Appendix

A1 Algorithms

The algorithm to train the generative adversarial network and doubly robust multitask network for counterfactual outcome calculation and ITE estimation are discussed in Algorithms 1 and 2 respectively.

A2 Performance Metrics

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:

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

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

$$R_{pol}(\pi) = \frac{1}{N}\sum_{n=0}^{N}\Big[1 - \Big(\sum_{i=1}^{k}\Big[\frac{1}{|\Pi_i \cap T_i \cap E|}\sum_{x(n)\in\Pi_i\cap\ T_i\cap E} y_i(n) \times \frac{|\Pi_n \cap E|}{|E|}\Big]\Big)\Big] \tag{3}$$

where π_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:

$$ATT = \frac{1}{|T_1 \cap E|}\sum_{x_i\in T_1\cap E} Y_1(x_i) - \frac{1}{|T_0 \cap E|}\sum_{x_i\in C\cap E} Y_0(x_i) \tag{4}$$

$$\epsilon_{ATT} = \big|ATT - \frac{1}{|T_1 \cap E|}\sum_{x_i\in T_1\cap E}\hat{Y}_1(x_i) - \hat{Y}_0(x_i)\big| \tag{5}$$

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

A3 Synthetic dataset of CEVAE

$$\begin{aligned} \mathbf{z}_i \sim \text{Bern}(0.5); \qquad \mathbf{x}_i|\mathbf{z}_i &\sim \mathcal{N}(\mathbf{z}_i, \sigma_5^2\mathbf{z}_i + \sigma_3^2(1-\mathbf{z}_i)) \\ t_i|\mathbf{z}_i &\sim \text{Bern}(0.75\mathbf{z}_i + 0.25(1-\mathbf{z}_i)) \\ \mathbf{y}_i|t_i, \mathbf{z}_i &\sim \text{Bern}(\text{Sigmoid}(3(\mathbf{z}_i + 2(2t_i - 1)))) \end{aligned} \tag{6}$$

A4 Synthetic dataset of DR-VIDAL

$$
\begin{aligned}
\mathbf{z}_x \sim \text{Bern}(0.5); \quad & \mathbf{z}_t \sim \text{Bern}(0.5) \\
\mathbf{z}_{yf} \sim \text{Bern}(0.5); \quad & \mathbf{z}_{ycf} \sim \text{Bern}(0.5) \\
\mathbf{x}_x|\mathbf{z}_x \sim \mathcal{N}(\mathbf{z}_x, 5(\mathbf{z}_x) + 3(1-\mathbf{z}_x)) & \\
\mathbf{x}_t|\mathbf{z}_t \sim \mathcal{N}(\mathbf{z}_x, 2(\mathbf{z}z_t) + 0.5(1-\mathbf{z}_t)) & \\
\mathbf{x}_{yf}|\mathbf{z}_{yf} \sim \mathcal{N}(\mathbf{z}_{yf}, 10(\mathbf{z}_{yf}) + 6(1-\mathbf{z}_{yf})) & \\
\mathbf{x}_{ycf}|\mathbf{z}_{ycf} \sim \mathcal{N}(\mathbf{z}_{ycf}, 10(\mathbf{z}_{ycf}) + 6(1-\mathbf{z}_{ycf})) & \\
\mathbf{w}_t^T \sim \mathcal{U}((-0.1, 0.1)^{10\text{x}1}); \quad & \mathbf{n}_t \sim \mathcal{N}(0, 0.1) \\
\mathbf{w}_y^T \sim \mathcal{U}((-1, 1)^{10\text{x}2}); \quad & \mathbf{n}_y \sim \mathcal{N}(0^{2\text{x}1}, 0.1\text{x}\mathcal{I}^{2\text{x}2}) \\
t|x \sim \text{Bern}(\text{Sigmoid}(\mathbf{w}_t^T\mathbf{x} + \mathbf{n}_t)); \quad & \mathbf{y}|\mathbf{x} \sim \mathbf{w}_y^T\mathbf{x} + \mathbf{n}_y
\end{aligned} \tag{7}
$$

A5 Datasets

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[1,2,3,4]. 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.

A7 Differences with CEVAE 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 simplistic, 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. 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.

A8 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 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, in TARNet the weights corresponding to each sample 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, updating the parameters of the outcome head of the factual outcome during training. In contrast DR-VIDAL provides the entire potential outcome vector, comprising both the factual and the counterfactual outcomes. For each training sample, the loss function is calculated for both outcomes, and the corresponding parameters of both the outcome heads are updated.

A6 Derivation of the ELBO Loss for VAE

From Figure section 4.1 and the causal graph in figure 1 in the main text, $p_{\phi_d}(\mathbf{x}|\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{y_cf})$ and $p_{\phi_d}(\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{y_cf}|\mathbf{x})$ are the true likelihood and true posterior respectively. The posterior is hard to evaluate, so we have to approximate the true posterior to the product of the factorized known distributions $q_{\phi_x}(\mathbf{z}_x|\mathbf{x})$, $q_{\phi_t}(\mathbf{z}_t|\mathbf{x})$, $q_{\phi_{yf}}(\mathbf{z}_{yf}|\mathbf{x})$ and $q_{\phi_{ycf}}(\mathbf{z}_{ycf}|\mathbf{x})$ by minimising the KL divergence as follows,

$$
\begin{aligned}
&KL\big(q_{\phi_x}(\mathbf{z}_x|\mathbf{x})q_{\phi_t}(\mathbf{z}_t|\mathbf{x})q_{\phi_{yf}}(\mathbf{z}_{yf}|\mathbf{x})q_{\phi_{ycf}}(\mathbf{z}_{ycf}|\mathbf{x})||p_{\phi_d}(\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf}|\mathbf{x})\big) \\
&= \int\int\int\int q_{\phi_x}(\mathbf{z}_x|\mathbf{x})q_{\phi_t}(\mathbf{z}_t|\mathbf{x})q_{\phi_{yf}}(\mathbf{z}_{yf}|\mathbf{x})q_{\phi_{ycf}}(\mathbf{z}_{ycf}|\mathbf{x})\big[\log\frac{q_{\phi_x}(\mathbf{z}_x|\mathbf{x})q_{\phi_t}(\mathbf{z}_t|\mathbf{x})q_{\phi_{yf}}(\mathbf{z}_{yf}|x)}{p_{\phi_d}(\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf}|\mathbf{x})}\big]\mathbf{dz}_x\mathbf{dz}_t\mathbf{dz}_{yf}\mathbf{dz}_{ycf} \\
&= \int\int\int\int q_{\phi_x}(\mathbf{z}_x|\mathbf{x})q_{\phi_t}(\mathbf{z}_t|\mathbf{x})q_{\phi_{yf}}(\mathbf{z}_{yf}|\mathbf{x})q_{\phi_{ycf}}(\mathbf{z}_{ycf}|\mathbf{x}) \\
&\quad \big[\log q_{\phi_x}(\mathbf{z}_x|\mathbf{x}) + \log q_{\phi_t}(\mathbf{z}_t|\mathbf{x}) + \log q_{\phi_{yf}}(\mathbf{z}_{yf}|\mathbf{x}) + \log q_{\phi_{ycf}}(\mathbf{z}_{ycf}|\mathbf{x}) - \log p_{\phi_d}(\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf}|\mathbf{x})\big]\mathbf{dz}_x\mathbf{dz}_t\mathbf{dz}_{yf}\mathbf{dz}_{ycf} \\
&= \int\int\int\int q_{\phi_x}(\mathbf{z}_x|\mathbf{x})q_{\phi_t}(\mathbf{z}_t|\mathbf{x})q_{\phi_{yf}}(\mathbf{z}_{yf}|\mathbf{x})q_{\phi_{ycf}}(\mathbf{z}_{ycf}|\mathbf{x})\big[\log q_{\phi_x}(\mathbf{z}_x|\mathbf{x}) + \log q_{\phi_t}(\mathbf{z}_t|\mathbf{x}) + \\
&\quad \log q_{\phi_{yf}}(\mathbf{z}_{yf}|\mathbf{x}) + \log q_{\phi_{ycf}}(\mathbf{z}_{ycf}|\mathbf{x}) - \log p_{\phi_d}(x|\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf}) - \log p_{\phi_d}(\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf}) \\
&\quad + \log p_{\phi_d}(x)\big]\mathbf{dz}_x\mathbf{dz}_t\mathbf{dz}_{yf}\mathbf{dz}_{ycf} \\
&= \int q_{\phi_x}(\mathbf{z}_x|\mathbf{x})\log\frac{q_{\phi_x}(\mathbf{z}_x|\mathbf{x})}{p_{\phi_d}(\mathbf{z}_x)}\mathbf{dz}_x + \int q_{\phi_t}(\mathbf{z}_t|\mathbf{x})\log\frac{q_{\phi_t}(\mathbf{z}_t|\mathbf{x})}{p_{\phi_d}(\mathbf{z}_t)}\mathbf{dz}_t + \int q_{\phi_{yf}}(\mathbf{z}_{yf}|\mathbf{x})\log\frac{q_{\phi_{yf}}(\mathbf{z}_{yf}|\mathbf{x})}{p_{\phi_d}(\mathbf{z}_{yf})}\mathbf{dz}_{yf} \\
&\quad + \int q_{\phi_{ycf}}(\mathbf{z}_{ycf}|\mathbf{x})\log\frac{q_{\phi_x}(\mathbf{z}_{ycf}|\mathbf{x})}{p_{\phi_d}(\mathbf{z}_x)}\mathbf{dz}_{ycf} - \int\int\int\int \big[q_{\phi_x}(\mathbf{z}_x|\mathbf{x})q_{\phi_t}(\mathbf{z}_t|\mathbf{x})q_{\phi_{yf}}(\mathbf{z}_{yf}|\mathbf{x}) \\
&\quad q_{\phi_{ycf}}(\mathbf{z}_{ycf}|\mathbf{x})\log p_{\phi_d}(\mathbf{x}|\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf})\big]\mathbf{dz}_x\mathbf{dz}_t\mathbf{dz}_{yf}\mathbf{dz}_{ycf} + \log p_{\phi_d}(\mathbf{x}) \\
&= KL\big(q_{\phi_x}(\mathbf{z}_x|\mathbf{x})||p_{\phi_d}(\mathbf{z}_x))\big) + KL\big(q_{\phi_t}(\mathbf{z}_t|\mathbf{x})||p_{\phi_d}(\mathbf{z}_t))\big) + KL\big(q_{\phi_{yf}}(\mathbf{z}_{yf}|\mathbf{x})||p_{\phi_d}(\mathbf{z}_{yf}))\big) \\
&\quad + KL\big(q_{\phi_{ycf}}(\mathbf{z}_{ycf}|\mathbf{x})||p_{\phi_d}(\mathbf{z}_{ycf}))\big) - \mathbb{E}_{q_{\phi_x}, q_{\phi_t}, q_{\phi_{yf}}, q_{\phi_{ycf}}}[\log p(\mathbf{x}|\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf})] \\
&\quad + \log p_{\phi_d}(\mathbf{x})
\end{aligned}
$$

where, the distributions $q_{\phi_x}(\mathbf{z}_x|x)$, $q_{\phi_t}(\mathbf{z}_t|x)$, $q_{\phi_{yf}}(z_{yf}|x)$, $q_{\phi_{ycf}}(z_{ycf}|x)$ and $p_{\phi_d}(x|\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf})$ are parameterized by the parameters $\phi_x, \phi_t, \phi_{yf}, \phi_{ycf}, \phi_d$. The KL divergence of two distributions is always greater than or equal to zero. So,

$$
\begin{aligned}
&KL\big(q_{\phi_x}(\mathbf{z}_x|x)q_{\phi_t}(\mathbf{z}_t|x)q_{\phi_{yf}}(\mathbf{z}_{yf}|\mathbf{x})q_{\phi_{ycf}}(\mathbf{z}_{ycf}|x)||p_{\phi_d}(\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf}|x)\big) \geq 0, \\
&\log p_{\phi_d}(\mathbf{z}) \geq \mathcal{L}_{ELBO} \quad \text{where,} \\
&\mathcal{L}_{ELBO}(\phi_x, \phi_t, \phi_{yf}, \phi_{ycf}; \mathbf{x}, \mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf}) \\
&\quad = \mathbb{E}_{q_{\phi_x}, q_{\phi_t}, q_{\phi_{yf}}, q_{\phi_{ycf}}}[\log p(\mathbf{x}|\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf})] \\
&\qquad - KL\big(q_{\phi_x}(\mathbf{z}_x|\mathbf{x})||p_{\phi_d}(\mathbf{z}_x))\big) - KL\big(q_{\phi_t}(\mathbf{z}_t|\mathbf{x})||p_{\phi_d}(\mathbf{z}_t))\big) - KL\big(q_{\phi_{yf}}(\mathbf{z}_{yf}|\mathbf{x})||p_{\phi_d}(\mathbf{z}_{yf}))\big) \\
&\qquad - KL\big(q_{\phi_{ycf}}(\mathbf{z}_{ycf}|\mathbf{x})||p_{\phi_d}(\mathbf{z}_{ycf}))\big)
\end{aligned}
$$

A9 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 $q_{\phi_x}(\mathbf{z}_x|\mathbf{x})$, $q_{\phi_t}(\mathbf{z}_t|\mathbf{x})$, $q_{\phi_{yf}}(\mathbf{z}_{yf}|\mathbf{x})$, $q_{\phi_{ycf}}(\mathbf{z}_{ycf}|\mathbf{x})$. The shared neural network has 3 layers, each with 15 nodes. The networks with $q_{\phi_x}(\mathbf{z}_x|\mathbf{x})$,

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

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

	IHDP $\sqrt{\epsilon_{PEHE}^{out-of-s}}$	**Jobs** $R_{Pol}^{out-of-s}$	**Twins** $\sqrt{\epsilon_{PEHE}^{out-of-s}}$
DR-VIDAL	**0.62 ± 0.06**	**0.102 ± 0.01**	**0.318 ± 0.008**
DR-VIDAL (w/o DR loss)	0.85 ± 0.06	0.110 ± 0.01	0.324 ± 0.007
DR-VIDAL (w/o Info loss)	0.67 ± 0.04	0.109 ± 0.01	0.318 ± 0.012
DR-VIDAL (w/o DR + Info loss)	0.81 ± 0.05	0.113 ± 0.01	0.326 ± 0.008

Table 1: Performance of the all the different DR-VIDAL configurations on the IHDP, Jobs and Twins datasets (1000, 10, and 100 realizations, respectively). Results show the out-of-sample (mean ± st.dev) error (PEHE) and policy risk (R_{Pol}).

$q_{\phi_t}(\mathbf{z}_t|\mathbf{x})$, $q_{\phi_{yf}}(\mathbf{z}_{yf}|x)$, $q_{\phi_{ycf}}(\mathbf{z}_{ycf}|\mathbf{x})$ 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 $p_{\phi_d}(\mathbf{x}|\mathbf{z}_x, \mathbf{z}_t, \mathbf{z}_{yf}, \mathbf{z}_{ycf})$. 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 **??**) 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[5]. 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://bitbucket.org/goingdeep2406/DR-VIDAL/src/master/.

A10 Performance of all the various DR-VIDAL configurations

The performance of all the various DR-VIDAL configurations are mentioned in Table 1.

Methods	Out-Sample	In-Sample
OLS/LR1	0.08 ± 0.04	0.01 ± 0.00
OLS/LR2	0.08 ± 0.03	0.01 ± 0.01
BLR	0.08 ± 0.03	0.01 ± 0.011
k-NN	0.13 ± 0.05	0.21 ± 0.01
BART	0.08 ± 0.03	0.02 ± 0.00
R Forest	0.09 ± 0.04	0.03 ± 0.01
C Forest	0.07 ± 0.03	0.03 ± 0.01
BNN	0.09 ± 0.04	0.03 ± 0.01
TARNET	0.11 ± 0.04	0.05 ± 0.02
CFR$_{WASS}$	0.09 ± 0.03	0.04 ± 0.01
GANITE	0.06 ± 0.03	0.01 ± 0.01
CEVAE	0.03 ± 0.01	0.02 ± 0.01
DR-VIDAL	$\mathbf{0.05 \pm 0.02}$	$\mathbf{0.04 \pm 0.03}$

Table 2: Performance of various models on the Jobs dataset for ϵ_{ATT} (mean $\pm$ st.dev).

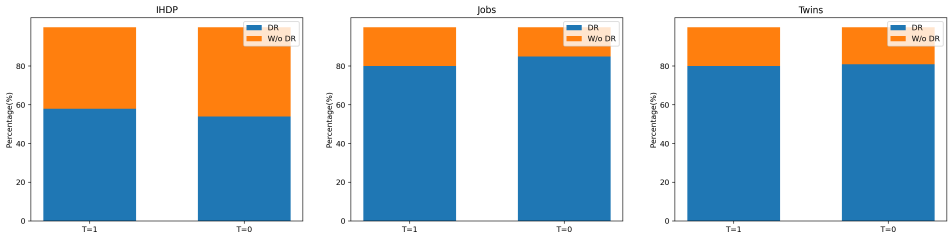

Figure 1: Performance comparison of doubly robust vs. non-doubly robust version of DR-VIDAL. Panels, from left to right, show results on IHDP, Jobs and Twins datasets (100, 10, 100 iterations), respectively.

A11 DR-VIDAL's performance on IHDP and Twins datasets for $\sqrt{\epsilon_{PEHE}}$ values

The performance of various models for $\sqrt{\epsilon_{PEHE}}$ values on the IHDP and Twins dataset are shown in Table 3.

A10 DR-VIDAL's in-sample performance on Jobs dataset for R_{Pol} values

The performance of various models on the Jobs dataset for R_{Pol} values are shown in Table 2.

A12 Performance comparison of doubly robust vs. non-doubly robust version of DR-VIDAL

Performance comparison of doubly robust vs. non-doubly robust version of DR-VIDAL is shown in Figure 1.The bar plots show how many times one model setup is better than the other in terms of error on the factual outcome (yf).

A13 t_SNE of representations

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 2. 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.

$Methods$	IHDP(ϵ_{ATE})		Twins(ϵ_{ATE})	
	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
OLS/LR2	0.31 ± 0.02	0.14 ± 0.01	0.0070 ± 0.0025	0.0039 ± 0.0025
BLR	0.93 ± 0.05	0.72 ± 0.04	0.0334 ± 0.0092	0.0057 ± 0.0036
k-NN	0.90 ± 0.05	0.14 ± 0.01	0.0051 ± 0.0039	0.0028 ± 0.0021
BART	0.34 ± 0.02	0.23 ± 0.01	0.1265 ± 0.0234	0.1206 ± 0.0236
R Forest	0.96 ± 0.06	0.73 ± 0.05	0.0080 ± 0.0051	0.0049 ± 0.0034
C Forest	0.40 ± 0.03	0.18 ± 0.01	0.0335 ± 0.0083	0.0286 ± 0.0035
BNN	0.42 ± 0.03	0.37 ± 0.03	0.0203 ± 0.0071	0.0056 ± 0.0032
TARNET	0.28 ± 0.01	0.26 ± 0.01	0.0151 ± 0.0018	0.0108 ± 0.0017
CFR$_{WASS}$	0.27 ± 0.01	0.25 ± 0.01	0.0284 ± 0.0032	0.0112 ± 0.0016
GANITE	0.49 ± 0.05	0.43 ± 0.05	0.0089 ± 0.0075	0.0058 ± 0.0017
CEVAE	0.46 ± 0.02	0.34 ± 0.01	n.r	n.r
DR-VIDAL	$\mathbf{0.69 \pm 0.06}$	$\mathbf{0.57 \pm 0.07}$	$\mathbf{0.0111 \pm 0.0137}$	$\mathbf{0.0102 \pm 0.0128}$

Table 3: Performance of various models on the IHDP and Twins datasets for ϵ_{ATE} (mean $\pm$ st.dev).

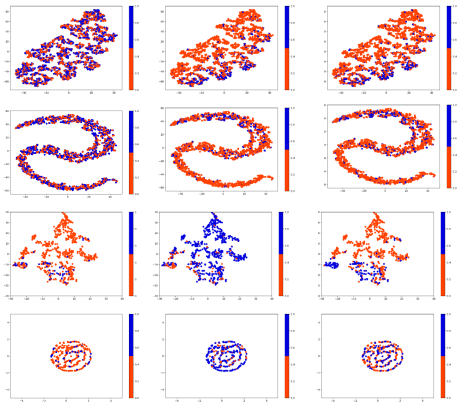

Figure 2: Visualization of the latent representation learned by the VAE module of DR-VIDAL for the Twins and Jobs dataset using t-SNE. The 1^{st} and 2^{nd} panels show the t-SNE before and after training the network for Twins dataset. The 3^{rd} and 4^{th} panels show the same for Jobs dataset. From left to right, the plots show the t-SNE of treatment, factual and counterfactual outcomes.

References

1. Hill JL. Bayesian nonparametric modeling for causal inference. Journal of Computational and Graphical Statistics. 2011;20(1):217-40.
2. Shalit U, Johansson FD, Sontag D. Estimating individual treatment effect: generalization bounds and algorithms. In: Precup D, Teh YW, editors. Proceedings of the 34th International Conference on Machine Learning. vol. 70 of Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR; 2017. p. 3076-85. Available from: http://proceedings.mlr.press/v70/shalit17a.html.
3. Louizos C, Shalit U, Mooij JM, Sontag D, Zemel R, Welling M. Causal effect inference with deep latent-variable models. In: Advances in neural information processing systems; 2017. p. 6446-56.
4. 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. .
5. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014.