DR-VIDAL - Doubly Robust Variational Information-theoretic Deep Adversarial Learning for Counterfactual Prediction and Treatment Effect Estimation

SHANTANU GHOSH (shg121@pitt.edu)
University of Pittsburgh https://orcid.org/0000-0003-4085-541X

Zheng Feng
University of Florida

Jiang Bian
University of Florida https://orcid.org/0000-0002-2238-5429

Kevin Butler
University of Florida

Mattia Prosperi
University of Florida https://orcid.org/0000-0002-9021-5595

Article

Keywords: Causal Inference, ITE Estimation, Deep Learning, Information Theory, Doubly Robust

Posted Date: December 6th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1055169/v1

DR-VIDAL - Doubly Robust Variational Information-theoretic Deep Adversarial Learning for Counterfactual Prediction and Treatment Effect Estimation

Abstract

Determining causal effects of interventions onto outcomes from observational (non-randomized) data, e.g., treatment repurposing using electronic health records, is challenging due to underlying bias. Causal deep learning has improved over traditional techniques for estimating individualized treatment effects. We present the Doubly Robust Variational Information-theoretic Deep Adversarial Learning (DR-VIDAL), a novel generative framework that combines two joint models of treatment and outcome, ensuring an unbiased estimation even when one of the two is misspecified. DR-VIDAL uses a variational autoencoder (VAE) to factorize confounders into latent variables according to causal assumptions; then, an information-theoretic generative adversarial network (Info-GAN) is used to generate counterfactuals; finally, a doubly robust block incorporates propensity matching/weighting into predictions. On synthetic and real-world datasets, DR-VIDAL achieves better performance than other non-generative and generative methods. In conclusion, DR-VIDAL uniquely fuses causal assumptions, VAE, Info-GAN, and doubly robustness into a comprehensive, performant framework. Code is available at: https://bitbucket.org/goingdeep2406/dr-vidal/src/master/

Keywords: Causal Inference, ITE Estimation, Deep Learning, Information Theory, Doubly Robust

1 Introduction

Understanding causal relationships and evaluating effects of interventions to achieve desired outcomes is key to progress in many fields, e.g., psychology, public health. A typical scenario in medicine is to determine whether a treatment (e.g., a lipid-lowering medication) is effective to reduce the risk of or cure an illness (e.g., cardiovascular disease). Randomized controlled trials (RCTs) are considered the best practice for evaluating causal effects [1]. However,

RCTs are not always feasible, e.g., due to ethical constraints. For instance, if one wanted to evaluate whether college education is the cause of good salary, it would not be ethical to randomly pick teenagers and randomize their admission to college. So, in many real-world use cases, observational data, i.e., real-world data collected retrospectively and not randomized, are the only usable data source. Unfortunately, observational data are often plagued with various biases –because the data generation processes are largely unknown– such as confounding (i.e., spurious causal effects on outcomes by features that are correlated with a true unmeasured cause) and colliders (i.e., mistakenly including effects of an outcome as predictors), making it difficult to infer causal claims [2]. Another problem is that, in both RCTs and observational datasets, only factual outcomes are available, since clearly an individual cannot be treated and not-treated at the same time. Counterfactuals are alternative predictions that respond to the question *"what outcome would have been observed if a person had been given a different treatment?"* If models are biased, counterfactual predictions can be wrong, and interventions can be ineffective or harmful [3].

Traditional statistical approaches for estimating treatment effects, taking into account possible bias from pre-treatment characteristics, include propensity score matching (PSM) and inverse probability weighting (IPW) [4]. The propensity score is a scalar estimate representing the conditional probability of receiving the treatment, given a set of measured pre-treatment covariates. By matching (or weighting) treated and control subjects according to their propensity score, a balance in pre-treatment covariates is induced, mimicking a randomization of the treatment assignment. However, the PSM approach only accounts for measured covariates, and latent bias may remain after matching [5]. PSM has been historically implemented with logistic-linear regression, coupled with different feature selection methods in the presence of high-dimensional datasets [6]. A problem with PSM is that it often decreases the sample size due to matching, while IPW can be affected by skewed, heavy-tailed weight distributions. Machine learning approaches have been introduced more recently, e.g., Bayesian additive regression trees [7] and counterfactual random forests [8]. Big data also led to the flourishing of causal deep learning [9]. Notable examples include the Treatment-Agnostic Representation Network (TARNet) [10], Dragonnet [11], Deep Counterfactual Network with Propensity-Dropout (DCN-PD) [12], Generative Adversarial Nets for inference of Individualized Treatment Effects (GANITE) [13], Causal Effect Variational Autoencoder (CEVAE) [14], and Treatment Effect by Disentangled Variational AutoEncoder (TEDVAE) [15].

1.1 Contribution

This work introduces a novel deep learning approach for treatment effect estimation and counterfactual prediction, named the *Doubly Robust Variational Information-theoretic Deep Adversarial Learning* (DR-VIDAL). Motivated from Makhzani *et al.* [16], we used a lower-dimensional neural representation of the input covariates to generate counterfactuals to improve convergence on

high-dimensional spaces. We assumed a causal graph on top of the covariates where the covariates X are generated from 4 independent latent variables Z_t, Z_{ycf}, Z_{yf} and Z_x indicating latents for treatment, counterfactual, factual outcomes and observed covariates respectively, shown in Figure 1. When generating the representations, we used a variational autoencoder (VAE) to infer the latent variables from the covariates in unsupervised manner and fed the learned lower dimensional representation from the VAE to a generative adversarial network (GAN). Also, to counter the loss of the predictive information while generating the counterfactuals, we aimed to maximize the mutual information between the learned representations and the output of the generator. We added this as a regularizer to the generator loss to generate more robust counterfactuals. Finally, we incorporated a doubly robust network head to estimate the ITE, improving also in loss convergence. As DR-VIDAL generates the counterfactual outcomes, we minimised the supervised loss for both the factual and the counterfactual outcomes to estimate ITE more accurately.

The main features of DR-VIDAL are, in summary:

- Incorporation of an underlying causal structure where the observed pre-treatment covariate set X is decomposed into four independent latent variables $Z_t, Z_X, Z_{yf}, Z_{ycf}$, inducing confounding on both the treatment and the outcome (Figure 1).
- Latent variables are inferred using a VAE [17].
- A GAN [18] with variational information maximization [19] generates (synthetic) complete tuples of covariates, treatment, factual and counterfactual outcomes.
- Individual treatment effects are estimated on complete datasets with a downstream, four-headed deep learning block which is doubly robust [20, 21].

To our knowledge, this is the first time in which VAE, GAN, information theory and doubly robustness are amalgamated into a counterfactual prediction method. By performing test runs on synthetic and real-world datasets, we show that DR-VIDAL can outperform a number of state-of-art tools for estimating ITE.

2 Problem Formulation

We use the *potential outcomes* framework [22, 23]. Let us consider a treatment t (binary for ease of reading, but the theory can be extended to multiple treatments) that can be prescribed to a population sample of size N. The individuals are characterized by a set of pre-treatment background covariates $\boldsymbol{X}$, and a health outcome Y is measured after treatment. We define each subject i with the tuple $\{\boldsymbol{X}, T, Y\}_{i=1}^{N}$, where Y_i^0 and Y_i^1 are the potential outcomes when applying treatments $T_i = 0$ and $T_i = 1$, respectively. The ITE $\tau(\mathbf{x})$ for subject i with pre-treatment covariates $\boldsymbol{X}_i = \mathbf{x}$, is defined as the difference in the average potential outcomes under both treatment interventions (i.e., treated vs. not treated), conditional on $\mathbf{x}$, i.e.,

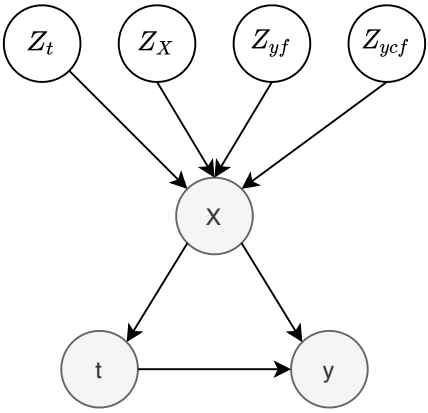

Fig. 1 Directed acyclic graph modeling the causal relationships among a treatment t, outcome y and pre-treatment covariates X, under a latent space Z.

$$\tau(\mathbf{x}) = \mathbb{E}[Y_i^1 - Y_i^0 \mid \boldsymbol{X}_i = \mathbf{x}] \tag{1}$$

The ITE cannot be calculated directly give the inaccessibility of both potential outcomes, as only factual outcomes can be observed, while the others (counterfactuals) can be considered as missing values. However, when the potential outcomes are made independent of the treatment assignment, conditionally on the pre-treatment covariates, i.e., $\{Y^1, Y^0\} \perp T \mid \boldsymbol{X}$, the ITE can then be estimated as $\tau(\mathbf{x}) = \mathbb{E}[Y^1 \mid T = 1, \boldsymbol{X} = \mathbf{x}] - \mathbb{E}[Y^0 \mid T = 0, \boldsymbol{X} = \mathbf{x}] = \mathbb{E}[Y \mid T = 1, \boldsymbol{X} = \mathbf{x}] - \mathbb{E}[Y \mid T = 0, \boldsymbol{X} = \mathbf{x}]$.

Such an assumption is called the strongly ignorable treatment assignment (SITA) assumption [24, 25]. By further averaging over the distribution of $\boldsymbol{X}$, the average treatment effect (ATE) τ_{01} can be calculated as

$$\tau_{01} = \mathbb{E}[\tau(\boldsymbol{X})] = \mathbb{E}[Y \mid T = 1] - \mathbb{E}[Y \mid T = 0] \tag{2}$$

ITE and ATE can be calculated with stratification matching of $\mathbf{x}$ in treatment and control groups, but the calculation becomes unfeasible as the covariate space increases in dimensions.

The propensity score $\pi(x)$ represents the probability of receiving the treatment $T = 1$ conditioned on the pre-treatment covariates $X = x$, denoted as $\pi(\mathbf{x}) = P(T = 1 \mid \boldsymbol{X} = \mathbf{x})$ [23].

The propensity score can be calculated using a regression function, e.g., logistic. ITE/ATE can then be calculated by matching (PSM) or weighting (IPW) instances through $\pi(\mathbf{x})$, in a doubly robust way [26], or through myraid approaches [8, 26–32]. In the next section, we describe approaches based on deep learning.

Algorithm 1 Training of the generative adversarial network for counterfactual outcome calculation

Input: Training set $\boldsymbol{X} = \{(\mathbf{x}^{(1)}, t^{(1)}, y_f^{(1)}),..., (\mathbf{x}^{(n)}, t^{(n)}, y_f^{(n)})\}$; hyper-parameters $\gamma > 0$; $\lambda > 0$; Encoders: E_{ϕ_x}, E_{ϕ_t}, $E_{\phi_{yf}}$, $E_{\phi_{ycf}}$ with parameters $\phi_x, \phi_t, \phi_{yf}, \phi_{ycf}$ respectively; Decoder D_{ϕ_d} with parameter D_{ϕ_d}; Generator G_{θ_g}, Discriminator D_{θ_d}, Q network D_{θ_q} with parameters $\theta_g, \theta_d, \theta_q$ respectively

Initialize parameters: $\phi_x, \phi_t, \phi_{yf}, \phi_{ycf}, \phi_d, \theta_g, \theta_d, \theta_q$
while training **do**
  $\mathbf{x} \leftarrow$ batch of samples from the dataset
  $\mathbf{z}_{\mu_x}, \mathbf{z}_{\sigma_x} \leftarrow E_{\phi_x}(\mathbf{x})$
  $\mathbf{z}_{\mu_t}, \mathbf{z}_{\sigma_t} \leftarrow E_{\phi_t}(\mathbf{x})$
  $\mathbf{z}_{\mu_{yf}}, \mathbf{z}_{\sigma_{yf}} \leftarrow E_{\phi_{yf}}(\mathbf{x})$
  $\mathbf{z}_{\mu_{ycf}}, \mathbf{z}_{\sigma_{ycf}} \leftarrow E_{\phi_{ycf}}(\mathbf{x})$
  $\mathbf{z}_x \leftarrow \mathbf{z}_{\mu_x} + \epsilon \mathbf{z}_{\sigma_x}$, where $\epsilon \sim \mathcal{N}(0, Id)$
  $\mathbf{z}_t \leftarrow \mathbf{z}_{\mu_t} + \epsilon \mathbf{z}_{\sigma_t}$, where $\epsilon \sim \mathcal{N}(0, Id)$
  $\mathbf{z}_{yf} \leftarrow \mathbf{z}_{\mu_{yf}} + \epsilon \mathbf{z}_{\sigma_{yf}}$, where $\epsilon \sim \mathcal{N}(0, Id)$
  $\mathbf{z}_{ycf} \leftarrow \mathbf{z}_{\mu_{ycf}} + \epsilon \mathbf{z}_{\sigma_{ycf}}$, where $\epsilon \sim \mathcal{N}(0, Id)$
  Concatenate $\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf}$ to form $\mathbf{z}_c$
  $\hat{\mathbf{x}} \leftarrow D_{\phi_d}(\mathbf{z}_c)$
  Calculate $\mathcal{L}_{VAE}(\phi_x, \phi_t, \phi_{yf}, \phi_{ycf};\ \mathbf{x}, \mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf})$
  $\phi_x \overleftarrow{-} \nabla_{\phi_x}\mathcal{L}_{VAE}$; $\phi_t \overleftarrow{-} \nabla_{\phi_t}\mathcal{L}_{VAE}$; $\phi_{yf} \overleftarrow{-} \nabla_{\phi_{yf}}\mathcal{L}_{VAE}$; $\phi_{ycf} \overleftarrow{-} \nabla_{\phi_{ycf}}\mathcal{L}_{VAE}$; $\phi_d \overleftarrow{-} \nabla_{\phi_d}\mathcal{L}_{VAE}$
  $\mathbf{z}_G \sim \mathcal{N}(0, Id)$
  $y_0, y_1 \leftarrow G_{\theta_g}(\mathbf{z}_G, \mathbf{z}_c)$
  $\hat{y}_0 = ((1-t) * y_f + t * y_0)$; $\hat{y}_1 = (t * y_f + (1-t) * y_1)$
  $d_{logit} \leftarrow D_{\theta_d}(\mathbf{x}, \hat{y}_0, \hat{y}_1)$
  Calculate $\mathcal{L}^D(\theta_d)$
  $\theta_d \overleftarrow{-} \nabla_{\theta_d}\mathcal{L}^D(\theta_d)$
  $\hat{y}_f \leftarrow t * y_1 + (1-t) * y_0$
  Compute$\mathcal{L}_S^G(y_f, \hat{y}_f)$
  Concatenate y_0, y_1 to form q_{input}
  $q_\mu, q_\sigma \leftarrow Q_{\theta_q}(q_{input})$
  Compute $\mathcal{L}_I(G, Q)$ by treating $Q(c|x)$ as factored Gaussian using q_μ, q_σ and z_c
  Compute $\mathcal{L}^G(\theta_g)$
  $\theta_g \overleftarrow{-} \nabla_{\theta_g}\mathcal{L}^G(\theta_g)$
end while

3 Related Work

Alaa and Van der Schaar [33] developed a comprehensive work on characterizing the conditions and the limits of treatment effect estimation using deep learning. The sample size plays an important role, e.g., estimations on small sample sizes are affected by selection bias, while on large sample sizes, they

Algorithm 2 Training of the doubly robust multitask network for ITE estimation

Input: Complete dataset $\tilde{X} = \{(\mathbf{x}^{(1)}, t^{(1)}, y_f^{(1)}, y_{cf}^{(1)}),..., (\mathbf{x}^{(n)}, t^{(n)}, y_f^{(n)}, y_{cf}^{(n)}) \}$ after training the GAN module for counterfactual prediction; hyper-parameters $\alpha > 0$; $\beta > 0$; outcome heads with shared parameters ϕ and outcome specific parameters θ_0, θ_1; propensity head with parameters θ_π; regressor head with parameters θ_μ

Initialize parameters: $\theta_0, \theta_1, \theta_\pi, \theta_\mu$
while training **do**
  $\mathbf{x} \leftarrow$ batch of samples from the dataset
  Calculate $\hat{y}_i^{(0)}, \hat{y}_i^{(1)}, \hat{y}_f^{(i)}, \hat{y}_{cf}^{(i)}$
  Calculate the predicted loss $\mathcal{L}_i^p(\theta_1, \theta_0, \phi)$
  Calculate $\hat{y}_{fDR}^{(i)}, \hat{y}_{cfDR}^{(i)}$
  Calculate the doubly Robust loss $\mathcal{L}_i^{DR}(\theta_1, \theta_0, \theta_\pi, \theta_\mu, \phi)$
  Calculate the final loss $\mathcal{L}_{ITE}(\theta_1, \theta_0, \theta_\pi, \theta_\mu, \phi)$
  Calculate gradients of the loss $\mathcal{L}_{ITE}(\theta_1, \theta_0, \theta_\pi, \theta_\mu, \phi)$
  Update the parameters $\theta_1, \theta_0, \theta_\pi, \theta_\mu, \phi$
end while

are affected by algorithmic design. Our work builds up on the ITE estimation approaches of CEVAE [14], DCN-PD [12], Dragonnet [11], GANITE [13], TARNet [10], and TEDVAE [15]. DCN-PD is a doubly robust, multitask network for counterfactual prediction, where propensity scores are used to determine a dropout probability of samples to regularize training, carried out in alternating phase, using treated and control batches. CEVAE uses VAE to identify latent variables from an observed pre-treatment vector and to generate counterfactuals. TARNet aims to provide an upper bound effect estimation by balancing the distributions of treated and controls –with a weight indemnifying group imbalance– within a high dimensional covariate space, but it does not exploit counterfactuals, and only minimises the factual loss function. Dragonnet is a modified TARNet with targeted regularization based on propensity scores. GANITE generates proxies of counterfactual outcomes from covariates and random noise using a GAN, and feeds them to an ITE generator. For both GANITE and TARNet, in presence of high-dimensional data, the loss could be hard to converge. TEDVAE [15] uses a variational autoencoder to infer hidden latent variables from proxies using a causal graph similar to CEVAE. In the next sections, we discuss in detail the novelty of DR-VIDAL and the differences in the architectural design and training mechanisms with respect to the aforementioned approaches.

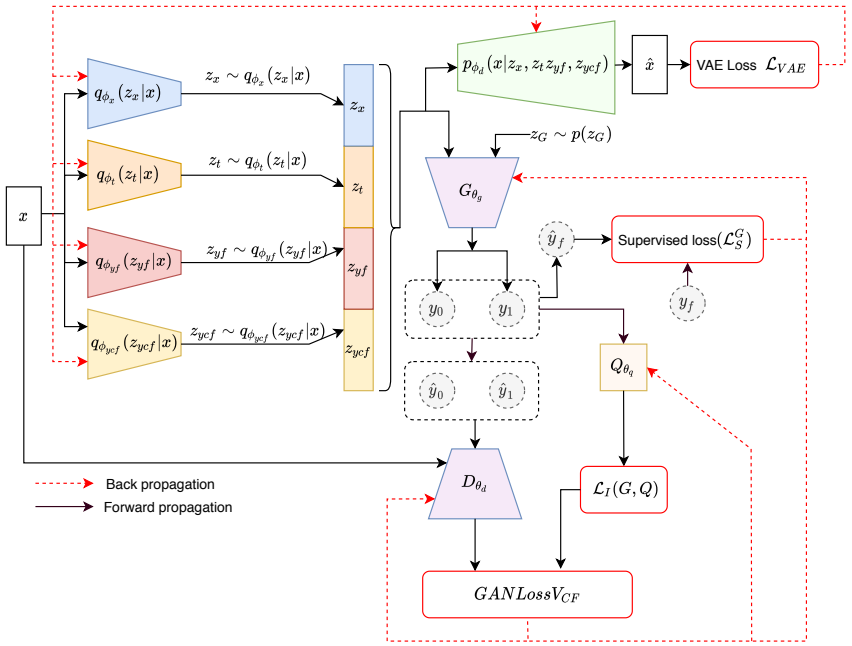

Fig. 2 Architecture of the counterfactual network to estimate the counterfactual outcome.

4 Proposed Methodology

DR-VIDAL architecture can be decomposed into three components: (1) a VAE infering the latent variables, (2) a GAN generating the counterfactual outcomes, and (3) a doubly robust module estimating ITE. The architectural layout and the training algorithm of the first two components (VAE and GAN) are illustrated in Figure 2 and Algorithm 1 , and for the third component (doubly robust estimator), are shown in Figure 3 and Algorithm 2

4.1 Latent variable inference with VAE

We assume that the observed covariates $\mathbf{X} = \mathbf{x}$ with treatment assignment $T = t$ factual and counterfactual outcomes $Y_f = y_f$ and $Y_{cf} = y_{cf}$ respectively, are generated from an independent latent space $\mathbf{z}$, composed by $\mathbf{z}_x \sim p(\mathbf{z}_x)$, $z_t \sim p(\mathbf{z}_t)$, $\mathbf{z}_{yf} \sim p(\mathbf{z}_{yf})$, and $\mathbf{z}_{ycf} \sim p(\mathbf{z}_{ycf})$, which denote the latent variables for the covariates $\mathbf{x}$, treatment indicator t, and factual outcomes y_f and y_{cf}, respectively. This decomposition follows the causal structure shown in Figure 1. The goal is to infer the posterior distribution $p(\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf} \mid \mathbf{x})$, which is harder to optimize. We use the theory of variational inference [34] to learn the variational posteriors $q_{\phi_x}(\mathbf{z}_x \mid \mathbf{x})$, $q_{\phi_t}(\mathbf{z}_t \mid \mathbf{x})$, $q_{\phi_{yf}}(\mathbf{z}_{yf} \mid \mathbf{x}), q_{\phi_{ycf}}(\mathbf{z}_{ycf} \mid \mathbf{x})$, using 4 different neural network encoders with parameters $\phi_x, \phi_t, \phi_{yf}$, and ϕ_{ycf}, respectively. Using the latent factors sampled from the learned variational posteriors, we reconstruct $\mathbf{x}$ by estimating

the likelihood $p_{\phi_d}(\mathbf{x} \mid \mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf})$ via a single decoder parameterized by ϕ_d. The latent factors, assumed to be Gaussian, are defined as follows:

$$p(\mathbf{z}_x) = \prod_{i=1}^{D_{z_x}} \mathcal{N}(z_{x_i} \mid 0, 1); \qquad p(\mathbf{z}_t) = \prod_{i=1}^{D_{z_t}} \mathcal{N}(z_{t_i} \mid 0, 1) \tag{3}$$

$$p(\mathbf{z}_{yf}) = \prod_{i=1}^{D_{z_{yf}}} \mathcal{N}(z_{yf_i} \mid 0, 1); \qquad p(\mathbf{z}_{ycf}) = \prod_{i=1}^{D_{z_{ycf}}} \mathcal{N}(z_{ycf_i} \mid 0, 1) \tag{4}$$

where $D_{z_x}, D_{z_t}, D_{z_{yf}}, D_{z_{ycf}}$ are the dimensions of the latent factors $\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf}$, respectively. The variational posteriors of the inference of models are defined as:

$$q_{\phi_x}(\mathbf{z}_x \mid \mathbf{x}) = \prod_{i=1}^{D_{z_x}} \mathcal{N}(\mu = \hat{\mu}_x, \sigma^2 = \hat{\sigma}_x^2) \tag{5}$$

$$q_{\phi_t}(\mathbf{z}_t \mid \mathbf{x}) = \prod_{i=1}^{D_{z_t}} \mathcal{N}(\mu = \hat{\mu}_t, \sigma^2 = \hat{\sigma}_t^2) \tag{6}$$

$$q_{\phi_{yf}}(\mathbf{z}_{yf} \mid \mathbf{x}) = \prod_{i=1}^{D_{z_{yf}}} \mathcal{N}(\mu = \hat{\mu}_{yf}, \sigma^2 = \hat{\sigma}_{yf}^2) \tag{7}$$

$$q_{\phi_{ycf}}(\mathbf{z}_{ycf} \mid \mathbf{x}) = \prod_{i=1}^{D_{z_{ycf}}} \mathcal{N}(\mu = \hat{\mu}_{ycf}, \sigma^2 = \hat{\sigma}_{ycf}^2) \tag{8}$$

where $\hat{\mu}_x, \hat{\mu}_t, \hat{\mu}_{yf}, \hat{\mu}_{ycf}$ and $\hat{\sigma}_x^2, \hat{\sigma}_t^2$, $\hat{\sigma}_{yf}^2$, $\hat{\sigma}_{ycf}^2$ are the means and variances of the Gaussian distributions parameterized by encoders $E_{\phi_x}, E_{\phi_t}, E_{\phi_{yf}}, E_{\phi_{ycf}}$ with parameters $\phi_x, \phi_t, \phi_{yf}, \phi_{ycf}$ respectively.

The overall evidence lower bound (ELBO) loss of the VAE is expressed as $\mathcal{L}_{ELBO}$ in the following equation,

$$\begin{aligned}
&\mathcal{L}_{ELBO}(\phi_x, \phi_t, \phi_{yf}, \phi_{ycf};\ \mathbf{x}, \mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf}) \\
&= \mathbb{E}_{q_{\phi_x}, q_{\phi_t}, q_{\phi_{yf}}, q_{\phi_{ycf}}}[\log p_{\phi_d}(\mathbf{x} \mid \mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf})] \\
&- KL\big(q_{\phi_x}(\mathbf{z}_x \mid \mathbf{x}) \mid\mid p_{\phi_d}(\mathbf{z}_x))\big) - KL\big(q_{\phi_t}(\mathbf{z}_t \mid \mathbf{x}) \mid\mid p_{\phi_d}(\mathbf{z}_t))\big) \\
&- KL\big(q_{\phi_{yf}}(\mathbf{z}_{yf} \mid \mathbf{x}) \mid\mid p_{\phi_d}(\mathbf{z}_{yf}))\big) - KL\big(q_{\phi_{ycf}}(\mathbf{z}_{ycf} \mid \mathbf{x}) \mid\mid p_{\phi_d}(\mathbf{z}_{ycf}))\big)
\end{aligned}$$

where KL denotes the Kullback–Leibler divergence of two probability distributions. We minimize the optimization function of the VAE as $\mathcal{L}_{VAE}$ to obtain the optimal parameter of the encoders $\phi_x, \phi_t, \phi_{yf}, \phi_{ycf}$, and of the decoder ϕ_d as $\mathcal{L}_{VAE}(\phi_x, \phi_t, \phi_{yf}, \phi_{ycf};\ \mathbf{x}, \mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf}) = -\mathcal{L}_{ELBO}(\phi_x, \phi_t, \phi_{yf}, \phi_{ycf};\ \mathbf{x}, \mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf})$. The detailed derivation is included in the Supplementary Information section A.

4.2 Generation of counterfactuals via GAN

After learning the hidden latent codes $\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf}$ from the VAE, we concatenate the latent codes to form z_c, passed to the generator of the GAN block G_{θ_g}, along with a random noise $\mathbf{z}_G \sim \mathcal{N}(0, Id)$. G_{θ_g} is parameterized by θ_g, and it outputs the vector $\overline{y}$ of the potential (factual and counterfactual) outcomes. We replace the factual outcome y_f in the generated outcome vector $\overline{y}$ to form $\hat{y}_0$ and $\hat{y}_1$, which are passed to the counterfactual discriminator D_{θ_d}, along with the true covariate vector $\mathbf{x}$. D_{θ_d} is parameterized by θ_d, and is responsible to predict the treatment variable, similarly to GANITE. The loss of the GAN block is defined as:

$$V_{GAN}(G, D) = \mathbb{E}_{\mathbf{x}, \mathbf{z}_G, \mathbf{z}_c}\left[t^T \log D(\mathbf{x}, G(\mathbf{z}_G, \mathbf{z}_c)) + (1-t)^T \log(1 - D(\mathbf{x}, G(\mathbf{z}_G, \mathbf{z}_c))\right]$$

where $\mathbf{x} \sim p(\mathbf{x}), \mathbf{z}_G \sim p(\mathbf{z}_G)$ and $\mathbf{z}_c$ denote the concatenated latent codes $\mathbf{z}_x \sim q_{\phi_x}(\mathbf{z}_x \mid \mathbf{x})$, $\mathbf{z}_t \sim q_{\phi_t}(\mathbf{z}_t \mid \mathbf{x})$, $\mathbf{z}_{yf} \sim q_{\phi_{yf}}(\mathbf{z}_{yf} \mid \mathbf{x})$ and $\mathbf{z}_{ycf} \sim q_{\phi_{ycf}}(\mathbf{z}_{ycf} \mid \mathbf{x})$. From $\overline{y}$, we also calculate the predicted factual outcome $\hat{y}_f$. As also done in GANITE, we make sure to include the supervised loss $\mathcal{L}_S^G(y_f, \hat{y}_f)$, which enforces the predicted factual outcome $\hat{y}_f$ to be as close as to the true factual outcome y_f.

$$\mathcal{L}_S^G(y_f, \hat{y}_f) = \frac{1}{n} \sum_{i=1}^{n} \left(y_f(i) - \hat{y}_f(i)\right)^2 \tag{9}$$

The complete loss function of counterfactual GAN is given by $V_{CF}(G, D) = V_{GAN}(G, D) + \gamma \mathcal{L}_S^G(y_f, \hat{y}_f)$.

We also employ an additional regularization $\lambda I(\mathbf{z}_c;\ G(\mathbf{z}_G, \mathbf{z}_c))$ to maximize the mutual information between the learned concatenated latent code $\mathbf{z}_c$ and the generated output by the generator $G(\mathbf{z}_G, \mathbf{z}_c)$, as in [19].We thus propose to solve the following minimax game:

$$\min_G \max_D V_{CF\_I}(G, D) = V_{CF}(G, D) + \lambda I(\mathbf{z}_c;\ G(\mathbf{z}_G, \mathbf{z}_c)) \tag{10}$$

$I(\mathbf{z}_c;\ G(\mathbf{z}_G, \mathbf{z}_c))$ is harder to solve because of the presence of the posterior $p(\mathbf{z}_c \mid \mathbf{x})$ [19], so we obtain the lower bound of it using an auxiliary distribution $Q(\mathbf{z}_c \mid \mathbf{x})$ to approximate $p(\mathbf{z}_c \mid \mathbf{x})$.

Finally, the optimization function of the counterfactual information-theoretic GAN –*InfoGAN*– incorporating the variational regularization of mutual information and hyperparameter λ is given by:

$$\min_{G,Q} \max_D V_{CF\_infoGAN}(G, D, Q) = V_{CF}(G, D) - \lambda \mathcal{L}_I(G, Q) \tag{11}$$

The counterfactual InfoGAN is used to generate the missing counterfactual outcome y_{cf} to form the quadruple $\{\mathbf{x},\ t,\ y_f,\ y_{cf}\}_{i=1}^N$ and sent to the doubly robust block to estimate the ITE.

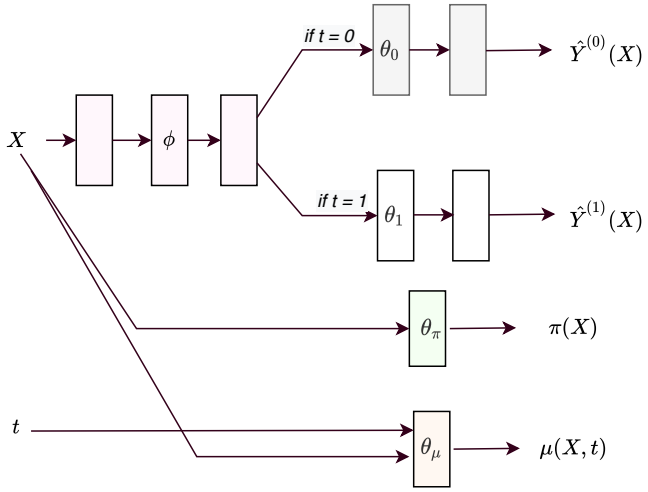

Fig. 3 Architecture of the four-headed, doubly robust neural network to calculate individual treatment effects.

4.3 Information-theoretic GAN optimization

The GAN generator G_{θ_g} works to fool the discriminator D_{θ_d}. To get the optimal Discriminator $D^*_{\theta_d}$, we maximize $V_{CF\_infoGAN}$

$$\max_D \mathcal{L}^D(\theta_d) = V_{CF\_infoGAN}(G, D, Q) \tag{12}$$

To get the optimal generator $G^*_{\theta_g}$, we maximize $V_{CF\_infoGAN}$

$$\min_{G,Q} \mathcal{L}^G(\theta_g) = V_{CF\_infoGAN}(G, D, Q) \tag{13}$$

4.4 Doubly robust ITE estimation

As introduced above, the propensity score $\pi(\mathbf{x})$ represents the probability of receiving a treatment $T = 1$ (over the alternative $T = 0$) conditioned on the pre-treatment covariates $X = x$. By combining IPW through $\pi(\mathbf{x})$ with outcome regression by both treatment variable and the covariates, Jonsson defined the doubly robust estimation of causal effect [20] as follows:

$$\hat{\delta}_{DR} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{y_i t_i - (t_i - \pi(x_i))\mu(x_i, t_i)}{\pi(x_i)} - \frac{y_i(1-t_i) - (t_i - \pi(x_i))\mu(x_i, t_i)}{1 - \pi(x_i)}\right] \tag{14}$$

where $\mu(x,t) = \hat{\alpha_0} + \hat{\alpha_1}x_1 + \hat{\alpha_2}x_2 + \cdots + \hat{\alpha_n}x_n + \hat{\delta}t$, and $(t_i - \pi(x_i))\mu(x_i, t_i)$ is used for the IPW estimator.

After getting the counterfactual outcome y_{cf} from the GAN to form the quadruple $\{\mathbf{x},\, t,\, y_f,\, y_{cf}\}_{i=1}^{N}$, we pass this as the input to the doubly robust network to estimate the ITE, using the architecture shown in Figure 3. To predict the outcomes $y^{(0)}$ and $y^{(1)}$, we use a configuration similar to TARNet, which contains a number of shared layers, denoted by f_ϕ, parameterized by ϕ, and two outcome-specific heads f_{θ_0} and f_{θ_1}, parameterized by θ_0 and θ_1.

To ensure doubly robustness, we introduce two more heads that predict the propensity score $\pi(\mathbf{x}) = \mathbb{P}(T = 1 \mid \mathbf{x})$ and the regressor $\mu(\mathbf{x}, t)$. These two are calculated using two neural networks, parameterized by θ_π and θ_μ respectively. The factual and counterfactual outcome $y_i^{(0)}$ and $y_i^{(1)}$ of the i^{th} sample are then calculated as:

$$\hat{y}_f^{(i)} = t_i(f_{\theta_1}(f_\phi(\mathbf{x}_i))) + (1 - t_i)(f_{\theta_0}(f_\phi(\mathbf{x}_i))) \tag{15}$$

$$\hat{y}_{cf}^{(i)} = (1 - t_i)(f_{\theta_1}(f_\phi(\mathbf{x}_i))) + t_i(f_{\theta_0}(f_\phi(\mathbf{x}_i))) \tag{16}$$

Next, the predicted loss will be

$$\mathcal{L}_i^p(\theta_1, \theta_0, \phi) = (\hat{y}_f^{(i)} - y_f^{(i)})^2 + (\hat{y}_{cf}^{(i)} - y_{cf}^{(i)})^2 + \alpha \text{BinaryCrossEntropy}(\pi(x_i), t_i)$$

where α is a hyperparameter. With the help of the propensity score $\pi(\mathbf{x})$ and the regressor $\mu(\mathbf{x}, T)$, the doubly robust outcomes are calculated as

$$\hat{y}_{f_{DR}}^{(i)} = t_i \left[\frac{t_i \hat{y}_i^{(1)} - (t_i - \pi(\mathbf{x}_i)\mu(\mathbf{x}_i, t_i))}{\pi(\mathbf{x}_i)} \right] \tag{17}$$

$$+ (1 - t_i) \left[\frac{(1 - t_i)\hat{y}_i^{(0)} - (t_i - \pi(\mathbf{x}_i)\mu(\mathbf{x}_i, t_i))}{1 - \pi(\mathbf{x}_i)} \right] \tag{18}$$

$$\hat{y}_{cf_{DR}}^{(i)} = (1 - t_i) \left[\frac{(1 - t_i)\hat{y}_i^{(1)} - (t_i - \pi(\mathbf{x}_i)\mu(\mathbf{x}_i, t_i))}{\pi(\mathbf{x}_i)} \right] \tag{19}$$

$$+ t_i \left[\frac{t_i \hat{y}_i^{(0)} - (t_i - \pi(\mathbf{x}_i)\mu(\mathbf{x}_i, t_i))}{1 - \pi(\mathbf{x}_i)} \right]$$

The doubly robust loss $\mathcal{L}_i^{DR}(\theta_1, \theta_0, \theta_\phi, \theta_\mu, \phi)$ is calculated as:

$$\mathcal{L}_i^{DR}(\theta_1, \theta_0, \theta_\pi, \theta_\mu, \phi) = (\hat{y}_{f_{DR}}^{(i)} - y_f^{(i)})^2 + (\hat{y}_{cf_{DR}}^{(i)} - y_{cf}^{(i)})^2 \tag{20}$$

Finally, the loss function of the ITE is:

$$\mathcal{L}^{ITE}(\theta_1, \theta_0, \theta_\pi, \theta_\mu, \phi) = \frac{1}{n} \sum_{i=1}^{n} \left(\mathcal{L}_i^p + \beta \mathcal{L}_i^{DR} \right) \tag{21}$$

where β is a hyperparameter and the whole network is trained using end-to-end strategy.

4.5 Differences with CEVAE, TEDVAE and GANITE

The counterfactual outcome predictor of DR-VIDAL uses both VAE and GAN in the same framework, while only VAE is used in CEVAE and only GAN is used GANITE. CEVAE also incorporates a causal graph, but it is simpler than ours, as it infers only the observed proxy X from Z. We instead considered multiple latent variables causally related to the treatment and the outcome in addition to the direct links to the pre-treatment covariates. TEDVAE's causal graph is more similar to ours, but our approach first generates the counterfactual outcomes and then minimises the supervised loss for both the factual and counterfactual outcomes using doubly robustness (for better estimation of ITE). Furthermore, we use GAN to generate counterfactual examples, but, unlike GANITE, we first infer the multiple latent factors using a VAE, then optimize the GAN with the mutual information, and finally generate the entire potential outcome vector.

4.6 Differences with TARNet and Dragonnet

The design of the doubly robust module block of DR-VIDAL is closely related to that of TARNet and Dragonnet. However, TARNet uses a two-headed network, which is not doubly robust. Dragonnet includes a third head that incorporates the propensity score. DR-VIDAL has four heads, and exploits the doubly robustness adding two heads, i.e., the propensity score and the regressor head, to the the basic two-headed TARNet configuration. Further, TARNet's sample weights are calculated as the crude probability of the treatment assignment, whereas DR-VIDAL accounts for the pre-treatment covariates. For Dragonnet, the targeted regularization is implemented without taking into account the regressed outcome, which instead is estimated by DR-VIDAL in the fourth head, as a function of treatment and pre-treatment covariates. Another major difference between TARNet/Dragonnet and DR-VIDAL is the training strategy. For both TARNet and Dragonnet, the counterfactual outcome does not exist, so for each sample the overall loss function has to be estimated with the factual outcome only. In contrast DR-VIDAL updates and optimizes parameters on the entire potential outcomes vector.

5 Experimental Setup

Synthetic datasets.

We conduct performance tests on two synthetic data experiments. The first uses the same data generation process devised for CEVAE [14]. We generate a marginal distribution $\mathbf{x}$ as a mixture of Gaussians from the 5-dimensional latent variable $\mathbf{z}$, indicating each mixture component as,

$$\begin{aligned} \mathbf{z}_i \sim \text{Bern}(0.5); \qquad \mathbf{x}_i \mid \mathbf{z}_i \sim \mathcal{N}(\mathbf{z}_i, \sigma_5^2 \mathbf{z}_i + \sigma_3^2(1-\mathbf{z}_i)) \\ t_i \mid \mathbf{z}_i \sim \text{Bern}(0.75\mathbf{z}_i + 0.25(1-\mathbf{z}_i)) \\ \mathbf{y}_i \mid t_i, \mathbf{z}_i \sim \text{Bern}(\text{Sigmoid}(3(\mathbf{z}_i + 2(2t_i - 1)))) \end{aligned} \tag{22}$$

Datasets of sample size {1000, 3000, 5000, 10000, 30000} are generated, and divided into 80-20 % train-test split.

In the second experimental setting, we amalgamate the synthetic data generation process by CEVAE with that of GANITE [13], to model the more complex causal structure illustrated in Figure 1. We sample 7-, 1-, 1-, and 1-dimensional vectors for $\mathbf{z}_x$, $\mathbf{z}_t$, $\mathbf{z}_{yf}$, and $\mathbf{z}_{ycf}$ from Bernoulli distributions, and then collate them into x, as specified below,

$$\begin{aligned} \mathbf{z}_x \sim \text{Bern}(0.5); \qquad \mathbf{z}_t \sim \text{Bern}(0.5) \\ \mathbf{z}_{yf} \sim \text{Bern}(0.5); \qquad \mathbf{z}_{ycf} \sim \text{Bern}(0.5) \\ \mathbf{x}_x \mid \mathbf{z}_x \sim \mathcal{N}(\mathbf{z}_x, 5(\mathbf{z}_x) + 3(1-\mathbf{z}_x)) \\ \mathbf{x}_t \mid \mathbf{z}_t \sim \mathcal{N}(\mathbf{z}_x, 2(\mathbf{z}z_t) + 0.5(1-\mathbf{z}_t)) \\ \mathbf{x}_{yf} \mid \mathbf{z}_{yf} \sim \mathcal{N}(\mathbf{z}_{yf}, 10(\mathbf{z}_{yf}) + 6(1-\mathbf{z}_{yf})) \\ \mathbf{x}_{ycf} \mid \mathbf{z}_{ycf} \sim \mathcal{N}(\mathbf{z}_{ycf}, 10(\mathbf{z}_{ycf}) + 6(1-\mathbf{z}_{ycf})) \\ \mathbf{w}_t^T \sim \mathcal{U}((-0.1, 0.1)^{10\text{x}1}); \qquad \mathbf{n}_t \sim \mathcal{N}(0, 0.1) \\ \mathbf{w}_y^T \sim \mathcal{U}((-1, 1)^{10\text{x}2}); \qquad \mathbf{n}_y \sim \mathcal{N}(0^{2\text{x}1}, 0.1\text{x}\mathcal{I}^{2\text{x}2}) \\ t \mid x \sim \text{Bern}(\text{Sigmoid}(\mathbf{w}_t^T \mathbf{x} + \mathbf{n}_t)); \qquad \mathbf{y} \mid \mathbf{x} \sim \mathbf{w}_y^T \mathbf{x} + \mathbf{n}_y \end{aligned} \tag{23}$$

From the covariates x, we simulate the treatment assignment t and the potential outcomes y as described in the GANITE paper. We generate multiple synthetic datasets for sample sizes {1000, 3000, 5000, 10000, 30000}, also divided into 80-20 % splits.

Real-world datasets.

We use three popular real-world benchmark datasets: the Infant Health and Development Program (IHDP) dataset [7], the Twins dataset [35], and the Jobs dataset [36]. The IHDP and Twins two are semi-synthetic, and simulated counterfactuals to the real factual data are available. These datasets have been also designed and collated to meet specific treatment overlap condition, nonparallel treatment assignment, and nonlinear outcome surfaces [7, 10, 13,

14]. The IHDP datasets is composed by 110 treated subjects and 487 controls, with 25 covariates. The Twins dataset comprises 4553 treated, 4567 controls, with 30 covariates. The Jobs dataset comprises 237 treated, 2333 controls, with 17 covariates. For all the real-world datasets, we use the same experimental settings described in GANITE, where the datasets are divided into 56/24/20 % train-validation-test splits. We run 1000, 10 and 100 realizations of IHDP, Jobs and Twins datasets, respectively.

Model fit and test details.

Consistent with prior studies [7, 10, 13], we report the error on the ATE ϵ_{ATE}, and the expected Precision in Estimation of Heterogeneous Effect (PEHE), ϵ_{PEHE}, for IHDP and Twins datasets, since factual and the counterfactual outcomes are available. For the Jobs dataset, as the counterfactual outcome does not exist, we report the policy risk $R_{pol}(\pi)$, and the error on the average treatment effect on the treated (ATT) ϵ_{ATT}, as indicated in [10, 13]. We compared DR-VIDAL with TARNet, CEVAE, and GANITE. In addition, for real-world datasets, we compare: least squares regression with treatment as a covariate (OLS/LR1); separate least squares regression for each treatment (OLS/LR2); balancing linear regression (BLR) [9]; k-nearest neighbor (k-NN) [32]; Bayesian additive regression trees (BART) [27]; random and causal forest (R Forest, C Forest) [8, 37]; balancing neural network (BNN) [9]; counterfactual regression with Wasserstein distance (CFR$_{WASS}$) [10].

5.1 Training and implementation of DR-VIDAL

Adversarial module.

To reduce the model complexity and parameters for the encoder of the VAE, we have a shared neural network connected to 4 other networks for estimating the four posterior distributions $q_{\phi_x}(\mathbf{z}_x \mid \mathbf{x})$, $q_{\phi_t}(\mathbf{z}_t \mid \mathbf{x})$, $q_{\phi_{yf}}(\mathbf{z}_{yf} \mid \mathbf{x})$, $q_{\phi_{ycf}}(\mathbf{z}_{ycf} \mid \mathbf{x})$. The shared neural network has 3 layers, each with 15 nodes. The networks with $q_{\phi_x}(\mathbf{z}_x \mid \mathbf{x})$, $q_{\phi_t}(\mathbf{z}_t \mid \mathbf{x})$, $q_{\phi_{yf}}(\mathbf{z}_{yf} \mid x)$, $q_{\phi_{ycf}}(\mathbf{z}_{ycf} \mid \mathbf{x})$ as outputs have a single layer with 5, 1, 1, 1 nodes, respectively. The decoder is a 4-layer neural network, each with 15 nodes to calculate the data likelihood $p_{\phi_d}(\mathbf{x} \mid \mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf})$. For the GAN, the generator network has 2 shared layers and 2 outcome-specific layers, each with 100 nodes. The discriminator and the network for information maximization (Q network in Figure 2) is a 3-layered neural network, each with 30 nodes and 8 nodes respectively. All the layers of the VAE and GAN use Rectified Linear Unit (ReLU) activation functions and the parameters are updated using the Adam optimizer [38]. The random noise $\mathbf{z}_G$ is sampled from a 92-dimensional standardized Gaussian distribution $\mathcal{N}(0, 1)$. The hyperparameter γ is set as 1 for all datasets, while λ is set as 0.2, 0.01 and 10 for IHDP, Jobs and Twins, respectively. The batch sizes of IHDP, Jobs, and Twins are 64, 64, and 256, respectively. The learning rates of the VAE, generator and discriminator are 1e-3, 1e-4, and 5e-4, respectively.

Doubly robust module.

For the doubly robust module, the shared network f_ϕ and outcome specific networks f_{θ_0} and f_{θ_1} are both 3-layer neural network, each with 200 and 100 nodes. The propensity network π has 2 layers each with 200 nodes. The regressor network μ has 6 layers with 200 nodes and 100 nodes in the first and last 3 layers. All the layers of the VAE and GAN use ReLU activation and the Adam optimizer. The batch sizes are the same as for the adversarial module. We set the learning rate of all the networks as 1e-4 and the hyperparameters α and β are set at 1 for all 3 datasets.

Implementation and availability.

DR-VIDAL is written in Pytorch (https://pytorch.org/) and is available under the MIT license at: https://github.com/Shantanu48114860/DR-VIDAL.

5.2 Performance Metrics

Consistent with prior studies [7, 10, 13], we report the error on the ATE ϵ_{ATE}, and the expected Precision in Estimation of Heterogeneous Effect (PEHE), ϵ_{PEHE}, for IHDP and Twins datasets, since factual and the counterfactual outcomes are available. For the Jobs dataset, as the counterfactual outcome does not exist, we report the policy risk $R_{pol}(\pi)$, and the error on the average treatment effect on the treated (ATT) ϵ_{ATT}, as indicated in [10, 13]. The error for PEHE, ATE, Policy Risk, ATT will be evaluated by estimating $\epsilon_{PEHE}, \epsilon_{ATE}, R_{pol}(\pi), \epsilon_{ATT}$ respectively as follows:

$$\epsilon_{PEHE} = \frac{1}{N}\sum_{n=0}^{N}\Big(\mathbb{E}_{y_j(n)\sim\mu_j(n)}\big[y_1(n) - y_0(n)\big] - \big[\hat{y_1}(n) - \hat{y_0}(n)\big]\Big)^2 \tag{24}$$

$$\epsilon_{ATE} = || \frac{1}{N}\sum_{n=0}^{N}\mathbb{E}_{y(n)\sim\mu(n)}[y(n)] - \frac{1}{N}\sum_{n=0}^{N}\hat{y}(n) \;||_2^2 \tag{25}$$

$$R_{pol}(\pi) = \frac{1}{N}\sum_{n=0}^{N}\Big[1 - \Big(\sum_{i=1}^{k}\Big[\frac{1}{\mid \Pi_i \cap T_i \cap E \mid}\sum_{x(n)\in\Pi_i\cap\ T_i\cap E} y_i(n) \times \frac{\mid \Pi_n \cap E \mid}{\mid E \mid}\Big]\Big)\Big] \tag{26}$$

where $\pi_i = \{\boldsymbol{x(n)} : i = \arg\max \hat{\boldsymbol{y}}$,
$T_i = \boldsymbol{x(n)} : t_i(n) = 1\}$, and E is the randomized sample.

The true average treatment effect on the treated (ATT) and its error ϵ_{ATT} are defined as follows:

$$ATT = \frac{1}{\mid T_1 \cap E \mid}\sum_{x_i\in T_1\cap E} Y_1(x_i) - \frac{1}{\mid T_0 \cap E \mid}\sum_{x_i\in C\cap E} Y_0(x_i) \tag{27}$$

$$\epsilon_{ATT} = \mid ATT - \frac{1}{\mid T_1 \cap E \mid} \sum_{x_i \in T_1 \cap E} \hat{Y}_1(x_i) - \hat{Y}_0(x_i) \mid \tag{28}$$

where T_1, T_0 and E are the subsets corresponding to treated, controlled samples, and randomized controlled trials, respectively.

6 Results

Synthetic datasets.

In the first synthetic dataset, which uses the generative assumptions of CEVAE defined in equation 22, the doubly robust version of DR-VIDAL demonstrates lower ATE error at all sample sizes with respect to all models, as shown in Figure 4-a. When comparing PEHE, DR-VIDAL (both with and without the doubly robust feature) largely outperforms GANITE, as displayed in Figure 4-b. In the second synthetic dataset, generated under the more complex assumptions according to equation 23, DR-VIDAL, both with and without the doubly robust feature, outperforms GANITE in terms of PEHE, as shown in Figure 4-b. It is worth noting the potential of DR-VIDAL to better infer hidden representations in comparison to GANITE irrespective of the presence of the doubly robust module.

Real world datasets.

In all three IHDP, Jobs and Twins datasets, across all realizations, the information-theoretic, doubly robust configuration of DR-VIDAL yields the best results against all other configurations –with/without information-theoretic optimization and with/without doubly robust loss. The doubly robust loss seems to be responsible for most of the improvement. The absolute gain is small, in the order of 1%, but the relative gain with respect to the non-doubly robust setup is significant, as shown in Figure B1 of Supplementary Information, where the doubly robust module always outperforms its non-doubly robust version (from 55-60% in IHDP to over 80% in Twins and Jobs datasets). The bar plots show how many times one model setup is better than the other in terms of error on the factual outcome (y_f). The performance for various configurations of DR-VIDAL is shown in table B1 of Supplementary Information.

Table 1 shows the comparison for the $\sqrt{\epsilon_{PEHE}}$ and R_{Pol} values with the state-of-the-art methods on the three datasets. DR-VIDAL outperforms the other methods on all datasets. On the IHDP and Jobs dataset, DR-VIDAL is the best overall by a larger margin. Instead, performance increment in the Twins dataset is mild. Even if DR-VIDAL has a large number of parameters, the Information (maximization) of hidden factors and the adversarial training make it appropriate for datasets with relatively small sample size like IHDP. It is worth noting that DR-VIDAL converges much faster than CEVAE and GANITE, possibly due to the doubly robustness. Table 2 shows the comparison for ϵ_{ATE} for IHDP and Twins dataset and ϵ_{ATT} for Jobs dataset respectively.

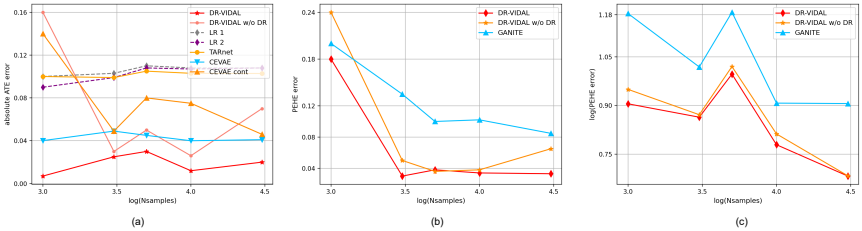

Fig. 4 From left to right a) Comparison of the performance (ATE) of DR-VIDAL vs. all other models on samples from the generative process of CEVAE, defined in equation 22; b-c) Performance comparison (PEHE) of GANITE vs. DR-VIDAL, with or without the doubly robust (DR, w/o DR) feature, on samples from the generative processes defined in equations 22(left panel) and 23 (right panel).

The t-distributed stochastic neighbor embedding (t-SNE) of representations learned by the VAE of the adversarial module of DR-VIDAL for Twins and Jobs datasets –before and after training– are shown in Figure B2 of Supplementary Information. For all datasets, the t-SNE shows reorganization and cluster tightness (i.e., the data reside on a smaller space) on the treatment, factual and counterfactual outcomes spaces.

7 Conclusions and Future Work

DR-VIDAL is a new deep learning approach to causal effect estimation and counterfactual prediction that combines adversarial representation learning, information-theoretic optimization, and doubly robust regression. Our approach fuses several key properties of existing methods with a distinctive causal structure and robust regression design. On all benchmark datasets, DR-VIDAL outperforms other tools, and both the doubly robust property and information-theoretic optimization improve performance over the basic adversarial setup.

This work has some limitations that warrant further development. First, the causal graph, even if more elaborate than CEVAE, is still relatively basic, with straightforward confounding and a unique adjustment set. For instance, a slight modification to the causal graph that connects the Z to X and only to their respective treatment, factual and counterfactual outcome nodes would already imply two adjustments set. Also, the encoded representation in the VAE does not employ any attention mechanism to identify the most important covariates for the propensity scores, especially with of high-dimensional datasets. Finally, one thing that would be worth evaluating is how Dragonnet would perform as a downstream module of DR-VIDAL, substituting it to our current four-head doubly-robust block. Another possible extension may be incorporating the causal graph in TEDVAE with DR-VIDAL and estimating the ITE with the doubly robust setup. Finally, we used variational information maximization, but it could be possible to estimate mutual information directly in the generator to avoid a bottleneck.

In conclusion, DR-VIDAL is a comprehensive framework for predicting counterfactuals and estimating ITE, and its flexibility (modifiable causal structure and modularity) allows for further expansion and improvement.

References

[1] Sibbald, B., Roland, M.: Understanding controlled trials: Why are randomised controlled trials important? BMJ **316**(7126), 201 (1998). https://doi.org/10.1136/bmj.316.7126.201

[2] Hernán, M.A., Robins, J.M.: Causal inference: what if. Boca Raton: Chapman & Hall/CRC (2020)

[3] Prosperi, M., Guo, Y., Sperrin, M., Koopman, J.S., Min, J.S., He, X., Rich, S., Wang, M., Buchan, I.E., Bian, J.: Causal inference and counterfactual prediction in machine learning for actionable healthcare. Nature Machine Intelligence **2**, 369–375 (2020). https://doi.org/10.1038/s42256-020-0197-y

[4] Austin, P.C.: An introduction to propensity score methods for reducing the effects of confounding in observational studies. Multivariate behavioral research **46**(3), 399–424 (2011)

[5] Garrido, M., *et al.*: Methods for constructing and assessing propensity scores. Health Services Research **49**(5), 1701–20 (2014)

[6] Tian, Y., Schuemie, M.J., Suchard, M.A.: Evaluating large-scale propensity score performance through real-world and synthetic data experiments. International journal of epidemiology **47**(6), 2005–2014 (2018)

[7] Hill, J.L.: Bayesian nonparametric modeling for causal inference. Journal of Computational and Graphical Statistics **20**(1), 217–240 (2011)

[8] Wager, S., Athey, S.: Estimation and inference of heterogeneous treatment effects using random forests. Journal of the American Statistical Association **113**(523), 1228–1242 (2018)

[9] Johansson, F., Shalit, U., Sontag, D.: Learning representations for counterfactual inference. In: International Conference on Machine Learning, pp. 3020–3029 (2016)

[10] Shalit, U., Johansson, F.D., Sontag, D.: Estimating individual treatment effect: generalization bounds and algorithms. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 3076–3085. PMLR, International Convention Centre, Sydney, Australia (2017). http://proceedings.mlr.press/v70/shalit17a.html

[11] Shi, C., Blei, D., Veitch, V.: Adapting neural networks for the estimation of treatment effects. In: Advances in Neural Information Processing Systems, pp. 2507–2517 (2019)

[12] Alaa, A.M., Weisz, M., Van Der Schaar, M.: Deep counterfactual networks with propensity-dropout. arXiv preprint arXiv:1706.05966 (2017)

[13] Yoon, J., Jordon, J., Van Der Schaar, M.: Ganite: Estimation of individualized treatment effects using generative adversarial nets. In: International Conference on Learning Representations (2018)

[14] Louizos, C., Shalit, U., Mooij, J.M., Sontag, D., Zemel, R., Welling, M.: Causal effect inference with deep latent-variable models. In: Advances in Neural Information Processing Systems, pp. 6446–6456 (2017)

[15] Zhang, W., Liu, L., Li, J.: Treatment effect estimation with disentangled latent factors. arXiv preprint arXiv:2001.10652 (2020)

[16] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. arXiv preprint arXiv:1511.05644 (2015)

[17] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)

[18] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27**, 2672–2680 (2014)

[19] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. Advances in neural information processing systems **29**, 2172–2180 (2016)

[20] Funk, M.J., Westreich, D., Wiesen, C., Stürmer, T., Brookhart, M.A., Davidian, M.: Doubly robust estimation of causal effects. American journal of epidemiology **173**(7), 761–767 (2011)

[21] Dudík, M., Erhan, D., Langford, J., Li, L., *et al.*: Doubly robust policy evaluation and optimization. Statistical Science **29**(4), 485–511 (2014)

[22] Rubin, D.B.: Estimating causal effects of treatments in randomized and nonrandomized studies. Journal of educational Psychology **66**(5), 688–701 (1974)

[23] Rosenbaum, P.R., Rubin, D.B.: The central role of the propensity score in observational studies for causal effects. Biometrika **70**(1), 41–55 (1983). https://doi.org/10.1093/biomet/70.1.41

[24] Imbens, G.W.: The role of the propensity score in estimating dose-response functions. Biometrika **87**(3), 706–710 (2000)

[25] Pearl, J., Glymour, M., Jewell, N.P.: Causal Inference in Statistics: A Primer. Wiley, ??? (2016). https://books.google.com/books?id=L3G-CgAAQBAJ

[26] Porter, K.E., Gruber, S., Van Der Laan, M.J., Sekhon, J.S.: The relative performance of targeted maximum likelihood estimators. The International Journal of Biostatistics **7**(1) (2011)

[27] Chipman, H.A., George, E.I., McCulloch, R.E., *et al.*: Bart: Bayesian additive regression trees. The Annals of Applied Statistics **4**(1), 266–298 (2010)

[28] Athey, S., Imbens, G.: Recursive partitioning for heterogeneous causal effects. Proceedings of the National Academy of Sciences **113**(27), 7353–7360 (2016)

[29] Lu, M., Sadiq, S., Feaster, D.J., Ishwaran, H.: Estimating individual treatment effect in observational data using random forest methods. Journal of Computational and Graphical Statistics **27**(1), 209–219 (2018)

[30] Dehejia, R.H., Wahba, S.: Propensity score-matching methods for non-experimental causal studies. Review of Economics and statistics **84**(1), 151–161 (2002)

[31] Lunceford, J.K., Davidian, M.: Stratification and weighting via the propensity score in estimation of causal treatment effects: a comparative study. Statistics in medicine **23**(19), 2937–2960 (2004)

[32] Crump, R.K., Hotz, V.J., Imbens, G.W., Mitnik, O.A.: Nonparametric tests for treatment effect heterogeneity. The Review of Economics and Statistics **90**(3), 389–405 (2008)

[33] Alaa, A., van der Schaar, M.: Limits of estimating heterogeneous treatment effects: Guidelines for practical algorithm design. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 129–138. PMLR, Stockholmsmässan, Stockholm Sweden (2018). http://proceedings.mlr.press/v80/alaa18a.html

[34] Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. Journal of the American statistical Association **112**(518), 859–877 (2017)

[35] Almond, D., Chay, K.Y., Lee, D.S.: The costs of low birth weight. The

Quarterly Journal of Economics **120**(3), 1031–1083 (2005)

[36] LaLonde, R.J.: Evaluating the econometric evaluations of training programs with experimental data. The American economic review, 604–620 (1986)

[37] Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)

[38] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

8 Competing Interests statement

All authors have no conflicts to declare.

9 Figure Legends

- Figure 1 Directed acyclic graph modeling the causal relationships among a treatment t, outcome y and pre-treatment covariates X, under a latent space Z.
- Figure 2 Architecture of the counterfactual network to estimate the counterfactual outcome.
- Figure 3 Architecture of the four-headed, doubly robust neural network to calculate individual treatment effects.
- Figure 4 From left to right a) Comparison of the performance (ATE) of DR-VIDAL vs. all other models on samples from the generative process of CEVAE, defined in equation 22; b-c) Performance comparison (PEHE) of GANITE vs. DR-VIDAL, with or without the doubly robust (DR, w/o DR) feature, on samples from the generative processes defined in equations 22(left panel) and 23 (right panel).

Table 1 Performance of $\sqrt{\epsilon_{PEHE}}$ and R_{Pol} (mean ± st.dev) of various models on the IHDP, Twins and Jobs datasets. TARNet was originally developed in TensorFlow. We re-implemented TARNet in Pytorch for IHDP and Jobs dataset. (*) is used to indicate methods that DR-VIDAL shows a statistically significant improvement over.

	IHDP($\sqrt{\epsilon_{PEHE}}$)		**Twins($\sqrt{\epsilon_{PEHE}}$)**		**Jobs(R_{Pol})**	
	Out-Sample	In-Sample	Out-Sample	In-Sample	Out-Sample	In-Sample
OLS/LR1	5.8 ± 0.3*	5.8 ± 0.3*	0.318 ± 0.007	0.319 ± 0.005*	0.23 ± 0.02*	0.22 ± 0.00*
OLS/LR2	2.5 ± 0.1*	2.4 ± 0.1*	0.320 ± 0.003*	0.320 ± 0.001*	0.24 ± 0.01*	0.21 ± 0.00*
BLR	5.8 ± 0.3*	5.8 ± 0.3*	0.323 ± 0.018*	0.312 ± 0.002*	0.25 ± 0.02*	0.22 ± 0.01*
k-NN	4.1 ± 0.2*	2.1 ± 0.1*	0.345 ± 0.007*	0.333 ± 0.003*	0.26 ± 0.02*	0.02 ± 0.00*
BART	2.3 ± 0.1*	2.1 ± 0.2*	0.338 ± 0.016*	0.347 ± 0.009*	0.25 ± 0.00*	0.23 ± 0.02*
R Forest	6.6 ± 0.3*	4.2 ± 0.2*	0.321 ± 0.005*	0.306 ± 0.002	0.28 ± 0.02*	0.23 ± 0.01*
C Forest	3.8 ± 0.2*	3.8 ± 0.2*	0.316 ± 0.011	0.366 ± 0.003*	0.20 ± 0.02*	0.19 ± 0.00*
BNN	2.1 ± 0.1*	2.2 ± 0.1*	0.321 ± 0.018*	0.325 ± 0.003*	0.24 ± 0.02*	0.20 ± 0.01*
TARNET (TensorFlow)	0.95 ± 0.02*	0.88 ± 0.02*	0.315 ± 0.003	0.317 ± 0.007	0.21 ± 0.01*	0.17 ± 0.01*
TARNeT (Pytorch)	1.10 ± 0.02*	-	-	-	0.29 ± 0.06*	-
CFR$_{WASS}$	0.76 ± 0.0*	0.71 ± 0.0*	0.313 ± 0.008	0.315 ± 0.007	0.21 ± 0.01*	0.17 ± 0.01*
GANITE	2.4 ± 0.4*	1.9 ± 0.4*	0.297 ± 0.05	0.289 ± 0.005	0.14 ± 0.01*	0.13 ± 0.01*
CEVAE	2.6 ± 0.1*	2.7 ± 0.1*	n.r	n.r	0.26 ± 0.0*	0.15 ± 0.0*
DR-VIDAL	**0.69 ± 0.06**	**0.69 ± 0.05**	**0.318 ± 0.008**	**0.317 ± 0.002**	**0.10 ± 0.01**	**0.09 ± 0.005**

Table 2 Performance of $\sqrt{\epsilon_{PEHE}}$ and R_{Pol} (mean ± st.dev) of various models on the IHDP, Twins and Jobs datasets. TARNet was originally developed in TensorFlow. We re-implemented TARNet in Pytorch for IHDP and Jobs dataset. (*) is used to indicate methods that DR-VIDAL shows a statistically significant improvement over.

	IHDP(ϵ_{ATE})		**Twins(ϵ_{ATE})**		**Jobs(ϵ_{ATT})**	
	Out-Sample	In-Sample	Out-Sample	In-Sample	Out-Sample	In-Sample
OLS/LR1	0.94 ± 0.06	0.73 ± 0.04	0.0069 ± 0.0056	0.0038 ± 0.0025	0.08 ± 0.04	0.01 ± 0.00
OLS/LR2	0.31 ± 0.02	0.14 ± 0.01	0.0070 ± 0.0025	0.0039 ± 0.0025	0.08 ± 0.03	0.01 ± 0.01
BLR	0.93 ± 0.05	0.72 ± 0.04	0.0334 ± 0.0092	0.0057 ± 0.0036	0.08 ± 0.03	0.01 ± 0.011
k-NN	0.90 ± 0.05	0.14 ± 0.01	0.0051 ± 0.0039	0.0028 ± 0.0021	0.13 ± 0.05	0.21 ± 0.01
BART	0.34 ± 0.02	0.23 ± 0.01	0.1265 ± 0.0234	0.1206 ± 0.0236	0.08 ± 0.03	0.02 ± 0.00
R Forest	0.96 ± 0.06	0.73 ± 0.05	0.0080 ± 0.0051	0.0049 ± 0.0034	0.09 ± 0.04	0.03 ± 0.01
C Forest	0.40 ± 0.03	0.18 ± 0.01	0.0335 ± 0.0083	0.0286 ± 0.0035	0.07 ± 0.03	0.03 ± 0.01
BNN	0.42 ± 0.03	0.37 ± 0.03	0.0203 ± 0.0071	0.0056 ± 0.0032	0.09 ± 0.04	0.03 ± 0.01
TARNET	0.28 ± 0.01	0.26 ± 0.01	0.0151 ± 0.0018	0.0108 ± 0.0017	0.09 ± 0.04	0.03 ± 0.01
CFR_{WASS}	0.27 ± 0.01	0.25 ± 0.01	0.0284 ± 0.0032	0.0112 ± 0.0016	0.09 ± 0.03	0.04 ± 0.01
GANITE	0.49 ± 0.05	0.43 ± 0.05	0.0089 ± 0.0075	0.0058 ± 0.0017	0.06 ± 0.03	0.01 ± 0.01
CEVAE	0.46 ± 0.02	0.34 ± 0.01	n.r	n.r	0.03 ± 0.01	0.02 ± 0.01
DR-VIDAL	**0.49 ± 0.06**	**0.49 ± 0.07**	**0.0111 ± 0.0137**	**0.0102 ± 0.0128**	**0.05 ± 0.02**	**0.04 ± 0.03**

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- DRVIDALNatureMILSupp.pdf