# CAUSAL AUTOENCODING AND TREATMENT CONDITIONING FOR COUNTERFACTUAL ESTIMATION OVER TIME

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Counterfactual estimation over time is important in various applications, such as personalized medicine. However, time-dependent confounding bias in observational data still poses a significant challenge in achieving accurate and efficient estimation. We introduce causal autoencoding and treatment conditioning (CAETC), a novel method for this problem. Built on adversarial representation learning, our method leverages an autoencoding architecture to learn a partially invertible and treatment-invariant representation, where the outcome prediction task is cast as applying a treatment-specific conditioning on the representation. Our design is independent of the underlying sequence model and can be applied to existing architectures such as long short-term memories (LSTMs) or temporal convolution networks (TCNs). We conduct extensive experiments on synthetic, semi-synthetic, and real-world data to demonstrate that CAETC yields significant improvement in counterfactual estimation over existing adversarial methods.

## 1 INTRODUCTION

Personalized medicine requires knowledge about individualized responses to potential treatments for effective treatment planning (Kent et al., 2018; Feuerriegel et al., 2024). While randomized controlled trials (RCTs) are the gold standard for treatment effect estimation, RCTs are often costly and difficult to implement in practice. Digitization enables more healthcare data to be recorded, especially via electronic health records. Combining large data volumes with appropriate modeling can enable accurate estimation of counterfactual outcomes and treatment effects, forming the basis for individualized decision making in health care (Bareinboim & Pearl, 2016) and other domains.

Modern counterfactual estimation methods integrate frameworks from causal inference and deep learning for counterfactual estimation over time (Lim et al., 2018; Bica et al., 2019; Melnychuk et al., 2022; Bouchattaoui et al., 2024; Wang et al., 2024). However, counterfactual estimation over time is difficult due to time-dependent confounding bias, where covariates that affect treatment choices evolve dynamically and are themselves influenced by prior treatments (Robins et al., 2000). This results in systematic differences in the distribution of confounders between treatment regimes, making traditional regression biased and complicating the estimation of unbiased causal effects. Conventional time-series models lack the capacity to adjust for this bias, relying on advanced methods to disentangle confounders from treatment assignment to reliably estimate treatment effects.

Learning to balance or disentangle time-dependent confounders is crucial for unbiased counterfactual estimation over time. Modern methods leverage sequence architectures like recurrent neural networks, transformers, or state-space models, with de-confounding mechanisms, like inverse probability of treatment weighting (IPTW) or adversarial balancing, to mitigate time-varying confounding bias, enabling better long-term treatment effect prediction in complex, sequential decision-making settings (Lim et al., 2018; Bica et al., 2019; Melnychuk et al., 2022; Bouchattaoui et al., 2024; Wang et al., 2024). For instance, recurrent marginal structure network (RMSN) (Lim et al., 2018) uses the LSTM architecture with IPTW to reweight the observed population toward balance across treatment assignments. Alternatively, counterfactual recurrent network (CRN) (Bica et al., 2019) uses the LSTM architecture with a gradient reversal layer, whereas causal transformer (CT) (Melnychuk et al., 2022)



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

uses the transformer architecture with domain confusion loss to learn an explicit treatment-invariant representation and reduce differences between treatment groups.

Despite these remarkable advances, several key limitations persist. It has been observed that existing balancing strategies, including adversarial representation learning used in CRN and CT, do not consistently improve counterfactual estimation in temporal settings (Huang et al., 2024). Therefore, a counterintuitive observation emerges: removing adversarial balancing (i.e., using only empirical risk minimization) improves counterfactual performance. Part of the phenomenon is attributed to historical covariate information loss, where reconstructing past inputs is more difficult when using adversarial learning. We interpret this covariate information loss as a consequence of learning a non-invertible representation. Representation invertibility is a common assumption to bound the generalization error for counterfactual estimation (Shalit et al., 2017; Johansson et al., 2022). Melnychuk et al. (2023) show that a non-invertible representation risks discarding covariate information and limits its ability to recover individual-level causal responses, i.e., suffers from loss of heterogeneity. More recent methods also identify invertible representations as an important assumption. Causal contrastive predictive coding (CCPC) (Bouchattaoui et al., 2024) tackles the problem implicitly by using the InfoMax principle to encourage representation invertibility. Conversely, Mamba-CDSP (Wang et al., 2024) uses an architecture-specific loss to decorrelate history from planned treatment to circumvent adversarial training. Encouraging full invertibility in the temporal setting requires the hidden representation to retain sufficient information to recover the history, which can be expensive and impact forecasting performance (MacKay et al., 2018). Therefore, we opt for a practical alternative: encouraging partial invertibility by reconstructing nearby time steps. Existing methods put little emphasis on modeling the interaction between the planned treatment and the balanced representation to decode future outcomes. While the balancing stage yields a theoretically valid representation, the interaction between the representation and planned treatment directly affects the future outcome prediction (see Fig. 2A). By making this interaction explicit, we propose an approach that simultaneously encourages partial invertibility and predicts future outcomes without excessive architecture changes or complex training procedures. Based on the aforementioned established evidence, our strategy tackles the problem of covariate information loss in adversarial representation learning methods and improves the counterfactual performance.

Inspired by these recent methods and their constraints, we present an architecture-agnostic method called *causal autoencoding and treatment conditioning (CAETC)* for counterfactual estimation over time. Our methodological contributions are the following:

- We design a model-agnostic method for counterfactual estimation over time that encourages a partially invertible representation via autoencoding and predicts future outcomes via treatment conditioning. We further encourage the conditioning layer to learn a treatment-specific transformation to improve performance. We apply the proposed design to LSTM and TCN.
- We propose an entropy maximization adversarial game that yields theoretically balanced representation across all treatment regimes. The adversarial game is equivalent to minimizing a generalized Jensen-Shannon divergence between treatment-conditional representation distributions. We further show that under certain assumptions and conditions, the potential outcome estimation error is bounded by observational estimation error and the aforementioned Jensen-Shannon divergence.
- We empirically validate CAETC on synthetic, semi-synthetic, and real-world datasets to demonstrate that our method achieves strong improvement over existing adversarial baselines.

## 2 Problem Formulation

### 2.1 Outcome forecasting

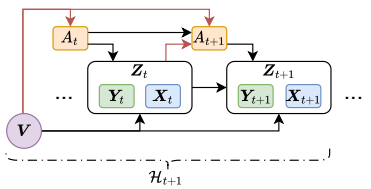


Figure 1: Causal graph for time-dependent confounding over $\mathcal{H}_t$.

Our method is designed in the context of the potential outcome framework Rubin (2005). Further details on assumptions for causal identification are in Appendix B. Consider a longitudinal dataset $\mathcal{D} = \left\{\{A_{i,t}, \boldsymbol{Y}_{i,t}, \boldsymbol{X}_{i,t}\}_{t=1}^{T}, \boldsymbol{V}_i\right\}_{i=1}^{N}$ of $N$ units and $T$ time steps with *categorical treatments* $A_{i,t}$, vector of *continuous outcomes of interest* $\boldsymbol{Y}_{i,t}$, vector of *continuous time-*

*varying covariates* $\boldsymbol{X}_{i,t}$, and vector of *static covariates* $\boldsymbol{V}_i$. Note that $\boldsymbol{Y}_{i,t}$ and $\boldsymbol{X}_{i,t}$ are disjoint covariates. Let $\boldsymbol{Z}_{i,t} = [\boldsymbol{X}_{i,t}, \boldsymbol{Y}_{i,t}]$ be the vector of all considered time-varying covariates. Since $\boldsymbol{X}_{i,t}$ may not be present for every dataset $\mathcal{D}$, $\boldsymbol{Z}_{i,t}$ is simply $\boldsymbol{Y}_{i,t}$ in that case. The history of a unit $i$ up to time $T_0$ is defined as $\mathcal{H}_{i,T_0} = \left\{ \{A_{i,t}, \boldsymbol{Z}_{i,t}\}_{t=1}^{T_0}, \boldsymbol{V}_i \right\}$. The treatment $A_{i,t}$ affects the time-varying covariates $\boldsymbol{Z}_{i,t}$. [1] For simplicity, we omit the unit index $i$ in the remainder of this article.

Consider the discrete treatment $A_t \in \{a^{(1)}, ..., a^{(K)}\}$ with $K$ possible values and denote the potential outcome under treatment $a^{(k)}$ as $\boldsymbol{Y}_t[a^{(k)}]$. Let the sequence of treatments from $t+1$ to $t+\tau$ be denoted by $A_{t+1:t+\tau}$. The task of interest is to estimate the outcome of the next $\tau$ time steps (prediction horizons) given the history up to $T_0$ and the non-random sequence of treatment $a_{T_0+1:T_0+\tau}$:

$$\mathbb{E}[\boldsymbol{Y}_{T_0+\tau}[a_{T_0+1:T_0+\tau}] \mid \mathcal{H}_{T_0}]. \tag{1}$$

Practically, using a neural network, the history $\mathcal{H}_t$ can be encoded into a latent space by a *representation network* $\Phi$ before being decoded into estimated outcomes by an *outcome head* $F_Y$, similar to prior works Shalit et al. (2017); Bica et al. (2019); Melnychuk et al. (2022); Bouchattaoui et al. (2024). Using autoregressive decoding, the estimation quantity can be expressed as:

$$F^Y(\Phi(\mathcal{H}_t), A_{t+1}) \approx \mathbb{E}[\boldsymbol{Y}_{t+1} \mid \Phi(\mathcal{H}_t), A_{t+1}]. \tag{2}$$

The existence of backdoor paths from static covariates $\boldsymbol{V}$ and current time-varying covariates $\boldsymbol{Z}_t$ to future treatment $A_{t+1}$, and the fact that current treatment $A_t$ also affects current time-varying covariates $\boldsymbol{Z}_t$ create time-dependent confounding. The confounding paths are shown in Fig. 1 in red. Under time-dependent confounding, estimating future outcomes is biased (Robins & Hernán, 2008).

# 3 METHOD

## 3.1 ARCHITECTURE OVERVIEW

As in prior works, we parametrize the representation layer $\Phi$ by a sequence network such as LSTM (CRN) or transformer (CT). After the representation layer $\Phi$, we make a crucial distinction in the architecture. In prior works, the learned representation is then forwarded to the respective treatment balancer $F^B$, or is concatenated with the next planned treatment $A_{t+1}$ to decode the next outcome $\boldsymbol{Y}_{t+1}$, as in Fig. 2A. In this case, if $\Phi(\mathcal{H}_t)$ is high-dimensional, then the influence of $A_{t+1}$ is also substantially reduced (Shalit et al., 2017).

Instead of the treatment $A_{t+1}$ being concatenated with $\Phi(\mathcal{H}_t)$ at the outcome decoder $F^Y$, we consider the treatment as conditioning information that transforms the representation at the $F^C$ layer, as in Fig. 2D. By explicitly modeling the treatment as conditioning information, more complex $F^C$ can be designed to modulate the representation while also enabling more flexible interactions that serve as building blocks for the rest of our method. Details on $F^C$ are deferred to Section 3.3.

## 3.2 INPUT AUTOENCODING

We start with a base autoencoding network to satisfy the representation invertibility as in Fig. 2B. At time step $T_0$, we encourage the decoding of current covariates $\{A_{T_0}, \boldsymbol{Z}_{T_0}\}$. Let $F^H$ be the time step decoding head, which includes three sub-heads: the *treatment decoding head* $F^A$, the *outcome of interest decoding head* $F^Y$, and the *time-varying covariate decoding head* $F^X$ if $\boldsymbol{X}_t$ exists in the dataset. Note that while we encourage the decoding of the current time step $T_0$ for simplicity and efficiency, more decoding heads $F^H$ can be added to decode the last $\nu$ time steps. However, existing sequence architectures have built-in mechanisms to selectively remove irrelevant information from the history, i.e., the forget gate in LSTM or the attention in transformer. Therefore, forcing decoding of many past inputs requires the network to remember those steps, which impairs the powerful learning ability of these mechanisms and can negatively impact performance (MacKay et al., 2018).

[1] In several works, the treatment $A_{i,t-1}$ is denoted to affect the covariates $\boldsymbol{Z}_{i,t}$. We depart from this notation to ensure the history $\mathcal{H}_{i,t}$ has the same number of time steps for both the treatment $A$ and covariates $\boldsymbol{Z}$. This change helps simplify the notation in several places while also easing implementation.

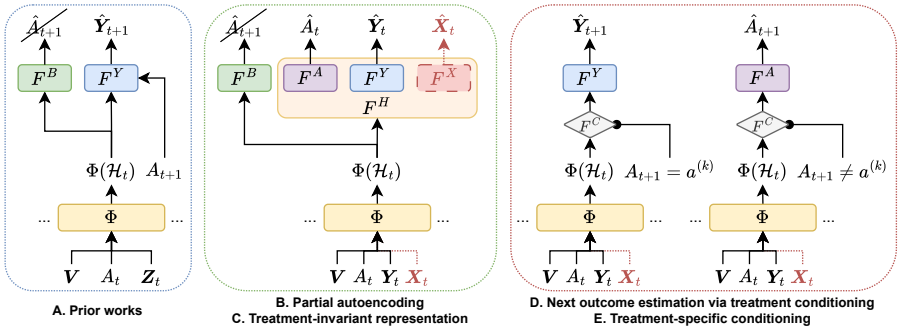


Figure 2: **A.** Prior works concatenate the representation $\Phi(\mathcal{H}_t)$ with treatment $A_{t+1}$ to predict next outcomes at $F^Y$. We make a distinction to model treatment $A_{t+1}$ as a transformation on the representation. **B.** The history $\mathcal{H}_t$ is encoded into representation $\Phi(\mathcal{H}_t)$ before being forwarded to respective heads for decoding. The treatment, outcome, and time-varying covariates estimators $F^A$, $F^Y$ and $F^X$ reconstruct $A_t$, $\boldsymbol{Y}_t$ and $\boldsymbol{X}_t$. **C.** Simultaneously, balancing is applied to $\Phi(\mathcal{H}_t)$ by maximizing the entropy of the treatment balancer $F^B$. **D.** $A_{t+1}$-specific transformation is applied to the $\Phi(\mathcal{H}_t)$ before being decoded to the next outcomes by $F^Y$. **E.** The conditioning layer $F^C$ is encouraged to learn treatment-specific transformations of $\Phi(\mathcal{H}_t)$.

We define the reconstruction loss for categorical treatments $A_t$ as the cross-entropy, and for continuous outcomes of interest $\boldsymbol{Y}_t$ and continuous time-varying covariates $\boldsymbol{X}$ as the mean squared error. For $\mathcal{H}_t, A_t, \boldsymbol{Y}_t, \boldsymbol{X}_t, \sim \mathcal{D}$, we have:

$$\mathcal{L}_t^{RA}(\theta_{F^A}, \theta_\Phi) = -\sum_{j=1}^{K} \mathbb{I}_{[A_t = a^{(j)}]} \log F_j^A(\Phi(\mathcal{H}_t)) \tag{3}$$

$$\mathcal{L}_t^{RY}(\theta_{F^Y}, \theta_\Phi) = \left\| F^Y(\Phi(\mathcal{H}_t)) - \boldsymbol{Y}_t \right\|_2^2 \tag{4}$$

$$\mathcal{L}_t^{RX}(\theta_{F^X}, \theta_\Phi) = \left\| F^X(\Phi(\mathcal{H}_t)) - \boldsymbol{X}_t \right\|_2^2 \tag{5}$$

where $\mathbb{I}_C$ is the indicator function for condition C. $\theta_{F^A}, \theta_{F^Y}, \theta_{F^X}, \theta_\Phi$ are the parameters of $F^A$, $F^Y$, $F^X$, and $\Phi$, accordingly.

The total reconstruction loss is then simply a combination of the above objectives with hyperparameters $\delta^A$ and $\delta^X$. If $\boldsymbol{X}_t$ is not present in the dataset, then $\delta^X = 0$.

$$\mathcal{L}_t^{R}(\theta_{F^A}, \theta_{F^Y}, \theta_{F^X}, \theta_\Phi) = \mathcal{L}_t^{RY}(\theta_{F^Y}, \theta_\Phi) + \delta^A \mathcal{L}_t^{RA}(\theta_{F^A}, \theta_\Phi) + \delta^X \mathcal{L}_t^{RX}(\theta_{F^X}, \theta_\Phi). \tag{6}$$

Prior work by (Huang et al., 2024) demonstrates that adversarial training (introduced in Section 3.4) is unstable and may cause the model to lose covariate information. Hence, the model risks violating representation invertibility (Shalit et al., 2017; Zhang et al., 2020; Johansson et al., 2022) and suffers loss of heterogeneity (Melnychuk et al., 2023). Despite being an important property to ensure causal identification, representation invertibility, in the context of longitudinal data, has rarely been enforced in practice (CRN (Bica et al., 2019), CT (Melnychuk et al., 2022)) or only been enforced implicitly (CCPC (Bouchattaoui et al., 2024)). Our design starts from explicit partial invertibility as a foundational building block instead.

### 3.3 Outcomes prediction via treatment conditioning

The task of interest is predicting the next outcome, which can be cast as applying a treatment-specific transformation on $\Phi(\mathcal{H}_t)$ that corresponds to the future outcomes. This conditioning mechanism can be considered as an inductive bias, i.e, the changes in $\boldsymbol{Z}_{t+1}$ are caused by applying $A_{t+1}$.

More specifically, we apply an element-wise affine transformation to the representation, also known as the feature-wise linear modulation (FiLM) (Perez et al., 2018). For each treatment $A_{t+1} = a^{(i)}$, FiLM learns a scaling vector $\xi^{(i)} = R^\xi(a^{(i)})$ and bias vector $\beta^{(i)} = R^\beta(a^{(i)})$, where $R^\xi, R^\beta$ are the generators that produce the corresponding vectors. The conditioning layer $F^C$ can then be defined as:

$$F^C(\Phi(\mathcal{H}_t), a^{(i)}) = \Phi(\mathcal{H}_t) \odot \xi^{(i)} \oplus \beta^{(i)} \tag{7}$$

where $\odot$ and $\oplus$ are element-wise multiplication and addition, respectively. Implementation-wise, $R^\xi$ and $R^\beta$ can be parameterized by a bag of embeddings for categorical treatment.

Reusing the outcome regressor $F^Y$, the next outcome prediction loss for $\mathcal{H}_t, A_{t+1}, \boldsymbol{Y}_{t+1} \sim \mathcal{D}$ is:

$$\mathcal{L}_t^Y(\theta_{F^Y}, \theta_{F^C}, \theta_\Phi) = \left\| F^Y(F^C(\Phi(\mathcal{H}_t), A_{t+1})) - \boldsymbol{Y}_{t+1} \right\|_2^2. \tag{8}$$

Prior works combine the treatment $A_{t+1}$ with the representation $\Phi(\mathcal{H}_t)$ via concatenation (Bica et al., 2019; Melnychuk et al., 2022; Bouchattaoui et al., 2024). After the next linear layer, this combination can be considered FiLM's special case, which is a learnable bias vector. However, this prior approach is limited in the expressiveness of the conditioning mechanism. Note that while we use an efficient FiLM mechanism, a more complex $F^C$ can be used.

### 3.4 Treatment-invariant representation learning

Leveraging adversarial domain adaptation, recent methods learn a treatment-invariant representation $\Phi(\mathcal{H}_t)$, breaking the association of history $\mathcal{H}_t$ and treatment $A_{t+1}$. Let $P^\Phi_{(j)}$ be the distribution of $\Phi(\mathcal{H}_t)$ conditioned on a $A_{t+1} = a^{(j)}$ for a fixed time step $t$, then recent methods aim to satisfy:

$$P^\Phi_{(1)} = P^\Phi_{(2)} = \cdots = P^\Phi_{(K)}. \tag{9}$$

This aspect has been explored in the literature. Let $P^A_{(j)}$ be the marginal distribution of $A_{t+1} = a^{(j)}$ for a fixed time step $t$, the simplified objective for each method is equivalent to:

$$\min_\Phi \textstyle\sum_{j=1}^K \frac{1}{K} \mathrm{KL}\left(P^\Phi_{(j)} \;||\; \frac{1}{K}\sum_{k=1}^K P^\Phi_{(k)}\right) \quad \text{( CRN (Bica et al., 2019, Appendix D))} \tag{10}$$

$$\min_\Phi \textstyle\sum_{j=1}^K \frac{1}{K} \mathrm{KL}\left(\sum_{k=1}^K P^A_{(k)} P^\Phi_{(k)} \;||\; P^\Phi_{(j)}\right) \quad \text{( CT (Melnychuk et al., 2022, Appendix F))} \tag{11}$$

$$\min_\Phi \textstyle\sum_{j=1}^K P^A_{(j)} \mathrm{KL}\left(P^\Phi_{(j)} \;||\; \sum_{k=1}^K P^A_{(k)} P^\Phi_{(k)}\right) + D \quad \text{(CCPC (Bouchattaoui et al., 2024, Appendix G.6))} \tag{12}$$

where equation 10 is equivalent to minimizing a multi-distribution Jensen-Shannon divergence with equal weights and equation 12 with weighting factors $P^A_{(1)}, ..., P^A_{(K)}$ (Lin, 1991) plus an additional expected divergence term $D$. The minima for equation 10, equation 11, and equation 12 satisfy equation 9.

**Adversarial entropy maximization.** Similar to existing works, we aim to learn a representation that satisfies equation 9. Consider the *balancing head* $F^B$, define the treatment classification loss as the cross-entropy for $\mathcal{H}_t, A_{t+1} \sim \mathcal{D}$:

$$\mathcal{L}_t^B(\theta_{F^B}) = -\textstyle\sum_{j=1}^K \mathbb{I}_{[A_{t+1}=a^{(j)}]} \log F_j^B(\Phi(\mathcal{H}_t)). \tag{13}$$

While the balancing head $F^B$ predicts the confounded planned treatment $A_{t+1}$, the representation $\Phi(\mathcal{H}_t)$ confuses the balancing head by removing such information. To learn a treatment-invariant representation, we propose to maximize the prediction entropy of $F^B$ for $\mathcal{H}_t \sim \mathcal{D}$:

$$\mathcal{L}_t^E(\theta_\Phi) = \textstyle\sum_{j=1}^K F_j^B(\Phi(\mathcal{H}_t)) \log F_j^B(\Phi(\mathcal{H}_t)). \tag{14}$$

Minimizing $\mathcal{L}_t^E$ is equivalent to maximizing the prediction entropy. For a fixed time step $t$, let $\boldsymbol{H} = \mathcal{H}_t$, we define the adversarial game as:

$$\begin{aligned} &\arg\min_{F^B} -\textstyle\sum_{j=1}^K \mathbb{E}_{\Phi(\boldsymbol{H})|A=a^{(j)}} \left[\log F_j^B(\Phi(\boldsymbol{H}))\right] P^A_{(j)} \\ &\arg\max_{\Phi} -\textstyle\sum_{j=1}^K \mathbb{E}_{\Phi(\boldsymbol{H})} \left[F_j^B(\Phi(\boldsymbol{H})) \log F_j^B(\Phi(\boldsymbol{H}))\right] \end{aligned} \quad \text{subject to } \textstyle\sum_{j=1}^K F_j^B(\Phi(\boldsymbol{H})) = 1. \tag{15}$$

Next, we show that the equilibrium of equation 15 satisfies equation 9, i.e, the representation is treatment-invariant.

**Theorem 3.1.** *For a fixed $t$, there exists a pair $\Phi$ and $F^B$ that satisfies the equilibrium in equation 15. The equilibrium holds if and only if $\Phi$ satisfies equation 9.*

We also obtained the simplified optimization objective for Eq. (15) as:

$$\min_\Phi \textstyle\sum_{j=1}^K P^A_{(j)} \mathrm{KL}\left(P^\Phi_{(j)} \;||\; \sum_{k=1}^K P^A_{(k)} P^\Phi_{(k)}\right). \tag{16}$$

This is similar to the objective of CCPC in Eq. (12) but with an efficient implementation as entropy maximization instead of minimizing the CLUB (Cheng et al., 2020) mutual information upper bound.

Ideally, achieving the equilibrium condition of equation 9 and satisfying representation invertibility results in a neural network with low estimation error for counterfactual outcomes. However, achieving the equilibrium of the adversarial game can be difficult in practice.

**Bound on prediction error.** While prior adversarial learning work has established the equilibrium case, the performance under a more general non-equilibrium case remains unknown. We can further show that the generalization error, under some assumptions, is bounded by the observational outcome prediction error and generalized Jensen-Shannon divergence between distributions inspired by the results from Shalit et al. (2017). Let $G^Y = F^Y \circ F^C$. Let $\bar{\mathcal{L}}^{G^Y,\Phi}(\boldsymbol{h}, a) = \mathbb{E}_{\boldsymbol{Y}|\boldsymbol{H}=\boldsymbol{h},A=a}\left[\mathcal{L}(G^Y(\Phi(\boldsymbol{h}), a), \boldsymbol{Y}[a])\right]$ for some non-negative loss function $\mathcal{L}$. The expected factual, counterfactual, and observational losses are defined as:

$$\epsilon^{\mathrm{F}}(G^Y, \Phi) = \int \textstyle\sum_{a=1}^{K} \bar{\mathcal{L}}^{G^Y,\Phi}(\boldsymbol{h}, a) P(\boldsymbol{H} = \boldsymbol{h}, A = a) d\boldsymbol{h} \tag{17}$$

$$\epsilon^{\mathrm{CF}}(G^Y, \Phi) = \int \textstyle\sum_{a=1}^{K} \sum_{a' \neq a} \bar{\mathcal{L}}^{G^Y,\Phi}(\boldsymbol{h}, a) P(\boldsymbol{H} = \boldsymbol{h}, A = a') d\boldsymbol{h} \tag{18}$$

$$\epsilon^{\mathrm{F}}_{(a)}(G^Y, \Phi) = \int \bar{\mathcal{L}}^{G^Y,\Phi}(\boldsymbol{h}, a) P(\boldsymbol{H} = \boldsymbol{h} | A = a) d\boldsymbol{h}. \tag{19}$$

**Theorem 3.2.** *Let $\Phi$ be an invertible representation function and $G^Y$ be a hypothesis. Let $S = \sup_{\boldsymbol{H},A} \left|\bar{\mathcal{L}}^{G^Y,\Phi}(\boldsymbol{h}, a)\right|$. Consider the weighting factors $\boldsymbol{\pi} \in \mathbb{R}^K$, $\boldsymbol{\pi} > 0$ , and $\sum_{a=1}^{K} \pi_a = 1$ and let $M = \left(\sqrt{\sum_{a=1}^{K} \frac{1}{\pi_a}} + (K-2)\sqrt{\sum_{a=1}^{K} \frac{P_{(a)}^{A\ 2}}{\pi_a}}\right)$, we then have:*

$$\epsilon^{\mathrm{F}}(G^Y, \Phi) + \epsilon^{\mathrm{CF}}(G^Y, \Phi) \leq \textstyle\sum_{a=1}^{K} \epsilon^{\mathrm{F}}_{(a)}(G^Y, \Phi) + SM\sqrt{2\,\mathrm{JS}_{\boldsymbol{\pi}}\left(P^{\Phi}_{(1)}, \ldots, P^{\Phi}_{(K)}\right)}. \tag{20}$$

It can be difficult to bound $S$ without additional assumptions, e.g., bounded loss functions. Alternatively, if we assume the domain of $\boldsymbol{V}$ and $\boldsymbol{Z}$ is bounded, then $S$ is bounded. In many applications, e.g, health care, we can typically assume naturally bounded distributions (e.g., heart rates, etc.).

### 3.5 Treatment-specific conditioning

By design, the conditioning layer $F^C$ should learn a treatment-specific transformation of the representation $\Phi(\mathcal{H}_t)$. However, during training, only the observable outcome corresponding to the factual treatment $A_{t+1} = a^{(k)}$ is optimized. Since the counterfactual outcomes cannot be observed, the conditioning with respect to remaining treatments $A_{t+1} \neq a^{(k)}$ does not receive any training signal.

However, by leveraging the same strategy for next outcome prediction, we synthetically applied the conditioning on the representation $\Phi(\mathcal{H}_t)$ for all remaining treatments. Reusing the treatment classifier $F^A$, for treatment $a^{(c)} \neq a^{(k)}$, we optimize:

$$\mathcal{L}^{C}_{t,(c)}(\theta_{F^A}, \theta_{F^C}, \theta_{\Phi}) = -\textstyle\sum_{j=1}^{K} \mathbb{I}_{[a^{(c)}=a^{(j)}]} \log F^A_j(F^C(\Phi(\mathcal{H}_t), a^{(c)})). \tag{21}$$

The final treatment-conditioning loss is the average of cross-entropy over all counterfactual treatments:

$$\mathcal{L}^{C}_{t}(\theta_{F^A}, \theta_{F^C}, \theta_{\Phi}) = \tfrac{1}{K-1} \textstyle\sum_{c \neq k} \mathcal{L}^{C}_{t,(c)}(\theta_{F^A}, \theta_{F^C}, \theta_{\Phi}). \tag{22}$$

Combining the treatment-conditioning loss $\mathcal{L}^C_t$ with the next outcome prediction loss $\mathcal{L}^Y_t$, for every sample in every time step, all possible configurations of the conditioning layer $F^C$ are optimized. While the $\mathcal{L}^C_t$ loss is easier to optimize compared to the $\mathcal{L}^Y_t$ loss, we observe that the additional training signal helps improve counterfactual estimation. Further, during training, to prevent overfitting to the treatment-conditioning loss, it is beneficial to add label smoothing (Müller et al., 2019), i.e., replacing $\mathbb{I}_{[a^{(c)}=a^{(j)}]}$ with $\mathbb{I}_{[a^{(c)}=a^{(j)}]}(1 - \alpha + \frac{\alpha}{K}) + \mathbb{I}_{[a^{(c)}\neq a^{(j)}]}\frac{\alpha}{K}$ in Eq. (21) for a hyperparameter $\alpha$.

### 3.6 Autoregressive decoding

At inference time, the predicted outcomes $\hat{\boldsymbol{Y}}_{t+1}$ are autoregressively reused to predict the next outcomes. However, if the dataset includes time-varying covariates $\boldsymbol{X}_t$, the network has a mismatched input dimension for future time steps, as $\boldsymbol{X}_{t+1}$ is not available. For certain types of data, a straightforward solution is also to learn to decode $\boldsymbol{X}_{t+1}$.

However, one of the potential problems with decoding $\boldsymbol{X}_t$ is that it can be difficult in many scenarios. Forcing the neural network to decode $\boldsymbol{X}_{t+1}$ in those cases may waste capacity on a less useful task. Existing works handle the problem by either using an encoder-decoder architecture (Lim et al., 2018; Bica et al., 2019; Bouchattaoui et al., 2024) or masking the corresponding attention of the decoder-only transformer (Melnychuk et al., 2022). We introduce a dropout-like model-agnostic approach called temporal cutoff that can be applied to a decoder-only model as a alternative to the encoder-decoder architecture in prior works.

**Temporal cutoff.** For each training data point, during training, all timesteps $t \geq T_{\text{cut}} \sim \text{Uniform}(1, T)$ are dropped. To represent the value of the dropped timesteps, a learnable missingness vector $\boldsymbol{M}$ is used. We apply the temporal cutoff to only $\boldsymbol{X}$ if it exists. To allow more capacity to represent missingness, we first linearly project $\boldsymbol{X}_t \in \mathbb{R}^u$ to a larger dimension using $\omega^{\boldsymbol{X}}$ so that $\omega^{\boldsymbol{X}}(\boldsymbol{X}_t) \in \mathbb{R}^v$ where $v > u$. For non-cutoff time steps, $\omega^{\boldsymbol{X}}(\boldsymbol{X}_t) \in \mathbb{R}^v$ is used, while for cutoff time steps, the missingness vector $\boldsymbol{M} \in \mathbb{R}^v$ is used as the input to the sequence model as in Fig. 3. We also linearly project $[\boldsymbol{V}, A_t, \boldsymbol{Y}_t]$ using $\omega^{\boldsymbol{VAY}}$ so that $\omega^{\boldsymbol{X}}(\boldsymbol{X}_t)$ do not dominate the input dimension. For simplicity, we let $\omega^{\boldsymbol{X}}$ and $\omega^{\boldsymbol{VAY}}$ projects to the same dimension, i.e., $\omega^{\boldsymbol{VAY}}([\boldsymbol{V}, A_t, \boldsymbol{Y}_t]) \in \mathbb{R}^v$.

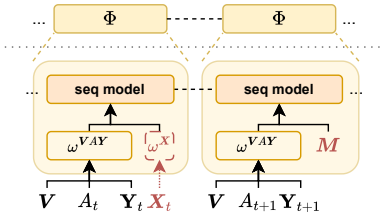


Figure 3: To handle input mismatch of the history $\mathcal{H}_{T_0} = \{\boldsymbol{V}, A_t, \boldsymbol{Y}_t, \boldsymbol{X}_t\}_{t=1}^{T_0}$ and autoregressively decoded sequences $\{\boldsymbol{V}, A_t, \boldsymbol{Y}_t\}_{t=T_0+1}^{T_0+\tau}$, we replace the future time-varying covariates $\{\boldsymbol{X}\}_{T_0+1}^{T_0+\tau}$ with a learnable vector $\boldsymbol{M}$.

### 3.7 Overall training objective

Reusing the hyperparameters $\delta^A$ from Eq. (6) (to reduce hyperparameter search effort) and adding an additional hyperparameter $\delta^E$, the overall training objective is defined as:

$$\begin{gathered} \min \delta^E \mathcal{L}_t^B(\theta_{F^B}) \\ \min \mathcal{L}_t^R(\theta_{F^A}, \theta_{F^Y}, \theta_{F^X}, \theta_\Phi) + \delta^E \mathcal{L}_t^E(\theta_\Phi) + \mathcal{L}_t^Y(\theta_{F^Y}, \theta_{F^C}, \theta_\Phi) + \delta^A \mathcal{L}_t^C(\theta_{F^A}, \theta_{F^C}, \theta_\Phi) \end{gathered}. \tag{23}$$

## 4 Experiments

### 4.1 Baselines

We evaluate the performance of CAETC against state-of-the-art methods for counterfactual estimation over time, including RMSN (Lim et al., 2018), CRN (Bica et al., 2019), CT (Melnychuk et al., 2022), and CCPC Bouchattaoui et al. (2024). As CAETC targets improving adversarial representation methods, we primarily compared against this family, while also providing an additional IPTW baseline (RMSN) for more comprehensive evaluation. Since our method is model-agnostic, we implement two variants: using LSTM (Hochreiter & Schmidhuber, 1997) and TCN (Bai et al., 2018), which we refer to as CAETC-LSTM and CAETC-TCN, respectively. We also provide a baseline using classical LSTM without balancing, which is equivalent to CRN with no adversarial training. Effective methods for counterfactual estimation should outperform classical LSTM.

### 4.2 Datasets

A challenge in evaluating counterfactual estimation methods is the lack of counterfactual outcomes in real-world datasets. Therefore, we use synthetic data to obtain counterfactual ground truths. To ensure comprehensive evaluation, we benchmark the performance against standard benchmarks used in prior works (Lim et al., 2018; Bica et al., 2019; Melnychuk et al., 2022), including fully synthetic data based on non-small cell lung cancer simulation Geng et al. (2017) and semi-synthetic data derived from the MIMIC-III dataset Johnson et al. (2016). We also evaluate CAETC on factual outcomes from real-world data in the MIMIC-III dataset to demonstrate performance in practice. More details are available in the Appendix.

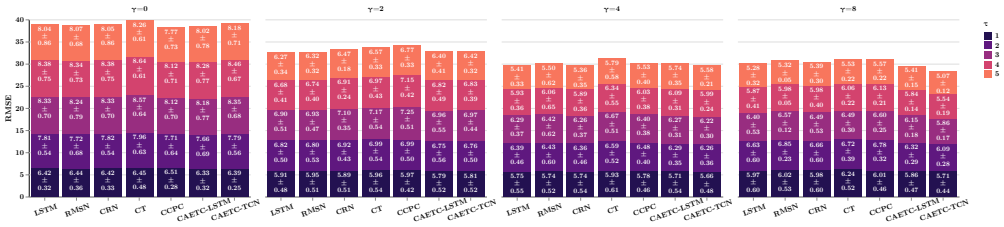


Figure 4: RMSEs for fully synthetic data on random trajectories with increasing time-dependent confounding during training. Each bar represents the RMSEs for 5-step predictions. ↓ is better.

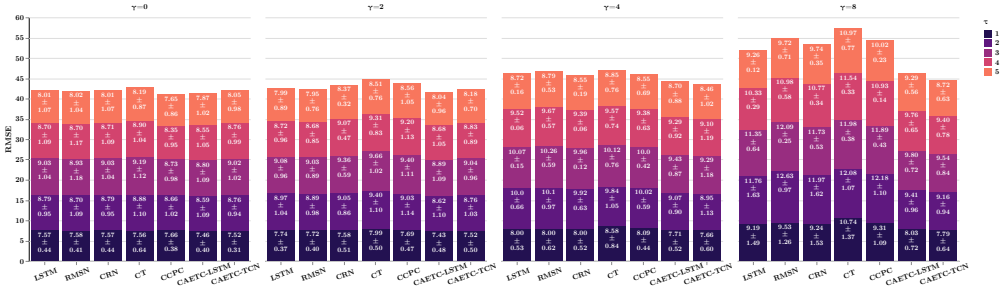


Figure 5: RMSEs for fully synthetic data on no-confounding test set ($\gamma = 0$) with increasing levels of time-dependent confounding during training. Training on $\gamma = 0$ and testing on $\gamma = 0$ represents the performance upper bound for each method. Due to time-dependent confounding, training on $\gamma \neq 0$ and testing on $\gamma = 0$ is expected to reduce performance. ↓ is better.

Table 1: RMSEs for semi-synthetic data with random trajectories. ↓ is better.

| | $\tau = 1$ | $\tau = 2$ | $\tau = 3$ | $\tau = 4$ | $\tau = 5$ | $\tau = 6$ | $\tau = 7$ | $\tau = 8$ | $\tau = 9$ | $\tau = 10$ | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LSTM | .218±.029 | .264±.022 | .319±.015 | .379±.010 | .439±.008 | .496±.007 | .552±.007 | .604±.009 | .655±.011 | .703±.014 | .463 |
| RMSN | .236±.040 | .287±.028 | .362±.016 | .438±.017 | .511±.020 | .579±.023 | .641±.025 | .699±.027 | .753±.029 | .803±.032 | .531 |
| CRN | .218±.028 | .269±.021 | .332±.014 | .399±.010 | .465±.009 | .527±.009 | .586±.011 | .641±.012 | .693±.014 | .743±.016 | .487 |
| CT | .174±.009 | .236±.009 | .310±.006 | .392±.005 | .477±.009 | .568±.012 | .662±.019 | .756±.026 | .845±.031 | .920±.037 | .534 |
| CCPC | .187±.015 | .236±.018 | .292±.018 | .351±.016 | .410±.015 | .467±.015 | .523±.015 | .577±.016 | .629±.018 | .679±.019 | 435 |
| CAETC LSTM | .197±.023 | .251±.022 | .299±.013 | .350±.006 | .401±.011 | .453±.019 | .504±.027 | .554±.034 | .603±.040 | .651±.046 | .426 |
| CAETC TCN | .179±.005 | .234±.011 | .294±.016 | .356±.019 | .418±.021 | .477±.022 | .536±.023 | .591±.023 | .646±.023 | .698±.025 | .443 |

### 4.3 Experiment with synthetic data

We build the fully synthetic data on the pharmacokinetics-pharmacodynamics model of non-small cell lung cancer (NSCLC) (Geng et al., 2017) under chemotherapy and radiotherapy. Simulation details are in Appendix E. We inject different levels of confounding from no confounding $\gamma = 0$ to strong confounding $\gamma = 8$. We set the maximum trajectory length $T$ to 60 and the prediction horizon $\tau$ to 5. We trained all methods on trajectories from 10000 patients and used a separate validation set of 1000 patients to select the best hyperparameters. We evaluate all methods under two scenarios: random trajectories and no-confounding. Further details on the test setting are available in the Appendix E.

**Random trajectories**. There are $K$ potential outcomes at every step. At prediction horizon $\tau$, there are an exponential $K^\tau$ outcomes, making simulation expensive. We randomly choose a subset of $k$ outcomes for evaluation. The setting is similar to Melnychuk et al. (2022). We set $k = 1$ and simulate 1000 patients.

**No confounding**. As $\gamma$ can be controlled, we evaluate the model on a non-confounded ($\gamma = 0$) test set to study if the model has learned unbiased estimation. This setting is more difficult than random trajectories, as neither the prediction horizon nor the input history resembles the confounded training set. The setting is similar to Lim et al. (2018). We use the trajectories of 1000 patients.

**Results.** CAETC-LSTM/TCN demonstrates strong performance with increasing time-dependent confounding as in Fig. 4 and Fig. 5. For all methods, training and testing on non-confounded datasets $\gamma = 0$ in Fig. 5 results in a very small difference in performance, which is desirable. Due to the limited amount of data, similar to real-world scenarios, LSTM and TCN architectures exhibit good performance compared to the transformer architecture. For small confounding, LSTM is stable and demonstrates strong performance compared to counterfactual methods. However, as the confounding

factor increases, only CAETC-LSTM/TCN exhibits stable counterfactual estimation performance. Prior works (CRN, CT) do not outperform an LSTM with empirical risk minimization due to covariate information loss, which is consistent with literature (Huang et al., 2024). Our method uses partial-autoencoding, which is less likely to suffer from the same issues. Further, it can be observed that under the challenging setting of Fig. 5, CAETC demonstrate strong improvement over existing works.

### 4.4 Experiment with semi-synthetic data

We employ a semi-synthetic dataset generated based on the MIMIC-III Johnson et al. (2016) intensive care units dataset. Simulation details are in Appendix F. We set the maximum trajectory length $T$ to 100 and the prediction horizon $\tau$ to 10. All methods are trained and validated on 2000 and 200 trajectories, respectively. We evaluate the method on random trajectories ($k = 1$) similar to Section 4.3, which includes 200 patients, resulting in 200 test horizons per time step $t$.

**Results.** CAETC-LSTM/TCN demonstrates strong improvement across all time steps, as in Table 1.

### 4.5 Experiment with real-world data

We further evaluate CAETC on real-world data based on the MIMIC-III dataset Johnson et al. (2016). However, for real-world data, counterfactual outcomes are not available. Nonetheless, based on Theorem 3.2, we can expect that a smaller sum of observational error indicates a tighter error bound. Therefore, the performance on observable outcomes is still a useful evaluation metric. We set the maximum trajectory length $T$ to 60 and the prediction horizon $\tau$ to 5. All methods are trained, validated, and tested on 5000, 500, and 500 trajectories sampled from the dataset, respectively.

Table 2: RMSEs for real-world data. ↓ is better.

| | $\tau = 1$ | $\tau = 2$ | $\tau = 3$ | $\tau = 4$ | $\tau = 5$ | Avg |
|---|---|---|---|---|---|---|
| LSTM | 6.81±0.03 | 7.36±0.04 | 7.64±0.04 | 7.84±0.02 | 8.01±0.04 | 7.53 |
| RMSN | 7.28±0.05 | 7.77±0.06 | 8.06±0.13 | 8.27±0.17 | 8.44±0.16 | 7.96 |
| CRN | 6.82±0.03 | 7.38±0.05 | 7.66±0.03 | 7.86±0.02 | 8.02±0.05 | 7.55 |
| CT | 6.70±0.04 | 7.37±0.06 | 7.71±0.06 | 7.94±0.05 | 8.12±0.03 | 7.57 |
| CCPC | 6.74±0.05 | 7.29±0.05 | 7.58±0.05 | 7.78±0.04 | 7.94±0.04 | 7.47 |
| CAETC LSTM | 6.73±0.03 | 7.28±0.06 | 7.57±0.08 | 7.77±0.1 | 7.93±0.12 | 7.46 |
| CAETC TCN | 6.67±0.07 | 7.24±0.08 | 7.51±0.08 | 7.70±0.07 | 7.85±0.05 | 7.39 |

**Results.** CAETC-LSTM/TCN demonstrates strong improvement across all time steps, as in Table 2.

### 4.6 Ablation

We examined the performance contribution of each component in CAETC using an LSTM backbone as in Table 3 with the same setting in Fig. 5 ($\gamma = 8$).

Table 3: Ablation with similar setting to Fig. 5 ($\gamma = 8$).

| | CAETC LSTM | $\tau = 1$ | $\tau = 2$ | $\tau = 3$ | $\tau = 4$ | $\tau = 5$ | Avg |
|---|---|---|---|---|---|---|---|
| (1) | Full | 8.03±0.72 | 9.41±0.96 | 9.80±0.72 | 9.76±0.65 | 9.29±0.56 | 9.26 |
| (2) | $\mathcal{L}^C = 0$ | 8.04±0.85 | 9.63±1.44 | 10.25±1.63 | 10.36±1.83 | 9.96±1.79 | 9.65 |
| (3) | $\mathcal{L}^C = 0$<br>$\mathcal{L}^R = 0$ | 9.37±1.13 | 11.28±0.82 | 11.29±0.22 | 10.79±0.22 | 10.10±0.25 | 10.57 |
| (4) | $\mathcal{L}^C = 0$<br>$\mathcal{L}^R = 0$<br>$\mathcal{L}^E = \mathcal{L}^B = 0$ | 8.67±0.39 | 10.65±0.68 | 11.13±1.07 | 11.04±1.39 | 10.53±1.46 | 10.40 |

**Results.** Observe that each objective contributes to the overall performance. We can see that the full CAETC (1) achieves the best performance. Removing the treatment conditioning loss as in (2) moderately affects the performance. Further removing reconstruction as in (3) negatively impact the performance. Without reconstruction loss, adding adversarial entropy maximization from (4) to (3) fails to improve the performance due to covariate information loss. Version (4) is the basic version with every objective removed and has similar performance to a regular LSTM.

## 5 Conclusion

We introduced a novel method for counterfactual estimation over time called causal autoencoding and treatment conditioning. CAETC utilizes a partial autoencoding architecture for representation invertibility and casts outcome prediction as treatment conditioning on the adversarially balanced representation. Extensive experiments show that CAETC improves significantly over baselines.

### AI USE STATEMENT

In this work, we used generative AI tools to aid or polish writing and assist in checking parts of mathematical proofs. We have reviewed all AI-assisted work.

## REFERENCES


Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271, April 2018.

Elias Bareinboim and Judea Pearl. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352, July 2016. doi: 10.1073/pnas.1510507113.

Ioana Bica, Ahmed M. Alaa, James Jordon, and Mihaela van der Schaar. Estimating counterfactual treatment outcomes over time through adversarially balanced representations. In *Proceedings of the International Conference on Learning Representations*, September 2019.

Mouad El Bouchattaoui, Myriam Tami, Benoit Lepetit, and Paul-Henry Cournède. Causal contrastive learning for counterfactual regression over time. In *Advances in Neural Information Processing Systems*, volume 37, pp. 1333–1369, 2024.

Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. CLUB: A contrastive log-ratio upper bound of mutual information. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 1779–1788, July 2020.

Frank C. Detterbeck and Christopher J. Gibson. Turning gray: The natural history of lung cancer over time. *Journal of Thoracic Oncology*, 3(7):781–792, July 2008. doi: 10.1097/JTO.0b013e31817c9230.

Stefan Feuerriegel, Dennis Frauen, Valentyn Melnychuk, Jonas Schweisthal, Konstantin Hess, Alicia Curth, Stefan Bauer, Niki Kilbertus, Isaac S. Kohane, and Mihaela van der Schaar. Causal machine learning for predicting treatment outcomes. *Nature Medicine*, 30(4):958–968, April 2024. doi: 10.1038/s41591-024-02902-1.

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 bio-mathematical model. *Scientific Reports*, 7(1):13542, October 2017. doi: 10.1038/s41598-017-13646-z.

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

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997. doi: 10.1162/neco.1997.9.8.1735.

Qiang Huang, Chuizheng Meng, Defu Cao, Biwei Huang, Yi Chang, and Yan Liu. An empirical examination of balancing strategy for counterfactual estimation on time series. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 20043–20062, June 2024.

Fredrik D. Johansson, Uri Shalit, Nathan Kallus, and David Sontag. Generalization bounds and representation learning for estimation of potential outcomes and causal effects. *Journal of Machine Learning Research*, 23(166):1–50, 2022.

Alistair E. W. 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):160035, May 2016. doi: 10.1038/sdata.2016.35.

Krzysztof Kacprzyk, Samuel Holt, Jeroen Berrevoets, Zhaozhi Qian, and Mihaela van der Schaar. ODE discovery for longitudinal heterogeneous treatment effects inference. In *Proceedings of the 12th International Conference on Learning Representations*, October 2023.

David M. Kent, Ewout Steyerberg, and David van Klaveren. Personalized evidence based medicine: Predictive approaches to heterogeneous treatment effects. *BMJ*, 363:k4245, December 2018. doi: 10.1136/bmj.k4245.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv:1412.6980, January 2017.

Rui Li, Stephanie Hu, Mingyu Lu, Yuria Utsumi, Prithwish Chakraborty, Daby M. Sow, Piyush Madan, Jun Li, Mohamed Ghalwash, Zach Shahn, and Li-wei H. Lehman. G-Net: A recurrent network approach to G-computation for counterfactual prediction under a dynamic treatment regime. In *Proceedings of Machine Learning for Health*, pp. 282–299, November 2021.

Bryan Lim, Ahmed Alaa, and Mihaela van der Schaar. Forecasting treatment responses over time using recurrent marginal structural networks. In *Advances in Neural Information Processing Systems*, volume 31, pp. 7483–7493, 2018.

J. Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, January 1991. doi: 10.1109/18.61115.

Matthew MacKay, Paul Vicol, Jimmy Ba, and Roger B Grosse. Reversible recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 31, pp. 9043–9054, 2018.

Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Causal transformer for estimating counterfactual outcomes. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 15293–15329, June 2022.

Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Bounds on representation-induced confounding bias for treatment effect estimation. In *Proceedings of the 12th International Conference on Learning Representations*, October 2023.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, volume 32, pp. 4694–4703, 2019.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, pp. 3942–3951, February 2018.

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):1393–1512, January 1986. doi: 10.1016/0270-0255(86)90088-6.

James Robins. A graphical approach to the identification and estimation of causal parameters in mortality studies with sustained exposure periods. *Journal of Chronic Diseases*, 40:139S–161S, January 1987. doi: 10.1016/S0021-9681(87)80018-8.

James M. Robins. Correcting for non-compliance in randomized trials using structural nested mean models. *Communications in Statistics - Theory and Methods*, 23(8):2379–2412, January 1994. doi: 10.1080/03610929408831393.

James M. Robins and Miguel A. Hernán. Estimation of the causal effects of time-varying exposures. In *Longitudinal Data Analysis*, 2008.

James M. Robins, Miguel Ángel Hernán, and Babette Brumback. Marginal structural models and causal inference in epidemiology. *Epidemiology*, 11(5):550, September 2000.

Donald B Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, March 2005. doi: 10.1198/016214504000001880.

Uri Shalit, Fredrik D. Johansson, and David Sontag. Estimating individual treatment effect: Generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3076–3085, July 2017.

Alexandre B. Tsybakov. Lower bounds on the minimax risk. In Alexandre B. Tsybakov (ed.), *Introduction to Nonparametric Estimation*, pp. 77–135. Springer, New York, NY, 2009. doi: 10.1007/978-0-387-79052-7_2.

Haotian Wang, Haoxuan Li, Hao Zou, Haoang Chi, Long Lan, Wanrong Huang, and Wenjing Yang. Effective and efficient time-varying counterfactual prediction with state-space models. In *Proceedings of the 13th International Conference on Learning Representations*, October 2024.

Shirly Wang, Matthew B. A. McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C. Hughes, and Tristan Naumann. MIMIC-Extract: A data extraction, preprocessing, and representation pipeline for MIMIC-III. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pp. 222–235, April 2020. doi: 10.1145/3368555.3384469.

Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, June 1989. doi: 10.1162/neco.1989.1.2.270.

Yao Zhang, Alexis Bellot, and Mihaela Schaar. Learning overlapping representations for the estimation of individualized treatment effects. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, pp. 1005–1014, June 2020.

## A Related Works

Counterfactual estimation has received great attention in both static and temporal settings. Here, we describe prior methods in the temporal domain.

**Counterfactual estimation over time:** Methods that control for time-dependent confounding were originally developed in epidemiology for longitudinal studies, such as the structural nested model (SNM) (Robins, 1994) and marginal structural model (MSM) (Robins et al., 2000). However, these methods have traditionally been implemented using simple parametric estimators with limited capacity to deal with complex, high-dimensional data. RMSN improves over MSM by incorporating LSTM to handle nonlinearities. However, RMSN still relies on IPTW and suffers from the same problems, such as high variance. Alternatively, G-Net (Li et al., 2021) adapts g-computation (Robins, 1986; 1987) to sequence model. Recent methods, such as CRN and CT (Bica et al., 2019; Melnychuk et al., 2022), use adversarial training to learn a treatment-invariant representation, removing the systematic differences between treatment regimes. Further progress has been made on diverse strategies to handle time-dependent confounding, e.g., adapting ODE discovery to treatment effect estimation (Kacprzyk et al., 2023).

**Learning treatment-invariant representation** Recent approaches adopt adversarial treatment-invariant representation learning (Bica et al., 2019; Melnychuk et al., 2022; Bouchattaoui et al., 2024) as a foundational building block to balance between treatment groups. More specifically, CRN (Bica et al., 2019) uses a gradient reversal layer Ganin et al. (2016) between an LSTM representation network and a treatment balancing head. The treatment balancing head is optimized to predict the treatment domain from the representation, whereas the gradient reversal layer encourages a treatment-invariant representation, forming an adversarial game. Similarly, CT (Melnychuk et al., 2022) replaces the LSTM backbone with a combination of three transformer sub-networks to capture long-range dependencies and proposes an adversarial game based on the domain confusion loss. However, an empirical study by Huang et al. (2024) demonstrates that adversarial training results in a significant loss of covariate information, which affects the outcome prediction performance of CRN and CT. Representation invertibility is used as a theoretical tool to ensure a valid representation (Shalit et al., 2017; Johansson et al., 2022; Melnychuk et al., 2023), which can help improve counterfactual estimation performance. CCPC (Bouchattaoui et al., 2024) implicitly encourages reconstructable representation by maximizing mutual information between the input and representation following the InfoMax principle. Alternatively, Mamba-CDSP (Wang et al., 2024) proposes an architecture-specific decorrelation loss to avoid the over-balancing problem of adversarial training. However, Mamba-CDSP is only equivalent to first-order moment matching of treatment regimes instead of matching the treatment conditional distribution.

# B ASSUMPTIONS FOR CAUSAL IDENTIFICATION

## B.1 ASSUMPTIONS

We make the standard assumptions for causal identification (Robins & Hernán, 2008) that are also used in related methods (Bica et al., 2019; Melnychuk et al., 2022; Bouchattaoui et al., 2024).

**Consistency**: If $A_{\leq t} = a_{\leq t}$ is a sequence of treatments for a given unit, then the potential outcome is equivalent to the observed outcome or $\boldsymbol{Y}_t[a_{\leq t}] = \boldsymbol{Y}_t$.

**Sequential Positivity**: For every time step, there is a strictly positive probability to observe any treatment level, given a realization of the history. Formally, if $P(\mathcal{H}_t = h_t) > 0$, then $0 < P(A_{t+1} = a_{t+1} | \mathcal{H}_t = h_t)$ for all $a_{t+1}$.

**Sequential Ignorability**: For every time step, the treatment assignment $A_{t+1}$ is independent of potential outcomes, conditioned on the history, or $\boldsymbol{Y}_{\geq t+1}[a_{\geq t+1}] \perp A_{t+1} | \mathcal{H}_t$ for all $a_{\geq t+1}$.

# C PROOF OF THEOREM 3.1

The following proofs are adapted from CRN (Bica et al., 2019) and CT (Melnychuk et al., 2022).

**Lemma C.1.** *For a fixed representation network $\Phi$, let $\boldsymbol{R} = \Phi(\boldsymbol{H})$, then the optimal predictor $F^B$ is:*

$$F_j^{B*}(\boldsymbol{R}) = \frac{P_{(j)}^A P_{(j)}^\Phi(\boldsymbol{R})}{\sum_{k=1}^K P_{(k)}^A P_{(k)}^\Phi(\boldsymbol{R})}. \tag{24}$$

*Proof.* The objective for the predictor $F^B$ in equation 15 is:

$$F^{B*} = \arg\min_{F^B} -\sum_{j=1}^K \int \log\left(F_j^B(\boldsymbol{r})\right) P_{(j)}^\Phi(\boldsymbol{r}) P_{(j)}^A d\boldsymbol{r} \quad \text{subject to} \quad \sum_{j=1}^K F_j^B(\boldsymbol{r}) = 1. \tag{25}$$

Minimizing the objective point-wise and applying the Lagrange multiplier:

$$F^{B*} = \arg\min_{F^B} -\sum_{j=1}^K \log\left(F_j^B(\boldsymbol{r})\right) P_{(j)}^\Phi(\boldsymbol{r}) P_{(j)}^A + \lambda \left(\sum_{j=1}^K F_j^B(\boldsymbol{r}) - 1\right). \tag{26}$$

Taking the derivative with respect to $F_j^B(\boldsymbol{r})$ and setting to 0:

$$F_j^{B*}(\boldsymbol{r}) = \frac{P_{(j)}^A P_{(j)}^\Phi(\boldsymbol{r})}{\lambda}. \tag{27}$$

Solve for $\lambda$ by the constraint $\sum_{j=1}^K F_j^B(\boldsymbol{r}) = 1$ to obtain equation 24. □

**Theorem 3.1.** *For a fixed $t$, there exists a pair $\Phi$ and $F^B$ that satisfies the equilibrium in equation 15. The equilibrium holds if and only if $\Phi$ satisfies equation 9.*

*Proof.* The objective for representation network $\Phi$ in equation 15 with $P^\Phi(\boldsymbol{r}) = P(\Phi(\boldsymbol{h}))$ is

$$\Phi^* = \arg\min_\Phi \sum_{j=1}^K \int F_j^B(\boldsymbol{r}) \log\left(F_j^B(\boldsymbol{r})\right) P^\Phi(\boldsymbol{r}) d\boldsymbol{r}. \tag{28}$$

Use the optimal predictor obtained from Lemma C.1 and $P^\Phi(\boldsymbol{r}) = \sum_{j=1}^K P_{(j)}^A P_{(j)}^\Phi(\boldsymbol{r})$:

$$\Phi^* = \arg\min_\Phi \sum_{j=1}^K \int \frac{P_{(j)}^A P_{(j)}^\Phi(\boldsymbol{r})}{\sum_{k=1}^K P_{(k)}^A P_{(k)}^\Phi(\boldsymbol{r})} \log\left(\frac{P_{(j)}^A P_{(j)}^\Phi(\boldsymbol{r})}{\sum_{k=1}^K P_{(k)}^A P_{(k)}^\Phi(\boldsymbol{r})}\right) \sum_{k=1}^K P_{(k)}^A P_{(k)}^\Phi(\boldsymbol{r}) d\boldsymbol{r} \tag{29}$$

$$= \arg\min_{\Phi} \sum_{j=1}^{K} \int P^A_{(j)} P^{\Phi}_{(j)}(\boldsymbol{r}) \log \frac{P^A_{(j)} P^{\Phi}_{(j)}(\boldsymbol{r})}{\sum_{k=1}^{K} P^A_{(k)} P^{\Phi}_{(k)}(\boldsymbol{r})} d\boldsymbol{r} \tag{30}$$

$$= \arg\min_{\Phi} \left( \sum_{j=1}^{K} \int P^A_{(j)} P^{\Phi}_{(j)}(\boldsymbol{r}) \log \frac{P^{\Phi}_{(j)}(\boldsymbol{r})}{\sum_{k=1}^{K} P^A_{(k)} P^{\Phi}_{(k)}(\boldsymbol{r})} d\boldsymbol{r} + \sum_{j=1}^{K} \int P^A_{(j)} P^{\Phi}_{(j)}(\boldsymbol{r}) \log P^A_{(j)} d\boldsymbol{r} \right) \tag{31}$$

$$= \arg\min_{\Phi} \left( \sum_{j=1}^{K} P^A_{(j)} \int P^{\Phi}_{(j)}(\boldsymbol{r}) \log \frac{P^{\Phi}_{(j)}(\boldsymbol{r})}{\sum_{k=1}^{K} P^A_{(k)} P^{\Phi}_{(k)}(\boldsymbol{r})} d\boldsymbol{r} + \underbrace{\sum_{j=1}^{K} P^A_{(j)} \log P^A_{(j)} \underbrace{\int P^{\Phi}_{(j)}(\boldsymbol{r}) d\boldsymbol{r}}_{=1}}_{=C} \right) \tag{32}$$

$$= \arg\min_{\Phi} \sum_{j=1}^{K} P^A_{(j)} \int P^{\Phi}_{(j)}(\boldsymbol{r}) \log \frac{P^{\Phi}_{(j)}(\boldsymbol{r})}{\sum_{k=1}^{K} P^A_{(k)} P^{\Phi}_{(k)}(\boldsymbol{r})} d\boldsymbol{r} \tag{33}$$

$$= \arg\min_{\Phi} \sum_{j=1}^{K} P^A_{(j)} \, \mathrm{KL} \left( P^{\Phi}_{(j)} \,||\, \sum_{k=1}^{K} P^A_{(k)} P^{\Phi}_{(k)} \right) \tag{34}$$

$$= \arg\min_{\Phi} \mathrm{JS}_{P^A_{(1)},...,P^A_{(K)}} \left( P^{\Phi}_{(0)}, ..., P^{\Phi}_{(K)} \right) \tag{35}$$

where $\mathrm{JS}_{P^A_1,...,P^A_{(k)}} \left( P^{\Phi}_{(1)}, ..., P^{\Phi}_{(k)} \right)$ is the generalized Jensen-Shannon divergence with weight $P^A_{(1)}, ..., P^A_{(k)}$ for distributions $P^{\Phi}_{(1)}, ..., P^{\Phi}_{(k)}$. The divergence is non-negative and equals zero if and only if all distributions are identical, which satisfy equation 9.

□

## D PROOF OF THEOREM 3.2

We restate (Shalit et al., 2017, Lemma A4) with a specific IPM metric, which is the total variation distance between two distributions $P$ and $Q$, denoted as $\mathrm{TV}\left(P \,||\, Q\right) = \frac{1}{2} \int |P(\boldsymbol{x}) - Q(\boldsymbol{x})| d\boldsymbol{x}$. Note that (Shalit et al., 2017, Lemma A4) can be extended to any categorical treatment pair $(a^{(i)}, a^{(j)})$ and is not limited to binary treatments. For every categorical treatment pair $(a^{(i)}, a^{(j)})$, we define the $ij$-factual, $ij$-counterfactual losses as follow:

$$\epsilon^{\mathrm{F}[ij]}(G^Y, \Phi) = \int \sum_{a \in \{a^{(i)}, a^{(j)}\}} \bar{\mathcal{L}}^{G^Y, \Phi}(\boldsymbol{h}, a) P(\boldsymbol{H} = \boldsymbol{h}, A = a) d\boldsymbol{h} \tag{36}$$

$$\epsilon^{\mathrm{CF}[ij]}(G^Y, \Phi) = \int \sum_{a \in \{a^{(i)}, a^{(j)}\}} \bar{\mathcal{L}}^{G^Y, \Phi}(\boldsymbol{h}, a) P(\boldsymbol{H} = \boldsymbol{h}, A = a') d\boldsymbol{h} \tag{37}$$

where $a' = a^{(i)}$ if $a = a^{(j)}$ and $a' = a^{(j)}$ if $a = a^{(i)}$.

**Lemma D.1.** *Let $\Phi : \mathrm{dom}\left(\boldsymbol{H}\right) \to \mathrm{dom}\left(\boldsymbol{R}\right)$ be an invertible representation. Consider the supremum $S^{[ij]} = \sup_{\boldsymbol{H}, \{a^{(i)}, a^{(j)}\}} \left| \bar{\mathcal{L}}^{G^Y, \Phi}(\boldsymbol{h}, a) \right|$, we then have:*

$$\epsilon^{\mathrm{CF}[ij]}(G^Y, \Phi) \leq P^A_{(i)} \epsilon^{\mathrm{F}}_{(j)}(G^Y, \Phi) + P^A_{(j)} \epsilon^{\mathrm{F}}_{(i)}(G^Y, \Phi) + 2 \left( P^A_{(i)} + P^A_{(j)} \right) S^{[ij]} \cdot \mathrm{TV} \left( P^{\Phi}_{(i)} \,||\, P^{\Phi}_{(j)} \right). \tag{38}$$

*Proof.* Let:

$$\epsilon^{\mathrm{CF}[ij]}_{(i)}(G^Y, \Phi) = \int \bar{\mathcal{L}}^{G^Y, \Phi}(\boldsymbol{h}, a^{(i)}) P(\boldsymbol{H} = \boldsymbol{h} | A = a^{(j)}) d\boldsymbol{h} \tag{39}$$

$$\epsilon^{\mathrm{CF}[ij]}_{(j)}(G^Y, \Phi) = \int \bar{\mathcal{L}}^{G^Y, \Phi}(\boldsymbol{h}, a^{(j)}) P(\boldsymbol{H} = \boldsymbol{h} | A = a^{(i)}) d\boldsymbol{h} \tag{40}$$

From (Shalit et al., 2017, Lemma A3), then

$$\epsilon^{\mathrm{F}[ij]}(G^Y,\Phi) = P^A_{(j)}\epsilon^{\mathrm{F}}_{(j)}(G^Y,\Phi) + P^A_{(i)}\epsilon^{\mathrm{F}}_{(i)}(G^Y,\Phi) \tag{41}$$

$$\epsilon^{\mathrm{CF}[ij]}(G^Y,\Phi) = P^A_{(i)}\epsilon^{\mathrm{CF}[ij]}_{(j)}(G^Y,\Phi) + P^A_{(j)}\epsilon^{\mathrm{CF}[ij]}_{(i)}(G^Y,\Phi). \tag{42}$$

Let $P_{(a)}(\boldsymbol{h}) = P(\boldsymbol{H}=\boldsymbol{h}|A=a)$, then:

$$\epsilon^{\mathrm{CF}[ij]}(G^Y,\Phi) - \left[P^A_{(i)}\epsilon^{\mathrm{F}}_{(j)}(G^Y,\Phi) + P^A_{(j)}\epsilon^{\mathrm{F}}_{(i)}(G^Y,\Phi)\right] \tag{43}$$

$$= \left[P^A_{(i)}\epsilon^{\mathrm{CF}[ij]}_{(j)}(G^Y,\Phi) + P^A_{(j)}\epsilon^{\mathrm{CF}[ij]}_{(i)}(G^Y,\Phi)\right] - \left[P^A_{(i)}\epsilon^{\mathrm{F}}_{(j)}(G^Y,\Phi) + P^A_{(j)}\epsilon^{\mathrm{F}}_{(i)}(G^Y,\Phi)\right] \tag{44}$$

$$= P^A_{(i)}\left[\epsilon^{\mathrm{CF}[ij]}_{(j)}(G^Y,\Phi) - \epsilon^{\mathrm{F}}_{(j)}(G^Y,\Phi)\right] + P^A_{(j)}\left[\epsilon^{\mathrm{CF}[ij]}_{(i)}(G^Y,\Phi) - \epsilon^{\mathrm{F}}_{(i)}(G^Y,\Phi)\right] \tag{45}$$

$$= P^A_{(i)}\int \bar{\mathcal{L}}^{G^Y,\Phi}(\boldsymbol{h},a^{(j)})\big(P_{(i)}(\boldsymbol{h}) - P_{(j)}(\boldsymbol{h})\big)d\boldsymbol{h} + P^A_{(j)}\int \bar{\mathcal{L}}^{G^Y,\Phi}(\boldsymbol{h},a^{(i)})\big(P_{(j)}(\boldsymbol{h}) - P_{(i)}(\boldsymbol{h})\big)d\boldsymbol{h} \tag{46}$$

$$\leq P^A_{(i)}\sup_{\boldsymbol{H}}\left|\bar{\mathcal{L}}^{G^Y,\Phi}(\boldsymbol{h},a^{(j)})\right|\int \left|P_{(i)}(\boldsymbol{h}) - P_{(j)}(\boldsymbol{h})\right|d\boldsymbol{h} + P^A_{(j)}\sup_{\boldsymbol{H}}\left|\bar{\mathcal{L}}^{G^Y,\Phi}(\boldsymbol{h},a^{(i)})\right|\int \left|P_{(j)}(\boldsymbol{h}) - P_{(i)}(\boldsymbol{h})\right|d\boldsymbol{h} \tag{47}$$

$$\leq P^A_{(i)}S\int \left|P_{(i)}(\boldsymbol{h}) - P_{(j)}(\boldsymbol{h})\right|d\boldsymbol{h} + P^A_{(j)}S\int_{\mathrm{dom}(\boldsymbol{H})} \left|P_{(j)}(\boldsymbol{h}) - P_{(i)}(\boldsymbol{h})\right|d\boldsymbol{h} \tag{48}$$

$$= (P^A_{(i)} + P^A_{(j)})\, S\int \left|P_{(i)}(\boldsymbol{h}) - P_{(j)}(\boldsymbol{h})\right|d\boldsymbol{h} \tag{49}$$

$$= (P^A_{(i)} + P^A_{(j)})\, S\int \left|P^\Phi_{(i)}(\boldsymbol{r}) - P^\Phi_{(j)}(\boldsymbol{r})\right|d\boldsymbol{r} \tag{50}$$

$$= (P^A_{(i)} + P^A_{(j)})\, 2S\,\mathrm{TV}\left(P^\Phi_{(i)} \,||\, P^\Phi_{(j)}\right) \tag{51}$$

where equation 47 is by the Hölder's inequality and equation 50 is by the change of variables formula. □

**Lemma D.2.** *Consider the generalized Jensen-Shannon divergence between $K$ distributions $P^\Phi_{(1)},...,P^\Phi_{(K)}$ with weighting factors $\boldsymbol{\pi}\in\mathbb{R}^K$, $\boldsymbol{\pi}>0$, $\sum_{a=1}^K \pi_a = 1$ and $M = \sum_{a=1}^K \pi_a P^\Phi_{(a)}$, then:*

$$\sum_{a=1}^K \mathrm{TV}\left(P^\Phi_{(a)} \,||\, M\right) \leq \sqrt{\sum_{a=1}^K \frac{1}{\pi_a}}\sqrt{\mathrm{JS}_{\boldsymbol{\pi}}\left(P^\Phi_{(1)},...,P^\Phi_{(K)}\right)/2} \tag{52}$$

$$\sum_{a=1}^K P^A_{(a)}\,\mathrm{TV}\left(P^\Phi_{(a)} \,||\, M\right) \leq \sqrt{\sum_{a=1}^K \frac{{P^A_{(a)}}^2}{\pi_a}}\sqrt{\mathrm{JS}_{\boldsymbol{\pi}}\left(P^\Phi_{(1)},...,P^\Phi_{(K)}\right)/2} \tag{53}$$

*Proof.* Use the Pinsker's inequality Tsybakov (2009) and Cauchy-Schwarz inequality to obtain:

$$\left(\sum_{a=1}^K \mathrm{TV}\left(P^\Phi_{(a)} \,||\, M\right)\right)^2 \leq \left(\sum_{a=1}^K \sqrt{\mathrm{KL}\left(P^\Phi_{(a)} \,||\, M\right)/2}\right)^2 \tag{54}$$

$$= \left(\sum_{a=1}^K \frac{1}{\sqrt{\pi_a}}\sqrt{\pi_a\,\mathrm{KL}\left(P^\Phi_{(a)} \,||\, M\right)/2}\right)^2 \leq \left(\sum_{a=1}^K \frac{1}{\pi_a}\right)\left(\sum_{a=1}^K \pi_a\,\mathrm{KL}\left(P^\Phi_{(a)} \,||\, M\right)/2\right) \tag{55}$$

$$\left(\sum_{a=1}^K P^A_{(a)}\,\mathrm{TV}\left(P^\Phi_{(a)} \,||\, M\right)\right)^2 \leq \left(\sum_{a=1}^K P^A_{(a)}\sqrt{\mathrm{KL}\left(P^\Phi_{(a)} \,||\, M\right)/2}\right)^2 \tag{56}$$

$$= \left(\sum_{a=1}^K \frac{P^A_{(a)}}{\sqrt{\pi_a}}\sqrt{\pi_a\,\mathrm{KL}\left(P^\Phi_{(a)} \,||\, M\right)/2}\right)^2 \leq \left(\sum_{a=1}^K \frac{{P^A_{(a)}}^2}{\pi_a}\right)\left(\sum_{a=1}^K \pi_a\,\mathrm{KL}\left(P^\Phi_{(a)} \,||\, M\right)/2\right) \tag{57}$$

Equality occurs when all distributions are identical. □

**Theorem 3.2.** *Let $\Phi$ be an invertible representation function and $G^Y$ be a hypothesis. Let $S = \sup_{\boldsymbol{H},A} \left|\bar{\mathcal{L}}^{G^Y,\Phi}(\boldsymbol{h}, a)\right|$. Consider the weighting factors $\boldsymbol{\pi} \in \mathbb{R}^K$, $\boldsymbol{\pi} > 0$ , and $\sum_{a=1}^K \pi_a = 1$ and let $M = \left(\sqrt{\sum_{a=1}^K \frac{1}{\pi_a}} + (K-2)\sqrt{\sum_{a=1}^K \frac{{P_{(a)}^A}^2}{\pi_a}}\right)$, we then have:*

$$\epsilon^{\mathrm{F}}(G^Y, \Phi) + \epsilon^{\mathrm{CF}}(G^Y, \Phi) \leq \sum_{a=1}^K \epsilon_{(a)}^{\mathrm{F}}(G^Y, \Phi) + SM\sqrt{2\,\mathrm{JS}_{\boldsymbol{\pi}}\left(P_{(1)}^{\Phi}, \ldots, P_{(K)}^{\Phi}\right)}. \tag{20}$$

*Proof.* By applying Lemma A3 of Shalit et al. (2017) and Lemma D.1:

$$\epsilon^{\mathrm{F}[ij]}(G^Y, \Phi) + \epsilon^{\mathrm{CF}[ij]}(G^Y, \Phi) \tag{58}$$

$$\leq P_{(j)}^A \epsilon_{(j)}^{\mathrm{F}}(G^Y, \Phi) + P_{(i)}^A \epsilon_{(i)}^{\mathrm{F}}(G^Y, \Phi) + P_{(i)}^A \epsilon_{(j)}^{\mathrm{F}}(G^Y, \Phi) + P_{(j)}^A \epsilon_{(i)}^{\mathrm{F}}(G^Y, \Phi) + (P_{(i)}^A + P_{(j)}^A)\, 2S^{[ij]}\,\mathrm{TV}\left(P_{(i)}^{\Phi} \,||\, P_{(j)}^{\Phi}\right) \tag{59}$$

$$= (P_{(i)}^A + P_{(j)}^A)\left(\epsilon_{(i)}^{\mathrm{F}}(G^Y, \Phi) + \epsilon_{(j)}^{\mathrm{F}}(G^Y, \Phi) + 2S^{[ij]}\,\mathrm{TV}\left(P_{(i)}^{\Phi} \,||\, P_{(j)}^{\Phi}\right)\right). \tag{60}$$

This is the bound for any categorical treatment pair $(a^{(i)}, a^{(j)})$. We perform a pairwise sum for all possible pairs, then apply the triangle inequality on the total variation distance:

$$\sum_{1\leq i<j\leq K} \epsilon^{\mathrm{F}[ij]}(G^Y, \Phi) + \epsilon^{\mathrm{CF}[ij]}(G^Y, \Phi) \tag{61}$$

$$\leq \sum_{1\leq i<j\leq K} (P_{(i)}^A + P_{(j)}^A)\left(\epsilon_{(i)}^{\mathrm{F}}(G^Y, \Phi) + \epsilon_{(j)}^{\mathrm{F}}(G^Y, \Phi) + 2S^{[ij]}\,\mathrm{TV}\left(P_{(i)}^{\Phi} \,||\, P_{(j)}^{\Phi}\right)\right) \tag{62}$$

$$\leq \sum_{1\leq i<j\leq K} (P_{(i)}^A + P_{(j)}^A)\left(\epsilon_{(i)}^{\mathrm{F}}(G^Y, \Phi) + \epsilon_{(j)}^{\mathrm{F}}(G^Y, \Phi) + 2S^{[ij]}\,\mathrm{TV}\left(P_{(i)}^{\Phi} \,||\, P_{(j)}^{\Phi}\right)\right) \tag{63}$$

$$\leq \sum_{1\leq i<j\leq K} (P_{(i)}^A + P_{(j)}^A)\left(\epsilon_{(i)}^{\mathrm{F}}(G^Y, \Phi) + \epsilon_{(j)}^{\mathrm{F}}(G^Y, \Phi) + 2S\,\mathrm{TV}\left(P_{(i)}^{\Phi} \,||\, M\right) + 2S\,\mathrm{TV}\left(P_{(j)}^{\Phi} \,||\, M\right)\right) \tag{64}$$

We can show that $\sum_{1\leq i<j\leq K}(P_{(i)}^A + P_{(j)}^A)(f_i + f_j) = \sum_{i=1}^K f_i + (K-2)\sum_{i=1}^K P_{(i)}^A f_i$, then

$$(K-1)\epsilon^{\mathrm{F}}(G^Y, \Phi) + \epsilon^{\mathrm{CF}}(G^Y, \Phi) \tag{65}$$

$$\leq \sum_{a=1}^K \left(\epsilon_{(a)}^{\mathrm{F}}(G^Y, \Phi) + 2S\,\mathrm{TV}\left(P_{(a)}^{\Phi} \,||\, M\right)\right) + (K-2)\sum_{a=1}^K P_{(a)}^A \left(\epsilon_{(a)}^{\mathrm{F}}(G^Y, \Phi) + 2S\,\mathrm{TV}\left(P_{(a)}^{\Phi} \,||\, M\right)\right) \tag{66}$$

$$\leq \sum_{a=1}^K \epsilon_{(a)}^{\mathrm{F}}(G^Y, \Phi) + 2S\sqrt{\sum_{a=1}^K \frac{1}{\pi_a}}\sqrt{\mathrm{JS}_{\boldsymbol{\pi}}\left(P_{(1)}^{\Phi}, ..., P_{(K)}^{\Phi}\right)/2} \tag{67}$$

$$+ (K-2)\epsilon^{\mathrm{F}}(G^Y, \Phi) + (K-2)2S\sqrt{\sum_{a=1}^K \frac{{P_{(a)}^A}^2}{\pi_a}}\sqrt{\mathrm{JS}_{\boldsymbol{\pi}}\left(P_{(1)}^{\Phi}, ..., P_{(K)}^{\Phi}\right)/2} \tag{68}$$

Rearrange the terms to obtain the final result.

□

# E Non-small Cell Lung Cancer PK-PD Simulation

## E.1 Simulation details

Similar to prior works (Lim et al., 2018; Bica et al., 2019; Melnychuk et al., 2022), we evaluate CAETC on a pharmacokinetics-pharmacodynamics model of non-small cell lung cancer (Geng et al., 2017).

The model simulates the tumor volume over time under chemotherapy and radiotherapy. Parameters in the form of $\zeta^{\text{param}}$ are either given or sampled from the distribution in Geng et al. (2017). Consider the discrete-time model of volume of the tumor $Y_t$, the chemotherapy drug concentration $C_t$, and the radiotherapy dose $D_t$:

$$Y_{t+1} = \left( \underbrace{1 + \zeta^{\rho} \log \frac{\zeta^{K}}{Y_t}}_{\text{tumor growth}} + \underbrace{\epsilon_t}_{\text{noise}} - \underbrace{\zeta^{\beta_c} C_{t+1}}_{\text{chemotherapy}} - \underbrace{\left( \zeta^{\alpha} D_{t+1} + \zeta^{\beta} D_{t+1}^2 \right)}_{\text{radiotherapy}} \right) Y_t \tag{69}$$

where one unit of time is one day. A noise term $\epsilon_t \sim \text{Normal}(0, 0.01^2)$ is added to account for randomness in tumor growth. The tumor is assumed to be spherical.

The chemotherapy concentration is modeled as an exponential decay with a half-life of one day as $C_t = \tilde{C}_t + C_{t-1}/2$. The dosage if used of chemotherapy $\tilde{C}_t$ and radiotherapy $D_t$ are 5.0 mg/m$^3$ of Vinblastine and 2.0 Gy fractions, respectively. Heterogeneous static features $V$ are generated by augmenting the prior means of $\zeta^{\alpha}, \zeta^{\beta}$. The patients are divided into three groups $V \sim \text{Uniform}\{1, 2, 3\}$. The augmented prior means $\zeta^{\mu'_{\alpha}}, \zeta^{\mu'_{\beta_C}}$ are then defined as:

$$\zeta^{\mu'_{\beta_C}} = \begin{cases} 1.1\zeta^{\mu_{\beta_C}} & \text{if } V = 3 \\ \zeta^{\mu_{\beta_C}} & \text{otherwise} \end{cases} \qquad \zeta^{\mu'_{\alpha}} = \begin{cases} 1.1\zeta^{\mu'_{\alpha}} & \text{if } V = 1 \\ \zeta^{\mu'_{\alpha}} & \text{otherwise} \end{cases} \tag{70}$$

Time-varying confounding is introduced by modeling treatment assignment as a Bernoulli variable:

$$A_{t+1}^{\text{chemo}}, A_{t+1}^{\text{radio}} \sim \text{Bernoulli}\left( \text{sigmoid}\left( \frac{\gamma}{\text{diam}(Y_{\max})} \left( \overline{\text{diam}(Y_{>t-15})} - \frac{\text{diam}(Y_{\max})}{2} \right) \right) \right) \tag{71}$$

where $\text{diam}(Y_{\max})$ is the maximum tumor diameter (13 cm), $\overline{\text{diam}(Y_{>t-15})}$ is the average tumor diameter over the last 15 days. The value of $\gamma$ directly affects the strength of time-dependent confounding, i.e., the larger the value of $\gamma$, the larger the time-dependent confounding.

Each tumor has an initial stage proportional to the value in Detterbeck & Gibson (2008). Given the initial stage, the initial volume $Y_1$ is correspondingly sampled from the stage distribution as in Geng et al. (2017). For treatment initial value, $C_1 = \tilde{C}_1 = 0$ mg/m$^3$ and $D_1 = 0$ Gy fractions.

Trajectories are measured until one of the following conditions happens: 1) The tumor diameter reaches $\text{diam}(Y_{\max}) = 13$ cm. 2) Patient recovers with probability $\exp(-Y_t \zeta^{\eta})$ where $\zeta^{\eta}$ is the tumor cell density as specified in Geng et al. (2017). 3) Termination after reaching the maximum time step $T = 60$ days.

### E.2 Experimental details

We generate 10000, 1000, and 1000 patients for training, validation, and testing. For training and testing, we simulate the full factual trajectories. All methods are tested on two settings: random trajectories and no confounding. We set $\tau = 5$, i.e., 5-step ahead prediction.

**Random trajectories**. At every time step $t$, we evaluate the methods by predicting potential horizons given the history $\mathcal{H}_t$. Therefore, for each $t$, we need to simulate an exponential of $K^{\tau}$ outcomes for a prediction horizon of $\tau$, which is prohibitively expensive. We can randomly sample a fixed subset of $k$ outcomes for evaluation, which includes both factual and counterfactual outcomes. This setting is similar to the setting in Melnychuk et al. (2022). We set $k = 1$ and simulate 1000 patients, resulting in 1000 test horizons per time step $t$. Since $T = 60$ and $\tau = 5$, we obtain 55 steps for history, resulting in $55 \times 1000 = 55000$ test horizons.

**No confounding**. At every time step $t$, we evaluate the methods by predicting factual horizons given the history $\mathcal{H}_t$. As the parameter $\gamma$ can be controlled, we evaluate the model on a non-confounded ($\gamma = 0$) test set. This setting is more difficult than random trajectories. In the random trajectories, the input history resembles the confounded training set distribution, while the test horizons do not. In the no-confounding setting, neither the history nor the test horizon resembles the training set distribution. For this strategy, we would only need to generate the factual outcomes for testing. The setting is similar to Lim et al. (2018). We use the trajectories of 1000 patients.

## F MIMIC-III Dataset

We generate semi-synthetic data based on the MIMIC-III dataset Johnson et al. (2016) using a similar protocol to Melnychuk et al. (2022) with some minor modifications to simplify the simulation. Similarly, the real-world data selection also follows Melnychuk et al. (2022). We used the MIMIC-Extract preprocessing pipeline Wang et al. (2020) to obtain the hourly aggregated data for the intensive care unit (ICU). All continuous time-varying covariates are further processed using forward filling, backward filling, and normalization.

From the hourly data, we extract 25 time-varying covariates $\boldsymbol{X}_t$ and 3 static covariates $\boldsymbol{V}$ as in Table 4. The static covariates are one-hot encoded into a 44-dimensional feature vector. Patients with less than 20 hours of ICU stay are removed.

Table 4: Selected time-varying covariates and static covariates from the MIMIC-Extract processed data.

| **Time-varying covariates** |
|---|
| heart rate |
| red blood cell count |
| sodium |
| mean blood pressure |
| systemic vascular resistance |
| glucose |
| chloride urine |
| glascow coma scale total |
| hematocrit |
| positive end-expiratory pressure set |
| respiratory rate |
| prothrombin time pt |
| cholesterol |
| hemoglobin |
| creatinine |
| blood urea nitrogen |
| bicarbonate |
| calcium ionized |
| partial pressure of carbon dioxide |
| magnesium |
| anion gap |
| phosphorous |
| venous pvo2 |
| platelets |
| calcium urine |
| **Static covariates** |
| gender |
| ethinicity |
| age |

### F.1 Semi-synthetic simulation details

The ICU maximum stay is cut off at 100 hours, creating time-varying covariates $\boldsymbol{X}_t$ with a maximum time step of $T = 100$.

We generate $N^Y = 2$ outcomes. For an outcome $m$, denoted as $Y_{i,t}^{(m)}$, the simulation first creates a non-treated outcomes $\tilde{Y}_{i,t}^{(m)}$. The non-treated outcome consists of an endogenous dependency by a combination of cubic spline and Gaussian process. For every outcome simulated, we randomly sampled a subset of 10 time-varying covariates $\boldsymbol{X}_{i,t}^{(m)}$ to be the exogenous dependency. Assuming the outcome $m$ is affected by a set of treatment $\boldsymbol{A}_{i,t}^{(m)}$ with treatment effect $\boldsymbol{E}_{i,t}^{(m)}$ (defined later), we then have:

$$\tilde{Y}_{i,t}^{(m)} = \underbrace{u_i^{(m)}}_{\text{initial value}} + \underbrace{\alpha^b b_i(t) + \alpha^g g_i^{(m)}(t)}_{\text{endogeneous}} + \underbrace{\alpha^f f^{(m)}(\boldsymbol{X}_{i,t}^{(m)})}_{\text{exogeneous}} + \underbrace{\epsilon_{i,t}^{(m)}}_{\text{noise}} \tag{72}$$

$$Y_{i,t}^{(m)} = \tilde{Y}_{i,t}^{(m)} + \boldsymbol{E}_{i,t}^{(m)\intercal}\mathbf{1} \tag{73}$$

where $\alpha^b$, $\alpha^g$, and $\alpha^f$ are weight for each component. The Gaussian process instance $g_i^m(t) \sim \mathcal{GP}(0, \text{Matern}_{\nu=2.5})$ represent smooth local trend. The spline $b(\cdot) \sim \text{Uniform}\{b^{\text{stable}}(\cdot), b^{\text{fast decline}}(\cdot), b^{\text{slow decline}}(\cdot), b^{\text{fast increase}}(\cdot), b^{\text{slow increase}}(\cdot)\}$ is a cubic spline sampled uniformly from a set of five cubic splines representing different global trends. Exogenous dependency is introduced by a non-linear function of the selected covariates, constructed by random Fourier features approximating an RBF kernel $f^{(m)}(\boldsymbol{X}_{i,t}^{(m)}) = \phi(\boldsymbol{X}_{i,t}^{(m)})^\intercal \boldsymbol{w}$ as in Hensman et al. (2018) with $\boldsymbol{w} \sim \text{Normal}(0, \sigma_w^2 \boldsymbol{I})$. A noise term is added to account for randomness $\epsilon_{i,t}^{(m)} \sim \text{Normal}(0, 0.05^2)$.

We generate $N^A = 3$ treatments. The confounded treatment $l$, denoted $A_{i,t}^{(l)}$ and its corresponding treatment effect $E_{i,t}^{(l)}$, is generated as:

$$U_{i,t}^{(l)} = \tanh\left(\overline{\boldsymbol{Y}_{i,>t-w^{(l)}}^{(l)}} + \gamma^X f^{(l)}(\boldsymbol{X}_{i,t})\right) \tag{74}$$

$$A_{i,t}^{(l)} \sim \text{Bernoulli}\left(\text{sigmoid}\left(\gamma^Y U_{i,t}^{(l)}\right)\right) \tag{75}$$

$$E_{i,t}^{(l)} = \sum_{v=0}^{w^{(l)}-1} \frac{\alpha^{(l)} U_{i,t}^{(l)} + \beta^{(l)}}{2^{v/5}} \tag{76}$$

where $\gamma^Y$ and $\gamma^X$ are confounding parameters. $U_{i,t}^{(l)}$ is the dependency on the past outcomes and covariates. $\overline{\boldsymbol{Y}_{i,>t-w^{(l)}}^{(l)}}$ is the average outcomes over the last $w^{(l)}$ hours. Note that $\boldsymbol{Y}_{i,t}^{(l)}$ is a subset of treated outcomes that confound with the treatment. Similar to the synthetic outcome, we also sampled a subset of 10 time-varying covariates $\boldsymbol{X}_{i,t}^{(l)}$ to be the confounding covariates for each treatment. $f^{(l)}$ is a non-linear function similar to $f^{(m)}$. The treatment assignment $A^{(l)}$ is modeled as a Bernoulli

random variable that depends on $U_{i,t}^{(l)}$. The treatment effect $E_{i,t}^{(l)}$ is just the sum of all treatments used within $\omega^{(l)}$ hours with scaling factor $\alpha^{(l)}$ and bias $\beta^{(l)}$. All treatments have maximum effect right after use but decay exponentially with a half-life of 5 hours.

We can observe that each outcome $Y_{i,t}^{(m)}$ is affected by multiple treatment $\boldsymbol{A}_{i,t}^{(m)}$ and each treatment $A_{i,t}^{(l)}$ is confounded with multiple outcomes $\boldsymbol{Y}_{i,t}^{(l)}$, creating a complex dynamic.

### F.2 Semi-synthetic experiment details

We simulate 2000 and 200 patients for training and validation. For training and testing, we simulate the full factual trajectories. For testing, we use the random trajectories strategy similar to Appendix E.2. We set $\tau = 10$, i.e., 10-step ahead prediction.

### F.3 Real-world data details

For real-world data, we extract two binary treatments, including `vaso` (vasopressor $A_t^{\text{vaso}}$) and `vent` (mechanical ventilation $A_t^{\text{vent}}$), and two outcomes $\boldsymbol{Y}_t$, including `diastolic blood pressure` and `oxygen saturation`. The ICU maximum stay is cut off at 60 hours, creating trajectories with a maximum time step of $T = 60$.

### F.4 Real-world experiment details

We select a subset of 5000 and 500 patients for training and validation. At every time step, we evaluate the methods by predicting the factual horizon $\tau$ given the history $\mathcal{H}_t$. We select a different subset of 500 patients for testing. We set $\tau = 5$, i.e., 5-step ahead prediction.

## G Baseline Methods

We select the baselines that use similar settings and assumptions to our work that also publicly released the source code for reproducibility, including RMSN, CRN, CT, and CCPC. Specifically, CRN, CT, and CCPC are highly relevant methods to CAETC as all these methods utilize adversarial representation learning to learn a treatment-invariant representation.

### G.1 Common structure

Table 5

(a) Architecture for $F^Y$

| **Input**: $[\Phi(\mathcal{H})_t, A_{t+1}]$ or $\Phi(\mathcal{H})_t$ or $F^C(\Phi(\mathcal{H})_t, A_{t+1})$ |
|---|
| Linear Layer |
| ELU Activation |
| Linear Layer |
| **Output**: $\hat{\boldsymbol{Y}}_{t+1}$ or $\hat{\boldsymbol{Y}}_t$ |

(b) Architecture for $F^B$

| **Input**: $\Phi(\mathcal{H})_t$ |
|---|
| Linear |
| ELU |
| Linear |
| Softmax |
| **Output**: $\hat{A}_{t+1}$ |

To ensure a fair comparison for all methods, we use a common structure including several components similar to Fig. 2A. All methods include a representation function $\Phi$ and an outcome prediction head $F^Y$ (Table 5a). For adversarial-based methods like CAETC, CRN, and CT, an additional balancing head $F^B$ is added (Table 5b). $F^Y$ and $F^B$ share the same architectures for all methods, with the only potential difference being the input size for $F^Y$ and $F^B$. All methods are trained with AdamW optimizers Kingma & Ba (2017) with weight decay of 0.01 and the teacher forcing technique Williams & Zipser (1989). The dimension of $\Phi(\mathcal{H}_t)$ is referred to as # hidden units in the hyperparameter search table in Appendix H. # hidden units are selected so that all methods have roughly the same total parameter range.

### G.2 CAETC

**Architecture**

CAETC uses a decoder-only architecture. The representation network $\Phi$ for CAETC is parametrized by either LSTM cells Hochreiter & Schmidhuber (1997) layers or TCN residual blocks Bai et al. (2018). We refer to both as layers. Since TCN is composed of dilated causal convolutions with an exponential receptive field as a function of depth, TCN requires more layers to cover the entire history. The architecture for $\Phi$ is in Table 6c. The output dimension for both projection layers is set to # hidden units.

Table 6

(a) Architecture for $\omega^{\boldsymbol{VAY}}$

| **Input**: $[\boldsymbol{V}, A_t, \boldsymbol{Y}_t]$ |
|---|
| Linear Layer |
| **Output**: $\omega^{\boldsymbol{VAY}}([\boldsymbol{V}, A_t, \boldsymbol{Y}_t])$ |

(b) Architecture for $\omega^{\boldsymbol{X}}$

| **Input**: $\boldsymbol{X}_t$ |
|---|
| Linear Layer |
| **Output**: $\omega^{\boldsymbol{X}}(\boldsymbol{X}_t)$ |

(c) Architecture for $\Phi$ (CAETC)

| **Input**: $\mathcal{H}_t$ |
|---|
| $\omega^{\boldsymbol{X},A,\boldsymbol{Y}}$ and $\omega^{\boldsymbol{X}}$<br>LSTM (# layers) or TCN (# layers)<br>Linear Layer<br>ELU Activation |
| **Output**: $\Phi(\mathcal{H})_t$ |

For CAETC, two additional heads $F^A$ (Table 7a) and $F^X$ (Table 7b) and a the conditioning layer $F^C$ (Table 7c) are added.

Table 7

(a) Architecture for $F^A$

| **Input**: $\Phi(\mathcal{H})_t$ or $F^C(\Phi(\mathcal{H})_t, A_{t+1})$ |
|---|
| Linear Layer<br>ELU Activation<br>Linear Layer<br>Softmax |
| **Output**: $\hat{A}_t$ or $\hat{A}_{t+1}$ |

(b) Architecture for $F^X$

| **Input**: $\Phi(\mathcal{H})_t$ |
|---|
| Linear Layer<br>ELU Activation<br>Linear Layer |
| **Output**: $\hat{\boldsymbol{X}}_t$ |

(c) Architecture for $F^C$

| **Input**: $\Phi(\mathcal{H}_t)$ and $a^{(k)}$ |
|---|
| 2x Embedding for $R^{\xi}$ and $R^{\beta}$ |
| **Output**: $\Phi(\mathcal{H}_t) \odot \xi^{(k)} \oplus \beta^{(k)}$ |

**Training algorithm**

The training process of CAETC is in Appendix G.2. We set $\delta^A = 0.1$, $\delta^X = 0.1$, $\delta^E = 0.001$, and $\alpha = 0.1$ for all experiments.

### G.3 RMSN

RMSN requires the estimation of $P(A_{t+1}|A_t)$ and $P(A_{t+1}|\mathcal{H}_t)$ for IPTW, each of which is parametrized by an LSTM. They are referred to as the propensity-treatment network and the propensity-history network. RMSN uses an encoder-decoder architecture. The propensity-treatment network and propensity-history network are both composed of a representation function $\Phi$ and a treatment classification head, for which we reuse the architecture of $F^B$ (Table 5b). The encoder and decoder are both composed of a representation function $\Phi$ and an outcome prediction head $F^Y$ (Table 5a). The architecture of $\Phi$ is similar for the propensity-treatment network, the propensity-history network, the encoder, and the decoder as in Table 8. In total, RMSN requires four different individual networks for training.

Table 8: Architecture for $\Phi$ (RMSN)

| **Input**: $\{A_{t'}\}_{t'=1}^{t}$ or $\mathcal{H}_t$ |
|---|
| LSTM (# layers)<br>Linear Layer<br>ELU Activation |
| **Output**: $\Phi(\{A_{t'}\}_{t'=1}^{t})$ or $\Phi(\mathcal{H}_t)$ |

### G.4 CRN

**Algorithm 1** Training algorithm for CAETC

**Input**: Dataset $\mathcal{D} = \left\{ \{A_{i,t}, \boldsymbol{Y}_{i,t}, \boldsymbol{X}_{i,t}\}_{t=1}^{T}, \boldsymbol{V}_i \right\}_{i=1}^{N}$, hyperparameters $\delta^A$, $\delta^X$, $\delta^E$
**for** epoch $e_j = 1$ to $e_{\max}$ **do**
  Compute loss:

$$\begin{aligned}\mathcal{L}(\theta_{F^A}, \theta_{F^Y}, \theta_{F_X}, \theta_{F^C}, \theta_\Phi) = &\left[\mathcal{L}^{RY}(\theta_{F^Y}, \theta_\Phi) + \mathcal{L}^{Y}(\theta_{F^Y}, \theta_{F^C}, \theta_\Phi)\right] \\ &+ \delta^A \left[\mathcal{L}^{RA}(\theta_{F^A}, \theta_\Phi) + \mathcal{L}^{C}(\theta_{F^A}, \theta_{F^C}, \theta_\Phi)\right] \\ &+ \delta^X \mathcal{L}^{RX}(\theta_{F^X}, \theta_\Phi) \\ &+ \delta^E \mathcal{L}^{E}(\theta_\Phi)\end{aligned}$$

  Compute gradient: $\nabla_{\theta_{F^A}, \theta_{F^Y}, \theta_{F_X}, \theta_{F^C}, \theta_\Phi} [\mathcal{L}(\theta_{F^A}, \theta_{F^Y}, \theta_{F^X}, \theta_{F^C}, \theta_\Phi)]$
  Compute loss: $\delta^E \mathcal{L}^B(\theta_{F^B})$
  Compute gradient: $\nabla_{\theta_{F^B}} [\delta^E \mathcal{L}^B(\theta_{F^B})]$
  Update parameters $\theta_{F^A}, \theta_{F^Y}, \theta_{F^X}, \theta_{F^C}, \theta_\Phi$ by gradient $\nabla_{\theta_{F^A}, \theta_{F^Y}, \theta_{F_X}, \theta_{F^C}, \theta_\Phi} [\mathcal{L}(\theta_{F^A}, \theta_{F^Y}, \theta_{F^X}, \theta_{F^C}, \theta_\Phi))]$
  Update parameters $\theta_{F^B}$ by gradient $\nabla_{\theta_{F^B}} [\delta^E \mathcal{L}^B(\theta_{F^B})]$
**end for**

CRN uses an encoder-decoder architecture with a gradient reversal layer for adversarial training. Both the encoder and decoder are composed of a representation function $\Phi$ (Table 9), an outcome prediction head $F^Y$ (Table 5a), and a treatment balancing head $F^B$ (Table 5b). CRN requires a hyperparameter for adversarial training, which is similar to our $\delta^B$. We set CRN's $\delta^B = 0.1$.

Table 9: Architecture for $\Phi$ (CRN)

| **Input**: $\mathcal{H}_t$ |
|---|
| LSTM (# layers) |
| Linear Layer |
| ELU Activation |
| **Output**: $\Phi(\mathcal{H})_t$ |

### G.5 CT

CT uses a decoder-only architecture and domain confusion loss for adversarial training. The decoder is composed of a representation function $\Phi$ (Table 10), an outcome prediction head $F^Y$ (Table 5a), and a treatment balancing head $F^B$ (Table 5b). CT also requires an exponential moving average (EMA) copy of the decoder. We reuse original hyperparameters in CT for the EMA copy. CT requires a hyperparameter for adversarial training, which is similar to our $\delta^B$. We set CT's $\delta^B = 0.01$.

Table 10: Architecture for $\Phi$ (CT)

| **Input**: $\mathcal{H}_t$ |
|---|
| 3x Transformer (# layers) |
| **Output**: $\Phi(\mathcal{H})_t$ |

### G.6 CCPC

CCPC uses an encoder-decoder architecture with (1) ICLUB loss for adversarial learning. Furthermore, CCPC utilizes (2) contrastive predictive coding and (3) an InfoMax regularization term to encourage representation invertibility. To ensure a fair comparison with a similar training process to all considered methods, we depart from CCPC's original implementation where appropriate. For example, CCPC does not use teacher forcing, while every method considered uses teacher forcing. Therefore, we opt to implement CCPC with teacher forcing. Similar to CRN and RMSN, we first train a CCPC encoder using teacher forcing and objectives (1), (2), and (3). Then the encoder is frozen and serves as a feature extractor for the decoder training, which also utilizes teacher forcing and objectives (1), (2), and (3).

Both the encoder and decoder are composed of a representation function $\Phi$ (Table 11), an outcome prediction head $F^Y$ (Table 5a), a treatment balancing head $F^B$ (Table 5b), and several linear layers to serve as input projections and context projections.

Table 11: Architecture for $\Phi$ (CCPC)

| **Input**: $\mathcal{H}_t$ |
|---|
| GRU (# layers) |
| Linear Layer |
| ELU Activation |
| **Output**: $\Phi(\mathcal{H})_t$ |

### G.7 ADVERSARIAL WEIGHT $\delta^B$

For all methods that use $\delta^B$, we perform an exponential increase in the value of $\delta_e^B$ as a function of the epoch until we reach $\delta^B$.

$$\delta_{e_j}^B = \delta^B \left( \frac{2}{1 + \exp(-10 \times e_j / e_{\max})} - 1 \right) \tag{77}$$

## H HYPERPARAMETER SEARCH

See Table 12 for hyperparameter search ranges.

## I LIMITATIONS

While CAETC demonstrates significant improvement, our design also suffers from similar limitations to existing methods. All methods considered in this paper are based on the potential outcome framework, which requires a set of strong assumptions, some of which are not testable in practice. i.e., sequential ignorability. Empirically, in minimal confounding settings, CAETC does not offer a substantial performance gain. Furthermore, CAETC introduces several hyperparameters, which require more complex validation to obtain the optimal results.

## J IMPACT STATEMENT

This paper proposes a novel method for counterfactual estimation over time, which can be beneficial for sequential decision-making systems in domains such as healthcare, economics, or public policy. Improved counterfactual and treatment effect estimation can reduce the bias when learning from observational data. However, misuse or incorrect estimates can result in potentially harmful outcomes, so careful evaluation is required in sensitive applications.

Table 12: Hyperparameter search range for every method & dataset. If multiple components (e.g., encoder & decoder) are in the same cell, the hyperparameter is shared for those components (e.g., encoder and decoder have the same number of layers) for the same run. Otherwise, the hyperparameter is component-specific for the same run. We run a random grid search with 10 combinations for every method for fair comparison. The final results reported are averaged over 3 random seeds.

| Method | Components | Hyperparameters | NSCLC range | MIMIC III range (semi-synthetic & real) |
|---|---|---|---|---|
| LSTM & CRN | Encoder & Decoder | # layers | 1 | 1,2 |
| | | # hidden units | 16, 32, 48 | 32, 48, 64 |
| | | Dropout rate | 0.0, 0.1 | |
| | Encoder | # epochs | 100 | 150 |
| | | Learning rate | 0.001, 0.0001 | |
| | | Batch size | 64, 128 | |
| | Decoder | # epochs | 50 | 100 |
| | | Learning rate | 0.001, 0.0001 | |
| | | Batch size | 512, 1024 | |
| RMSN | Propensity [1] & Encoder & Decoder | # layers | 1 | 1,2 |
| | | # hidden units | 16, 32, 48 | 32, 48, 64 |
| | | Dropout rate | 0.0, 0.1 | |
| | Propensity[1] & Encoder | # epochs | 100 | 150 |
| | | Learning rate | 0.001, 0.0001 | |
| | Decoder | # epochs | 50 | 100 |
| | | Learning rate | 0.001, 0.0001 | |
| CT | | # layers | 1 | 1,2 |
| | | # hidden units | 16, 24, 32 | 18, 24, 36 |
| | | # attention heads | 2 | 2,3 |
| | | Max positional encoding | 15 | 30 |
| | | # epochs | 150 | 250 |
| | | Dropout rate | 0.0, 0.1 | |
| | | Learning rate | 0.001, 0.0001 | |
| | | Batch size | 64, 128 | |
| CAETC LSTM | | # layers | 1 | 1,2 |
| | | # hidden units | 16, 32, 48 | 32, 48, 64 |
| | | # epochs | 150 | 250 |
| | | Dropout rate | 0.0, 0.1 | |
| | | Learning rate | 0.001, 0.0001 | |
| | | Batch size | 64, 128 | |
| CAETC TCN | | # layers | 3,4, 5 | 4,5, 6 |
| | | # hidden units | 16, 24, 32 | 24, 32, 40 |
| | | Kernel size | 3, 5 | |
| | | Dilation factor | 2 | |
| | | # epochs | 150 | 250 |
| | | Dropout rate | 0.0, 0.1 | |
| | | Learning rate | 0.001, 0.0001 | |
| | | Batch size | 64, 128 | |

[1] Both the propensity-treatment & propensity-history networks are referred to as "propensity".