SyncTwin: Treatment Effect Estimation with Longitudinal Outcomes

Anonymous Author(s)
Affiliation
Address
email

Abstract

Most of the medical observational studies estimate the causal treatment effects using electronic health records (EHR), where a patient's covariates and outcomes are both observed longitudinally. However, previous methods focus only on adjusting for the covariates while neglecting the temporal structure in the outcomes. To bridge the gap, this paper develops a new method, SyncTwin, that learns a patient-specific time-constant representation from the pre-treatment observations. SyncTwin issues counterfactual prediction of a target patient by constructing a synthetic twin that closely matches the target in representation. The reliability of the estimated treatment effect can be assessed by comparing the observed and synthetic pre-treatment outcomes. The medical experts can interpret the estimate by examining the most important contributing individuals to the synthetic twin. In the real-data experiment, SyncTwin successfully reproduced the findings of a randomized controlled clinical trial using observational data, which demonstrates its usability in the complex real-world EHR.

1 Introduction

Estimating the causal individual treatment effect (ITE) using observational data has become increasingly common in the medical literature due to the popularization of electronic health records (EHR). The EHR is a *longitudinal* collection of records: it contains repeated measurements of a patient's health condition over irregular time intervals. Although the treatments may also vary over time, many studies consider a *point treatment*, where the treatment allocation is performed at some observed time and stays fixed during the study [12, 42, 33, 5, 48]. Point treatment is widely applicable to problems involving one-off treatments (e.g. surgical operations) or treatments that do not change frequently (e.g. long term medication for chronic disease). We will refer to the setting above as Longitudinal and Irregularly sampled data with Point treatment setting, or *LIP*. The LIP setting will be the focus of this work (Figure 1 A).

The LIP setting is different from the conventional settings in causal inference. This is because we observe the *pre-treatment* outcome y_t over time leading to the treatment allocation. These pre-treatment outcomes may unveil the inherent temporal structure in the outcome time series (e.g. trend and seasonality), leading to better ITE estimation over time. In contrast, the conventional settings only consider the outcomes *after* treatment initiation [51, 49, 35, 41, 6]. This is true even for works that deal with dynamic treatment allocations [38, 34, 11].

A naive way to incorporate pre-treatment outcomes into the standard ITE methods is to treat them as additional covariates. This approach is viable but inadequate: the pre-treatment outcomes are arguably much more closely linked to the outcomes after treatment than the covariates — they hence deserve special considerations, e.g. modifying the architecture or loss function to reflect their importance.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

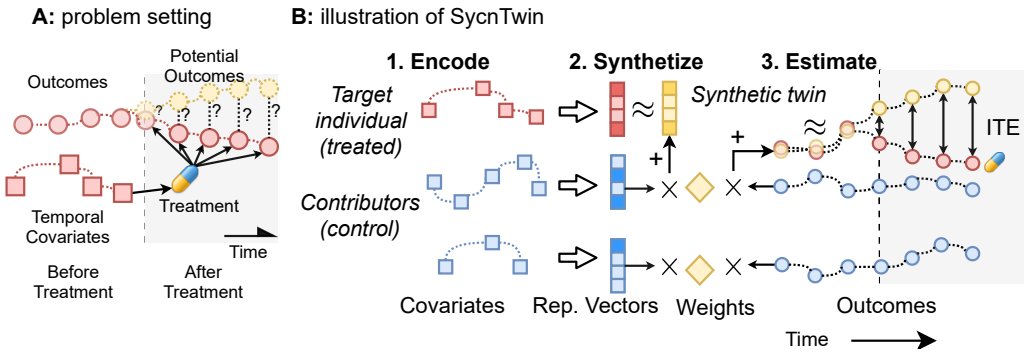

Figure 1: **A**: Illustration of the LIP setting. Yellow dots: the potential outcomes. **B**: Illustration of SyncTwin (shaded area: the time points after the treatment allocation). 1. Temporal covariates are encoded as representation vectors. 2. The synthetic twin of a treated target individual is constructed as the weighted average of the few contributors from the control group. 3. The difference between the observed outcome and the synthetic twin outcome estimates ITE. Similar procedures can be carried out for control or new individuals.

Without such modification, the ITE methods will not incorporate our prior belief on the importance of the pre-treatment outcomes, which may lead to worse performance.

To bridge this gap, we propose SyncTwin, a novel ITE estimation method tailored for the LIP setting. We assume that the temporal outcomes are generated by individualized latent variables and time-varying latent trends. This assumption is similar to the "factor model" assumption commonly used in Econometrics [1] and allows us to achieve the right balance between the parametric assumption and modeling flexibility. Figure 1 B illustrates the schematics of SyncTwin using an example with two treatment options (treated and control). SyncTwin first uncovers the individualized latent variables using representation learning. For a target individual, SyncTwin selects and weights a few contributors based on their latent variables and a sparsity constraint. It proceeds to construct a synthetic twin whose temporal outcomes are the weighted average of the contributors. Finally, the ITE is estimated as the difference between the target individual and the synthetic twin's outcomes.

Unlike conventional methods, SyncTwin does not use the learned latent variables to directly predict the outcomes; instead it uses these variables to find the contributors and their importance weights so as to construct the synthetic twin. This approach brings two bonus features; both are important to medical applications. (1) We can calculate an individualized estimation error bound based on the actual and the synthetic outcomes *before* treatment. In practice, the clinician can accept the recommended treatment when the error bound is below a threshold and resort to expert knowledge otherwise. (2) We can identify the most important contributors to an estimate as the ones who receive the highest weights. The clinician can further examine these contributors (e.g. whether they are indeed similar to the target individual) to validate and interpret the estimate.

Contributions. (1) We develop SyncTwin to leverage the temporal structure in the outcomes for better ITE estimation. (2) We provide theoretical justifications for each step involved in SyncTwin and prove an individualized error bound that can be used to identify untrustworthy estimates. (3) In addition to extensive simulations, we conduct an observational study using real EHR data and successfully reproduced the findings of a clinical trial with SyncTwin. This provides solid evidence for the usability of SyncTwin in real-world medical research.

2 Problem setting

We consider an observational study with N individuals indexed by $i \in [N] = \{1, \ldots, N\}$. Let $\mathrm{a}_i \in \{0, 1\}$ be the treatment indicator with $\mathrm{a}_i = 1$ if i received the treatment at some time and $\mathrm{a}_i = 0$ otherwise[1]. We realign the time steps such that all treatments were initiated at time $t = 0$. Let $\mathcal{I}_1 = \{i \in [N] \mid \mathrm{a}_i = 1\}$ and $\mathcal{I}_0 = \{i \in [N] \mid \mathrm{a}_i = 0\}$ be the set of the treated and the control individuals respectively. Denote $N_0 = |\mathcal{I}_0|$ and $N_1 = |\mathcal{I}_1|$ as the sizes of the groups.

[1] SyncTwin can model more than two treatment options, but we focus on binary treatments for illustration.

Let $\mathbf{X}_i = [\mathbf{x}_{is}]_{s\in[S_i]}$ be the temporal covariates consisting of a sequence of observations $\mathbf{x}_{is} \in \mathbb{R}^D$. Let $\mathbf{M}_i = [\mathbf{m}_{is}]_{s\in[S_i]}$ be the sequence of binary masking vectors $\mathbf{m}_{is} \in \{0,1\}^D$, where $[\mathbf{m}_{is}]_d = 1$ indicates the dth element of $\mathbf{x}_{is}$ is measured and $[\mathbf{m}_{is}]_d = 0$ otherwise. The entire sequence $\mathbf{X}_i$ contains $S_i \in \mathbb{N}$ observations taken before treatment at times $\mathcal{T}_i = [t_{is}]_{s\in[S_i]}$, $t_{is} < 0$. The maximum sequence length $S = \max_i(S_i)$.

The outcome $\mathrm{y}_{it} \in \mathbb{R}$ is observed at time $t \in \mathcal{T}^- \cup \mathcal{T}^+$. Let $\mathcal{T}^- = \{-M, \ldots, -1\}$ and $\mathcal{T}^+ = \{0, \ldots, H-1\}$ be the observation times before and after treatment allocation. We arrange the outcomes after treatment into a H-dimensional vector denoted as $\mathbf{y}_i = [\mathrm{y}_{it}]_{t\in\mathcal{T}^+} \in \mathbb{R}^H$. Similarly define pre-treatment outcome vector $\mathbf{y}_i^- = [\mathrm{y}_{it}]_{t\in\mathcal{T}^-} \in \mathbb{R}^M$.

Using the potential outcome framework [39], let $\mathrm{y}_{it}(\mathrm{a}_i) \in \mathbb{R}$ denote the potential outcome at time t in a world where i received the treatment as indicated by a_i. Let $\mathbf{y}_i(1) = [\mathrm{y}_{it}(1)]_{t\in\mathcal{T}^+} \in \mathbb{R}^H$, and $\mathbf{y}_i^-(1) = [\mathrm{y}_{it}(1)]_{t\in\mathcal{T}^-} \in \mathbb{R}^M$. Similarly, let $\mathbf{y}_i(0) = [\mathrm{y}_{it}(0)]_{t\in\mathcal{T}^+}$ and $\mathbf{y}_i^-(0) = [\mathrm{y}_{it}(0)]_{t\in\mathcal{T}^-}$. The individual treatment effect (ITE) is defined as $\tau_i = \mathbf{y}_i(1) - \mathbf{y}_i(0) \in \mathbb{R}^H$.

3 ITE estimation via SyncTwin

SyncTwin estimates the ITE by predicting both potential outcomes $\mathbf{y}_i(1)$ and $\mathbf{y}_i(0)$ and take the difference. Note that for an individual in the dataset $i \in [N]$, it is only necessary to predict the counterfactual outcome $\mathbf{y}_i(1-\mathrm{a}_i)$ because the factual outcome $\mathbf{y}_i(\mathrm{a}_i) = \mathbf{y}_i$ is observed (under the consistency assumption to be discussed later). Without loss of generality, we will discuss the way to estimate $\mathbf{y}_i(0)$ as an example in the following sections. One can estimate $\mathbf{y}_i(1)$ using the same method.

3.1 Assumptions

SyncTwin relies on three assumptions. **(1)** *Stable Unit Treatment Value Assumption* [39]: $\mathrm{y}_{it}(\mathrm{a}_i) = \mathrm{y}_{it}$, $\forall i \in [N]$, $t \in \mathcal{T}^- \cup \mathcal{T}^+$. **(2)** *No anticipation*, also known as causal systems [4]: $\mathrm{y}_{it} = \mathrm{y}_{it}(1) = \mathrm{y}_{it}(0)$, $\forall t \in \mathcal{T}^-, i \in [N]$. **(3)** *Data generating process (DGP)*. The DGP assumption involves two parts. First, the causal directed acyclic graph (DAG) is specified in Figure 2 [36], which involves a latent variable $\mathbf{c}_i \in \mathbb{R}^K$. Secondly, we assume the potential outcomes have a parametric form:

$$\mathrm{y}_{it}(0) = \mathbf{q}_t^\top \mathbf{c}_i + \xi_{it}, \quad \forall t \in \mathcal{T}^- \cup \mathcal{T}^+, \tag{1}$$

where $\mathbf{q}_t \in \mathbb{R}^K$, $K < \min(M, H)$ is a weight vector and ξ_{it} is the white noise. Equation 1 is commonly referred to as a *latent factor model* with $\mathbf{c}_i$ as the individualized latent variable and $\mathbf{q}_t$ as the time-varying latent trend [10]. Without loss of generality, we require the weight vectors $||\mathbf{q}_t|| = 1$, $\forall t \in \mathcal{T}^- \cup \mathcal{T}^+$ [47].

Discussion. SyncTwin differs from the nonparametric ITE estimation methods [27] because it assumes the parametric form in Equation 1. Importantly, since the time-varying latent trend $\mathbf{q}_t$ is common to all individuals, any linear combination of different individuals' outcomes will automatically preserve $\mathbf{q}_t$, i.e. $\sum_i b_i \mathrm{y}_{it}(0) = \mathbf{q}_t^\top \sum_i b_i \mathbf{c}_i + \sum_i b_i \xi_{it}$, for any weights $b_i \in \mathbb{R}$, $\forall i \in [N]$. For this reason, SyncTwin attempts to find the weights b_i's to issue counterfactual predictions rather than making prediction directly from the learned representation. As we will show later, doing so allows us to derive various theoretical results to inform the design of SyncTwin . It also leads to a checking procedure to validate estimation quality and improvements in interpretability.

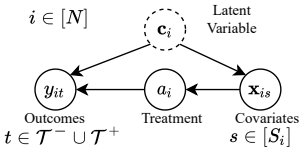

Figure 2: The DAG of the data generating model.

The DAG in Figure 2 is different from the one commonly used in the static setting [39]. We assume the latent variable $\mathbf{c}_i$ captures the physiological factors that impact both the outcome and the covariates. Here, the covariates $\mathbf{x}_{is}$ is a "confounder" in a general sense because it opens up a backdoor path from treatment to outcome, i.e. $a_i \to \mathbf{x}_{is} \to \mathbf{c}_i \to \mathrm{y}_{is}$ [36]. Based on the backdoor criterion [36], adjusting for the covariates $\mathbf{x}_{is}$ is sufficient to identify the treatment effect from observational data.

We compare our assumptions with those used in the related works in Appendix A.3. In particular, we show that our assumption is weaker than the one used by Synthetic Control, a widely-applied method in Econometrics [2]. In Appendix A.5, we further demonstrate the plausibility of our assumptions in real-world scenarios. In the simulation study (Section 5.1) we show experimentally that SyncTwin performs well even when the data is not generated from the assumed DGP exactly but instead from a set of differential equations. Finally, and perhaps most importantly, we show that based on our assumptions, SyncTwin is able to reproduce the findings of a large-scale randomized controlled trial using observational data (Section 5.2).

3.2 Learning to represent temporal covariates

The latent variable $\mathbf{c}_i$ plays an important role as it affects the covariates $\mathbf{x}_{is}$ and the outcomes y_{it}. SyncTwin uses deep neural networks to learn the representation $\tilde{\mathbf{c}}_i$ as a proxy for $\mathbf{c}_i$.

SyncTwin is agnostic to the exact choice of architecture as long as the network translates the irregularly sampled temporal covariates into a fixed-sized representation vector. For this reason, we use the well-proven sequence-to-sequence architecture (Seq2Seq) [44] with a standard attentive encoder [9] and a LSTM decoder [26]. The learned representation $\tilde{\mathbf{c}}_i = f_e(\mathbf{X}_i, \mathbf{M}_i, \mathcal{T}_i; \theta_e)$, where f_e is the encoder with trainable weights θ_e. The reconstructed covariates $\hat{\mathbf{X}}_i = f_d(\tilde{\mathbf{c}}_i, \mathcal{T}_i; \theta_d)$, where f_d is the decoder with trainable weights θ_d. To ensure the learned representation $\tilde{\mathbf{c}}_i$ is a *linear* predictor of the potential outcome $\mathbf{y}_i(0)$ (Equation 1), we introduce a trainable parameter $\tilde{\mathbf{Q}} \in \mathbb{R}^{H \times K}$ and define $\tilde{\mathbf{y}}_i(0) := \tilde{\mathbf{Q}} \cdot \tilde{\mathbf{c}}_i$. Note that using a nonlinear function to map from $\tilde{\mathbf{c}}_i$ to $\tilde{\mathbf{y}}_i(0)$ will be *inconsistent* with the DGP and will not uncover the latent variable $\mathbf{c}_i$ as desired.

We train the networks end-to-end by optimizing the loss function $\mathcal{L}^{tr} = \lambda_r \mathcal{L}_r + \lambda_p \mathcal{L}_s$, where λ_r and λ_p are hyper-parameters[2] balancing the supervised loss $\mathcal{L}_s$ and the reconstruction loss $\mathcal{L}_r$:

$$\mathcal{L}_s(\mathcal{D}_0) = \sum_{i \in \mathcal{D}_0} ||\tilde{\mathbf{y}}_i(0) - \mathbf{y}_i(0)||^2; \quad \mathcal{L}_r(\mathcal{D}_0, \mathcal{D}_1) = \sum_{i \in \mathcal{D}_0 \cup \mathcal{D}_1} ||(\tilde{\mathbf{X}}_i - \mathbf{X}_i) \odot \mathbf{M}_i||^2, \tag{2}$$

where $\mathcal{D}_0 \subseteq \mathcal{I}_0$ and $\mathcal{D}_1 \subseteq \mathcal{I}_1$ are training data, $\mathbf{m}_{is}$ is the masking vector, $\odot$ represents element-wise product and $|| \cdot ||$ is the L_2 norm. In Proposition 1, we show that minimizing the supervised loss $\mathcal{L}_s$ will reduce the error bound on the learned representations, making them closer to the true latent variable $\mathbf{c}_i$. The proof and the motivations for $\mathcal{L}_r$ are shown in A.1.1.

Proposition 1 (Error bound on the learned representations)**.** *Under the assumptions in Section 3.1, the total error on the learned representations for the control is bounded by:*

$$\sum_{j \in \mathcal{I}_0} \|\mathbf{c}_j - \tilde{\mathbf{c}}_j\| \leq \beta \mathcal{L}_s + \sum_{j \in \mathcal{I}_0} \|\xi_j\|, \tag{3}$$

where $\mathcal{L}_s$ is the supervised loss, β is a constant depending on $\mathbf{q}_t$ and $\tilde{\mathbf{Q}}$, and ξ_j is the white noise (Equation 1 and 2).

3.3 Synthesizing the twin

At this point, one may be temped to predict the counterfactual outcome $\mathbf{y}_i(0)$ of a treated individual $i \in \mathcal{I}_1$ with the output of the neural network $\tilde{\mathbf{y}}_i(0)$. However, $\tilde{\mathbf{y}}_i(0)$ is issued by a black-box neural network, which is not easily interpretable. As a remedy, SyncTwin explicitly constructs a synthetic twin who matches the target in representations. As we will show, this approach is able to control estimation bias and is more interpretable.

The synthetic twin of a target individual is defined by a set of weights $\boldsymbol{b}_i := [b_{ij}]_{j \in \mathcal{I}_0} \in \mathbb{R}^{N_0}$, each associated with a contributor in the control group $j \in \mathcal{I}_0$ (or treatment group when predicting $\hat{\mathbf{y}}_i(1)$). SyncTwin solves the following optimization problem to find the weights:

$$\boldsymbol{b}_i = \arg\min_{\tilde{\boldsymbol{b}}_i} \|\tilde{\mathbf{c}}_i - \sum_{j \in \mathcal{I}_0} \tilde{b}_{ij} \tilde{\mathbf{c}}_j\|^2 \quad \text{s.t. } \tilde{b}_{ij} \geq 0, \ \forall j \in \mathcal{I}_0 \text{ and } \sum_{j \in \mathcal{I}_0} \tilde{b}_{ij} = 1, \tag{4}$$

where $\tilde{\mathbf{c}}_j$ is the representation learned by the encoder. Denote the loss function in Equation 4 as $\mathcal{L}_m$. We will call $\mathcal{L}_m$ the *matching loss* because it indicates how well the synthetic twin matches the

[2]The hyperparamter sensitivity is studied in A.13. λ_r and λ_p do not significantly impact the performance.

target individual in representations. Let d_i^c denote the optimal value of $\mathcal{L}_m$ and let $\hat{\mathbf{c}}_i$ be the synthetic representation given by the solution $\boldsymbol{b}_i$:

$$\mathrm{d}_i^c := \|\tilde{\mathbf{c}}_i - \sum_{j \in \mathcal{I}_0} b_{ij} \tilde{\mathbf{c}}_j\|^2; \quad \hat{\mathbf{c}}_i := \sum_{j \in \mathcal{I}_0} b_{ij} \tilde{\mathbf{c}}_j \tag{5}$$

The optimization procedure is detailed in Appendix A.8. SyncTwin predicts the potential outcome $\mathrm{y}_{it}(0)$ for a target individual i using the same weights $\boldsymbol{b}_i$: $\forall t \in \mathcal{T}^- \cup \mathcal{T}^+$,

$$\hat{\mathrm{y}}_{it}(0) = \sum_{j \in \mathcal{I}_0} b_{ij} \mathrm{y}_{jt}(0) = \sum_{j \in \mathcal{I}_0} b_{ij} \mathrm{y}_{jt}, \tag{6}$$

where the last equality follows from the consistency assumption (Section 3.1). Denote $\hat{\mathbf{y}}_i(0) = [\hat{\mathrm{y}}_{it}(0)]_{t \in \mathcal{T}^+}$ as the predicted potential outcomes after treatment initiation.

The estimator in Equation 6 follows directly from the DGP assumption (Equation 1). To see this, remember that $\mathrm{y}_{it}(0)$ is *linear* in the latent variable $\mathbf{c}_i$. Hence, if we find a set of weights b_{ij}^* to match the latent variable, i.e. $\mathbf{c}_i = \sum_j b_{ij}^* \mathbf{c}_j$, the same set of weights will also match the outcome $\mathrm{y}_{it}(0) \approx \sum_j b_{ij}^* \mathrm{y}_{jt}(0)$ up to random noise. In practice, since we do not observe $\mathbf{c}_i$, we have to perform matching on the learned representation $\tilde{\mathbf{c}}_i$ as in Equation 4.

Therefore we can see that the quality of the estimator $\hat{\mathrm{y}}_{it}(0)$ depends on two things: **(1)** whether the learned representation $\tilde{\mathbf{c}}$ is close to the true latent variable $\mathbf{c}$ and **(2)** whether good matching weights $\mathbf{b}_i$ can be found on the learned representation. SyncTwin attempts to fulfill the first condition by representation learning in Section 3.2. The second condition is facilitated by solving the optimization problem in Equation 4. Proposition 2 formalizes this intuition and shows that when a perfectly-matching twin is found, the estimation error only depends on the quality of the learned representations (proved in A.1.1).

Proposition 2 (Bias bound on counterfactual prediction)**.** *Suppose that* $\mathrm{d}_i^c = 0$ *for some* $i \in \mathcal{I}_1$ *(Equation 1 and 5), the absolute value of the counterfactual prediction bias for* i *is bounded by:*

$$|\mathbb{E}[\hat{\mathbf{y}}_i(0) - \mathbf{y}_i(0)]| \leq |\mathcal{T}^+| \Big(\|\mathbf{c}_i - \tilde{\mathbf{c}}_i\| + \sum_{j \in \mathcal{I}_0} \|\mathbf{c}_j - \tilde{\mathbf{c}}_j\| \Big).$$

Notes on Interpretability. In addition to reducing bias by learning and matching representations, SyncTwin is also more interpretable. In particular, the weights $\boldsymbol{b}_i$ can be interpreted as the "contribution" or "importance" of a contributor j to the target i due to the optimization constraints. We can create a shortlist of the most important contributors based on $\boldsymbol{b}_i$. A domain expert can understand the rationale behind the estimate by examining the shortlist. For instance, they can check whether the important contributors share similar disease progression patterns as the target individual.

3.4 Calculating individualized error bound with pre-treatment outcomes

Since Equation 6 applies to all time points before and after treatment, let $\hat{\mathbf{y}}_i^-(0) = [\hat{\mathrm{y}}_{it}(0)]_{t \in \mathcal{T}^-}$ be the predicted *pre-treatment* potential outcome vector. We define d_i^y as the estimation error in the pre-treatment period:

$$\mathrm{d}_i^y = \|\hat{\mathbf{y}}_i^-(0) - \mathbf{y}_i^-(0)\|_1 = \|\hat{\mathbf{y}}_i^-(0) - \mathbf{y}_i^-\|_1, \tag{7}$$

where $|| \cdot ||_1$ is the vector ℓ_1-norm. The second equality allows d_i^y to be evaluated for all individuals. It holds because of the no anticipation assumption (Section 3.1). Achieving a small error d_i^y implies that the synthetic twin $\hat{\mathbf{c}}_i$ matches well with the *true* latent variable $\mathbf{c}_i$ due to the linearity between $\mathbf{y}_i^-(0)$ and $\mathbf{c}_i$ (Equation 1). Since $\mathbf{c}_i$ is assumed to be constant over time, the pre-treatment error can be used to assess the post-treatment error. We formalize this intuition in Proposition 3 and use d_i^y to control the error in the post-treatment period (proved in A.1.1).

Proposition 3 (Error control under no hidden confounders)**.** *Given any target error threshold* $\delta > 0$*, define the acceptance group of treated individuals as*

$$\mathcal{A}_\delta = \left\{ i \in \mathcal{I}_1 | \mathrm{d}_i^y \leq \delta |\mathcal{T}^-|/|\mathcal{T}^+| \right\}.$$

Under the assumptions in Section 3.1, the post-treatment estimation error $|\mathbb{E}[\hat{\mathbf{y}}_i(0)] - \mathbb{E}[\mathbf{y}_i(0)]| \leq \delta$*,* $\forall i \in \mathcal{A}_\delta$*.*

Table 1: **Problem settings considered in the literature**. "Static": observed (or allocated) only once; "Regular": observed (or allocated) over time at a regular frequency; "Irregular": observed over time irregularly; "-": not observed or modeled. * can be extended to Irregular. † can be extended to Regular. LIP: Longitudinal, Irregular, Point treatment.

Setting	Example	Pre-treatment $\mathbf{X}$	Pre-treatment $\mathbf{y}^-$	Treatment a	Post-treatment $\mathbf{y}$	Nonlinear f: $\mathbf{y} = f(\mathbf{X})$
Static	[41]	Static*	-	Static	Static†	✓
DT	[11]	Regular*	-	Regular	Regular	✓
SC	[2]	Regular	Regular	Static	Regular	×
LIP (This work)	This work	Irregular	Regular	Static	Regular	✓

Proposition 3 shows that we can control the estimation error to be below a certain threshold δ by rejecting the estimate if its error d_i^y during the pre-treatment period is larger than $\delta|\mathcal{T}^-|/|\mathcal{T}^+|$. Alternatively, we can rank the estimation trustworthiness for the individuals based on d_i^y. This is helpful when the user is willing to accept a percentage of estimations which are deemed most trustworthy.

3.5 Training, validation and inference

We perform model training, validation and inference (testing) on three disjoint datasets. In summary, we train the encoder and decoder on the training data using the loss functions described in Section 3.2. The validation data is then used to validate and tune the hyper-parameters of the encoder and decoder. Then, we solve the optimization problem in Section 3.3 to find the weights $\mathbf{b}_i$ for the target individuals in the testing data. Finally, we compute the potential outcome estimator (Section 3.3) and the individualized error bound (Section 3.4). The pseudocode is described in A.7.

4 Related work

Synthetic Control. Similar to SyncTwin, Synthetic control (SC) [1] and its extensions [8, 7] estimate ITE based on synthetic outcomes. However, SC needs to *flatten* the temporal covariates $[\mathbf{x}_{is}]_{s\in[S_i]}$ into a fixed-sized (high-dimensional) vector $\underline{\mathbf{x}}_i$ and use it to construct the twin. As a result, SC does not allow the covariates to be variable-length or sampled at irregular frequencies (Table 1); otherwise $\underline{\mathbf{x}}_i$'s dimensionality will vary across individuals. This severely limits SC's usability in medical observational studies because such sampling irregularities are common in EHR. Moreover, SC assumes $\mathrm{y}_{it}(0) = \mathbf{q}_t^\top \underline{\mathbf{x}}_i + \xi_{it}$, i.e. the flattened covariates $\underline{\mathbf{x}}_i$ *linearly* predicts $\mathrm{y}_{it}(0)$, which is a special case of SyncTwin's assumption (to be discussed later in Equation 1) and seems unlikely in many medical applications (e.g. body mass index does not linearly relate to blood pressure).

Static nonparametric ITE estimation. Most ITE estimation methods in the literature consider the static setting and do not pay special attention to the outcomes before treatment $\mathbf{y}_i^-$ — these outcomes are treated as standard covariates $\mathbf{X}_i$, if at all (Table 1). However, the pre-treatment outcomes $\mathbf{y}_i^-$ may encode certain temporal structure that will persist even on the outcomes $\mathbf{y}_i$ after treatment (e.g. trend and seasonality). Explicitly unveiling such temporal structure will be very beneficial to accurately estimating and validating the ITE after treatment. Moreover, these methods make no parametric assumption on the data generating model [27] and they learn to predict the two potential outcomes using deep neural networks [41, 51, 22]. Despite the recent progress in interpretable machine learning, these methods are still in general less interpretable than the parametric models such as SyncTwin whose coefficients can be directly understood by an expert.

ITE estimation for dynamic treatments. Several works consider ITE estimation for dynamic treatments (DT) [38]. In the DT setting (Table 1), *multiple* treatment decisions are made over time for each individual. ITE estimation is much more challenging for DT because a prior treatment a_{t-1} may influence the temporal covariates $\mathbf{x}_t$, which may in turn confound a latter treatment a_t (known as *temporal confounding*). Marginal structural model [38] and related works [40, 34, 11] have been proposed to overcome temporal confounding. In comparison, our setting considers a point treatment and therefore do not suffer from temporal confounding. As a result, although the works in the DT

setting are applicable, they might be over-complicated for our problem. In addition, our previous discussion about the inadequacy in leveraging pre-treatment outcomes also applies to DT methods.

Works with similar terminology. Several *unrelated* works in the literature use terms such as "twin", which might cause confusion. We discuss these works in Appendix A.4.

5 Experiments

5.1 Simulation Study

Data generation. In this simulation study, we evaluate SyncTwin on the task of estimating the LDL cholesterol-lowering effect of statins, a common drug prescribed to hypercholesterolaemic patients. Following our convention, the individuals are enrolled at $t = 0$, the covariates are observed in $\mathcal{T} = [-S, 0)$, where $S \in \{15, 25, 45\}$, and the ITE is to be estimated in the period $\mathcal{T}^+ = [0, 4]$. We start by generating a covariate k_t^{in} for from a mixture distribution:

$$\mathrm{k}_{it}^{in} = \delta_i \mathbf{f}_{it} + (1 - \delta_i)\mathbf{g}_{it}; \quad \delta_i \overset{\text{iid}}{\sim} \text{Bern}(p_i) \tag{8}$$

where $\text{Bern}(p)$ is the Bernoulli distribution with success probability p and $\mathbf{f}_{it},\ \mathbf{g}_{it}$ are drawn from two different distributions (specified in A.10). To introduce confounding bias, we vary p_i for the treated and the control: $p_i = p_0, \forall i \in \mathcal{I}_0$ and $p_i = 1, \forall i \in \mathcal{I}_1$. The constant p_0 controls the degree of confounding bias (smaller p_0, larger bias). We simulate the outcome y_t using the widely adopted Pharmacology model in the literature [16, 50, 31]. It is obtained by solving the differential equation 9 and adding independent white noise $\epsilon \sim \mathrm{N}(0, 0.1)$.

$$\dot{\mathrm{p}}_t = \mathrm{k}_t^{in} - k \cdot \mathrm{p}_t; \quad \dot{\mathrm{d}}_t = \mathrm{a}_t - h \cdot \mathrm{d}_t; \quad \dot{\mathrm{y}}_t = k \cdot \mathrm{p}_t - k\mathrm{d}_t(\mathrm{d}_t + d_{50})^{-1} \cdot \mathrm{y}_t. \tag{9}$$

where $\dot{\mathrm{y}}_t$ is the derivative of y_t and a_t is the indicator of statins treatment. The covariates includes $\mathbf{x}_t = \{\mathrm{k}_t^{in}, \mathrm{y}_t, \mathrm{p}_t\}$. The interpretation of all constants involved are presented in Appendix A.10. Finally, we introduce irregular sampling by creating masks $\mathbf{m}_{it} \sim \text{Bern}(m)$, where probability $m \in \{0.3, 0.5, 0.7, 1\}$. It is worth highlighting that the simulation data are not generated from SyncTwin's assumed DGP (1) but from domain-specific ODEs (9).

Benchmarks. From the Synthetic Control literature, we considered the original Synthetic Control method (SC) [2], Robust Synthetic Control (RSC) [7] and MC-NNM [8]. From the dynamic treatment literature, we compared against Counterfactual Recurrent Network (CRN) [11], Recurrent Marginal Structural Network (RMSN) [34], and counterfactual Gaussian Process (CGP) [40]. We included a modified CFRNet, which was originally developed for the static setting [41]. We replaced its fully-connected encoder with the encoder used by SyncTwin to to allow CFRNet to model temporal covariates (Section 3.2). We use One-nearest Neighbour Matching (1NN) [43] as a baseline. The implementation details of all benchmarks are available in Appendix A.9. We compared with two ablated versions of SyncTwin. SyncTwin-$\mathcal{L}_r$ is trained only with reconstruction loss and SyncTwin-$\mathcal{L}_s$ only with supervised loss.

Main results. We evaluate the mean absolute error (MAE) on counterfactual prediction: $\frac{1}{N_1}\sum_{i=1}^{N_1} ||\mathbf{y}_i(0) - \hat{\mathbf{y}}_i(0)||_1$. Table 2 presents the results for various levels of confounding bias p_0. Additional results for different sequence length S and sampling irregularity m are shown in Appendix A.11. SyncTwin achieves the best or equally-best performance in all cases despite the assumed DGP does not exactly hold. The full SyncTwin with both loss functions also consistently outperforms the versions trained only with $\mathcal{L}_r$ or $\mathcal{L}_s$. In comparison, SC, RSC and MC-NNM underperform because their assumption that the flattened covariates $\underline{\mathbf{x}}_i$ *linearly* predict the outcome is violated (Section 4).

Error bound. SyncTwin allows the user to reject untrustworthy estimates based on d_i^y. In Figure 3 (A) below, we show the scenarios of using various rejection thresholds on d_i^y. As expected from Proposition 3, when the user increases the threshold, more estimates will be accepted but they will have a higher MAE. The fact that the two curves in the figure share the similar increasing trend suggests d_i^y is a good indicator of individual error: rejection based on a non-informative criterion will lead to a flat line of MAE. In practice, the user can use Figure 3 (A) to calibrate the threshold in order to achieve balance between the accuracy and the workload. For instance, if the user would like SyncTwin to accept 75% of the estimate, he or she should set threshold $\delta = 0.12$ according to the pre-treatment error d_i^y and expect a MAE of around 0.105 for the post-treatment outcome.

Table 2: Mean absolute error on ITE under different levels of confounding bias p_0. $m = 1$ and $S = 25$ are used. Estimated standard deviations are shown in the parentheses. The best performer is in bold. Additional quantitative results are shown in Appendix A.11

Method	$N_0 = 200$			$N_0 = 1000$		
	$p_0 = 0.1$	$p_0 = 0.25$	$p_0 = 0.5$	$p_0 = 0.1$	$p_0 = 0.25$	$p_0 = 0.5$
SyncTwin-Full	**0.324 (.038)**	**0.144 (.012)**	**0.119 (.008)**	**0.141 (.012)**	**0.106 (.006)**	**0.093 (.005)**
SyncTwin-$\mathcal{L}_r$	0.353 (.039)	0.170 (.015)	0.139 (.010)	0.256 (.026)	0.145 (.012)	0.101 (.006)
SyncTwin-$\mathcal{L}_s$	0.336 (.039)	0.170 (.015)	0.120 (.008)	0.144 (.012)	0.113 (.007)	0.127 (.010)
SC	0.340 (.041)	0.151 (.024)	0.149 (.018)	0.258 (.050)	0.166 (.034)	0.214 (.036)
RSC	0.837 (.044)	0.360 (.020)	0.321 (.018)	0.310 (.016)	0.298 (.014)	0.302 (.014)
MC-NNM	1.160 (.059)	0.612 (.031)	0.226 (.011)	0.527 (.029)	0.159 (.008)	0.124 (.006)
CFRNet	0.895 (.077)	0.411 (.037)	0.130 (.007)	0.411 (.038)	0.175 (.013)	0.106 (.007)
CRN	1.045 (.064)	0.546 (.039)	0.360 (.024)	0.864 (.052)	0.767 (.040)	0.357 (.021)
RMSN	0.390 (.031)	0.362 (.028)	0.332 (.026)	0.447 (.041)	0.386 (.034)	0.385 (.032)
CGP	0.660 (.043)	0.610 (.039)	0.561 (.035)	0.826 (.056)	0.693 (.047)	0.602 (.038)
1NN	1.866 (.099)	1.721 (.091)	1.614 (.078)	2.446 (.131)	1.746 (.106)	1.384 (.083)

Sensitivity to omitted covariates. In Figure 3 (B), we show the situation when some covariates are omitted from the analysis, thus making it harder to adjust for the confounding bias. In particular, we set $\mathbf{x}_t = \{\mathrm{k}_t^{in}, \mathrm{y}_t, \mathrm{p}_t\}$, $\mathbf{x}_t = \{\mathrm{y}_t, \mathrm{p}_t\}$ or $\mathbf{x}_t = \{\mathrm{y}_t\}$ (omitting 0,1,2 covariates). We observe that SyncTwin is still the best-performing model when some covariates are omitted.

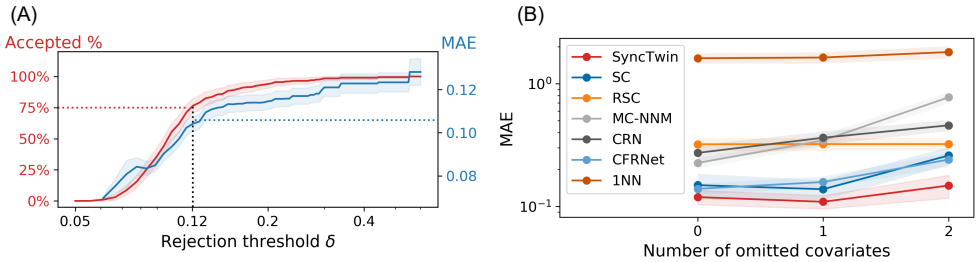

Figure 3: **(A)** The percentage of accepted estimates (red) and their MAE (blue) using various rejection thresholds δ on d_t^y. Eg. setting $\delta = 0.12$ gives 75% acceptance rate and MAE around 0.105. **(B)** Performance when certain covariates are omitted (hidden confounders). Shaded area: 95% confidence interval.

5.2 Experiment on real data

To validate the usability of SyncTwin in real-world medical problems, we conduct an observational study to emulate a large-scale randomized controlled trial (RCT)—Heart Protection Study (HPS) [21, 20]. Although most existing methods have been validated in synthetic or semi-synthetic experiments [41, 11], few of them have been demonstrated to successfully reproduce the findings of a RCT from observational data. Since RCT is the gold standard of treatment effect estimation, the ability to reproduce a RCT provides strong evidence about the method's usability in medical research and facilitates the method's adoption in the medical community.

The HPS was conducted to investigate the treatment effect of statins, a drug commonly used to lower the LDL Cholesterol (LDL). It reported an a change of -1.26 mmol/L (SD=0.06) in LDL for participants randomised to statins versus placebo at the end of the first year after treatment [37]. The study enrolled 20,536 individuals and lasted for eight years, making it one of the largest studies to investigate the treatment effect of statins.

Data Source. We used medical records from English National Health Service general practices that contributed anonymised primary care electronic health records to the Clinical Practice Research Datalink (CPRD), covering approximately 6.9 percent of the UK population [25]. CPRD was linked to secondary care admissions from Hospital Episode Statistics, and national mortality records from the Office for National Statistics. We defined treatment initiation as the date of first CPRD prescription

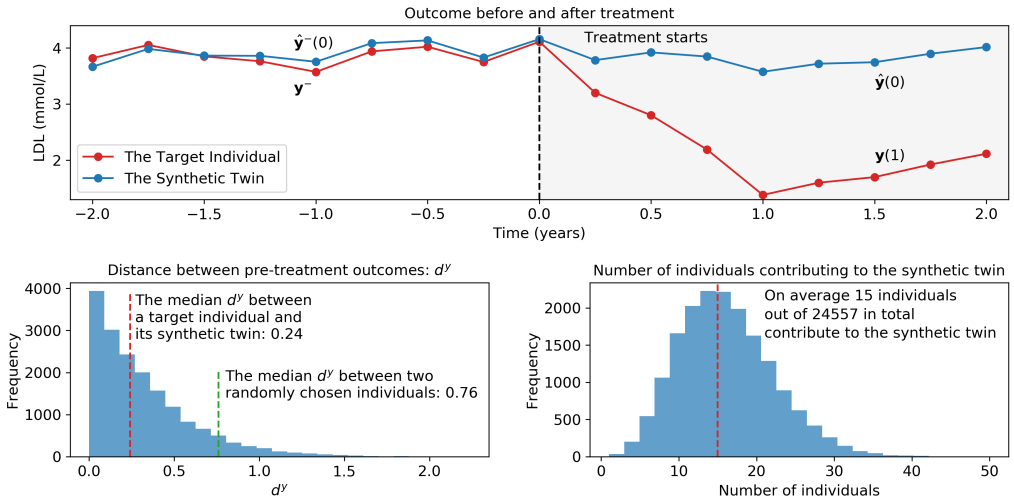

Figure 4: Top: the outcomes (LDL) before and after treatment of a target individual and its synthetic twin. Bottom left: histogram of distance d^y (Equation 7). Bottom right: histogram of number of contributors used to construct the synthetic twin.

and the outcome of interest was the measured LDL. Known risk factors for LDL were selected as temporal covariates measured before treatment initiation: HDL Cholesterol, Systolic Blood Pressure, Diastolic Blood Pressure, Body Mass Index, Pulse, Creatinine, Triglycerides and smoking status. Our analysis is based on a subset of 125,784 individuals who met the enrollment criterion of HPS (Appendix A.15). They were split into three equally-sized subsets for training, validation and testing, each with 17,371 treated and 24,557 controls.

Main findings. SyncTwin estimates the average treatment effect as **-1.25** mmol/L (SD 0.01) by averaging the estimated ITE on testing data $\sum_{i \in \mathcal{D}^{te}} \hat{\tau}_{it}/|\mathcal{D}^{te}|$. The estimate is very close to the result reported in HPS, -1.26 mmol/L (SD=0.06), after taking into account the standard errors. In contrast, CRN and RMSN estimate the effect to be **-0.72** mmol/L (SD 0.01) and **-0.83** mmol/L (SD 0.01) respectively, which are significantly smaller than the trial's finding. Other benchmark methods either cannot handle irregularly-measured covariates or do not scale to the size of the dataset. Our result suggests SyncTwin is able to overcome the confounding bias in the complex real-world datasets. Note that SyncTwin is also able to estimate ITE and the effect over different time horizons. However, since these results are not reported in HPS, we are not able to quantitatively evaluate them.

Qualitative evaluation of ITE over time. For each individual, we can visualize the outcomes before treatment and compare them with the synthetic twin to sense-check the estimate. The individual shown in Figure 4 (top) has a sensible ITE estimate because the synthetic twin matches its pre-treatment outcomes closely. In addition to visualization, we can calculate the individualized error bound based on d_i^y (Equation 7). From Figure 4 (bottom left) we can see in most cases d_i^y is small with a median of 0.24 mmol/L (compared to the average between any two randomly chosen individuals 0.76 mmol/L). This means if the expert can only tolerate an error of 0.24 mmol/L on ITE estimation, half of the estimates (those with $\mathrm{d}_i^y \leq 0.24$ mmol/L) can be accepted (Section 3). As shown in Figure 4 (bottom right) on average only 15 (out of 24,557) individuals contribute to the synthetic twin ($b_{ij} > 0.01$). The medical experts can check these few contributors to interpret the estimate.

6 Conclusion and future work

We present SyncTwin, an ITE estimation method tailored for temporal outcomes with point treatment. SyncTwin achieves interpretability and strong performance in both simulated and real data experiments. In future works, we plan to extend SyncTwin to the dynamic treatment setting and to model the outcomes in continuous time. Developing additional assumptions to guarantee the robustness to certain types of unobserved confounders is also an interesting avenue for future research.

References

[1] Alberto Abadie. Using synthetic controls: Feasibility, data requirements, and methodological aspects. *Journal of Economic Literature*, 2019.

[2] Alberto Abadie, Alexis Diamond, and Jens Hainmueller. Synthetic control methods for comparative case studies: Estimating the effect of california's tobacco control program. *Journal of the American statistical Association*, 105(490):493–505, 2010.

[3] Alberto Abadie and Javier Gardeazabal. The economic costs of conflict: A case study of the basque country. *American economic review*, 93(1):113–132, 2003.

[4] Jaap H Abbring and Gerard J Van den Berg. The nonparametric identification of treatment effects in duration models. *Econometrica*, 71(5):1491–1517, 2003.

[5] Linda H Aiken, Douglas M Sloane, Luk Bruyneel, Koen Van den Heede, Peter Griffiths, Reinhard Busse, Marianna Diomidous, Juha Kinnunen, Maria Kózka, Emmanuel Lesaffre, et al. Nurse staffing and education and hospital mortality in nine european countries: a retrospective observational study. *The lancet*, 383(9931):1824–1830, 2014.

[6] Ahmed M Alaa and Mihaela van der Schaar. Bayesian inference of individualized treatment effects using multi-task gaussian processes. In *Advances in Neural Information Processing Systems*, pages 3424–3432, 2017.

[7] Muhammad Amjad, Devavrat Shah, and Dennis Shen. Robust synthetic control. *The Journal of Machine Learning Research*, 19(1):802–852, 2018.

[8] Susan Athey, Mohsen Bayati, Nikolay Doudchenko, Guido Imbens, and Khashayar Khosravi. Matrix completion methods for causal panel data models. Technical report, National Bureau of Economic Research, 2018.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[10] Jushan Bai and Serena Ng. *Large dimensional factor analysis*. Now Publishers Inc, 2008.

[11] Ioana Bica, Ahmed M Alaa, James Jordon, and Mihaela van der Schaar. Estimating counterfactual treatment outcomes over time through adversarially balanced representations. In *International Conference on Learning Representations*, 2020.

[12] Melissa DA Carlson and R Sean Morrison. Study design, precision, and validity in observational studies. *Journal of palliative medicine*, 12(1):77–82, 2009.

[13] Barbra A Dickerman, Xabier García-Albéniz, Roger W Logan, Spiros Denaxas, and Miguel A Hernán. Avoidable flaws in observational analyses: an application to statins and cancer. *Nature Medicine*, 25(10):1601–1606, 2019.

[14] Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. Why does unsupervised pre-training help deep learning? In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 201–208, 2010.

[15] Dumitru Erhan, Pierre-Antoine Manzagol, Yoshua Bengio, Samy Bengio, and Pascal Vincent. The difficulty of training deep architectures and the effect of unsupervised pre-training. In *Artificial Intelligence and Statistics*, pages 153–160, 2009.

[16] Demiana William Faltaos, Saïk Urien, Valérie Carreau, Marina Chauvenet, Jean Sebastian Hulot, Philippe Giral, Eric Bruckert, and Philippe Lechat. Use of an indirect effect model to describe the ldl cholesterol-lowering effect by statins in hypercholesterolaemic patients. *Fundamental & clinical pharmacology*, 20(3):321–330, 2006.

[17] Steven E Finkel. *Causal analysis with panel data*. Sage, 1995.

[18] Jared C Foster, Jeremy MG Taylor, and Stephen J Ruberg. Subgroup identification from randomized clinical trial data. *Statistics in medicine*, 30(24):2867–2880, 2011.

[19] GPy. GPy: A gaussian process framework in python. http://github.com/SheffieldML/GPy, since 2012.

[20] Heart Protection Study Collaborative Group et al. Mrc/bhf heart protection study: randomised placebo-controlled trial of cholesterol-lowering with simvastatin in 20,536 high-risk individuals. *Lancet*, 360(9326):7–22, 2002.

[21] Heart Protection Study Collaborative Group et al. Randomized trial of the effects of cholesterol-lowering with simvastatin on peripheral vascular and other major vascular outcomes in 20,536 people with peripheral arterial disease and other high-risk conditions. *Journal of vascular surgery*, 45(4):645–654, 2007.

[22] Negar Hassanpour and Russell Greiner. Learning disentangled representations for counterfactual regression. In *International Conference on Learning Representations*, 2020.

[23] Harshad Hegde, Neel Shimpi, Aloksagar Panny, Ingrid Glurich, Pamela Christie, and Amit Acharya. Mice vs ppca: Missing data imputation in healthcare. *Informatics in Medicine Unlocked*, 17:100275, 2019.

[24] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, pages 15663–15674, 2019.

[25] Emily Herrett, Arlene M Gallagher, Krishnan Bhaskaran, Harriet Forbes, Rohini Mathur, Tjeerd Van Staa, and Liam Smeeth. Data resource profile: clinical practice research datalink (cprd). *International journal of epidemiology*, 44(3):827–836, 2015.

[26] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[27] Guido W Imbens. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and statistics*, 86(1):4–29, 2004.

[28] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[29] Fredrik D Johansson, Nathan Kallus, Uri Shalit, and David Sontag. Learning weighted representations for generalization across designs. *arXiv preprint arXiv:1802.08598*, 2018.

[30] Nathan Kallus. Deepmatch: Balancing deep covariate representations for causal inference using adversarial training. *arXiv preprint arXiv:1802.05664*, 2018.

[31] Jimyon Kim, Byung-Jin Ahn, Hong-Seok Chae, Seunghoon Han, Kichan Doh, Jeongeun Choi, Yong K Jun, Yong W Lee, and Dong-Seok Yim. A population pharmacokinetic–pharmacodynamic model for simvastatin that predicts low-density lipoprotein-cholesterol reduction in patients with primary hyperlipidaemia. *Basic & clinical pharmacology & toxicology*, 109(3):156–163, 2011.

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[33] Matthieu Legrand, Claire Dupuis, Christelle Simon, Etienne Gayat, Joaquim Mateo, Anne-Claire Lukaszewicz, and Didier Payen. Association between systemic hemodynamics and septic acute kidney injury in critically ill patients: a retrospective observational study. *Critical care*, 17(6):1–8, 2013.

[34] Bryan Lim, Ahmed M Alaa, and Mihaela van der Schaar. Forecasting treatment responses over time using recurrent marginal structural networks. In *Advances in Neural Information Processing Systems*, pages 7483–7493, 2018.

[35] Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*, pages 6446–6456, 2017.

[36] Judea Pearl. *Causality*. Cambridge university press, 2009.

[37] Paul M Ridker and Nancy R Cook. Statins: new american guidelines for prevention of cardiovascular disease. *The Lancet*, 382(9907):1762–1765, 2013.

[38] James M Robins, Miguel Angel Hernan, and Babette Brumback. Marginal structural models and causal inference in epidemiology, 2000.

[39] Donald B Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, 2005.

[40] Peter Schulam and Suchi Saria. Reliable decision support using counterfactual models. In *Advances in Neural Information Processing Systems*, pages 1697–1708, 2017.

[41] Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pages 3076–3085. PMLR, 2017.

[42] Jae W Song and Kevin C Chung. Observational studies: cohort and case-control studies. *Plastic and reconstructive surgery*, 126(6):2234, 2010.

[43] Elizabeth A Stuart. Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1):1, 2010.

[44] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[45] Timo Teräsvirta, Dag Tjøstheim, Clive William John Granger, et al. *Modelling nonlinear economic time series*. Oxford University Press Oxford, 2010.

[46] Madeleine Udell and Alex Townsend. Nice latent variable models have log-rank. *arXiv preprint arXiv:1705.07474*, 1, 2017.

[47] Yiqing Xu. Generalized synthetic control method: Causal inference with interactive fixed effects models. *Political Analysis*, 25(1):57–76, 2017.

[48] Xiaobo Yang, Yuan Yu, Jiqian Xu, Huaqing Shu, Hong Liu, Yongran Wu, Lu Zhang, Zhui Yu, Minghao Fang, Ting Yu, et al. Clinical course and outcomes of critically ill patients with sars-cov-2 pneumonia in wuhan, china: a single-centered, retrospective, observational study. *The Lancet Respiratory Medicine*, 8(5):475–481, 2020.

[49] Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. Representation learning for treatment effect estimation from observational data. In *Advances in Neural Information Processing Systems*, pages 2633–2643, 2018.

[50] Koutaro Yokote, Hideaki Bujo, Hideki Hanaoka, Masaki Shinomiya, Keiji Mikami, Yoh Miyashita, Tetsuo Nishikawa, Tatsuhiko Kodama, Norio Tada, and Yasushi Saito. Multicenter collaborative randomized parallel group comparative study of pitavastatin and atorvastatin in japanese hypercholesterolemic patients: collaborative study on hypercholesterolemia drug intervention and their benefits for atherosclerosis prevention (chiba study). *Atherosclerosis*, 201(2):345–352, 2008.

[51] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Ganite: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations*, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Section 6

(c) Did you discuss any potential negative societal impacts of your work? [Yes] Appendix A.15

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] Section 3.1

(b) Did you include complete proofs of all theoretical results? [Yes] Appendix A.1

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Access to the CPRD is regulated. Researcher has to sign an end user license before access to the data is granted. The code will be released upon acceptance.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Appendix A.9

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Section 5

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Appendix A.14

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] Section 5.2

(b) Did you mention the license of the assets? [Yes] We have signed the end user agreement for CPRD.

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] It is discussed in detail in the publication by the dataset creators, which we cited in Section 5.2.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Section 5.2

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]