Encode-Decoder-based GAN for Estimating Counterfactual Outcomes under Sequential Selection Bias and Combinatorial Explosion

Yoshiyuki Norimatsu NORIMATSU.YOSHIYUKI@DY.MITSUBISHIELECTRIC.CO.JP
Information Technology R&D Center, Mitsubishi Electric Corporation, Kanagawa, Japan

Masaaki Imaizumi IMAIZUMI@G.ECC.U-TOKYO.AC.JP
The University of Tokyo / RIKEN Center for Advanced Intelligence Project, Tokyo, Japan

Editors: Biwei Huang and Mathias Drton

Abstract

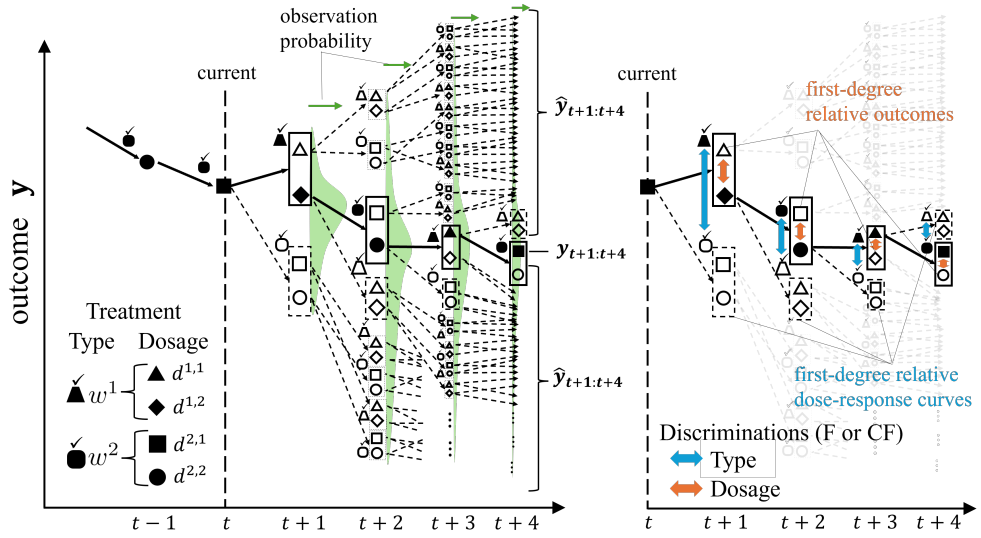
Estimating counterfactual outcomes of time-varying treatment types and associated dosages is important for addressing medical problems. This task becomes more challenging when both the treatment type and dosage assignment are biased due to the presence of time-varying confounders, as compared to estimating outcomes for treatment types alone. Specifically, the setup yields the following two obstacles: first, treatment types and dosages are selected sequentially, causing observed outcomes to be biased at each time step, leading to $2\times\tau$ biases for a τ-step-ahead prediction (sequential selection bias); second, the number of treatment trajectories increases exponentially with τ (combinatorial explosion). In this paper, we introduce Encoder-Decoder Time-SCIGAN (EDTS), which combines a longitudinal encoder-decoder transformer with a Generative Adversarial Network (GAN) for estimating counterfactuals. The encoder-decoder architecture predicts outcomes for one-step- and multi-step-ahead predictions separately, while the GAN generates counterfactual outcomes that cannot be distinguished from observed outcomes by the discriminators to handle sequential selection bias. To address combinatorial explosion, we propose a novel discrimination method, Sequential Counterfactual Discrimination (SCD) for EDTS discriminators. Our evaluation of synthetic and semi-synthetic datasets demonstrate that EDTS outperforms the current baselines. To the best of our knowledge, this is the first study to propose an architecture for estimating counterfactual outcomes of both time-varying treatment types and dosages. Implementation is available at https://github.com/ynorimat/EDTS.

Keywords: Causal Inference, GAN, Time-varying treatment types and dosages

1. Introduction

Estimating counterfactual outcomes of time-varying treatments has been an important challenge for machine learning (Lim et al., 2018; Bica et al., 2020a; Melnychuk et al., 2022). A typical example is a problem of medical treatments. For various medical conditions (e.g., medication treatment, cancer therapy, and COVID-19 vaccinations), the estimation allows us to select the optimal treatment trajectories for each patient. The treatment effect varies not only with treatment types (e.g., chemotherapy or radiotherapy) but also with treatment dosages (i.e., amount of chemotherapy, intensity of radiotherapy). It has been an important issue to choose the best treatment type and optimal dosage for each treatment (Imbens, 2000; Wang et al., 2019; Vegetabile et al., 2021).

To determine the optimal treatment type-dosage pair at a given point in time, we often need to estimate all possible outcomes of treatment type-dosage pair over the future. This requires estimating the impact of treatment type-dosage pair over time on the outcomes, a relationship referred to as a *dose-response curve*. Since we can only observe one outcome from each treatment type-dosage pair



(a) Time-varying treatment types and dosages setting (b) Sequential Counterfactual Discrimination

Figure 1: (a) This is an example setting for 4-step-ahead prediction, and there are two types of treatment with two dosages. $\mathbf{y}_{t+1:t+4}$ (solid line) and $\hat{\mathbf{y}}_{t+1:t+4}$ (dashed line) are the outcomes of the factual trajectory and counterfactual treatment trajectories, respectively. The solid and dashed line boxes denote the factual and counterfactual dose-response curves, and outline characters indicate the counterfactual treatment type and dosage options, respectively. The green densities represent the observation probabilities of the outcomes. (b) By using SCD, we reduce the number of counterfactuals for discriminations in GAN. In this case, we limit only 4 counterfacutal outcomes and 4 dose-response curves for discriminations with SCD.

in patients at a time, we need to estimate unobserved outcomes of other treatment type-dosage pairs, which are called *counterfactual outcomes*.

Several methods have been proposed to estimate outcomes of treatment type-dosage pairs (or dose-response curves for one treatment) in the static setting (Imbens, 2000; Imai and Van Dyk, 2004; Hirano and Imbens, 2004; Schwab et al., 2020; Bica et al., 2020b; Nie et al., 2021; Wang et al., 2022).

An important challenge is to estimate counterfactuals in the longitudinal setting. Specifically, there are mainly two difficulties.

(i). **Sequential selection bias**: When the assignments of treatment type and dosage are sequentially biased due to the time-varying confounders (sequential selection bias), we rarely observe counterfactual treatment trajectories that significantly deviate from the factual trajectory over time or in other patients (Figure 1 (a) green densities). Estimating counterfactual outcomes for long-step-ahead prediction becomes more difficult. State-of-the-art methods build upon long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) or transformer (Vaswani et al., 2017) and handle treatment assignment bias using propensity score (Robins et al., 2000; Lim et al., 2018), g-computation (Li et al., 2021), and balancing methods (Bica et al., 2020a; Melnychuk et al., 2022). However, these methods only consider time-varying treatment types, and not both time-varying treatment types and dosages.

(ii). **Combinatorial explosion**: The number of possible treatment trajectories increases exponentially with τ-step-ahead prediction. As shown in (Figure 1(a)), this is an example setting for 4-step-ahead prediction, and there are two types of treatment with two dosages (e.g. half-dose and full-dose). There are $256 = 4^4$ possible treatment trajectories ($t+1 \sim t+4$) in the longitudinal setting, whereas there are only $4 = 4^1$ possible treatment trajectories ($t+1$) in the static setting.

In this paper, we introduce the Encoder-Decoder Time-SCIGAN (EDTS) that builds upon encoder-decoder transformer and GAN (Goodfellow et al., 2014). EDTS solves the two difficulties by the following two ideas,

(i). **To address the sequential selection bias**, the GAN generates counterfactual outcomes for all treatment type-dosage pairs that cannot be distinguished from observed outcomes by two different discriminators (one for treatment type and one for dosage). This approach can remove sequential selection bias of treatment type and dosage, simultaneously.

(ii). **To address the combinatorial explosion** for the long-step-ahead prediction, we propose a novel discrimination method, Sequential Counterfactual Discrimination (SCD) for EDTS's GAN discriminations, shown in Figure 1 (b). Using SCD, we generate only the first-degree relative counterfactuals of a direct line (the factual trajectory). This approach reduces the number of discriminations, whether observed or generated, to 4 (treatment type) and 4 (dosage), and the GAN can be effectively trained by using only predictable counterfactuals for training.

We evaluate our EDTS on synthetic and semi-synthetic datasets, and it achieves superior performance over the current baselines. To the best of our knowledge, this is the first study to estimate counterfactual outcomes of both time-varying treatment types and dosages using GAN to handle the bias due to time-varying confounders.

2. Related work

2.1. Estimating dose-response curves in the static setting

Several studies have proposed methods to estimate treatment effects (Louizos et al., 2017; Yoon et al., 2018; Ghosh et al., 2021, 2023) or dose-response curves, such as DRNets (Dose Response networks) by Schwab et al. (2020) and SCIGAN (eStimating the effects of Continuous Interventions using GAN) by Bica et al. (2020b).

DRNets estimate dose-response curves of continuous treatment (treatment with a continuous dosage) using a balancing representations method to handle treatment type assignment bias. This method learns an invariant treatment representation, which is simultaneously predictive of the outcome but non-predictive of the treatment type.

SCIGAN estimates dose-response curves using GAN to handle both treatment type and dosage assignment bias. SCIGAN treats input dosages as discrete values for training, so the choice of dosage-level is limited to the number of dosage samples. After training, the trained generator can generate treatment outcomes at any continuous dosage value. Since SCIGAN is for the static setting and the input is given by cross-sectional data, our longitudinal setup with time-varying covariates, treatments, and outcomes cannot be handled.

2.2. Estimating counterfactual outcomes in the longitudinal setting

Methods for estimating time-varying outcomes were originally introduced in epidemiology. Examples of such methods include G-computation, marginal structural models (MSMs), and structural nested models (Robins, 1986; Robins et al., 2000; Hernán et al., 2001; Robins and Hernán, 2009). Here, state-of-the-art methods are RMSNs (Recurrent Marginal Structural Networks) (Lim et al., 2018), CRN (Causal Recurrent Network) (Bica et al., 2020a), G-Net (Li et al., 2021), CT (Causal Transformer), and EDCT (Encoder-Decoder Causal Transforer) (Melnychuk et al., 2022). These methods address bias due to time-varying confounding in different ways. RMSNs combine two propensity networks and use the predicted inverse probability of treatment weighting (IPTW) scores for training the prediction networks. G-Net aims to predict both outcomes and time-varying covariates, and then performs G-computation for multiple-step-ahead prediction. CRN, CT and EDCT use balancing representaions method. However, the previous studies consider only time-varying treatment types and their assignment bias, without considering time-varying dosages and dosage assignment bias.

3. Problem formulation

We build upon the standard setting for estimating counterfactual outcomes in the longitudinal setting as in Robins and Hernán (2009); Lim et al. (2018); Bica et al. (2020a); Li et al. (2021); Melnychuk et al. (2022).

Suppose that there are N patients and an i-th patient has T times steps for $i = 1, ..., N$. We also consider a space of treatment types. There are $n_{\mathcal{W}}$ types of treatments in total, and thus we define $\mathcal{W} = \{w^k\}_{k=1}^{n_{\mathcal{W}}}$ as a set of indices of the treatments. Further, for each treatment $k = 1, ..., n_{\mathcal{W}}$, we define a space of discretized dosages $\mathcal{D}^k = \{d^{k,j}\}_{j=1}^{n_{\mathcal{D}}^k}$ with $n_{\mathcal{D}}^k$ values and its combination $\mathcal{D} = \mathcal{D}^1 \times \cdots \times \mathcal{D}^{n_{\mathcal{W}}}$.

We define our observations. For the patient i and each time step $t = 1, ..., T$, we observe time-varying covariates $\mathrm{X}_t^i \in \mathbb{R}^{d_x}$, outcomes $\mathrm{y}_t^i \in \mathbb{R}^{d_y}$, treatments $(\mathrm{W}_t^i, \mathrm{D}_t^i) \in \mathcal{W} \times \mathcal{D}$, and static covariates describing a patient $\mathrm{V}^i \in \mathbb{R}^{d_v}$, such as gender and age. We have access to the following observations

$$\mathfrak{D} = \left\{ \left\{ \mathrm{X}_t^i, \mathrm{y}_t^i, \mathrm{W}_t^i, \mathrm{D}_t^i \right\}_{t=1}^{T} \cup \mathrm{V}^i \right\}_{i=1}^{N}. \tag{1}$$

Here, we summarize the observations and define histories of the patients as

$$\mathbf{H}_{1:t} = \{\mathbf{X}_{1:t}, \mathbf{y}_{1:t}, \mathbf{W}_{1:t-1}, \mathbf{D}_{1:t-1}, \mathbf{V}\}, \tag{2}$$

where $\mathbf{X}_t = \{X_t^i\}_{i=1}^N$ and $\mathbf{X}_{1:t} = \{\mathbf{X}_1, \ldots, \mathbf{X}_t\}$. $\mathbf{y}_{1:t}, \mathbf{W}_{1:t}, \mathbf{D}_{1:t}$ and $\mathbf{V}$ are defined in the same way. Note that if some patient has a shorter treatment period than T, we mask the remaining time steps.

We build on the potential outcome framework (Neyman, 1923; Rubin, 1978) and its extension to time-varying treatments (Robins and Hernán, 2009). Let $\tau \geq 1$ denote the prediction horizon for τ-step-ahead prediction, and let $(\mathbf{w}_{t:t+\tau-1}, \mathbf{d}_{t:t+\tau-1}) \in (\mathcal{W} \times \mathcal{D})^{\times \tau}$ denote a given (non-random) treatment intervention over the horizon. Then, we define a potential outcome $\mathbf{Y}_{t+\tau}$ with the invention $(\mathbf{w}_{t:t+\tau-1}, \mathbf{d}_{t:t+\tau-1})$, which is typically not observed.

Our goal is thus to estimate potential outcomes $\mathbf{Y}_{t+\tau}$, after applying a treatment intervention $(\mathbf{w}_{t:t+\tau-1}, \mathbf{d}_{t:t+\tau-1})$ for a given patient history $\mathbf{H}_{1:t}$. Formally, we aim to estimate:

$$\mathbb{E}(\mathbf{Y}_{t+\tau} \mid (\mathbf{w}_{t:t+\tau-1}, \mathbf{d}_{t:t+\tau-1}), \mathbf{H}_{1:t}). \tag{3}$$

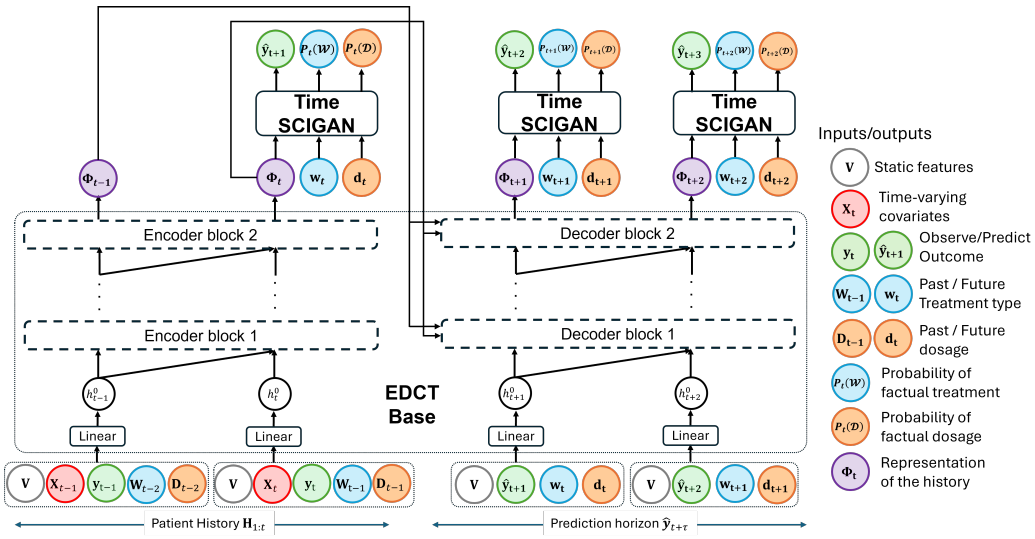


Figure 2: Overview of Encoder-Decoder Time-SCIGAN (EDTS).

Note that the potential counterfactual outcomes are identifiable from factual observational data $\mathfrak{D}$ under three standard assumptions: (1) consistency, (2) sequential ignorability, and (3) sequential overlap (see Appendix A for details).

4. Encoder-Decoder Time-SCIGAN (EDTS)

4.1. Overview of EDTS

We first give an overview of our proposal, *Encoder-Decoder Time-SCIGAN (EDTS)*, which is an Encoder-Decoder architecture and consists of EDCT-Base $\mathscr{R}$ and Time-SCIGAN as shown in Figure 2. The encoder predicts one-step-ahead prediction $\hat{\mathbf{y}}_{t+1}(\mathbf{w}_t, \mathbf{d}_t)$, whereas the decoder predicts multi-step-ahead prediction $\hat{\mathbf{y}}_{t+\tau}(\mathbf{w}_{t+\tau-1}, \mathbf{d}_{t+\tau-1})$ with $\tau > 1$.

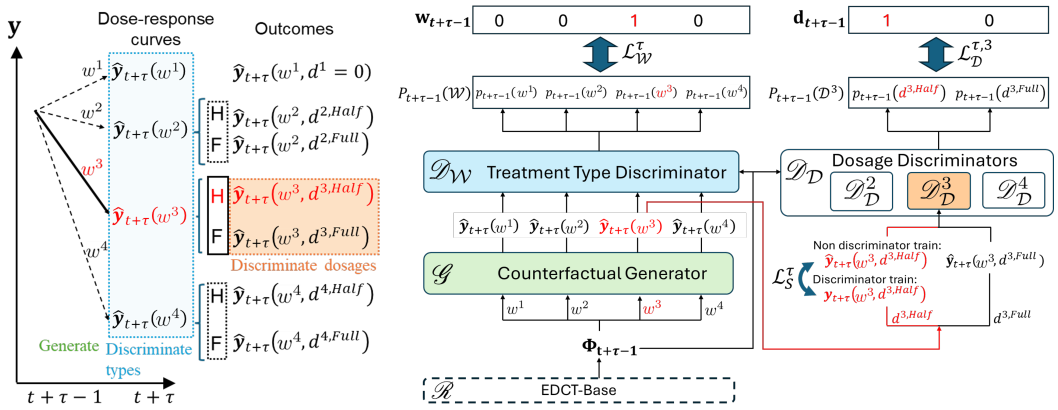
EDCT-Base $\mathscr{R}$ takes $\mathbf{H}_{1:t}$ as an argument and outputs its representation $\mathbf{\Phi}_t$. EDCT-Base is identical to the original EDCT (Melnychuk et al., 2022) excluding the outcome prediction and treatment classifier networks. The detailed architecture of EDCT-Base is presented in Appendix B.

Time-SCIGAN is an extension of SCIGAN (Bica et al., 2020b), which is designed for static settings, to accommodate longitudinal settings. Time-SCIGAN generates counterfactual outcomes of treatment type-dosage pairs from $\mathbf{\Phi}_t$. The details are presented in subsection 4.3.

We note that a straightforward combination of Causal Transformer (EDCT) and SCIGAN, which we applied for our EDTS, resulted in poor training. Adversarial training models such as GANs are difficult to train, especially in the longitudinal setting. GAN models that work well in the static setting do not easily generalize to the longitudinal setting due to the instability of the training of GANs. To address this problem, we introduce SCD and two different weights into Time-SCIGAN.

4.2. Sequential counterfactual discrimination (SCD)

For multi-step ahead prediction (Figure 1(a)), there are numerous counterfactuals (outcomes and dose-response curves) for possible treatment trajectories. If we generate all counterfactuals and then



(a) Example setting (b) Architecture of Time-SCIGAN

Figure 3: Overview of Time-SCIGAN. (a) Example setting with four treatment types and two dosages at τ step ahead prediction from current t. w^1 represents no treatment and $w^2 \sim w^4$ represent treatment types with two dosages (H: a half-dose, F: a full-dose) each. Factual treatment type $\mathbf{w}_{t+\tau-1}$ is w^3 and factual dosage $\mathbf{d}_{t+\tau-1}$ is $d^{3,Half}$. For notation, red denotes factual objects, and black denotes counterfactual objects.

discriminate whether they were observed (factual) or generated (counterfactual) for training, the traning of GANs can be ineffective. This is because counterfactuals that significantly deviate from the factual trajectory are infrequently observed over time or in other patients, making them difficult to predict. To train the GAN effectively, we should select only counterfactuals that are closely related to the factual trajectory.

To address this problem, we propose *Sequential Counterfactual Discrimination* (SCD) to limit the number of counterfactuals used for discrimination. SCD focuses on first-degree relatives (counterfactuals) of the factual trajectory, which are direct siblings of the factuals (as shown in Figure 1(b)) and excludes more distant relatives (etc. nieces and nephews). SCD limits counterfactuals vertically (relatives) along the time step, not horizontally (descendants).

SCD focuses on partial trajectories, but it has more utility than ignoring less common trajectories. Most counterfactual trajectories are infrequently observed over time or in other patients, hence ignoring them does not have a significant impact on discrimination and generation. In addition, this approach has the advantage of avoiding the exponential computational costs. In the case of Figure 1, we generate only 4 counterfactual outcomes and 4 counterfactual dose-response curves with SCD (Figure 1(b)), compared to 336 and 156 without SCD (Figure 1(a)). With SCD, we discriminate only 4 times by treatment type and dosage discriminators, and by using only predictable counterfactuals, the GAN can be effectively trained. When including counterfactuals of second-degree relatives or higher, the GAN become untrainable.

4.3. Time-SCIGAN

Time-SCIGAN consists of counterfactual generator $\mathscr{G}$, treatment type discriminator $\mathscr{D}_{\mathcal{W}}$, and dosage discriminators $\mathscr{D}_{\mathcal{D}} = \{\mathscr{D}_{\mathcal{D}}^k\}_{k=1}^{n_{\mathcal{W}}}$. Note that we utilize the dosage discriminator $\mathscr{D}_{\mathcal{D}}^k$ for a treatment $k = 1, ..., n_{\mathcal{W}}$ when the treatment k has more than one dosage option, since no discriminator is

necessary for a treatment with only one dosage option. An overview of Time-SCIGAN is shown in Figure 3, and the detailed architectures of $\mathscr{G}$, $\mathscr{D}_{\mathcal{W}}$ and $\mathscr{D}_{\mathcal{D}}$ are presented in Appendix C.

At τ step ahead prediction (encoder: $\tau = 1$, decoder: $\tau = 2 \sim \tau^{max}$), Time-SCIGAN takes $\mathbf{\Phi}_{t+\tau-1}$, $\mathbf{w}_{t+\tau-1}$ and $\mathbf{d}_{t+\tau-1}$ as input, and outputs $\hat{\mathbf{y}}_{t+\tau}$, $P_{t+\tau-1}(\mathcal{W})$ and $P_{t+\tau-1}(\mathcal{D})$, where $P_{t+\tau-1}(\mathcal{W})$ represents the probability that the dose-response curve corresponds to either factual or counterfactual treatment, and $P_{t+\tau-1}(\mathcal{D})$ represents the probability that the outcome corresponds to either factual or counterfactual dosage (shown in Figure 3(b)).

In the original SCIGAN, the same weight is used to calculate the discriminator loss. In our Time-SCIGAN, we introduce two different weights: scale and factual weights. The scale weights address the unbalanced scales caused by significant outcome changes in the longitudinal setting. The factual weights handle the unbalanced labels between factual and counterfactual treatment types or dosages.

COUNTERFACTUAL GENERATOR

The counterfactual generator $\mathscr{G}$ receives $\mathbf{\Phi}_t$ from EDCT-Base $\mathscr{R}$, and outputs all dose-response curves $\hat{\mathbf{y}}_{t+\tau}(\mathcal{W}) = \{\hat{\mathbf{y}}_{t+\tau}(w^k)\}_{k=1}^{n_{\mathcal{W}}}$, where $\hat{\mathbf{y}}_{t+\tau}(w^k) = \{\hat{\mathbf{y}}_{t+\tau}(w^k, d^{k,j})\}_{j=1}^{n_{\mathcal{D}}^k}$. The supervised loss $\mathcal{L}_S^\tau$ is defined via the mean squared error (MSE) as follows,

$$\begin{aligned}\mathcal{L}_S^\tau(\mathscr{G}, \mathscr{R}) =& |\mathbf{y}_{t+\tau}(\mathbf{w}_{t+\tau-1}, \mathbf{d}_{t+\tau-1}) - \mathscr{G}(\mathbf{\Phi}_{t+\tau-1})(\mathbf{w}_{t+\tau-1}, \mathbf{d}_{t+\tau-1})|^2 \\ =& |\mathbf{y}_{t+\tau}(\mathbf{w}_{t+\tau-1}, \mathbf{d}_{t+\tau-1}) - \hat{\mathbf{y}}_{t+\tau}(\mathbf{w}_{t+\tau-1}, \mathbf{d}_{t+\tau-1})|^2. \end{aligned} \tag{4}$$

When training $\mathscr{D}_{\mathcal{W}}$ and $\mathscr{D}_{\mathcal{D}}$, we replace the predicted factual outcome $\hat{\mathbf{y}}_{t+\tau}(\mathbf{w}_{t+\tau-1}, \mathbf{d}_{t+\tau-1}) \in \hat{\mathbf{y}}_{t+\tau}(\mathbf{w}_{t+\tau-1})$ with the observed factual outcome $\mathbf{y}_{t+\tau}(\mathbf{w}_{t+\tau-1}, \mathbf{d}_{t+\tau-1})$. For notation, let $\mathbf{y}(w)$ denote a dose-response curve, $\mathbf{y}(w, d)$ denote an outcome, and $\hat{}$ denote a predicted value.

TREATMENT TYPE DISCRIMINATOR

The treatment type discriminator $\mathscr{D}_{\mathcal{W}}$ receives all dose-response curves $\hat{\mathbf{y}}_{t+\tau}(\mathcal{W})$ from counterfactual generator $\mathscr{G}$ and $\mathbf{\Phi}_{t+\tau-1}$ from EDCT-Base $\mathscr{R}$, and outputs $P_{t+\tau-1}(\mathcal{W})$. The treatment type discriminator loss $\mathcal{L}_{\mathcal{W}}^\tau$ is calculated as follows,

$$\mathcal{L}_{\mathcal{W}}^\tau(\mathscr{D}_{\mathcal{W}}; \mathscr{G}, \mathscr{R}, r_{\mathcal{W}}) \tag{5}$$

$$\begin{aligned} =& \sum_{k=1}^{n_{\mathcal{W}}} \frac{\mathbf{y}_{t+\tau}(\mathbf{w}_{t+\tau-1}, \mathbf{d}_{t+\tau-1})}{\mathbf{y}^{max}} \Big[\mathbb{1}_{\{w^k = \mathbf{w}_{t+\tau-1}\}} r_{\mathcal{W}} \log \mathscr{D}_{\mathcal{W}} \left(\hat{\mathbf{y}}_{t+\tau}(w^k), \mathbf{\Phi}_{t+\tau-1} \right) \\ & + \mathbb{1}_{\{w^k \neq \mathbf{w}_{t+\tau-1}\}} (1 - r_{\mathcal{W}}) \log \left(1 - \mathscr{D}_{\mathcal{W}} \left(\hat{\mathbf{y}}_{t+\tau}(w^k), \mathbf{\Phi}_{t+\tau-1} \right) \right) \Big], \end{aligned} \tag{6}$$

where $\mathbb{1}_{[\cdot]}$ is the indicator function, $r_{\mathcal{W}}$ is factual weights for treatment types, respectively, and $\mathbf{y}^{max}$ is the maximum of $\mathbf{y}$.

$\mathbf{y}_{t+\tau}(\mathbf{w}_{t+\tau-1}, \mathbf{d}_{t+\tau-1})/\mathbf{y}^{max}$ are scale weights to address the unbalanced scales between the supervised loss $\mathcal{L}_S^\tau$ and the discriminator loss $\mathcal{L}_{\mathcal{W}}^\tau$ due to significant outcome changes in the longitudinal setting. $r_{\mathcal{W}}$ and $(1 - r_{\mathcal{W}})$ are factual and counterfactual weights to address the unbalanced labels between factual and counterfactual treatment types. During training $\mathscr{D}_{\mathcal{W}}$, we set $r_{\mathcal{W}} = (n_{\mathcal{W}} - 1)/n_{\mathcal{W}}$ (in the case of Figure 3(a), $r_{\mathcal{W}}$ is 0.75).

DOSAGE DISCRIMINATOR

The dosage discriminators $\mathscr{D}_{\mathcal{D}}$ receive the factual dose-response curve $\hat{\mathbf{y}}_{t+\tau}(\mathbf{w}_{t+\tau-1})$ from $\mathscr{G}$ and $\mathbf{\Phi}_{t+\tau-1}$ from $\mathscr{R}$, and output $P_t(\mathcal{D})$. We calculate the dosage discriminator loss $\mathcal{L}_{\mathcal{D}}^{\tau,k}$ for treatment w^k as follows,

$$\mathcal{L}_{\mathcal{D}}^{\tau,k}(\mathscr{D}_{\mathcal{D}}^k;\mathscr{G},\mathscr{R},r_{\mathcal{D}}^k) \tag{7}$$

$$= \mathbb{1}_{\{w^k=\mathbf{w}_{t+\tau-1}\}} \sum_{j=1}^{n_{\mathcal{D}}^k} \frac{\mathbf{y}_{t+\tau}(\mathbf{w}_{t+\tau-1},\mathbf{d}_{t+\tau-1})}{\mathbf{y}^{max}} \Big[\mathbb{1}_{\{d^{k,j}=\mathbf{d}_{t+\tau-1}\}} r_{\mathcal{D}}^k \log\left(\mathscr{D}_{\mathcal{D}}^k\left(\hat{\mathbf{y}}_{t+\tau}(w^k,d^{k,j}),\mathbf{\Phi}_{t+\tau-1}\right)\right)$$
$$+ \mathbb{1}_{\{d^{k,j}\neq\mathbf{d}_{t+\tau-1}\}}\left(1-r_{\mathcal{D}}^k\right)\log\left(1-\mathscr{D}_{\mathcal{D}}^k\left(\hat{\mathbf{y}}_{t+\tau}(w^k,d^{k,j}),\mathbf{\Phi}_{t+\tau-1}\right)\right)\Big], \tag{8}$$

where $\mathbb{1}_{\{w^k=\mathbf{w}_{t+\tau-1}\}}$ term ensures that only factual dose-response curve is used for $\mathscr{D}_{\mathcal{D}}^k$. $r_{\mathcal{D}}^k$ and $(1-r_{\mathcal{D}}^k)$ are factual and counterfactual weights for dosages of treatment type w^k. During training $\mathscr{D}_{\mathcal{D}}^k$, we set $r_{\mathcal{D}}^k = (n_{\mathcal{D}}^k - 1)/n_{\mathcal{D}}^k$ (in the case of Figure 3(a), $r_{\mathcal{D}}^k$ is 0.5).

OVERALL DISCRIMINATOR LOSS

To train the non-discriminator parameters $(\mathscr{G},\mathscr{R})$, we calculate the overall discriminator loss $\mathcal{L}_{\mathcal{O}}^{\tau}$ by summing $\mathcal{L}_{\mathcal{W}}^{\tau}$ and $\mathcal{L}_{\mathcal{D}}^{\tau,k}$, as follows,

$$\mathcal{L}_{\mathcal{O}}^{\tau}(\mathscr{D}_{\mathcal{W}},\mathscr{D}_{\mathcal{D}};\mathscr{G},\mathscr{R}) = \mathcal{L}_{\mathcal{W}}^{\tau}(\mathscr{D}_{\mathcal{W}};\mathscr{G},\mathscr{R},r_{\mathcal{W}}) + \sum_{k=1}^{n_{\mathcal{W}}} \mathcal{L}_{\mathcal{D}}^{\tau,k}(\mathscr{D}_{\mathcal{D}}^k;\mathscr{G},\mathscr{R},r_{\mathcal{D}}^k). \tag{9}$$

The factual outcomes are trained using the supervised loss in Eq. (4) and the overall discriminator loss in Eq. (9). However, the counterfactual outcomes are trained only via the overall discriminator loss. We want to estimate the counterfactual outcomes rather than factual outcomes, therefore, we set factual weights equal to counterfactual weights ($r_{\mathcal{W}} = 0.5, r_{\mathcal{D}}^k = 0.5$) unlike when training the treatment type or dosage discriminators.

4.4. Training of EDTS

ENCODER

We perform the encoder-decoder two-stage training. To address confounding bias, we use the GAN with two simultaneous objectives, non-discriminator $(\mathscr{G},\mathscr{R})$ and discriminator $(\mathscr{D}_{\mathcal{W}},\mathscr{D}_{\mathcal{D}})$. $(\mathscr{G},\mathscr{R})$ and $(\mathscr{D}_{\mathcal{W}},\mathscr{D}_{\mathcal{D}})$ are trained according to the minimax game defined as follows,

$$(\hat{\mathscr{G}},\hat{\mathscr{R}}) = \arg\min_{\mathscr{G},\mathscr{R}} \mathcal{L}_S^1(\mathscr{G},\mathscr{R}) - \alpha\mathcal{L}_{\mathcal{O}}^1(\hat{\mathscr{D}}_{\mathcal{W}},\hat{\mathscr{D}}_{\mathcal{D}};\mathscr{G},\mathscr{R}) \tag{10}$$

$$(\hat{\mathscr{D}}_{\mathcal{W}},\hat{\mathscr{D}}_{\mathcal{D}}) = \begin{cases} \hat{\mathscr{D}}_{\mathcal{W}} = \arg\min_{\mathscr{D}_{\mathcal{W}}} \alpha\mathcal{L}_{\mathcal{W}}^1(\mathscr{D}_{\mathcal{W}};\hat{\mathscr{G}},\hat{\mathscr{R}}) \\ \hat{\mathscr{D}}_{\mathcal{D}}^k = \arg\min_{\mathscr{D}_{\mathcal{D}}^k} \alpha\mathcal{L}_{\mathscr{D}}^{1,k}(\mathscr{D}_{\mathcal{D}}^k;\hat{\mathscr{G}},\hat{\mathscr{R}}) \ for\ each\ w^k \end{cases} \tag{11}$$

where $\alpha \geq 0$ is the Overall Discriminator (OD) coefficient.

DECODER

Each loss is summarized in Eq. (10-11) for τ_{max}-step-ahead prediction using $\mathcal{L}^{2:\tau_{max}} = \sum_{\tau=2}^{\tau_{max}} \mathcal{L}^{\tau}$, where $\mathcal{L}^{\tau} = \{\mathcal{L}_S^{\tau}, \mathcal{L}_O^{\tau}, \mathcal{L}_{\mathcal{W}}^{\tau}, \mathcal{L}_{\mathcal{D}}^{\tau,k}\}$. The decoder parameters $(\mathcal{G}, \mathcal{R}, \mathcal{D}_{\mathcal{W}}, \mathcal{D}_{\mathcal{D}})$ are trained in the same manner as the encoder. During the training of the decoder, we use the teacher forcing technique (Williams and Zipser, 1989), and we switch off the teacher forcing and autoregressively feed model predictions during the evaluation of the multiple-step-ahead prediction.

4.5. Theoretical insights

We develop a theoretical result to state that the overall discriminator loss (eq. 9) aims to remove the time-varing confounding bias ($p(\mathbf{W}_t|\mathbf{H}_{1:t})$ and $p(\mathbf{D}_t|\mathbf{H}_{1:t})$). This result implies that the GAN allows us to obtain a low error in estimating counterfactuals for all treatments.

Let $\Phi(\mathbf{H}_{1:t}) = \mathcal{G}(\mathcal{R}(\mathbf{H}_{1:t}))$. Let P_k and P_k^{Φ} denote the distributions of $\mathbf{H}_{1:t}$ and $\Phi(\mathbf{H}_{1:t})$ conditional on $\mathbf{W}_t = w^k$, respectively. Let $P_{k,j}$ and $P_{k,j}^{\Phi}$ denote the distributions of $\mathbf{H}_{1:t}$ and $\Phi(\mathbf{H}_{1:t})$ conditional on $\mathbf{W}_t = w^k$ and $\mathbf{D}_t = d^{k,j}$, respectively. Let $\mathcal{D}_{\mathcal{W}}^k$ and $\mathcal{D}_{\mathcal{D}}^{k,j}$ denote the output of $\mathcal{D}_{\mathcal{W}}$ corresponding to w^k and $\mathcal{D}_{\mathcal{D}}^k$ corresponding to $d^{k,j}$, respectively.

Theorem 1 *The minimax game is defined as follows:*

$$\min_{\Phi} \max_{\mathcal{D}_{\mathcal{W}}, \mathcal{D}_{\mathcal{D}}} \sum_{k=1}^{n_{\mathcal{W}}} \left(\mathbb{E}_{\boldsymbol{H}_t \sim P_k} \left[\log \left(\mathcal{D}_{\mathcal{W}}^k \left(\Phi(\boldsymbol{H}_{1:t}) \right) \right) \right] + \sum_{j=1}^{n_{\mathcal{D}}^k} \mathbb{E}_{\boldsymbol{H}_t \sim P_{k,j}} \left[\log \left(\mathcal{D}_{\mathcal{D}}^{k,j} \left(\Phi(\boldsymbol{H}_{1:t}) \right) \right) \right] \right) \tag{12}$$

$$\textit{subject to } \sum_{k=1}^{n_{\mathcal{W}}} \mathcal{D}_{\mathcal{W}}^k \left(\Phi(\boldsymbol{H}_{1:t}) \right) = 1 \textit{ and } \sum_{j=1}^{n_{\mathcal{D}}^k} \mathcal{D}_{\mathcal{D}}^{k,j} \left(\Phi(\boldsymbol{H}_{1:t}) \right) = 1 \tag{13}$$

The minimax game has a global minimum of Φ which is attained if and only if $P_1^{\Phi} = P_2^{\Phi} = \cdots = P_{n_{\mathcal{W}}}^{\Phi}$ and $P_{k,1}^{\Phi} = P_{k,2}^{\Phi} = \cdots = P_{k,n_{\mathcal{D}}^k}^{\Phi}$ for each treatment w^k, i.e. when the learned counterfactual generators are invariant across all treatment type and dosage pairs.

Proof See Appendix D. ■

5. Experiments

Baselines

In our study, the baselines are identical to those in the previous, state-of-the-art literature for estimating counterfactual outcomes of time-varying treatment types. These include MSMs (Robins et al., 2000), RMSNs (Lim et al., 2018), CRN (Bica et al., 2020a), G-Net (Li et al., 2021), CT and EDCT (Melnychuk et al., 2022). For the new setting of time-varying treatment types and dosages, we modified those baseline methods to add dosages to their inputs. Furthermore, we modified CRN, CT and EDCT to add a dosage classifier network to handle dosage assignment bias (CRN+, CT+, and EDCT+). The baseline methods are detailed in Appendix E,

Training

We applied exponential growth to α, and used an exponential moving average (EMA) (Yazıcı et al., 2018) during training as well as CT (Melnychuk et al., 2022), in order to avoid the difficulty of training adversarial models such as GANs especially in the longitudinal setting. The details of training procedures and hyperparameter tuning are described in Appendix F.

Table 1: Normalized RMSE for τ-step-ahead prediction in the setting of time-varying treatment types and dosages. Shown: mean ± standard deviation over five runs (lower is better). γ_w and γ_d represent the amount of time-varying confounding parameters, with higher values indicating larger assignment bias of treatment and dosage, respectively.

		$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$	$\tau = 6$
$\gamma_w = 0$	MSMs	0.88 (±0.08)	1.35 (±0.10)	1.89 (±0.14)	1.68 (±0.12)	1.49 (±0.11)	1.32 (±0.10)
$\gamma_d = 0$	RMSNs	0.53 (±0.02)	0.66 (±0.03)	0.72 (±0.04)	0.73 (±0.06)	0.72 (±0.06)	0.70 (±0.07)
	CRN+	0.52 (±0.04)	0.66 (±0.04)	0.73 (±0.05)	0.74 (±0.06)	0.73 (±0.05)	0.70 (±0.05)
	G-Net	0.53 (±0.03)	0.69 (±0.05)	0.75 (±0.05)	0.77 (±0.06)	0.76 (±0.06)	0.74 (±0.06)
	CT+	0.63 (±0.05)	0.77 (±0.05)	0.81 (±0.05)	0.82 (±0.06)	0.79 (±0.06)	0.76 (±0.06)
	EDCT+	0.50 (±0.05)	0.63 (±0.05)	0.68 (±0.05)	0.68 (±0.06)	0.66 (±0.05)	0.63 (±0.05)
	EDTS($\alpha = 0$)	0.49 (±0.05)	0.62 (±0.05)	0.68 (±0.05)	0.69 (±0.06)	0.67 (±0.05)	0.64 (±0.05)
	EDTS	**0.48 (±0.04)**	**0.61 (±0.04)**	**0.67 (±0.05)**	**0.67 (±0.05)**	**0.65 (±0.05)**	**0.62 (±0.05)**
$\gamma_w = 1$	MSMs	1.09 (±0.10)	1.90 (±0.16)	2.61 (±0.19)	2.30 (±0.17)	2.01 (±0.15)	1.77 (±0.13)
$\gamma_d = 1$	RMSNs	0.58 (±0.06)	0.83 (±0.10)	0.92 (±0.10)	0.94 (±0.10)	0.92 (±0.10)	0.88 (±0.10)
	CRN+	0.54 (±0.05)	0.77 (±0.09)	0.86 (±0.11)	0.90 (±0.12)	0.89 (±0.12)	0.86 (±0.11)
	G-Net	0.58 (±0.09)	0.85 (±0.18)	0.97 (±0.23)	1.04 (±0.28)	1.06 (±0.33)	1.05 (±0.35)
	CT+	0.67 (±0.05)	0.89 (±0.08)	0.97 (±0.09)	0.99 (±0.09)	0.98 (±0.10)	0.95 (±0.09)
	EDCT+	0.68 (±0.05)	0.91 (±0.09)	0.98 (±0.10)	0.99 (±0.10)	0.98 (±0.10)	0.94 (±0.10)
	EDTS($\alpha = 0$)	**0.49 (±0.05)**	**0.70 (±0.07)**	0.78 (±0.07)	0.80 (±0.07)	0.79 (±0.08)	0.76 (±0.08)
	EDTS	0.50 (±0.05)	**0.70 (±0.07)**	**0.77 (±0.08)**	**0.79 (±0.09)**	**0.78 (±0.09)**	**0.75 (±0.09)**
$\gamma_w = 2$	MSMs	1.46 (±0.12)	2.69 (±0.23)	3.57 (±0.30)	3.10 (±0.28)	2.68 (±0.25)	2.32 (±0.21)
$\gamma_d = 2$	RMSNs	0.66 (±0.05)	1.01 (±0.11)	1.14 (±0.12)	1.17 (±0.12)	1.14 (±0.11)	1.09 (±0.10)
	CRN+	0.64 (±0.05)	0.98 (±0.11)	1.13 (±0.13)	1.19 (±0.13)	1.18 (±0.10)	1.14 (±0.12)
	G-Net	0.61 (±0.06)	0.94 (±0.12)	1.09 (±0.14)	1.16 (±0.15)	1.17 (±0.15)	1.14 (±0.15)
	CT+	0.70 (±0.05)	1.04 (±0.11)	1.18 (±0.13)	1.24 (±0.14)	1.24 (±0.16)	1.22 (±0.16)
	EDCT+	0.62 (±0.06)	0.91 (±0.10)	1.00 (±0.11)	1.03 (±0.12)	1.00 (±0.12)	0.96 (±0.11)
	EDTS($\alpha = 0$)	0.59 (±0.10)	**0.86 (±0.11)**	**0.96 (±0.11)**	0.99 (±0.11)	0.97 (±0.11)	**0.92 (±0.10)**
	EDTS	**0.58 (±0.08)**	**0.86 (±0.10)**	**0.96 (±0.11)**	**0.98 (±0.11)**	**0.96 (±0.11)**	**0.92 (±0.10)**
$\gamma_w = 3$	MSMs	1.91 (±0.16)	3.47 (±0.30)	4.50 (±0.38)	3.74 (±0.33)	3.07 (±0.29)	2.54 (±0.25)
$\gamma_d = 3$	RMSNs	**0.76 (±0.10)**	1.43 (±0.18)	1.57 (±0.17)	1.61 (±0.14)	1.60 (±0.12)	1.57 (±0.11)
	CRN+	0.86 (±0.09)	1.34 (±0.16)	1.53 (±0.16)	1.58 (±0.15)	1.57 (±0.16)	1.53 (±0.17)
	G-Net	0.76 (±0.06)	1.25 (±0.12)	1.47 (±0.16)	1.56 (±0.19)	1.57 (±0.21)	1.54 (±0.23)
	CT+	0.93 (±0.13)	1.39 (±0.17)	1.60 (±0.16)	1.68 (±0.15)	1.70 (±0.12)	1.68 (±0.11)
	EDCT+	0.81 (±0.13)	1.35 (±0.16)	1.52 (±0.15)	1.54 (±0.12)	1.52 (±0.11)	1.49 (±0.11)
	EDTS($\alpha = 0$)	**0.73 (±0.10)**	**1.18 (±0.13)**	1.36 (±0.15)	1.41 (±0.16)	1.41 (±0.17)	1.37 (±0.18)
	EDTS	0.75 (±0.12)	**1.18 (±0.13)**	**1.34 (±0.13)**	**1.38 (±0.13)**	**1.35 (±0.13)**	**1.29 (±0.12)**
$\gamma_w = 4$	MSMs	2.37 (±0.18)	3.89 (±0.34)	4.58 (±0.44)	3.46 (±0.34)	2.77 (±0.28)	2.45 (±0.25)
$\gamma_d = 4$	RMSNs	1.07 (±0.04)	2.04 (±0.17)	2.19 (±0.23)	2.18 (±0.24)	2.10 (±0.23)	1.97 (±0.22)
	CRN+	1.01 (±0.16)	1.72 (±0.20)	2.03 (±0.23)	2.18 (±0.25)	2.24 (±0.28)	2.23 (±0.31)
	G-Net	0.98 (±0.17)	1.79 (±0.25)	2.12 (±0.26)	2.26 (±0.24)	2.31 (±0.23)	2.29 (±0.23)
	CT+	1.39 (±0.26)	2.05 (±0.36)	2.33 (±0.39)	2.42 (±0.37)	2.43 (±0.33)	2.38 (±0.31)
	EDCT+	1.11 (±0.17)	2.20 (±0.27)	2.36 (±0.28)	2.32 (±0.26)	2.23 (±0.25)	2.15 (±0.29)
	EDTS($\alpha = 0$)	1.13 (±0.17)	1.72 (±0.27)	1.91 (±0.28)	1.95 (±0.28)	1.93 (±0.29)	1.91 (±0.36)
	EDTS	**0.90 (±0.10)**	**1.57 (±0.13)**	**1.82 (±0.18)**	**1.91 (±0.21)**	**1.90 (±0.22)**	**1.85 (±0.22)**

5.1. Experiments with fully-synthetic data

SIMULATION MODEL

We use the pharmacokinetic-pharmacodynamic model of tumor growth (Geng et al., 2017). This model simulates the combined effects of chemotherapy and radiotherapy in patients with lung cancer. The same model was previously used to evaluate RMSNs (Lim et al., 2018), CRN (Bica et al., 2020a), G-Net (Li et al., 2021), CT and EDCT (Melnychuk et al., 2022). In the experimental setting of time-varying treatment types and dosages, full (regular) and half dosages were chosen for each treatment type at each time step, and the treatment dosages were used as model inputs. The simulation details are provided in Appendix G.

RESULTS

Table 1 presents the results. EDTS has a consistent and significant improvement across all prediction horizons τ. There is a notable performance gain in EDTS over the state-of-the-art baselines, especially pronounced for larger confounding γ and larger τ. Furthermore, by comparing our EDTS against EDTS ($\alpha = 0$), we see clear performance gains. Positive α is effective in the case of high selection bias (γ_w, $\gamma_d = 3$ and 4). It becomes difficult to predict the counterfactual outcomes using only the transformer architecture without discriminators ($\alpha = 0$), so there is a remarkable difference (about 2.3% and 5.7% when γ_w, $\gamma_d = 3$ and 4, respectively) between the proposed model with $\alpha = 0$ and with positive values on α.

In Appendix G.4, we also compared our EDTS performance with a prior studies setting of time-varying treatment types. For this setting, we modified EDTS to omit the input of dosages and dosage discriminators (EDTS-).

5.2. Experiments with semi-synthetic data

DATASET DESCRIPTIONS

To validate baseline models with multi-output outcomes, high-dimensional covariates and complex feature interactions, CT (Melnychuk et al., 2022) use a semi-synthetic dataset based on real-world medical data from MIMIC-III dataset (Johnson et al., 2016). Instead of MIMIC-III dataset, we use high-dimensional real-world vital data from ElectroEncephaloGraphy (EEG) dataset (Zhang et al., 1995).

We use a similar approach to CT to create semi-synthetic dataset from real-world data. We generate patient trajectories with outcomes under endogeneous and exogeneous dependencies while considering treatment effects. This allows us control for the amount of confounding and access ground-truth counterfactuals for evaluation. Details are provided in Appendix H.

RESULT

Table 2 shows the results. Again, EDTS demonstrates consistent and significant improvement across all prediction horizons τ with an average improvement of 25.4% over the baselines. Our EDTS performs well on high-dimentional data with complex interactions.

Table 2: Results for semi-synthetic data for τ-step-ahead prediction based on real-world vital data (EEG). Shown: RMSE as mean ± standard deviation over five runs.

	$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$	$\tau = 6$
MSMs	0.54 (±0.00)	1.57 (±0.03)	2.48 (±0.06)	2.30 (±0.06)	2.07 (±0.05)	1.89 (±0.05)
RMSNs	0.47 (±0.01)	**0.91 (±0.01)**	**0.96 (±0.01)**	1.07 (±0.02)	1.17 (±0.04)	1.28 (±0.05)
CRN+	0.54 (±0.01)	1.13 (±0.02)	1.17 (±0.03)	1.22 (±0.03)	1.27 (±0.04)	1.31 (±0.05)
G-Net	0.57 (±0.01)	1.21 (±0.01)	1.18 (±0.01)	1.22 (±0.03)	1.28 (±0.05)	1.34 (±0.07)
CT+	0.51 (±0.01)	1.05 (±0.01)	1.12 (±0.03)	1.22 (±0.05)	1.34 (±0.07)	1.45 (±0.09)
EDCT+	0.58 (±0.01)	1.09 (±0.02)	1.10 (±0.03)	1.19 (±0.05)	1.28 (±0.07)	1.36 (±0.09)
EDTS (α=0)	0.48 (±0.00)	0.94 (±0.02)	0.97 (±0.02)	1.05 (±0.01)	**1.12 (±0.02)**	**1.19 (±0.03)**
EDTS	**0.41 (±0.03)**	**0.91 (±0.02)**	**0.96 (±0.03)**	**1.04 (±0.05)**	**1.12 (±0.07)**	**1.19 (±0.10)**

5.3. Ablation study

Subsequently, we performed an ablation study using full-synthetic data to examine the effectiveness of the treatment type and dosage discriminators.

a. The effectiveness of the treatment type discriminator: We removed the dosage discriminators ($\mathscr{D}_{\mathcal{D}}$) and trained the model with or without treatment type discriminator using data with only treatment type assignment bias ($\gamma_w = 5$, $\gamma_d = 0$).
b. The effectiveness of the dosage discriminators: (i) We removed the treatment type discriminator ($\mathscr{D}_{\mathcal{W}}$) and trained the model with dosage discriminators using data biased with only dosage assignment bias ($\gamma_w = 0$, $\gamma_d = 5$). (ii) We trained EDTS without discriminators, with only the treatment type discriminator, or with both treatment type and dosage discriminators using data with treatment type and dosage assignment bias ($\gamma_w = 4$, $\gamma_d = 4$).

Table 3 presents the results. Our treatment type and dosage discriminators improved over the baseline values. Therefore, their effectiveness has been verified.

Table 3: Ablation study of EDTS. Shown: Normalized RMSE as mean ± standard deviation over five runs.

		$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$	$\tau = 6$
a	$\gamma_w = 5, \gamma_d = 0$						
	None	1.51 (±0.24)	2.30 (±0.12)	2.60 (±0.13)	2.68 (±0.12)	2.64 (±0.14)	2.56 (±0.19)
	$\mathscr{D}_{\mathcal{W}}$	**1.37 (±0.17)**	**2.15 (±0.19)**	**2.47 (±0.19)**	**2.56 (±0.21)**	**2.57 (±0.23)**	**2.53 (±0.29)**
b	$\gamma_w = 0, \gamma_d = 5$						
	None	**0.44 (±0.04)**	**0.70 (±0.05)**	0.80 (±0.05)	0.82 (±0.05)	0.80 (±0.05)	0.76 (±0.05)
	$\mathscr{D}_{\mathcal{D}}$	0.45 (±0.03)	**0.70 (±0.07)**	**0.79 (±0.08)**	**0.81 (±0.08)**	**0.79 (±0.08)**	**0.75 (±0.08)**
	$\gamma_w = 4, \gamma_d = 4$						
	None	1.13 (±0.17)	1.72 (±0.27)	1.91 (±0.28)	1.95 (±0.28)	1.93 (±0.29)	1.91 (±0.36)
	$\mathscr{D}_{\mathcal{W}}$	1.06 (±0.19)	1.67 (±0.12)	1.96 (±0.18)	2.09 (±0.26)	2.13 (±0.35)	2.12 (±0.43)
	$\mathscr{D}_{\mathcal{W}}, \mathscr{D}_{\mathcal{D}}$	**0.90 (±0.10)**	**1.57 (±0.13)**	**1.82 (±0.18)**	**1.91 (±0.21)**	**1.90 (±0.22)**	**1.85 (±0.22)**

6. Limitations.

Training GANs by a minimax optimization becomes unstable if the assignment bias is too large ($\gamma_w = \gamma_d > 4$) or if the dose-response curves are complex (i.e., not monotonically increasing or

decreasing). To ensure stable training and reduce the complexity of EDTS, we imposed the following restrictions on the experimental settings:

a. We avoid using a large selection bias such as $\gamma_w = \gamma_d > 4$.
b. We simply make the dose-response curves monotone.
c. We only use two options for dosages for each treatment: half-dose and full-dose.

Although these approaches have improved the stability of the training, there is still room for more improvement.

7. Conclusion

We propose a starte-of-the-art method Encoder-Decoder Time-SCIGAN (EDTS) to estimate the counterfactual outcomes of time-varying treatment types and dosages. We also developed a GAN's discrimination method, Sequential Counterfactual Discrimination (SCD) and introduce two different weights: scale and factual weights into SCIGAN, confirming GAN's effectiveness for time-varying confounders. GANs generate counterfactual outcomes via adversarial training, while other methods MSMs, G-methods and balancing representaion do not directly consider them during training. Since GANs can directly train the prediction of counterfactual outcomes, GANs would be preferred over these methods. We believe that our proposed EDTS represents an important advancement in the setting of time-varying treatment types and dosages.

In future work, we aim to explore more challenging setting, including time-varying treatment types and continuous dosages instead of discrete ones. We also intend to develop more stable methods than GANs for addressing treatment assignment bias, potentially utilizing generative models like VAE and diffusion models as proposed by Wu et al. (2024).

References

Onur Atan, William R Zame, and Mihaela van der Schaar. Learning optimal policies from observational data. *arXiv preprint arXiv:1802.08679*, 2018.

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 (ICLR)*, 2020a.

Ioana Bica, James Jordon, and Mihaela van der Schaar. Estimating the effects of continuous-valued interventions using generative adversarial networks. In *Advances in neural information processing systems (NeurIPS)*, pages 16434–16445, 2020b.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Changran Geng, Harald Paganetti, and Clemens Grassberger. Prediction of treatment response for combined chemo-and radiation therapy for non-small cell lung cancer patients using a biomathematical model. *Scientific reports*, 7(1):13542, 2017.

Shantanu Ghosh, Christina Boucher, Jiang Bian, and Mattia Prosperi. Propensity score synthetic augmentation matching using generative adversarial networks (pssam-gan). *Computer methods and programs in biomedicine update*, 1:100020, 2021.

Shantanu Ghosh, Zheng Feng, Jiang Bian, Kevin Butler, and Mattia Prosperi. Dr-vidal-doubly robust variational information-theoretic deep adversarial learning for counterfactual prediction and treatment effect estimation on real world data. In *AMIA Annual Symposium Proceedings*, volume 2022, page 485, 2023.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

James Hensman, Nicolas Durrande, and Arno Solin. Variational fourier features for gaussian processes. *Journal of Machine Learning Research*, 18(151):1–52, 2018.

Miguel A Hernán, Babette Brumback, and James M Robins. Marginal structural models to estimate the joint causal effect of nonrandomized treatments. *Journal of the American Statistical Association*, 96(454):440–448, 2001.

Keisuke Hirano and Guido W Imbens. The propensity score with continuous treatments. *Applied Bayesian modeling and causal inference from incomplete-data perspectives*, 226164:73–84, 2004.

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

Kosuke Imai and David A Van Dyk. Causal inference with general treatment regimes: Generalizing the propensity score. *Journal of the American Statistical Association*, 99(467):854–866, 2004.

Guido W Imbens. The role of the propensity score in estimating dose-response functions. *Biometrika*, 87(3):706–710, 2000.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

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

Rui Li, Stephanie Hu, Mingyu Lu, Yuria Utsumi, Prithwish Chakraborty, Daby M Sow, Piyush Madan, Jun Li, Mohamed Ghalwash, Zach Shahn, et al. G-net: a recurrent network approach to g-computation for counterfactual prediction under a dynamic treatment regime. In *Machine Learning for Health*, pages 282–299. PMLR, 2021.

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 624–639, 2018.

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 7493–7503, 2018.

Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. *Advances in neural information processing systems*, 30, 2017.

Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Causal transformer for estimating counterfactual outcomes. In *International Conference on Machine Learning*, pages 15293–15329. PMLR, 2022.

Jerzy Neyman. On the application of probability theory to agricultural experiments. essay on principles. *Ann. Agricultural Sciences*, pages 1–51, 1923.

Lizhen Nie, Mao Ye, Qiang Liu, and Dan Nicolae. Vcnet and functional targeted regularization for learning causal effects of continuous treatments. *arXiv preprint arXiv:2103.07861*, 2021.

J Robins and MA Hernán. Estimation of the causal effects of time-varying exposures longitudinal data analysis, 2009.

James Robins. 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 (9-12):1393–1512, 1986.

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

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

Donald B Rubin. Bayesian inference for causal effects: The role of randomization. *The Annals of statistics*, pages 34–58, 1978.

Patrick Schwab, Lorenz Linhardt, Stefan Bauer, Joachim M Buhmann, and Walter Karlen. Learning counterfactual representations for estimating individual dose-response curves. In *AAAI Conference on Artificial Intelligence*, 2020.

Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE international conference on computer vision*, pages 4068–4076, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Brian G Vegetabile, Beth Ann Griffin, Donna L Coffman, Matthew Cefalu, Michael W Robbins, and Daniel F McCaffrey. Nonparametric estimation of population average dose-response curves using entropy balancing weights for continuous exposures. *Health Services and Outcomes Research Methodology*, 21:69–110, 2021.

Li Wang, Daijin Ren, Tianlun Huang, Xin Liu, and Gaosi Xu. The effectiveness and safety of full-dose versus half-dose corticosteroid plus renin-angiotensin system blockers for iga nephropathy. *Therapeutic Advances in Chronic Disease*, 10:2040622319887875, 2019.

Xin Wang, Shengfei Lyu, Xingyu Wu, Tianhao Wu, and Huanhuan Chen. Generalization bounds for estimating causal effects of continuous treatments. *Advances in Neural Information Processing Systems*, 35:8605–8617, 2022.

Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

Shenghao Wu, Wenbin Zhou, Minshuo Chen, and Shixiang Zhu. Counterfactual generative models for time-varying treatments. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3402–3413, 2024.

Yasin Yazıcı, Chuan-Sheng Foo, Stefan Winkler, Kim-Hui Yap, Georgios Piliouras, and Vijay Chandrasekhar. The unusual effectiveness of averaging in gan training. *arXiv preprint arXiv:1806.04498*, 2018.

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.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Xiao Lei Zhang, Henri Begleiter, Bernice Porjesz, Wenyu Wang, and Ann Litke. Event related potentials during object recognition tasks. *Brain research bulletin*, 38(6):531–538, 1995.

Appendix A. Assumptions

We build on the potential outcome framework (Rubin, 1974) and extend it to the time-varying treatments and outcomes (Robins and Hernán, 2009). The potential outcome framework has been widely used in earlier studies with objectives similar to (Robins and Hernán, 2009; Lim et al., 2018; Bica et al., 2020a). To achieve this, three standard assumptions for data generating mechanism are necessary to identify a counterfactual outcome distribution in the longitudinal setting. Specifically, we consider the average τ-step-ahead potential outcome conditioned on history from Eq. (3):

- Assumption A.1. (Consistency).

  If $(\mathbf{W}_{1:t}, \mathbf{D}_{1:t}) = (\mathbf{w}_{1:t}, \mathbf{d}_{1:t})$ is a given sequence of treatments for a patient, then $\mathbf{Y}_{t+1}(\mathbf{w}_{1:t}, \mathbf{d}_{1:t}) = \mathbf{Y}_{t+1}(\mathbf{W}_{1:t}, \mathbf{D}_{1:t})$. This implies that the potential outcome under the given treatment sequence $(\mathbf{w}_{1:t}, \mathbf{d}_{1:t})$ coincides with the observed (factual) outcome, with the condition $(\mathbf{W}_{1:t}, \mathbf{D}_{1:t}) = (\mathbf{w}_{1:t}, \mathbf{d}_{1:t})$.

- Assumption A.2. (Sequential Overlap).

  There is always a nonzero probability of receiving or not receiving any treatment across the entire history space over time: $0 < \mathbb{P}(\mathbf{W}_t = \mathbf{w}_t, \mathbf{D}_t = \mathbf{d}_t) \mid \mathbf{H}_{1:t} = \mathbf{h}_{1:t}) < 1$, if $\mathbb{P}(\mathbf{H}_{1:t} = \mathbf{h}_{1:t}) > 0$, where $\mathbf{h}_{1:t}$ is the realization of a patient history.

- Assumption A.3. (Sequential Ignorability) or No Unobserved Confounding.

  The current treatment is independent of the potential outcome, conditioning on the observed history: $(\mathbf{W}_t, \mathbf{D}_t) \perp\!\!\!\perp \mathbf{Y}_{t+1}(\mathbf{w}_t, \mathbf{d}_t) \mid \mathbf{H}_{1:t}, \forall(\mathbf{w}_t, \mathbf{d}_t)$. This implies that no unobserved confounders affect either treatment or outcome.

Appendix B. Details for EDCT-Base

EDCT-Base consists of a transformer-based encoder and decoder. The encoder builds a treatment invariant sequence of representations of the history $\mathbf{\Phi}_{1:t}$, balanced with a custom adversarial objective. The decoder subsequently uses $\mathbf{\Phi}_{1:t}$ as cross-attention keys and values for estimating outcomes of future treatments.

We start by mapping the concatenated past outcomes $\mathbf{y}_t$, time-varying covariates $\mathbf{X}_t$, static covariates $\mathbf{V}$, past treatment $\mathbf{W}_{t-1}$, and past treatment dosages $\mathbf{D}_{t-1}$ to a hidden state space of dimensionality d_h via a fully-connected linear layer as follows:

$$\mathbf{h}_t^0 = \text{Linear}(\text{Concat}(\mathbf{W}_{t-1}, \mathbf{D}_{t-1}, \mathbf{y}_t, \mathbf{X}_t, \mathbf{V})) \tag{14}$$

In the case of the decoder, we apply a similar input transformation

$$\mathbf{h}_t^0 = \text{Linear}(\text{Concat}(\mathbf{w}_{t-1}, \mathbf{d}_{t-1}, \hat{\mathbf{y}}_t, \mathbf{V})) \tag{15}$$

where $\hat{\mathbf{y}}_t$ are autoregressively-fed model outputs.

We then stack B identical encoder/decoder blocks or layers, which transform the whole sequence of hidden states $(\mathbf{h}_1^0, \ldots, \mathbf{h}_t^0)$ in quadratic time, depending on sequence length l. This is given by as follows,

$$\mathrm{H}^b = (\mathbf{h}_1^b, \ldots, \mathbf{h}_t^b)^\top \in \mathbb{R}^{l \times d_h}, \tag{16}$$

$$\mathrm{H}^b = \text{Block}_b(\mathrm{H}^{b-1}), \quad b \in \{1, \ldots, B\}, \tag{17}$$

where B is the total number of blocks.

The encoder uses hidden states to infer keys K, queries Q, and values V (self-attention). In contrast, the decoder has both self- and cross-attention. Subsequently, we use keys and values, inferred from a sequence of balanced representations of history.

B.1. Encoder block

The encoder block is defined in the following way:

$$\tilde{\mathrm{H}}^{b-1} = \text{LN}\left(\text{MHA}\left(Q(\mathrm{H}^{b-1}), K(\mathrm{H}^{b-1}), V(\mathrm{H}^{b-1})\right) + \mathrm{H}^{b-1}\right), \tag{18}$$

$$\mathrm{H}^b = \text{LN}\left(\text{FF}(\tilde{\mathrm{H}}^{b-1}) + \tilde{\mathrm{H}}^{b-1}\right), \tag{19}$$

where LN is the layer norm, MHA is the multi-head attention, and FF is the feed-forward layer.

We take the representations H^b from the last transformer block as the outputs. Here, we apply an additional fully connected linear layer and exponential linear unit (ELU) nonlinearity, as follows,

$$\mathbf{\Phi}_t = \text{ELU}(\text{Linear}(\mathrm{H}^b)) \tag{20}$$

where fully-connected linear layer is followed by dropout, $\mathbf{\Phi}_t \in \mathbb{R}^{d_r}$ and d_r is the dimensionality of the representation.

B.2. Decoder block

The decoder block adds a cross-attention layer after self-attention (that is between Eq. (18) and Eq. (19)). Note that the dimensionality of the hidden decoder state is set such that it matches the size of the balanced representations of the encoder, i.e., $d_h = d_r$. This is formalized as follows:

$$\tilde{\mathrm{H}}^{b-1} = \mathrm{LN}\left(\mathrm{MHA}\left(Q(\mathrm{H}^{b-1}), K(\mathrm{H}^{b-1}), V(\mathrm{H}^{b-1})\right) + \mathrm{H}^{b-1}\right), \tag{21}$$

$$\tilde{\tilde{\mathrm{H}}}^{b-1} = \mathrm{LN}\left(\mathrm{MHA}\left(Q(\tilde{\mathrm{H}}^{b-1}), K((\boldsymbol{\Phi}_{1:t})^\top), V((\boldsymbol{\Phi}_{1:t})^\top)\right) + \tilde{\mathrm{H}}^{b-1}\right), \tag{22}$$

$$\mathrm{H}^{b} = \mathrm{LN}\left(\mathrm{FF}(\tilde{\tilde{\mathrm{H}}}^{b-1}) + \tilde{\tilde{\mathrm{H}}}^{b-1}\right), \tag{23}$$

where $\boldsymbol{\Phi}_{1:t}$ is a sequence of the encoder representations (i.e., the encoded history $\mathbf{H}_{1:t}$), as transformed according to Eq. (16).

We calculate $\boldsymbol{\Phi}_{t+\tau}$ in the same manner of the encoder block using Eq. (20).

Appendix C. Details for Time-SCIGAN

C.1. Counterfactual Generator

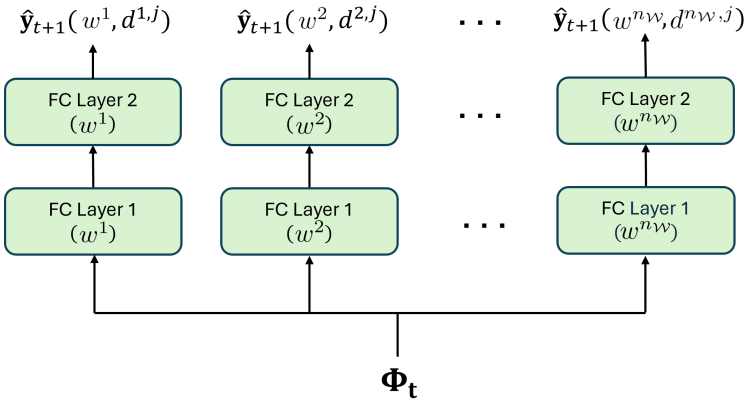


Figure 4: Architecture of Generator $\mathcal{G}$

The counterfactual generator $\mathcal{G}$ is a multi-head model of fully-connected layers for each treatment. We concatenate representations $\mathbf{\Phi}_t$, dosage $d^{k,j}$ and some noises Z (a uniform distribution in the interval [0, 1]) as input, and transform them to outcomes $\hat{\mathbf{y}}_{t+1}(w^k, d^{k,j})$ via a double fully-connected linear layer:

$$\hat{\mathbf{y}}_{t+1}(w^k, d^{k,j}) = \text{Linear}(\text{ELU}(\text{Linear}(\text{Concat}(\mathbf{\Phi}_t, \text{Z}, d^{k,j})))) \tag{24}$$

We calculate the outcomes for all treatment-dosage pairs $\{\{(w^k, d^{k,j})\}_{j=1}^{n_\mathcal{D}^k}\}_{k=1}^{n_\mathcal{W}}$, and obtain the dose-response curve $\hat{\mathbf{y}}_{t+1}(w^k) = \{\hat{\mathbf{y}}_{t+1}(w^k, d^{k,j})\}_{j=1}^{n_\mathcal{D}^k}$ for each treatment w^k. The factual outcome corresponds to factual treatment and dosage, the factual dose-response curve corresponds to factual treatment, and the other outcomes and dose-response curves are counterfactual (Eq.25, Eq.26).

$$\hat{\mathbf{y}}_{t+1}(w, d) = \begin{cases} \hat{\mathbf{y}}^f_{t+1}(w, d) \; if \; w \; = \mathbf{w}_t \; and \; d = \mathbf{d}_t \\ \hat{\mathbf{y}}^{cf}_{t+1}(w, d) \; else \end{cases} \tag{25}$$

$$\hat{\mathbf{y}}_{t+1}(w) = \begin{cases} \hat{\mathbf{y}}^f_{t+1}(w) \; if \; w \; = \mathbf{w}_t \\ \hat{\mathbf{y}}^{cf}_{t+1}(w) \; else \end{cases} \tag{26}$$

C.2. Discriminators

To ensure that the discriminators act as functions of sets, SCIGAN use ideas from Zaheer et al. (2017) to create permutation-invariant for the treatment discriminator and permutation-equivariant networks for the dosage discriminators.

C.2.1. Treatment Type Discriminator

We want the treatment discriminator $\mathcal{D}_\mathcal{W}$, to be permutation-invariant with respect to $\hat{\mathbf{y}}_{t+1}(w^k)$ for each treatment w^k. To achieve this, we define $h_t : \sum_{k=1}^{n_\mathcal{W}} (\mathcal{D}^k \times \mathcal{Y})^{n_\mathcal{D}^k} \rightarrow \mathcal{H}_H$ and require that h_t be permutation invariant w.r.t. each of the spaces $(\mathcal{D}^k \times \mathcal{Y})^{n_\mathcal{D}^k}$. To construct h_t, we concatenate

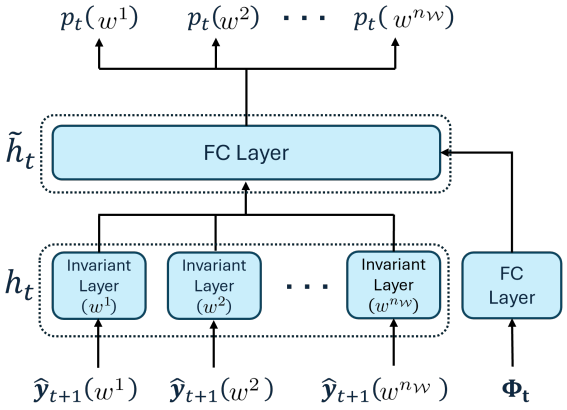


Figure 5: Treatment discriminator $\mathscr{D}_{\mathcal{W}}$

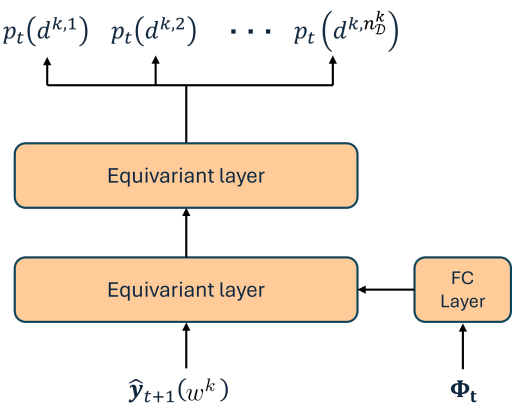


Figure 6: Dosage Discriminator $\mathscr{D}_{\mathcal{D}}^k$

Figure 7: Architecture of discriminators

the outputs of several invariant layers, and each individual acts in space $(\mathcal{D}^k \times \mathcal{Y})^{n_{\mathcal{D}}^k}$. That is, for each treatment w^k, we define a map $h_t^k : (\mathcal{D}^k \times \mathcal{Y})^{n_{\mathcal{D}}^k} \to \mathcal{H}_h^k$. We then define $\mathcal{H}_H = \sum_{k=1}^{n_W} \mathcal{H}_H^k$ and $h_t(\hat{\mathbf{y}}_{t+1}) = (h_t^1(\hat{\mathbf{y}}_{t+1}(w^1)), \ldots, h_t^{n_W}(\hat{\mathbf{y}}_{t+1}(w^{n_W})))$.

We begin by mapping the concatenated $\hat{\mathbf{y}}_{t+1}(w^k, d^{k,j})$ and dosages $d^{k,j}$ to the hidden state space of dimensionality d_{inv} via a fully-connected linear layer:

$$h_t^{k,j} = \text{ELU}\left(\text{Linear}\left(\text{Concat}\left(\hat{\mathbf{y}}_{t+1}(w^k, d^{k,j}), d^{k,j}\right)\right)\right) \tag{27}$$

We summarize the hidden value $h_t^{k,j}$ for each treatment w^k,

$$h_t^k = \sum_{j=1}^{n_{\mathcal{D}}^k} h_t^{k,j} \tag{28}$$

We transformed $\mathbf{\Phi}_t$ to a hidden state space of dimensionality d_{inv} via a fully connected linear layer, as shown in Eq. (29). Then, we concatenate h_t^k and $\tilde{\mathbf{\Phi}}_t$, and apply an additional fully-connected linear and ELU layer as shown in Eq. (30).

$$\tilde{\mathbf{\Phi}}_t = \text{ELU}(\text{Linear}(\mathbf{\Phi}_t)) \tag{29}$$

$$\tilde{h}_t = \text{ELU}(\text{Linear}(\text{Concat}(h_t^1, h_t^2, \ldots, h_t^{n_W}, \tilde{\mathbf{\Phi}}_t))) \tag{30}$$

We finally pass $\tilde{h}_t$ through a fully connected network $\tilde{h}_t \to [0, 1]^k$.

$$P_t(\mathcal{W}) = \text{Linear}(\tilde{h}_t) \tag{31}$$

C.2.2. DOSAGE DISCRIMINATOR

We want each dosage discriminator, $\mathscr{D}_{\mathcal{D}}^k$, to be permutation equivariant with respect to $\hat{\mathbf{y}}_{t+1}(w^{k,j}, d^{k,j}) \in \hat{\mathbf{y}}_{t+1}(w^k)$ for each dosage $d^{k,j}$. The basic building block for equivariant functions is defined in terms of equivariance input, $\mathbf{u}$, and auxiliary input, $\mathbf{v}$, by:

$$f_{equi}(\mathbf{u}, \mathbf{v}) = \sigma(\lambda \mathbf{I}_m \mathbf{u} + \gamma(\mathbf{1}_m \mathbf{1}_m^T)\mathbf{u} + (\mathbf{1}_m \Theta^T)\mathbf{v}), \tag{32}$$

where $\mathbf{I}_m$ is the $m \times m$ identity matrix, λ and γ are scalar parameters and Θ is a vector of weights.

To achieve this, each $\mathscr{D}_{\mathcal{D}}^k$ consists of two layers, as given in Eq. 32 (as shown in Figure 6).

Equivariant Layer 1 receives $\hat{\mathbf{y}}_{t+1}(w^k)$ for the equivariance input and $\mathbf{\Phi}_t$ for the auxiliary input.

$$I_t^{k,j} = \text{Concat}(\hat{\mathbf{y}}_{t+1}(w^k, d^{k,j}), d^{k,j}) \tag{33}$$

$$\lambda_t^k = \text{Linear}(\sum_{j=1}^{n_{\mathcal{D}}^k} I_t^{k,j}) \qquad \gamma_t^{k,j} = \text{Linear}(I_t^{k,j}) \qquad \tilde{\mathbf{\Phi}}_t = \text{ELU}(\text{Linear}(\mathbf{\Phi}_t)) \tag{34}$$

$$h_t^{k,j} = \text{ELU}(-\lambda_t^k + \gamma_t^{k,j} + \tilde{\mathbf{\Phi}}_t) \tag{35}$$

Equivariant Layer 2 receives the output of the first layer $h_t^{k,j}$ for the equivariance input.

$$\tilde{\lambda}_t^k = \text{Linear}(\sum_{j=1}^{n_{\mathcal{D}}^k} h_t^{k,j}) \qquad \tilde{\gamma}_t^{k,j} = \text{Linear}(h_t^{k,j}) \tag{36}$$

$$\tilde{h}_t^{k,j} = \text{ELU}(-\tilde{\lambda}_t^k + \tilde{\gamma}_t^{k,j}) \tag{37}$$

We finally pass the output of the second layer through a fully connected network $\tilde{h}_t^{k,j} \to [0, 1]$.

$$P_t(d^{k,j}) = \text{Linear}(\tilde{h}_t^{k,j}) \tag{38}$$

Appendix D. Proof of theorem

We extend the theorem of Bica et al. (2020a) for time-varying treatment types to our theorem for time-varying treatment types and dosages.

We first prove the following proposition.

Proposition 1. For fixed Φ, let $x' = \Phi(\mathbf{H}_t)$. Then the optimal prediction probabilities of $\mathscr{D}_{\mathcal{W}}$ and $\mathscr{D}_{\mathcal{D}}^{k}$ are given as follows:

$$\mathscr{D}_{\mathcal{W}}^{k*}(x') = \frac{P_k^{\Phi}(x')}{\sum_{k=1}^{n_{\mathcal{W}}} P_k^{\Phi}(x')} \qquad \mathscr{D}_{\mathcal{D}}^{k,j*}(x') = \frac{P_{k,j}^{\Phi}(x')}{\sum_{j=1}^{n_{\mathcal{D}}^k} P_{k,j}^{\Phi}(x')} \tag{39}$$

Proof: For fixed Φ, the optimal prediction probabilities are given as follows:

$$\mathscr{D}_{\mathcal{W}}^{*} = \max_{\mathscr{D}_{\mathcal{W}}} \sum_{k=1}^{n_{\mathcal{W}}} \int_{x'} \log\left(\mathscr{D}_{\mathcal{W}}^{k}(x')\right) P_k^{\Phi}(x') dx' \quad subject\ to \sum_{k=1}^{n_{\mathcal{W}}} \mathscr{D}_{\mathcal{W}}^{k}(x') = 1 \tag{40}$$

$$\mathscr{D}_{\mathcal{D}}^{k*} = \max_{\mathscr{D}_{\mathcal{D}}} \sum_{j=1}^{n_{\mathcal{D}}^k} \int_{x'} \log\left(\mathscr{D}_{\mathcal{D}}^{k,j}(x')\right) P_{k,j}^{\Phi}(x') dx' \quad subject\ to \sum_{j=1}^{n_{\mathcal{D}}^k} \mathscr{D}_{\mathcal{D}}^{k,j}(x') = 1 \tag{41}$$

By maximizing the value function pointwise and applying Lagrange multiples, we get

$$\mathscr{D}_{\mathcal{W}}^{*} = \max_{\mathscr{D}_{\mathcal{W}}} \sum_{k=1}^{n_{\mathcal{W}}} \log\left(\mathscr{D}_{\mathcal{W}}^{k}(x')\right) P_k^{\Phi}(x') + \lambda_w \left(\sum_{k=1}^{n_{\mathcal{W}}} \mathscr{D}_{\mathcal{W}}^{k}(x') - 1\right) \tag{42}$$

$$\mathscr{D}_{\mathcal{D}}^{k*} = \max_{\mathscr{D}_{\mathcal{D}}^{k}} \sum_{j=1}^{n_{\mathcal{D}}^k} \log\left(\mathscr{D}_{\mathcal{D}}^{k,j}(x')\right) P_{k,j}^{\Phi}(x') + \lambda_d^k \left(\sum_{j=1}^{n_{\mathcal{D}}^k} \mathscr{D}_{\mathcal{D}}^{k,j}(x') - 1\right) \tag{43}$$

Setting the derivative (w.r.t. $\mathscr{D}_{\mathcal{W}}^{k*}(x')$ and $\mathscr{D}_{\mathcal{D}}^{k,j*}(x')$) to zero and solving for $\mathscr{D}_{\mathcal{W}}^{k*}(x')$ and $\mathscr{D}_{\mathcal{D}}^{k,j*}(x')$, we get

$$\mathscr{D}_{\mathcal{W}}^{k*}(x') = -\frac{P_k^{\Phi}(x')}{\lambda_w} \qquad \mathscr{D}_{\mathcal{D}}^{k,j*}(x') = -\frac{P_{k,j}^{\Phi}(x')}{\lambda_d^k} \tag{44}$$

where λ_w and λ_d^k can now be solved for using the constraint to be $\lambda_w = -\sum_{k=1}^{n_{\mathcal{W}}} P_k^{\Phi}(x')$ and $\lambda_d^k = -\sum_{j=1}^{n_{\mathcal{D}}^k} P_{k,j}^{\Phi}(x')$. This gives the result.

Proof. (**Theorem 1**): by substituting the expression from Proposition 1 into the minimax game defined in Eq. 6, the objective for Φ becomes

$$\min_{\Phi} \sum_{k=1}^{n_{\mathcal{W}}} \left(\mathbb{E}_{x' \sim P_k^{\Phi}} \left[\log\left(\frac{P_k^{\Phi}(x')}{\sum_{k'=1}^{n_{\mathcal{W}}} P_{k'}^{\Phi}(x')} \right) \right] + \sum_{j=1}^{n_{\mathcal{D}}^k} \mathbb{E}_{x' \sim P_{k,j}^{\Phi}} \left[\log\left(\frac{P_{k,j}^{\Phi}(x')}{\sum_{j'=1}^{n_{\mathcal{D}}^k} P_{k,j'}^{\Phi}(x')} \right) \right] \right) \tag{45}$$

We then note that

$$\sum_{k=1}^{n_{\mathcal{W}}} \mathbb{E}_{x' \sim P_k^{\Phi}} \left[\log \left(\frac{P_k^{\Phi}(x')}{\sum_{k'=1}^{n_{\mathcal{W}}} P_{k'}^{\Phi}(x')} \right) \right] + n_{\mathcal{W}} \log n_{\mathcal{W}}$$
$$+ \sum_{k=1}^{n_{\mathcal{W}}} \left(\sum_{j=1}^{n_{\mathcal{D}}^k} \mathbb{E}_{x' \sim \mathrm{P}_{k,j}^{\Phi}} \left[\log \left(\frac{P_{k,j}^{\Phi}(x')}{\sum_{j'=1}^{n_{\mathcal{D}}^k} P_{k,j'}^{\Phi}(x')} \right) \right] + n_{\mathcal{D}}^k \log n_{\mathcal{D}}^k \right) \tag{46}$$

$$= \sum_{k=1}^{n_{\mathcal{W}}} \mathbb{E}_{x' \sim P_k^{\Phi}} \left[\log \left(\frac{P_k^{\Phi}(x')}{\frac{1}{n_{\mathcal{W}}} \sum_{k'=1}^{n_{\mathcal{W}}} P_{k'}^{\Phi}(x')} \right) \right] + \sum_{k=1}^{n_{\mathcal{W}}} \left(\sum_{j=1}^{n_{\mathcal{D}}^k} \mathbb{E}_{x' \sim P_{k,j}^{\Phi}} \left[\log \left(\frac{P_{k,j}^{\Phi}(x')}{\frac{1}{n_{\mathcal{D}}^k} \sum_{j'=1}^{n_{\mathcal{D}}^k} P_{k,j'}^{\Phi}(x')} \right) \right] \right) \tag{47}$$

$$= \sum_{k=1}^{n_{\mathcal{W}}} KL \left(P_k^{\Phi}(x') \| \frac{1}{n_{\mathcal{W}}} \sum_{k'=1}^{n_{\mathcal{W}}} P_{k'}^{\Phi}(x') \right) + \sum_{k=1}^{n_{\mathcal{W}}} \left(\sum_{j=1}^{n_{\mathcal{D}}^k} KL \left(P_{k,j}^{\Phi}(x') \| \frac{1}{n_{\mathcal{D}}^k} \sum_{j'=1}^{n_{\mathcal{D}}^k} P_{k,j'}^{\Phi}(x') \right) \right) \tag{48}$$

$$= K \cdot JSD(P_1^{\Phi}, \ldots, P_{n_{\mathcal{W}}}^{\Phi}) + \sum_{k=1}^{n_{\mathcal{W}}} K \cdot JSD(P_{k,1}^{\Phi}, \ldots, P_{k,n_{\mathcal{D}}^k}^{\Phi}) \tag{49}$$

where $KL(\cdot\|\cdot)$ is the Kullback-Leibler divergence and $JSD(\cdot, \ldots, \cdot)$ is the multi-distribution Jensen-Shannon Divergence (Li et al., 2018). Because $n_{\mathcal{W}} \log n_{\mathcal{W}}$ and $n_{\mathcal{D}}^k \log n_{\mathcal{D}}^k$ are constants and the multi-distribution JSD is non-negative and 0 if and only if all distributions are equal, we have that $P_1^{\Phi} = P_2^{\Phi} = \cdots = P_{n_{\mathcal{W}}}^{\Phi}$ and $P_{k,1}^{\Phi} = P_{k,2}^{\Phi} = \cdots = P_{k,n_{\mathcal{D}}^k}^{\Phi}$ for any treatment w^k.

Appendix E. Baseline methods

We select six methods as baselines, which make use of the previous setting of time-varying treatment types. These methods are as follows: (1) Marginal Structural Models (MSMs) (Robins et al., 2000; Hernán et al., 2001), (2) Recurrent Marginal Structural Networks (RMSNs) (Lim et al., 2018), (3) Counterfactual Recurrent Network (CRN) (Bica et al., 2020a), (4) G-Net (Li et al., 2021), (5) Causal Transformer (CT) and, (6) Encoder-Decoder Causal Transformer (EDCT) (Melnychuk et al., 2022).

a. For the prior setting of time-varying treatment types, we follow the same approach used in Melnychuk et al. (2022) for the baseline methods.

b. For our new settings of time-varying treatment types and dosages, we modify the baseline methods to include dosages in their inputs. Additionally, we extend CRN, CT, and EDCT to add a dosage classifier network to address dosage assignment bias. These modified baseline methods are denoted as CRN+, CT+, and EDCT+, respectively.

We provide details for each baseline and the modification for the new setting in the following sections.

E.1. Marginal Structural Models (MSMs)

MSMs (Robins et al., 2000; Hernán et al., 2001) is a standard baseline in epidemiology, which aims to estimate counterfactual outcomes using inverse probability of treatment weights (IPTW). Time-varying confounding bias is addressed by applying stabilized weights,

$$SW(t,\tau) = \frac{\prod_{n=t}^{t+\tau} f(\mathbf{W}_n|\mathbf{W}_{1:n-1})}{\prod_{n=t}^{t+\tau} f(\mathbf{W}_n|\mathbf{H}_{1:n})}. \tag{50}$$

Here, τ ranges from 1 to τ_{max}, and $f(\mathbf{W}_n|\mathbf{W}_{1:n-1})$ and $f(\mathbf{W}_n|\mathbf{H}_{1:n})$ denote the conditional probability mass functions for discrete treatments of $\mathbf{W}_n$ given $\mathbf{W}_{1:n-1}$ and $\mathbf{H}_{1:n}$, respectively.

Both are estimated using logistic regressions, which depend on the sum of previous treatment type applications, two previous time-varying covariates and static covariates, as follows,

$$f(\mathbf{W}_t|\mathbf{W}_{t-1}) = \sigma\left(\sum_{k=1}^{n_{\mathcal{W}}} \omega_k \mathbb{1}_{[\mathbf{W}_{t-1}=w^k]}\right) \tag{51}$$

$$f(\mathbf{W}_t|\mathbf{H}_{t-1:t}) = \sigma\left(a_{1,x}\mathbf{X}_t + a_{2,x}\mathbf{X}_{t-1} + a_{1,y}\mathbf{y}_t + a_{2,y}\mathbf{y}_{t-1} + a_v\mathbf{V} + \sum_{k=1}^{n_{\mathcal{W}}} \phi_k \mathbb{1}_{[\mathbf{W}_{t-1}=w^k]}\right) \tag{52}$$

where $\sigma(\cdot)$ is a sigmoid function and ω, ϕ, a are logistic regression parameters. After the stabilized weights are estimated, they are normalized and truncated at their 1st and 99th percentiles as carried out by Lim et al. (2018).

Counterfactual outcome regressions are defined, as follows,

$$f(\mathbf{y}_{t+1}|\mathbf{H}_{t-1:t}, \mathbf{W}_t) = b_{1,x}\mathbf{X}_t + b_{2,x}\mathbf{X}_{t-1} + b_{1,y}\mathbf{y}_t + b_{2,y}\mathbf{y}_{t-1} + b_v\mathbf{V} + \sum_{k=1}^{n_{\mathcal{W}}} \beta_k \sum_{n=t-1}^{t} \mathbb{1}_{[\mathbf{W}_n=w^k]} \tag{53}$$

where β and b are regression coefficients. These outcome regressions are fitted separately for each prediction horizon. We consider six projection horizons and create a separate model for each one. Each model is fitted independently, and we do not adjust the weight based on the projection horizon.

FOR THE SETTING OF TIME-VARYING TREATMENT TYPES AND DOSAGES

We add the previous dosage $\mathbf{D}_{t-1}$ to $f(\mathbf{W}_t|\mathbf{H}_t)$ (Eq. (54)), and previous dosage $\mathbf{D}_{t-1}$ and current dosage $\mathbf{D}_t$ to $f(\mathbf{y}_{t+1}|\mathbf{H}_t)$ (Eq. (55)) as inputs.

$$f(\mathbf{W}_t|\mathbf{H}_{t-1:t}) = \sigma\Big(a_{1,x}\mathbf{X}_t + a_{2,x}\mathbf{X}_{t-1} + a_{1,y}\mathbf{y}_t + a_{2,y}\mathbf{y}_{t-1} + a_v\mathbf{V} + \sum_{k=1}^{n_W} \phi_k \mathbb{1}_{[\mathbf{W}_{t-1}=w^k]} + a_{2,d}\mathbf{D}_{t-1}\Big) \tag{54}$$

$$f(\mathbf{y}_{t+1}|\mathbf{H}_{t-1:t}, \mathbf{W}_t, \mathbf{D}_t) = b_{1,x}\mathbf{X}_t + b_{2,x}\mathbf{X}_{t-1} + b_{1,y}\mathbf{y}_t + b_{2,y}\mathbf{y}_{t-1} + b_v\mathbf{V} + \sum_{k=1}^{n_W} \beta_k \sum_{n=t-1}^{t} \mathbb{1}_{[\mathbf{W}_n=w^k]} + b_{1,d}\mathbf{D}_t + b_{2,d}\mathbf{D}_{t-1} \tag{55}$$

E.2. Recurrent Marginal Structural Networks (RMSNs)

RMSNs refer to sequence-to-sequence architectures consisting of four LSTM subnetworks: a propensity treatment network, a propensity history network, an encoder, and a decoder. RMSNs are designed to handle multiple binary treatments. To perform one-step-ahead prdiction, the encoder first learns a representation $\mathbf{\Phi}_t$ of the observed history $\mathbf{H}_{1:t}$. To perform τ-step-ahead prediction, the decoder uses this representation to compute $f(\mathbf{y}_{t+\tau}|\mathbf{\Phi}_t, \mathbf{w}_{t+1:t+\tau-1})$. A fully-connected linear layer (memory adapter) is used to match the size of the representation of the encoder with the hidden units of the decoder.

In RMSNs, time-varying confounding is addressed by reweighting the objective using IPTW (Robins et al., 2000) during training. IPTW creates a pseudo-population that mimics a randomized controlled trial. As demonstrated in Lim et al. (2018), we use the stabilized weights (Eq. (50)). Both $f(\mathbf{W}_t|\mathbf{W}_{1:t-1})$ and $f(\mathbf{W}_t|\mathbf{H}_{1:t})$ are learned from the data using LSTM networks, which are called propensity treatment network (nominator) and propensity history network (denominator).

During training, the propensity networks are initially trained to estimate the stabilized weights $SW(t, \tau)$. The encoder is trained using a mean squared error (MSE) weighted by $SW(\cdot, 1)$. Similar to MSMs, the stabilized weights are subsequently normalized and truncated. Finally, the decoder is trained by minimizing the loss using the the stabilized weights $SW(\cdot, \tau_{max})$.

FOR THE SETTING OF TIME-VARYING TREATMENT TYPES AND DOSAGES

We add past dosage $\mathbf{D}_{1:t-1}$ to the observed history $\mathbf{H}_{1:t}$, and future dosage $\mathbf{d}_{t:t+\tau-1}$ to the prediction $\mathbf{y}_{t+\tau}[\mathbf{w}_{t:t+\tau-1}, \mathbf{d}_{t:t+\tau-1}]$ for τ-step-ahead prediction.

E.3. Counterfactual Recurrent Network (CRN)

CRN consists of an encoder-decoder architecture. The encoder and decoder each consist of a single LSTM-layer (G_R). At each time step t, the output of G_R is passed through a fully-connected linear layer to build a representation $\mathbf{\Phi}_t$. Then, two fully-connected networks G_Y and G_W (as G_A in original paper (Bica et al., 2020a)) are on top of $\mathbf{\Phi}_t$, to predict the next outcome $\mathbf{y}_{t+1}$ and the current treatment type $\mathbf{w}_t$, respectively. CRN builds balanced representations that predict outcomes, and

do not predict the treatment type assignment. This is achieved by adopting an adversarial learning technique, known as gradient reversal (Ganin et al., 2016). Both the encoder and decoder is trained by minimizing the following loss,

$$\mathcal{L}(G_Y, G_{\mathcal{W}}, G_R) = |\mathbf{y}_{t+1} - G_Y(\mathbf{\Phi}_t, \mathbf{w}_t)|^2 - \lambda \sum_{k=1}^{n_{\mathcal{W}}} \mathbb{1}_{[w^k = \mathbf{w}_t]} \log G_W(\mathbf{\Phi}_t) \tag{56}$$

with hyperparameter λ.

Loss $\mathcal{L}$ is based on a gradient reversal layer (Ganin et al., 2016), which forces G_W to minimize the cross-entropy between the predicted and current treatments, but $\mathbf{\Phi}_t$ to maximize it. In our experiments, we kept $\lambda = 1$, similar to Atan et al. (2018); Bica et al. (2020a); Melnychuk et al. (2022).

E.4. G-Net

G-Net is based on the G-computation formula in Eq. (57), which expresses the average counterfactual outcome $\mathbf{y}_{t+\tau}[\mathbf{w}_{t:t+\tau-1}]$ conditioned on the history $\mathbf{H}_{1:t}$ in terms of the observational data distribution.

$$\mathbb{E}(\mathbf{y}_{t+\tau}[\mathbf{w}_{t:t+\tau-1}] \mid \mathbf{H}_{1:t}) = \int_{\mathbb{R}^{d_x} \times \cdots \times \mathbb{R}^{d_x}} \mathbb{E}(\mathbf{y}_{t+\tau} | \mathbf{H}_{1:t}, \mathbf{y}_{t+1:t+\tau-1}, \mathbf{w}_{t+1:t+\tau-1}) \times$$
$$\prod_{j=t+1}^{t+\tau-1} \mathbb{P}(\mathbf{y}_j | \mathbf{H}_{1:t}, \mathbf{y}_{t+1:j-1}, \mathbf{w}_{t:j-1}) \, \mathrm{d}\mathbf{y}_{t+1:t+\tau-1} \tag{57}$$

G-Net performs counterfactual outcome prediction in two steps. First, the conditional distributions $\mathbb{P}(\mathbf{y}_j | \mathbf{H}_{1:t}, \mathbf{y}_{t+1:j-1}, \mathbf{w}_{t:j-1})$ are estimated. Subsequently, Monte Carlo simulations are performed using Eq. (57), by sampling from the estimated distributions. Later, $\mathbf{y}_{t+\tau}[\mathbf{w}_{t:t+\tau-1}]$ is predicted by calculating the empirical mean over the Monte Carlo samples (M = 50).

The conditional distributions $\mathbb{P}(\mathbf{y}_j | \mathbf{H}_{1:t}, \mathbf{x}_{t+1:j-1}, \mathbf{w}_{t:j-1})$ are learned by estimating the respective conditional expectations $\mathbb{E}(\mathbf{y}_j | \mathbf{H}_{1:t}, \mathbf{y}_{t+1:j-1}, \mathbf{w}_{t:j-1})$, which are in turn learned via a single LSTM jointly with outcome prediction. We can then sample from $\mathbb{P}(\mathbf{y}_j | \mathbf{H}_{1:t}, \mathbf{y}_{t+1:j-1}, \mathbf{w}_{t:j-1})$ by drawing from the empirical distributions of the residuals on a holdout set that is not used to estimate the conditional expectations. We used 10% of the training data as the holdout dataset.

FOR THE SETTING OF TIME-VARYING TREATMENT TYPES AND DOSAGES

We add past dosage $\mathbf{D}_{1:t-1}$ to the observed history $\mathbf{H}_{1:t}$, and future dosage $\mathbf{d}_{t:t+\tau-1}$ to the prediction $\mathbf{y}_{t+\tau}[\mathbf{w}_{t,t+\tau-1}, \mathbf{d}_{t:t+\tau-1}]$ for τ-step-ahead prediction.

E.5. Causal Transformer (CT)

The state-of-the-art study (Melnychuk et al., 2022) proposes two transformer models, CT and EDCT. CT is a single multi-input architecture that combines separate transformer subnetworks for each feature, whereas EDCT consists of a transformer-based encoder and decoder, with all input sequences are fed into a single subnetwork.

CT and EDCT fed outputs of transformers G_R into a fully-connected linear layer that builds a representation $\mathbf{\Phi}_t$. Two fully-connected networks G_Y and $G_{\mathcal{W}}$ (as G_A in Melnychuk et al. (2022)) are placed on top of $\mathbf{\Phi}_t$ to predict the next outcome $\mathbf{y}_{t+1}$ and the current treatment $\mathbf{w}_t$, respectively.

$G_{\mathcal{W}}$ is fit to predict the current treatment, as follows,

$$\mathcal{L}_{\mathcal{W}}(G_{\mathcal{W}}, G_R) = -\sum_{k=1}^{n_{\mathcal{W}}} \mathbb{1}_{[w^k=\mathbf{w}_t]} \log G_{\mathcal{W}}(\mathbf{\Phi}_t). \tag{58}$$

This minimizes the classification loss of the current treatment type assignment given by $\mathbf{\Phi}_t$. However, while $G_{\mathcal{W}}$ can predict the current treatment, the actual representation $\mathbf{\Phi}_t$ should be non-predictive to address treatment type assignment bias.

To handle adversarial objectives, CT and EDCT use an adversarial learning technique called the counterfactual domain confusion (CDC) loss (Melnychuk et al., 2022) to minimize the cross-entropy between a uniform distribution over the treatment categorical space and the predictions of $G_{\mathcal{W}}$, as follows,

$$\mathcal{L}_{conf}(G_{\mathcal{W}}, G_R) = -\sum_{k=1}^{n_{\mathcal{W}}} \frac{1}{n_{\mathcal{W}}} \log G_{\mathcal{W}}(\mathbf{\Phi}_t). \tag{59}$$

Overall, $\hat{G}_Y$, $\hat{G}_R$ and $\hat{G}_{\mathcal{W}}$ are trained according to the minimax game defined as follows,

$$(\hat{G}_Y, \hat{G}_R) = \underset{G_Y, G_R}{\arg\min}\, \mathcal{L}_Y(G_Y, G_R) + \alpha \mathcal{L}_{conf}(\hat{G}_{\mathcal{W}}, G_R) \tag{60}$$

$$\hat{G}_{\mathcal{W}} = \underset{G_W}{\arg\min}\, \alpha \mathcal{L}_{\mathcal{W}}(G_{\mathcal{W}}, \hat{G}_R) \tag{61}$$

where α is a hyperparameter for domain confusion, and $\mathcal{L}_Y$ is the factual loss of the outcome.

The optimal values of $\hat{G}_Y$, $\hat{G}_R$ and $\hat{G}_{\mathcal{W}}$ achieve an equilibrium between the factual outcome prediction and domain confusion.

E.6. CRN+, CT+, EDCT+

We add the previous dosage to the sequential network (LSTM (CRN) or transformer (CT)) and the current dosage to G_Y, as inputs. Moreover, we add treatment dosage classifier network $G_{\mathcal{D}}$ to the baseline methods (CRN, CT and EDCT) (Figure 8).

CRN+

The encoder and decoder of CRN+ are trained by minimizing the loss, as follows:

$$\begin{aligned}\mathcal{L}(G_Y, G_{\mathcal{W}}, G_{\mathcal{D}}, G_R) = |\mathbf{y}_{t+1} - G_Y(\mathbf{\Phi}_t, \mathbf{w}_t, \mathbf{d}_t)|^2 - \lambda \sum_{w=1}^{n_{\mathcal{W}}} \mathbb{1}_{[w=\mathbf{w}_t]} \log G_W(\mathbf{\Phi}_t) \\ - \lambda \sum_{j=1}^{n_{\mathcal{D}}^{w_t}} \mathbb{1}_{[d^{k,j}=\mathbf{d}_t]} \log G_D(\mathbf{\Phi}_t, \mathbf{w}_t).\end{aligned} \tag{62}$$

CT+ AND EDCT+

We define the dosage discriminator loss and CDC loss for dosages, as follows:

$$\mathcal{L}_{\mathcal{D}}(G_D, G_R) = -\sum_{j=1}^{n_{\mathcal{D}}^{w_t}} \mathbb{1}_{[d^{w_t,j}=\mathbf{d}_t]} \log G_{\mathcal{D}}(\mathbf{\Phi}_t, \mathbf{w}_t) \tag{63}$$

$$\mathcal{L}_{conf}(G_D, G_R) = -\sum_{j=1}^{n_{\mathcal{D}}^{w_t}} \frac{1}{n_{\mathcal{D}}^{w_t}} \log G_{\mathcal{D}}(\mathbf{\Phi}_t, \mathbf{w}_t) \tag{64}$$

Then, (G_Y, G_R) and $(G_{\mathcal{W}}, G_{\mathcal{D}})$ are trained according to the minimax game defined as follows,

$$(\hat{G}_Y, \hat{G}_R) = \arg\min_{G_Y, G_R} \mathcal{L}_Y(G_Y, G_R) + \alpha \mathcal{L}_{conf}(\hat{G}_{\mathcal{W}}, G_R) + \alpha \mathcal{L}_{conf}(\hat{G}_{\mathcal{D}}, G_R) \tag{65}$$

$$(\hat{G}_{\mathcal{W}}, \hat{G}_{\mathcal{D}}) = \begin{cases} \hat{G}_{\mathcal{W}} = \underset{G_{\mathcal{W}}}{\text{argmin}}\ \alpha \mathcal{L}_{\mathcal{W}}(G_{\mathcal{W}}, \hat{G}_R) \\ \hat{G}_{\mathcal{D}} = \underset{G_{\mathcal{D}}}{\text{argmin}}\ \alpha \mathcal{L}_{\mathcal{D}}(G_{\mathcal{D}}, \hat{G}_R) \end{cases} \tag{66}$$

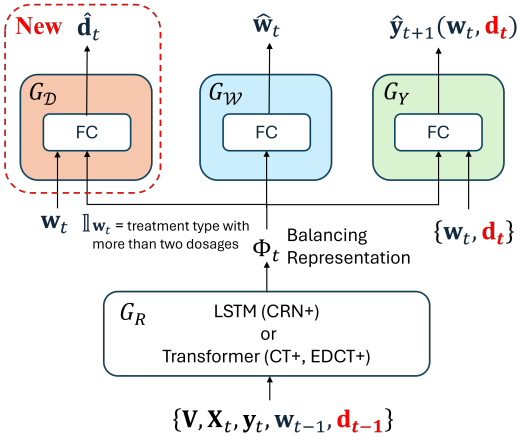


Figure 8: Architecture of CRN+, CT+, EDCT+

Appendix F. Implementation

F.1. Training

We implemented EDTS in PyTorch, and conduced training on AMD Ryzen Threadripper 3960X CPU and a 2 × Nvidia RTX 3090 GPUs. Baseline models were trained using Adam (Kingma and Ba, 2014) optimized with a learning rate η and a specified number of epochs n_e. During training, we employes an exponential decay of the learning rate η. The number of epochs (n_e) was set to 100 ~ 300 to account for the complexities of the baselines (Tables 4 and 5). Additionally, we used the teacher forcing technique (Williams and Zipser, 1989) when training the models for multiple-step-ahead predictions. During the evaluation of the multiple-step-ahead predictions, we switched off the teacher forcing and fed the model predictions autoregressively.

Adversarial Training

For the parameters α (in CDC loss of CT (EDCT) or OD loss of EDTS) and λ (in gradient reversal of CRN) of adversarial training, we chose $\alpha = 0.01$ and $\lambda = 1.00$, as chosen in the original works (Tzeng et al., 2015; Yazıcı et al., 2018). We applied exponential growth to both α and λ as follows:

$$\alpha_e = \alpha \cdot \left(\frac{2}{1 + \exp(-10 \times e/n_e)} - 1 \right), \qquad \lambda_e = \lambda \cdot \left(\frac{2}{1 + \exp(-10 \times e/n_e)} - 1 \right), \tag{67}$$

where $e \in 1, \ldots, n_e$ represents the current epoch index.

To enhance the stability of the adversarial training, in the original paper (Melnychuk et al., 2022), CT and EDCT use an exponential moving average (EMA) (Yazıcı et al., 2018) during training. We also use EMA for EDTS, CT and EDCT in this study. The detail of EMA is described in Melnychuk et al. (2022).

F.2. Hyperparameter tuning.

p, η, and all other hyperparameters (number of blocks B, minibatch size, number of attention heads n_h, size of hidden units d_h, size of representaion or balanced representation d_r, size of hidden units in fully-connected networks n_{FC}) are subjected to tuning. We conducted hyperparameter optimization for all benchmarks using a random grid search based on the factual RMSE of the validation set. The ranges of the hyperparameter grids are listed in Tables 4 and 5. Additional information on the model-specific hyperparameters is provided in Table 6. In specific cases such as in the LSTM hidden units or transformer blocks, we employed one- or two-layer sequential models. We conducted hyperparameter tuning separately for all the models, different values of the confounding amount using the 1,000 factual patient time-series from the validation set.

Table 4: Ranges for hyperparameter tuning across experiments of data using (1) the tumor growth (TG) simulator, (2) data from the semi-synthetic data from real-world EEG data. C is the input size. d_r is the size of BR or the output of the LSTM (in the case of G-Net).

Model	Sub-model	Hyperparameter	Range
RMSNs	Propensity Treatment Network	LSTM layers (B)	1, 2
		Learning late (η)	0.01, 0.001, 0.0001
		Minibatch size	64, 128, 256
		LSTM hidden units (d_h)	0.5C, 1C, 2C, 3C, 4C
		LSTM dropout rate (p)	0.1, 0.2, 0.3, 0.4, 0.5
		Max gradient norm	0.5, 1.0, 2.0
		Number of epochs (n_e)	100 (TG), 200 (EEG)
	Propensity History Network /Encoder	LSTM layers (B)	1, 2
		Learning late (η)	0.01, 0.001, 0.0001
		Minibatch size	64, 128, 256
		LSTM hidden units (d_h)	0.5C, 1C, 2C, 3C, 4C
		LSTM dropout rate (p)	0.1, 0.2, 0.3, 0.4, 0.5
		Max gradient norm	0.5, 1.0, 2.0
		Number of epochs (n_e)	100 (TG), 200 (EEG)
	Decoder	LSTM layers (B)	1, 2
		Learning late (η)	0.01, 0.001, 0.0001
		Minibatch size	256, 512, 1024
		LSTM hidden units (d_h)	1C, 2C, 4C, 8C, 16C
		LSTM dropout rate (p)	0.1, 0.2, 0.3, 0.4, 0.5
		Max gradient norm	0.5, 1.0, 2.0, 4.0
		Number of epochs (n_e)	100 (TG), 200 (EEG)
CRN	Encoder	LSTM layers (B)	1, 2
		Learning late (η)	0.01, 0.001, 0.0001
		Minibatch size	64, 128, 256
		LSTM hidden units (d_h)	0.5C, 1C, 2C, 3C, 4C
		BR size (d_r)	0.5C, 1C, 2C, 3C, 4C
		FC hidden units (n_{FC})	$0.5d_r$, $1d_r$, $2d_r$, $3d_r$, $4d_r$
		LSTM dropout rate (p)	0.1, 0.2, 0.3, 0.4, 0.5
		Number of epochs (n_e)	100 (TG), 200 (EEG)
	Decoder	LSTM layers (B)	1, 2
		Learning late (η)	0.01, 0.001, 0.0001
		Minibatch size	256, 512, 1024
		LSTM hidden units (d_h)	BR size of encoder
		BR size (d_r)	0.5C, 1C, 2C, 3C, 4C
		FC hidden units (n_{FC})	$0.5d_r$, $1d_r$, $2d_r$, $3d_r$, $4d_r$
		LSTM dropout rate (p)	0.1, 0.2, 0.3, 0.4, 0.5
		Number of epochs (n_e)	100 (TG), 200 (EEG)
G-Net	-	LSTM layers (B)	1, 2
		Learning late (η)	0.01, 0.001, 0.0001
		Minibatch size	64, 128, 256
		LSTM hidden units (d_h)	0.5C, 1C, 2C, 3C, 4C
		LSTM output size (d_r)	0.5C, 1C, 2C, 3C, 4C
		FC hidden units (n_{FC})	$0.5d_r$, $1d_r$, $2d_r$, $3d_r$, $4d_r$
		LSTM dropout rate (p)	0.1, 0.2, 0.3, 0.4, 0.5
		Number of epochs (n_e)	100 (TG), 200 (EEG)
CT	-	Transformer blocks (B)	1, 2
		Learning late (η)	0.01, 0.001, 0.0001
		Minibatch size	64, 128, 256
		Attention head (n_h)	2, 3
		Transformer units (d_h)	1C, 2C, 3C, 4C
		BR size (d_r)	0.5C, 1C, 2C, 3C, 4C
		FC hidden units (n_{FC})	$0.5d_r$, $1d_r$, $2d_r$, $3d_r$, $4d_r$
		Sequential dropout rate (p)	0.1, 0.2, 0.3, 0.4, 0.5
		Max positional encoding (l_{max})	15 (TG), 20 (EEG)
		Number of epochs (n_e)	150 (TG), 300 (EEG)

Table 5: Ranges for hyperparameter tuning across experiments. C is the input size. d_r is the size of BR in EDCT or representation in EDTS.

Model	Sub-model	Hyperparameter	Range
EDCT	Encoder	Transformer blocks (B)	1, 2
		Learning late (η)	0.01, 0.001, 0.0001
		Minibatch size	64, 128, 256
		Attention head (n_h)	2, 3
		Transformer units (d_h)	1C, 2C, 3C, 4C
		BR size (d_r)	0.5C, 1C, 2C, 3C, 4C
		FC hidden units (n_{FC})	$0.5d_r$, $1d_r$, $2d_r$, $3d_r$, $4d_r$
		Sequential dropout rate (p)	0.1, 0.2, 0.3, 0.4, 0.5
		Max positional encoding (self-attention) (l_{max})	15 (TG), 20 (EEG)
		Number of epochs (n_e)	100 (TG), 200 (EEG)
	Decoder	Transformer blocks (B)	1, 2
		Learning late (η)	0.01, 0.001, 0.0001
		Minibatch size	256, 512, 1024
		Attention head (n_h)	2, 3
		Transformer units (d_h)	BR size of encoder
		BR size (d_r)	0.5C, 1C, 2C, 3C, 4C
		FC hidden units (n_{FC})	$0.5d_r$, $1d_r$, $2d_r$, $3d_r$, $4d_r$
		Sequential dropout rate (p)	0.1, 0.2, 0.3, 0.4, 0.5
		Max positional encoding (cross-attention) (l_{max})	τ_{max}
		Max positional encoding (self-attention) (l_{max})	15 (TG), 20 (EEG)
		Number of epochs (n_e)	100 (TG), 200 (EEG)
EDTS	Encoder	Transformer blocks (B)	1, 2
		Learning late (η)	0.01, 0.001, 0.0001
		Minibatch size	64, 128, 256
		Attention head (n_h)	2, 3
		Transformer units (d_h)	1C, 2C, 3C, 4C
		R size (d_r)	0.5C, 1C, 2C, 3C, 4C
		FC hidden units (n_{FC})	$0.5d_r$, $1d_r$, $2d_r$, $3d_r$, $4d_r$
		Invariant/Eqivariant hidden units (n_{IE})	FC hidden units
		Sequential dropout rate (p)	0.1, 0.2, 0.3, 0.4, 0.5
		Max positional encoding (self-attention) (l_{max})	15 (TG), 20 (EEG)
		Number of epochs (n_e)	100 (TG), 200 (EEG)
	Decoder	Transformer blocks (B)	1, 2
		Learning late (η)	0.01, 0.001, 0.0001
		Minibatch size	256, 512, 1024
		Attention head (n_h)	2, 3
		Transformer units (d_h)	R size of encoder
		R size (d_r)	0.5C, 1C, 2C, 3C, 4C
		FC hidden units (n_{FC})	$0.5d_r$, $1d_r$, $2d_r$, $3d_r$, $4d_r$
		Invariant/Eqivariant hidden units (n_{IE})	FC hidden units
		Sequential dropout rate (p)	0.1, 0.2, 0.3, 0.4, 0.5
		Max positional encoding (self-attention) (l_{max})	τ_{max}
		Max positional encoding (cross-attention) (l_{max})	15 (TG), 20 (EEG)
		Number of epochs (n_e)	100 (TG), 200 (EEG)

Table 6: Additional information on model-specific hyperparameters (constant for all experiments).

Model	Sub-model	Hyperparameter	Value
RMSNs	Propensity Treatment Network	Random search iterations	50
		Input size (C)	d_w
		Output size	d_w
	Propensity History Network	Random search iterations	50
		Input size (C)	$d_y + d_x + d_v + d_w + d_d$
		Output size	d_w
	Encoder	Random search iterations	50
		Input size (C)	$d_y + d_x + d_v + d_w + d_d$
		Output size	d_y
	Decoder	Random search iterations	20
		Input size (C)	$d_y + d_v + d_w + d_d$
		Output size	d_y
CRN	Encoder	Random search iterations	50
		Input size (C)	$d_y + d_x + d_v + d_w + d_d$
		Output size	$d_y + d_w + d_d$
		Gradient reversal coefficient	1.0
	Decoder	Random search iterations	50
		Input size (C)	$d_y + d_v + d_w + d_d$
		Output size	$d_y + d_w + d_d$
		Gradient reversal coefficient	1.0
G-Net	-	Random search iterations	50
		Input size (C)	$d_y + d_x + d_v + d_w + d_d$
		Output size	d_y
		MC samples (M)	50
		Number of covariate groups	1
		Holdout dataset radio (empirical residuals)	10%
CT	-	Random search iterations	50
		Input size (C)	$\max\{d_y, d_x, d_v, d_w, d_d\}$
		Output size	$d_y + d_w$
		CDC coefficient	0.01
		EMA of model weights	0.99
		Positional encoding	relative, trainable
EDCT	Encoder	Random search iterations	50
		Input size (C)	$d_y + d_x + d_v + d_w + d_d$
		Output size	$d_y + d_w$
		CDC coefficient	0.01
		EMA of model weights	0.99
		Positional encoding	relative, trainable
	Decoder	Random search iterations	50
		Input size (C)	$d_y + d_v + d_w + d_d$
		Output size	$d_y + d_w$
		CDC coefficient	0.01
		EMA of model weights	0.99
		Positional encoding	relative, trainable
EDTS	Encoder	Random search iterations	50
		Input size (C)	$d_y + d_x + d_v + d_w + d_d$
		Output size	$d_y + d_w + d_d$
		OD coefficient	0.01
		EMA of model weights	0.99
		Positional encoding	relative, trainable
	Decoder	Random search iterations	50
		Input size (C)	$d_y + d_v + d_w + d_d$
		Output size	$d_y + d_w + d_d$
		OD coefficient	0.01
		EMA of model weights	0.99
		Positional encoding	relative, trainable

Appendix G. Details on experiments with synthetic data

G.1. Summary of Tumor Growth Simulator

The tumor growth simulator (Geng et al., 2017) models the volume of the tumor y_{t+1} for $t+1$ days after cancer diagnosis, resulting in a one-dimensional outcome, i.e., $d_y = 1$. The model includes two binary treatments: (i) radiotherapy (w_t^r) and (ii) chemotherapy (w_t^c), which are modeled as follows: (i) Radiotherapy when assigned to a patient has an immediate effect on the outcome. (ii) Chemotherapy affects several future outcomes with an exponentially decaying effect.

Model of tumor growth.

The volume of tumor t days after diagnosis is modelled as follows:

$$\underbrace{y_{t+1}}_{\substack{Volume\,of\\ Tumor}} = \Big\{ \underbrace{1 + \rho \log(K/y_t)}_{Growth} - \underbrace{\beta_c d_t^c}_{Chemotherapy} - \underbrace{\left(\alpha_r d_t^r + \beta_r {d_t^r}^2\right)}_{Radiatiotherapy} + \underbrace{\epsilon_t}_{Noise} \Big\} y_t \tag{68}$$

where the parameters $\rho, K, \beta_c, \alpha_r, \beta_r$ are samples from the prior distributions described in Geng et al. (2017), and $\epsilon_t \sim \mathcal{N}(0, 0.01^2)$ is independently sampled noise.

Patient type $V = \{1, 2, 3\}$ is randomly assigned to each patient. Type 1 patients respond more favorably to radiotherapy, denoted as $\mu'(\alpha_r) = 1.1 \times \mu(\alpha_r)$, while type 3 patients respond more favorably to chemotherapy, denoted as $\mu'(\beta_c) = 1.1 \times \mu(\beta_c)$. Here, $\mu(\alpha_r)$ represents the prior mean of α_r, and $\mu(\beta_c)$ represents the prior mean of β_c.

d_t^c is the chemotherapy dosage and d_t^r is the radiotherapy dosage. The dose-response curve for chemotherapy follows a linear function, while that for radiotherapy is quadratic.

Time-varying treatment types

In the setting of time-varying treatment types, the chemotherapy dosage $d_t^c = 5.0\ mg/cm^3$ and radiotherapy dosage $d_t^r = 2.0\ Gy$ are constant, and the treatmet dosages are not used as model inputs. Time-varying confounding is introduced by biased treatment type assignment, which is identical for both treatment types. The probabilities of the chemotherapy and radiotherapy treatment type assignments are as follows:

$$p(w_t^c = 1),\ p(w_t^r = 1) = \sigma\left(\frac{\gamma_w}{D_{max}}\left(D_{t-14:t} - \frac{D_{max}}{2}\right)\right) \tag{69}$$

where $w_t^c, w_t^r \in \{0, 1\}$, $D_{t-14:t}$ is the average tumor diameter over the last 15 days, D_{max} is the maximum tumor diameter 13 cm, and σ is the sigmoid.

γ_w controls the degree of time-dependent confounding of treatment type. For $\gamma_w = 0$, the treatment type assignment is fully randomized. With increasing values, the amount of time-varying confounding becomes also increases. Chemotherapy and radiotherapy assignment are independent, and there are four treatment type options: no treatment ($w_t^c = 0, w_t^r = 0$), chemotherapy ($w_t^c = 1, w_t^r = 0$), radiotherapy ($w_t^c = 0, w_t^r = 1$), and both ($w_t^c = 1, w_t^r = 1$) at each time step.

For the test data for 1-step-ahead prediction, we use counterfactual simulation data (CSD) that are simulated with inverted probabilities of factual treatment type assignments $p^{inv}(w_t^c = 1) = p(w_t^c = 0)$ and $p^{inv}(w_t^r = 1) = p(w_t^r = 0)$. For τ-step-ahead prediction ($\tau > 1$), CSD is highly counterfactual, making it difficult to evaluate the baseline methods accurately. Subsequently, the

counterfactual trajectory data after 2-step from each time step of CSD is simulated with a random treatment type assignment (random trajectory used in CT).

TIME-VARYING TREATMENT TYPES AND DOSAGES

In the setting of time-varying treatment types and dosages, full (regular) and half dosages can be chosen for each treatment type at each time step (Table 7), and the treatment dosages are used as model inputs.

Table 7: Treatment type-dosage pair

	Dosage	
Type	Full (Regular)	Half
no treatment	0	-
chemotherapy	5	2.5
radiotherapy	2	1
both	7	3.5

The probabilities of full dosage assignments are given in Eq.70 and Eq.71. γ_d is the amount of time-varying confounding parameter of the dosage assignment.

$$p(d_t^c = Full),\ p(d_t^r = Full) = \sigma\left(\frac{\gamma_d}{D_{max}}\left(D_{t-14:t} - \frac{D_{max}}{2}\right)\right) \tag{70}$$

$$p(d_t^b = Full) = \frac{p(d_t^c = Full) + p(d_t^r = Full)}{2} \tag{71}$$

where $d_t^c,\ d_t^r,\ d_t^b \in \{\text{Half}, \text{Full}\}$ are dosage assignments of chemotherapy, radiotherapy, and both treatments, respectively.

For the test data for 1-step-ahead prediction, CSD are simulated with an inverted factual treatment and dosage assignment (i.e., $p^{inv}(d_t^c = Full) = p(d_t^c = Half)$). The counterfactual trajectory data after 2-step from each time step of the CSD are simulated using a random treatment and dosage assignment.

G.2. Dataset.

For each level of confounding γ, we simulate 10,000 patient trajectories for training, and 1,000 each for validation and testing. We limit the length of the trajectories to a maximum of 60 time steps. Some patients have shorter trajectories due to recovery or death. For training, we only use periods before death or recovery, so these events are not considered in this study.

G.3. Performance measurement.

We retrain the models on five simulated datasets using different random seeds. We then report the averaged root mean square error (RMSE) on the test set, i.e., for the hold-out data. We report a normalized RMSE, normalized by the maximum tumor volume, $y^{max} = 1150\ cm^3$.

G.4. Additional Results (prior setting)

We compared our EDTS performance with a prior studies setting of time-varying treatment types. For this setting, we modified EDTS to omit the input of dosages and dosage discriminators (EDTS-).

Table 8 presents the results. Compared with the baseline methods, EDTS improved the prediction accuracy of counterfactual outcomes. This proves that GAN is effective for handling treatment assignment bias in the longitudinal setting, and not just in the static setting.

Table 8: Normalized RMSE for τ-step-ahead prediction in the setting of time-varying treatment types (setting in prior studies). Shown: mean ± standard deviation over five runs (lower is better). γ_w represents the amount of time-varying confounding parameters, with higher values indicating larger assignment bias of treatments.

		$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$	$\tau = 6$
$\gamma_w = 0$	MSMs	1.02 (±0.06)	1.44 (±0.10)	1.80 (±0.13)	1.51 (±0.10)	1.28 (±0.09)	1.09 (±0.08)
	RMSNs	0.66 (±0.06)	0.76 (±0.07)	0.77 (±0.07)	0.74 (±0.07)	0.68 (±0.06)	0.62 (±0.06)
	CRN	**0.63 (±0.06)**	**0.73 (±0.07)**	**0.74 (±0.07)**	0.72 (±0.07)	0.66 (±0.06)	**0.60 (±0.06)**
	G-Net	0.65 (±0.07)	0.77 (±0.08)	0.78 (±0.07)	0.75 (±0.07)	0.70 (±0.06)	0.64 (±0.05)
	CT	0.72 (±0.06)	0.83 (±0.07)	0.84 (±0.07)	0.81 (±0.06)	0.75 (±0.06)	0.69 (±0.05)
	EDCT	0.65 (±0.07)	0.75 (±0.07)	0.76 (±0.07)	0.72 (±0.07)	0.67 (±0.06)	0.61 (±0.06)
	EDTS-	**0.63 (±0.06)**	**0.73 (±0.08)**	**0.74 (±0.07)**	**0.71 (±0.07)**	**0.65 (±0.07)**	**0.60 (±0.06)**
$\gamma_w = 2$	MSMs	1.48 (±0.08)	2.38 (±0.15)	2.81 (±0.21)	2.33 (±0.20)	1.94 (±0.18)	1.61 (±0.16)
	RMSNs	0.78 (±0.08)	1.08 (±0.09)	1.14 (±0.10)	1.10 (±0.10)	1.01 (±0.10)	0.92 (±0.09)
	CRN	**0.71 (±0.07)**	1.00 (±0.09)	1.06 (±0.10)	1.02 (±0.11)	0.94 (±0.11)	0.85 (±0.11)
	G-Net	0.76 (±0.08)	1.06 (±0.10)	1.14 (±0.11)	1.11 (±0.12)	1.04 (±0.12)	0.95 (±0.12)
	CT	0.75 (±0.06)	1.01 (±0.08)	1.07 (±0.09)	1.04 (±0.09)	0.97 (±0.09)	0.88 (±0.09)
	EDCT	0.72 (±0.09)	1.01 (±0.07)	1.05 (±0.08)	1.01 (±0.09)	0.94 (±0.10)	0.85 (±0.10)
	EDTS-	**0.71 (±0.08)**	**0.98 (±0.09)**	**1.04 (±0.11)**	**1.00 (±0.12)**	**0.93 (±0.12)**	**0.84 (±0.12)**
$\gamma_w = 4$	MSMs	2.23 (±0.12)	3.63 (±0.24)	4.27 (±0.31)	3.46 (±0.29)	2.81 (±0.26)	2.31 (±0.23)
	RMSNs	1.16 (±0.24)	1.76 (±0.22)	1.83 (±0.23)	1.76 (±0.22)	1.63 (±0.19)	1.48 (±0.17)
	CRN	1.04 (±0.10)	1.54 (±0.16)	1.62 (±0.18)	1.57 (±0.19)	**1.45 (±0.18)**	**1.33 (±0.16)**
	G-Net	**0.94 (±0.14)**	**1.50 (±0.19)**	1.67 (±0.23)	1.69 (±0.23)	1.63 (±0.24)	1.54 (±0.23)
	CT	1.09 (±0.13)	**1.50 (±0.18)**	1.61 (±0.21)	1.62 (±0.23)	1.57 (±0.25)	1.49 (±0.26)
	EDCT	1.01 (±0.21)	1.82 (±0.15)	1.85 (±0.15)	1.69 (±0.14)	1.50 (±0.13)	1.34 (±0.12)
	EDTS-	0.99 (±0.12)	**1.50 (±0.15)**	**1.60 (±0.17)**	**1.55 (±0.17)**	**1.45 (±0.16)**	**1.33 (±0.15)**

Appendix H. Details on experiments with semi-synthetic data

H.1. Data

We used the EEG dataset (Zhang et al., 1995), which can be downloaded from https://kdd.ics.uci.edu/databases/eeg/eeg.html. This dataset originatess from a large study examining EEG correlates of genetic predisposition to alcoholism. It contains measurements from 64 electrodes placed on the scalp sampled at 256 Hz (3.9-millisecond epochs) for one second. For our semi-synthetic data, we randomly select 24 of 64 electrodes as time-varying covariates, and use first 100 of 256 time steps.

There were two groups of subjects: alcoholic and control. Each subject was exposed to either a single stimulus (S1) or to two stimuli (S1 and S2). We use the groups and exposures as static covariates. We one-hot-encode all static covariates and use them later further for generating noise. Altogether, this results into a 5-dimensional feature vector ($d_v = 5$).

Our simulation of semi-synthetic data for time-varying treatment types and dosages is based on the framework establised by Melnychuk et al. (2022) for time-varying treatment types. We begin by generating untreated outcome trajectories under endogeneous and exogeneous dependencies. Then, we sequentially apply treatments to these trajectories. We assume that the dependencies between treatments, outcomes, and time-varying covariates are sparse, meaning that outcomes are influenced by only a few treatments and covariates. Treatment assignments depend on a limited number of outcomes and covariates.

Our semi-synthetic simulator proceeds as follows.

First, we randomly select 1,000 trials of the EEG data set that contains all 120 trials for 122 subjects (all over 14,640 trials).

Second, we simulate d_y untreated outcomes $\mathbf{Z}_t^{j,(i)}$, $j = 1 \dots, d_y$, for each trial i. Therein, we combine (1) an endogenous component (B-spline (t) and random function $g^{j,(i)}$); (2) an exogenous dependency $f_Z^j(\mathbf{X}_t^{(i)})$ on a subset of current time-varying covariates; and (3) independent random noise ϵ_t. Formally, we generate the simulations via

$$\mathbf{Z}_t^{j,(i)} = \underbrace{\alpha_S^j \text{B-spline}(t) + \alpha_g^j g^{j,(i)}(t)}_{\text{endogenous}} + \underbrace{\alpha_f^j f_Z^j(\mathbf{X}_t^{(i)})}_{\text{exogenous}} + \underbrace{\epsilon_t}_{\text{noise}} \tag{72}$$

with $\epsilon_t \sim N(0, 0.005^2)$ and where α_S^j, α_g^j, and α_f^j are weight parameters.

Further, the B-spline (t) is sampled from a mixture of three cubic splines: one with a rapid decline throughout the intensive care unit stay, one with a mild decline, and one that remains stable. $g^{j,(i)}(\cdot)$ is independently sampled for each patient from Gaussian process with Matérn kernel; and $f_Z^j(\cdot)$ is sampled using a random Fourier features (RFF) approximation of a Gaussian process (Hensman et al., 2018). We specifically use to avoid the tedious Cholesky decomposition when sampling random functions at many points in the time-varying feature space $\mathbb{R}^{d_x}$. By combining all three components, we aim to simulate outcomes that exhibit endogenous dependencies at different resolutions (global trends of B-splines and the local correlation structure of Gaussian processes) and arbitrarily chosen exogeneous dependencies on other time-varying features.

Third, we sequentially simulate three synthetic treatments $\mathbf{A}_t^l = \{\mathbf{W}_t^l, \mathbf{D}_t^l\}, l = 1, \dots, d_a$ ($d_a = 4$, one is no treatment) as shown in Table 9. We add confounding to the treatments by a subset of current time-varying covariates via a random function $f_Y^l(\mathbf{X}_t)$. Subsequently, we average of the subset of previous T_l treated outcomes $\bar{A}_{T_l}(\bar{\mathbf{y}}_{t-1})$. Formally, we compute $\mathbf{A}_t^l$ via

$$p_{\mathbf{A}_t^l} = \sigma\left(\gamma_A^l \bar{A}_{T_l}(\bar{\mathbf{y}}_{t-1}) + \gamma_X^l f_Y^l(\mathbf{X}_t) + b_j\right), \tag{73}$$

$$\mathbf{A}_t^l = \begin{cases} \mathbf{W}_t^l \sim \text{Softmax}(p_{\mathbf{A}_t^1}, p_{\mathbf{A}_t^2}, \ldots, p_{\mathbf{A}_t^{d_a}}), \\ \mathbf{D}_t^l \sim \text{Berunoulli}(p_{\mathbf{A}_t^l}), \end{cases} \tag{74}$$

where $\sigma(\cdot)$ is the sigmoid activation, γ_A^l and γ_X^l are confounding parameters, b_l is a fixed bias, and $f_Y^l(\cdot)$ is sampled from an RFF approximation of a Gaussian process (similar to $f_Z^j(\cdot)$).

Table 9: Treatment type-dosage pair
Half-dose effect is the ratio of the treatment effect of a half-dose to that of a full-dose.

	Dosage		Half-dose effect
Type	Full	Half	
no treatment	0	-	-
treatment 1	1	0.5	0.25
treatment 2	1	0.5	0.50
treatment 3	1	0.5	0.75

Fourth, we apply treatments to the untreated outcomes. For this, we set $\mathbf{y}_1 = \mathbf{Z}_1$. Each treatment l is modeled to have a long-lasting effect on some outcome j, with a maximal additive effect β_{lj} immediately after application. We assume that the treatment has an effect within the time window $t - w^l, \ldots, t$. Additionally, we assume that the effect size of treatments follows an inverse-square decay over time. We also scale the effect by the probability $p_{\mathbf{A}_t^l}$. Afterward, the effects of multiple treatments are aggregated by taking the minimum across the treatment effects. Formally, we model this via

$$E^j(t) = \sum_{i=t-w^t}^{t} \frac{\min_{l=1,\ldots,d_a} \mathbb{1}_{[\mathbf{A}_i^l=1]} p_{\mathbf{A}_i^l} \beta_{lk}}{(w^l - i)^2}, \tag{75}$$

where β_{lj} is the maximum effect size of treatment l. This is either constant for all the outcomes j, or zero, so that the treatment does not influence the outcome.
Fifth, we combine the above. That is, we simply add the simulated treatment effect $E^j(t)$ to untreated outcome; i. e.,

$$\mathbf{y}_t^j = \mathbf{Z}_t^j + E^j(t) \tag{76}$$

Sixth, we generate our semi-synthetic dataset based on the above simulator. After simulating no treatment and three synthetic treatments ($d_a = 4$), and two synthetic outcomes ($d_y = 2$),

We split 1,000 trials into train/validation/test subsets using a 60% / 20% / 20 % split. The test data for 1-step-ahead prediction is simulated by inverting the actual treatment and dosage assignments (i.e., $p_{\mathbf{A}_t^l}^{inv} = 1 - p_{\mathbf{A}_t^l}$). For multiple-step-ahead prediction with $\tau_{max} = 6$, the test data after the 2-step ($\tau = 2, \ldots, \tau_{max}$) is simulated using random treatment and dosage assignments. We then sample 5 random trajectories for each patient at each time step.

H.2. Experimental Details

HYPERPARAMETER TUNING.

We perform hyperparameter tuning separately for all models. For this, we use the 200 factual time-series from the validation subset. Details are in Appendix F.

PERFORMANCE MEASUREMENT.

We retrain the models on five simulated datasets with different random seeds (the random seeds for sampling from Gaussian processes are kept the same). We then report the averaged root mean square error (RMSE) on the test set, which is the hold-out data. RMSE is calculated for standardized outcomes.