COSTAR: IMPROVED TEMPORAL COUNTERFACTUAL ESTIMATION WITH SELF-SUPERVISED LEARNING

Chuizheng Meng [*†]
chuizhem@usc.edu

Yihe Dong [†]
yihed@google.com

Sercan Ö. Arık [†]
soarik@google.com

Yan Liu [*†]
yanliu.cs@usc.edu

Tomas Pfister [†]
tpfister@google.com

February 13, 2024

ABSTRACT

Estimation of temporal counterfactual outcomes from observed history is crucial for decision-making in many domains such as healthcare and e-commerce, particularly when randomized controlled trials (RCTs) suffer from high cost or impracticality. For real-world datasets, modeling time-dependent confounders is challenging due to complex dynamics, long-range dependencies and both past treatments and covariates affecting the future outcomes. In this paper, we introduce Counterfactual Self-Supervised Transformer (COSTAR), a novel approach that integrates self-supervised learning for improved historical representations. We propose a component-wise contrastive loss tailored for temporal treatment outcome observations and explain its effectiveness from the view of unsupervised domain adaptation. COSTAR yields superior performance in estimation accuracy and generalization to out-of-distribution data compared to existing models, as validated by empirical results on both synthetic and real-world datasets.

1 Introduction

Accurate estimation of treatment outcomes over time conditioning on the observed history is a fundamental problem in causal analysis and decision making in various applications (Mahar et al., 2021; Ye et al., 2023; Wang et al., 2023). For example, in medical domains, doctors are interested in knowing how a patient reacts to a treatment or multi-step treatments; in e-commerce, retailers are concerned about how future sales change if adjusting the price of an item. While randomized controlled trials (RCTs) are the gold standard for treatment outcome estimation, most often than not such trials are either too costly or even impractical to conduct. Therefore, utilizing available observed data (such as electronic health records (EHRs) and historical sales) for accurate treatment outcome estimation, has drawn increasing interest in the community.

Compared to the well-studied i.i.d cases, treatment outcome estimation from time series observations not only finds more applications in the real world but also pose significant more challenges, due to the complex dynamics and the long-range dependencies in time series. Existing works along this endeavors explore various architectures with improved capacity and training strategies to alleviate time-dependent confounding[3]. Recurrent marginal structural networks (RMSNs) (Lim, 2018), counterfactual recurrent networks (CRN) (Bica et al., 2020), and G-Net (Li et al., 2021) utilize architectures based on recurrent neural networks. To mitigate time-dependent confounding, they train proposed models with inverse probability of treatment weighting (IPTW), treatment invariant representation through gradient reversal, and G-computation respectively, in addition to the factual estimation loss on observed data. Causal Transformer (CT) (Melnychuk et al., 2022) further improves capturing long-range dependencies in the observational data

[*]University of Southern California
[†]Google Cloud AI Research
Code available at https://github.com/google-research/google-research/tree/master/COSTAR
[3]We leave a more detailed review of related work in Sec. 2 of the appendix.

with a tailored transformer-based architecture and overcomes the temporal confounding with balanced representations trained through counterfactual domain confusion loss.

While existing methods achieve performance gain in empirical evaluation, they rely on the fully supervised loss of future outcomes to learn representations of history and thus suffer from its limitations. In many practical applications, we are confronted with the cold case challenge, where no or limited observations of testing time series are accessible. Figure 1 shows an example in healthcare: after training a vital sign estimator with historical health records (seen as outcomes) and drug usage (seen as treatments) sequences of patients in the youth age group, the model is asked to estimate the potential vital signs in the elderly age group after applying a treatment plan, with no or very limited observations of elderly people collected beforehand. Existing methods based on supervised learning have difficulty generalizing to different domains and handling cold cases in test time.

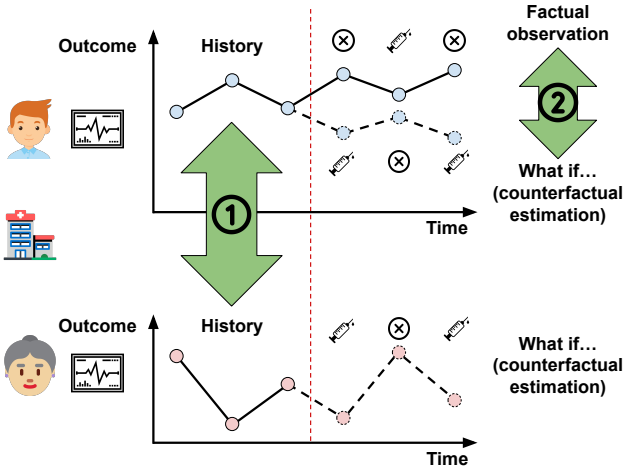


Figure 1: We illustrate the problem of treatment outcome estimation over time with an example in healthcare. We propose COSTAR as a temporal counterfactual estimator enhanced with self-supervised learning, inducing transferrability to both ① cold-start cases from unseen subpopulations and ② counterfactual outcome estimation.

In this work, we propose a paradigm shift from supervised learning to self-supervised training for temporal treatment outcome estimation. Our proposed model, **Counterfactual Self-Supervised Transformer (COSTAR)**, addresses the aforementioned limitations. To enhance the model capacity, we propose an encoder architecture composed of alternating temporal and feature-wise attention, capturing dependencies among both time steps and features. To learn expressive and transferable representations of the observed history, we refine the contrastive loss in self-supervised learning to a finer-grained level: both the entire history and each of the covariate/treatment/outcome components are contrasted when constructing the loss. Moreover, we view the counterfactual outcome estimation problem from the unsupervised domain adaptation (UDA) perspective and provide the theoretical analysis of the error bound for a counterfactual outcome estimator that gives estimation based on representations from self-supervised learning.

Our main contributions are summarized as follows:

1. We adapt self-supervised learning (SSL) together with component-wise contrastive losses tailored for temporal observations to learn more expressive representations of the history in temporal counterfactual outcome estimation.
2. We explain the boost from self-supervised learning on the counterfactual outcome estimation problem with the view of unsupervised domain adaptation (UDA) perspective and provide the theoretical analysis of the error bound of a counterfactual outcome estimator that predicts with representations from self-supervised learning.
3. Empirical results show that our proposed framework outperforms existing baselines across both synthetic and real-world datasets in both estimation accuracy and generalization. In addition, we demonstrate that the learned representations are balanced towards treatments and thus address the temporal confounding issue.

2 Related Work

Counterfactual treatment outcome estimation over time. Early works in counterfactual treatment outcome estimation were first developed for epidemiology and can be considered under 3 major groups: G-computation, marginal structural models (MSMs), and structural nested models (Robins, 1986, 1994; Robins et al., 2000; Robins & Hernan, 2008). One major shortcoming of these is that they are built on linear models and suffer from the limited model capacity when facing time series data with complex temporal dependencies. Follow-up works address the limitation in expressiveness with Bayesian non-parametric methods (Xu et al., 2016; Soleimani et al., 2017; Schulam & Saria, 2017) or more expressive deep neural networks (DNNs) such as recurrent neural networks (RNNs). For example, recurrent marginal structural networks (RMSNs) (Lim, 2018) replace the linear model in MSM with an RNN-based architecture for forecasting treatment outcomes. G-Net (Li et al., 2021) also adopts RNN instead of classical regression models in the g-computation framework. Inspired by the success of representation learning for domain adaptation

and generalization (Ganin et al., 2016; Tzeng et al., 2015), more recent works explore learning representations that are both predictive for outcome estimation and balanced regardless of the treatment bias in training data. Counterfactual recurrent network (CRN) (Bica et al., 2020) trains an RNN-based model with both the factual outcome regression loss and the gradient reversal (Ganin et al., 2016) w.r.t. the treatment prediction loss. The former loss encourages the learned representations to be predictive of outcomes while the latter encourages the representations to be homogeneous given different treatments. The joint training target leads to informative and balanced representations. With similar motivations, (Melnychuk et al., 2022) replaces the RNN-based architecture with a Transformer-based (Vaswani et al., 2017) one along with the domain confusion loss (Tzeng et al., 2015) to learn treatment-agnostic representations. Given the flexibility of the choice of model architectures, recent works extend temporal counterfactual outcome estimation to irregular time series (Seedat et al., 2022; Cao et al., 2023), temporal point process (Zhang et al., 2022b), and graph-structured spatiotemporal data (Jiang et al., 2023) with the help of (Kidger et al., 2020) and (Huang et al., 2020). While existing works claim that both predictive and balanced representations are critical in accurate counterfactual outcome estimation, we empirically find that the impact of representation balancing is inconsistent and marginal. In contrast, improving the expressiveness of representations brings more robust improvements.

Self-supervised learning of time series. Being widely studied first for computer vision tasks (He et al., 2020; Chen et al., 2020; Grill et al., 2020; Chen* et al., 2021), self-supervised learning achieves strong performance with the advantage of not relying on labeled data. Recent works (Yue et al., 2022; Tonekaboni et al., 2021; Woo et al., 2022; Zhang et al., 2022a) further generalize and adapt self-supervised learning methods for time series, including classification, forecasting, and anomaly detection tasks. However, existing works of counterfactual outcome estimation over time have neglected self-supervised learning of time series as an effective way of learning informative representations. Meanwhile, existing models for self-supervised learning of time series are not tailored for counterfactual outcome estimation. Hence we propose COSTAR to mitigate the gap.

3 Problem Formulation

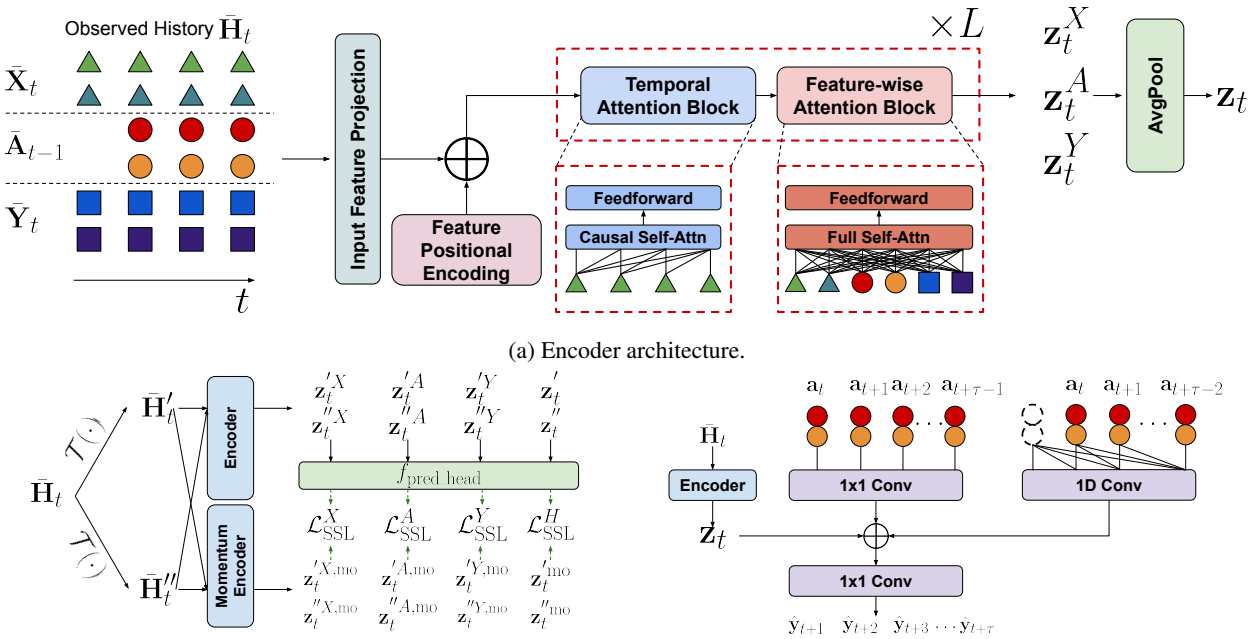
Our task is estimating the outcomes of subjects with observed history after being applied a sequence of treatments from observational data (Lim, 2018; Bica et al., 2020; Melnychuk et al., 2022). We represent the available observed dataset as $\{\{\boldsymbol{x}_t^{(i)}, \boldsymbol{a}_t^{(i)}, \boldsymbol{y}_t^{(i)}\}_{t=1}^{T^{(i)}}, \boldsymbol{v}^{(i)}\}_{i=1}^{N}$ of N independently sampled subjects, where $T^{(i)} \in \mathbb{N}^+$ denotes the length of the observed history of subject i, and $\boldsymbol{x}_t^{(i)} \in \mathbb{R}^{d_X}$, $\boldsymbol{a}_t^{(i)} \in \mathbb{R}^{d_A}$, and $\boldsymbol{y}_t^{(i)} \in \mathbb{R}^{d_Y}$ stand for the observed vector of covariates, treatments, and outcomes respectively, at time t of subject i. $\boldsymbol{v}^{(i)} \in \mathbb{R}^{d_V}$ contain all static features of subject i. We omit the subject index i in the following text for notational simplicity.

Following the potential outcomes (Splawa-Neyman et al., 1990; Rubin, 1978) framework extended to time-varying treatments and outcomes (Robins & Hernan, 2008), our target is to estimate $\mathbb{E}(\boldsymbol{y}_{t+\tau}[\bar{\boldsymbol{a}}_{t:t+\tau-1}]|\bar{\boldsymbol{H}}_t)$ for $\tau \geq 1$, where $\bar{\boldsymbol{H}}_t = (\bar{\boldsymbol{X}}_t, \bar{\boldsymbol{A}}_{t-1}, \bar{\boldsymbol{Y}}_t, \boldsymbol{V})$ is the observed history. $\bar{\boldsymbol{X}}_t = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t)$, $\bar{\boldsymbol{A}}_{t-1} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_{t-1})$, $\bar{\boldsymbol{Y}}_t = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_t)$, $\boldsymbol{V} = \boldsymbol{v}$. $\bar{\boldsymbol{a}}_{t:t+\tau-1} = (\boldsymbol{a}_t, \boldsymbol{a}_{t+1}, \ldots, \boldsymbol{a}_{t+\tau-1})$ is the sequence of the applied treatments in the future τ discrete time steps. In factual data, $\bar{\boldsymbol{H}}_t$ and $\boldsymbol{a}_t$ are correlated, leading to the treatment bias in counterfactual outcome estimation. In addition, the distribution of $\bar{\boldsymbol{H}}_t$ can also vary between training and test data: $P_{\mathcal{D}_{tr}}(h) \neq P_{\mathcal{D}}(h)$, causing the feature distribution shifts. Following the tradition in domain adaptation, we name $P_{\mathcal{D}_{tr}}(h)$ and $P_{\mathcal{D}}(h)$ the source/target domains respectively. Table 4 describes feature distribution shifts in our datasets.

To ensure the identifiability of treatment effects from observational data, we take the standard assumptions used in existing works (Bica et al., 2020; Melnychuk et al., 2022): (1) consistency, (2) positivity and (3) sequential strong ignorability (See Appendix B).

4 Counterfactual Self-Supervised Transformer

We illustrate the detailed design of our proposed Counterfactual Self-Supervised Transformer (COSTAR). Our main goal is to learn representations of observed history sequences that are informative for counterfactual treatment outcome estimation, which we achieve by tailoring both the representation encoder architecture and the self-supervised training loss. On top of the representation learning, we also propose a simple yet effective decoder for non-autoregressive outcome prediction and demonstrate improvements in both the accuracy and the speed of multi-step estimation. Fig. 2 overviews the proposed framework.



(a) Encoder architecture.

(b) Self-supervised learning of history.

(c) Non-autoregressive outcome predictor.

Figure 2: Overview of COSTAR. (a) Encoder architecture. The Temporal Attention Block applies temporal causal attention along the time dimension in parallel for each feature, while the Feature-wise Attention Block calculates full self-attention along the feature dimension in all time steps. (b) Self-supervised learning of the history representations. Positive pairs are generated by applying random transformations $\mathcal{T}(\cdot)$ on the same sample. We construct component-wise contrastive losses of historical covariates, treatments and outcomes in addition to the standard contrastive loss of the entire sequence. (c) Non-autoregressive outcome predictor architecture.

4.1 Encoder architecture

For a given sequence of the observed history $\bar{\boldsymbol{H}}_t \in \mathbb{R}^{t \times d_{\text{input}}}$ concatenated from $(\bar{\boldsymbol{X}}_t, \bar{\boldsymbol{A}}_{t-1}, \bar{\boldsymbol{Y}}_t)$ (we omit static variables $\boldsymbol{V}$ here for simplicity and leave the processing details in Sec. 4.1; $d_{\text{input}} = d_X + d_A + d_Y$), the encoder in Fig. 2a maps the entire history to representations $\{\boldsymbol{z}_t^i \in \mathbb{R}^{d_{\text{model}}}\}_{i=1}^{d_{\text{input}}}$ for each feature f_i. Then, we employ average pooling for feature-wise representations from corresponding features to get the representations of covariate, treatment, and outcome components, and all features for the representation of the entire observed history. Denote the set of covariate, treatment, and outcome variables as $\mathbf{F}_X$, $\mathbf{F}_A$, and $\mathbf{F}_Y$ respectively. We have:

$$\begin{aligned} \boldsymbol{z}_t^X &= \text{avg}(\{\boldsymbol{z}_t^i\}_{f_i \in \mathbf{F}_X}), \quad \boldsymbol{z}_t^A = \text{avg}(\{\boldsymbol{z}_t^i\}_{f_i \in \mathbf{F}_A}), \\ \boldsymbol{z}_t^Y &= \text{avg}(\{\boldsymbol{z}_t^i\}_{f_i \in \mathbf{F}_Y}), \quad \boldsymbol{z}_t = \text{avg}(\{\boldsymbol{z}_t^i\}_{i=1}^{d_{\text{input}}}). \end{aligned} \tag{1}$$

We describe the detailed design of the encoder architecture in Appendix A.

4.2 Self-supervised representation learning of the observed history

We employ pretraining for the encoder in a self-supervised way with the contrastive learning objectives $\mathcal{L}_{\text{SSL}}^X$, $\mathcal{L}_{\text{SSL}}^A$, $\mathcal{L}_{\text{SSL}}^Y$, $\mathcal{L}_{\text{SSL}}^H$ for the component representations $\boldsymbol{z}_t^X, \boldsymbol{z}_t^A, \boldsymbol{z}_t^Y$ and the overall representation $\boldsymbol{z}_t$ respectively. The overall self-supervised learning loss is given as:

$$\mathcal{L}_{\text{SSL}} = \mathcal{L}_{\text{SSL}}^H + (\mathcal{L}_{\text{SSL}}^X + \mathcal{L}_{\text{SSL}}^A + \mathcal{L}_{\text{SSL}}^Y)/3. \tag{2}$$

Self-supervised training. We train our proposed encoder to learn representations of the history with a self-supervised learning framework modified based on MoCo v3 (Chen* et al., 2021) that achieves the state-of-the-art performance in self-supervised vision transformer training. Following MoCo v3, we set up our proposed encoder f_{enc} as combination of a momentum encoder with the same architecture and initial weights $f_{\text{enc}}^{\text{mo}}$, and a multi-layer perceptron (MLP) as the prediction head $f_{\text{pred\_head}} : \mathbb{R}^{d_{\text{model}}} \to \mathbb{R}^{d_{\text{model}}}$. We first apply the augmentations in (Woo et al., 2022),

which includes scaling, shifting and jittering, on each sample in the input batch $\{\bar{\boldsymbol{H}}_t^{(i)}\}_{i=1}^B$ (B is the batch size) and generates the positive sample pair $\{\bar{\boldsymbol{H}}_t^{'(i)}\}_{i=1}^B$, $\{\bar{\boldsymbol{H}}_t^{''(i)}\}_{i=1}^B$. Their representations are encoded as follows:

$$\begin{aligned}
\boldsymbol{z}_t^{'X(i)}, \boldsymbol{z}_t^{'A(i)}, \boldsymbol{z}_t^{'Y(i)}, \boldsymbol{z}_t^{'(i)} &= f_{\text{enc}}(\bar{\boldsymbol{H}}_t^{'(i)}),\\
\boldsymbol{z}_t^{''X(i)}, \boldsymbol{z}_t^{''A(i)}, \boldsymbol{z}_t^{''Y(i)}, \boldsymbol{z}_t^{''(i)} &= f_{\text{enc}}(\bar{\boldsymbol{H}}_t^{''(i)}),\\
\boldsymbol{z}_t^{'X,\text{mo}(i)}, \boldsymbol{z}_t^{'A,\text{mo}(i)}, \boldsymbol{z}_t^{'Y,\text{mo}(i)}, \boldsymbol{z}_t^{'\text{mo}(i)} &= f_{\text{enc}}^{\text{mo}}(\bar{\boldsymbol{H}}_t^{'(i)}),\\
\boldsymbol{z}_t^{''X,\text{mo}(i)}, \boldsymbol{z}_t^{''A,\text{mo}(i)}, \boldsymbol{z}_t^{''Y,\text{mo}(i)}, \boldsymbol{z}_t^{''\text{mo}(i)} &= f_{\text{enc}}^{\text{mo}}(\bar{\boldsymbol{H}}_t^{''(i)}).
\end{aligned} \tag{3}$$

The vanilla MoCo v3 method adopts the InfoNCE contrastive loss (Oord et al., 2018) as the training objective:

$$\begin{aligned}
\mathcal{L}_{\text{SSL}}^H =& \mathcal{L}_{\text{InfoNCE}}(\{f_{\text{pred_head}}(\boldsymbol{z}_t^{'(i)})\}_{i=1}^B, \{\boldsymbol{z}_t^{''\text{mo}(i)}\}_{i=1}^B)\\
&+ \mathcal{L}_{\text{InfoNCE}}(\{f_{\text{pred_head}}(\boldsymbol{z}_t^{''(i)})\}_{i=1}^B, \{\boldsymbol{z}_t^{'\text{mo}(i)}\}_{i=1}^B),
\end{aligned} \tag{4}$$

where

$$\begin{aligned}
&\mathcal{L}_{\text{InfoNCE}}(\{\boldsymbol{q}^{(i)}\}_{i=1}^B, \{\boldsymbol{k}^{(i)}\}_{i=1}^B)\\
=& -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\cos\langle \boldsymbol{q}^{(i)}, \boldsymbol{k}^{(i)}\rangle)}{\sum_{j=1}^B \exp(\cos\langle \boldsymbol{q}^{(i)}, \boldsymbol{k}^{(j)}\rangle)}.
\end{aligned} \tag{5}$$

Component-wise contrastive loss. In addition to the contrastive loss of the overall representations, we enhance the training with contrastive losses on each subset of covariates, treatments and outcomes:

$$\begin{aligned}
&\mathcal{L}_{\text{SSL}}^{(\cdot)}\\
=&\mathcal{L}_{\text{InfoNCE}}(\{f_{\text{pred_head}}(\boldsymbol{z}_t^{'(\cdot)(i)})\}_{i=1}^B, \{\boldsymbol{z}_t^{''(\cdot),\text{mo}(i)}\}_{i=1}^B)\\
&+ \mathcal{L}_{\text{InfoNCE}}(\{f_{\text{pred_head}}(\boldsymbol{z}_t^{''(\cdot)(i)})\}_{i=1}^B, \{\boldsymbol{z}_t^{'(\cdot),\text{mo}(i)}\}_{i=1}^B),
\end{aligned} \tag{6}$$

where $(\cdot)$ is X, A, Y.

4.3 Non-autoregressive outcome predictor

The architecture of the proposed predictor model is shown in Fig. 2c. At the prediction stage, we first encode the observed history $\bar{\boldsymbol{H}}_t$ with the pretrained encoder, the treatment $\boldsymbol{a}_{t'-1}$ is modeled right before time $t' = t+1, \ldots, t+\tau-1$ with a 1x1 convolution layer, and the remaining treatment sequence $(\boldsymbol{a}_t, \boldsymbol{a}_{t+1}, \ldots, \boldsymbol{a}_{t'-2})$ with a 1D convolution layer. Then, the concatenated encoding is fed into a multi-layer perceptron (MLP) to predict future outcomes $(\hat{\boldsymbol{y}}_{t+1}, \hat{\boldsymbol{y}}_{t+2}, \ldots, \hat{\boldsymbol{y}}_{t+\tau})$. We jointly train the predictor layers and fine tune the pretrained encoder with the L_2 loss of factual outcome estimation weighted for each step:

$$\mathcal{L}_{\text{est}} = \sum_{i=1}^{\tau} w_i \left\|\hat{\boldsymbol{y}}_{t+i} - \boldsymbol{y}_{t+i}\right\|_2^2, \tag{7}$$

where each w_i is a hyperparameter satisfying $\sum_{i=1}^{\tau} w_i = 1$ and various strategies of setting w_is can be selected via validation errors, which we will discuss in the ablation study (Sec. 5.3).

4.4 Unsupervised domain adaptation view of counterfactual outcome estimation

For the given observed history $\bar{\boldsymbol{H}}_t$, treatment sequence $\bar{\boldsymbol{a}}_{t:t+\tau-1}$ to apply, and the outcome $\boldsymbol{y}_{t+\tau}[\boldsymbol{a}_{t:t+\tau-1}](\bar{\boldsymbol{H}}_t)$ to estimate, we notice that learning a counterfactual treatment outcome estimator $f_{\boldsymbol{a}}(\bar{\boldsymbol{H}}_t) = \mathbb{E}(\boldsymbol{y}_{t+\tau}[\boldsymbol{a}]|\bar{\boldsymbol{H}}_t)$ with factual data specifically for a certain treatment sequence $\boldsymbol{a}$ can be viewed as an unsupervised domain adaptation (**UDA**) problem with the treatment value being discrete – any sample in the factual dataset can be categorized into one of the two subsets (1) $\mathcal{S}_{\boldsymbol{a}} = \{(\bar{\boldsymbol{H}}_t^{(i)}, \bar{\boldsymbol{a}}_{t:t+\tau-1}^{(i)}, \boldsymbol{y}_{t+\tau}^{(i)})\}_{\bar{\boldsymbol{a}}_{t:t+\tau-1}^{(i)} = \boldsymbol{a}}$ and (2) $\mathcal{T}_{\bar{\boldsymbol{a}}} = \{(\bar{\boldsymbol{H}}_t^{(i)}, \bar{\boldsymbol{a}}_{t:t+\tau-1}^{(i)}, \boldsymbol{y}_{t+\tau}^{(i)})\}_{\bar{\boldsymbol{a}}_{t:t+\tau-1}^{(i)} \neq \boldsymbol{a}}$. With Assumption B.1, we have $\boldsymbol{y}_{t+\tau}^{(i)} = \boldsymbol{y}_{t+\tau}[\boldsymbol{a}](\bar{\boldsymbol{H}}_t^{(i)})$ in $\mathcal{S}_{\boldsymbol{a}}$ and thus $\boldsymbol{y}_{t+\tau}^{(i)}$ is a label of $f_{\boldsymbol{a}}(\bar{\boldsymbol{H}}_t)$. This does not hold for $\mathcal{T}_{\boldsymbol{a}}$, where $\bar{\boldsymbol{a}}_{t:t+\tau-1}^{(i)} \neq \boldsymbol{a}$. Therefore, $\mathcal{S}_{\boldsymbol{a}}$ and $\mathcal{T}_{\boldsymbol{a}}$ correspond to the **labeled** and the **unlabeled** dataset in UDA. Considering the existence of treatment bias, $P_{\mathcal{T}_{\boldsymbol{a}}}(\bar{\boldsymbol{H}}_t) = P(\bar{\boldsymbol{H}}_t|\bar{\boldsymbol{a}}_{t:t+\tau-1} \neq \boldsymbol{a}) \neq P(\bar{\boldsymbol{H}}_t|\bar{\boldsymbol{a}}_{t:t+\tau-1} = \boldsymbol{a}) = P_{\mathcal{S}_{\boldsymbol{a}}}(\bar{\boldsymbol{H}}_t)$, which corresponds to the distribution shift between the labeled source domain and the unlabeled target domain in UDA. Notice that the source/target domains here are used for describing the labeled and unlabeled subsets regarding a

treatment value to help us analyze the error of counterfactual outcome estimation in UDA framework, and are different from the definitions we use in Section 3. In the latter case, source/target domains describe the different distributions of $\bar{\boldsymbol{H}}_t$ between train/test data. As a natural generalization, we analyze the upper bound of contrastive learning for counterfactual outcome estimation based on the transferability analysis of contrastive learning in UDA (HaoChen et al., 2022):

Theorem 4.1 (Upper bound of counterfactual outcome estimator)**.** *Suppose that Assumptions D.2, D.3, and D.4 hold for the set of observed history $\mathcal{H}$ and its positive-pair graph $G(\mathcal{H}, w)$, and the representation dimension $k \geq 2m$. Let r be a minimizer of the generalized spectral contrastive loss on factual data and the regression head $f_{\boldsymbol{a}}$ be constructed in Alg. 1 with labeled data. We have*

$$\begin{aligned}\mathcal{E}_{\mathcal{D}}(f_{\boldsymbol{a}}) \lesssim & P(\boldsymbol{a})\mathcal{E}_{\mathcal{S}_{\boldsymbol{a}}}(f_{\boldsymbol{a}}) + (1 - P(\boldsymbol{a})) \\ & \cdot \left[\epsilon^2 + (4B^2 - \epsilon^2)\frac{r}{\alpha^2\gamma^4} \cdot \exp(-\Omega(\frac{\rho\gamma^2}{\alpha^2}))\right],\end{aligned} \tag{8}$$

where $P(\boldsymbol{a})$ is the prior probability of the treatment $\boldsymbol{a}$ to apply. $\mathcal{E}_{\mathcal{S}_{\boldsymbol{a}}}(f_{\boldsymbol{a}})$ is the outcome estimation error of $f_{\boldsymbol{a}}$ in $\mathcal{S}_{\boldsymbol{a}}$, which can be optimized with supervised learning. α, r, γ, ρ are parameters in Assumptions D.2, D.3, D.4. ϵ is the hyperparameter in Alg. 1. B is the upper bound of outcome and predicted outcome values[1]. When $\gamma \geq \alpha^{1/2}$ and ρ is comparable to α, $\rho\gamma^2 \gg \alpha^2$ and lead to a small upper bound.

Sketch of the proof For the given observed history $\bar{\boldsymbol{H}}_t$, treatment sequence $\bar{\boldsymbol{a}}_{t:t+\tau-1}$ to apply, and the outcome $\boldsymbol{y}_{t+\tau}[\boldsymbol{a}_{t:t+\tau-1}](\bar{\boldsymbol{H}}_t)$ to estimate, we slightly abuse scalar/vector/matrix notations and denote them as $h, a, y[a](h)$ for simplicity. With discrete treatments[2], we notice that the counterfactual outcome prediction of each type of treatments from factual data can be viewed as an unsupervised domain adaptation (**UDA**) problem:

For a treatment type a, we aim at finding a function $f_a(h)$ that specifically estimates $\mathbb{E}(y[a](h))$. Any sample (h_i, a_i, y_i) from the observed dataset $\mathcal{D}_{tr}$ can be categorized into one of (1) **Labeled subset** $\mathcal{S}_a = \{(h_i, a_i, y_i)\}_{a_i=a}$ and (2) **Unlabeled subset** $\mathcal{T}_a = \{(h_i, a_i, y_i)\}_{a_i\neq a}$. According to Assumption B.1, for any $(h_i, a_i, y_i) \in \mathcal{S}_a$, we have $y_i = y[a](h_i)$ and thus y_i is a label of $f_a(h)$. In contrast, for any $(h_i, a_i, y_i) \in \mathcal{T}_a$, $y_i = y[a_i](h_i)$ and $a_i \neq a$, resulting in that y_i is no longer a valid label of $f_a(h)$. For simplicity, we omit the treatment symbol a as well as the y_i in $\mathcal{T}_a$: $\mathcal{S}_a = \{(h_i, y_i)\}_{a_i=a}$, $\mathcal{T}_a = \{h_i\}_{a_i\neq a}$.

Considering the existence of treatment bias, there exists at least a $a' \neq a$ satisfying $P(h|a) \neq P(h|a')$, which potentially leads to $P_{\mathcal{T}_a}(h) = P_{\mathcal{D}_{tr}}(h_i|a_i \neq a) \neq P_{\mathcal{D}_{tr}}(h_i|a_i = a) = P_{\mathcal{S}_a}(h)$. In counterfactual estimation, we aim at minimizing the estimation error without treatment bias:

$$\mathcal{E}_{\mathcal{D}}(f_a) = \mathbb{E}_{h\sim P_{\mathcal{D}}(h), y\sim P_{\mathcal{D}}(y[a]|h)}\ell(f_a(h), y). \tag{9}$$

Here, we focus on the case where no covariate and concept shifts happen across datasets [3]: $P_{\mathcal{D}}(h) = P_{\mathcal{D}_{tr}}(h) = \sum_{a'} P(a')P_{\mathcal{D}_{tr}}(h|a')$, $P_{\mathcal{D}}(y[a]|h) = P_{\mathcal{D}_{tr}}(y[a]|h) = P_{\mathcal{S}_a}(y[a]|h) = P_{\mathcal{T}_a}(y[a]|h)$. Then Eq. 9 becomes:

$$\begin{aligned}\mathcal{E}_{\mathcal{D}}(f_a) =& \sum_{a'} P(a')\mathbb{E}_{h\sim P_{\mathcal{D}_{tr}}(h|a'), y\sim P_{\mathcal{D}_{tr}}(y[a]|h)}\ell(f_a(h), y) \\ =& P(a)\underbrace{\mathbb{E}_{h\sim P_{\mathcal{S}_a}(h), y\sim P_{\mathcal{D}_{tr}}(y[a]|h)}\ell(f_a(h), y)}_{\mathcal{E}_{\mathcal{S}_a}(f_a)} + \\ & (1 - P(a))\underbrace{\mathbb{E}_{h\sim P_{\mathcal{T}_a}(h), y\sim P_{\mathcal{D}_{tr}}(y[a]|h)}\ell(f_a(h), y)}_{\mathcal{E}_{\mathcal{T}_a}(f_a)}.\end{aligned} \tag{10}$$

So far we can see that for the task of finding an outcome estimator for treatment type a, the counterfactual estimation error is bounded by estimation errors on both $\mathcal{S}_a$ (denoted as $\mathcal{E}_{\mathcal{S}_a}(f_a)$) and $\mathcal{T}_a$ (denoted as $\mathcal{E}_{\mathcal{T}_a}(f_a)$). Per our analysis above, $\mathcal{S}_a$ and $\mathcal{T}_a$ corresponds to the labeled source domain data and the unlabeled target domain data in UDA problems, where potential distribution shifts exist between $\mathcal{S}_a$ and $\mathcal{T}_a$ due to treatment bias. While $\mathcal{E}_{\mathcal{S}_a}(f_a)$ can be optimized with supervised learning using factual data in $\mathcal{S}_a$, we cannot directly optimize $\mathcal{E}_{\mathcal{T}_a}(f_a)$ with labeled data directly.

[1] Bounded outcome values can be achieved through normalization.

[2] For a sequence of discrete treatments, we can always map it to a single discrete variable with a proper encoding.

[3] The distribution shift between $P_{\mathcal{D}}(h)$ v.s. $P_{\mathcal{D}_{tr}}(h)$ (covariate shift) and $P_{\mathcal{D}}(y[a]|h)$ v.s. $P_{\mathcal{D}_{tr}}(y[a]|h)$ (concept shift) can be viewed as the general covariate shift/concept shift and fit into existing theories of domain adaptation/generalization (Farahani et al., 2021).

Recent works (Thota & Leontidis, 2021; Sagawa et al., 2021; Park et al., 2020; Wang et al., 2021) show that contrastive learning, as an effective self-supervised representation learning method, demonstrates strong transferability in UDA and leads to simple state-of-the-art algorithms. Considering the close connection between counterfactual outcome estimation and UDA, we develop our model COSTAR based on contrastive learning and our analysis of the counterfactual estimation error bound from the recent work (HaoChen et al., 2022), where the authors provide theoretical analysis of the transferability of contrastive learning in UDA.

We provide the complete proof in Appendix D.

5 Experiments

Datasets. Following the common evaluation setup for counterfactual treatment outcome estimation overtime in (Lim, 2018; Bica et al., 2020; Melnychuk et al., 2022), we use datasets from both synthetic simulation and real-world observation in our experiments. We provide more detailed dataset description in Appendix E. **(1) Tumor growth.** Following previous work (Lim, 2018; Bica et al., 2020; Melnychuk et al., 2022), we run the pharmacokinetic-pharmacodynamic(PK-PD) tumor growth simulation and generates a fully-synthetic dataset. The PK-PD simulation (Geng et al., 2017) is a state-of-the-art bio-mathematical model simulating the combined effect of chemotherapy and radiotherapy on tumor volumes. **(2) Semi-synthetic MIMIC-III.** (Melnychuk et al., 2022) constructs a semi-synthetic dataset by simulating outcomes under endogenous dependencies on time and exogenous dependencies on observational patient trajectories that are high dimensional and contain long-range dependencies. We include it in our evaluation as a more challenging synthetic dataset. **(3) M5.** The M5 Forecasting dataset (Makridakis et al., 2022) contains daily sales of Walmart stores across three US states, along with the metadata of items and stores, as well as explanatory variables such as price and special events. We transform it to treatment outcome estimation task with the treatment variable of item price, and the outcome variable of the sales of items. Covariate variables include all remaining features. With synthetic data, we report the counterfactual outcome estimation errors and compare the performance of COSTAR with baselines. However, with real-world data, the counterfactual outcome cannot be observed or simulated, thus, we only report the prediction errors of factual outcome.

Feature distribution shifts. To achieve a comprehensive evaluation of counterfactual outcome estimation performance, we introduce feature distribution shifts into the datasets. For each dataset, we split based on static characteristics of subjects into a subset in the source domain and a subset in the target domain. Each subset is further divided into train/validation/test sets. We summarize the main statistics of datasets in Table 4 in the appendix.

Baselines. We select comprehensive methods for estimating counterfactual outcomes over time as baselines, including MSM (Robins et al., 2000), RMSN (Lim, 2018), CRN (Bica et al., 2020), G-Net (Li et al., 2021), and Causal Transformer (CT) (Melnychuk et al., 2022). MSM has difficulty converging when trained with high-dimensional input in semi-synthetic MIMIC-III and M5 datasets and we thus only evaluate it for tumor growth. We empirically find that the balanced representation training losses proposed in CRN and CT do not bring a robust improvement over their variants trained only with the empirical risk minimization (ERM) on factual outcomes. Therefore, we also include these variants, CRN(ERM) and CT(ERM), as baselines.

5.1 Zero-shot transfer setup

To showcase cold-start prediction capabilities, in this setup, we focus on the performance on the target domain after training the model in the source domain, with distributional difference in features. Results are shown in Table 1. COSTAR demonstrates the state-of-the-art performance in a majority of horizons across datasets (4/6 in Tumor growth, 6/6 in Semi-synthetic MIMIC-III and M5). On average, COSTAR decreases the outcome estimation errors by over 6.2%, 22.5% and 26.3% compared to baselines. Results demonstrate the strong transferability of COSTAR in the zero-shot transfer setting.

5.2 Data-efficient transfer learning setup

Effectively utilizing small amount of target domain data can be important, and we showcase that it is indeed one of the key strengths of the proposed approach.

To demonstrate this, for the Tumor Growth Dataset, we fine-tune each method trained on the source domain with 100 sequences from the target domain. For the semi-synthetic MIMIC-III and M5 datasets, we set the number of target domain samples for fine-tuning to be 10% of the number of samples of the target domain in the original dataset. To

Table 1: Results of zero-shot transfer setup for multi-step outcome estimation. We report the mean ± standard deviation of Rooted Mean Squared Errors (RMSEs ↓) over 5 runs. **Bold**: the best results. Underline: the 2nd best results.

Dataset	Method	$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$	$\tau = 6$	Avg	Gain(%)
Tumor growth	MSM	1.0515±0.0674	**0.5048±0.0591**	**0.7583±0.0831**	0.9685±0.1066	1.1561±0.1243	1.3372±0.1356	0.9627±0.0923	6.2%
	RMSN	1.2406±0.1301	1.0914±0.0346	1.1315±0.0634	1.1583±0.0810	1.1674±0.0913	1.1531±0.0919	1.1571±0.0660	22.0%
	CRN(ERM)	1.2924±0.0772	1.1769±0.1058	1.1728±0.1136	1.1906±0.1106	1.1997±0.1061	1.1883±0.0985	1.2035±0.0901	25.0%
	CRN	1.6047±0.0487	2.0846±0.1665	2.0963±0.1274	2.1574±0.1255	2.2609±0.0569	2.3704±0.1368	2.0957±0.0514	68.9%
	CT(ERM)	0.9729±0.0718	1.0217±0.0292	1.1173±0.0457	1.1904±0.0395	1.2359±0.0618	1.2913±0.0939	1.1383±0.0251	20.7%
	CT	1.0272±0.1077	1.1428±0.2182	1.2708±0.2471	1.3608±0.2681	1.4166±0.2935	1.4322±0.3138	1.2751±0.2326	29.2%
	G-Net	1.0492±0.0529	1.0125±0.0767	1.1271±0.0876	1.2153±0.0777	1.2549±0.0727	1.2543±0.0678	1.1522±0.0537	21.7%
	COSTAR	**0.8767±0.0492**	0.7995±0.0853	0.8282±0.0801	**0.9021±0.1062**	**0.9888±0.1280**	**1.0210±0.1168**	**0.9027±0.0814**	(-)
Semi-synthetic MIMIC-III	RMSN	0.2551±0.0303	0.6641±0.1092	0.9107±0.1915	1.1217±0.2916	1.2821±0.3603	1.3950±0.4038	0.9381±0.2210	44.7%
	CRN(ERM)	0.2506±0.0303	0.5545±0.0917	0.7581±0.1112	0.9018±0.1547	1.0113±0.1941	1.1068±0.2324	0.7639±0.1238	32.1%
	CRN	0.4041±0.0537	0.8256±0.1767	1.0439±0.1958	1.1807±0.1725	1.3121±0.2229	1.4374±0.3089	1.0340±0.1606	49.8%
	CT(ERM)	0.2762±0.0804	0.5397±0.1181	0.6765±0.1417	0.7728±0.1636	0.8451±0.1850	0.9028±0.2070	0.6688±0.1472	22.5%
	CT	0.3138±0.0458	0.5992±0.0492	0.7576±0.0694	0.8695±0.0921	0.9510±0.1118	1.0128±0.1274	0.7506±0.0797	30.9%
	G-Net	0.5514±0.1502	0.9398±0.2384	1.2461±0.3321	1.4985±0.4024	1.7045±0.4463	1.8731±0.4660	1.3022±0.3367	60.2%
	COSTAR	**0.2266±0.0249**	**0.4501±0.0893**	**0.5406±0.0987**	**0.5964±0.1020**	**0.6344±0.1040**	**0.6637±0.1052**	**0.5186±0.0869**	(-)
M5	RMSN	15.1616±2.0027	13.9966±0.5316	13.4899±1.2632	13.5162±1.7437	13.8004±2.0637	14.3366±2.3891	13.8280±1.5526	47.5%
	CRN(ERM)	9.8859±1.2980	20.8199±3.9049	38.2653±8.9897	59.4192±16.2788	82.9515±26.1928	105.8120±35.5325	61.4536±17.5760	88.2%
	CRN	8.1119±0.3183	10.3741±2.2616	12.9356±3.1588	15.4168±3.8002	18.1382±4.6750	21.1337±5.4694	15.5997±3.7687	53.5%
	CT(ERM)	7.1253±0.5777	8.3438±1.0313	9.2014±1.4146	9.9409±1.7572	10.6726±2.1718	11.3597±2.5966	9.9037±1.7852	26.7%
	CT	7.1239±0.5770	8.2939±0.9702	9.1465±1.3397	9.9091±1.7198	10.6311±2.0328	11.3032±2.4185	9.8568±1.6959	26.3%
	G-Net	7.5358±0.1605	8.6077±0.3166	9.7167±0.4861	10.8993±0.6902	12.3477±0.8940	13.8200±1.1193	10.4879±0.6078	30.8%
	COSTAR	**6.4054±0.0547**	**6.9328±0.0634**	**7.2428±0.0700**	**7.4585±0.0580**	**7.7012±0.0627**	**7.8278±0.0651**	**7.2614±0.0609**	(-)

achieve a fair comparison, we fine-tune each method until it reaches the lowest factual outcome estimation error on a separate validation set in the target domain.

Table 2 compares the performance of all methods in data-efficient transfer learning setup. For the majority of horizons (4/6 in Tumor growth, 5/6 in Semi-synthetic MIMIC-III and M5), we observe that COSTAR achieves the state-of-the-art performance after fine-tuning. Again, COSTAR reduces the outcome estimation errors by at least 7.8%, 9.9% and 4.11% in the three datasets respectively.

Table 2: Results of the data-efficient transfer learning setup for multi-step outcome estimation. We report the mean ± standard deviation of Rooted Mean Squared Errors (RMSEs ↓) over 5 runs. **Bold**: the best results. Underline: the 2nd best results.

Dataset	Method	$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$	$\tau = 6$	Avg	Gain(%)
Tumor growth	MSM	1.0436±0.0671	**0.5023±0.0588**	**0.7475±0.0829**	0.9537±0.1060	1.1376±0.1233	1.3146±0.1338	0.9499±0.0915	7.8%
	RMSN	1.1839±0.0842	1.0912±0.0405	1.1215±0.0593	1.1538±0.0688	1.1728±0.0773	1.1740±0.0830	1.1495±0.0529	23.8%
	CRN(ERM)	1.2648±0.0689	1.1740±0.1015	1.1507±0.1016	1.1474±0.1070	1.1414±0.1041	1.1203±0.0906	1.1664±0.0786	24.9%
	CRN	1.5019±0.0587	1.5362±0.0248	1.7824±0.1060	1.9842±0.1707	2.1317±0.2431	2.2546±0.3414	1.8651±0.1318	53.0%
	CT(ERM)	0.8947±0.0668	0.8700±0.0857	0.9507±0.1309	1.0031±0.1502	1.0371±0.1545	1.0668±0.1565	0.9704±0.1098	9.7%
	CT	0.9545±0.0782	0.9494±0.1597	1.0225±0.1562	1.1062±0.1377	1.1455±0.1192	1.1562±0.0953	1.0557±0.1136	17.0%
	G-Net	1.0335±0.0622	1.0154±0.1100	1.1105±0.1476	1.1859±0.1620	1.2257±0.1693	1.2198±0.1508	1.1318±0.1118	22.6%
	COSTAR	**0.8654±0.0328**	0.7945±0.0532	0.8248±0.0751	**0.8754±0.0987**	**0.9378±0.1176**	**0.9594±0.1062**	**0.8762±0.0720**	(-)
Semi-synthetic MIMIC-III	RMSN	0.2100±0.0192	0.6084±0.1114	0.7745±0.1180	0.8908±0.1402	0.9776±0.1505	1.0440±0.1529	0.7509±0.1123	31.0%
	CRN(ERM)	**0.1946±0.0158**	0.4770±0.0808	0.5983±0.0923	0.6786±0.1004	0.7315±0.1047	0.7690±0.1070	0.5748±0.0823	9.9%
	CRN	0.2955±0.0256	0.5051±0.0748	0.6361±0.0786	0.7277±0.0783	0.7919±0.0764	0.8379±0.0759	0.6324±0.0656	18.1%
	CT(ERM)	0.2704±0.0631	0.5347±0.1061	0.6712±0.1252	0.7679±0.1433	0.8402±0.1607	0.8968±0.1784	0.6635±0.1279	21.9%
	CT	0.3105±0.0459	0.5840±0.0633	0.7414±0.0887	0.8530±0.1157	0.9348±0.1392	0.9974±0.1608	0.7368±0.0971	29.7%
	G-Net	0.3814±0.0556	0.6519±0.0856	0.8183±0.1122	0.9413±0.1365	1.0359±0.1592	1.1117±0.1795	0.8234±0.1191	37.1%
	COSTAR	0.2288±0.0229	**0.4496±0.0877**	**0.5393±0.0962**	**0.5946±0.0990**	**0.6326±0.1013**	**0.6626±0.1026**	**0.5179±0.0844**	(-)
M5	RMSN	13.9705±0.3867	13.6233±0.8150	13.3291±1.2900	13.1984±1.3892	13.0889±1.2605	13.0108±1.1173	13.2501±1.1696	45.92%
	CRN(ERM)	6.3558±0.0594	7.0530±0.0433	7.3452±0.0447	7.5541±0.0392	7.7636±0.0450	7.9247±0.0561	7.5281±0.0447	4.82%
	CRN	6.2868±0.0471	7.0282±0.0482	7.3327±0.0610	7.5378±0.0521	7.7492±0.0586	7.9094±0.0676	7.5115±0.0572	4.60%
	CT(ERM)	**6.1720±0.0354**	6.9309±0.0571	7.2855±0.0889	7.5418±0.1191	7.7839±0.1283	7.9425±0.1430	7.4969±0.1058	4.42%
	CT	6.2041±0.0252	7.0022±0.0372	7.3675±0.0513	7.6394±0.0894	7.8932±0.1153	8.0701±0.1456	7.5945±0.0845	5.65%
	G-Net	6.7077±0.1006	7.0479±0.1069	7.3872±0.1349	7.6545±0.1596	7.9188±0.1800	8.1186±0.2058	7.4725±0.1461	4.11%
	COSTAR	6.3026±0.0519	**6.8364±0.0560**	**7.1464±0.0674**	**7.3634±0.0619**	**7.6058±0.0640**	**7.7393±0.0637**	**7.1656±0.0592**	(-)

Table 3: Ablation studies for multi-step outcome estimation. We report the mean ± standard deviation of Rooted Mean Squared Errors (RMSEs ↓) over 5 runs. **Bold**: the best results.

Dataset	Component	Choice	$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$	$\tau = 6$	Avg	Gain(%)
Semi-synthetic MIMIC-III	COSTAR		0.2266±0.0249	0.4501±0.0893	**0.5406±0.0987**	**0.5964±0.1020**	**0.6344±0.1040**	0.6637±0.1052	0.5186±0.0869	(-)
	Encoder	w/ VT	0.4897±0.0888	0.6161±0.1139	0.6978±0.1200	0.7428±0.1217	0.7705±0.1196	0.7910±0.1182	0.6846±0.1130	24.2%
		w/ CT	0.3519±0.0584	0.4936±0.0897	0.5762±0.0967	0.6279±0.1017	0.6634±0.1045	0.6874±0.1029	0.5667±0.0919	8.5%
		TB only	0.2729±0.0409	0.4711±0.0836	0.5607±0.0937	0.6160±0.0995	0.6553±0.1038	0.6831±0.1043	0.5432±0.0856	4.5%
		FB only	1.1210±0.0827	1.1287±0.0855	1.1755±0.1076	1.2055±0.1251	1.2296±0.1418	1.2547±0.1566	1.1858±0.1155	56.3%
	FPE	w/ abs	0.2981±0.0444	0.4679±0.0940	0.5561±0.1050	0.6091±0.1079	0.6446±0.1115	0.6694±0.1128	0.5409±0.0951	4.1%
	SSL Loss	none	0.2998±0.0466	0.4718±0.0905	0.5579±0.1007	0.6117±0.1035	0.6460±0.1060	0.6680±0.1060	0.5425±0.0914	4.4%
		w/o comp	0.2884±0.0421	0.4603±0.0898	0.5475±0.1032	0.6013±0.1077	0.6353±0.1079	**0.6610±0.1075**	0.5323±0.0926	2.6%
	SupL Loss	w/ uni	0.2910±0.0355	0.4656±0.0873	0.5547±0.0998	0.6094±0.1039	0.6440±0.1059	0.6681±0.1070	0.5884±0.1006	11.9%
		w/ sq.inv.	**0.1968±0.0148**	**0.4456±0.0815**	0.5415±0.0906	0.6021±0.0954	0.6431±0.0956	0.6761±0.0948	**0.5175±0.0780**	-0.2%
	Decoder	w/ autoreg	0.2049±0.0118	0.7036±0.1422	1.0234±0.2214	1.2023±0.2745	1.4692±0.3711	1.6577±0.4959	1.0435±0.2437	50.3%
M5	COSTAR		6.4054±0.0547	**6.9328±0.0634**	**7.2428±0.0700**	**7.4585±0.0580**	**7.7012±0.0627**	**7.8278±0.0651**	**7.2614±0.0609**	(-)
	Encoder	w/ VT	17.8226±4.7807	17.6769±4.5561	17.5937±4.4219	17.5279±4.2936	17.4113±4.1662	17.2706±4.0453	17.5505±4.3765	58.6%
		w/ CT	6.7386±0.2326	7.1911±0.2474	7.4549±0.2322	7.6524±0.2061	7.8488±0.2021	7.9589±0.2006	7.4741±0.2176	2.8%
		TB only	6.4085±0.0538	6.9547±0.0535	7.2673±0.0453	7.4825±0.0388	7.7167±0.0380	7.8328±0.0430	7.2771±0.0409	0.2%
		FB only	6.8805±0.0333	7.6298±0.0212	7.9706±0.0254	8.1215±0.0298	8.3989±0.0411	8.5303±0.0435	7.9219±0.0311	8.3%
	FPE	w/ abs	6.4089±0.0693	6.9648±0.0617	7.2776±0.0528	7.4834±0.0430	7.7214±0.0421	7.8479±0.0370	7.2840±0.0486	0.3%
	SSL Loss	none	6.4296±0.1193	6.9434±0.0796	7.2548±0.0699	7.4744±0.0753	7.7117±0.0728	7.8429±0.0817	7.2761±0.0827	0.2%
		w/o comp	6.4637±0.0926	6.9847±0.0764	7.2934±0.0705	7.5093±0.0661	7.7497±0.0669	7.8748±0.0608	7.3126±0.0715	0.7%
	SupL Loss	w/ sq.inv.	**6.3425±0.0461**	6.9760±0.0523	7.3170±0.0538	7.5366±0.0614	7.8015±0.0658	7.9439±0.0736	7.3196±0.0583	0.8%
		w/ inv	6.3575±0.0473	6.9427±0.0381	7.2766±0.0422	7.4932±0.0447	7.7431±0.0531	7.8785±0.0628	7.2819±0.0464	0.3%
	Decoder	w/ autoreg	6.3572±0.0621	> 20	> 20	> 20	> 20	> 20	> 20	> 60%

5.3 Ablation studies

We conduct ablation studies in the zero-shot transfer setup to validate the design of COSTAR. We choose the feature-rich datasets: semi-synthetic MIMIC-III and M5 since they contain complex dynamics and thus are more viable for evaluating components capturing temporal and feature-wise interactions.

Encoder. To validate the impact of our proposed encoder architecture, we replace the it with the following variants: (1) Vanilla Transformer **(w/ VT)**. A vanilla transformer with temporal causal attention, which takes the history with all features concatenated as multivariate time series input. (2) CT **(w/ CT)**. The encoder architecture proposed by Causal Transformer (Melnychuk et al., 2022) that concatenates features grouped by covariates/treatments/outcomes into 3 subsets first, then applies self-attention/cross-attention among sequences with each group of features/each pair of feature groups in an alternating way. (3) Temporal Attention Block only (**TB only**). The variant that only includes temporal attention blocks but not feature-wise attention blocks. (4) Feature-wise Attention Block only (**FB only**). The variant that only includes feature-wise attention blocks.

Rows "Encoder | w/VT(w/CT)" in Table 3 demonstrate the superior performance of our proposed encoder architecture. We observe that both methods (CT, COSTAR) processing features respectively outperform VT that simply concatenates all features, marking the importance of explicitly modeling feature interactions. Moreover, the finer-grained modeling of feature interactions between each pair of features in COSTAR further improves the estimation performance compared to the coarser modeling of interactions between feature subsets in CT.

Rows "Encoder | TB only(FB only)" in Table 3 show that the temporal attention blocks are the most critical for temporal outcome estimation, while the feature-wise attention blocks further boost the performance by 4.5% and 0.2% on Semi-synthetic MIMIC-III and M5 datasets respectively.

Feature positional encoding (FPE). We replace the tree-based feature positional encoding with its absolute variant (**w/abs**): each feature maps to a separate learnable encoding vector. We observe that the tree-based positional encoding has gains of 4.1% and 0.3% over the absolute variant in the two datasets respectively.

Self-supervised loss (SSL). To validate the improvement brought by introducing self-supervised learning as well as the choice of its training loss, we compare COSTAR with two variants: (i) **none.** A model with the same architecture as COSTAR but trained with factual estimation losses only; and (ii) **w/o comp.** with vanilla MoCo v3 training loss in Eq. 4 for self-supervised learning. Rows "SSL Loss | none(w/o comp)" in Table 3 compare the estimation performance of the aforementioned choices of self-supervised learning losses and validate the effectiveness of our component-wise contrastive loss in self-supervised learning.

Supervised loss (SupL). We consider different choices of the hyperparameter in the supervised training loss of Eq. 7: (1) **w/ uni**: a uniform weight with each $w_i = 1/\tau$; (2) **w/ inv**: weights in proportion to the inverse of horizon

$w_i = \frac{1/i}{\sum_{j=1}^{\tau} 1/j}$; (3) **w/ sq.inv.**: weights in proportion to the inverse of the squared horizon $w_i = \frac{1/i^2}{\sum_{j=1}^{\tau} 1/j^2}$. Both (2) and (3) are designed to enhance the short-term outcome estimation performance. We select the weights by validation error for each dataset (w/inv for Semi-synthetic MIMIC-III and w/uni for M5), and compare it to the other two variants. While the relative performance order varies across datasets, all variants can outperform the best baseline results in Table 1.

Decoder. We validate the effectiveness of our non-autoregressive design of the decoder and compare it with an autoregressive alternative (**w/ autoreg**) by including the previous outcome in input features. While results in rows "Decoder | w/ autoreg" show good performance in very short horizons ($\tau = 1$), multistep outcome estimation errors quickly diverges with the horizon increasing.

6 Conclusion

In this work, we propose a self-supervised learning framework - Counterfactual Self-Supervised Transformer - to tackle the challenges associated with accurately estimating treatment outcomes over time using observed history, which is a crucial component in areas where randomized controlled trials (RCTs) are not feasible. By integrating self-supervised learning and the Transformer-based encoder combining temporal with feature-wise attention, we've achieved notable advances in estimation accuracy and cross-domain generalization performance.

References

Ioana Bica, Ahmed M Alaa, James Jordon, and Mihaela van der Schaar. Estimating counterfactual treatment outcomes over time through adversarially balanced representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJg866NFvB.

Defu Cao, James Enouen, Yujing Wang, Xiangchen Song, Chuizheng Meng, Hao Niu, and Yan Liu. Estimating treatment effects from irregular time series observations with hidden confounders. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):6897–6905, Jun. 2023. doi: 10.1609/aaai.v37i6.25844. URL https://ojs.aaai.org/index.php/AAAI/article/view/25844.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Xinlei Chen*, Saining Xie*, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia. A brief review of domain adaptation. *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020*, pp. 877–894, 2021.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 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, 2017.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Jeff Z. HaoChen, Colin Wei, Ananya Kumar, and Tengyu Ma. Beyond separability: Analyzing the linear transferability of contrastive representations to related subpopulations. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=vmjckXzRXmh.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Zijie Huang, Yizhou Sun, and Wei Wang. Learning continuous system dynamics from irregularly-sampled partial observations. *Advances in Neural Information Processing Systems*, 33:16177–16187, 2020.

Song Jiang, Zijie Huang, Xiao Luo, and Yizhou Sun. Cf-gode: Continuous-time causal inference for multi-agent dynamical systems. *arXiv preprint arXiv:2306.11216*, 2023.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33:6696–6707, 2020.

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

Bryan Lim. Forecasting treatment responses over time using recurrent marginal structural networks. *Advances in neural information processing systems*, 31, 2018.

Robert K Mahar, Myra B McGuinness, Bibhas Chakraborty, John B Carlin, Maarten J IJzerman, and Julie A Simpson. A scoping review of studies using observational data to optimise dynamic treatment regimens. *BMC medical research methodology*, 21:1–13, 2021.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 38(4):1346–1364, 2022. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2021.11.013. URL https://www.sciencedirect.com/science/article/pii/S0169207021001874. Special Issue: M5 competition.

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

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Changhwa Park, Jonghyun Lee, Jaeyoon Yoo, Minhoe Hur, and Sungroh Yoon. Joint contrastive learning for unsupervised domain adaptation. *arXiv preprint arXiv:2006.10297*, 2020.

James Robins. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical modelling*, 7(9-12):1393–1512, 1986.

James Robins and Miguel Hernan. Estimation of the causal effects of time-varying exposures. *Chapman & Hall/CRC Handbooks of Modern Statistical Methods*, pp. 553–599, 2008.

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, 1994.

James M Robins, Miguel Angel Hernan, and Babette Brumback. Marginal structural models and causal inference in epidemiology. *Epidemiology*, pp. 550–560, 2000.

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

Shiori Sagawa, Pang Wei Koh, Tony Lee, Irena Gao, Sang Michael Xie, Kendrick Shen, Ananya Kumar, Weihua Hu, Michihiro Yasunaga, Henrik Marklund, et al. Extending the wilds benchmark for unsupervised adaptation. *arXiv preprint arXiv:2112.05090*, 2021.

Peter Schulam and Suchi Saria. Reliable decision support using counterfactual models. *Advances in neural information processing systems*, 30, 2017.

Nabeel Seedat, Fergus Imrie, Alexis Bellot, Zhaozhi Qian, and Mihaela van der Schaar. Continuous-time modeling of counterfactual outcomes using neural controlled differential equations. *arXiv preprint arXiv:2206.08311*, 2022.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 464–468, 2018.

Vighnesh Shiv and Chris Quirk. Novel positional encodings to enable tree-based transformers. *Advances in neural information processing systems*, 32, 2019.

Hossein Soleimani, Adarsh Subbaswamy, and Suchi Saria. Treatment-response models for counterfactual reasoning with continuous-time, continuous-valued interventions. *arXiv preprint arXiv:1704.02038*, 2017.

Jerzy Splawa-Neyman, Dorota M Dabrowska, and Terrence P Speed. On the application of probability theory to agricultural experiments. essay on principles. section 9. *Statistical Science*, pp. 465–472, 1990.

Zhiquan Tan, Yifan Zhang, Jingqin Yang, and Yang Yuan. Contrastive learning is spectral clustering on similarity graph, 2023.

Mamatha Thota and Georgios Leontidis. Contrastive domain adaptation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 2209–2218, 2021.

Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*, 2021.

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

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

Chao Wang, Xiaowei Shi, Shuai Xu, Zhe Wang, Zhiqiang Fan, Yan Feng, An You, and Yu Chen. A multi-stage framework for online bonus allocation based on constrained user intent detection. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5028–5038, 2023.

Shirly Wang, Matthew BA 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, 2020.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1780–1790, 2021.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=PilZY3omXV2.

Yanbo Xu, Yanxun Xu, and Suchi Saria. A bayesian nonparametric approach for estimating individualized treatment-response curves. In *Machine learning for healthcare conference*, pp. 282–300. PMLR, 2016.

Hangting Ye, Zhining Liu, Wei Cao, Amir M Amiri, Jiang Bian, Yi Chang, Jon D Lurie, Jim Weinstein, and Tie-Yan Liu. Web-based long-term spine treatment outcome forecasting. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3082–3092, 2023.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8980–8987, 2022.

Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems*, 35:3988–4003, 2022a.

Yizhou Zhang, Defu Cao, and Yan Liu. Counterfactual neural temporal point process for estimating causal influence of misinformation on social media. *Advances in Neural Information Processing Systems*, 35:10643–10655, 2022b.

A Model Architecture

Input feature projection. Assume the concatenation of time-varying variables $(\bar{\boldsymbol{X}}_t, \bar{\boldsymbol{A}}_{t-1}, \bar{\boldsymbol{Y}}_t)$ in history as $\bar{\boldsymbol{S}}_t \in \mathbb{R}^{t\times d_S}$, $d_S = d_X + d_A + d_Y$, and the static variables $\boldsymbol{V} \in \mathbb{R}^{d_V}$. We adopt a linear transformation $f_{\text{input}} : \mathbb{R} \to \mathbb{R}^{d_{\text{model}}}$ to map $\bar{\boldsymbol{S}}_t$ and $\boldsymbol{V}$ to the embedding space as $\boldsymbol{E}^S \in \mathbb{R}^{t\times d_S\times d_{\text{model}}}$ and $\boldsymbol{E}^V \in \mathbb{R}^{d_V\times d_{\text{model}}}$ respectively, where

$$\boldsymbol{E}^S[i,j] = f_{\text{input}}(\bar{\boldsymbol{S}}_t[i,j]),\ 1 \le i \le t,\ 1 \le j \le d_S; \tag{11}$$

$$\boldsymbol{E}^V[j] = f_{\text{input}}(\boldsymbol{V}[j]),\ 1 \le j \le d_V. \tag{12}$$

Feature positional encoding. A shared feature projection function among all features is not sufficient to encode the feature-specific information since the same scalar value represents different semantics in different features. Meanwhile, feature-specific information is also critical for modeling the interactions among features. Therefore we enhance the input embedding with a positional encoding along the feature dimension. Since the features can be grouped into covariates, treatments and outcomes and form a hierarchical structure with 2 levels, we model it with a learnable tree positional encoding (Shiv & Quirk, 2019; Wang et al., 2021). Denote the lists of covariate, treatment, outcome, and static features as $\boldsymbol{F}_X$, $\boldsymbol{F}_A$, $\boldsymbol{F}_Y$, and $\boldsymbol{F}_V$. For the i-th feature $f_i^{\boldsymbol{F}}$ in a certain feature list $\boldsymbol{F} \in \{\boldsymbol{F}_X, \boldsymbol{F}_A, \boldsymbol{F}_Y, \boldsymbol{F}_V\}$, its positional encoding is:

$$\begin{aligned}
&\boldsymbol{E}_{\text{fea_pos}}(f_i^{\boldsymbol{F}}) = \boldsymbol{E}_{\text{fea}} \cdot \text{Concat}(\boldsymbol{e}_{\boldsymbol{F}}, \boldsymbol{e}_i),\ \text{where} \\
&\boldsymbol{e}_{\boldsymbol{F}} = \begin{cases} (1,0,0,0) \text{ if } \boldsymbol{F} \text{ is } \boldsymbol{F}_X \\ (0,1,0,0) \text{ if } \boldsymbol{F} \text{ is } \boldsymbol{F}_A \\ (0,0,1,0) \text{ if } \boldsymbol{F} \text{ is } \boldsymbol{F}_Y \\ (0,0,0,1) \text{ if } \boldsymbol{F} \text{ is } \boldsymbol{F}_V \end{cases},
\end{aligned} \tag{13}$$

$\boldsymbol{e}_i \in \mathbb{R}^{\max(d_X,d_A,d_Y,d_V)}$ is a one-hot vector with only $\boldsymbol{e}_i[i] = 1$. $\boldsymbol{E}_{\text{fea}} \in \mathbb{R}^{d_{\text{model}}\times(4+\max(d_X,d_A,d_Y,d_V))}$ are learnable tree embedding weights. After obtaining the stacked feature positional embeddings $\boldsymbol{E}^S_{\text{fea\_pos}} \in \mathbb{R}^{d_S\times d_{\text{model}}}$, $\boldsymbol{E}^V_{\text{fea\_pos}} \in$

$\mathbb{R}^{d_V \times d_{\text{model}}}$ of time-varying and static features, we broadcast them to the shape of $\boldsymbol{E}^S$ and $\boldsymbol{E}^V$ respectively along the time dimension. The embedded input is then the sum of input feature projection and feature positional encoding:

$$\boldsymbol{Z}^{S,(0)} = \boldsymbol{E}^S + \text{Broadcast}(\boldsymbol{E}^S_{\text{fea\_pos}}), \tag{14}$$

$$\boldsymbol{Z}^{V,(0)} = \boldsymbol{E}^V + \text{Broadcast}(\boldsymbol{E}^V_{\text{fea\_pos}}). \tag{15}$$

Temporal attention block. The temporal attention block is designed to capture the temporal dependencies within each feature. We construct the block based on the self-attention part in the conventional Transformer decoder (Vaswani et al., 2017). Considering the importance of relative time interval in modeling treatment outcomes, we adopt the relative positional encoding (Shaw et al., 2018; Melnychuk et al., 2022) along the time dimension.

The temporal attention block in the l-th layer receives $\boldsymbol{Z}^{S,(l-1)}$ from the previous layer, reshapes it to d_S sequences with lengths t, and passes them through the block in parallel. The outputs $\boldsymbol{Z}^{S,(l)}_{\text{tmp}}$ have the same shape as $\boldsymbol{Z}^{S,(l-1)}$. Since $\boldsymbol{Z}^{V,(l-1)}$ is static, we only pass it through the point-wise feed-forward module and get $\boldsymbol{Z}^{V,(l)}_{\text{tmp}}$

Feature-wise attention block. The feature-wise attention block models interactions among different features. We reuse the architecture of the conventional Transformer encoder but replace the positional encoding with the feature positional encoding as described. The block in the l-th layer receives $\boldsymbol{Z}^{S,(l)}_{\text{tmp}}$ from the temporal attention block and reshapes it to t sequences, each with a length d_S. We broadcast $\boldsymbol{Z}^{V,(l)}_{\text{tmp}}$ and concatenate it with each sequence along the feature dimension to enable the attention among both time-varying and static features. The concatenated t sequences that we apply attention to are:

$$\begin{aligned}\boldsymbol{Z}^{SV,(l)}_{\text{tmp}} &= \text{Concat}(\boldsymbol{Z}^{S,(l)}_{\text{tmp}}, \text{Broadcast}(\boldsymbol{Z}^{V,(l)}_{\text{tmp}})) \\ &\in \mathbb{R}^{t \times (d_S + d_V) \times d_{\text{model}}}.\end{aligned} \tag{16}$$

We apply full attention across all features and get $\boldsymbol{Z}^{SV,(l)}$ with the same shape as $\boldsymbol{Z}^{SV,(l)}_{\text{tmp}}$. The propagated embeddings of time-varying features are obtained as:

$$\boldsymbol{Z}^{S,(l)} = \boldsymbol{Z}^{SV,(l)}[:, : d_S, :] \in \mathbb{R}^{t \times d_S \times d_{\text{model}}}. \tag{17}$$

To keep the $\boldsymbol{Z}^{V,(l)}$ static after feature-wise attention, we only propagate $\boldsymbol{Z}^{V,(l)}_{\text{tmp}}$ with full attention among static features only. The updated embeddings of the static features $\boldsymbol{Z}^{V,(l)} \in \mathbb{R}^{d_V \times d_{\text{model}}}$ have the same shape as $\boldsymbol{Z}^{V,(l)}_{\text{tmp}}$.

B Identifiability assumptions

Assumption B.1 (Consistency)**.** *The potential outcome of any treatment $\boldsymbol{a}_t$ is always the same as the factual outcome when a subject is given the treatment $\boldsymbol{a}_t$: $\boldsymbol{y}_{t+1}[\boldsymbol{a}_t] = \boldsymbol{y}_{t+1}$.*

Assumption B.2 (Positivity)**.** *If $P(\bar{\boldsymbol{A}}_{t-1} = \bar{\boldsymbol{a}}_{t-1}, \bar{\boldsymbol{X}}_t = \bar{\boldsymbol{x}}_t) \neq 0$, then $P(\boldsymbol{A}_t = \boldsymbol{a}_t | \bar{\boldsymbol{A}}_{t-1} = \bar{\boldsymbol{a}}_{t-1}, \bar{\boldsymbol{X}}_t = \bar{\boldsymbol{x}}_t) > 0$ for any $\bar{\boldsymbol{a}}_t$.*

Assumption B.3 (Sequential strong ignorability)**.** $\boldsymbol{Y}_{t+1}[\boldsymbol{a}_t] \perp\!\!\!\perp \boldsymbol{A}_t | \bar{\boldsymbol{A}}_{t-1}, \bar{\boldsymbol{X}}_t, \forall \boldsymbol{a}_t, t$.

C Why use temporally causal attention in COSTAR?

The embeddings of time-varying features $\boldsymbol{Z}^{S,(L)} \in \mathbb{R}^{t \times (d_X + d_A + d_Y) \times d_{\text{model}}}$ from the final layer is re-organized to step-wise representations $(\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_t)$. Each $\boldsymbol{Z}_{t'} = \{\boldsymbol{z}^i_{t'} \in \mathbb{R}^{d_{\text{model}}}\}_{i=1}^{d_X + d_A + d_Y}$ is further aggregated to $\boldsymbol{z}^X_{t'}, \boldsymbol{z}^A_{t'}, \boldsymbol{z}^Y_{t'}, \boldsymbol{z}_{t'}$ in Equation 1. When the encoder satisfies the temporal causality (i.e. $\boldsymbol{Z}_{t'}$ only depends on $\bar{\boldsymbol{H}}_{t'}$), they can be seen as a sequence of representations for the observed history $\bar{\boldsymbol{H}}_1, \bar{\boldsymbol{H}}_2, \ldots, \bar{\boldsymbol{H}}_t$ truncated at each time step.

When we feed the encoder with factual data in training stages, a major advantage of encoders satisfying temporal causality is that we can estimate the outcomes and evaluate the factual estimation losses in every time step of the input sequence at a single forward pass. Evaluating of counterfactual data, the encoder only needs to keep $\boldsymbol{Z}_t$ representing the entire observed history, conditioning on which the predictor rolls out outcome estimations given counterfactual treatments.

In contrast, feeding the entire history in one pass for training is error-prone for architectures and can violate the temporal causality (e.g. transformers with fully temporal attention or frequency-based methods), since it leaks information of future steps into the representations in previous steps. Predictors trained with such representations converge quickly

to a trivial model that simply copies future steps as estimations. When we evaluate counterfactual data where future counterfactual outcomes are no longer available in input, the performance degenerates. As a result, we have to explicitly unroll the observed sequence to t truncated sequences and run t forward passes to get the representations and factual errors in every step when training non-temporally-causal models. This leads to a $\times T$ increase in training time when the batch size remains unchanged due to hardware restrictions, where T is the maximum length of sequences in training data. We have $T \geq 50$ in our experiments and find that none of the architecture violating temporal causality can finish training in a reasonable time.

D Proof of generalization bound of contrastive learning in counterfactual outcome estimation

D.1 Preliminaries

Positive pairs. Pairs of semantically related/similar data samples are positive pairs in contrastive learning. In contrastive learning, positive pairs are commonly generated by applying randomized transformation on the same input (He et al., 2020; Woo et al., 2022).

For exposition simplicity, we assume the set of factual observed history $\mathcal{H}$ is a finite but large dataset of size N. We use P_+ to denote the distribution of positive pairs. P_+ satisfies $P_+(h, h') = P_+(h', h),\ \forall\, h, h' \in \mathcal{H}$. $P_{\mathcal{H}}$ denotes the marginal distribution of P_+: $P_{\mathcal{H}}(h) = \sum_{h' \in \mathcal{H}} P_+(h, h')$.

Positive-pair graph. Following the definition in (HaoChen et al., 2022), we introduce the *postive-pair graph* as a weighted undirected graph $G(\mathcal{H}, w)$ with the vertex set $\mathcal{H}$ and the edge weight $w(h, h') = P_+(h, h')$. $w(h) = P_{\mathcal{H}}(h) = \sum_{h' \in \mathcal{H}} w(h, h')$. For any vertex subset A, $w(A) = \sum_{h \in A} w(h)$. For any vertex subsets A, B, $w(A, B) = \sum_{h \in A, h' \in B} w(h, h')$. For any vertex h and vetex subset B, $w(h, B) = w(\{h\}, B)$.

Generalized spectral contrastive loss. Let $r : \mathcal{H} \to \mathbb{R}^k$ be a mapping from the input data to k-dimensional features. For the convenience of proof, we consider the (generalized) spectral contrastive loss proposed in (HaoChen et al., 2022):

$$\mathcal{L}_\sigma(r) = \mathbb{E}_{(h, h^+) \sim P_+} \left[\left\| r(h) - r(h^+) \right\|_2^2 \right] + \sigma \cdot R(r), \tag{18}$$

where the regularizer is defined as $R(r) = \left\| \mathbb{E}_{h \in P_{\mathcal{H}}} [r(h) r(h)^T] - I_{k \times k} \right\|_F^2$ and $I_{k \times k}$ is the k-dimensional identity matrix. Notice that the InfoNCE loss is more commonly used in empirical study (He et al., 2020; Chen et al., 2020; Chen* et al., 2021) instead of the spectral contrastive loss, and their equivalence is still an open problem with some preliminary results (Tan et al., 2023).

D.2 Definitions and assumptions

We reiterate the following definitions and assumptions in (HaoChen et al., 2022) for self-containment:

Definition D.1 (Expansion)**.** *Let A, B be two disjoint subsets of $\mathcal{H}$. We denote the expansion, max-expansion and min-expansion from A to B as follows:*

$$\begin{aligned} \phi(A, B) &= \frac{w(A, B)}{w(A)}, \\ \bar{\phi}(A, B) &= \max_{h \in A} \frac{w(h, B)}{w(h)}, \\ \underline{\phi}(A, B) &= \min_{h \in A} \frac{w(h, B)}{w(h)}. \end{aligned} \tag{19}$$

Assumption D.2 (Cross-cluster connections)**.** *For some $\alpha \in (0, 1)$, we assume that vertices of the positive-pair graph $G(\mathcal{H}, w)$ can be partitioned into m disjoint clusters $C_1, \ldots, C_m$ such that for any $i \in [m]$,*

$$\bar{\phi}(C_i, \mathcal{H} \backslash C_i) \leq \alpha. \tag{20}$$

Assumption D.3 (Intra-cluster conductance)**.** *For all $i \in [m]$, assume the conductance of the subgraph restricted to C_i is large, i.e., every subset A of C_i with at most half the size of C_i expands to the rest:*

$$\forall\, A \subset C_i \text{ satisfying } w(A) \leq w(C_i)/2,\ \phi(A, C_i \backslash A) \geq \gamma. \tag{21}$$

Assumption D.4 (Relative expansion)**.** *Let S and T be two disjoint subsets of $\mathcal{H}$, each is formed by r clusters among $C_1, C_2, \ldots, C_m$ for $r \leq m/2$. Let $\rho = \min_{i\in[r]} \underline{\phi}(T_i, S_i)$ be the minimum min-expansions from T_i to S_i. For some sufficiently large universal constant c, we assume that $\rho \geq c \cdot \alpha^2$ and that*

$$\rho = \min_{i\in[r]} \underline{\phi}(T_i, S_i) \geq c \cdot \max_{i\neq j} \bar{\phi}(T_i, S_j). \tag{22}$$

D.3 Proof of Theorem 4.1

We adapt the preconditioned featurer averaging classifier in (HaoChen et al., 2022) for regression in our proof:

Algorithm 1 Preconditioned feature averaging (PFA).

Require: Pretrained representation extractor r, unlabeled data $P_{\mathcal{H}}$, source domain labeled data P_S, target domain test data $\tilde{h}$, integer $t \in \mathbb{Z}^+$, outcome discretization granularity ϵ.

1: Compute the preconditioner matrix $\Sigma = \mathbb{E}_{h\in P_{\mathcal{H}}}[r(h)r(h)^T]$.
2: **for** every outcome value y_i corresponding to the cluster $C_i, i \in [r]$ **do**
3: Compute the mean feature of outcome y_i: $b_i = \mathbb{E}_{(h,y)\sim P_S}[\mathbb{1}[\|y - y_i\|_2 \leq \epsilon] \cdot r(h)]$.
4: **end for**
5: **return** prediction y_{i^*}, $i^* = \arg\max_{i\in[r]} \langle r(h), \sum^{t-1} b_i \rangle$.

For any PFA regressor f constructed with Alg. 1, we can transform it to a corresponding classifier by defining its 0-1 classification error on the target domain T as:

$$\mathcal{E}_T^{01}(f) = \mathbb{E}_{(h,y)\sim P_T}[\mathbb{1}[\|y - f(h)\|_2 > \epsilon]]. \tag{23}$$

We can directly apply the main result in (HaoChen et al., 2022) and get an upper bound of the 0-1 error on the target domain:

Theorem D.5 (Upper bound of 0-1 error on the target domain (HaoChen et al., 2022))**.** *Suppose that Assumption D.2, Assumption D.3, and Assumption D.4 holds for the set of observed history $\mathcal{H}$ and its positive-pair graph $G(\mathcal{H}, w)$, and the representation dimension $k \geq 2m$. Let r be a minimizer of the generalized spectral contrastive loss and the regression head f be constructed in Alg. 1. We have*

$$\mathcal{E}_T^{01}(f) \lesssim \frac{r}{\alpha^2\gamma^4} \cdot \exp(-\Omega(\frac{\rho\gamma^2}{\alpha^2})). \tag{24}$$

Lemma D.6 (Relation between the L2 regression error and 0-1 classification error)**.** *Suppose that both $\|f(h)\|_2 \leq B$ and $\|y\|_2 \leq B$, $\epsilon < 2B$. The L2 regression error $\mathcal{E}_T(f)$ of the PFA regressor on the target domain T is bounded by $\mathcal{E}_T^{01}(f)$ as:*

$$\mathcal{E}_T(f) \leq \epsilon^2 + (4B^2 - \epsilon^2)\mathcal{E}_T^{01}(f). \tag{25}$$

Proof.

$$\begin{aligned}
\mathcal{E}_T(f) =& \mathbb{E}_{(h,y)\in P_T} \|y - f(h)\|_2^2 \\
\leq& \sum_{(h,y)\in T} P(h,y) \left[\mathbb{1}[\|y - f(h)\|_2 > \epsilon] \|y - f(h)\|_2^2\right] \\
&+ \sum_{(h,y)\in T} P(h,y)(1 - \mathbb{1}[\|y - f(h)\|_2 > \epsilon])\epsilon^2 \\
\leq& \sum_{(h,y)\in T} P(h,y) \left[\mathbb{1}[\|y - f(h)\|_2 > \epsilon] 4B^2\right] \\
&+ \sum_{(h,y)\in T} P(h,y)(1 - \mathbb{1}[\|y - f(h)\|_2 > \epsilon])\epsilon^2 \\
=& \epsilon^2 + (4B^2 - \epsilon^2)\mathbb{E}_{(h,y)\in P_T} \mathbb{1}[\|y - f(h)\|_2 > \epsilon] \\
=& \epsilon^2 + (4B^2 - \epsilon^2)\mathcal{E}_T^{01}(f).
\end{aligned}$$

□

Lemma D.6 connects the L2 error and the 0-1 error. Combining Eq. 10, Eq. 24, Eq. 25, we immediately get Theorem 4.1.

E Dataset description

Table 4: Statistics of datasets.

Dataset	Domain	Property	Seq Length	Train/Validation/Test Seq Num
Tumor growth	source	$\gamma = 10$	60	10000/1000/1000
	target	$\gamma = 0$	60	100/1000/1000
Semi-synthetic MIMIC-III	source	age in [20,45]	99	3704/926/926
	target	age$\geq$85	99	138/347/1737
M5	source	food items	50	39606/7048/7048
	target	household items	50	3623/3512/18005

We summarize the statistics and the way of introducing feature distribution shifts in Table 4.

Tumor growth. We refer readers to (Bica et al., 2020; Melnychuk et al., 2022) for the complete descriptions of the pharmacokinetic-pharmacodynamic (PK-PD) model. Here we focus on how we introduce distribution shifts by adjusting the treatment bias coefficient γ.

The volume of tumor after t days of diagnosis is:

$$\begin{aligned} & V(t+1) \\ =&(1+\rho \log(\frac{K}{V(t)}) - \beta_c C(t) - (\alpha_r d(t) + \beta_r d(t)^2) + e_t) \\ & \cdot V(t), \end{aligned} \tag{26}$$

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

PK-PD model constructs time-varying confounding by connecting the probability of assigning chemotherapy and radiotherapy with the outcome - tumor diameter:

$$\begin{aligned} p_c(t) &= \sigma(\frac{\gamma_c}{D_{\text{max}}}(\bar{D}(t) - \delta_c)), \\ p_r(t) &= \sigma(\frac{\gamma_r}{D_{\text{max}}}(\bar{D}(t) - \delta_r)). \end{aligned} \tag{27}$$

$\bar{D}(t)$ is the mean tumor diameter in the past 15 days and $D_{\text{max}} = 13$. σ is the sigmoid function. $\delta_c = \delta_r = D_{\text{max}}/2$. γ_c and γ_r controls the importance of tumor diameter history on treatment assignment, thus control the strength of time-dependent confounding.

In Tumor growth dataset, we set $\gamma_c = \gamma_r = \gamma = 10$ to generate data in the source domain, and $\gamma_c = \gamma_r = \gamma = 0$ for the target domain. As a result, both treatment bias and the data distribution of history differs between source and target domains.

Semi-synthetic MIMIC-III. We split the semi-synthetic MIMIC-III dataset introduced in (Melnychuk et al., 2022) by ages of patients to the source/target domain. More specifically, we generate simulation data from patients with ages falling in $[20, 45]$ as the source domain data and simulation based on patients with ages over 85 as the target domain data. Missing values in MIMIC-III dataset is imputed with the so-called "Simple Imputation" described in Wang et al. (2020). Missing values are first forward filled and then set to individual-specific mean if there are no previous values. If the variable is always missing for a patient, we set it to the global mean.

M5. We adapt the M5 forecasting dataset (https://www.kaggle.com/competitions/m5-forecasting-accuracy) for treatment effect estimation over time. In M5, we select the item pricing as treatment, its sales as outcome and all other features as covariates. We aggregate the item sales by week to reduce the sequence length to the same level as the other two datasets for the convenience of evaluation. We also discretize the continuous pricing by mapping $(p_{t,i} - p_{0,i})/p_{0,i}$ to buckets divided by its 20-quantiles, where $p_{t,i}, p_{0,i}$ are the prices of item i at time t and at its initial sale.

To introduce the feature distribution shift, we select 5000 items in the food category as the source domain data and another 5000 items in the household category as the target domain data.

F Baselines

Baseline implementation. We reuse the implementation in (Melnychuk et al., 2022) for evaluating all the baselines, including: MSM (Robins et al., 2000), RMSN (Lim, 2018), CRN (Bica et al., 2020), G-Net (Li et al., 2021), and Causal Transformer (CT) (Melnychuk et al., 2022).

Hyperparameter tuning. For all baselines, we follow the ranges of hyperparameter tuning in (Melnychuk et al., 2022) and select the hyperparameters with the lowest factual outcome estimation error on the validation set from the source domain. For each method and each dataset, the same set of hyperparameters are used in the zero-shot transfer/data-efficient transfer/standard supervised learning settings. The detailed hyperparameters used for baselines and COSTAR are listed in the configuration files in our code repository. Here we list the main hyperparameters for reference.

MSM. There is no tuneable hyperparameter in MSM.

RMSN. We list the hyperparameters of RMSN in Table 5.

Table 5: RMSN hyperparameters.

		Tumor growth	Semi-synthetic MIMIC-III	M5
Propensity Treatment	RNN Hidden Units	8	6	44
	Dropout	0.5	0.1	0.4
	Layer Num	1	2	1
	Max Gradient Norm	1.0	0.5	2.0
	Batch Size	128	256	128
	Learning Rate	0.01	0.01	0.001
Propensity History	RNN Hidden Units	24	74	92
	Dropout	0.1	0.5	0.5
	Layer Num	1	2	2
	Max Gradient Norm	2.0	1.0	0.5
	Batch Size	128	64	128
	Learning Rate	0.01	0.001	0.01
Encoder	RNN Hidden Units	24	74	46
	Dropout	0.1	0.1	0.1
	Layer Num	1	1	2
	Max Gradient Norm	0.5	0.5	0.5
	Batch Size	64	1024	128
	Learning Rate	0.01	0.001	0.0001
Decoder	RNN Hidden Units	48	196	45
	Dropout	0.1	0.1	0.1
	Layer Num	1	1	1
	Max Gradient Norm	0.5	0.5	4.0
	Batch Size	256	1024	256
	Learning Rate	0.0001	0.0001	0.0001

CRN(ERM). See Table 6.

CRN. See Table 7.

CT(ERM). See Table 8.

CT. See Table 9.

G-Net. See Table 10.

COSTAR. See Table 11.

Comparison of numbers of model parameters. Here we list the number of trainable parameters in each baseline as well as COSTAR in the experiments of each dataset.

Table 6: CRN(ERM) hyperparameters.

		Tumor growth	Semi-synthetic MIMIC-III	M5
Encoder	RNN Hidden Units	24	74	46
	Balancing Representation Size	18	74	46
	FC Hidden Units	18	37	46
	Layer Num	1	1	2
	Dropout	0.1	0.1	0.1
	Batch Size	256	64	128
	Learning Rate	0.01	0.001	0.001
Decoder	RNN Hidden Units	18	74	46
	Balancing Representation Size	6	98	90
	FC Hidden Units	6	98	22
	Layer Num	1	2	1
	Dropout	0.1	0.1	0.1
	Batch Size	256	256	256
	Learning Rate	0.001	0.0001	0.0001

Table 7: CRN hyperparameters.

		Tumor growth	Semi-synthetic MIMIC-III	M5
Encoder	RNN Hidden Units	18	74	46
	Balancing Representation Size	3	74	46
	FC Hidden Units	12	37	46
	Layer Num	1	1	2
	Dropout	0.2	0.1	0.1
	Batch Size	256	64	128
	Learning Rate	0.001	0.001	0.001
Decoder	RNN Hidden Units	3	74	46
	Balancing Representation Size	3	98	90
	FC Hidden Units	3	98	22
	Layer Num	1	2	1
	Dropout	0.2	0.1	0.1
	Batch Size	256	256	256
	Learning Rate	0.001	0.0001	0.0001

Table 8: CT(ERM) hyperparameters.

	Tumor growth	Semi-synthetic MIMIC-III	M5
Transformer Hidden Units	12	24	24
Balancing Representation Size	2	88	94
FC Hidden Units	12	44	47
Layer Num	1	1	2
Head Num	2	3	2
Max Relative Position	15	20	30
Dropout	0.1	0.1	0.1
Batch Size	64	64	64
Learning Rate	0.001	0.01	0.001

Table 9: CT hyperparameters.

	Tumor growth	Semi-synthetic MIMIC-III	M5
Transformer Hidden Units	16	24	24
Balancing Representation Size	16	88	94
FC Hidden Units	16	44	47
Layer Num	1	1	2
Head Num	2	3	2
Max Relative Position	15	20	30
Dropout	0.2	0.1	0.1
Batch Size	64	64	64
Learning Rate	0.001	0.01	0.001

Table 10: G-Net hyperparameters.

	Tumor growth	Semi-synthetic MIMIC-III	M5
RNN Hidden Units	24	148	144
FC Hidden Units	48	74	72
Dropout	0.1	0.1	0.1
Layer Num	1	1	2
Batch Size	128	256	256
Learning Rate	0.001	0.01	0.001

Table 11: COSTAR hyperparameters.

		Tumor growth	Semi-synthetic MIMIC-III	M5
Encoder	Transformer Hidden Units	24	36	36
	Encoder Momentum	0.99	0.99	0.99
	Temperature	1.0	1.0	1.0
	Layer Num	1	1	2
	Head Num	2	3	2
	Dropout	0.1	0.1	0.1
	Batch Size	64	64	64
	Learning Rate	0.001	0.001	0.001
Decoder	Hidden Units	128	128	128
	Batch Size	32	32	32
	Learning Rate	0.001	0.001	0.001

Table 12: Number of trainable parameters.

#trainable params	Tumor growth	semi-synthetic MIMIC-III	M5
MSM	<1K	(-)	(-)
RMSN	18.8K	387K	213K
CRN(ERM)	6.5K	164K	78K
CRN	2.3K	164K	78K
CT(ERM)	5.2K	45K	80.3K
CT	9.4K	45K	80.3K
G-Net	3.4K	151K	323K
COSTAR	20.7K	43.6K	77.5K

G Results of supervised learning setup

Table 13 shows the performance in standard supervised learning setting, with both train and test data from the source domain. Overall, COSTAR outperforms other baselines in tumor growth and semi-synthetic MIMIC-III datasets. With M5, COSTAR also shows comparable performance to the CT(ERM) with a 1.3% relative difference.

Table 13: Results in standard supervised learning setting, with source and target datasets coming from the same distribution for multi-step outcome estimation. We report the mean +- standard deviation of Rooted Mean Squared Errors (RMSEs) over 5 runs. **Bold**: the best results. Underline: the 2nd best results.

Dataset	Method	$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$	$\tau = 5$	$\tau = 6$	Avg	Gain(%)
Tumor growth	MSM	5.8368±0.6157	**2.0400±0.6719**	**3.0385±0.9990**	3.8701±1.2736	4.6173±1.5246	5.3823±1.7839	4.1308±1.1211	12.3%
	RMSN	4.8388±0.7770	5.4447±1.9202	5.9261±2.1096	5.9817±2.1270	5.8705±2.0544	5.5461±1.8865	5.6013±1.7727	35.4%
	CRN(ERM)	5.1601±0.5222	6.0784±2.3196	6.4721±2.4221	6.6142±2.4206	6.5648±2.3455	6.2939±2.1955	6.1972±2.0226	41.6%
	CRN	4.8130±0.2296	6.3126±2.9523	6.6993±3.8805	6.7520±3.8551	6.8386±3.5630	6.8852±3.1150	6.3834±2.8863	43.3%
	CT(ERM)	5.1286±1.3377	5.7262±2.7601	6.5085±2.9886	6.9248±3.0009	7.1971±2.9346	7.2369±2.7570	6.4537±2.5904	43.9%
	CT	6.5485±1.5221	7.5382±2.8528	7.9030±2.9569	7.9828±2.9332	7.8244±2.8075	7.4418±2.6103	7.5398±2.5976	52.0%
	G-Net	3.9371±0.4023	3.7697±1.1861	4.6054±1.4181	4.9730±1.4773	5.0491±1.4410	4.8745±1.3153	4.5348±1.1778	20.1%
	COSTAR	**3.7403±0.3695**	3.0067±0.9065	3.4619±1.1557	**3.8501±1.3127**	**3.9160±1.3142**	**3.7525±1.1493**	**3.6212±1.0040**	(-)
Semi-synthetic MIMIC-III	RMSN	0.2107±0.0261	0.5352±0.0842	0.6722±0.1096	0.7669±0.1203	0.8309±0.1280	0.8764±0.1331	0.6487±0.0976	21.7%
	CRN(ERM)	**0.1951±0.0202**	0.4426±0.0799	0.5530±0.0859	0.6113±0.0842	0.6478±0.0828	0.6708±0.0819	0.5201±0.0713	2.4%
	CRN	0.3276±0.0301	0.5234±0.0839	0.6531±0.0985	0.7234±0.0985	0.7618±0.0921	0.7825±0.0854	0.6286±0.0801	19.2%
	CT(ERM)	0.2130±0.0164	0.4426±0.0766	0.5495±0.0836	0.6191±0.0851	0.6669±0.0856	0.7010±0.0834	0.5320±0.0695	4.6%
	CT	0.2175±0.0178	0.4421±0.0757	0.5458±0.0854	0.6161±0.0925	0.6670±0.0993	0.7047±0.1040	0.5322±0.0765	4.6%
	G-Net	0.3418±0.0290	0.6015±0.0653	0.7542±0.0758	0.8620±0.0825	0.9429±0.0875	1.0035±0.0915	0.7510±0.0686	32.4%
	COSTAR	0.2286±0.0265	**0.4417±0.0876**	**0.5288±0.0957**	**0.5825±0.0991**	**0.6190±0.1018**	**0.6458±0.1030**	**0.5077±0.0848**	(-)
M5	RMSN	35.7795±4.3603	33.2570±2.3870	33.4138±4.0678	33.4169±4.4289	33.3104±4.4017	33.3819±4.1602	33.7599±3.9379	52.2%
	CRN(ERM)	13.8445±0.1550	15.7926±0.1278	16.7071±0.2240	17.0887±0.1724	17.2709±0.0923	17.9759±0.0880	16.4466±0.1367	1.8%
	CRN	13.5907±0.0859	15.5242±0.0692	16.2694±0.1157	16.7355±0.0719	17.0095±0.0388	17.6874±0.0558	16.1361±0.0673	-0.1%
	CT(ERM)	**13.4887±0.1335**	15.3397±0.2922	16.3415±0.4096	17.0545±0.5603	17.4828±0.5609	18.5832±0.4930	**15.9414±0.3804**	-1.3%
	CT	13.6721±0.3574	15.9384±1.0910	17.2781±1.6049	18.1796±1.8134	18.8805±2.2159	19.2510±1.9642	16.7897±1.4144	3.8%
	G-Net	13.7187±0.0833	**14.9851±0.1205**	**15.9578±0.1701**	16.8278±0.2229	17.4833±0.3111	18.1665±0.3795	16.1898±0.2070	0.3%
	COSTAR	14.2556±0.1792	15.6151±0.2287	16.1743±0.2076	**16.4791±0.1276**	**16.9037±0.1322**	**17.4379±0.1845**	16.1443±0.1742	(-)

H Visualization of the Learned Representations

Fig. 3 depicts the representations learned for the Semi-synthetic MIMIC-III dataset after each of the 4 stages: (a) **Pre-trained**: representations after the self-supervised learning stage of source data. (b) **Non-Cold-Start**: representations fine-tuned with factual outcome estimation loss of source data. (c) **Cold-Start**: representations of target data when directly applying the encoder trained in (b). (d) **Transfer**: representations of target data after fine-tuned with small amount of target data. We use T-SNE to map each representation to a 2D space and color each point with values of its upcoming treatment and outcomes.

As shown in the first row, representations with different types of upcoming treatments overlap, indicating that the learned representations after each stage are balanced towards treatments. In the second and the third rows, we observe clusters of representations corresponding to similar outcome values, which indicates that the learned representations are informative about the upcoming outcomes, even including the representations trained only with self-supervised loss (column "Pretrained"). Such clustered structures also persist when moving from the source domain data (column "Non-Cold-Start") to the target domain data ("Cold-Start"), showing that the learned representations can generalize to cold-start cases.

I Examples of counterfactual treatment outcome estimation

Fig. 4 qualitatively compare the counterfactual outcome estimation performance differences between COSTAR and baselines in the zero-shot transfer setting. We randomly select a sequence from the observed data until time $t = 4$ (x-axis), then apply sequences of treatments sampled uniformly (i.e. no treatment bias) and simulate the step-wise outcomes for 10 times as the ground truth. We compare the ground truth of each simulation with all methods tested with semi-synthetic MIMIC-III dataset. In Fig. 4 we find that the gaps between estimations and ground truth outcomes are obvious in columns of baseline results. Instead, they closely match each other in the estimation results (the rightmost column) given by COSTAR, demonstrating its superior performance.

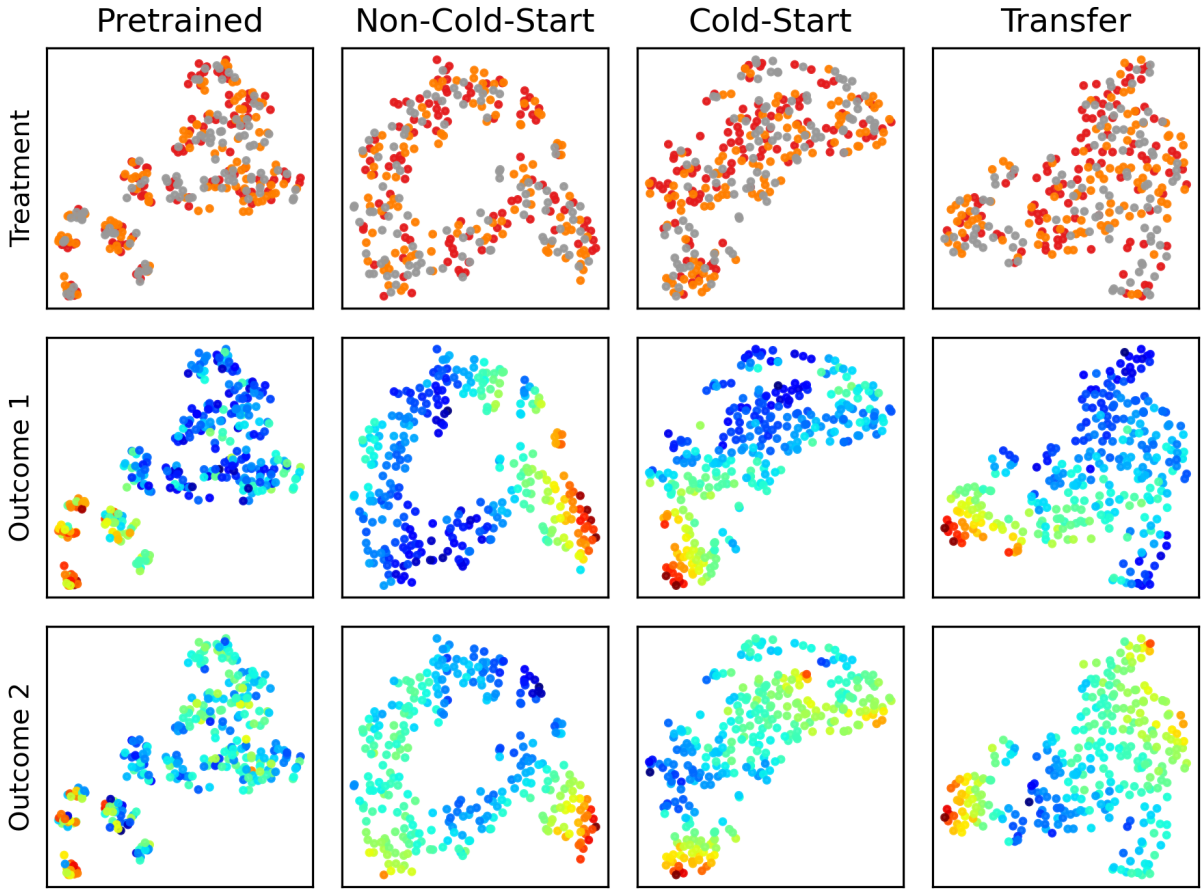


Figure 3: T-SNE visualization of learned representations in Semi-synthetic MIMIC-III dataset.

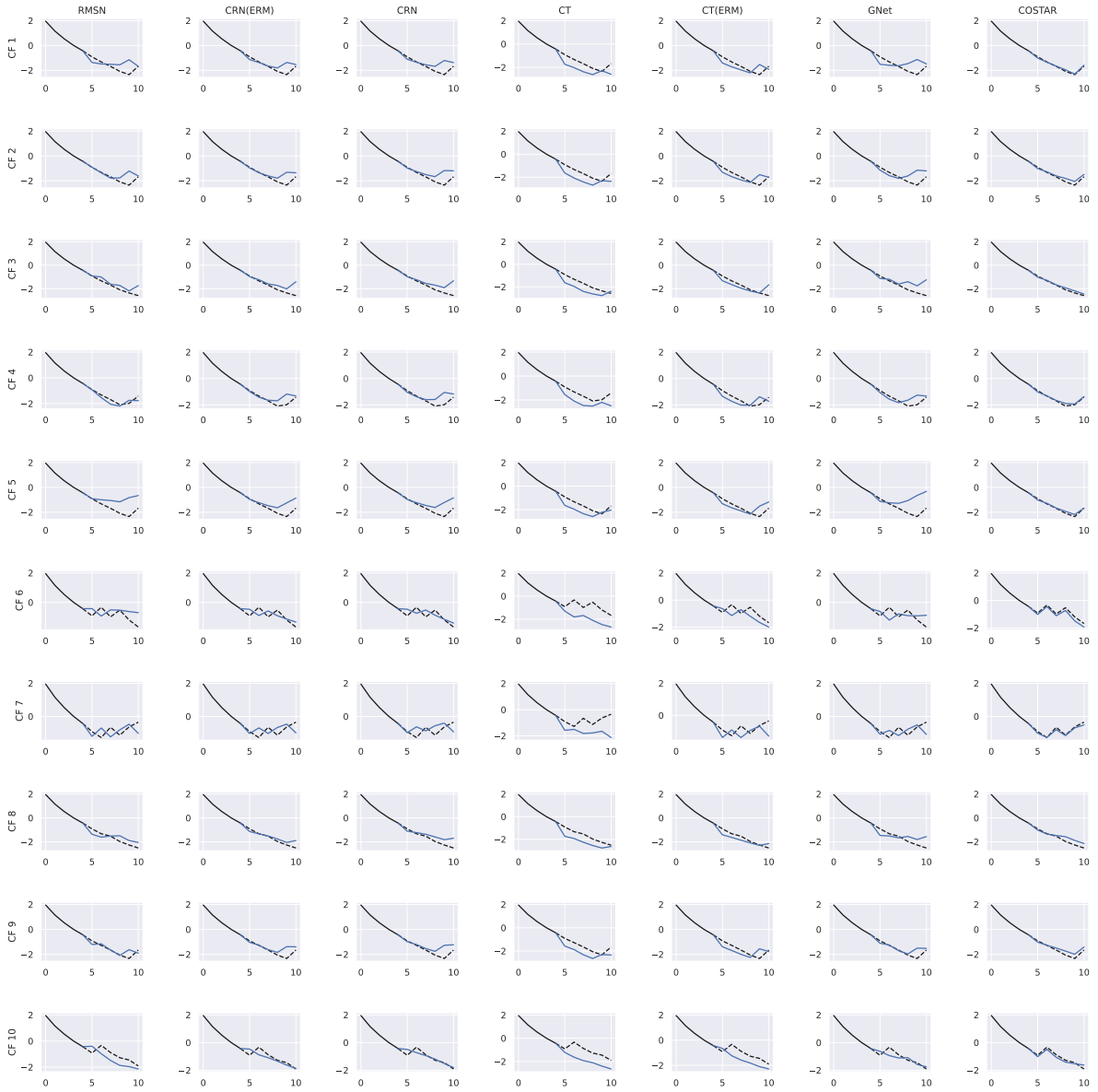


Figure 4: Examples of counterfactual treatment outcome estimation with semi-synthetic MIMIC-III data in the zero-shot transfer setting. We plot one of the two output dimensions for clarity. Each row lists the results of a counterfactual treatment sequence, while each column shows the estimations of one method across all treatment sequences tested. In each sub-figure, the observed historical outcomes are plotted in black solid lines, and the ground truth counterfactual outcomes in black dash lines. The blue solid lines show the estimated outcomes.