data \(X_{dise}\) accordingly.

Stage III: Treatment Assignment. Randomly set up multiple medicine lines for each disease, and build up the causality function from medicines to the responded disease, i.e. treatments affect disease severity. Update all patients’ treated data \(X_{trea}\) accordingly, and in each iteration, record the difference of after and before updating to be the value of \(\Delta\).