Leveraging a Simulator for Learning Causal Representations from Post-Treatment Covariates for CATE

Anonymous authors
Paper under double-blind review

Abstract

Treatment effect estimation involves assessing the impact of different treatments on individual outcomes. Current methods estimate Conditional Average Treatment Effect (CATE) using observational datasets where covariates are collected before treatment assignment and outcomes are observed afterward, under assumptions like positivity and unconfoundedness. In this paper, we address a more realistic scenario where both covariates and outcomes are gathered after treatment. We show that post-treatment covariates render CATE unidentifiable, and recovering CATE requires learning treatment-independent causal representations. Prior work shows that such representations can be learned through contrastive learning if counterfactual supervision is available in observational data. However, since counterfactuals are rare, other works have explored using simulators that offer synthetic counterfactual supervision. Our goal in this paper is to systematically analyze the role of simulators in estimating CATE. We analyze the CATE error of several baselines and highlight their limitations. We then establish a generalization bound that characterizes the CATE error from jointly training on real and simulated distributions, as a function of the real-simulator mismatch. Finally, we introduce SimPONet, a novel method whose loss function is inspired from our generalization bound. We further show how SimPONet adjusts the simulator's influence on the learning objective based on the simulator's relevance to the CATE task. We experiment with various DGPs, by systematically varying the real-simulator distribution gap to evaluate *SimPONet*'s efficacy against state-of-the-art CATE baselines.

1 Introduction

In Conditional Average Treatment Effect (CATE) task, the goal is to estimate the difference in an outcome Y, as an individual Z is subject to different treatments T. The gold standard to estimating such effects is Randomized Control Trials, which are often expensive, and with the easy availability of observational data, there is extensive interest in harnessing them for deriving these estimates. The first step in estimating treatment effects from observational datasets is to determine the set of covariates that, when conditioned upon, make the treatment effects identifiable. Prior works (Künzel et al., 2019; Nie & Wager, 2021; Curth & van der Schaar, 2023; Shalit et al., 2017; Nie et al., 2021; Shalit et al., 2016; Chauhan et al., 2023; Curth & van der Schaar, 2021; Zhang et al., 2020; Shi et al., 2019; Stuart, 2010), assume that such covariates are observed, and gathered prior to treatment, with outcomes Y observed after the treatment is given. However, collecting such datasets is challenging as it requires tracking the same individuals over two distinct time points. As a result, readily available observational datasets often contain both covariates X and the outcomes Y recorded together post-treatment for individuals characterized by a latent representation Z. For example, in economics, policymakers implement different taxation policies T aimed at improving an outcomes Y like Gross Domestic Expenditure of the individuals Z. To identify effective policies, they may rely on domain expertise or conduct small-scale RCTs where collecting pre-treatment covariates is feasible. However, the true effectiveness of a policy is only revealed through large-scale testing on the population after its implementation. Collecting such datasets typically involves obtaining post-treatment covariates X and their associated outcomes Y together (Ashenfelter, 1978; Angrist, 1995). Even in other domains such as voluntary healthcare surveys, only post-treatment data about patients might be accessible. In medical imaging, an image taken under a specific instrument setting (treatment) may be evaluated to determine whether switching to a different setting would improve a subsequent diagnosis (outcome).

We present our setup in the top panel of Fig. 1 marked RealDGP (Data Generating Process for the real distribution), where the latent variables Z causally produce the observed treatment T, outcome Y, and covariates X. We begin this paper by presenting an impossibility result in this context.

Lemma 1. *The causal effect of T on Y is not identifiable given i.i.d. samples of the observed nodes from the real DGP shown in top panel of Fig. 1.*

proof. Since X is a collider, conditioning on it opens the backdoor path $T \to X \leftarrow Z \to Y$. Furthermore, as Z is latent, this backdoor path remains open, making CATE unidentifiable from X, T, and Y alone (Pearl, 2015).

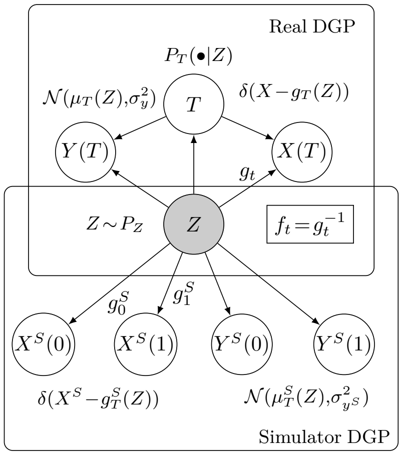

Figure 1: The Data Generating process for Real and Simulator.

The main takeaway from the lemma is that certain additional assumptions are unavoidable for achieving identifiability. The lemma further emphasizes that the key to identifiability lies in extracting treatment-independent causal representations from the post-treatment X that affect Y. One such assumption that allows for the recovery of causal representations is counterfactual supervision in real data. Prior work (Von Kügelgen et al., 2021) demonstrates that, under such an assumption, contrastive losses can be applied to pairs of covariates that differ by treatment to extract Z from X and T. While some works (Nagalapatti et al., 2022; Bachman et al., 2019) assume direct access to such counterfactual supervision in real data, others rely on simulators that generate high-quality synthetic counterfactuals (Von Kügelgen et al., 2021; Zimmermann et al., 2021). However, these are strong assumptions since counterfactuals are rarely available in real-world scenarios, and while simulators are more feasible, assessing their quality or relevance to the downstream task during training is challenging. Therefore, our goal in this paper is to leverage simulators only to the extent they remain relevant to the CATE task. We conduct a theoretical analysis to derive generalization bounds that show how CATE error worsens as the mismatch between real and simulated distributions increases. These insights motivate our proposed algorithm, *SimPONet*, which uses simulated data to apply regularizers inspired by our generalization bound. *SimPONet*'s aim is to enhance CATE estimates beyond what is achievable with observational data alone. Through experiments, we systematically vary the distributional gap between real and synthetic data across various DGPs, demonstrating that *SimPONet* consistently outperforms multiple baselines in estimating CATE.

Contributions: 1) We address Treatment Effect Estimation with post-treatment covariates —- a *non-identifiable* challenge – by leveraging a simulator that offers synthetic counterfactual supervision. 2) We assess the CATE errors for three baselines that can be trained on real/simulated data, and highlight their limitations. 3) Next, we consider a joint training framework, and derive a generalization bound that characterizes the CATE error as a function of real-simulator distributional mismatch. 4) This analysis motivates our method, *SimPONet*, a novel algorithm that uses simulated samples to improve CATE estimates beyond what can be achieved from observational data alone. 5) To our knowledge, this is the *first* systematic study on the role of simulators in CATE estimation. 6) Experiments across various DGPs confirm *SimPONet*'s effectiveness.

2 Related Work

CATE estimation with post-treatment covariates is not explored much in the literature. We provide a comprehensive discussion of related work on CATE methods with both pre-treatment and post-treatment covariates in Appendix B, along with real-world examples of how simulators are used in pharmacology and electrochemistry to assess CATE from post-treatment covariates in Appendix B.3.

3 Problem Formulation

We use random variables X,T,Y to denote post-treatment covariates, binary treatments, and outcomes respectively. The observational dataset has n samples: $D_{\text{trn}} = \{(\boldsymbol{x}_i, t_i, y_i)\}_{i=1}^n$ where $t_i \in \mathcal{T} = \{0,1\}$ denotes treatment, $\boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^{n_x}$ denotes covariates observed after t_i is applied, and y_i the resulting outcome. We use the Neyman-Rubin potential outcomes framework to denote $Y_i(t), X_i(t)$ as the potential outcome and covariate for unit i under a treatment t. The main challenge is the absence of counterfactuals in D_{trn}, i.e., for each unit i, we observe covariates and outcomes under only one treatment t_i.

We use the random variable $Z \in \mathcal{Z} \subset \mathbb{R}^{n_z}$ to denote the causal representations of covariates X. Z generates X via treatment-specific covariate generating functions $g_t : \mathcal{Z} \mapsto \mathcal{X}$ for $t \in \{0,1\}$. We assume that g_t is diffeomorphic (Locatello et al., 2019a;b; Von Kügelgen et al., 2021); i.e., it is smooth, invertible, and has a smooth inverse. Diffeomorphism ensures that all factors involved in generating X are preserved within X so that there exists inverse functions $f_t : \mathcal{X} \mapsto \mathcal{Z}, \forall t \in \{0,1\}$ that could recover the causal representations Z back. A sample is obtained from the real DGP as follows: (1) $z_i \sim P_Z$, (2) $t_i \sim P(T|z_i)$, (3) $\boldsymbol{x}_i \sim P(X|z_i,t_i) = \delta(X - g_{t_i}(z_i))$, where δ denotes the dirac-delta distribution, (4) $y_i \sim P(Y|z_i,t_i) = \mathcal{N}(\mu_{t_i}(z_i), \sigma_y^2)$ is sampled from a Gaussian with mean $\mu_{t_i}(z_i)$ and constant variance σ_y^2. Here, $\mu_t : \mathcal{Z} \mapsto \mathcal{Y}\ \forall t$ generates responses for individuals with latent representations z under treatment t. We express the *factual* observed outcome for i as $Y_i(t_i) = \mu_{t_i}(f_{t_i}(\boldsymbol{x}_i))$, and the missing *counterfactual* (CF) outcome under $1-t_i$ as $Y_i(1-t_i) = \mu_{1-t_i}(f_{t_i}(\boldsymbol{x}_i))$.

Our Goal is Conditional Average Treatment Effect (CATE) estimation which quantifies the difference in outcomes due to a change in treatment. Given a test unit $(\boldsymbol{x}_j, t_j)$, its CATE is given by $\tau_j = \mathbb{E}[Y_j(T = 1) - Y_j(T = 0)|\boldsymbol{x}_j, t_j]$. As argued earlier, estimating τ from post-treatment data involves the sub goal of learning causal representations of observed covariates X using a function $f_t : X \mapsto Z$. We use $\tau : \mathcal{Z} \mapsto \mathcal{Y}$ to express the treatment effect using the latent z_j as $\tau(z_j) = \mu_1(z_j) - \mu_0(z_j)$. Since f_t inverts X to give Z, the same effect can also be expressed for $(\boldsymbol{x}_j, t_j)$ using τ_X as $\tau_X(\boldsymbol{x}_j, t_j) = \mu_1(f_{t_j}(\boldsymbol{x}_j)) - \mu_0(f_{t_j}(\boldsymbol{x}_j))$

where $\tau_X : \mathcal{X} \times \mathcal{T} \mapsto \mathcal{Y}$. Notice that $\tau_X(\bullet,t) = \tau \circ f_t(\bullet)$. When estimating τ_X, the factual outcome is easy, all we need to do is fit a regression model on the observation data. The main challenge lies in estimating the counterfactual outcome under treatment $1-t_j$.

Theorem 1 in (Locatello et al., 2019a) presents an impossibility result stating that f_t which maps covariates X to their treatment-independent causal representations Z is *not* identifiable solely using D_{trn}. The main hurdle is that multiple DGPs can yield the same marginal distribution $P(X,T)$, making it impossible to isolate the true DGP. However, prior work has shown how to learn f_t with *counterfactuals*, requiring that D_{trn} includes both covariates $X_i(t_i)$ and $X_i(1-t_i)$. Theorem 4.4 of (Von Kügelgen et al., 2021) shows that such counterfactual supervision allows for recovery of Z up to a diffeomorphic transformation h using contrastive learning. Proposition 2 in (Zimmermann et al., 2021) further shows that h is, in particular, a rotation in an n_z dimensional unit-normalized hypersphere. While some prior works assume direct access to counterfactual supervision in real data (Nagalapatti et al., 2022; Bachman et al., 2019), others rely on high-quality synthetic counterfactuals from simulators (Xie, 2018; Kaur et al., 2021). In contrast, our approach seeks to leverage simulators only to the extent that they improve the downstream CATE task. We next formally define the simulator's data generating process.

Simulator DGP The simulator generates paired instances giving rise to a counterfactual dataset $D_{\text{syn}} = \{\boldsymbol{x}_i^S(0), \boldsymbol{x}_i^S(1), y_i^S(0), y_i^S(1)\}$ generated using the DGP as shown in the lower panel in Figure 1. The simulated instances are obtained as follows: (1) $z_i \sim P_Z$; i.e., Z is sampled from the same distribution as real, (2) post-treatment covariates $\boldsymbol{x}_i^S(t) \sim P(X^S | Z = z_i, T = t) = \delta(X^S - g_t^S(z_i))$ under *both* treatments $t = \{0,1\}$. $g_t^S : \mathcal{Z} \mapsto \mathcal{X}\ \forall t$ are diffeomorphic functions, and (3) corresponding outcomes $y_i^S(0)$, $y_i^S(1)$ are sampled from $P(Y^S | Z = z_i, T = t) = \mathcal{N}(\mu_t^S(z_i), \sigma^2_{y^S})$, where $\mu_t^S : \mathcal{Z} \mapsto \mathcal{Y}, \forall t$. Note that z_i remains hidden even in D_{syn}. We use "S" in the superscript to indicate a simulator component. Now we describe some metrics that assess the distance between real and simulator DGP.

Definition 1 [$d_{\boldsymbol{x}|t}(f_t, f_t^S)$] We assess the distance between the real and synthetic causal representation extractors f_t and f_t^S using the following expected distance: $d_{\boldsymbol{x}|t}(f_t, f_t^S) = \mathbb{E}_{\boldsymbol{x} \sim P(X|t)}\left[||f_t(\boldsymbol{x}) - f_t^S(\boldsymbol{x})||_2^2\right]$.

Definition 2 [$d_z(\tau, \tau^S)$] We assess the distance between the real and simulator CATE functions on the P_Z distribution as: $d_z(\tau, \tau^S) = \mathbb{E}_{z \sim P_Z}\left[(\tau(z) - \tau^S(z))^2\right]$. Under composition with a diffeomorphic function h, we write $d_h(\tau, \tau^S) = \mathbb{E}_{z \sim P_Z}\left[(\tau(h(z)) - \tau^S(h(z)))^2\right]$.

Assumptions for Identifying CATE τ_X. We summarise the assumptions that are needed on the real dataset D_{trn} and simulated counterfactual dataset D_{syn} to identify the CATE function τ_X: (A1) *Positivity:* $P(T = t | Z = z) > 0,\ \forall t \in \mathcal{T}, z \in \mathcal{Z}$. (A2) *Diffeomorphic Covariate Generation:* Covariates in both real and synthetic distributions are obtained through diffeomorphic transformations of Z under any treatment T. (A3) *Identifiability of* τ *given* Z*:* The causal factors Z that generate X form a sufficient adjustment set, blocking backdoor paths between T and Y, thus making τ identifiable from Z. Note that A2 and A3 together ensure that X contains information about all the relevant latent factors that affect the outcome Y and is a weaker notion of the commonly used *unconfoundedness* assumption.

CATE Error ($\mathcal{E}_{\text{CATE}}$). Given a test dataset $D_{\text{tst}} = \{(\boldsymbol{x}_j, t_j, y_j(0), y_j(1))\}_{j=1}^m$, with each $\boldsymbol{x}_j$ rendered under t_j, we compute the empirical error incurred in estimating CATE using mean squared error as $\mathcal{E}_{\text{CATE}} = \frac{1}{m}\sum_{j \in D_{\text{tst}}}[\tau_j - \widehat{\tau}_j]^2$ where $\tau_j = y_j(1) - y_j(0)$ is the true effect and $\widehat{\tau}_j$ is the predicted effect for the instance $(\boldsymbol{x}_j, t_j)$.

The CATE error in general can be decomposed across treatment T as

$$\mathcal{E}_{\text{CATE}} = \sum_{t\in\mathcal{T}} P(T=t)\ \mathcal{E}^t_{\text{CATE}} \quad \text{where } \mathcal{E}^t_{\text{CATE}} = \int_{\boldsymbol{x}\in\mathcal{X}} [\tau_X(\boldsymbol{x},t) - \widehat{\tau_X}(\boldsymbol{x},t)]^2 P(\boldsymbol{x}|t) d\boldsymbol{x}$$

Definition 3. Let us define *factual* error $\mathcal{E}^t_F$ and *counterfactual* error $\mathcal{E}^t_{CF}$ on samples with observed treatment t and missing treatment $1-t$ as follows:

$$\mathcal{E}^t_F = \int_{\boldsymbol{x}\in\mathcal{X}} [\mu_t(f_t(\boldsymbol{x})) - \widehat{\mu}_t(\widehat{f}_t(\boldsymbol{x}))]^2 P(\boldsymbol{x}|t) d\boldsymbol{x} \text{ and } \mathcal{E}^t_{CF} = \int_{\boldsymbol{x}\in\mathcal{X}} [\mu_{1-t}(f_t(\boldsymbol{x})) - \widehat{\mu}_{1-t}(\widehat{f}_t(\boldsymbol{x}))]^2 P(\boldsymbol{x}|t) d\boldsymbol{x}$$

Lemma 2. *The CATE error is related to the factual and counterfactual error as:* $\mathcal{E}^t_{CATE} \leq 2\mathcal{E}^t_F + 2\mathcal{E}^t_{CF}$ *[Proof in Appendix B.6.1]*

4 Learning Causal Representations for CATE

Our task involves learning four functions: $\widehat{f}_t$ that extracts the causal representations from $X(t)$ and $\widehat{\mu}_t$ that estimates the outcomes $Y(t)$ for $t\in\{0,1\}$. With access to *counterfactual simulated* data D_{syn} and *observational real* data D_{trn}, one can come up with the following approaches for estimating CATE: 1) SimOnly, which only uses D_{syn}, and 2) RealOnly which only uses D_{trn} to estimate μ_t, *(3)* $\text{Real}_\mu\text{Sim}_f$, which uses D_{syn} to estimate f_t and subsequently, D_{trn} to estimate μ_t. We now discuss the training approach for each of these methods, delve into their shortcomings, and then present our proposed method *SimPONet*.

To illustrate the shortcomings, we consider a test instance $\boldsymbol{x}^\star$ generated under treatment $T=1$ (without loss of generality) and derive the CATE error expression for it in the population setting, as $|D_{\text{trn}}| \to \infty$ and $|D_{\text{syn}}| \to \infty$.

4.1 The SimOnly Estimator

SimOnly solely uses D_{syn}. It leverages the counterfactual supervision provided by the simulator and identifies the simulator's DGP as follows:

(Step 1) Estimate the synthetic causal representation extractor f^S from covariate pairs $\{\boldsymbol{x}^S_i(0), \boldsymbol{x}^S_i(1)\}$ using contrastive learning Von Kügelgen et al. (2021):

$$\{\widetilde{f}^S_0, \widetilde{f}^S_1\} = \underset{\{\widehat{f}^S_0, \widehat{f}^S_1\}}{\text{argmin}} \sum_{i=1}^{|D_{\text{syn}}|} \left[-\log \frac{\exp(\text{sim}(\hat{z}_i(1), \hat{z}_i(0))}{\sum_{j\neq i}\sum_{t,t'} \exp(\text{sim}(\hat{z}_i(t), \hat{z}_j(t')))} \right] \text{ where } \hat{z}_i(t) = \widehat{f}^S_t(\boldsymbol{x}^S_i(t)) \quad (1)$$

where $\text{sim}(\bullet,\bullet)$ is cosine similarity, $(\boldsymbol{x}^S_i(t), \boldsymbol{x}^S_i(1-t))$ denotes a positive pair with the same underlying latent z_i. A negative pair $(\boldsymbol{x}^S_i(t), \boldsymbol{x}^S_j(t'))$ has different (z_i, z_j). Contrastive learning increases similarity of representations of positive pairs $(\hat{z}_i(0), \hat{z}_i(1))$ while pushing apart the negative pairs $(\hat{z}_i(t), \hat{z}_{j\neq i}(t'))$.

Lemma 3. *As* $|D_{syn}| \to \infty$*, contrastive training with paired counterfactual covariates as shown in Eq. 1 recovers* $\widetilde{f}^S_t = h \circ f^S_t$ *where* h *is a diffeomorphic transformation. Moreover, when the latent space* $\mathcal{Z} \subset \mathbb{S}^{(n_z-1)}$ *(unit-norm hypersphere in* $\mathbb{R}^{n_z}$*),* h *is a rotation transform by Extended Mazur-Ulam Theorem as shown in (Zimmermann et al., 2021) (Proposition 2).*

[Refer Appendix B.6.2 for more details.]

The main insight from the above lemma is that, given counterfactual supervision, it is possible to recover causal representations Z from post-treatment covariates X up to a rotation h, making CATE identifiable in the simulated distribution, as we demonstrate below.

(Step 2) Estimate $\widetilde{\tau}^S(z) = \widetilde{\mu}_1^S(z) - \widetilde{\mu}_0^S(z)$ with supervision on difference of outcomes $\tau^S(f_t^S(\boldsymbol{x}_i^S(t))) = y_i^S(1) - y_i^S(0)$ as

$$\widetilde{\tau}^S = \operatorname*{argmin}_{\widehat{\tau^S}} \sum_{\boldsymbol{x}^S \in D_{\text{syn}}} \left[\tau^S(f_t^S(\boldsymbol{x}^S(t))) - \widehat{\tau^S}(\widetilde{f}_t^S(\boldsymbol{x}^S(t)))\right]^2 \tag{2}$$

The SimOnly method uses these estimates as-is on real data, i.e. $\widehat{\tau} = \widetilde{\tau}^S$ and $\widehat{f}_t = \widetilde{f}_t^S,\ \forall t \in \mathcal{T}$. We analyze below the error incurred with such estimates on real data.

CATE error: In the population setting, since $\widetilde{f}_t^S = h \circ f_t^S$, we see that the optimization problem in Eq. 2 yields $\widetilde{\tau}^S = \tau^S \circ h^{-1}$ as its solution. Thus, for an instance $\boldsymbol{x}^\star$ from the real distribution under treatment 1, the true CATE is $\tau(f_1(\boldsymbol{x}^\star))$. The CATE error using SimOnly becomes:

$$\left[\tau(f_1(\boldsymbol{x}^\star)) - \tau^S \circ h^{-1}(h \circ f_1^S(\boldsymbol{x}^\star))\right]^2 = \left[\tau(f_1(\boldsymbol{x}^\star)) - \tau^S(f_1^S(\boldsymbol{x}^\star))\right]^2.$$

This shows that for SimOnly to provide accurate CATE estimates, the simulator must perfectly align with the real world; i.e., $\tau^S = \tau$ and $f_t = f_t^S$ for all t. However, designing such simulators is highly challenging in practice, making this method unsuitable for CATE.

4.2 The RealOnly Estimator

RealOnly solely uses real observational data D_{trn}. Since D_{trn} lacks counterfactual covariates, this model cannot apply contrastive training and therefore cannot explicitly supervise the recovery of causal representations. Instead, it focuses on regressing the factual outcomes $y_i(t_i)$ from post-treatment covariates $\boldsymbol{x}_i(t_i)$. In terms of the four learning parameters, its learning objective is:

$$\operatorname*{argmin}_{\{\widehat{\mu}_0, \widehat{\mu}_1, \widehat{f}_0, \widehat{f}_1\}} \sum_{i=1}^{|D_{\text{trn}}|} (y_i - \widehat{\mu}_{t_i}(\widehat{f}_{t_i}(\boldsymbol{x}_i)))^2$$

However, since $\widehat{\mu}_t, \widehat{f}_t$ are not individually supervised, we might as well collapse them into a composition $\mu_t^F = \mu_t \circ f_t$; yielding $\mu_{t_i}^F(\boldsymbol{x}_i) = y_i$, and thereby CATE as $\widehat{\tau_X}(\boldsymbol{x}, t) = \widehat{\mu}_1^F(\boldsymbol{x}) - \widehat{\mu}_0^F(\boldsymbol{x})$.

RealOnly is consistent in estimating the factual outcomes, because as $|D_{\text{trn}}| \to \infty$, we have $\widehat{\mu}_t^F = \operatorname{argmin}_{\widehat{\mu}_t^F} \mathbb{E}_{\boldsymbol{x} \sim P(\boldsymbol{x}|t)} \left[\left(\widehat{\mu}_t^F(\boldsymbol{x}) - \mu_t^F(\boldsymbol{x})\right)^2\right] = \mu_t^F$ and therefore, the factual error $\mathcal{E}_F^t = 0$. However, RealOnly incurs a significant error when estimating the counterfactual outcome, which in turn contributes to the CATE error, as shown below.

CATE error: The true CATE for $\boldsymbol{x}^\star$ obtained using treatment 1 can be written as $\tau(f_1(\boldsymbol{x}^\star)) = \mu_1(f_1(\boldsymbol{x}^\star)) - \mu_0(f_1(\boldsymbol{x}^\star))$. Then, the CATE error for RealOnly is computed as:

$$\left[\left(\mu_1(f_1(\boldsymbol{x}^\star)) - \mu_0(f_1(\boldsymbol{x}^\star))\right) - \left(\widehat{\mu}_1^F(\boldsymbol{x}^\star) - \widehat{\mu}_0^F(\boldsymbol{x}^\star)\right)\right]^2 = \left[\left(\mu_1(f_1(\boldsymbol{x}^\star)) - \widehat{\mu}_1^F(\boldsymbol{x}^\star)\right) - \left(\mu_0(f_1(\boldsymbol{x}^\star)) - \widehat{\mu}_0^F(\boldsymbol{x}^\star)\right)\right]^2$$

In the population setting, using the consistency of factual estimates, the CATE error reduces to $\big[\mu_0(f_1(\boldsymbol{x}^\star)) - \mu_0^F(\boldsymbol{x}^\star)\big]^2$. This error is zero when $f_1(\boldsymbol{x}^\star) = f_0(\boldsymbol{x}^\star)$. Thus, for RealOnly to provide accurate CATE estimates, the treatment must not affect the post-treatment covariates, i.e., $g_0(z) = g_1(z)\ \forall z$ in which case their inverse are equal $f_0 = f_1$. However, this assumption is often unrealistic. For instance, in pharmacology, different drugs typically induce distinct effects on patient covariates, limiting the applicability of this model in such settings.

4.3 The Real$_\mu$Sim$_f$ Estimator

Unlike SimOnly, which uses D_{syn} to learn both $\widehat{f}_t$ and $\widehat{\mu}_t$, this approach leverages D_{syn} solely to learn the representation extractor $\widehat{f}_t$. Specifically, it assumes that $\widehat{f}_t = \widetilde{f}^S_t$, as obtained from Eq. 1. Thereafter, it learns the $\widehat{\mu}_t$ parameters by applying a factual loss on D_{trn} to estimate

$$\widehat{\mu}_0, \widehat{\mu}_1 = \underset{\{\widehat{\mu}_0, \widehat{\mu}_1\}}{\operatorname{argmin}} \sum_{D_{\text{trn}}} (y_i - \widehat{\mu}_{t_i}(\widetilde{f}^S_{t_i}(\boldsymbol{x}_i)))^2$$

We call this method Real$_\mu$Sim$_f$ since it learns the outcome parameters μ from real samples while learning representation extractor f_t from the simulator.

CATE error: One condition under which the Real$_\mu$Sim$_f$ model achieves zero CATE error is when $\widetilde{f}^S_t = f_t$ for each treatment t. This requires that the simulator aligns with real-world covariates, specifically $\boldsymbol{x}_t = g_t(z) = g^S_t(z) = \boldsymbol{x}^S_t$. This limitation arises because the model learns the representation extractor solely from D_{syn}, without making adjustments for real covariates.

In summary, we described three possible CATE estimators and showed that each method would provide accurate CATE estimates under certain strong assumptions about the real and simulator DGPs. Given that none of these assumptions would hold in practice, we now turn to exploring a joint training framework that learns simultaneously from both real and simulated samples.

4.4 The *SimPONet* Estimator

We first conduct a theoretical analysis to derive a generalization bound that characterizes the CATE error as a function of the mismatch between the real and simulator distributions. This analysis forms the basis for our proposed method, *SimPONet*, whose loss function is inspired by the bound.

Lemma 4. *Assume τ is K_τ-Lipschitz, and $\widetilde{f}^S$ and $\widetilde{\tau}^S$ are estimates from the simulator DGP obtained from the optimization in Eq. 1, 2. Then, the CATE error on the estimates $\widehat{f}_t$ and $\widehat{\tau}$ admits the following bound:*

$$\mathcal{E}^t_{CATE}(\widehat{f}_t, \widehat{\tau}) \le [8\mathcal{E}^t_F + 12 d_h(\widehat{\tau}, \widetilde{\tau}^S) + 12K_\tau^2 d_{\boldsymbol{x}|t}(\widehat{f}_t, \widetilde{f}^S_t)] + [12 d_z(\tau, \tau^S) + 12K_\tau^2 d_{\boldsymbol{x}|t}(f_t, f^S_t)]$$

where $d_{\boldsymbol{x}|t}, d_z, d_{h(z)}$ are distance functions in Sec. 3 and $\mathcal{E}^t_F$ is the factual loss. [Proof in Appendix B.6.2.]

The expressions in blue are constants that capture the discrepancy between real and simulated distributions and cannot be minimized. In contrast, the remaining terms can be minimized by training on D_{trn} and D_{syn}. As $|D_{\text{trn}}|$ approaches infinity, the factual error $\mathcal{E}^t_F$ can be made to approach zero, while the other minimizable distance terms act as regularizers. The term $d_h(\widehat{\tau}, \widetilde{\tau}^S)$ can assist in regularizing the outcome parameters $\widehat{\mu}_t$, whereas $d_{\boldsymbol{x}|t}(\widehat{f}_t, \widetilde{f}^S_t)$ can aid in regularizing the parameters of the causal representation extractor functions $\widehat{f}_t$. This analysis leads to our proposed approach *SimPONet* whose overall loss is as follows:

$$\min_{\{\widehat{\mu}_t,\widehat{f}_t\}} \underbrace{\sum_{D_{\text{trn}}} \Big(y_i - \widehat{\mu}_{t_i}(\widehat{f}_{t_i}(\boldsymbol{x}_i))\Big)^2}_{\text{Factual Loss on } D_{\text{trn}}} + \underbrace{\lambda_f \sum_{D_{\text{trn}}} \|\widetilde{f}^S_{t_i}(\boldsymbol{x}_i) - \widehat{f}_{t_i}(\boldsymbol{x}_i)\|_2^2}_{d(\widetilde{f}^S_t,\widehat{f}_t) \text{ regularizer}} + \underbrace{\lambda_\tau \sum_{D_{\text{syn}}} \sum_{t\in\{0,1\}} \Big(\tau^S_i - \widehat{\tau}(\widetilde{f}^S_t(\boldsymbol{x}^S_i(t)))\Big)^2}_{\tau^S \text{ regularizer on } D_{\text{syn}}} \tag{3}$$

where $\tau^S_i = y^S_i(1) - y^S_i(0)$ and $\lambda_\tau, \lambda_f > 0$ are loss weights. $\widehat{\tau}(\bullet) = \widehat{\mu}_1(\bullet) - \widehat{\mu}_0(\bullet)$ denotes the estimated CATE.

SimPONet relaxes the strict equality $\widehat{f}_t = \widetilde{f}^S_t$ used by $\text{Real}_\mu\text{Sim}_f$, and instead uses $\widetilde{f}^S_t$ as a regularizer, while ensuring that $\widehat{\mu}_t$ accurately predicts the factual outcomes for instances in D_{trn}. It also imposes the τ^S loss on simulated instances to leverage any potential similarity between the true treatment effect, τ, and the simulated treatment effect, τ^S. Furthermore, the τ^S loss is essential to prevent degenerate solutions that would cause *SimPONet* to collapse to the $\text{Real}_\mu\text{Sim}_f$ estimator. This is because applying regularization solely on $\widehat{f}_t$ can drive the regularizer $||\widehat{f}_t(\boldsymbol{x}) - \widetilde{f}^S_t(\boldsymbol{x})||_2^2$ to zero, leading to $\widehat{f}_t = \widetilde{f}^S_t$, while still minimizing the factual error $\mathcal{E}^t_F$ by updating $\widehat{\mu}_t$ accordingly. Consequently, *SimPONet* would collapse into the $\text{Real}_\mu\text{Sim}_f$ estimator, making the τ^S loss critical in avoiding such degeneracies.

Adjusting Loss Weights. *SimPONet* adjusts the loss weight λ_f for learning $\widehat{f}_t$ by comparing the *factual errors* of the RealOnly model, which trains on X, with those of the $\text{Real}_\mu\text{Sim}_f$ model, trained using simulated causal representations $\widetilde{f}^S_t(\boldsymbol{x})$. If RealOnly consistently outperforms $\text{Real}_\mu\text{Sim}_f$ in factual error, we infer that the simulated representations may not generalize well to the real distribution, prompting *SimPONet* to reduce λ_f. By default, λ_f is set to 1; however, if $\text{Real}_\mu\text{Sim}_f$ exhibits a notably higher factual error, *SimPONet* lowers λ_f to 10^{-4}.

In contrast, tuning λ_τ requires τ supervision on real data, which is unavailable. Prior work (Curth & van der Schaar, 2021; Nagalapatti et al., 2024b; Künzel et al., 2019) argue that while outcome functions μ_t can be complex, the difference function $\tau = \mu_1 - \mu_0$ is often simpler. For instance, if we consider $\mu^S_t = \mu_t + c$ (with $c > 0$), we can make the factual outcomes to diverge arbitrarily while their corresponding τ and τ^S remain equal. Therefore, while comparing the factual errors between SimOnly and RealOnly models to set λ_τ is appealing, it maybe a poor choice in practice. So, *SimPONet* always sets λ_τ to its default 1.

We present the *SimPONet*'s pseudocode in Appendix B.5.

5 Experiments

We conduct experiments that are designed to address the following research questions:

RQ1 How do different methods compare with varying discrepancies between real and simulator in settings with closed form estimates; i.e., without errors due to finite-sample training?

RQ2 How does *SimPONet* compare to other SOTA baselines that assume pre-treatment covariates?

RQ3 What are the contributions of individual loss terms in *SimPONet*?

RQ4 How does *SimPONet* fare against the baselines when μ_t exhibits complex non-linear behavior?

RQ5 How do CATE methods perform when trained directly on (a) post-treatment covariates X, or (b) pre-treatment Z, or (c) simulated causal representations $\widetilde{f}^S$ from Eq. 1?

5.1 Neural Architecture and Hyperparameters

For the Linear experiments in RQ1, we omit shared layers in Fig. 3, and set $\widehat{f}_0$ and $\widehat{f}_1$ as $n_x \times n_x$ matrices, and $\widehat{\mu}_0$ and $\widehat{\mu}_1$ as $n_x \times 1$ vectors. For the real-world experiments in RQ2, with both f_t, μ_t as non-linear functions, we set $\widehat{\mu}_0$ and $\widehat{\mu}_1$ as 2-layer MLPs with hidden layers of 100 and 50 neurons. The shared layers have 2 hidden layers with 50 and n_x neurons, respectively. We impose the $d(\widehat{f}_t, \widetilde{f}_t^S)$ loss for *SimPONet* on outputs of the shared layers. For experiments in RQ4, we omit the shared layers while setting $\widehat{f}_t$ as linear layer. But since μ is non-linear, we use an MLP with one hidden layer of 50 neurons and ReLU activations for each $\widehat{\mu}_t$.

Hyperparameters: We implemented all baselines and *SimPONet* using `jax` within CATENets (Curth et al., 2021), a standard library for benchmarking state-of-the-art CATE estimation methods. To ensure consistency, we used the same MLP architecture, learning rates, optimizers, and other hyperparameters as the default settings in CATENets for baseline approaches. The unique hyperparameters for *SimPONet* are the loss weights λ_f and λ_τ. As described in Sec. 4.4, *SimPONet* tunes λ_f by comparing the factual errors of the RealOnly and $\text{Real}_\mu\text{Sim}_f$ models, while λ_τ is always set to 1. CATENets applies early stopping based on factual error in the validation dataset, a common practice in CATE training. To ensure a fair comparison, we maintained consistent training and validation splits across all methods.

Assessing Statistical Significance: For CATE experiments, standard deviation is sometimes misleading to comment on the statistical significance of empirical results as noted in (Curth et al., 2021). So, for all experiments, we conduct a one-sided paired t-test with *SimPONet* as the baseline and enclose *p-values in brackets* to indicate statistical significance. Lower p-values favor *SimPONet.*

5.2 RQ1: Assessing Baselines Under Settings Without Finite Training Sample Errors

To address RQ1, we consider a setting where both the real and simulator DGPs as shown in Fig. 1 are linear. In particular, we generate the training datasets D_{trn} and D_{syn} as follows: (1) Latent variables $z \in \mathbb{R}^{n_z}$ are sampled from distribution P_Z. (2) Real and simulated covariates for a treatment t are computed as $g_t(z) = z\boldsymbol{R}_t$ and $g_t^S(z) = z\boldsymbol{S}_t$, where $\boldsymbol{R}_t$ and $\boldsymbol{S}_t$ are invertible matrices. (3) Outcomes are generated as $\mu_t(z) = z^\top w_t$ and $\mu_t^S(z) = z^\top w_t^S$, where w_t and w_t^S are vectors in $\mathbb{R}^{n_z}$. We consider two datasets for RQ1: (1) Synthetic-Gaussian and (2) Real World-IHDP, differing in how Z is obtained. In setting (1), $z \in \mathbb{R}^{10}$ is sampled from a standard Gaussian $\mathcal{N}(0,1)$, while for (2), Z is taken from the real-world IHDP dataset (Appendix B.8) as-is.

Now, to systematically control the real-simulator mismatch, we need means to vary the following distances: $d(\boldsymbol{R}_0^{-1}, \boldsymbol{R}_1^{-1})$, $d(\boldsymbol{R}_t^{-1}, \boldsymbol{S}_t^{-1})$, and $d(\tau, \tau^S)$. We achieve this as follows: (1) Initialize $\boldsymbol{R}_0^{-1}, w_0 \sim \mathcal{N}(0,1)$. (2) To inject a distance $\gamma_R \in (0,0.5)$ between $\boldsymbol{R}_0^{-1}$ and $\boldsymbol{R}_1^{-1}$, set $\boldsymbol{R}_1^{-1} = (1-\gamma_R)\boldsymbol{R}_0^{-1} + \gamma_R \mathcal{N}(0,1)$. (3) Set $w_1 \sim \gamma w_0 + (1-\gamma)\mathcal{N}(0,1)$. We use $\gamma = 0.4$ in all experiments. (4) Similarly, inject a γ_{RS} gap between $\boldsymbol{R}_t^{-1}$ and $\boldsymbol{S}_t^{-1}$. (5) For treatment effect parameters $w_\tau = w_1 - w_0$ in the real DGP, we sample its simulator counterpart with a gap γ_τ as $w_\tau^S = (1-\gamma_\tau)w_\tau + \gamma_\tau \mathcal{N}(0,1)$ and set w_t^S accordingly.

In linear settings, the optimization problems for the CATE estimators SimOnly, RealOnly, and $\text{Real}_\mu\text{Sim}_f$ admit closed-form solutions. We show the closed-form solutions in Table 4 in the Appendix, and a detailed derivation in Appendix B.7.

Table 1: RQ1: In a linear DGP setting, we vary the gaps using γ_R for $d(f_0,f_1)$ in the first column, γ_{RS} for $d(f_t,f_t^S)$, and γ_τ for $d(\tau,\tau^S)$. "low" refers to 0.1 that simulates small distance, while "high" refers to 0.4. We run all experiments with five different seeds and report *p*-values of comparing the mean performance in bracket.

			Synthetic-Gaussian				Real World-IHDP			
$d(f_0,f_1)$	$d(f_t,f_t^S)$	$d(\tau,\tau^S)$	SimOnly	RealOnly	Real$_\mu$Sim$_f$	*SimPONet*	SimOnly	RealOnly	Real$_\mu$Sim$_f$	*SimPONet*
0.00	high	high	2.82 (0.27)	**0.00 (1.00)**	15.75 (0.01)	2.58 (0.00)	3.57 (0.11)	**0.00 (1.00)**	48.76 (0.05)	3.20 (0.00)
low	low	low	0.63 (0.00)	2.47 (0.02)	1.19 (0.01)	**0.54 (0.00)**	1.00 (0.44)	3.43 (0.02)	2.73 (0.00)	**0.97 (0.00)**
low	low	high	1.57 (0.16)	2.47 (0.08)	**1.19 (0.83)**	1.39 (0.00)	1.62 (0.26)	3.43 (0.04)	2.73 (0.02)	**1.49 (0.00)**
low	high	low	2.14 (0.22)	2.47 (0.00)	15.75 (0.01)	**1.85 (0.00)**	3.67 (0.31)	3.43 (0.48)	48.76 (0.05)	**3.37 (0.00)**
low	high	high	2.82 (0.26)	**2.47 (0.56)**	15.75 (0.01)	2.57 (0.00)	3.57 (0.11)	3.43 (0.39)	48.76 (0.05)	**3.19 (0.00)**
high	low	low	0.63 (0.00)	13.86 (0.02)	1.19 (0.01)	**0.54 (0.00)**	1.00 (0.47)	47.78 (0.06)	2.73 (0.00)	**0.98 (0.00)**
high	low	high	1.57 (0.16)	13.86 (0.03)	**1.19 (0.83)**	1.39 (0.00)	1.62 (0.27)	47.78 (0.06)	2.73 (0.02)	**1.50 (0.00)**
high	high	low	2.14 (0.21)	13.86 (0.03)	15.75 (0.01)	**1.85 (0.00)**	3.67 (0.31)	47.78 (0.06)	48.76 (0.05)	**3.38 (0.00)**
high	high	high	2.82 (0.26)	13.86 (0.04)	15.75 (0.01)	**2.57 (0.00)**	3.57 (0.11)	47.78 (0.06)	48.76 (0.05)	**3.19 (0.00)**

For *SimPONet*, a closed-form solution is not possible, so we solve it to a local optimum using alternating minimization over $\widehat{\mu}_t$ and $\widehat{f}_t$, with each alternating update in closed-form. We show the *SimPONet*'s update equations in Appendix B.7.4. In summary, the setting of RQ1 allows study of the impact of varying discrepancies between the real and simulator distributions without approximation errors due to finite training samples.

We show the results comparing *SimPONet* with the three baselines in Table 1 where we observe: (a) Across both synthetic and real-world settings, *SimPONet* achieves either the best or second-best performance. The CATE error for *SimPONet* remains controlled primarily due to its ability to bound errors in the counterfactual distribution. (b) In contrast, the RealOnly and Real$_\mu$Sim$_f$ models perform well only under certain restrictive conditions favorable to them, providing zero error on the factual distribution but very high counterfactual error, leading to poor CATE estimates.

5.3 RQ2: Comparing *SimPONet* with State-of-the-art CATE Baselines

We conduct experiments using semi-synthetic observational datasets commonly used to assess efficacy of treatment effect estimation methods in the literature: the Infant Health Development Program (IHDP) and the Atlantic Causal Inference Conference (ACIC) datasets. These datasets contain real-world *pre-treatment* covariates (Z). Please refer Appendix B.8 for more details on these datasets.

To align these datasets with our study, we apply RealNVP Normalizing Flows (Dinh et al., 2016) to transform pre-treatment covariates Z into post-treatment X. Flows are non-linear deep neural networks that ensure invertibility of the covariate generating functions g_t,g_t^S. We consider randomly initialized flows with two coupling layers. We used the flows g_0,g_1 on real data, and two other distinct flows g_0^S,g_1^S to obtain covariates in synthetic data from Z. We borrow the real outcomes as-is from the ground truth dataset. However, we synthesize simulator outcomes with a gap of γ_τ as follows: (1) sample $w_\tau^S \in \mathbb{R}^{n_z} \sim \mathcal{N}(0,1)$ (2) set $\tau^S(z) = \tau(z) + (\sigma(\tau) \cdot \gamma_\tau \cdot z^\top w_\tau^S)$, where $\sigma(\tau)$ is the standard deviation of CATE labels in the real dataset. Scaling by $\sigma(\tau)$ ensured comparability between τ and τ^S. Thus, when $\gamma_\tau = 0$, $\tau = \tau^S$; when $\gamma_\tau = 1$, τ is disparate from τ^S.

We evaluated *SimPONet* against various baselines from the well-known CATENets (Curth et al., 2021), a benchmarking library for CATE estimation. Since these methods require pre-treatment covariates, we provided them with representations extracted by simulated causal representation extractor $\widetilde{f}_t^S(\boldsymbol{x})$ as

Table 2: RQ2: Comparison of *SimPONet* with several pre-treatment baselines and post-treatment proposals. p-values for paired t-tests against *SimPONet* are in brackets. Lower p-values indicate statistical significance. *SimPONet* outperforms others overall, while SimOnly performs best on ACIC-2 since $\tau = \tau^S$.

Method	IHDP	ACIC-2	ACIC-7	ACIC-26
RNet (Nie & Wager, 2021)	1.54 (0.00)	3.30 (0.00)	5.91 (0.04)	6.06 (0.18)
XNet (Künzel et al., 2019)	1.0 (0.00)	0.43 (0.15)	5.49 (0.17)	5.1 (0.38)
DRNet (Schwab et al., 2020)	0.96 (0.00)	0.24 (0.59)	5.53 (0.15)	5.08 (0.39)
CFRNet (Shalit et al., 2017)	0.96 (0.00)	0.36 (0.26)	5.55 (0.15)	5.09 (0.38)
FlexTENet (Curth & van der Schaar, 2021)	0.96 (0.00)	0.32 (0.32)	5.46 (0.19)	5.04 (0.40)
DragonNet (Shi et al., 2019)	0.96 (0.00)	0.29 (0.41)	5.57 (0.14)	5.09 (0.38)
IPW (Robins et al., 1994)	0.96 (0.00)	0.36 (0.24)	5.56 (0.15)	5.09 (0.38)
k-NN (Stuart, 2010)	0.96 (0.00)	0.33 (0.33)	5.48 (0.18)	5.13 (0.37)
PerfectMatch (Schwab et al., 2018)	0.98 (0.00)	0.56 (0.11)	5.75 (0.08)	5.13 (0.37)
StableCFR (Wu et al., 2023)	1.01 (0.00)	1.09 (0.03)	5.56 (0.15)	5.08 (0.43)
ESCFR (Wang et al., 2024)	0.96 (0.00)	0.27 (0.47)	5.55 (0.15)	5.79 (0.21)
PairNet (Nagalapatti et al., 2024b)	0.97 (0.00)	0.12 (0.85)	5.46 (0.23)	5.05 (0.37)
SimOnly	0.94 (0.00)	**0.00 (0.98)**	6.65 (0.00)	6.60 (0.12)
RealOnly	0.83 (0.13)	11.23 (0.01)	14.81 (0.05)	8.18 (0.01)
$\text{Real}_\mu\text{Sim}_f$	0.96 (0.00)	0.17 (0.76)	5.57 (0.14)	5.09 (0.38)
SimPONet	**0.79 (0.00)**	0.26 (0.00)	**5.04 (0.00)**	**4.67 (0.00)**

input. Running these baselines with post-treatment covariates $\boldsymbol{x}$ directly as input yielded much poorer results as shown in Fig. 5. We also compared *SimPONet* with SimOnly, RealOnly, and $\text{Real}_\mu\text{Sim}_f$ baselines that we developed in our theoretical analysis. We present the results in Table 2 for $\gamma_\tau = 0.1$, and defer the results for larger γ_τ to Appendix B.10. We make the following key observations:

(a) **IHDP**: This dataset contains 25 pre-treatment covariates, 19 of which are binary. Contrastive training struggled to capture these binary features, causing $\text{Real}_\mu\text{Sim}_f$, which uses $\widetilde{f}_t^S(\boldsymbol{x})$ as input, to consistently underperform RealOnly, which directly uses $\boldsymbol{x}$. As shown in Fig 2, the p-value is zero for the IHDP dataset, but significantly larger for the others. As a result, we set the weight of the regularizer $d(\widehat{f}_t, \widetilde{f}_t^S)$, controlled by λ_f, to 1e-4 for IHDP, while keeping λ_f at its default value of 1 for ACIC. Overall, *SimPONet* achieved the best performance.

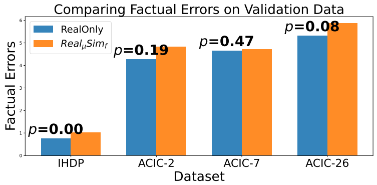

Figure 2: Factual errors with p-values shown above bars. For IHDP, RealOnly consistently outperforms $\text{Real}_\mu\text{Sim}_f$.

(b) **ACIC-2:** This dataset is unique in that the true CATE, τ, is constant across all individuals in the observational data, implying that its standard deviation $\sigma(\tau) = 0$ for real samples. As a result, our approach to synthesizing the simulated CATE, τ^S, given by $\tau^S(z) = \tau(z) + \left(\sigma(\tau) \cdot \gamma_\tau \cdot z^\top w_\tau^S\right)$, yields $\tau^S(z) = \tau(z)$ for all z. This leads to perfect alignment between the synthetic and true CATE, causing SimOnly to outperform all other methods on this dataset. Although *SimPONet* could have improved by assigning a higher weight to the τ^S regularizer, tuning this weight would typically require supervision on τ, which we avoid.

(c) **ACIC-7 and ACIC-26:** The CATENets baselines significantly outperform the RealOnly model, because of high-quality causal representations extracted by $\widetilde{f}_t^S$. This is demonstrated by $\text{Real}_\mu\text{Sim}_f$ outperforming RealOnly on factual error (see Fig. 2). In ACIC-7 and ACIC-26, *SimPONet* achieves the best results by leveraging the closeness between τ and τ^S. FlexTENet (Curth & van der Schaar, 2021), which shares parameters between $\widehat{\mu}_0$ and $\widehat{\mu}_1$, and PairNet (Nagalapatti et al., 2024b), which applies losses on pairs of close-by samples are strong contenders to *SimPONet*.

Table 3: RQ3: Impact of regularizers. Here, $-d(\widehat{f}_t,\widetilde{f}_t^S)$ represents our loss 3 with $\lambda_f = 0$, and $-\tau^S$ means $\lambda_\tau = 0$. A negative value implies *SimPONet* with all regularizers outperforms the ablation where one regularizer is disabled.

			IHDP - Linear f_t, Linear μ_t		GP - Linear f_t, Non-Linear μ_t	
$d(f_0,f_1)$	$d(f_t,f_t^S)$	$d(\tau,\tau^S)$	$-d(\widehat{f_t},\widetilde{f}^S)$	$-\tau^S$	$-d(\widehat{f_t},\widetilde{f}^S)$	$-\tau^S$
0.00	high	high	+1.29 (1.00)	-1.07 (0.18)	-0.55 (0.31)	-0.62 (0.26)
high	low	low	-0.64 (0.04)	+0.01 (0.51)	-0.29 (0.24)	-0.04 (0.47)
high	low	high	-0.40 (0.11)	+0.00 (0.50)	-0.42 (0.10)	-0.19 (0.34)
high	high	low	+1.74 (0.99)	-0.03 (0.48)	-0.66 (0.27)	-0.71 (0.25)
high	high	high	+1.30 (1.00)	-0.03 (0.45)	-0.72 (0.21)	-0.97 (0.18)

We also include experiments with non-linear, non-invertible covariate transformations using randomly initialized multi-layer perceptrons in Appendix B.11. Despite MLPs violating the diffeomorphism assumption required by our theory, *SimPONet* proved robust to these violations in practice.

5.4 RQ3: Ablation of *SimPONet* Losses

We evaluate the impact of the $d(\widehat{f}_t,\widetilde{f}_t^S)$ and τ^S regularizers in *SimPONet*'s objective 3. We experiment with the Linear IHDP dataset (Sec. 5.2) and the Non-Linear Gaussian process dataset (Sec. B.9). For the Linear IHDP dataset, in the $-\tau^S$ case, we added an L_2 penalty on w_t for the alternating minimization to work. Table 3 presents the *difference* in CATE errors of *SimPONet* and the ablation, averaged over five seeds with p-values. A negative entry means *SimPONet* does better than the ablation. We observe that τ^S loss is very effective since *SimPONet* outperforms the $-\tau^S$ in both datasets. For IHDP, removing $d(\widehat{f}_t,\widetilde{f}_t^S)$ loss helps. We could not set λ_f weight to be small because both RealOnly and $\text{Real}_\mu\text{Sim}_f$ achieved zero factual error. Despite this, *SimPONet* with both regularizers comfortably outperformed the other proposals demonstrating it as a better candidate for our task.

5.5 RQ4, RQ5

We detail the experiments for **RQ4** in Appendix B.9, testing linear transformations for g_t,g_t^S and non-linear Gaussian Processes for μ_t,μ_t^S. Overall, baselines performed well only under specific DGP conditions, while *SimPONet* consistently excelled across most settings. Appendix B.11 covers **RQ5** results, showing CATE baselines improved significantly with pre-treatment covariates Z, highlighting the need for causal representation recovery. Performance dropped for all methods when trained on X instead of $\widetilde{f}^S(\boldsymbol{x})$, underscoring simulators' value in extracting causal representations.

6 Conclusion

This paper addressed the challenge of estimating treatment effects from post-treatment covariates a setting not identifiable from observational data alone. We proposed to tackle this task using off-the-shelf simulators that synthesize counterfactuals, in contrast to prior work that relied on real-world counterfactuals, which limited their practical applicability. Our theoretical analysis established a bound on the CATE error based on the distributional mismatch between real and simulated data. Notable, ours is the first work to systematically analyze the role of simulators in CATE estimation. We introduced *SimPONet*, a framework that jointly learns from real and simulated samples to enhance CATE estimates beyond what could be achieved from observational data alone. Extensive experiments across various DGPs demonstrated that *SimPONet* is a robust and effective method for estimating CATE from post-treatment data.

References

Joshua Angrist. Estimating the labor market impact of voluntary military service using social security data on military applicants, 1995.

Orley Ashenfelter. Estimating the effect of training programs on earnings. *The Review of Economics and Statistics*, pp. 47–57, 1978.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019.

Elias Bareinboim and Judea Pearl. Controlling selection bias in causal inference. In *Artificial Intelligence and Statistics*, pp. 100–108. PMLR, 2012.

Elias Bareinboim and Jin Tian. Recovering causal effects from selection bias. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Nitay Calderon, Eyal Ben-David, Amir Feder, and Roi Reichart. DoCoGen: Domain counterfactual generation for low resource domain adaptation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7727–7746, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.533. URL `https://aclanthology.org/2022.acl-long.533`.

Vinod K Chauhan, Soheila Molaei, Marzia Hoque Tania, Anshul Thakur, Tingting Zhu, and David A Clifton. Adversarial de-confounding in individualised treatment effects estimation. In *International Conference on Artificial Intelligence and Statistics*, pp. 837–849. PMLR, 2023.

Zeming Chen, Qiyue Gao, Antoine Bosselut, Ashish Sabharwal, and Kyle Richardson. DISCO: Distilling counterfactuals with large language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5514–5528, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.302. URL `https://aclanthology.org/2023.acl-long.302`.

Alexander Coppock. Avoiding post-treatment bias in audit experiments. *Journal of Experimental Political Science*, 6(1):1–4, 2019.

Juan Correa, Jin Tian, and Elias Bareinboim. Generalized adjustment under confounding and selection biases. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Alicia Curth and Mihaela van der Schaar. On inductive biases for heterogeneous treatment effect estimation. *Advances in Neural Information Processing Systems*, 34:15883–15894, 2021.

Alicia Curth and Mihaela van der Schaar. In search of insights, not magic bullets: Towards demystification of the model selection dilemma in heterogeneous treatment effect estimation. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, 2023.

Alicia Curth, David Svensson, Jim Weatherall, and Mihaela van der Schaar. Really doing great at estimating cate? a critical look at ml benchmarking practices in treatment effect estimation. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 2)*, 2021.

Chiara Dalla Man, Robert A Rizza, and Claudio Cobelli. Meal simulation model of the glucose-insulin system. *IEEE Transactions on biomedical engineering*, 54(10):1740–1749, 2007.

Nikhil J Dhinagar, Sophia I Thomopoulos, Emily Laltoo, and Paul M Thompson. Counterfactual mri generation with denoising diffusion models for interpretable alzheimer's disease effect detection. *bioRxiv*, pp. 2024–02, 2024.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Tanay Dixit, Bhargavi Paranjape, Hannaneh Hajishirzi, and Luke Zettlemoyer. CORE: A retrieve-then-edit framework for counterfactual data generation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2964–2984, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.216. URL `https://aclanthology.org/2022.findings-emnlp.216`.

Zijun Gao and Yanjun Han. Minimax optimal nonparametric estimation of heterogeneous treatment e ffects. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. L in (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21751–21762. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/f75b757d3459\c3e93e98ddab7b903938-Paper.pdf`.

Muhammad Waleed Gondal, Manuel Wuthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. *Advances in Neural Information Processing Systems*, 32, 2019.

Yu Gu, Jianwei Yang, Naoto Usuyama, Chunyuan Li, Sheng Zhang, Matthew P Lungren, Jianfeng Gao, and Hoifung Poon. Medjourney: Counterfactual medical image generation by instruction-learning from multimodal patient journeys. 2023.

Negar Hassanpour and Russell Greiner. Counterfactual regression with importance sampling weights. In *IJCAI*, pp. 5880–5887, 2019a.

Negar Hassanpour and Russell Greiner. Learning disentangled representations for counterfactual regression. In *International Conference on Learning Representations*, 2019b.

JONATHAN HOMOLA, MIGUEL M. PEREIRA, and MARGIT TAVITS. Fixed effects and post-treatment bias in legacy studies. *American Political Science Review*, 118(1):537–544, 2024. doi: 10.1017/S0003055423001351.

Qiang Huang, Defu Cao, Yi Chang, and Yan Liu. Extracting post-treatment covariates for heterogeneous treatment effect estimation. 2023.

Stefano M Iacus, Gary King, and Giuseppe Porro. Causal inference without balance checking: Coarsened exact matching. *Political analysis*, 20(1):1–24, 2012.

Guillaume Jeanneret, Loïc Simon, and Frédéric Jurie. Diffusion models for counterfactual explanations. In *Proceedings of the Asian Conference on Computer Vision*, pp. 858–876, 2022.

Yonghan Jung, Jin Tian, and Elias Bareinboim. Learning causal effects via weighted empirical risk minimization. *Advances in neural information processing systems*, 33:12697–12709, 2020.

Nathan Kallus. Deepmatch: Balancing deep covariate representations for causal inference using adversarial training. In *International Conference on Machine Learning*, pp. 5067–5077. PMLR, 2020.

Prabhjot Kaur, Samira Taghavi, Zhaofeng Tian, and Weisong Shi. A survey on simulators for testing self-driving cars. In *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*, pp. 62–70. IEEE, 2021.

Edward H Kennedy. Towards optimal doubly robust estimation of heterogeneous causal effects. *arXiv preprint arXiv:2004.14497*, 2020.

Gary King. A hard unsolved problem? post-treatment bias in big social science questions. In *Hard Problems in Social Science" Symposium, April*, volume 10, 2010.

Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10):4156–4165, 2019.

Oran Lang, Ilana Traynis, and Yun Liu. Explaining counterfactual images. *Nature Biomedical Engineering*, 2023. URL `https://rdcu.be/dwVKK`.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124. PMLR, 2019a.

Francesco Locatello, Michael Tschannen, Stefan Bauer, Gunnar R¨¨ätsch, Bernhard Schölkopf, and Olivier Bachem. Disentangling factors of variations using few labels, 2019b. URL `https://openreview.net/forum?id=SkGy6hjvPE`.

Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. Generate your counterfactuals: Towards controlled counterfactual generation for text. In *AAAI Conference on Artificial Intelligence*, 2020. URL `https://api.semanticscholar.org/CorpusID:228063841`.

Lokesh Nagalapatti, Guntakanti Sai Koushik, Abir De, and Sunita Sarawagi. Learning recourse on instance environment to enhance prediction accuracy. In *Advances in Neural Information Processing Systems*, 2022.

Lokesh Nagalapatti, Akshay Iyer, Abir De, and Sunita Sarawagi. Continuous treatment effect estimation using gradient interpolation and kernel smoothing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(13):14397–14404, Mar. 2024a. doi: 10.1609/aaai.v38i13.29353. URL `https://ojs.aaai.org/index.php/AAAI/article/view/29353`.

Lokesh Nagalapatti, Pranava Singhal, Avishek Ghosh, and Sunita Sarawagi. Pairnet: Training with observed pairs to estimate individual treatment effect. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024b. URL `https://openreview.net/forum?id=o5SVr80Rgg`.

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

Xinkun Nie and Stefan Wager. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2):299–319, 2021.

Michal Ozery-Flato, Pierre Thodoroff, and Tal El-Hay. Adversarial balancing for causal inference. *ArXiv*, abs/1810.07406, 2018.

Yushu Pan and Elias Bareinboim. Counterfactual image editing. *arXiv preprint arXiv:2403.09683*, 2024.

Nick Pawlowski, Daniel Coelho de Castro, and Ben Glocker. Deep structural causal models for tractable counterfactual inference. *Advances in neural information processing systems*, 33:857–869, 2020.

J. Pearl and Cambridge University Press. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000. ISBN 9780521773621. URL `https://books.google.co.in/books?id=wnGU_TsW3BQC`.

Judea Pearl. Conditioning on post-treatment variables. *Journal of Causal Inference*, 3(1):131–137, 2015.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

Marcel Robeer, Floris Bex, and Ad Feelders. Generating realistic natural language counterfactuals. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3611–3625, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.306. URL `https://aclanthology.org/2021.findings-emnlp.306`.

James M Robins, Andrea Rotnitzky, and Lue Ping Zhao. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American statistical Association*, 89(427): 846–866, 1994.

Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Axel Sauer and Andreas Geiger. Counterfactual generative networks. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=BXewfAYMmJw`.

Patrick Schwab, Lorenz Linhardt, and Walter Karlen. Perfect match: A simple method for learning representations for counterfactual inference with neural networks. *arXiv preprint arXiv:1810.00656*, 2018.

Patrick Schwab, Lorenz Linhardt, Stefan Bauer, Joachim M Buhmann, and Walter Karlen. Learning counterfactual representations for estimating individual dose-response curves. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5612–5619, 2020.

Uri Shalit, Fredrik D. Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms, 2016. URL `https://arxiv.org/abs/1606.03976`.

Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pp. 3076–3085. PMLR, 2017.

Claudia Shi, David Blei, and Victor Veitch. Adapting neural networks for the estimation of treatment effects. *Advances in neural information processing systems*, 32, 2019.

Elizabeth A Stuart. Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1):1, 2010.

Jayaraman Thiagarajan, Vivek Sivaraman Narayanaswamy, Deepta Rajan, Jia Liang, Akshay Chaudhari, and Andreas Spanias. Designing counterfactual generators using deep model inversion. *Advances in Neural Information Processing Systems*, 34:16873–16884, 2021.

Julius Von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. *Advances in neural information processing systems*, 34:16451–16467, 2021.

Clinton J. Wang, Natalia S. Rost, and Polina Golland. Spatial-intensity transform gans for high fidelity medical image-to-image translation. In Anne L. Martel, Purang Abolmaesumi, Danail Stoyanov, Diana Mateus, Maria A. Zuluaga, S. Kevin Zhou, Daniel Racoceanu, and Leo Joskowicz (eds.), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, pp. 749–759, Cham, 2020. Springer International Publishing. ISBN 978-3-030-59713-9.

Hao Wang, Jiajun Fan, Zhichao Chen, Haoxuan Li, Weiming Liu, Tianqiao Liu, Quanyu Dai, Yichao Wang, Zhenhua Dong, and Ruiming Tang. Optimal transport for treatment effect estimation. *Advances in Neural Information Processing Systems*, 36, 2024.

Anpeng Wu, Kun Kuang, Ruoxuan Xiong, Bo Li, and Fei Wu. Stable estimation of heterogeneous treatment effects. In *International Conference on Machine Learning*, pp. 37496–37510. PMLR, 2023.

Jinyu Xie. Simglucose v0.2.1 (2018) [Online], 2018. URL `https://github.com/jxx123/simglucose`. Accessed on September 25, 2023.

Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. Representation learning for treatment effect estimation from observational data. *Advances in Neural Information Processing Systems*, 31, 2018.

Jinsung Yoon, James Jordon, and Mihaela Van Der Schaar. Ganite: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations*, 2018.

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

Yi-Fan Zhang, Hanlin Zhang, Zachary C. Lipton, Li Erran Li, and Eric P. Xing. Exploring transformer backbones for heterogeneous treatment effect estimation, 2022. URL `https://arxiv.org/abs/2202.01336`.

Roland S. Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. 139:12979–12990, 2021. URL `http://proceedings.mlr.press/v139/zimmermann21a.html`.

A Appendix / supplemental material

A.1 Code

We have released the code in the anonynous URL `https://anonymous.4open.science/r/catenets-simponet/README.md`. We have also uploaded the code along with our submission.

A.2 Learning Counterfactual Simulators

Here we discuss prior works that train generative models for synthesizing counterfactuals. In general, to obtain counterfactuals in the real distribution, we need to follow three steps (Pawlowski et al., 2020): (a) *abduction*, inverting X to obtain Z, (b) *action*, applying a new treatment, and (c) *prediction* generating a new X under the new treatment. These steps require prior knowledge of the DGP specifications, which are often difficult to define and cannot be learned from observational data alone (Pawlowski et al., 2020). Consequently, many methods bypass the principled approach and use pre-trained models like Diffusion models and Large Language models to generate pseudo counterfactuals from a related synthetic domain. Such simulators are proposed across various modalities, including images (Pawlowski et al., 2020; Gu et al., 2023; Thiagarajan et al., 2021; Pan & Bareinboim, 2024; Sauer & Geiger, 2021; Jeanneret et al., 2022), text (Madaan et al., 2020; Calderon et al., 2022; Chen et al., 2023; Dixit et al., 2022; Robeer et al., 2021), and healthcare (Lang et al., 2023; Wang et al., 2020; Dhinagar et al., 2024). Prior research (Gondal et al., 2019) shows that while such simulated data is not directly usable for downstream tasks, they provide strong inductive biases that transfer well to the real distribution. Our method can incorporate any such counterfactual generators as simulators, provided they contribute to learning causal representations that are predictive of CATE.

B Related Work

B.1 CATE with Pre-Treatment Covariates

The primary challenge addressed here is handling confounding that arises out of biased treatment assignment in observational datasets. The main ideas explored include: estimating pseudo-outcomes for missing treatments in the training dataset and then using these to train effect predictors (Gao & Han, 2020; Curth & van der Schaar, 2021; Nie & Wager, 2021; Kennedy, 2020; Yoon et al., 2018; Zhang et al., 2020; Nagalapatti et al., 2024a); adding targeted regularizers to ensure consistent ITE estimates (Shi et al., 2019; Nie et al., 2021; Zhang et al., 2022); learning balanced representation of covariates across treatment groups (Shalit et al., 2016; 2017; Yao et al., 2018; Chauhan et al., 2023; Wang et al., 2024; Wu et al., 2023); matching to near-by covariates (Stuart, 2010; Rosenbaum & Rubin, 1983; Iacus et al., 2012; Schwab et al., 2018; Kallus, 2020; Nagalapatti et al., 2024b); and weighing losses to mitigate confounding (Hassanpour & Greiner, 2019a;b; Jung et al., 2020; Ozery-Flato et al., 2018).

B.2 CATE with Post-Treatment Covariates

This is our setting, and is more challenging because it falls into the third rung (counterfactual) of Pearl's causal ladder (Pearl & Press, 2000). Please refer (Pearl, 2015) for a formal proof. In economics, post-treatment variables in trials are known to exacerbate estimated causal effects (Coppock, 2019; HOMOLA et al., 2024; King, 2010). Post-treatment variables have been used to estimate selection bias $P(T|Z)$ in observational data (Bareinboim & Pearl, 2012; Bareinboim & Tian, 2015; Correa et al., 2018). A closely related work is (Huang et al., 2023) that leverages post-treatment variables for estimating treatment effects but differs from us since they assume: (1) covariates X causally affect Y, and (2) an entangled version of X,Z is observed; they simply focus on disentangling Z through representation learning.

B.3 Real-World Applications of Simulators for Estimating CATE

We provide two examples from medicine and electrochemistry to show how simulators aid CATE estimation in practice:

Medicine. Simulators play a crucial role in pharmacology, particularly for assessing drug efficacies. For instance, the SimBiology toolbox [1] in MATLAB is commonly used to predict the effects of `SGLT2` inhibitors (T) on type-2 diabetes (Y) while considering post-treatment covariates (X) such as plasma glucose levels, gut glucose levels, urinary glucose excretion, and liver insulin levels. SimBiology enables modeling these effects using differential and algebraic equations that are often calibrated on target populations to minimize the real-simulator mismatch. Despite not perfectly replicating reality, such simulators are invaluable for early-stage clinical trial decisions and have demonstrated utility in modeling short-term treatment effects (Dalla Man et al., 2007).

Electrochemistry. Another application involves recommending optimal electrode materials to maximize battery capacity (Y). By observing Y under various electrode materials (T) and post-treatment variables like charge/discharge rate, internal resistance, and temperature distribution (X), the Ansys Battery Cell and Electrode Simulator [2] provides realistic electrochemical simulations. This tool has been used by Volkswagen Motorsport for comprehensive multiphysics simulations to design and validate battery models. Such simulators are highly relevant for practical decision-making in industries.

These examples illustrate the practical relevance of simulators across different fields. While simulators cannot fully replace real data or randomized controlled trials (RCTs), they offer valuable insights that can reduce the number of RCTs needed for optimal treatment identification. Our paper aims to characterize the CATE error when using imperfect simulators in conjunction with real observational data. Additionally, *SimPONet* maximizes the utility of simulators by leveraging the highly correlated simulator's treatment effects with real-world effects, without relying on the exact correlation of individual potential outcomes.

B.4 *SimPONet* Architecture

We present an overview of the *SimPONet* model architecture in Figure 3. Our model has four primary parameters: $\widehat{f}_0$ and $\widehat{f}_1$ for extracting causal representations, and $\widehat{\mu}_0$ and $\widehat{\mu}_1$ for predicting outcomes. Shared layers project $\widehat{f}_0$ and $\widehat{f}_1$ into a common space.

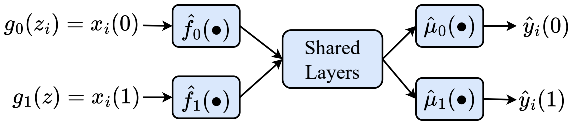

Figure 3: *SimPONet*'s model architecture.

B.5 *SimPONet* Pseudocode

Here, we present the *SimPONet* pseudocode. The steps involved in our algorithm are:

---

[1] https://in.mathworks.com/videos/series/simbiology-tutorials-for-qsp-pbpk-and-pk-pd-modeling-and-analysis.html

[2] https://www.ansys.com/applications/battery/battery-cell-and-electrode

line 1 First we use the simulator dataset D_{syn} to apply contrastive losses on the counterfactual covariates using Eq. 1. This optimization gives us a Z extractor in the simulator distribution, which we denote as $\widetilde{f}_t^S$.

line 2 We partition the training dataset into train, validation dataset using stratified split based on T. We then initialize the loss weigts λ_f, λ_τ to their defaults.

lines 4,5 We now decide upon the loss weight λ_f. To do so, we train RealOnly and $\text{Real}_\mu\text{Sim}_f$ models. We then assess the factual prediction errors of these models on the validation split of the training dataset. If RealOnly model performs much better than the $\text{Real}_\mu\text{Sim}_f$ model, it means that the $\widetilde{f}_t^S$ obtained from line2 above is of inferior quality. Therefore, we scale λ_f that regularizes the *SimPONet*'s $\widehat{f}_t$ based on $\widetilde{f}_t^S$ to a very small value, 1e-4.

line 6 We can now apply gradient descent algorithm on the *SimPONet*'s objective in Eq. 3 to train the $\widehat{\mu}_t, \widehat{f}_t$ parameters of the model.

Algorithm 1 *SimPONet* Algorithm

```
Require: Observational Data $D_{\text{trn}}$: $\{(\boldsymbol{x}_i, t_i, y_i)\}$, Simulator Data $D_{\text{syn}}$: $\{(\boldsymbol{x}_i^S(0), \boldsymbol{x}_i^S(1), y_i^S(0), y_i^S(1))\}$
1: Let $\{\widehat{f}(\bullet, t)\} \leftarrow Z$ extraction functions, and $\{\widehat{\mu}(\bullet, t)\} \leftarrow$ outcome functions for $t = 0,1$.
2: Let $\widetilde{f}_t^S \leftarrow$ Eq. 1 (Minimize Contrastive loss on $D_{\text{syn}}$)
3: Set $D_{\text{trn}}, D_{\text{val}} \leftarrow \textsc{split}(D_{\text{trn}}, pc = 0.3, \text{stratify} = T)$, and init default hyperparameters $\lambda_f, \lambda_\tau \leftarrow 1,1$
4: RealOnly $\leftarrow \min_{\{\widehat{\mu}, \widehat{f}\}} \sum_{D_{\text{trn}}} (y_i - \widehat{\mu}_{t_i}(\widehat{f}_{t_i}(\boldsymbol{x}_i)))^2$; $\text{Real}_\mu\text{Sim}_f \leftarrow \min_{\widehat{\mu}} \sum_{D_{\text{trn}}} (y_i - \widehat{\mu}_{t_i}(\widetilde{f}_{t_i}^S(\boldsymbol{x}_i)))^2$
5: Set $\lambda_f \leftarrow$ 1e-4 if FactualErr(RealOnly, $D_{\text{val}}$) >> FactualErr($\text{Real}_\mu\text{Sim}_f$, $D_{\text{val}}$)
6: $\{\widehat{f}_t, \widehat{\mu}_t\} \leftarrow$ Eq. 3 (perform gradient descent on SimPONet's objective using $D_{\text{trn}}, D_{\text{syn}}$ while early stopping
   using Factual Error on $D_{\text{val}}$)
7: Return $\{\widehat{f}_t, \widehat{\mu}_t\}$ for $t = 0,1$
```

We present the pseudocode for *SimPONet* in Alg. 1.

B.6 Theoretical Analysis

In this section, we present the proofs for our theoretical results.

B.6.1 Proof of Lemma 2

The CATE error is related to the factual and counterfactual error as: $\mathcal{E}_{\text{CATE}}^t \leq 2\mathcal{E}_F^t + 2\mathcal{E}_{CF}^t$

Proof. We decompose the CATE error into factual and counterfactual estimation error as follows:

$$
\begin{aligned}
\mathcal{E}_{\text{CATE}}^t &= \int_{\boldsymbol{x} \in \mathcal{X}} [\tau_X(\boldsymbol{x}, t) - \widehat{\tau_X}(\boldsymbol{x}, t)]^2 P(\boldsymbol{x}|t) d\boldsymbol{x} = \int_{\boldsymbol{x} \in \mathcal{X}} [\tau(f_t(\boldsymbol{x})) - \widehat{\tau}(\widehat{f}_t(\boldsymbol{x}))]^2 P(\boldsymbol{x}|t) d\boldsymbol{x} \\
&= \int_{\boldsymbol{x} \in \mathcal{X}} [(\mu_1(f_t(\boldsymbol{x})) - \mu_0(f_t(\boldsymbol{x}))) - (\widehat{\mu}_1(\widehat{f}_t(\boldsymbol{x})) - \widehat{\mu}_0(\widehat{f}_t(\boldsymbol{x})))]^2 P(\boldsymbol{x}|t) d\boldsymbol{x} \\
&= \int_{\boldsymbol{x} \in \mathcal{X}} [(\mu_1(f_t(\boldsymbol{x})) - \widehat{\mu}_1(\widehat{f}_t(\boldsymbol{x}))) - (\mu_0(f_t(\boldsymbol{x})) - \widehat{\mu}_0(\widehat{f}_t(\boldsymbol{x})))]^2 P(\boldsymbol{x}|t) d\boldsymbol{x}
\end{aligned}
$$

Let $t'=1-t$ denote the *counterfactual* treatment. We can then rewrite the above expression as:

$$\mathcal{E}^t_{\text{CATE}}=\int_{\boldsymbol{x}\in\mathcal{X}}[(\mu_t(f_t(\boldsymbol{x}))-\widehat{\mu}_t(\widehat{f}_t(\boldsymbol{x})))-(\mu_{t'}(f_t(\boldsymbol{x}))-\widehat{\mu}_{t'}(\widehat{f}_t(\boldsymbol{x})))]^2P(\boldsymbol{x}|t)d\boldsymbol{x}$$

Now using the inequality $(a-b)^2\leq 2a^2+2b^2$, we can separate the *factual* and *counterfactual* terms:

$$\begin{aligned}\mathcal{E}^t_{\text{CATE}}&\leq 2\int_{\boldsymbol{x}\in\mathcal{X}}[\mu_t(f_t(\boldsymbol{x}))-\widehat{\mu}_t(\widehat{f}_t(\boldsymbol{x}))]^2P(\boldsymbol{x}|t)d\boldsymbol{x}+2\int_{\boldsymbol{x}\in\mathcal{X}}[\mu_{t'}(f_t(\boldsymbol{x}))-\widehat{\mu}_{t'}(\widehat{f}_t(\boldsymbol{x}))]^2P(\boldsymbol{x}|t)d\boldsymbol{x}\\&=2\mathcal{E}^t_F+2\mathcal{E}^t_{CF}\end{aligned}$$

B.6.2 Recovery of f^S upto a diffeomorphic transformation

Lemma 5. *As $|D_{syn}|\rightarrow\infty$, contrastive training with paired covariates recovers $\widetilde{f}^S_t=h\circ f^S_t$ while paired outcome supervision recovers $\widetilde{\tau}^S=\tau^S\circ h^{-1}$ where h is a diffeomorphic transformation. Moreover, when the latent space $\mathcal{Z}\subset\mathbb{S}^{(n_z-1)}$ (unit-norm hypersphere in $\mathbb{R}^{n_z}$), h is a rotation transform by Extended Mazur-Ulam Theorem as shown in (Zimmermann et al., 2021) (Proposition 2).*

Proof. Theorem 4.4 of (Von Kügelgen et al., 2021) shows that contrastive training with covariate pairs $\{\boldsymbol{x}^S_i(0),\boldsymbol{x}^S_i(1)\}$ recovers Z upto a diffeomorphic transformation h, i.e. for the simulator DGP our estimate $\hat{z}_i=\widetilde{f}^S(\boldsymbol{x}^S_i(t),t)=h(z_i)=h(f^S(\boldsymbol{x}^S_i(t),t),\forall t\in\mathcal{T}$. Moreover for unit-norm latent representations, $\mathcal{Z}\subset\mathbb{S}^{d_z-1}$, (Zimmermann et al., 2021) show that h is an isometric (norm-preserving) function and therefore, a rotation transform by an extension of Mazur-Ulam Theorem. Mazur-Ulam Theorem states that any smooth, invertible and isometric function is necessarily affine. Moreover, in our setting, the norm of z as well as $\hat{z}$ is always one and thus, h is necessarily a rotation. Therefore, we recover $\widetilde{f}^S=h\circ f^S$ upto a rotation of the true inverse map f^S with sufficient paired samples from the simulator.

Next, we recover $\widetilde{\tau}^S$ from the following minimisation:

$$\widetilde{\tau}^S=\underset{\widehat{\tau}^S}{\operatorname{argmin}}\mathbb{E}_{\boldsymbol{x}^S}\left[\widehat{\tau}^S(\widetilde{f}^S(\boldsymbol{x}^S(t),t))-\tau^S(f^S(\boldsymbol{x}^S(t),t))\right]^2=\underset{\widehat{\tau}^S}{\operatorname{argmin}}\mathbb{E}_z\left[\widehat{\tau}^S(h(z))-\tau^S(z)\right]^2$$

The above optimization gives $\widetilde{\tau}^S=\tau^S\circ h^{-1}$ and hence we recover the CATE function τ^S for the simulator DGP composed with h^{-1}.

Proof of Lemma 4.

Assume τ is K_τ-Lipschitz, and $\widetilde{f}^S$ and $\widetilde{\tau}^S$ are estimates from the simulator DGP obtained from the optimization in Eq. 1, 2. Then, the CATE error on the estimates $\widehat{f}_t$ and $\widehat{\tau}$ admits the following bound:

$$\mathcal{E}^t_{\text{CATE}}(\widehat{f}_t,\widehat{\tau})\leq[8\mathcal{E}^t_F+12d_h(\widehat{\tau},\widetilde{\tau}^S)+12K^2_\tau d_{\boldsymbol{x}|t}(\widehat{f}_t,\widetilde{f}^S_t)]+[12d_z(\tau,\tau^S)+12K^2_\tau d_{\boldsymbol{x}|t}(f_t,f^S_t)]$$

where $d_{\boldsymbol{x}|t},d_z,d_{h(z)}$ are distance functions in Sec. 3 and $\mathcal{E}^t_F$ is the factual loss.

Proof. We now construct at upper bound on *counterfactual* error $\mathcal{E}_{CF}^t$ that relies on both observational data and simulator estimates to motivate the *SimPONet* objective:

$$\begin{aligned}
\mathcal{E}_{CF}^t &= \int_{\boldsymbol{x}\in\mathcal{X}} [\mu_{t'}(f_t(\boldsymbol{x})) - \widehat{\mu}_{t'}(\widehat{f}_t(\boldsymbol{x}))]^2 P(\boldsymbol{x}|t) d\boldsymbol{x} \\
&= \int_{\boldsymbol{x}\in\mathcal{X}} [(\mu_{t'}(f_t(\boldsymbol{x})) - \mu_t(f_t(\boldsymbol{x}))) - (\widehat{\mu}_{t'}(\widehat{f}_t(\boldsymbol{x})) - \widehat{\mu}_t(\widehat{f}_t(\boldsymbol{x}))) + \mu_t(f_t(\boldsymbol{x})) - \widehat{\mu}_t(\widehat{f}_t(\boldsymbol{x}))]^2 P(\boldsymbol{x}|t) d\boldsymbol{x} \\
&= \int_{\boldsymbol{x}\in\mathcal{X}} [(2\mathbf{1}_{t=0} - 1) \cdot (\tau(f_t(\boldsymbol{x})) - \widehat{\tau}(\widehat{f}_t(\boldsymbol{x}))) + \mu_t(f_t(\boldsymbol{x})) - \widehat{\mu}_t(\widehat{f}_t(\boldsymbol{x}))]^2 P(\boldsymbol{x}|t) d\boldsymbol{x}
\end{aligned}$$

Where $\mathbf{1}_{t=0} = 1$ when $t = 0$ and zero otherwise, and thus, $(2\mathbf{1}_{t=0} - 1) = \pm 1$ adjusting the sign of CATE terms. Now we utilise the inequality $(a+b+c)^2 \leq 3(a^2+b^2+c^2)$ to obtain:

$$\begin{aligned}
\mathcal{E}_{CF}^t &= \int_{\boldsymbol{x}\in\mathcal{X}} [(2\mathbf{1}_{t=0} - 1) \cdot (\tau(f_t(\boldsymbol{x})) - \widehat{\tau}(h \circ f_t(\boldsymbol{x})) + \widehat{\tau}(h \circ f_t(\boldsymbol{x})) - \widehat{\tau}(\widehat{f}_t(\boldsymbol{x}))) + \mu_t(f_t(\boldsymbol{x})) - \widehat{\mu}_t(\widehat{f}_t(\boldsymbol{x}))]^2 P(\boldsymbol{x}|t) d\boldsymbol{x} \\
&\leq 3 \int_{\boldsymbol{x}\in\mathcal{X}} [\tau(f_t(\boldsymbol{x})) - \widehat{\tau}(h \circ f_t(\boldsymbol{x}))]^2 P(\boldsymbol{x}|t) d\boldsymbol{x} + 3 \int_{\boldsymbol{x}\in\mathcal{X}} [\widehat{\tau}(h \circ f_t(\boldsymbol{x})) - \widehat{\tau}(\widehat{f}_t(\boldsymbol{x}))]^2 P(\boldsymbol{x}|t) d\boldsymbol{x} \\
&\quad + 3 \int_{\boldsymbol{x}\in\mathcal{X}} [\mu_t(f_t(\boldsymbol{x})) - \widehat{\mu}_t(\widehat{f}_t(\boldsymbol{x}))]^2 P(\boldsymbol{x}|t) d\boldsymbol{x} \\
&= 3 \int_{z\in\mathcal{Z}} [\tau(z) - \widehat{\tau}(h(z))]^2 P(z|t) dz + 3 \int_{\boldsymbol{x}\in\mathcal{X}} [\widehat{\tau}(h(f_t(\boldsymbol{x}))) - \widehat{\tau}(\widehat{f}_t(\boldsymbol{x}))]^2 P(\boldsymbol{x}|t) d\boldsymbol{x} + 3\mathcal{E}_F^t
\end{aligned}$$

Here h denotes the unknown rotation transformation that relates the estimated simulator functions $(\widetilde{f}^S, \widetilde{\tau}^S)$ with the ground-truth simulator functions (f^S, τ^S) as shown in Lemma 5. Let K_τ be the Lipschitz constant for $\widehat{\tau}$. We can bound the second term in the above expression as follows:

$$\begin{aligned}
\mathcal{E}_{CF}^t &\leq 3 \int_{z\in\mathcal{Z}} [\tau(z) - \widehat{\tau}(h(z))]^2 P(z|t) dz + 3K_\tau^2 \int_{\boldsymbol{x}\in\mathcal{X}} ||h(f_t(\boldsymbol{x})) - \widehat{f}_t(\boldsymbol{x})||^2 P(\boldsymbol{x}|t) d\boldsymbol{x} + 3\mathcal{E}_F^t \\
&= 3 d_z(\tau, \widehat{\tau} \circ h) + 3K_\tau^2 d_{\boldsymbol{x}|t}(h \circ f_t, \widehat{f}_t) + 3\mathcal{E}_F^t
\end{aligned}$$

Now we can add and subtract simulator function estimates to bound the two distance terms as follows:

$$\begin{aligned}
\mathcal{E}_{CF}^{t} \leq & 3\int_{z\in\mathcal{Z}}[\tau(z)-\tau^{S}(z)+\tau^{S}(z)-\widehat{\tau}(h(z))]^{2}P(z|t)dz \\
& +3K_{\tau}^{2}\int_{\boldsymbol{x}\in\mathcal{X}}||h(f_t(\boldsymbol{x}))-h(f_t^{S}(\boldsymbol{x}))+h(f_t^{S}(\boldsymbol{x}))-\widehat{f}_t(\boldsymbol{x})||^{2}P(\boldsymbol{x}|t)d\boldsymbol{x}+3\mathcal{E}_{F}^{t} \\
\leq & 6\int_{z\in\mathcal{Z}}[\tau(z)-\tau^{S}(z)]^{2}P(z|t)dz+6\int_{z\in\mathcal{Z}}[\tau^{S}(z)-\widehat{\tau}(h(z))]^{2}P(z|t)dz \\
& +6K_{\tau}^{2}\int_{\boldsymbol{x}\in\mathcal{X}}||h(f_t(\boldsymbol{x}))-h(f_t^{S}(\boldsymbol{x})||^{2}P(\boldsymbol{x}|t)d\boldsymbol{x}+6K_{\tau}^{2}\int_{\boldsymbol{x}\in\mathcal{X}}||h(f_t^{S}(\boldsymbol{x}))-\widehat{f}_t(\boldsymbol{x})||^{2}P(\boldsymbol{x}|t)d\boldsymbol{x}+3\mathcal{E}_{F}^{t} \\
= & 6d_z(\tau,\tau^{S})+6d_z(\widehat{\tau}\circ h,\tau^{S})+6K_{\tau}^{2}d_{\boldsymbol{x}|t}(h\circ f_t,h\circ f_t^{S})+6K_{\tau}^{2}d_{\boldsymbol{x}|t}(\widehat{f}_t,h\circ f_t^{S})+3\mathcal{E}_{F}^{t}
\end{aligned}$$

Now, using Lemma 5, we can rewrite $\tau^{S}=\widetilde{\tau}^{S}\circ h$ in the second term. Thus, $d_z(\widehat{\tau}\circ h,\tau^{S})=d_z(\widehat{\tau}\circ h,\widetilde{\tau}^{S}\circ h)$. Now making use of Definition 2, we can rewrite this as $d_{h(z)}(\widehat{\tau},\widetilde{\tau}^{S})$ which is a distance function defined on the space of rotated latents $h(z)$. We also rewrite $h\circ f^{S}$ as $\widetilde{f}^{S}$ in the fourth term.

Moreover, $d_{\boldsymbol{x}|t}(h\circ f_t,h\circ f_t^{S})=d_{\boldsymbol{x}|t}(f_t,f_t^{S})$ since h is a rotation transform and preserves the distance between any two vectors. Thus, $||f_t(\boldsymbol{x})-f_t^{S}(\boldsymbol{x})||_2=||h\circ f_t(\boldsymbol{x})-h\circ f_t^{S}(\boldsymbol{x})||_2$. Combining these results, we can evaluate the above bound to the following:

$$\mathcal{E}_{CF}^{t}\leq[6d_{h(z)}(\widehat{\tau},\widetilde{\tau}^{S})+6K_{\tau}^{2}d_{\boldsymbol{x}|t}(\widehat{f}_t,\widetilde{f}_t^{S})+3\mathcal{E}_{F}^{t}]+[6d_z(\tau,\tau^{S})+6K_{\tau}^{2}d_{\boldsymbol{x}|t}(f_t,f_t^{S})]$$

B.7 Linear DGP Derivation

We derive expressions for CATE estimates $\widehat{\tau_X}(\boldsymbol{x},t)$ as well as $\mathcal{E}_{\text{CATE}}^{t}$ for each of our proposed estimators in the linear setting below. Note that ground truth CATE $\tau_X(\boldsymbol{x},t)=\boldsymbol{x}\boldsymbol{R}_t^{-1}(w_1-w_0)$. We consider factual treatment $t=1$ to illustrate the errors.

B.7.1 SimOnly

For SimOnly, we use $\hat{\boldsymbol{R}}_t^{-1}=\boldsymbol{S}_t^{-1}$ and $\hat{w}_t=w_t^{S}$ which are obtained by training on simulator data. Thus, the CATE estimate $\widehat{\tau_X}(\boldsymbol{x}^*,t)=\boldsymbol{x}^*\boldsymbol{S}_t^{-1}(w_1^{S}-w_0^{S})$. The CATE error on a sample $\boldsymbol{x}^*$, with treatment $t=1$ is given by $[\widehat{\tau_X}(\boldsymbol{x}^*,1)-\tau_X(\boldsymbol{x}^*,1)]^2=[(\boldsymbol{x}^*(\boldsymbol{S}_1^{-1}(w_1^{S}-w_0^{S})-\boldsymbol{R}_1^{-1}(w_1-w_0))]^2$

B.7.2 RealOnly

For RealOnly, the factual objective $\mathcal{E}_F^{t}=||\boldsymbol{x}\hat{\boldsymbol{R}}_t^{-1}\hat{w}_t-y||_2^2=||\boldsymbol{x}\hat{\boldsymbol{R}}_t^{-1}\hat{w}_t-\boldsymbol{x}\boldsymbol{R}_t^{-1}w_t||_2^2$. Thus, the closed form solution of the estimator $\hat{\boldsymbol{R}}_t^{-1}\hat{w}_t=\boldsymbol{R}_t^{-1}w_t,\forall t\in\mathcal{T}$. Since we can't decouple the terms $\hat{\boldsymbol{R}}_t^{-1}$ and $\hat{w}_t$, the CATE estimate is given by $\widehat{\tau_X}(\boldsymbol{x}^*,t)=\boldsymbol{x}^*\hat{\boldsymbol{R}}_1^{-1}\hat{w}_1-\boldsymbol{x}^*\hat{\boldsymbol{R}}_0^{-1}\hat{w}_0=\boldsymbol{x}^*\boldsymbol{R}_1^{-1}w_1-\boldsymbol{x}^*\boldsymbol{R}_0^{-1}w_0$.
CATE error on sample $\boldsymbol{x}^*$ with treatment $t=1$ is given by $[\widehat{\tau_X}(\boldsymbol{x}^*,1)-\tau_X(\boldsymbol{x}^*,1)]^2=[(\boldsymbol{x}^*\boldsymbol{R}_1^{-1}w_1-\boldsymbol{x}^*\boldsymbol{R}_0^{-1}w_0)-\boldsymbol{x}\boldsymbol{R}_1^{-1}(w_1-w_0)]^2=[\boldsymbol{x}(\boldsymbol{R}_1^{-1}-\boldsymbol{R}_0^{-1})w_0]^2$

B.7.3 Real$_\mu$Sim$_f$

For Real$_\mu$Sim$_f$, we first set $\hat{\boldsymbol{R}}_t^{-1}=\boldsymbol{S}_t^{-1}$ which is obtained by training on simulator data. Next, we train $\hat{w}_t$ on the factual objective: $||\boldsymbol{x}\hat{\boldsymbol{R}}_t^{-1}\hat{w}_t-\boldsymbol{x}\boldsymbol{R}_t^{-1}w_t||_2^2=||\boldsymbol{x}\boldsymbol{S}_t^{-1}\hat{w}_t-\boldsymbol{x}\boldsymbol{R}_t^{-1}w_t||_2^2$. This, gives us a closed form solution for the minimising $\hat{w}_t = \boldsymbol{S}_t \boldsymbol{R}_t^{-1} w_t$. The CATE estimate $\widehat{\tau_X}(\boldsymbol{x}^*, t) = \boldsymbol{x}^* \boldsymbol{S}_t^{-1}(\hat{w}_1 - \hat{w}_0) = \boldsymbol{x}^* \boldsymbol{S}_t^{-1}(\boldsymbol{S}_1 \boldsymbol{R}_1^{-1} w_1 - \boldsymbol{S}_0 \boldsymbol{R}_0^{-1} w_0)$. Fixing treatment $t = 1$, this simplifies further: $\widehat{\tau_X}(\boldsymbol{x}^*, 1) = \boldsymbol{x}^* \boldsymbol{S}_1^{-1}(\boldsymbol{S}_1 \boldsymbol{R}_1^{-1} w_1 - \boldsymbol{S}_0 \boldsymbol{R}_0^{-1} w_0) = \boldsymbol{x}^*(\boldsymbol{R}_1^{-1} w_1 - \boldsymbol{S}_1^{-1} \boldsymbol{S}_0 \boldsymbol{R}_0^{-1} w_0)$. CATE Error is given by $[\widehat{\tau_X}(\boldsymbol{x}^*, 1) - \tau_X(\boldsymbol{x}^*, 1)]^2 = [\boldsymbol{x}^*(\boldsymbol{R}_1^{-1} w_1 - \boldsymbol{S}_1^{-1} \boldsymbol{S}_0 \boldsymbol{R}_0^{-1} w_0) - \boldsymbol{x}^* \boldsymbol{R}_1^{-1}(w_1 - w_0)]^2 = [\boldsymbol{x}^* \boldsymbol{R}_1^{-1} w_0 - \boldsymbol{x}^* \boldsymbol{S}_1^{-1} \boldsymbol{S}_0 \boldsymbol{R}_0^{-1} w_0]^2 = [\boldsymbol{x}^*(\boldsymbol{R}_1^{-1} - \boldsymbol{S}_1^{-1} \boldsymbol{S}_0 \boldsymbol{R}_0^{-1}) w_0]^2$

Table 4: This table presents the predicted CATE and the corresponding CATE errors obtained from the three CATE proposals computed analytically for a test instance $\boldsymbol{x}^\star$ observed under treatment 1.

Method	Estimate for CATE $\widehat{\tau_X}(\boldsymbol{x}^\star,1)$	CATE Error $[\widehat{\tau_X}(\boldsymbol{x}^\star,1) - \tau(\boldsymbol{x}^\star,1)]^2$
SimOnly	$\boldsymbol{x}^\star \boldsymbol{S}_1^{-1} w_\tau^S$	$\left[\boldsymbol{x}^\star\left(\boldsymbol{R}_1^{-1} w_\tau - \boldsymbol{S}_1^{-1} w_\tau^S\right)\right]^2$
RealOnly	$\boldsymbol{x}^\star\left(\boldsymbol{R}_1^{-1} w_1 - \boldsymbol{R}_0^{-1} w_0\right)$	$\left[\boldsymbol{x}^\star(\boldsymbol{R}_0^{-1} - \boldsymbol{R}_1^{-1}) w_0\right]^2$
Real$_\mu$Sim$_f$	$\boldsymbol{x}^\star \boldsymbol{S}_1^{-1} \boldsymbol{S}_1 \boldsymbol{R}_1^{-1} w_1 - \boldsymbol{x}^\star \boldsymbol{S}_1^{-1} \boldsymbol{S}_0 \boldsymbol{R}_0^{-1} w_0$	$\left[\boldsymbol{x}^\star(\boldsymbol{R}_1^{-1} - \boldsymbol{S}_1^{-1} \boldsymbol{S}_0 \boldsymbol{R}_0^{-1}) w_0\right]^2$

B.7.4 *SimPONet*

We train both $\hat{\boldsymbol{R}}_t^{-1}, \hat{w}_t$ on the following objective jointly:

$$\mathcal{L}(\{\hat{\boldsymbol{R}}_t^{-1}, \hat{w}_t\}_{t=0,1}) = \left[\sum_{t=0,1} ||\boldsymbol{x}\hat{\boldsymbol{R}}_t^{-1}\hat{w}_t - \boldsymbol{x}\boldsymbol{R}_t^{-1} w_t||_2^2 + \lambda_f \sum_{t=0,1} ||\boldsymbol{x}\hat{\boldsymbol{R}}_t^{-1} - \boldsymbol{x}\boldsymbol{S}_t^{-1}||_F^2 + \lambda_\tau ||\mathbf{z}(\hat{w}_1 - \hat{w}_0) - \mathbf{z}(w_1 - w_0)||_2^2\right]$$

Here, $\mathbf{z} = \boldsymbol{x}_{t'}^S \boldsymbol{S}_{t'}^{-1}$ are the latents for simulated covariates $\boldsymbol{x}_{t'}^S$ (which are identifiable from D_{syn}). Due to the joint nature of this optimisation, it is not possible to derive closed form solutions for the optimum. However, one can compuet gradients of the objective with respect to $\hat{\boldsymbol{R}}_t^{-1}$ and $\hat{w}_t$ separately. This, gives us an alternating minimisation algorithm with closed form updates.

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{R}}_t^{-1}} &= \frac{\partial}{\partial \hat{\boldsymbol{R}}_t^{-1}}\left[||\boldsymbol{x}\hat{\boldsymbol{R}}_t^{-1}\hat{w}_t - y||_2^2 + \lambda_f ||\boldsymbol{x}\hat{\boldsymbol{R}}_t^{-1} - \boldsymbol{x}\boldsymbol{S}_t^{-1}||_F^2\right] \\
&= 2\boldsymbol{x}^T \boldsymbol{x}\hat{\boldsymbol{R}}_t^{-1}(\hat{w}_t \hat{w}_t{}^T + \lambda_f \boldsymbol{I}) - 2\boldsymbol{x}^T y \hat{w}_t + -2\lambda_f \boldsymbol{x}^T \boldsymbol{x}\boldsymbol{S}_t^{-1}
\end{aligned}$$

Setting the derivative to zero, we obtain the following update rule:

$$\hat{\boldsymbol{R}}_t^{-1} \leftarrow (\boldsymbol{x}^\dagger y \hat{w}_t + \lambda_f \boldsymbol{S}_t^{-1}) \cdot (\hat{w}_t \hat{w}_t{}^T + \lambda_f \boldsymbol{I})^{-1}$$

where $\boldsymbol{x}^\dagger = (\boldsymbol{x}^T \boldsymbol{x})^{-1} \boldsymbol{x}^T$ is the pseudoinverse of $\boldsymbol{x}$.

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \hat{w}_t} &= \frac{\partial}{\partial \hat{w}_t}\left[||\boldsymbol{x}\hat{\boldsymbol{R}}_t^{-1}\hat{w}_t - y||_2^2 + \lambda_\tau ||\mathbf{z}(\hat{w}_t - \hat{w_{t'}}) - (y_1^S - y_0^S)||_2^2\right] \\
&= 2(\hat{z}^T \hat{z})\hat{w}_t - 2\hat{z}^T y + 2\lambda_\tau (z^T z \hat{w}_t - z^T (z\hat{w_{t'}} + (y_1^S - y_0^S))) \\
&= 2[(\hat{z}^T \hat{z}) + \lambda_\tau (z^T z)]\hat{w}_t - 2(\hat{z}^T y + \lambda_\tau z^T (z\hat{w_{t'}} + (y_1^S - y_0^S)))
\end{aligned}$$

Where $\hat{z} = \boldsymbol{x}\hat{\boldsymbol{R}}_t^{-1}$. Setting the derivative to zero, we obtain the following update rule:

$$\hat{w}_t \leftarrow ((\hat{z}^T \hat{z}) + \lambda_\tau (z^T z))^{-1} \cdot (\hat{z}^T y + \lambda_\tau z^T (z\hat{w_{t'}} + (y_1^S - y_0^S)))$$

For *SimPONet*, we perform alternating updates of $\hat{w}_t$ and $\hat{\boldsymbol{R}}_t^{-1}$ fixing the other estimate.

B.8 Summary of Datasets

IHDP. The Infant Health and Development Program (IHDP) is a randomized controlled trial designed to assess the impact of physician home visits on the cognitive test performance of premature infants. The dataset exhibits selection bias due to the deliberate removal of non-random subsets of treated individuals from the training data. Since outcomes are observed for only one treatment, we generate both observed and counterfactual outcomes using a synthetic outcome generation function based on the original covariates for both treatments, making the dataset suitable for causal inference.

The IHDP dataset includes 747 subjects and 25 variables. While the original dataset discussed in (Shalit et al., 2017) had 1000 versions, our work uses a smaller version with 100 iterations, aligning with the CATENets benchmark. Each version varies in the complexity of the assumed outcome generation function, treatment effect heterogeneity, etc. As outlined in (Curth et al., 2021), reporting the standard deviation of performance across the 100 different seeds is inappropriate. Therefore, we calculate p-values through paired t-tests between our method (*SimPONet*) and other baseline methods, using *SimPONet* as the baseline for all experiments. We follow setting D of the IHDP dataset as mentioned in (Curth & van der Schaar, 2021) where response surfaces are modified to suppress the extremely high variance of potential outcomes in certain versions of the IHDP dataset.

ACIC. The Atlantic Causal Inference Conference (ACIC) competition dataset (2016)[3] consists of 77 datasets, all containing the same 58 covariates derived from the Collaborative Perinatal Project. Each dataset simulates binary treatment assignments and continuous outcome variables, with variations in the complexity of the treatment assignment mechanism, treatment effect heterogeneity, the ratio of treated to control observations, overlap between treatment and control groups, dimensionality of the confounder space, and the magnitude of the treatment effect.

All datasets share common characteristics, such as independent and identically distributed observations conditional on covariates, adherence to the ignorability assumption (selection on observables with all confounders measured and no hidden bias), and the presence of non-true confounding covariates. Of the 77 datasets, we selected a subset of three: versions 2, 7, and 26, aligning with the CATENets benchmark. These versions present non-linear covariate-to-outcome relationships and maximum variability in treatment effect heterogeneity. Version 2, notably, exhibits no heterogeneity, meaning the treatment effect is constant across all individuals. However, accurately estimating outcome differences even for this version is challenging due to the inherent noise in potential outcome realizations in the dataset.

B.9 RQ4: Linear covariate function g and non-linear outcome function μ

Now, we consider a more complex setup where the covariate functions g_t and g_t^S remain linear, but the outcome functions μ_t and μ_t^S are nonlinear in Z. In particular, we sample the outcomes y and y^S using Gaussian Processes (GPs) (Rasmussen & Williams, 2005). Let $GP(0,\mathcal{K}_\gamma)$ denote a GP with an RBF kernel of width γ, so a higher γ results in a more complex function. To sample the μ_0,μ_1 such that their difference τ has a gap γ, we follow: (1) Sample τ using a GP: $\tau \sim GP(0,\mathcal{K}_\gamma)$. (2) Sample $\mu_0 \sim GP(0,\mathcal{K}_1)$. (3) Set $\mu_1 \sim \mu_0 + \tau$. As before, we set $\gamma = 0.4$. Now, to sample μ_0^S,μ_1^S such that $d(\tau,\tau^S) = \gamma_\tau$: (1) Set $\tau^S \sim \tau + GP(0,\mathcal{K}_{\gamma_\tau})$. (2) Sample $y_0^S \sim GP(0,\mathcal{K}_1)$. (3) Set $y_1^S = y_0^S + \tau^S$.

---

[3] `https://jenniferhill7.wixsite.com/acic-2016/competition`

Table 5: RQ4: Results for linear covariate and GP-based nonlinear outcome functions. We run each experiment 5 times and show the p-values. *SimPONet* outperforms others in many settings. RealOnly is a strong contender.

$d(f_0,f_1)$	$d(f_t,f_t^S)$	$d(\tau,\tau^S)$	SimOnly	RealOnly	$\text{Real}_\mu\text{Sim}_f$	*SimPONet*
0.00	high	high	5.30 (0.09)	2.99 (0.11)	3.17 (0.09)	**1.94 (0.00)**
low	low	low	1.77 (0.15)	1.06 (0.73)	**1.05 (0.77)**	1.26 (0.00)
low	low	high	1.56 (0.08)	**0.98 (0.65)**	1.07 (0.44)	1.05 (0.00)
low	high	low	4.60 (0.04)	2.87 (0.07)	3.12 (0.04)	**1.88 (0.00)**
low	high	high	4.15 (0.04)	2.96 (0.02)	3.11 (0.01)	**1.60 (0.00)**
high	low	low	2.39 (0.14)	1.19 (0.68)	**1.12 (0.77)**	1.37 (0.00)
high	low	high	1.94 (0.11)	**0.98 (0.83)**	1.16 (0.58)	1.21 (0.00)
high	high	low	7.42 (0.09)	2.65 (0.38)	2.95 (0.25)	**2.37 (0.00)**
high	high	high	5.70 (0.07)	2.80 (0.17)	3.16 (0.09)	**2.10 (0.00)**

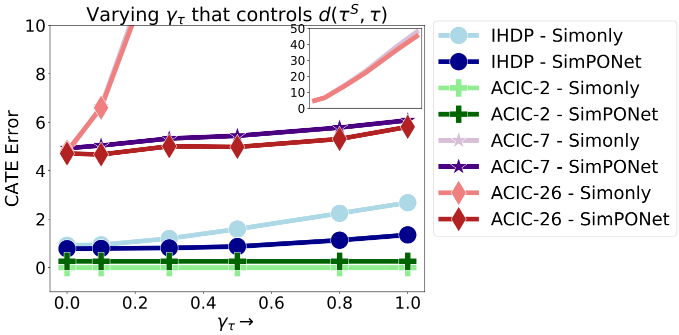

Figure 4: We vary γ_τ, which controls the gap between the synthetic CATE, τ^S, and the real CATE, τ. Each dataset is represented by a distinct color, where the pale version of the color indicates SimOnly and the darker version denotes *SimPONet*. For ACIC-7 and ACIC-26, as γ_τ increases, the CATE error grows significantly. Therefore, we present these results as an inset figure in the top-right corner.

We estimate $\widetilde{f}^S$ in Eq. 1 in closed-form, whereas we learn other parameters using gradient descent. We show the results in Table 5 where we observe: (a) *SimPONet* outperforms the other baselines in five out of nine settings (b) In alignment with our theory, $\text{Real}_\mu\text{Sim}_f$ performs better than others only when $d(f_t, f_t^S)$ is small. In summary, when the properties of the underlying DGP are unclear, *SimPONet* proves to be an effective approach for learning τ.

B.10 Varying γ_τ in Arbitrary DGP Experiment

The results of the varying γ_τ experiment are presented in Fig. 4, where we compare two approaches that leverage simulator data during training: SimOnly and *SimPONet*. Across all γ_τ gaps, we observe that *SimPONet* consistently outperforms SimOnly in three out of four datasets, with ACIC-2 being the exception. This is expected, as ACIC-2 satisfies the condition $\tau_s = \tau$. The performance gains of *SimPONet* are particularly notable in the ACIC-7 and ACIC-26 datasets, where the CATE error for SimOnly escalates significantly at larger γ_τ values. The exact error values for these cases are shown in the inset subplot at the top-right. These findings underscore our argument: while SimOnly can perform well on simulators closely aligned with the real world, it struggles with real-world simulators that diverge from reality. In contrast, *SimPONet*'s adjustment strategies—enabled by theoretically grounded regularizers derived from the CATE error analysis—yield much more reliable CATE estimates.

B.11 RQ5: Comparing CATE methods when trained on Z vs X vs $\widetilde{f}^S(X)$

To address RQ5, we use the IHDP dataset to evaluate the baseline models when trained on post-treatment covariates X directly, and we compare these results with those from Table 2, where the baselines were trained on simulated causal representations, $\widetilde{f}_t^S(\boldsymbol{x})$. We also evaluate the baselines trained on pre-treatment Z to explicitly show the detrimental impact of post-treatment covariates on the CATE error.

In addition, we extend Table 2 by considering non-invertible, non-linear transformations with Multi-Layer Perceptrons (MLPs) on the IHDP dataset to assess the robustness of *SimPONet* and baselines when the diffeomorphism assumption is violated. To this end, we used two-layer MLPs for g_t and g_t^S to generate covariates from Z. We present the results in Fig. 5, and make the following observations:

In the pre-treatment setting, causal representation extraction is unnecessary, making simulator supervision inconsequential; thus, we omit the four post-treatment CATE methods introduced in this paper. For other baselines, Fig. 5 shows that they perform significantly worse with post-treatment X than with pre-treatment Z, underscoring the importance of causal representation recovery. DragonNet achieved the best performance with pre-treatment covariates.

For Normalizing Flow-generated covariates, CATE error consistently decreased when baselines used simulator-based causal representations, $\widetilde{f}_t^S(X)$, rather than X, validating the utility of simulators in extracting causal representations. *SimPONet* achieved the best results, with larger gains in this setting.

In the MLP-based covariate experiments, *SimPONet* continued to outperform all baselines, displaying trends similar to those observed with Normalizing Flow-generated covariates. This indicates that the diffeomorphism assumption required for our theoretical results may be inconsequential in practice.

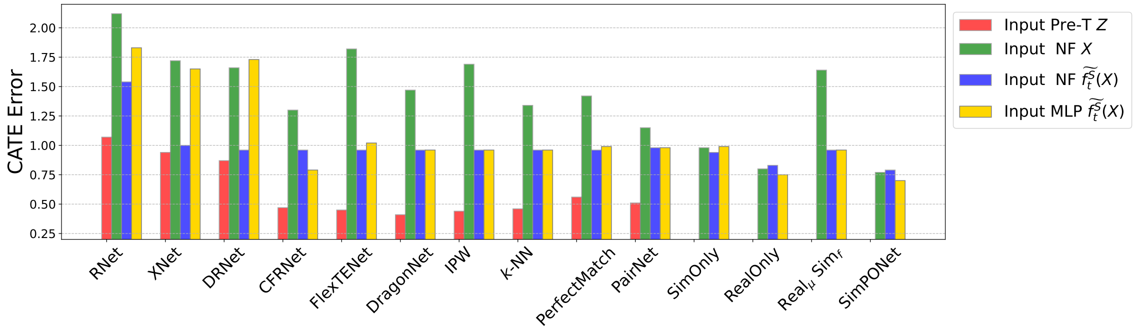

Figure 5: Comparing CATE errors under pre-treatment Z, and MLP, Normalizing flow generated post-treatment covariates X.

B.12 Table of Symbols

Symbol	Definition
X	Real post-treatment covariates: Random Variable
Y	Real outcomes: Random Variable
X^S	Simulator post-treatment covariates: Random Variable
Y^S	Simulator outcomes: Random Variable
T	Treatment: Random Variable
Z	Latent (unobserved) pre-treatment representations: Random Variable
D_{trn}	Observational training dataset from Real DGP
D_{syn}	Counterfactual dataset from Simulator DGP
D_{tst}	Test dataset from Real DGP
$\boldsymbol{x}$,$\boldsymbol{x}^S$,z,t,y,y^S	Realisations of random variables X,X^S,Z,T,Y,Y^S respectively
$\mathcal{X}$	Space of post-treatment covariate values: Set
$\mathcal{T}$	Space of treatment values: Set $=\{0,1\}$
$\mathcal{Z}$	Space of latents: Set
$\mathcal{Y}$	Space of outcomes: Set
n_z,n_x	Dimensions of vector spaces in which $\mathcal{Z}$,$\mathcal{X}$ lie
$Y_i(t)$	Potential outcome for i^{th} unit under treatment t
$X_i(t)$	Potential post-treatment covariate for i^{th} unit under treatment t
g_t	Mapping from $\mathcal{Z} \mapsto \mathcal{X}$, transforms latents to real post-treatment covariates under t
g_t^S	Mapping from $\mathcal{Z} \mapsto \mathcal{X}$, transforms latents to simulated post-treatment covariates under t
f_t	Mapping from $\mathcal{X} \mapsto \mathcal{Z}$, transforms real post-treatment covariates under t to latents
f_t^S	Mapping $\mathcal{X} \mapsto \mathcal{Z}$, transforms simulated post-treatment covariates under t to latents
P_Z	Probability distribution of latents Z
μ_t	Outcome function for real data under t
μ_t^S	Outcome function for simulated data under t
τ	Conditional Average Treatment Effect for real data, $\mu_1 - \mu_0$, Mapping $\mathcal{Z} \mapsto \mathcal{Y}$
τ^S	Conditional Average Treatment Effect for simulated data, $\mu_1^S - \mu_0^S$, Mapping $\mathcal{Z} \mapsto \mathcal{Y}$
$\circ$	Composition of functions
$\tau_X(\boldsymbol{x},t)$	Conditional Average Treatment Effect for real data, $\tau \circ f_t(\boldsymbol{x})$, Mapping $\mathcal{X} \times \mathcal{T} \mapsto \mathcal{Y}$
$\tau_X^S(\boldsymbol{x}^S,t)$	Conditional Average Treatment Effect for simulated data, $\tau^S \circ f_t^S(\boldsymbol{x}^S)$, Mapping $\mathcal{X} \times \mathcal{T} \mapsto \mathcal{Y}$
h	Diffeomorphic transformation, arises due to contrastive learning
$\mathbb{S}^d$	Unit-norm hypersphere of dimension d, Subset of $\mathbb{R}^{(d+1)}$
$d_{\boldsymbol{x} \mid t}$	Expected squared-distance between two functions on $P(X \mid T)$, see Section 3 for definition
d_z	Expected squared-distance between two functions on P_Z, see Section 3 for definition
$d_{h(z)}$	d_z under transformation h on z, see Section 3 for definition
$\text{sim}(\bullet,\bullet)$	Cosine similarity
$\widehat{f}_t$	Estimate for f_t
$\widehat{f}_t^S$	Estimate for f_t^S
$\widehat{\mu}_t$	Estimate for μ_t
$\widehat{\mu}_t^S$	Estimate for μ_t^S
$\widetilde{f}_t^S$	Estimate for f_t^S recovered from contrastive learning
$\widetilde{\mu}_t^S$	Estimate for μ_t^S on recovering Simulator DGP
$\mathcal{E}_{\text{CATE}}$	CATE estimation error
$\mathcal{E}_{\text{CATE}}^t$	CATE estimation error on covariates $\boldsymbol{x}$ under treatment t
$\mathcal{E}_F^t$	Factual error on treatment t samples
$\mathcal{E}_{CF}^t$	Counterfactual error on treatment t samples
K_μ	Lipschitz constant for μ_t,$\widehat{\mu}_t$
K_τ	Lipschitz constant for τ,$\widehat{\tau}$
K_{μ^S}	Lipschitz constant for μ_t^S,$\widehat{\mu}_t^S$,$\widetilde{\mu}_t^S$
K_{τ^S}	Lipschitz constant for τ^S,$\widehat{\tau}^S$,$\widetilde{\tau}^S$
$\boldsymbol{R}_t$	g_t for linear DGP: Matrix
$\boldsymbol{S}_t$	g_t^S for linear DGP: Matrix
w_t	μ_t for linear DGP: Vector
w_t^S	μ_t^S for linear DGP: Vector
w_τ	τ for linear DGP: Vector
w_τ^S	τ^S for linear DGP: Vector