Estimation of Individual Treatment Effect in Latent Confounder Models via Adversarial Learning

Changhee Lee, ¹ Nicholas Mastronarde, ² and Mihaela van der Schaar ^{3, 1}

Department of Electrical and Computer Engineering, University of California, Los Angeles, USA
Department of Electrical Engineering, University at Buffalo, New York, USA
Department of Engineering Science, University of Oxford, UK
chl8856@ucla.edu, nmastron@buffalo.edu, mihaela.vanderschaar@oxford-man.ox.ac.uk

Abstract

Estimating the individual treatment effect (ITE) from observational data is essential in medicine. A central challenge in estimating the ITE is handling confounders, which are factors that affect both an intervention and its outcome. Most previous work relies on the *unconfoundedness assumption*, which posits that all the confounders are measured in the observational data. However, if there are unmeasurable (latent) confounders, then *confounding bias* is introduced. Fortunately, noisy proxies for the latent confounders are often available and can be used to make an unbiased estimate of the ITE. In this paper, we develop a novel adversarial learning framework to make unbiased estimates of the ITE using noisy proxies.

Introduction

Understanding the individual treatment effect (ITE) on an outcome $\bf y$ of an intervention $\bf t$ on an individual with features $\bf x$ is a challenging problem in medicine. When inferring the ITE from observational data, it is common to assume that all of the confounders – factors that affect both the intervention and the outcome – are measurable and captured in the observed data as shown in Figure 1(a). However, in practice, there are often unobserved (latent) confounders $\bf z$ as shown in Figure 1(b). For example, socio-economic status cannot be directly measured, but can influence the types of medications that a subject has access to; therefore, it acts as a confounder between the medication and the patient's health. If such latent confounders are not appropriately accounted for, then the estimated ITE will be subject to *confounding bias*, making it impossible to estimate the effect of the intervention on the outcome without bias [1, 2]. A common technique for mitigating confounding bias is using *proxy variables*, which are measurable proxy for the latent confounders that can enable unbiased estimation of the ITE. For instance, in the causal diagram in Figure 1(b), $\bf x$ can be viewed as providing noisy proxies of the latent confounders $\bf z$.

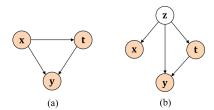


Figure 1: Causal diagrams. (a) x is an observed confounder between the intervention t and outcome y. (b) z is a latent confounder and x serves as a proxy that provides noisy views of z. Shaded and unshaded nodes denote observed and unobserved (latent) variables, respectively.

Contributions: We introduce an adversarial framework to infer the complex non-linear relationship between the proxy variables (i.e., observations) and the latent confounders via approximately recovering posterior distributions that can be used to infer the ITE. Our experiments on synthetic and

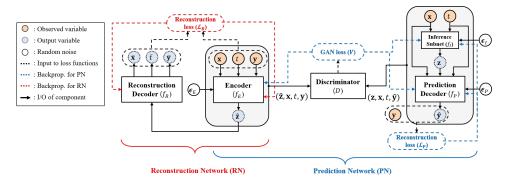


Figure 2: An illustration of the proposed network architecture.

semi-synthetic observational datasets show that the proposed method is competitive with – and often outperforms – state-of-the-art methods when there are no latent confounders or when the proxy noise is small, and outperforms all tested benchmarks when the proxy variables become noisy.

Causal Effect with Latent Confounders

Our goal is to estimate the ITE from an observational dataset $\mathcal{D} = \{(\mathbf{x}_i, t_i, \mathbf{y}_i)\}_{i=1}^N$, where \mathbf{x}_i, t_i , and \mathbf{y}_i , denote the *i*-th subject's feature vector, treatment (we assume that the treatment is binary, i.e., $t \in \{0, 1\}$), and outcome vector, respectively, and N is the number of subjects. The ITE for a subject with observed potential confounder \mathbf{x} is defined as

$$ITE(\mathbf{x}) = \mathbb{E}\left[\mathbf{y}|\mathbf{x}, do(t=1)\right] - \mathbb{E}\left[\mathbf{y}|\mathbf{x}, do(t=0)\right].$$
 (1)

To recover the ITE under the latent confounder model in Figure 1(b), we need to identify $p(\mathbf{y}|\mathbf{x}, do(t=1))$ and $p(\mathbf{y}|\mathbf{x}, do(t=0))$. The former can be calculated as follows:

$$p(\mathbf{y}|\mathbf{x},do(t=1)) = \int_{\mathbf{z}} p(\mathbf{y}|\mathbf{z},\mathbf{x},do(t=1))p(\mathbf{z}|\mathbf{x},do(t=1))d\mathbf{z} = \int_{\mathbf{z}} p(\mathbf{y}|\mathbf{z},\mathbf{x},t=1)p(\mathbf{z}|\mathbf{x})d\mathbf{z}, \quad (2)$$

where the second equality follows from the rules of do-calculus¹ applied to the causal graph in Figure 1(b) [3]. $(p(\mathbf{y}|\mathbf{x}, do(t=0)))$ can be derived similarly.) It is worth to highlight that $p(\mathbf{y}|\mathbf{x}, do(t=1))$ is equivalent to $p(\mathbf{y}|\mathbf{x}, t=1)$ if the unconfoundedness assumption holds as in Figure 1(a).

Thus, from (1) and (2), we can make estimates of the ITE without confounding bias using the estimates of the conditional distributions $p(\mathbf{y}|\mathbf{z}, \mathbf{x}, t)$ and $p(\mathbf{z}|\mathbf{x})$. Since \mathbf{z} is unobservable, we assume that the joint distribution $p(\mathbf{z}, \mathbf{x}, t, \mathbf{y})$ can be approximately recovered solely from the observations $(\mathbf{x}, t, \mathbf{y})$ as justified in [4].

Adversarial Learning for Causal Effect

In this section, we propose a method that estimates *Causal Effect using a Generative Adversarial Network (CEGAN)*. CEGAN's objective is to estimate the conditional posteriors in (2) under the causal graph in Figure 1(b) so that we can estimate the ITE (1) for new subjects. However, since we cannot measure the *true* latent confounder, we are unable to directly learn the posterior distribution $p(\mathbf{z}|\mathbf{x})$. Instead, we learn a mapping between the data (observations) and an arbitrary latent space following an adversarial learning framework similar to those developed in [5] and [6].

Our model, depicted in Figure 2, comprises a prediction network (right) and a reconstruction network (left). Each network includes an encoder-decoder pair, where the encoder is shared between them. The posterior distributions that are required to solve (2) can be estimated using bidirectional models [5] and [6] via factorizing the posterior distribution as $p(\mathbf{y}|\mathbf{z},\mathbf{x},t) \approx q_P(\mathbf{y}|\mathbf{z},\mathbf{x},t)$ and $p(\mathbf{z}|\mathbf{x}) \approx \sum_{t \in \{0,1\}} q_I(\mathbf{z}|\mathbf{x},t) q(t|\mathbf{x})$, where q_P and q_I are the components of the prediction network and q is the propensity score. Meanwhile, the reconstruction network is a denoising autoencoder [7], which helps the prediction network find a meaningful mapping to the latent space that preserves information in the data space.

¹The *do*-operator [1] simulates physical interventions by deleting certain functions from the model, replacing them with a constant value, while keeping the rest of the model unchanged.

Prediction Network

The prediction network has two components: a generator (which consists of the encoder f_E , the prediction decoder f_P , and the inference subnetwork f_I) and a discriminator D.

The **encoder** (f_E) , which is employed in both the reconstruction and prediction networks, maps the data space to the latent space. Thus, the output of the encoder $\hat{\mathbf{z}}$ is given by $\hat{\mathbf{z}} = f_E(\mathbf{x}, t, \mathbf{y}, \epsilon_E)$. The **inference subnetwork** (f_I) is introduced to infer \mathbf{z} based on \mathbf{x} and given t; its output \mathbf{z} is given by $\mathbf{z} = f_I(\mathbf{x}, t, \epsilon_I)$. The **prediction decoder** (f_P) is a function that outputs the estimated outcome $\hat{\mathbf{y}}$ given a sample (\mathbf{x}, t) drawn from the data distribution $p_d(\mathbf{x}, t)$ and a latent variable $\mathbf{z} \sim q_I(\mathbf{z}|\mathbf{x}, t)$ inferred by f_I ; thus, $\hat{\mathbf{y}} = f_P(\mathbf{z}, \mathbf{x}, t, \epsilon_P)$. Note that the outputs of the generator are randomized by the noise term $\epsilon_E, \epsilon_I, \epsilon_P \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ using the universal approximator technique described in [8].

With the conditional probabilities $q_E(\mathbf{z}|\mathbf{x},t,\mathbf{y}), q_I(\mathbf{z}|\mathbf{x},t)$, and $q_P(\mathbf{y}|\mathbf{z},\mathbf{x},t)$ obtained from the generator, we are able to define two joint distributions: $q_E(\mathbf{z},\mathbf{x},t,\mathbf{y}) = p_d(\mathbf{x},t,\mathbf{y})q_E(\mathbf{z}|\mathbf{x},t,\mathbf{y})$ for the encoder and $q_P(\mathbf{z},\mathbf{x},t,\mathbf{y}) = p_d(\mathbf{x},t)q_I(\mathbf{z}|\mathbf{x},t)q_P(\mathbf{y}|\mathbf{z},\mathbf{x},t)$ for the prediction decoder. Using tuples drawn from the two joint distributions, CEGAN attempts to match these distribution by playing an adversarial game between the generator and the discriminator. To do so, the **prediction discriminator** (D) maps tuples $(\mathbf{z},\mathbf{x},t,\mathbf{y})$ to a probability in [0,1]. Specifically, $D(\mathbf{z},\mathbf{x},t,\mathbf{y})$ and $1-D(\mathbf{z},\mathbf{x},t,\mathbf{y})$ denote estimates of the probabilities that the tuple $(\mathbf{z},\mathbf{x},t,\mathbf{y})$ is drawn from $q_E(\mathbf{z},\mathbf{x},t,\mathbf{y})$ and $q_P(\mathbf{z},\mathbf{x},t,\mathbf{y})$, respectively. The discriminator tries to distinguish between tuples $(\mathbf{z},\mathbf{x},t,\mathbf{y})$ that are drawn from $q_E(\mathbf{z},\mathbf{x},t,\mathbf{y})$ and $q_P(\mathbf{z},\mathbf{x},t,\mathbf{y})$. Following the framework in [5], the two distributions can be matched (i.e., they reach the same saddle point) by solving the following min-max problem between the generator and the discriminator:

$$\min_{(\theta_E, \theta_I, \theta_P)} \max_{\theta_D} \ \mathbb{E}_{q_E(\mathbf{z}, \mathbf{x}, t, \mathbf{y})} \Big[\log \left(D(\hat{\mathbf{z}}, \mathbf{x}, t, \mathbf{y}) \right) \Big] + \mathbb{E}_{q_P(\mathbf{z}, \mathbf{x}, t, \mathbf{y})} \Big[\log \left(1 - D(\mathbf{z}, \mathbf{x}, t, \hat{\mathbf{y}}) \right) \Big].$$
(3)

Reconstruction Network

The relationship between the data and the latent space is not specified in the prediction network. Consequently, the network may converge to an undesirable matched joint distribution. For instance, it may learn to match the joint distributions $q_E(\mathbf{z}, \mathbf{x}, t, \mathbf{y})$ and $q_P(\mathbf{z}, \mathbf{x}, t, \mathbf{y})$ while inferring latent variables \mathbf{z} that provide no information about the data samples $(\mathbf{x}, t, \mathbf{y})$. We introduce a reconstruction network to nudge the prediction network toward learning a meaningful mapping between the data and latent spaces. We utilize a denoising autoencoder for the reconstruction network, which employs the same encoder as the prediction network, f_E , and a **reconstruction decoder** f_R . f_R reconstructs the original input of the encoder f_E from the output of f_E ; the output can be given as $(\bar{\mathbf{x}}, \bar{t}, \bar{\mathbf{y}}) = f_R(\hat{\mathbf{z}})$.

Then, we define the following reconstruction loss:

$$\mathcal{L}_R(\mathbf{w}, \bar{\mathbf{w}}) = \ell(\mathbf{x}, \bar{\mathbf{x}}) + \ell(t, \bar{t}) + \ell(\mathbf{y}, \bar{\mathbf{y}}), \tag{4}$$

where $\mathbf{w} = [\mathbf{x}, t, \mathbf{y}], \bar{\mathbf{w}} = [\bar{\mathbf{x}}, \bar{t}, \bar{\mathbf{y}}], \ell(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|^2$ for continuous values, and $\ell(\mathbf{a}, \mathbf{b}) = -\mathbf{a}^T \log \mathbf{b} - (\mathbf{1} - \mathbf{a})^T \log (\mathbf{1} - \mathbf{b})$ for binary values. Here, log denotes the element-wise logarithm. By minimizing (4) iteratively with the min-max problem (3), f_E is able to map data samples into the latent space while preserving information that is available in the data space.

Experiments

Ground truth counterfactual outcomes are never available in observational datasets, which makes it difficult to evaluate causal inference methods. Thus, we evaluate CEGAN against various benchmarks using a semi-synthetic dataset where we model the proxy mechanism to generate latent confounding. In the appendix, we perform further comparisons using a semi-synthetic dataset suggested in [4] and a synthetic dataset.

Performance Metric: We use two different performance metrics in our evaluations – expected precision in the estimation of heterogeneous effect (PEHE) and average treatment effect (ATE) [9]:

$$\epsilon_{\text{PEHE}} = \frac{1}{N} \sum_{i=1}^{N} \Big(\big(y_i(1) - y_i(0) \big) - \big(\hat{y}_i(1) - \hat{y}_i(0) \big) \Big)^2, \ \ \epsilon_{\text{ATE}} = \Big| \frac{1}{N} \sum_{i=1}^{N} \big(y_i(1) - y_i(0) \big) - \big(\hat{y}_i(1) - \hat{y}_i(0) \big) \Big|,$$

Table 1: Comparison of $\sqrt{\epsilon_{\rm PEHE}}$ and $\epsilon_{\rm ATE}$ (mean \pm std) on the TWINS dataset.

Method	$\sqrt{\epsilon_{ ext{PEHE}}}$				$\epsilon_{ m ATE}$			
	no latent confounding		latent confounding		no latent confounding		latent confounding	
	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample
LR-1	0.365 ± 0.00	0.367 ± 0.00	0.413 ± 0.01	0.423 ± 0.02	0.045 ± 0.02	0.186 ± 0.03	0.064 ± 0.02	0.206 ± 0.03
LR-2	0.404 ± 0.02	0.411 ± 0.02	0.442 ± 0.02	0.454 ± 0.02	0.128 ± 0.03	0.206 ± 0.04	0.148 ± 0.03	0.227 ± 0.04
kNN	0.486 ± 0.02	0.506 ± 0.02	0.492 ± 0.02	0.515 ± 0.02	0.254 ± 0.04	0.264 ± 0.04	0.271 ± 0.04	0.285 ± 0.04
CForest	0.356 ± 0.01	0.372 ± 0.01	0.417 ± 0.02	0.429 ± 0.02	0.025 ± 0.02	0.188 ± 0.03	0.023 ± 0.02	0.186 ± 0.03
BART	0.569 ± 0.06	0.562 ± 0.06	0.877 ± 0.08	0.871 ± 0.08	0.432 ± 0.08	0.429 ± 0.08	0.790 ± 0.09	0.786 ± 0.09
CMGP	0.367 ± 0.01	0.365 ± 0.01	0.430 ± 0.05	0.438 ± 0.05	0.034 ± 0.03	0.036 ± 0.04	0.192 ± 0.09	0.213 ± 0.09
CFR _{WASS}	0.371 ± 0.03	0.371 ± 0.03	0.427 ± 0.05	0.438 ± 0.05	0.056 ± 0.06	0.071 ± 0.06	0.205 ± 0.07	0.226 ± 0.07
CEVAE	0.363 ± 0.00	0.364 ± 0.00	0.423 ± 0.00	0.428 ± 0.00	0.071 ± 0.01	0.165 ± 0.01	0.088 ± 0.01	0.183 ± 0.01
CEGAN	0.363 ± 0.00	0.362±0.00	0.369 ± 0.00	0.369 ± 0.00	0.018 ± 0.01	0.017 ± 0.01	0.022 ± 0.01	0.021 ± 0.02

where $y_i(1)$ and $y_i(0)$ are the ground truth of the treated and controlled outcomes for the *i*-th sample and $\hat{y}_i(1)$ and $\hat{y}_i(0)$ are their estimates.

We compare CEGAN against benchmarks (see appendix for details of the tested benchmarks) using a semi-synthetic dataset (**TWINS**) which is similar to that was first proposed in [4]. Based on records of twin births in the USA from 1989-1991 [10], we artificially create a binary treatment such that t=1 (t=0) denotes being born the heavier (lighter). The binary outcome is the mortality of each of the twins in their first year. (Since we have records for both twins, we treat their outcomes as two potential outcomes, i.e., $\mathbf{y}(1)$ and $\mathbf{y}(0)$, with respect to the treatment assignment of being born heavier.) Due to its high correlation with the outcome [11, 12], we select the feature 'GESTAT', which is the gestational age in weeks. The treatment assignment is based only on this single variable, i.e., $t_i|z_i \sim \text{Bern}(\sigma(wz_i))$, where $w \sim \mathcal{N}(10, 0.1^2)$ and z is the min-max normalized value of 'GESTAT'. The data generation process is not exactly equivalent to that proposed in [4] as i) it includes artificial proxies of the latent variable in the observational dataset, which is less realistic and ii) the treatment assignment is not only based on the latent variable but also on the observed variables which is not consistent with the causal model in Figure 1(b). (In the appendix, we reported details and results for the TWINS dataset with the same data generation process in [4].)

To assess the performance of causal inference methods in the presence of latent confounding, we test them on two datasets: "no latent confounding" which contains 'GESTAT' and relies on the unconfoundedness assumption as depicted in Figure 1(a) and "latent confounding" which excludes 'GESTAT' from the observational dataset and follows the latent causal graph in Figure 1(b).

Throughout the evaluation, we average over 100 Monte Carlo samples from the estimated posteriors derived using each method to compute $\mathbb{E}(\mathbf{y}|\mathbf{x},do(t=1))$ and $\mathbb{E}(\mathbf{y}|\mathbf{x},do(t=0))$ in (1) for CEVAE and CEGAN. The reported values in Table 1 are averaged over 50 realizations with the same 64/16/20 train/validation/test splits.

The performance of $\sqrt{\epsilon_{\text{PEHE}}}$ and ϵ_{ATE} is reported in Table 1, for both within-sample and out-of-sample tests. The ITE estimation accuracy decreases for all of the evaluated methods after removing 'GESTAT' from the observational dataset due to information loss and confounding bias due to the latent confounder. CEGAN provides competitive performance compared to the state-of-the-art when there is no latent confounding, while outperforming all benchmarks under latent confounding for both $\sqrt{\epsilon_{\text{PEHE}}}$ and ϵ_{ATE} . Under the circumstances when there is latent confounding and the treatment assignment is solely based on this latent confounder, CEGAN provides more robust performance than CEVAE.

Conclusion

In this paper, we studied the problem of estimating causal effects in the latent confounder model. In order to obtain unbiased estimates of the ITE, we introduced a novel method, CEGAN, which utilizes an adversarially learned bidirectional model along with a denoising autoencoder. CEGAN achieves competitive performance with numerous state-of-the-art benchmarks when the *unconfoundedness assumption* holds or the proxy noise is small, while outperforming state-of-the-art causal inference methods when latent confounding is present. CEGAN performs especially well when the proxy noise is large and the treatment is determined based solely on the latent confounders.

References

- [1] Judea Pearl. Causality: Models, Reasoning, and Inference. Cambridge University Press, 2000.
- [2] Manabu Kuroki and Judea Pearl. Measurement bias and effect restoration in causal inference. *Biometrika*, 101(2):423437, March 2014.
- [3] Judea Pearl. On measurement bias in causal inference. In Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (AUAI 08), page 425432, 2010.
- [4] Christos Louizos, Uri Shalit, Joris Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. *In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.
- [5] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, 2017.
- [6] Jeff Donahue, Philipp Krhenbhl, and Trevor Darrell. Adversarial feature learning. *In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, 2017.
- [7] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408, 2010.
- [8] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *Proc.* 2016, 2016.
- [9] Jennifer L. Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217240, 2011.
- [10] Douglas Almond, Kenneth Y. Chay, and David S. Lee. The costs of low birth weight. *The Quarterly Journal of Economics*, 120(3):10311083, 2005.
- [11] Kath Moser, Alison Macfarlane, Yuan Huang Chow, Lisa Hilder, and Nirupa Dattani. Introducing new data on gestation-specific infant mortality among babies born in 2005 in england and wales. *Health Stat Q.*, 35:13–27, 2007.
- [12] M. J. Platt. Outcomes in preterm infants. *Health Stat Q.*, 128(5):399–403, 2014.
- [13] Richard K. Crump, V. Joseph Hotz, Guido W. Imbens, and Oscar A. Mitnik. Nonparametric tests for treatment effect heterogeneity. *The Review of Economics and Statistics*, 90(3):389405, 2008.
- [14] Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.
- [15] Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 2017.
- [16] Uri Shalit, Fredrik D. Johansson, and David Sontag. Estimating individual treatment effect: Generalization bounds and algorithms. *In Proceedings of the 33rd International Conference on International Conference on Machine Learning (ICML 2016)*, 2016.
- [17] Ahmed M Alaa and Mihaela van der Schaar. Bayesian inference of individualized treatment effects using multi-task gaussian processes. *In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.
- [18] Elizabeth S Allman, Catherine Matias, and John A Rhodes. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, pages 3099–3132, 2009.

Appendix

Optimization of CEGAN

CEGAN is trained in an iterative fashion: we alternate between optimizing the reconstruction network and the prediction network until convergence. In this section, we describe the empirical loss functions that are used to optimize each component. Pseudo-code for training CEGAN is provided in Algorithm

Algorithm 1 Pseudo-code of CEGAN

Input: Observational dataset \mathcal{D}

Output: CEGAN parameters $(\theta_E, \theta_I, \theta_P, \theta_D, \theta_R)$

Initialize $(\theta_E, \theta_I, \hat{\theta}_P, \theta_D, \theta_R)$

repeat

1) Reconstruction network optimization

Sample minibatch of k_r data and noise samples

Update f_E , f_R using stochastic gradient descent (SGD) with gradient:

$$\nabla_{(\theta_E, \theta_R)} \frac{1}{k_r} \sum_{i=1}^{k_r} \mathcal{L}_R(\mathbf{w}_i, \bar{\mathbf{w}}_i)$$

2) Prediction network optimization

Sample minibatch of k_d data and noise samples

Update *D* using SGD with gradient:

$$-\nabla_{\theta_D} \frac{1}{k_d} \sum_{i=1}^{k_d} V(\mathbf{w}_i, \hat{\mathbf{z}}_i, \hat{\mathbf{y}}_i)$$

Sample minibatch of k_g data and noise samples

Update f_E , f_I , f_P using SGD with gradient:

$$\nabla_{(\theta_E, \theta_I, \theta_P)} \frac{1}{k_g} \sum_{i=1}^{k_g} \left[V(\mathbf{w}_i, \hat{\mathbf{z}}_i, \hat{\mathbf{y}}_i) + \alpha \mathcal{L}_P(\mathbf{y}_i, \hat{\mathbf{y}}_i) \right]$$

until convergence

We train the reconstruction network (f_E, f_R) by optimizing the following objective in a supervised fashion:

$$\underset{(\theta_E,\theta_R)}{\text{minimize}} \quad \sum_{i=1}^m \mathcal{L}_R(\mathbf{w}_i, \bar{\mathbf{w}}_i)$$

where $\mathbf{w}_i = [\mathbf{x}_i, t_i, \mathbf{y}_i]$ and $\bar{\mathbf{w}}_i = [\bar{\mathbf{x}}_i, \bar{t}_i, \bar{\mathbf{y}}_i]$. For the prediction network, we define an empirical value function for the min-max optimization problem in (3):

$$V(\mathbf{w}_i, \hat{\mathbf{z}}_i, \hat{\mathbf{y}}_i) = \log D(\hat{\mathbf{z}}_i, \mathbf{x}_i, t_i, \mathbf{y}_i) + \log(1 - D(\mathbf{z}_i, \mathbf{x}_i, t_i, \hat{\mathbf{y}}_i))$$
.

In addition, we define the following reconstruction loss at the prediction decoder f_P :

$$\mathcal{L}_P(\mathbf{y}_i, \hat{\mathbf{y}}_i) = \ell(\mathbf{y}_i, \hat{\mathbf{y}}_i),$$

where $\ell(\cdot)$ is defined as in (4). Overall, the discriminator and generator iteratively optimize the following objectives, where α is a trade-off parameter:

minimize
$$-\sum_{i=1}^{m} V(\mathbf{w}_{i}, \hat{\mathbf{z}}_{i}, \hat{\mathbf{y}}_{i})$$
minimize
$$\sum_{i=1}^{m} \left[V(\mathbf{w}_{i}, \hat{\mathbf{z}}_{i}, \hat{\mathbf{y}}_{i}) + \alpha \mathcal{L}_{P}(\mathbf{y}_{i}, \hat{\mathbf{y}}_{i}) \right]$$
(5)

Table 2: Comparison of	FREHE and FATE	(mean + std) on the	e TWINS dataset	with Scenario 1.

	$\sqrt{\epsilon_{ m PEHE}}$				$\epsilon_{ ext{ATE}}$			
Method	p = 0.1		p = 0.5		p = 0.1		p = 0.5	
	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample
LR-1	0.373 ± 0.00	0.365 ± 0.00	0.379 ± 0.00	0.370 ± 0.00	0.025 ± 0.01	0.021 ± 0.01	0.069 ± 0.02	0.064 ± 0.02
LR-2	0.376 ± 0.00	0.374 ± 0.01	0.384 ± 0.01	0.381 ± 0.01	0.021 ± 0.01	0.016 ± 0.01	0.069 ± 0.02	0.063 ± 0.02
kNN	0.385 ± 0.01	0.398 ± 0.01	0.409 ± 0.01	0.422 ± 0.01	0.020 ± 0.02	0.019 ± 0.02	0.116 ± 0.03	0.111 ± 0.03
CForest	0.400 ± 0.30	0.410 ± 0.26	0.409 ± 0.33	0.416 ± 0.27	0.016 ± 0.01	0.024 ± 0.01	0.055 ± 0.02	0.066 ± 0.02
BART	0.400 ± 0.02	0.397 ± 0.02	0.455 ± 0.03	0.456 ± 0.03	0.074 ± 0.05	0.078 ± 0.05	0.234 ± 0.06	0.241 ± 0.07
CMGP	0.377 ± 0.01	0.370 ± 0.02	0.380 ± 0.01	0.371 ± 0.01	0.054 ± 0.02	0.049 ± 0.02	0.078 ± 0.03	0.072 ± 0.03
CFR _{WASS}	0.373 ± 0.00	0.366 ± 0.00	0.379 ± 0.01	0.373 ± 0.01	0.024 ± 0.02	0.021 ± 0.02	0.068 ± 0.03	0.063 ± 0.03
CEVAE	0.369 ± 0.00	0.367 ± 0.00	0.373 ± 0.00	0.368 ± 0.01	0.015 ± 0.01	0.020 ± 0.01	0.032 ± 0.02	0.027 ± 0.02
CEGAN	0.371 ± 0.00	0.364±0.00	0.372 ± 0.00	0.366 ± 0.00	0.021 ± 0.01	0.016 ± 0.01	0.030 ± 0.01	0.026±0.01

When optimizing CEGAN, the prediction network's loss \mathcal{L}_P is used as a regularizer to improve training compared to using only the GAN loss (5), and the reconstruction network's loss \mathcal{L}_R drives the learning process. Specifically, we train (f_E, f_R) to minimize \mathcal{L}_R iteratively with the GAN loss (5), which forces f_E to learn a latent mapping \mathbf{z} that is informative enough to reconstruct $(\mathbf{x}, \mathbf{y}, t)$ and, thus, drives f_I to favor a meaningful latent structure over a trivial one.

Additional Experiments

Benchmarks

We compare CEGAN with several cutting-edge methods including logistic regression using treatment as a feature (**LR-1**), logistic regression separately trained for each treatment assignment (**LR-2**), k-nearest neighbor (**kNN**) [13], Bayesian additive regression trees (**BART**) [14], causal forests (**CForest**) [15], counterfactual regression with Wasserstein distance (**CFR**_{WASS}) [16]², multi-task Gaussian process (**CMGP**) [17] and Causal Effect VAE (**CEVAE**) [4]³. For continuous outcomes, logistic regressions are replaced with least squares linear regressions. We also compare against CEGAN trained only with \mathcal{L}_P (**CEGAN**(\mathcal{L}_P)), which is equivalent to a feed-forward network consisting of f_I and f_P . Consequently, CEGAN(\mathcal{L}_P) does not exploit adversarial learning and does not account for latent confounders (i.e., the **z** inferred by f_I is not trained to have a meaningful relationship with the data samples (**x**, t, **y**)).

Simulation Settings

Unless otherwise specified, we set $\alpha=1$ in (5) and assume a 20-dimensional latent space for ${\bf z}$. A fully-connected network is used for each component of the prediction network (i.e., f_E , f_P , f_I , and D) and a multi-output network is used for the reconstruction network f_R . Each of these networks comprise 3 layers, 200 hidden units in each layer, and ReLU activation functions. The networks are trained using an Adam optimizer with a minibatch size of 64 and a learning rate of 10^{-4} . A dropout probability of 0.6 is assumed, and Xavier and zero initializations are applied for weight matrices and bias vectors, respectively. CEGAN is implemented using Tensorflow.

Semi-Synthetic Dataset: TWINS proposed in [4]

In this subsection, we compare CEGAN against the aforementioned benchmarks using a semi-synthetic dataset that was first proposed in [4]. The dataset is based on records of twin births in the USA from 1989-1991 [10]. Using this real-world dataset, we artificially create a binary treatment such that t=1 (t=0) denotes being born the heavier (lighter) twin. The binary outcome corresponds to the mortality of each of the twins in their first year. Since we have records for both twins, we treat their outcomes as two potential outcomes, i.e., $\mathbf{y}(1)$ and $\mathbf{y}(0)$, with respect to the treatment assignment of being born heavier. To make a semi-synthetic dataset, we choose same-sex twins, discard features that are only available after birth, and focus on cases where both twins have birth weights below 2 kg. Overall, we have a dataset of 10,286 twins with 49 features related to the parents,

²https://github.com/clinicalml/cfrnet

³https://github.com/AMLab-Amsterdam/CEVAE

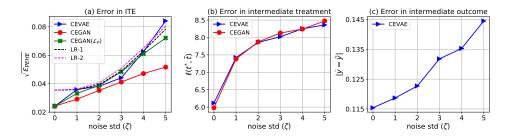


Figure 3: Performance evaluation on the synthetic dataset. The x-axes denote the standard deviation of the noise (ζ) in the proxy mechanism mapping z to x. (a) PEHE vs. ζ for CEGAN and CEVAE. LR-1, LR-2, and CEGAN(\mathcal{L}_P) are included for reference. (b) Cross-entropy between t^* and \tilde{t} . (c) Absolute error between \dot{y} and \tilde{y} .

the pregnancy, and the birth.⁴ The mortality rate of the lighter twin (t = 0) is 21.64% and the heavier twin (t = 1) is 15.32%, which yields an average treatment effect of -6.32%.

For the TWINS dataset whose data generation process is equivalent to what was proposed in [4], we base our treatment assignment on the feature 'GESTAT10', which is a categorical value from 0 to 9 representing the number of gestation weeks. (In this experiment, 'GESTAT' is discarded; see the description in the manuscript.) We then follow the treatment and noisy proxy generation procedures reported in [4]. Specifically, we let $t_i|\mathbf{x}_i,z_i\sim \mathrm{Bern}(\sigma(w_o^T\mathbf{x}+w_h(\frac{z}{10}-0.1)))$, where $\sigma(\cdot)$ denotes the sigmoid function, $w_o\sim\mathcal{N}(0,0.1\cdot I)$, and $w_h\sim\mathcal{N}(9,0.1)^5$, and we artificially generate noisy proxies by using three randomly flipped replicas of one-hot encoded 'GESTAT10' with flipping probability p. It is worth to highlight that, compared to the TWINS dataset proposed in the manuscript, artificial proxy variables are created based on the gestational age feature and included as additional observed features, and the treatment depends not only on the latent variable but also on these artificial proxies.

The $\sqrt{\epsilon_{\text{PEHE}}}$ and ϵ_{ATE} results are reported in Table 2 for both in-sample and out-of-sample tests. When the proxy noise is relatively small (p=0.1), CEGAN and CEVAE achieve comparable performance to other benchmarks. This aligns with the well-known result that three independent views of a latent feature guarantee that it can be recovered [18], so even techniques that do not account for latent confounders can make accurate predictions. In contrast, when the artificial proxy variables become too noisy to be useful (p=0.5), CEGAN and CEVAE achieve comparable performance to each other and outperform the other benchmarks due to their robustness to latent confounders. However, the artificially generated treatments t_i in this data set are conditioned on both \mathbf{x}_i and z_i , which is inconsistent with the causal diagram in Figure (b), where t_i only depends on the latent features z_i .

Synthetic dataset: toy example

To further illustrate the robustness of CEGAN to latent confounders, we generate a synthetic dataset as follows:

$$z_{ij} \sim \mathcal{N}(3(\mu - 1), 1^2) \text{ for } j = 1, \dots, d_z$$

$$\mathbf{x}_i | \mathbf{z}_i = \mathbf{z}_i + \mathbf{n}$$

$$t_i | \mathbf{z}_i \sim \text{Bern}(\sigma(0.25 \cdot z_{id_z}))$$

$$y_i | \mathbf{z}_i, t_i = \sigma(\mathbf{1}^T \mathbf{z}_i + (2t_i - 1)),$$
(6)

where $\mu \sim \text{Bern}(0.5)$, $\mathbf{n} \sim \mathcal{N}(0, \zeta^2 \mathbf{I})$ $(0 \le \zeta \le 5)$, and $\sigma(\cdot)$ is the sigmoid function. We assume a binary treatment $t \in \{0, 1\}$, a one-dimensional $y \in [0, 1]$, and 5-dimensional \mathbf{z} and \mathbf{x} , i.e., $d_z = d_x = 5$.

Since we only have access to observations (\mathbf{x}, y, t) , where \mathbf{x} is a noisy proxy of \mathbf{z} , the above generation process introduces latent confounding between t and y through \mathbf{z} as illustrated in Figure

⁴We made every effort to faithfully reproduce the dataset from its source [10] using the same criteria as in [4], but did not end up with the same number of twins or features.

⁵Since we have four more features, we did not obtain comparable ϵ_{ATE} using $w_h \sim \mathcal{N}(5, 0.1)$ as reported in [4]. So, we calibrated the mean of w_h from 5 to 9 to achieve similar results.

1(b). Without measuring z, we expect causal inference methods will suffer from confounding bias. In our experiments, we evaluate over a sample size N=5000 and average over 50 realizations of the outcomes with the same 64/16/20 train/validation/test splits.

In Figure 3(a), using out-of-sample tests, we illustrate how $\sqrt{\epsilon_{\text{PEHE}}}$ varies with the standard deviation of the noise (ζ) in the proxy mechanism that maps \mathbf{z} to \mathbf{x} . The PEHE increases with the noise under all evaluated benchmarks because the conditional entropy of \mathbf{z} given the proxy \mathbf{x} , i.e., $H(\mathbf{z}|\mathbf{x}) = -\mathbb{E}_{p(\mathbf{x},\mathbf{z})} [\log p(\mathbf{z}|\mathbf{x})]$, is proportional to $\log \zeta$. However, CEGAN is more robust to the noise than the other benchmarks – including CEVAE, which considers latent confounders.

Following the previous discussion regarding its relationship to CEVAE, we believe that CEGAN performs better than CEVAE because it does not require as many intermediate steps to infer \mathbf{z} when estimating the ITE (1). In particular, both CEVAE and CEGAN need to predict an intermediate treatment assignment, i.e., $\tilde{t} \sim q(t|\mathbf{x}=\mathbf{x}^*)$, while CEVAE also needs to predict an intermediate outcome, i.e., $\tilde{y} \sim q(y|\mathbf{x}=\mathbf{x}^*,t=\tilde{t})$, where (\mathbf{x}^*,t^*,y^*) denote the true observations and (\tilde{t},\tilde{y}) denote the intermediate predictions. Since the treatment is binary, we adopt the cross-entropy $\ell(t^*,\tilde{t})$ defined in (4) between t^* and \tilde{t} to quantify the error in the predicted intermediate treatment \tilde{t} . This error affects both CEVAE and CEGAN. To evaluate the error in predicting the intermediate outcome \tilde{y} we compute the absolute difference between \dot{y} and \tilde{y} , i.e., $|\dot{y}-\tilde{y}|$, where $\dot{y} \sim q(y|\mathbf{x}=\mathbf{x}^*,t=t^*)$ is the intermediate outcome conditioned on t^* instead of \tilde{t} . This error only affects CEVAE. In Figure 3(b) and 3(c), we show how $\ell(t^*,\tilde{t})$ and $|\dot{y}-\tilde{y}|$ vary with respect to the standard deviation of the noise (ζ) , respectively.

Figure 3(b) and 3(c) demonstrate that the intermediate predictions made in both CEGAN and CEVAE become less accurate as the noise increases. We conjecture that this error is accumulated and propagated to the inference of **z** and eventually decreases the accuracy of the ITE estimates. Consequently, since CEVAE requires more intermediate steps to infer **z**, it performs worse than CEGAN. Note that, we omit error measurements in the latent space because (i) differences between latent variables do not necessarily correspond to the accuracy of predictions based on them and (ii) we cannot directly compare errors in the different latent spaces generated by CEVAE and CEGAN.