COSTAR: Improved Temporal Counterfactual Estimation with Self-Supervised Learning

Anonymous authors
Paper under double-blind review

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[1]. 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 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

[1] We leave a more detailed review of related work in Sec. 2 of the appendix.

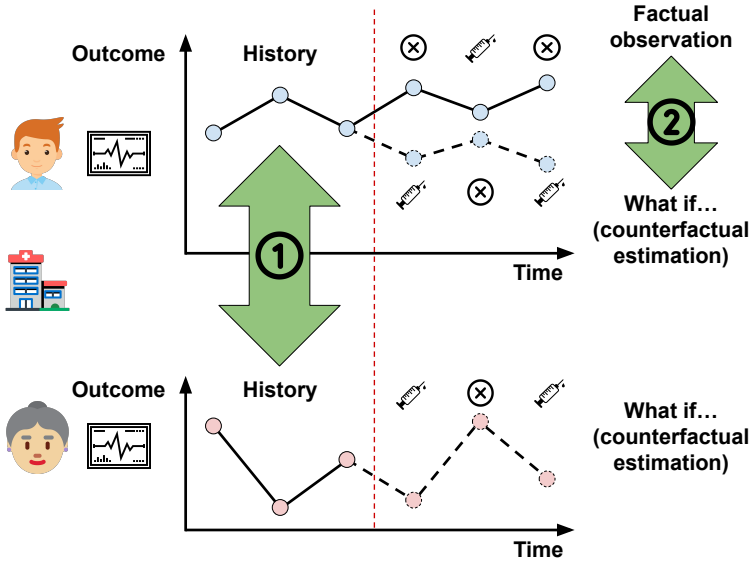


Figure 1: We illustrate the problem of treatment outcome estimation over time with an example in healthcare. We propose COSTAR as a temporal counterfactual estimator enhanced with self-supervised learning, inducing transferrability to both ① cold-start cases from unseen subpopulations and ② counterfactual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

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.

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

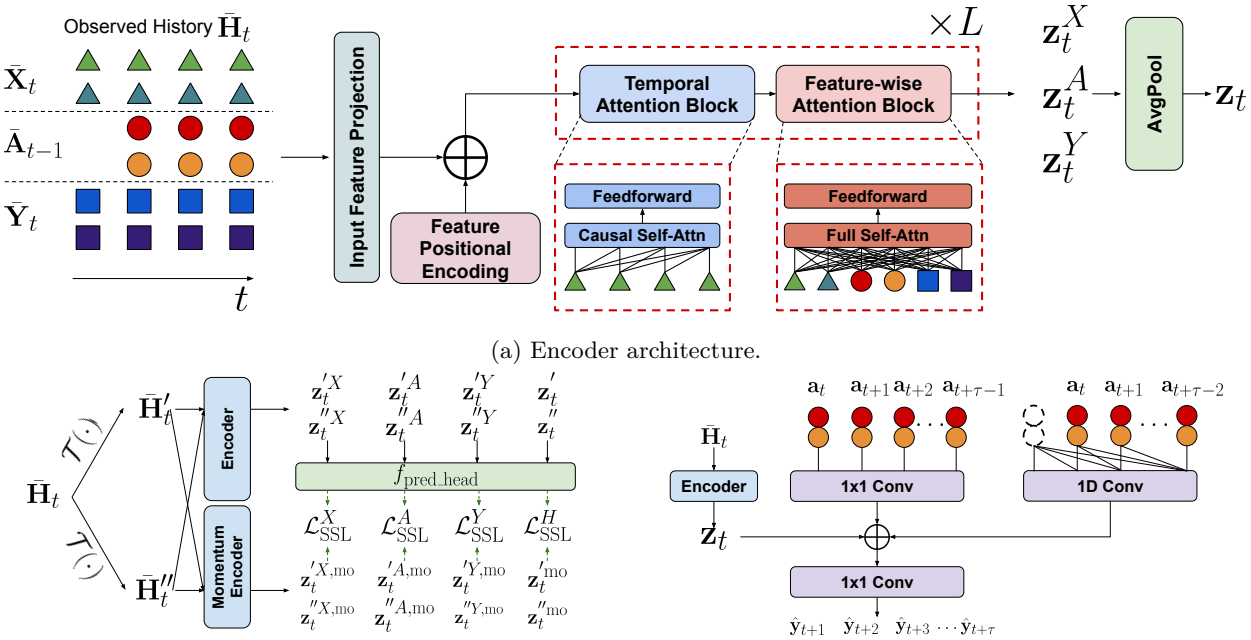
Counterfactual treatment outcome estimation over time. Early works in counterfactual treatment outcome estimation were first developed for epidemiology and can be considered under 3 major groups: G-computation, marginal structural models (MSMs), and structural nested models (Robins, 1986; 1994; Robins 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



(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.

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 5 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.

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[2]. 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[3], we notice that the counterfactual outcome

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

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

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[4]: $P_{\mathcal{D}}(h) = P_{\mathcal{D}_{tr}}(h) = \sum_{a'} P(a') P_{\mathcal{D}_{tr}}(h|a')$, $P_{\mathcal{D}}(y[a]|h) = P_{\mathcal{D}_{tr}}(y[a]|h) = P_{\mathcal{S}_a}(y[a]|h) = P_{\mathcal{T}_a}(y[a]|h)$. 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.

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

[4]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).

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 5 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 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.

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.

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

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

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-

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

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.

5.4 Standard Supervised Learning Setup

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

5.5 Visualization of Learned Representations

Fig. 3 depicts the representations learned for the Semi-synthetic MIMIC-III dataset after each of the 4 stages: (a) **Pretrained**: 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.

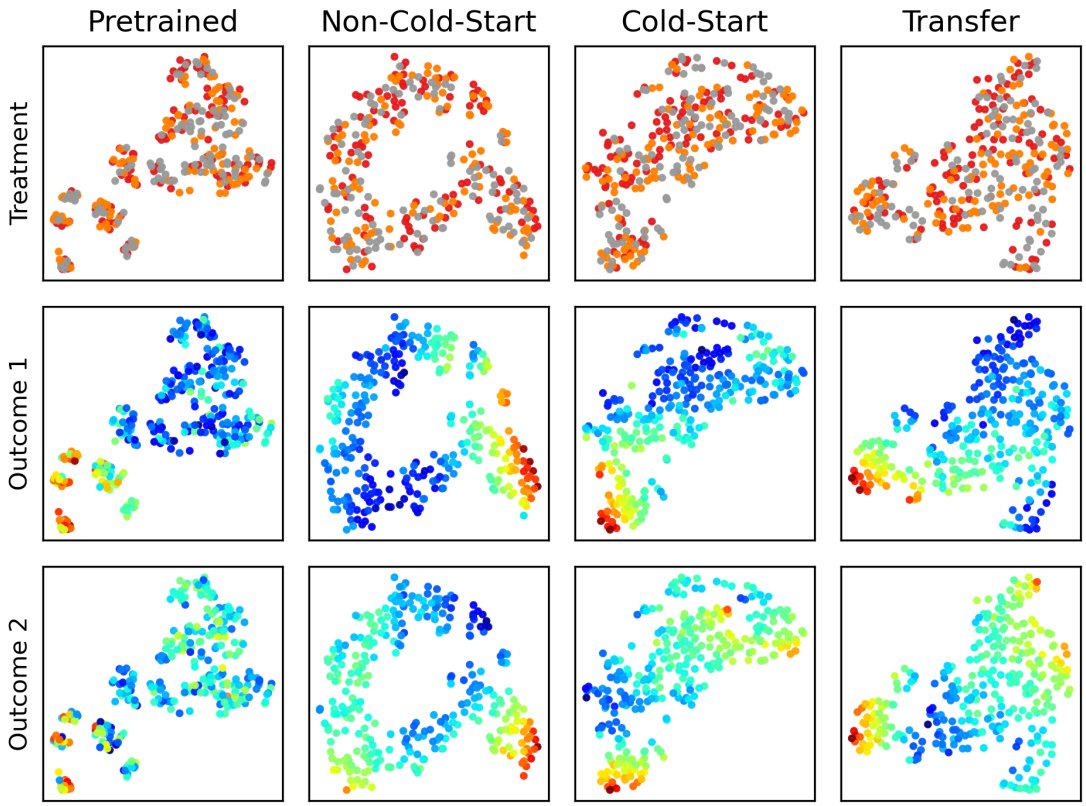


Figure 3: T-SNE visualization of learned representations in Semi-synthetic MIMIC-III dataset.

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.