Estimating Average Causal Effects from Patient Trajectories

Dennis Frauen[1, 2], **Tobias Hatt**[3], **Valentyn Melnychuk**[1, 2], **Stefan Feuerriegel**[1, 2]

[1] LMU Munich
[2] Munich Center for Machine Learning
[3] ETH Zurich
frauen@lmu.de, thatt@ethz.ch, melnychuk@lmu.de, feuerriegel@lmu.de

Abstract

In medical practice, treatments are selected based on the expected causal effects on patient outcomes. Here, the gold standard for estimating causal effects are randomized controlled trials; however, such trials are costly and sometimes even unethical. Instead, medical practice is increasingly interested in estimating causal effects among patient (sub)groups from electronic health records, that is, observational data. In this paper, we aim at estimating the average causal effect (ACE) from observational data (patient trajectories) that are collected over time. For this, we propose DeepACE: an end-to-end deep learning model. DeepACE leverages the iterative G-computation formula to adjust for the bias induced by time-varying confounders. Moreover, we develop a novel sequential targeting procedure which ensures that DeepACE has favorable theoretical properties, i. e., is doubly robust and asymptotically efficient. To the best of our knowledge, this is the first work that proposes an end-to-end deep learning model tailored for estimating time-varying ACEs. We compare DeepACE in an extensive number of experiments, confirming that it achieves state-of-the-art performance. We further provide a case study for patients suffering from low back pain to demonstrate that DeepACE generates important and meaningful findings for clinical practice. Our work enables practitioners to develop effective treatment recommendations based on population effects.

Introduction

Causal effects of treatments on patient outcomes are hugely important for decision-making in medical practice (Yazdani and Boerwinkle 2015; Melnychuk, Frauen, and Feuerriegel 2022b). These estimates inform medical practitioners in the expected effectiveness of treatments and thus guide treatment selection. Notwithstanding, information on causal effects is also relevant for other decision-making of other domains such as public health (Glass et al. 2013) and marketing (Varian 2016).

Causal effects can be estimated from either randomized controlled trials (RCTs) or observational studies (Robins, Hernán, and Brumback 2000; Frauen and Feuerriegel 2022). Even though RCTs present the gold standard for estimating causal effects, they are often highly costly, unfeasible in practice, or even unethical (e. g., medical professionals cannot withhold effective treatment to patients in need) (Robins, Hernán, and Brumback 2000). Therefore, medical practice is increasingly relying on observational data to study causal relationships between treatments and patient outcomes. Nowadays, electronic health records are readily available, capturing patient trajectories with high granularity, and thus providing rich observational data in medical practice (Allam et al. 2021).

In this paper, we aim at estimating causal effects from observational data in form of patient trajectories. Patient trajectories encode medical histories in a time-resolved manner and are thus of longitudinal form. However, estimating causal effects from observational data is subject to challenges (Robins and Hernán 2009). The reason is that the underlying treatment assignment mechanism is usually confounded with the patient outcomes. Another reason is that confounders may vary over time, which introduces additional dependencies and treatment-confounder feedback.

While there are works on estimating individualized causal effects (Lim, Alaa, and van der Schaar 2018; Bica et al. 2020; Melnychuk, Frauen, and Feuerriegel 2022a), we are interested in **average causal effects (ACEs)**. As such, ACEs give the expected difference in health outcomes when applying different treatment interventions at the population level. ACEs are important in many applications ranging from marketing to epidemiology, where policies are always based on (sub)population effects (Varian 2016; Naimi, Cole, and Kennedy 2017). A prominent example is public health: here, a government might be interested in the different average effects of a stay-home order on COVID-19 spread for vaccinated vs. non-vaccinated people.

Proposed method: We propose an end-to-end deep learning model to estimate time-varying ACEs, called DeepACE. DeepACE combines a recurrent neural network and feedforward neural networks to learn conditional expectations of factual and counterfactual outcomes under complex nonlinear dependencies, based on which we then estimate time-varying ACEs. In DeepACE, we address time-varying confounding by leveraging the G-formula, which expresses the ACE as a sequence of nested conditional expectations based on observational data as. Existing methods are limited in that these learn the nested conditional expectations *separately* by performing an iterative procedure (van der Laan and Rose

2018). In contrast, our end-to-end model DeepACE makes it possible to learn them *jointly*, leading to a more efficient use of information across time.

We further develop a *sequential targeting procedure* by leveraging results from semi-parametric estimation theory in order to improve the estimation quality of DeepACE. The sequential targeting procedure perturbs ("targets") the outputs of DeepACE so that our estimator satisfies a semi-parametric efficient estimating equation. To achieve this, we propose a targeting layer and a targeted regularization loss for training. We then derive that DeepACE provides a doubly robust and asymptotically efficient estimator.

Our main **contributions**[1]: (i) We propose DeepACE: the first end-to-end neural network for estimating time-varying average causal effects using observational data. DeepACE builds upon the iterative G-computation formula to address time-varying confounding. (ii) We develop a novel sequential targeting procedure which ensures that DeepACE provides a doubly robust and asymptotically efficient estimator. (iii) We perform an extensive series of computational experiments using state-of-the-art models for time-varying ACE estimation, establishing that DeepACE achieves a superior performance. We further demonstrate that DeepACE generates important findings based on a medical case study for patients suffering from low back pain.

Related Work

Estimating causal effects from observational data can be grouped into static and longitudinal settings (Table 1).

		Static setting	**Longitudinal setting**
Effects	Individual (=ITE)	e. g., TARNet, causal forest	RMSNs, CRN, G-Net
	Average (=ACE)	e. g., TMLE, DragonNet	G-methods, LTMLE **DeepACE (ours)**

Table 1: Key methods for causal effect estimation.

Causal Effect Estimation in the Static Setting

Extensive works focus on treatment effect estimation in static settings. Two important methods that adopt machine learning for *average* treatment effect estimation in the static setting are: (i) targeted maximum likelihood estimation (TMLE) (van der Laan and Rubin 2006) and (ii) DragonNet (Shi, Blei, and Veitch 2019). TMLE is a plugin estimator that takes a (machine learning) model as input and perturbs the predicted outcomes, so that the final estimator satisfies a certain efficient estimating equation. This idea builds upon semi-parametric efficiency theory and is often called *targeting*. Any estimator satisfying the efficient estimating equation is guaranteed to have desirable asymptotic properties. On the other hand, DragonNet is a neural network that incorporates a targeting procedure into the training process by extending the network architecture and adding a tailored regularization term to the loss function. This allows the model parameters to adapt simultaneously to provide a targeted estimate. To the best of our knowledge, there exists no similar targeting procedure for longitudinal data, and, to fill this gap, we later develop a tailored *sequential* targeting procedure.

In general, causal effect estimation for static settings is different from longitudinal settings due to time-varying confounding and treatment-confounder feedback (Robins and Hernán 2009). Hence, methods for static causal effect estimation are biased when they are applied to longitudinal settings, thus leading to an inferior performance. For this reason, we later use methods for time-varying causal effect estimation as our prime baselines. Results for static baselines are reported in the Appendix[2].

Causal Effect Estimation in Longitudinal Settings

Causal effect estimation in longitudinal settings is often also called *time-varying* causal effect estimation. Here, we distinguish individual and average causal effects.

Individual causal effects: There is a growing body of work on adapting neural networks for estimating *individual* causal effects in the longitudinal setting (also individual treatment effects or ITE). These methods predict counterfactual outcomes conditioned on individual patient histories. Recurrent marginal structural networks (RMSNs) (Lim, Alaa, and van der Schaar 2018) use inverse probability weighting to learn a sequence-to-sequence model that addresses bias induced by time-varying confounders. The counterfactual recurrent network (CRN) (Bica et al. 2020) adopts adversarial learning to build balanced representations. The G-Net (Li et al. 2021) incorporates the G-formula into a recurrent neural network and applies Monte Carlo sampling.

ITE methods are not optimal for ACE estimation: Even though ITE methods can be used for ACE estimation by averaging individual effects, this is **not** optimal for two reasons: (i) There is a well-established efficiency theory for ACE estimation (Kennedy 2016). In particular, methods that average ITE estimators suffer from so-called *plug-in bias* (Curth, Alaa, and van der Schaar 2020). (ii) ITE methods for the longitudinal setting exclude covariates at time steps after the start of intervention from training and prediction (often by using an encoder-decoder architecture) in order to estimate effects of different interventions on individual patients. This is because these methods aim at estimating individual effects for unseen patients or time steps (out-of-sample). G-computation for unseen patients is hard because post-intervention covariates need to be predicted, leading to problems if X is high-dimensional. This is not needed for the ACE as we average over the observed population. Still, we later include the above state-of-the-art methods from ITE estimation (i. e., RMSNs, CRN, and G-Net) as baselines (we average individual estimates). There are other methods for predicting counterfactual outcomes over time (e. g., (Schulam and Saria 2017; Soleimani, Subbaswamy, and Saria 2017; Qian et al. 2021; Berrevoets et al. 2021)), which are not applicable due to different settings or assumptions.

[1]Code available at https://github.com/DennisFrauen/DeepACE.

[2]Appendix available at https://arxiv.org/abs/2203.01228.

Average causal effects: Several methods for time-varying ACE estimation originate from epidemiological literature. Here, common are so-called g-methods (Naimi, Cole, and Kennedy 2017). Examples of g-methods include marginal structural models (Robins, Hernán, and Brumback 2000), G-computation via the G-formula (Robins and Hernán 2009), and structural nested models (Robins 1994). The previous methods make linearity assumptions and are consequently not able to exploit nonlinear dependencies within the data. Nevertheless, we include the g-methods as baselines.

We are aware of only one work that leverages machine learning methods for time-varying ACE estimation: longitudinal targeted maximum likelihood estimation (LTMLE) (van der Laan and Gruber 2012). LTMLE has two components: (i) LTMLE uses iterative G-computation. Mathematically, the G-formula can be expressed in terms of nested conditional expectations, which can then be learned successively by using arbitrary regression models. As such, each conditional expectation is learned *separately*, which is known as iterative G-computation. Our method, DeepACE, follows a similar approach but estimates the conditional expectations *jointly*. (ii) LTMLE targets the estimator by perturbing the predicted outcomes in each iteration to make them satisfy an efficient estimating equation. In contrast to LTMLE, our sequential targeting procedure is incorporated into the model training process, which allows all model parameters to be learned simultaneously. Later, we implement two variants of LTMLE as baselines, namely LTMLE with a generalized linear model (glm) (van der Laan and Gruber 2012) and LTMLE with a super learner (van der Laan and Rose 2018).

Research gap: To the best of our knowledge, there exists no *end-to-end* machine learning model tailored for *time-varying ACE estimation*. Hence, DeepACE is the first neural network that simultaneously learns to perform iterative G-computation and to apply a sequential targeting procedure. By learning all parameters jointly, we expect our end-to-end model to provide more accurate estimation results.

Problem Setup

Setting

We build upon the standard setting for estimating time-varying ACEs (Robins and Hernán 2009; van der Laan and Gruber 2012). For each time step $t \in \{1, \dots, T\}$, we observe (time-varying) patient covariates $X_t \in \mathbb{R}^p$, treatments $A_t \in \{0, 1\}$, and outcomes $Y_{t+1} \in \mathbb{R}$. For example, we would model critical care for COVID-19 patients by taking blood pressure and heart rate as time-varying patient covariates, ventilation as treatment, and respiratory frequency as outcome. Modeling the treatments A_t as binary variables is consistent with prior works (van der Laan and Rubin 2006; Shi, Blei, and Veitch 2019) and is standard in medical practice (Robins, Hernán, and Brumback 2000): should one apply a treatment or not?

At each time step t, the treatment A_t directly affects the next outcome Y_{t+1}, and the covariates X_t may affect both A_t and Y_{t+1}. All X_t, A_t, and Y_{t+1} may have direct effects on future treatments, covariates, and outcomes. The corresponding causal graph is shown in Fig. 1. For notation, we denote the observed trajectory at time t by $\mathcal{H}_t = (\bar{X}_t, \bar{A}_{t-1})$, where $\bar{X}_t = (X_1, \dots, X_t)$ and $\bar{A}_{t-1} = (A_1, \dots, A_{t-1})$. We always consider the lagged outcomes Y_t to be included in the covariates X_t.

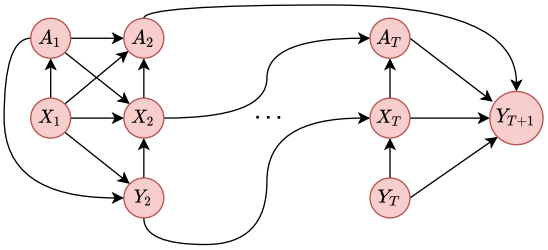

Figure 1: One possible example of a causal graph describing the data-generating process.

We have further access to an observational dataset $\mathcal{D}$, that consists of N independent patient trajectories $\mathcal{D}$ for patients $i \in \{1, \dots, N\}$, i. e., $\mathcal{D} = (\{x_t^{(i)}, a_t^{(i)}, y_{t+1}^{(i)}\}_{t=1}^{T})_{i=1}^{N}$. Such patient trajectories are nowadays widely available in electronic health records (Allam et al. 2021). For notation, we use a superscript (i) to refer to patients (we omit it unless needed).

We build upon the potential outcomes framework (Rubin 1978) and denote $Y_{t+1}(\bar{a}_t)$ as the potential outcome, which would have been observed at time $t+1$ if a treatment intervention $\bar{a}_t = (a_1, \dots, a_t)$ was applied. Note that $Y_{t+1}(\bar{a}_t)$ is unobserved if the treatment intervention $\bar{a}_t$ does not coincide with the treatment assignments $\bar{A}_t$ in the observational dataset. This is also known as the fundamental problem of causal inference (Pearl 2009).

ACE Estimation

Given two intended treatment interventions $\bar{a}_T = (a_1, \dots, a_T)$ and $\bar{b}_T = (b_1, \dots, b_T)$, we define the expected potential outcomes as

$$\theta^a = \mathrm{E}\left[Y_{T+1}(\bar{a}_T)\right] \text{ and } \theta^b = \mathrm{E}\left[Y_{T+1}\left(\bar{b}_T\right)\right]. \quad (1)$$

Objective: We aim at estimating the average causal effect (ACE) $\psi = \theta^a - \theta^b$. To do so, we impose three standard causal inference assumptions (Robins, Hernán, and Brumback 2000): consistency, positivity, and sequential ignorability (see Appendix). Together, these assumptions allow to identify the ACE ψ from observational data (Pearl 2009).

Iterative G-computation

In contrast to the static setting, simple covariate adjustment is not sufficient for identification of the ACE because the time-varying covariates may be caused by previous treatments (Pearl 2009). The well-known *G-formula* (Robins 1986) offers a remedy by successively integrating out post-treatment covariates while conditioning on the intervention of interest. For our setting, we consider a variant that uses iterated conditional expectations (Robins 1999; Bang and Robins 2005). That is, we can write the parameter θ^a as the

Algorithm 1: Iterative G-computation (Robins 1999; van der Laan and Rose 2018)

$\hat{Q}^a_{T+2} \leftarrow Y_{T+1}$
for $t \in \{T+1, T, \ldots, 2\}$ **do**
  $\hat{Q}_t(\cdot) \leftarrow$ Regress $\hat{Q}^a_{t+1}$ on $\left(\bar{X}_{t-1}, \bar{A}_{t-1}\right)$
  $\hat{Q}^a_t \leftarrow \hat{Q}_t\left(\bar{X}_{t-1}, \bar{a}_{t-1}\right)$
end
$\hat{\theta}^a \leftarrow \frac{1}{N}\sum_{i=1}^{N} \hat{Q}_2^{a^{(i)}}$

result of an iterative process. More precisely, we introduce recursively defined conditional expectations Q^a_t that depend on the covariates $\bar{X}_{t-1}$ and interventions $\bar{a}_{t-1}$ via

$$Q^a_t = Q_t\left(\bar{X}_{t-1}, \bar{a}_{t-1}\right) = \mathrm{E}\left[Q^a_{t+1} \mid \bar{X}_{t-1}, \bar{A}_{t-1} = \bar{a}_{t-1}\right] \quad (2)$$

for $t \in \{2, \ldots, T+1\}$, and initialize $Q^a_{T+2} = Y_{T+1}$. Then, the expected potential outcome can be written as

$$\theta^a = \mathrm{E}\left[Q^a_2\right]. \quad (3)$$

In Eq. (2), the covariates $X_T, X_{T-1}, \ldots$ are successively integrated out, and the θ^a is obtained in Eq. (3) by averaging. For a derivation, we refer to the Appendix.

In the following, we review iterative G-computation (van der Laan and Rose 2018), which is an iterative procedure that leverages Eq. (2) and Eq. (3) to estimate the expected potential outcome θ^a and, subsequently, the average causal effect ψ. Iterative G-computation estimates the conditional expectations Q^a_t by using a regression model for all $t \in \{2, \ldots, T+1\}$. Then, θ^a can be estimated by taking the empirical mean in Eq. (3). The full algorithm is in Alg. 1.

Iterative G-computation in the above form is subject to drawbacks. In particular, one has to specify T separate regression models that are trained separately. Each model uses the predictions of its already trained predecessor as training labels. This may lead to a training procedure which is comparatively unstable.

Need for an end-to-end model: We postulate that an end-to-end model that learns all regression models jointly should overcome the above drawbacks. In particular, an end-to-end model can share information across time steps and, thereby, should be able to generate more accurate estimates for time-varying ACEs. Motivated by this, our end-to-end model learns all parameters jointly.

DeepACE

Overview: We propose a novel end-to-end deep learning model for ACE estimation, called DeepACE. DeepACE is motivated by the idea of iterative G-computation. It is trained with observational data from patient trajectories and a specific treatment intervention $\bar{a}_T$, based on which it learns the conditional expectations Q^a_t from Eq. (2).

DeepACE consists of two main components: (i) a **G-computation layer** (Sec. G-computation layer), which produces initial estimates of the Q^a_t by minimizing a tailored G-computation loss, and (ii) a **targeting layer** (Sec. Targeting layer), which applies perturbations in a way that the final estimator satisfies an efficient estimating equation. The overall model architecture is shown in Fig. 2.

DeepACE is trained by combining (i) a *G-computation loss* $\mathcal{L}_Q$, (ii) a *propensity loss* $\mathcal{L}_g$, and (iii) a *targeting loss* $\mathcal{L}_{\text{tar}}$ into a joint loss $\mathcal{L}$ as described later. The outputs of DeepACE can then be used to provide ACE estimates (Sec. ACE estimation). We further show that DeepACE provides a doubly robust and asymptotically efficient estimator (Sec. Theoretical results). Finally, we provide implementation details are described in Sec. Model training.

G-computation Layer

The G-computation layer takes the observational data $\mathcal{D}$ and a specific treatment intervention $\bar{a}_T$ as input. For each time step $t \in \{2, \ldots, T+1\}$, it generates two outputs: (i) a factual output $\hat{Q}^A_t$ for $Q_t\left(\bar{X}_{t-1}\bar{A}_{t-1}\right)$, and (ii) a counterfactual output $\hat{Q}^a_t$ for $Q_t\left(\bar{X}_{t-1}, \bar{a}_{t-1}\right)$ according to Eq. (2). The factual outputs $\hat{Q}^A_t$ are trained to estimate the one-step shifted counterfactual outputs $\hat{Q}^a_{t+1}$, while the counterfactual outputs $\hat{Q}^a_t$ are obtained by implicitly evaluating $\hat{Q}_t\left(\bar{X}_{t-1}, \cdot\right)$ at $\bar{A}_{t-1} = \bar{a}_{t-1}$ as done in Alg. 1.

Architecture The architecture of the **G-computation layer** is shown in Fig. 2 (bottom). In the G-computation layer, we use a long short-term-memory (LSTM) layer (Hochreiter and Schmidhuber 1997) to process the input data. We choose an LSTM due to its ability to learn complex non-linear dynamics from patient trajectories while addressing the vanishing gradient problem which frequently occurs when using recurrent neural networks.

We feed the data twice into the LSTM: (i) with the observed treatments $\bar{A}_T$ (factual forward pass), and (ii) once with the treatment intervention $\bar{a}_T$ (counterfactual forward pass). Based on this, we computed the hidden LSTM states as follows. At each time step t, the factual forward pass leads to a factual hidden LSTM state h^A_t depending on the factual trajectory $\mathcal{H}_t = (\bar{X}_t, \bar{A}_{t-1})$, and the counterfactual forward pass leads to a counterfactual hidden LSTM state h^a_t depending on the past covariates $\bar{X}_t$ and interventions $\bar{a}_{t-1}$.

Both hidden states h^A_t and h^a_t are processed further. In the factual forward pass, we feed the factual hidden state h^A_t together with the current observed treatment A_t into a fully-connected feed-forward network FF^Q_t. The network FF^Q_t generates a factual output $\hat{Q}^A_{t+1}$ for $Q_{t+1}(\bar{X}_t, \bar{A}_t)$ according to Eq. (2). In the counterfactual forward pass, we feed the counterfactual hidden state h^a_t also FF^Q_t and replace the treatment input A_t with the current intervention a_t. As a result, the network FF^Q_t generates a counterfactual output $\hat{Q}^a_{t+1}$ for $Q^a_t = Q_{t+1}(\bar{X}_t, \bar{a}_t)$.

G-computation Loss We design a tailored loss function, such that we mimic Algorithm 1. For this, we denote the outputs of the G-computation layer for a patient i at time t by $\hat{Q}^{A^{(i)}}_{t+1}(\eta)$ and $\hat{Q}^{a^{(i)}}_{t+1}(\eta)$. Here, we explicitly state the dependence on the model parameters (i. e., the LSTM and feed-forward layers), which we denote by η. We define the

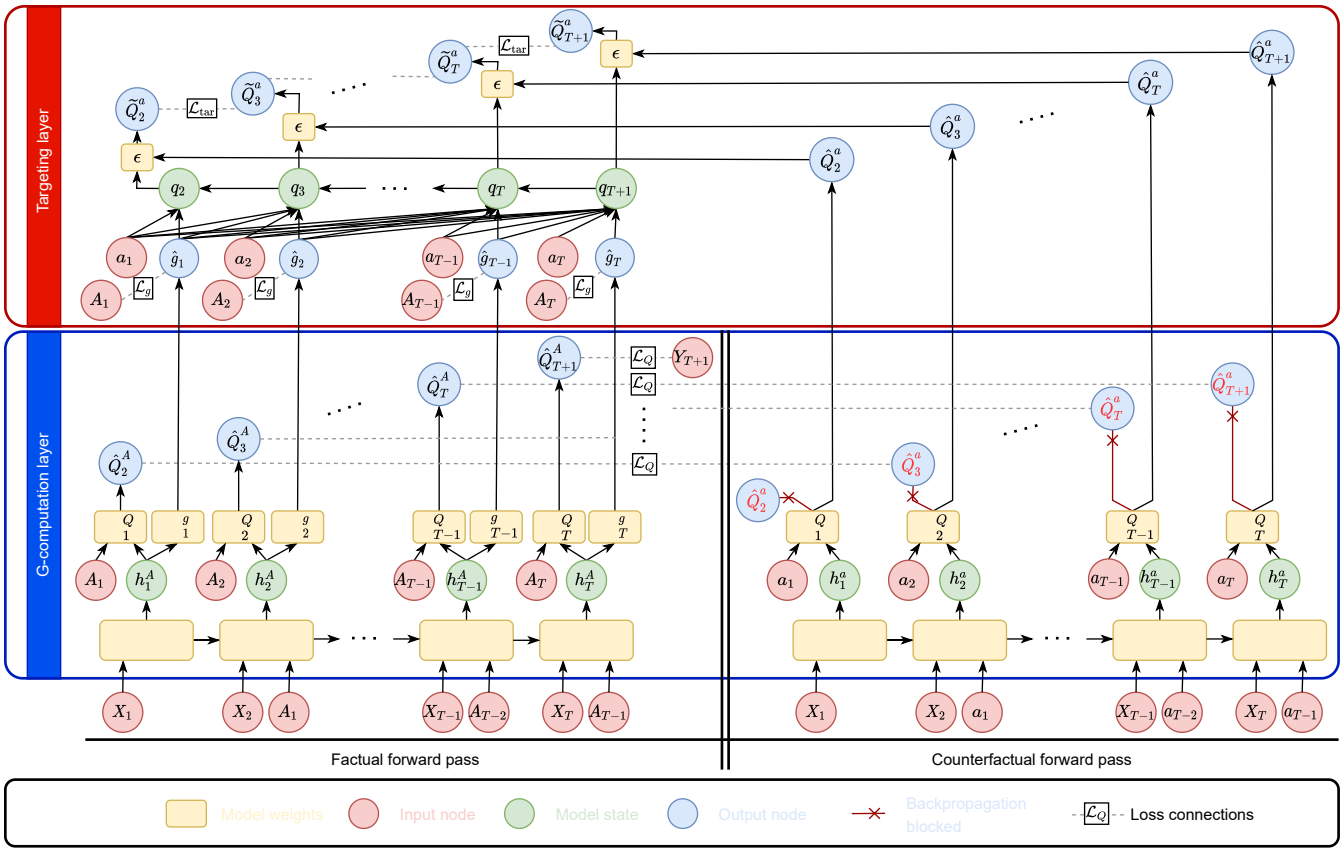

Figure 2: DeepACE consisting of the G-computation layer and the targeting layer.

G-computation loss as

$$\mathcal{L}_Q(\eta) = \frac{1}{N}\frac{1}{T}\sum_{i=1}^{N}\sum_{t=2}^{T+1}\left(\hat{Q}_t^{A^{(i)}}(\eta) - \hat{Q}_{t+1}^{a^{(i)}}(\eta)\right)^2, \qquad (4)$$

where we defined $\hat{Q}_{T+2}^{a^{(i)}}(\eta) = y_{T+1}^{(i)}$.

Blocking counterfactual backpropagation: Each counterfactual output $\hat{Q}_{t+1}^a$ is used as a prediction objective by the previous factual output $\hat{Q}_t^A$. Recall that, in Algorithm 1, the counterfactual estimates $\hat{Q}_{t+1}^a$ are obtained only by evaluating the learned conditional expectation at $\bar{A}_t = \bar{a}_t$. Therefore, we only want the factual outputs $\hat{Q}_t^A$ to learn the counterfactual outputs $\hat{Q}_{t+1}^a$ and not vice-versa. Hence, when training the model with gradient descent, we block the gradient backpropagation through the counterfactual FF_t^Q during the counterfactual forward pass.

Targeting Layer

For our sequential targeting procedure, we now introduce a **targeting layer**. The motivation is as follows: In principle, we could estimate the expected potential outcome θ^a by first training the G-computation layer, and subsequently following Eq. (3) and taking the empirical mean over the first counterfactual outputs $\hat{Q}_2^a$. Instead, we propose to leverage results from semi-parametric estimation theory, as this allows us to construct an estimator with better theoretical properties, namely double robustness and asymptotic efficiency.[3] For this purpose, we design our targeting layer so that it estimates the *propensity scores* $g_t(\mathcal{H}_t) = P(A_t \mid \mathcal{H}_t)$.

We then use the propensity scores to perturb the counterfactual outputs $\hat{Q}_t^a$ to make them satisfy an efficient estimating equation. To formalize this, we first provide the mathematical background and subsequently describe how we implement the targeting layer.

Mathematical background In the following, we summarize the general framework under which semi-parametric efficient estimators can be obtained. Let $\hat{Q}^a = (\hat{Q}_2^a, \ldots, \hat{Q}_{T+1}^a)$ be estimators of the conditional expectations $(Q_2^a, \ldots, Q_{T+1}^a)$ from Eq. 2, and let $\hat{g} = (\hat{g}_1, \ldots, \hat{g}_T)$ be estimators of the propensity scores $(g_1, \ldots, g_T)$, where $g_t = g_t(\mathcal{H}_t)$. Furthermore, let $\hat{\theta}^a$ be an estimator of θ^a.

Ideally, we would like to obtain a tuple of estimators $(\hat{Q}^a, \hat{g}, \hat{\theta}^a)$ with the following properties: (1) *Double robustness*: If either $\hat{Q}^a$ or $\hat{g}$ are consistent, $\hat{\theta}^a$ is a consistent estimator of θ^a. (2) *(Semi-parametric) asymptotic efficiency*: If both $\hat{Q}^a$ and $\hat{g}$ are consistent, $\hat{\theta}^a$ achieves the smallest vari-

[3] For an overview on semi-parametric estimation theory, we refer to (Kennedy 2016).

ance among all asymptotically linear estimators of θ^a. It can be shown that, asymptotically, the tuple $(\hat{Q}^a, \hat{g}, \hat{\theta}^a)$ fulfills properties (1) and (2) if it satisfies the following *efficient estimating equation* (Kennedy 2016)

$$\frac{1}{N}\sum_{i=1}^{N} \phi\left(\hat{Q}^{a^{(i)}}, \hat{g}^{(i)}, \hat{\theta}^a\right) = 0, \tag{5}$$

where ϕ is the efficient influence function of $(\hat{Q}^a, \hat{g}, \hat{\theta}^a)$. We call an estimator that satisfies Eq. (5) "targeted". For the longitudinal setting, ϕ has a closed form (derived in van der Laan and Gruber (2012)) that is given by

$$\phi\left(Q^a, g, \theta^a\right) = \left(Q_2^a - \theta^a\right) + \sum_{t=2}^{T+1}\left(Q_{t+1}^a - Q_t^a\right)\left(\prod_{\ell=1}^{t-1}\frac{\mathbb{1}(A_\ell = a_\ell)}{g_\ell(\mathcal{H}_\ell)}\right), \tag{6}$$

where we used the convention that $Q_{T+2}^a = Y_{T+1}$ and where $\mathbb{1}(\cdot)$ denotes the indicator function.

Implementation We propose a sequential targeting layer to perturb the initial estimates produced by the G-computation layer in order to satisfy Eq. (5). Specifically, we add a model parameter that is jointly trained with the other model parameters. A tailored regularization term ensures that the efficient estimating estimation from Eq. (5) is satisfied.

Inputs to the targeting layer are (i) the counterfactual outputs $\hat{Q}_{t+1}^a(\eta)$ of the G-computation layer and (ii) predictions $\hat{g}_t(\eta)$ of the propensity scores $g_t(\mathcal{H}_t)$, where, η denotes the trainable parameters of the G-computation layer. To allow for gradient backpropagation, we obtain the counterfactual outputs $\hat{Q}_{t+1}^a(\eta)$ from a second (identical) output of FF_t^Q, where the gradient flow is not blocked during training. Furthermore, we generate the propensity estimates $\hat{g}_t(\eta)$ by adding separate feed-forward networks FF_t^g on top of the factual hidden states h_t^A.

In the following, we describe how the targeting layer applies perturbations to generate targeted outputs $\widetilde{Q}_{t+1}$. We recursively define perturbation values $q_{T+2}(\eta) = 0$ and

$$q_t(\eta) = q_{t+1}(\eta) - \prod_{\ell=1}^{t-1}\frac{\mathbb{1}(A_\ell = a_\ell)}{\hat{g}_\ell(\eta)} \tag{7}$$

for $t \in \{2, \ldots, T+1\}$. The perturbation values are used to create the targeted network outputs via

$$\widetilde{Q}_t^a(\eta, \epsilon) = \hat{Q}_t^a(\eta) + \epsilon q_t(\eta) \tag{8}$$

for $t \in \{2, \ldots, T+2\}$, where ϵ is an additional network parameter that is trained together with η. Note that, by definition, we have that $\widetilde{Q}_{T+2}^a(\eta, \epsilon) = Y_{T+1}$.

Loss We use two regularization terms in order to train the targeting layer. First, we define the *propensity loss*

$$\mathcal{L}_g(\eta) = \frac{1}{N}\frac{1}{T}\sum_{i=1}^{N}\sum_{t=1}^{T}\mathrm{BCE}\left(\hat{g}_t^{(i)}(\eta), a_t^{(i)}\right), \tag{9}$$

where BCE denotes binary cross-entropy loss. The propensity loss ensures that the propensity networks learn to predict the propensity scores $g_t(\mathcal{H}_t)$. Second, we define our *targeting loss*

$$\mathcal{L}_{\mathrm{tar}}(\eta, \epsilon) = \frac{1}{N}\frac{1}{T}\sum_{i=1}^{N}\sum_{t=2}^{T+1}\left(\widetilde{Q}_{t+1}^{a^{(i)}}(\eta, \epsilon) - \widetilde{Q}_t^{a^{(i)}}(\eta, \epsilon)\right)^2. \tag{10}$$

We show in Sec. Theoretical results that our targeting loss forces the outputs $\widetilde{Q}_{t+1}$ to satisfy the efficient estimating from Eq. (5) and thus makes them "targeted".

In contrast to other sequential targeting methods (e. g., LTMLE (van der Laan and Gruber 2012)) that apply targeting perturbations iteratively over time, our procedure allows the entire model to be learned jointly. We later show that this gives more accurate ACE estimates.

Model Training and ACE Estimation

Overall loss: To train DeepACE, we combine the above into an overall loss

$$\mathcal{L}(\eta, \epsilon) = \mathcal{L}_Q(\eta) + \alpha\mathcal{L}_g(\eta) + \beta\mathcal{L}_{\mathrm{tar}}(\eta, \epsilon), \tag{11}$$

where α and β are constants that control the amount of propensity and targeting regularization, respectively. Details on our implementation, training, and hyperparameter tuning are in the Appendix.

ACE estimation: Given two treatment interventions $\bar{a}_T$ and $\bar{b}_T$, we train two separate DeepACE models for each $\bar{a}_T$ and $\bar{b}_T$. Then, we estimate the ACE ψ via $\hat{\psi} = \frac{1}{N}\sum_{i=1}^{N}\left(\widetilde{Q}_2^{a^{(i)}} - \widetilde{Q}_2^{b^{(i)}}\right)$, where $\widetilde{Q}_2^{a^{(i)}}$ and $\widetilde{Q}_2^{b^{(i)}}$ denote the two targeted DeepACE outputs for patient i at time $t = 2$.

Theoretical Results

The following theorem ensures that our combination of targeting layer and regularization in DeepACE indeed produces a targeted estimator.

Theorem 1. *Let $(\hat{\eta}, \hat{\epsilon})$ be a stationary point of $\mathcal{L}(\eta, \epsilon)$. Then, for any $\beta > 0$, the estimator $\widetilde{\theta}^a = \frac{1}{N}\sum_{i=1}^{N}\widetilde{Q}_2^{a^{(i)}}(\hat{\eta}, \hat{\epsilon})$ is targeted, i. e., DeepACE satisfies the efficient estimating equation from Eq.* (5).

Proof. See Appendix. □

By Theorem 1, the DeepACE estimator $\widetilde{\theta}^a$ is doubly robust, i. e., $\widetilde{\theta}^a$ is consistent, even if either the targeted outputs $\widetilde{Q}_{t+1}^a$ or the propensity estimates $\hat{g}_t$ are misspecified.

Assuming that the true conditional expectations Q_{t+1}^a and propensity scores g_t are contained in a suitable hypothesis class, the initial outputs $\hat{Q}_{t+1}^a$ from the G-computation layer and the propensity estimates $\hat{g}$ will converge to Q_{t+1}^a and g_t with growing sample size due to the construction of $\mathcal{L}_Q$ and $\mathcal{L}_g$ (provided that $\alpha > 0$). The next corollary shows that this implies asymptotic efficiency of $\widetilde{\theta}^a$.

Corollary 1. *If the initial outputs $\hat{Q}_{t+1}^a$ from the G-computation layer and propensity estimates $\hat{g}_t$ of DeepACE are consistent for Q_{t+1}^a and g_t, then also the targeted outputs $\widetilde{Q}_{t+1}^a$ are consistent for Q_{t+1}^a. In particular, the DeepACE estimator $\widetilde{\theta}^a$ is asymptotically efficient for θ^a.*

Proof. See Appendix. □

Experiments

Baselines

We compare DeepACE against state-of-the-art methods for time-varying causal effect estimation, see Table 1. The baselines are selected from recent literature on causal effect estimation (Li et al. 2021; van der Laan and Gruber 2012). The baselines can be categorized into three groups: (1) **G-methods**, (2) Longitudinal targeted maximum likelihood estimation (**LTMLE**), and (3) **deep learning models for ITEs**. Implementation and hyperparameter tuning details for all baselines are in the Appendix.

Experiments Using Synthetic Data

Setting: Synthetic data are commonly used to evaluate the effectiveness of causal inference methods because they provide access to the counterfactual outcomes (e. g., (van der Laan and Rubin 2006; Bica, Alaa, and van der Schaar 2020; Shi, Blei, and Veitch 2019)). Therefore, we can successfully compute the ground-truth ACE and thus benchmark the performance of all methods.

Results: We generate a synthetic dataset with $N = 1000$ patient trajectories over $T = 15$ time steps and sample $p = 6$ time-varying covariates. We then evaluate our baselines on three different setups which correspond to different treatment interventions [4]. For each method and setup, we calculate the absolute error between estimated and ground-truth averaged over 5 different runs with random seeds. We refer to the Appendix for details regarding the data generating process and method evaluation.

The results are shown in Table 2. All baselines are clearly outperformed by DeepACE on all three experiments. The best-performing baseline is LTMLE with the super learner. This is reasonable as it is the only baseline available that is both tailored for ACE estimation and makes use of machine learning. The linear methods (i. e., g-methods, LTMLE, and glm) are not able to capture the non-linear dependencies within the data and thus achieve an inferior performance. The ITE baselines (except CT) use recurrent neural networks and should thus be able to learn non-linearities but, nevertheless, are inferior. This is unsurprising and attributed to the fact that they are designed for estimating individual rather than average causal effects (see Section Related work).

To compare DeepACE with iterative G-computation, we compute the estimation errors of both methods over different time lags $h \in \{1, \dots, 5\}$, corresponding to level of long-range dependencies within the data (see Appendix). The results are shown in Fig. 3, showing the effectiveness of DeepACE for long-range dependencies (common in EHRs).

Ablation study: We also analyze DeepACE but where the targeting layer is removed (Table 2). Both variants of DeepACE outperform the baselines. The variant without targeting performs well but we are careful with interpretations due to the simple nature of the synthetic data and rather relegate conclusions to real-world data as in the following.

[4] Here, $\bar{b}_T$ is fixed to the zero-intervention (no treatment applied), and $\bar{a}_T$ is chosen as $(\mathbb{1}(k \leq i \leq \ell))_{i \in \{1,\dots,T\}}$ for $k \in \{1, 3, 5\}$ and $\ell \in \{10, 13, 15\}$.

Method	Setup 1	Setup 2	Setup 3
(1) G-METHODS			
MSM	0.24 ± 0.18	0.27 ± 0.19	0.21 ± 0.10
Iter. G-computation	0.26 ± 0.23	0.34 ± 0.26	0.32 ± 0.27
G-formula	0.14 ± 0.10	0.44 ± 0.25	0.86 ± 0.45
SNMM	0.39 ± 0.04	0.47 ± 0.02	0.35 ± 0.02
(2) LTMLE			
LTMLE (glm)	0.20 ± 0.19	0.25 ± 0.19	0.33 ± 0.24
LTMLE (super l.)	0.13 ± 0.08	0.14 ± 0.12	0.19 ± 0.13
(3) DEEP L. (ITE)			
RMSNs	0.20 ± 0.10	0.52 ± 0.32	0.95 ± 0.52
CRN	0.20 ± 0.10	0.52 ± 0.32	0.95 ± 0.52
G-Net	0.18 ± 0.08	0.46 ± 0.26	0.97 ± 0.38
DeepACE w/o tar.	$\mathbf{0.04 \pm 0.03}$	0.10 ± 0.09	$\mathbf{0.15 \pm 0.11}$
DeepACE	$\mathbf{0.04 \pm 0.02}$	$\mathbf{0.09 \pm 0.07}$	0.18 ± 0.08

lower = better (best in bold)

Table 2: Results on synthetic data (mean $\pm$ std. dev.).

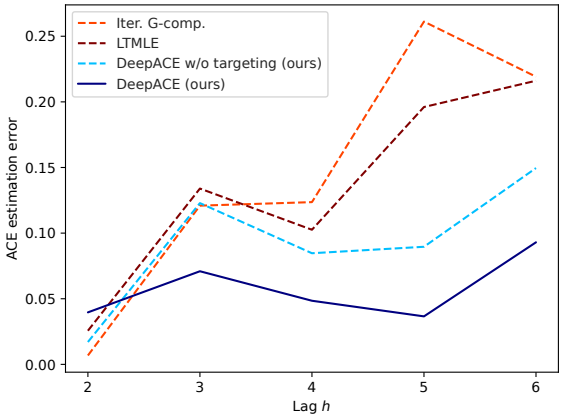

Figure 3: Performance comparison. Shown: mean estimation error averaged over 5 runs.

Experiments Using Semi-synthetic Data

Setting: We create semi-synthetic data that enables us to evaluate DeepACE using real-world data while having access to the ground-truth ACE. For this purpose, we use the MIMIC-III dataset (Johnson et al. 2016), which includes electronic health records from patients admitted to intensive care units. We generate $N = 1000$ patient trajectories. Again, compare three setups with different treatment interventions. For details, we refer to the Appendix.

Results: The results are shown in Table 3. Again, DeepACE outperforms all baselines by a large margin.

Ablation study: We repeat the experiments with DeepACE but where the targeting layer is removed (Table 3). This thus demonstrates the importance of our targeting procedure for achieving a superior performance.

Case Study Using Real-world Data

Setting: We demonstrate the value of DeepACE for real-world patient trajectories collected in a clinical study. For

Method	Setup 1	Setup 2	Setup 3
(1) G-METHODS			
MSM	2.35 ± 0.64	3.06 ± 0.67	2.47 ± 0.95
Iter. G-computation	0.81 ± 0.35	0.72 ± 0.72	1.90 ± 1.06
G-formula	0.32 ± 0.27	0.31 ± 0.20	0.32 ± 0.27
SNMM	0.28 ± 0.33	0.52 ± 0.26	1.96 ± 2.36
(2) LTMLE			
LTMLE (glm)	0.82 ± 0.35	0.72 ± 0.72	1.84 ± 1.13
LTMLE (super l.)	0.96 ± 1.01	0.92 ± 1.17	0.76 ± 0.47
(3) DEEP L. (ITE)			
RMSNs	2.35 ± 0.14	2.32 ± 0.18	2.36 ± 0.14
CRN	2.53 ± 0.03	2.53 ± 0.04	2.52 ± 0.04
G-Net	0.67 ± 0.15	0.65 ± 0.17	0.67 ± 0.15
DeepACE w/o tar.	$\mathbf{0.18 \pm 0.17}$	0.25 ± 0.11	0.21 ± 0.07
DeepACE	$\mathbf{0.18 \pm 0.14}$	$\mathbf{0.12 \pm 0.14}$	$\mathbf{0.16 \pm 0.10}$

lower = better (best in bold)

Table 3: Results on semi-synthetic data (mean $\pm$ std. dev.).

this purpose, we analyze data from $N = 928$ patients with low back pain (LBP) (Nielsen et al. 2017). Here, we are interested in the causal effect of whether patients have been allowed/disallowed to do physical labor (binary treatments) on pain intensity (outcome). Thus, the treatment interventions of interest are $a = (1, 1, 1)$ (physical labor) and $b = (0, 0, 0)$ (stop physical labor) with $T = 3$. Medical research is interested in identifying different phenotypes whereby LBP is classified into different subgroups with clinical meaningful interpretation. We thus estimate average causal effects $\psi_i = \theta_i^a - \theta_i^b$ for two patient cohorts, namely (1) severe LBP ($i = 1$) and (2) mild LBP ($i = 2$) (Nielsen et al. 2017).

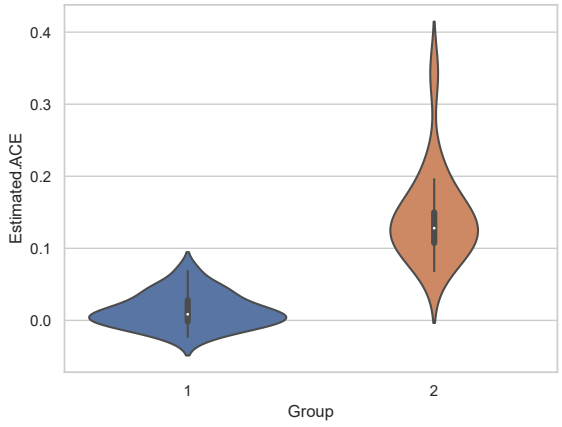

Figure 4: Violin charts (boxplots) showing the distributions of DeepACE estimates ψ_i (standardized) for two patient subgroups $i \in \{1, 2\}$ The distributions are obtained using Monte Carlo dropout during model prediction.

Results: We make several observations relevant for medical practice (Fig. 4): (i) Both ACEs are positive, which is line with medical knowledge implying that physical labor may worsen LBP progression. (ii) Surprisingly, the ACE is larger for the group with mild LBP. Hence, our results suggest that physical labor should be also stopped for mild LBP, as this would eliminate negative effects for observed pain. (iii) The variance is much larger for mild LBP, indicating that impact of physical labor is more heterogeneous.

Discussion

Estimating causal effects from observational data requires custom methods that adjust for time-varying confounding. Recently proposed deep learning models aim at estimating time-varying *individual treatment effects* (ITEs) and are not able to provide efficient *average causal effect* (ACE) estimators. DeepACE fills this gap by improving on state-of-the-art methods for time-varying ACE estimation.

Future work: Our work could be extended in several ways. Two possible directions are: (1) DeepACE could be extended to account for dynamic treatment regimes (Robins and Hernán 2009) instead of prespecified static ones. (2) The performance of targeted estimation under violation of identification assumptions (ignorability, positivity) could be of interest.

References

Allam, A.; Feuerriegel, S.; Rebhan, M.; and Krauthammer, M. 2021. Analyzing patient trajectories with artificial intelligence. *Journal of Medical Internet Research*, 23(12): e29812.

Bang, H.; and Robins, J. M. 2005. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61(4): 962–973.

Berrevoets, J.; Curth, A.; Bica, I.; McKinney, E.; and van der Schaar, M. 2021. Disentangled counterfactual recurrent networks for treatment effect inference over time. *arXiv preprint*, arXiv:2112.03811.

Bica, I.; Alaa, A. M.; Jordon, J.; and van der Schaar, M. 2020. Estimating counterfactual treatment outcomes over time through adversarially balanced representations. In *ICLR*.

Bica, I.; Alaa, A. M.; and van der Schaar, M. 2020. Time series deconfounder: Estimating treatment effects over time in the presence of hidden confounders. In *ICML*.

Curth, A.; Alaa, A. M.; and van der Schaar, M. 2020. Estimating structural target functions using machine learning and influence functions. *arXiv preprint*, arXiv:2008.06461.

Frauen, D.; and Feuerriegel, S. 2022. Estimating individual treatment effects under unobserved confounding using binary instruments. *arXiv preprint*, arXiv:2208.08544.

Glass, T. A.; Goodman, S. N.; Hernán, M. A.; and Samet, J. M. 2013. Causal inference in public health. *Annual Review of Public Health*, 34: 61–75.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation*, 9(8): 1735–1780.

Johnson, A. E. W.; Pollard, T. J.; Shen, L.; Lehman, L.-w. H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L. A.; and Mark, R. G. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1): 160035.

Kennedy, E. H. 2016. Semiparametric theory and empirical processes in causal inference. *Statistical Causal Inferences and their Applications in Public Health Research*, 141–167.

Li, R.; Hu, S.; Lu, M.; Utsumi, Y.; Chakraborty, P.; Sow, D. M.; Madan, P.; Li, J.; Ghalwash, M.; Shahn, Z.; and Lehman, L.-w. 2021. G-Net: A recurrent network approach to G-computation for counterfactual prediction under a dynamic treatment regime. In *ML4H*.

Lim, B.; Alaa, A. M.; and van der Schaar, M. 2018. Forecasting treatment responses over time using recurrent marginal structural networks. In *NeurIPS*.

Melnychuk, V.; Frauen, D.; and Feuerriegel, S. 2022a. Causal transformer for estimating counterfactual outcomes. In *ICML*.

Melnychuk, V.; Frauen, D.; and Feuerriegel, S. 2022b. Normalizing flows for interventional density estimation. *arXiv preprint*, arXiv:2209.06203.

Naimi, A. I.; Cole, S. R.; and Kennedy, E. H. 2017. An introduction to g methods. *International Journal of Epidemiology*, 46(2): 756–762.

Nielsen, A. M.; Kent, P.; Hestbaek, L.; Vach, W.; and Kongsted, A. 2017. Identifying subgroups of patients using latent class analysis: Should we use a single-stage or a two-stage approach? A methodological study using a cohort of patients with low back pain. *BMC Musculoskeletal Disorders*, 18(1): 57.

Pearl, J. 2009. Causal inference in statistics: An overview. *Statistics Surveys*, 3: 96–146.

Qian, Z.; Zhang, Y.; Bica, I.; Wood, A. M.; and van der Schaar, M. 2021. SyncTwin: Treatment effect estimation with longitudinal outcomes. In *NeurIPS*.

Robins, J. M. 1986. A new approach to causal inference in mortality studies with a sustained exposure period: Application to control of the healthy worker survivor effect. *Mathematical Modelling*, 7: 1393–1512.

Robins, J. M. 1994. Correcting for non-compliance in randomized trials using structural nested mean models. *Communications in Statistics - Theory and Methods*, 23(8): 2379–2412.

Robins, J. M. 1999. Robust estimation in sequentially ignorable missing data and causal inference models. *Proceedings of the American Statistical Association on Bayesian Statistical Science*, 6–10.

Robins, J. M.; and Hernán, M. A. 2009. *Estimation of the causal effects of time-varying exposures*. Chapman & Hall/CRC handbooks of modern statistical methods. Boca Raton: CRC Press. ISBN 9781584886587.

Robins, J. M.; Hernán, M. A.; and Brumback, B. 2000. Marginal structural models and causal inference in epidemiology. *Epidemiology*, 11(5): 550–560.

Rubin, D. B. 1978. Bayesian inference for causal effects: The role of randomization. *Annals of Statistics*, 6(1): 34–58.

Schulam, P.; and Saria, S. 2017. Reliable decision support using counterfactual models. In *NeurIPS*.

Shi, C.; Blei, D. M.; and Veitch, V. 2019. Adapting neural networks for the estimation of treatment effects. In *NeurIPS*.

Soleimani, H.; Subbaswamy, A.; and Saria, S. 2017. Treatment-response models for counterfactual reasoning with continuous-time, continuous-valued interventions. In *UAI*.

van der Laan, M. J.; and Gruber, S. 2012. Targeted minimum loss based estimation of causal effects of multiple time point interventions. *The International Journal of Biostatistics*, 8(1).

van der Laan, M. J.; and Rose, S. 2018. *Targeted learning in data science*. Cham: Springer. ISBN 978-3-319-65303-7.

van der Laan, M. J.; and Rubin, D. B. 2006. Targeted maximum likelihood learning. *The International Journal of Biostatistics*, 2(1).

Varian, H. R. 2016. Causal inference in economics and marketing. *Proceedings of the National Academy of Sciences (PNAS)*, 113(27): 7310–7315.

Yazdani, A. M.; and Boerwinkle, E. 2015. Causal inference in the age of decision medicine. *Journal of Data Mining in Genomics & Proteomics*, 6(1).