Supplementary Materials of Counterfactual Fairness with Disentangled Causal Effect Variational Autoencoder

Hyemi Kim[1], **Seungjae Shin**[1], **JoonHo Jang**[1], **Kyungwoo Song**[1], **Weonyoung Joo**[1], **Wanmo Kang**[2], **Il-Chul Moon**[1]
[1] Department of Industrial and Systems Engineering
[2] Department of Mathematical Sciences
Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea
{khm0308,tmdwo0910,adkto8093,gtshs2,es345,wanmo.kang,icmoon}@kaist.ac.kr

1 Theoretical Analysis of Covariance Structure

In this section, we analyze the covariance structure of DCEVAE, CEVAE, and mCEVAE. We assume the linear Variational Autoencoder (VAE) structure to figure out the stationary point of Evidence Lower Bound (ELBO) in VAE [13].

1.1 DCEVAE

Network structure and ELBO

$$\begin{aligned} p(x_r|u_r) &= \mathcal{N}(W_r u_r + \mu_r, \sigma^2 I), \\ p(x_d|a,u_d) &= \mathcal{N}(W_d[a,u_d] + \mu_d, \sigma^2 I), \\ p(y|a,u) &= \mathcal{N}(W_y[a,u] + \mu_y, \sigma^2 I), \\ q(u|a,x,y) &= \mathcal{N}(V_u([a,x,y] - \mu), \Sigma), \\ \bar{q}(u|a,x,y) &= \mathcal{N}(\bar{\mu}, \bar{\Sigma}), \\ p(u) &= \mathcal{N}(0, I), \end{aligned}$$

where $u = [u_r, u_d]$, $[\cdot,\cdot]$ means concatenation of vectors, Σ is the covariance matrix of the joint distribution of latent variables, and $\bar{\Sigma}$ is the covariance matrix of the permuted u. The detail of permuted u is specified in line 8-11, Algorithm 1, Appendix.

We should optimize the loss function in Eq. 1 to train the DCEVAE. For simplification, we first derive the optimal discriminator D_{ψ^*} which maximizes $\mathcal{M}$ in *Maximization Step*, and then derive the Σ^* which minimizes $\mathcal{L}$ in *Minimzation Step*.

$$\begin{aligned} \min_{W,V,\Sigma} \mathcal{L} &= -E_{q(u|a,x,y)}[\log p(x_r|u_r) + \log p(x_d|a,u_d) + \log p(y|a,u)] + KL(q(u|a,x,y)||p(u)) \\ &\quad + \beta E_{q(u|a,x,y)}[\log \frac{D_\psi(u)}{1 - D_\psi(u)}] \\ \max_{\psi} \mathcal{M} &= E_{u\sim q(u|a,x,y)}[\log D_\psi(u)] + E_{u\sim \bar{q}(u|a,x,y)}[\log 1 - D_\psi(u)]. \end{aligned} \tag{1}$$

Maximization Step First, similar to GAN [5], we consider the optimal discriminator D_ψ for any given parameter sets W, V, and Σ. Then, the optimal discriminator D_{ψ^*} is expressed as

$$D_{\psi^*} = \frac{q(u|x)}{q(u|x) + \bar{q}(u|x)}. \tag{2}$$

Minimization Step With optimal discriminator D_{ψ^*} in Eq. 2, the minimization step is expressed as

$$\begin{aligned} \min_{W,V,\Sigma} \mathcal{L} &= -\underbrace{E_{q(u|a,x,y)}[\log p(x_r|u_r) + \log p(x_d|a,u_d) + \log p(y|a,u)]}_{\text{A}} + \underbrace{KL(q(u|a,x,y)||p(u))}_{\text{B}} \\ &\quad + \beta \underbrace{KL(q(u|a,x,y)||\bar{q}(u|a,x,y))}_{\text{C}}. \end{aligned} \tag{3}$$

Each of the terms (A-C) in Eq. 3 is expressed in a closed form of the linear VAE. The term (A) is expressed as

$$\begin{aligned}
&E_{q(u|a,x,y)}[\log p(x_r|u_r)] \\
&= E_{q(u|a,x,y)}[-(W_r u_r - (x_r - \mu_r))^T (W_r u_r - (x_r - \mu_r))/2\sigma^2 - \frac{d}{2}\log 2\pi\sigma^2] \\
&= E_{q(u|a,x,y)}[\frac{-(W_r u_r)^T (W_r u_r) + 2(x_r - \mu_r)^T W_r u - (x_r - \mu_r)^T (x - \mu_r)}{\sigma^2} - \frac{d}{2}\log 2\pi\sigma^2].
\end{aligned}$$

Note that $W_r u_r \sim \mathcal{N}(W_r V_r, W_r \Sigma_r W_r^T)$, we compute $E_{q(u|a,x,y)}[\log p(x_r|u)]$ as

$$\begin{aligned}
E_{q(u|a,x,y)}[\log p(x_r|u)] = \frac{1}{2\sigma^2}[&-tr(W_r M_r \Sigma W_r^T) - V_r^T W_r^T W_r V_r \\
&+ 2(x_r - \mu_r)^T W_r V_r - (x_r - \mu_r)^T (x_r - \mu_r)] - \frac{d}{2}\log 2\pi\sigma^2.
\end{aligned}$$

Similarly, we compute $E_{q(u|a,x,y)}[\log p(x_d|a,u_d)]$ as

$$\begin{aligned}
E_{q(u|a,x,y)}[\log p(x_d|a,u_d)] = \frac{1}{2\sigma^2}[&-tr(W_d M_d \Sigma W_d^T) - V_d^T W_d^T W_d V_d \\
&+ 2(x_d - \mu_d)^T W_d V_d - (x_d - \mu_d)^T (x_d - \mu_d)] - \frac{d}{2}\log 2\pi\sigma^2,
\end{aligned}$$

where

$$M_r = \begin{pmatrix} I_{u_r \times u_r} & 0 \\ 0 & 0 \end{pmatrix} \text{ and } M_d = \begin{pmatrix} 0 & 0 \\ 0 & I_{u_d \times u_d} \end{pmatrix}.$$

Also, $E_{q(u|a,x,y)}[\log p(y|a,u)]$ is expressed as

$$\begin{aligned}
E_{q(u|a,x,y)}[\log p(y|a,u)] = \frac{1}{2\sigma^2}[&-tr(W_Y \Sigma W_Y^T) - V_d^T W_Y^T W_Y V_Y \\
&+ 2(y - \mu_Y)^T W_Y V - (y - \mu_Y)^T (y - \mu_Y)] - \frac{d}{2}\log 2\pi\sigma^2,
\end{aligned}$$

where $V = [v_u, v_d]$.
The term (B) is expressed as

$$KL(q(u|a,x,y)||p(u)) = \frac{1}{2}(-\log det\Sigma + (x-\mu)^T V^T V (x-\mu) + tr(\Sigma) - q),$$

where q is the dimension of covariance matrix.
The term (C) is expressed as

$$KL(q(u|a,x,y)||\bar{q}(u|a,x,y)) = \frac{1}{2}(-\log \frac{det\Sigma}{det\bar{\Sigma}} + (\bar{\mu}-\mu)^T \bar{\Sigma}^{-1}(\bar{\mu}-\mu) + tr(\bar{\Sigma}^{-1}\Sigma) - q).$$

Finding stationary point To compute the stationary point Σ^*, we take the derivative with respect to Σ on Eq. 3. With terms (A), (B), and (C), the stationary point Σ^* is derived as

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \Sigma} &= \frac{n}{2}\left(\frac{1}{\sigma^2} W_r^T M_r^T W_r + \frac{1}{\sigma^2} W_d^T M_d^T W_d + \frac{1}{\sigma^2} diag\left(W_y^T W_y\right) - \Sigma^{-T} + I - \beta\Sigma^{-T} + \beta\bar{\Sigma}^{-T}\right) = 0, \\
\Sigma^* &= \left\{\frac{1}{1+\beta}\left(\frac{1}{\sigma^2} W_r^T M_r^T W_r + \frac{1}{\sigma^2} W_d^T M_d^T W_d + \frac{1}{\sigma^2} diag\left(W_y^T W_y\right) + I + \beta\bar{\Sigma}^{-T}\right)\right\}^{-1}.
\end{aligned}$$

1.2 CEVAE

Network structure and ELBO

$$\begin{aligned}
p(a|u) &= \mathcal{N}(W_A u + \mu_A, \sigma^2 I), \\
p(x|u) &= \mathcal{N}(W_X u + \mu_X, \sigma^2 I), \\
p(y|a,u) &= \mathcal{N}(a(W_{Y_1} u + \mu_{Y_1}) + (1-a)(W_{Y_0} u + \mu_{Y_0}), \sigma^2 I), \\
q(a|x) &= \text{Bern}(p_A), p_A = \sigma(V_a x + b_a) \\
q(y|a,x) &= \text{Bern}(p_Y), p_Y = \sigma(a(V_{y_1} x + b_{y_1}) + (1-a)(V_{y_0} x + b_{y_0})) \\
q(u|a,x,y) &= \mathcal{N}(aV_1([x,y] - \mu_1) + (1-a)V_0([x,y] - \mu_0), a\Sigma_1 + (1-a)\Sigma_0), \\
p(u) &= \mathcal{N}(0, I),
\end{aligned}$$

where Σ_0 and Σ_1 are the covaraince structure of each encoder with the intervention $a = 0$ and $a = 1$.

To train the CEVAE, we should optimize the following loss function.

$$\begin{aligned}\min_{W,V,\Sigma} \mathcal{L} = &- \underbrace{E_{q(u,a,y|x)}[\log p(x|u) + \log p(a|u) + \log p(y|a,u)]}_{\text{A}} + \underbrace{E_{q(u,a,y|x)}[\log \frac{q(u,a,y|x)}{p(u)}]}_{\text{B}} \\ &+ \underbrace{\log q(a^*|x^*) + \log q(y^*|a^*,x^*)}_{\text{C}},\end{aligned} \tag{4}$$

where x^*, a^*, and y^* are the observed values in the training data.
The term (A) is derived in a similar way to the DCEVAE.

$$\begin{aligned}&E_{q(a,y,u|x)}[\log p(x|u)] \\ &= E_{q(a,y,u|x)}[\frac{-(W_x u - (x-\mu_x))^T(W_x u - (x-\mu_x))}{2\sigma^2} - \frac{d}{2}\log 2\pi\sigma^2] \\ &= E_{q(a,y,u|x)}[\frac{-(W_X u)^T(W_X u) + 2(x-\mu_X)^T W_X u - (x-\mu_X)^T(x-\mu_X)}{2\sigma^2} - \frac{d}{2}\log 2\pi\sigma^2].\end{aligned}$$

Note that $W_X u \sim \mathcal{N}(W_X V', W_X \Sigma' W_X^T)$, $V' = aV_1([x,y]-\mu_1) + (1-a)V_0([x,y]-\mu_0)$, and $\Sigma' = a\Sigma_1 + (1-a)\Sigma_0$. Then, we compute $E_{q(a,y,u|x)}[\log p(x|u)]$ as

$$\begin{aligned}E_{q(a,y,u|x)}[\log p(x|u)] &= E_{q(a|x)q(y|a,x)q(u|a,x,y)}[\log p(x|u)] \\ &= E_{q(a|x)q(y|a,x)}[\frac{1}{2\sigma^2}[-tr(W_X\Sigma' W_X^T) - V'^T W_X^T W_X V' \\ &+ 2(x-\mu_X)^T W_X V' - (x-\mu_X)^T(x-\mu_X)] - \frac{d}{2}\log 2\pi\sigma^2] \\ &= \sum_{a\in 0,1}\sum_{y\in 0,1} p_A^a(1-p_A)^{(1-a)} p_Y^y (1-p_Y)^{(1-y)}[\frac{1}{2\sigma^2}[-tr(W_X\Sigma' W_X^T) - V'^T W_X^T W_X V' \\ &+ 2(x-\mu_X)^T W_X V' - (x-\mu_X)^T(x-\mu_X)] - \frac{d}{2}\log 2\pi\sigma^2].\end{aligned}$$

Similarly, $E_{q(a,y,u|x)}[\log p(y|u)]$ and $E_{q(a,y,u|x)}[\log p(a|u)]$ are computed as

$$\begin{aligned}E_{q(a,y,u|x)}[\log p(y|u)] = &\sum_{a\in 0,1}\sum_{y\in 0,1} p_A^a(1-p_A)^{(1-a)} p_Y^y (1-p_Y)^{(1-y)}\big[\frac{1}{2\sigma^2}[-tr(W_{Y_a}\Sigma' W_{Y_a}^T) - V'^T W_{Y_a}^T W_{Y_a} V' \\ &+ 2(y-\mu_{Y_a})^T W_{Y_a} V' - (y-\mu_{Y_a})^T(y-\mu_{Y_a})] - \frac{1}{2}\log 2\pi\sigma^2\big], \\ E_{q(a,y,u|x)}[\log p(a|u)] = &\sum_{a\in 0,1}\sum_{y\in 0,1} p_A^a(1-p_A)^{(1-a)} p_Y^y (1-p_Y)^{(1-y)}[\frac{1}{2\sigma^2}[-tr(W_A\Sigma' W_A^T) - V'^T W_A^T W_A V' \\ &+ 2(a-\mu_A)^T W_A V' - (a-\mu_A)^T(a-\mu_A)] - \frac{1}{2}\log 2\pi\sigma^2\big].\end{aligned}$$

The term (B) is expressed as

$$\begin{aligned}E_{q(u,a,y|x)}[\log \frac{q(u,a,y|x)}{p(u)}] &= E_{q(a|x)q(y|a,x)q(u|a,x,y)}[\log \frac{q(u,a,y|x)}{p(u)}] \\ &= E_{q(a|x)q(y|a,x)}[KL(q(u|a,x,y)||p(u)] \\ &= E_{q(a|x)q(y|a,x)}\big[\frac{1}{2}(-\log det\Sigma' + V'^T V' + tr(\Sigma') - q)\big] \\ &= \sum_{a\in 0,1}\sum_{y\in 0,1} p_A^a(1-p_A)^{(1-a)} p_Y^y (1-p_Y)^{(1-y)}\big[\frac{1}{2}(-\log det\Sigma' + V'^T V' + tr(\Sigma') - q)\big],\end{aligned}$$

where q is the dimension of the covariance matrix.
The term (C) is expressed as

$$\log q(a^*|x^*) + \log q(y^*|a^*,x^*) = a^*\log p_A + (1-a^*)\log(1-p_A) + y^*\log p_Y + (1-y^*)\log(1-p_Y).$$

Finding stationary points Similar to DCEVAE, we take the derivative on Eq. 4 with respect to Σ_0 and Σ_1 for finding stationary point Σ_0^* and Σ_1^*.

$$\frac{\partial \mathcal{L}}{\partial \Sigma_0} = \sum_{i=1}^{n_0} p_A p_Y^{y_i^*} (1-p_Y)^{(1-y_i^*)} \frac{1}{2} (\frac{1}{\sigma^2} W_X^T W_X + \frac{1}{\sigma^2} W_A^T W_A + \frac{1}{\sigma^2} W_{Y_0}^T W_{Y_0} - \Sigma_0^{-T} + I) = 0,$$

$$\Sigma_0^* = \{\frac{1}{\sigma_x^2} diag(W_X^T W_X) + \frac{1}{\sigma^2} diag(W_A^T W_A) + \frac{1}{\sigma^2} diag(W_{Y_0}^T W_{Y_0}) + I\}^{-1},$$

where n_0 is the number of samples with intervention $a = 0$.
Without loss of generality, Σ_1^* is expressed as

$$\Sigma_1^* = \{\frac{1}{\sigma^2} diag(W_X^T W_X) + \frac{1}{\sigma^2} diag(W_A^T W_A) + \frac{1}{\sigma^2} diag(W_{Y_1}^T W_{Y_1}) + I\}^{-1}.$$

1.3 mCEVAE

Network structure and ELBO

$$\begin{aligned}
p(x|u) &= \mathcal{N}(W_X u + \mu_X, \sigma^2 I), \\
p(y|a,u) &= \mathcal{N}(a(W_{Y_1} u + \mu_{Y_1}) + (1-a)(W_{Y_0} u + \mu_{Y_0}), \sigma^2 I), \\
q(u|a,x,y) &= \mathcal{N}(aV_1([x,y] - \mu_1) + (1-a)V_0([x,y] - \mu_0), a\Sigma_1 + (1-a)\Sigma_0), \\
p(u) &= \mathcal{N}(0, I),
\end{aligned}$$

where Σ_0 and Σ_1 are the covaraince structure of each encoder with the intervention $a = 0$ and $a = 1$. To train the mCEVAE, we should optimize the following loss function.

$$\begin{aligned}
\min_{W,V,\Sigma} \mathcal{L} = &- \underbrace{E_{q(u|a,x,y)}[\log p(x|u) + \log p(y|a,u)]}_{\text{A}} + \underbrace{E_{q(u|a,x,y)}[\log \frac{q(u|a,x,y)}{p(u)}]}_{\text{B}} \\
&+ \lambda_{\text{MMD}} \underbrace{\text{MMD}(q(u)||p(u))}_{\text{C}} + \lambda_{\text{MMD}_A} \underbrace{\sum_{a \in 0,1} \text{MMD}(q(u|a)||p(u))}_{\text{D}},
\end{aligned} \tag{5}$$

where MMD is a Maximum Mean Discrepancy.

Closed form with Linear VAE The term (A) and (B) are computed in a similar way to DCEVAE and CEVAE. Note that $W_X u \sim \mathcal{N}(W_X V', W_X \Sigma' W_X^T)$, $V' = aV_1([x,y] - \mu_1) + (1-a)V_0([x,y] - \mu_0)$, and $\Sigma' = a\Sigma_1 + (1-a)\Sigma_0$. Then, we compute $E_{q(u|a,x,y)}[\log p(x|u)]$ as

$$\begin{aligned}
E_{q(u|a,x,y)}[\log p(x|u)] = \frac{1}{2\sigma^2} [&-tr(W_X \Sigma' W_X^T) - V'^T W_X^T W_X V' \\
&+ 2(x - \mu_X)^T W_X V' - (x - \mu_X)^T (x - \mu_X)] - \frac{d}{2} \log 2\pi\sigma^2],
\end{aligned}$$

$$\begin{aligned}
E_{q(u|a,x,y)}[\log p(y|u)] = \frac{1}{2\sigma^2} [&-tr(W_{Y_a} \Sigma' W_{Y_a}^T) - V'^T W_{Y_a}^T W_{Y_a} V' \\
&+ 2(y - \mu_{Y_a})^T W_{Y_a} V' - (y - \mu_{Y_a})^T (y - \mu_{Y_a})] - \frac{1}{2} \log 2\pi\sigma^2.
\end{aligned}$$

The term (B) is expressed as

$$E_{q(u|a,x,y)}[\log \frac{q(u|a,x,y)}{p(u)}] = \frac{1}{2}(-\log det\Sigma' + V'^T V' + tr(\Sigma') - q),$$

where q is the dimension of the covariance matrix.
The term (C) is expressed as

$$\text{MMD}(q(u)||p(u)) = \underbrace{E_{v \sim p(u), w \sim p(u)}[k(v,w)]}_{\text{C1}} - 2 \underbrace{E_{v \sim q(u), w \sim p(u)}[k(v,w)]}_{\text{C2}} + \underbrace{E_{v \sim q(u), w \sim q(u)}[k(v,w)]}_{\text{C3}},$$

where $k(x,y) = \exp(\frac{||x-y||^2}{2\gamma^2})$, the gaussian kernel. We follow Rustamov [16] to generalize (C) with a multivariate gaussian distribution. The term (C2) is expressed as

$$
\begin{aligned}
&E_{v\sim p(u),w\sim q(u)}[k(u,v)] = \int_{R^q}\int_{R^q} \big(\frac{1}{n}\sum_{i=1}^{n}(2\pi)^{-\frac{q}{2}}|\Sigma_i|^{-\frac{1}{2}}e^{-\frac{1}{2}(w-\mu_i)^T|\Sigma_i|^{-1}(w-\mu_i)}\big)e^{-\frac{(u-v)^T(u-v)}{2\gamma^2}}(2\pi)^{-\frac{q}{2}}e^{-\frac{1}{2}v^Tv}dudv \\
&= \frac{1}{n}\sum_{i=1}^{n}\int_{R^q}\int_{R^q}\big\{(2\pi)^{-\frac{q}{2}}|\Sigma_i|^{-\frac{1}{2}}e^{-\frac{1}{2}(w-\mu_i)^T|\Sigma_i|^{-1}(w-\mu_i)}\big\}e^{-\frac{1}{2}(u-v)^T|\gamma^2 I|^{-1}(u-v)}(2\pi)^{-\frac{q}{2}}e^{-\frac{1}{2}v^Tv}dudv \\
&= (2\pi)^{\frac{q}{2}}\frac{1}{n}\sum_{i=1}^{n}\int_{R^q}\int_{R^q}\big\{(2\pi)^{-\frac{q}{2}}|\Sigma_i|^{-\frac{1}{2}}e^{-\frac{1}{2}(w-\mu_i)^T|\Sigma_i|^{-1}(w-\mu_i)}\big\}\big\{(2\pi)^{-\frac{q}{2}}e^{-\frac{1}{2}(u-v)^T|\gamma^2 I|^{-1}(u-v)}\big\}\big\{(2\pi)^{-\frac{q}{2}}e^{-\frac{1}{2}v^Tv}\big\}dudv,
\end{aligned}
\tag{6}
$$

where $\mu_i = a_i V_1([x_i,y_i]-\mu_1) + (1-a_i)V_0([x_i,y_i]-\mu_0)$ and $\Sigma_i = a_i\Sigma_1 + (1-a_i)\Sigma_0$. The integral in Eq. 6 gives the probability density function (PDF) of $W = A+B+C$, where $A\sim\mathcal{N}(0,\Sigma)$, $B\sim\mathcal{N}(0,\gamma^2 I)$, and $C\sim\mathcal{N}(0,I)$. Thus, the integral can be represented by the PDF of $W\sim\mathcal{N}(0,\Sigma+\gamma^2 I+I)$. Plugging in the multiplier in front of the integral, and replacing $w=\mu$, then we get

$$
E_{v\sim p(u),w\sim q(u)}[k(u,v)] = \frac{1}{n}\sum_{i=1}^{n}(2\pi)^{-\frac{q}{2}}|\Sigma_i+\gamma^2 I+I|^{-1}e^{-\frac{1}{2}\mu_i^T|\Sigma_i+\gamma^2 I+I|^{-1}\mu_i}.
$$

Similar to (C2), we compute the term (C1) and (C3) as

$$
\begin{aligned}
E_{v\sim p(u),w\sim p(u)}[k(u,v)] &= (2\pi)^{-\frac{q}{2}}|\gamma^2 I+2I|^{-1}, \\
E_{v\sim q(u),w\sim q(u)}[k(u,v)] &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(2\pi)^{-\frac{q}{2}}|\Sigma_i+\gamma^2 I+\Sigma_j|^{-1}e^{-\frac{1}{2}\mu_i^T|\Sigma_i+\gamma^2 I+\Sigma_j|^{-1}\mu_j}.
\end{aligned}
$$

The term (C) is expressed as

$$
\begin{aligned}
MMD(q(u)||p(u)) &= (2\pi)^{-\frac{q}{2}}|\gamma^2 I+2I|^{-1} + \frac{1}{n}\sum_{i=1}^{n}(2\pi)^{-\frac{q}{2}}|\Sigma_i+\gamma^2 I+I|^{-1}e^{-\frac{1}{2}\mu_i^T|\Sigma_i+\gamma^2 I+I|^{-1}\mu_i}, \\
&+ \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(2\pi)^{-\frac{q}{2}}|\Sigma_i+\gamma^2 I+\Sigma_j|^{-1}e^{-\frac{1}{2}\mu_i^T|\Sigma_i+\gamma^2 I+\Sigma_j|^{-1}\mu_j}.
\end{aligned}
\tag{7}
$$

The term (D) is also expressed as a closed form equation similar to the term (C).

$$
\begin{aligned}
MMD(q(u|a=0)||p(u)) &= (2\pi)^{-\frac{q}{2}}|\gamma^2 I+2I|^{-1} + \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{a_i=0}(2\pi)^{-\frac{q}{2}}|\Sigma_i+\gamma^2 I+I|^{-1}e^{-\frac{1}{2}\mu_i^T|\Sigma_i+\gamma^2 I+I|^{-1}\mu_i} \\
&+ \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{1}_{a_i=0}\mathbb{1}_{a_j=0}(2\pi)^{-\frac{q}{2}}|\Sigma_i+\gamma^2 I+\Sigma_j|^{-1}e^{-\frac{1}{2}\mu_i^T|\Sigma_i+\gamma^2 I+\Sigma_j|^{-1}\mu_j}, \\
MMD(q(u|a=1)||p(u)) &= (2\pi)^{-\frac{q}{2}}|\gamma^2 I+2I|^{-1} + \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{a_i=1}(2\pi)^{-\frac{q}{2}}|\Sigma_i+\gamma^2 I+I|^{-1}e^{-\frac{1}{2}\mu_i^T|\Sigma_i+\gamma^2 I+I|^{-1}\mu_i} \\
&+ \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{1}_{a_i=1}\mathbb{1}_{a_j=1}(2\pi)^{-\frac{q}{2}}|\Sigma_i+\gamma^2 I+\Sigma_j|^{-1}e^{-\frac{1}{2}\mu_i^T|\Sigma_i+\gamma^2 I+\Sigma_j|^{-1}\mu_j}.
\end{aligned}
\tag{8}
$$

Finding stationary points We take the derivative with respect to Σ on ELBO similar to DCEVAE and CEVAE. First, the derivative Eq. 7 with respect to Σ_0 is given as

$$
\begin{aligned}
&\frac{\partial MMD(q(u)||p(u))}{\partial\Sigma_0} = \frac{\partial}{\partial\Sigma_0}\big((2\pi)^{-\frac{q}{2}}|\Sigma_0+\gamma^2 I+I|^{-1}e^{-\frac{1}{2}\mu_0^T|\Sigma_0+\gamma^2 I+I|^{-1}\mu_0}\big) \\
&+ \frac{1}{n^2}(2\pi)^{-\frac{q}{2}}\frac{\partial}{\partial\Sigma_0}\big(2n_0n_1|\Sigma_0+\gamma^2 I+\Sigma_1|^{-1}e^{-\frac{1}{2}\mu_0^T|\Sigma_0+\gamma^2 I+\Sigma_1|^{-1}\mu_1} + n_0^2|2\Sigma_0+\gamma^2 I|^{-1}e^{-\frac{1}{2}\mu_0^T|2\Sigma_0+\gamma^2 I|^{-1}\mu_0}\big),
\end{aligned}
$$

where n_0 and n_1 are the number of samples with intervention $a=0$ and $a=1$. $n=n_0+n_1$ is the number of whole samples. Also, derivation on Eq. 8 with respect to Σ_0 is expressed as

$$
\begin{aligned}
&\frac{\partial MMD(q(u|a=0)||p(u))}{\partial\Sigma_0} = \frac{\partial}{\partial\Sigma_0}\big((2\pi)^{-\frac{q}{2}}|\Sigma_0+\gamma^2 I+I|^{-1}e^{-\frac{1}{2}\mu_1^T|\Sigma_0+\gamma^2 I+I|^{-1}\mu_1}\big) \\
&+ \frac{1}{n^2}(2\pi)^{-\frac{q}{2}}\frac{\partial}{\partial\Sigma_0}\big(n_1^2|2\Sigma_0+\gamma^2 I|^{-1}e^{-\frac{1}{2}\mu_1^T|2\Sigma_0+\gamma^2 I|^{-1}\mu_0}\big).
\end{aligned}
$$

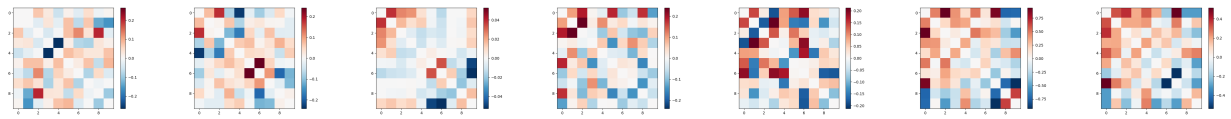
Then, the stationary points Σ_0 satisfies the following equation.

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \Sigma_0} &= \sum_{i=1}^{n} \frac{1}{2}\left(\frac{1}{\sigma^2} W_X^T W_X + \frac{1}{\sigma^2} W_{Y_0}^T W_{Y_0} - \Sigma_0^{-T} + I\right) \\ &+ \lambda_{MMD} \frac{\partial MMD(q(u|a=1)||p(u))}{\partial \Sigma_0} + \lambda_{MMD_A} \frac{\partial MMD(q(u|a=0)||p(u))}{\partial \Sigma_0} = 0.\end{aligned} \tag{9}$$

We cannot derive the closed form solution on Σ_0. Without loss of generality, $\frac{\partial \mathcal{L}}{\partial \Sigma_1}$ can be expressed as similar as Eq. 9.

1.4 Covariance Structure Comparison

The sections from 1.1 to 1.3 show the covariance structure of DCEVAE, CEVAE, and mCEVAE. In DCEVAE, the stationary point Σ^* is expressed as $\left\{\frac{1}{1+\beta}\left(\frac{1}{\sigma^2} W_r^T M_r^T W_r + \frac{1}{\sigma^2} W_d^T M_d^T W_d + \frac{1}{\sigma^2} diag\left(W_y^T W_y\right) + I + \beta \bar{\Sigma}^{-T}\right)\right\}^{-1}$. The term $\bar{\Sigma}$ enforces Σ^* to have two block diagonal matrices, which means Σ^* is disentangled. On the other hand, in CEVAE, Σ_0 is expressed as $\{\frac{1}{\sigma_x^2} diag(W_X^T W_X) + \frac{1}{\sigma^2} diag(W_A^T W_A) + \frac{1}{\sigma^2} diag(W_{Y_0}^T W_{Y_0}) + I\}^{-1}$. There is no term forces the stationary points Σ_0^* and Σ_1^* to be disentangled. We cannot derive the closed form of Σ_0 and Σ_1 in mCEVAE, but we empirically visualize the covariance structures in Figure 1f and Figure 1g. Figure 1 compares the covariance structure of DCEVAE, CEVAE, and mCEVAE. There are no block diagonals in CEVAE and mCEVAE, so it cannot disentangle the latent variables, unlike DCEVAE.



(a) $\beta = 0$ (b) $\beta = 3.2$ (c) $\beta = 1000$ (d) CEVAE Σ_0 (e) CEVAE Σ_1 (f) mCEVAE Σ_0 (g) mCEVAE Σ_1

Figure 1: (a,b,c) The covariance matrix of sampled latent values from DCEVAE with different $\beta = 0$, 3.2, and 1000. (c,d,e,f) The sampled covariance matrix Σ_0 and Σ_1 from CEVAE and mCEVAE.

2 Preliminary for Causal Inference

Since we introduce a VAE alternative interpreting a causal inference with an intervention, we review causal inference emphasis on interventions. A causal model builds the causality structure between variables with a graphical representation, a.k.a. *Causal Graph* ($\mathcal{G}$). Each node corresponds to a variable (V_i), and each edge indicates a causality that specifies the causal effect in a deterministic function (f_{V_i}) from causal variables, which is a collection of parent nodes, to an effect variable, which is a destination node. Additionally, this paper focuses on the exogenous uncertainties (U) because this paper ultimately aims at discerning the effect of U between V. The definition of a causal model is as follows:

Definition .1 *(Probabilistic Causal Model). A probabilistic causal model [15] $\mathcal{M}$ is a tuple $\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathbf{F}, \mathbf{P}(\mathbf{u}) \rangle$, where:*

1. $\mathbf{U}$ *is a set of* ***exogenous variables*** *which cannot be observed or experimented on, however, they affect the rest of the model. An example of such exogenous variables is the stochastic noises in an observation dataset.*
2. $P(\mathbf{u})$ *is an arbitrary joint probability distribution defined over* U.
3. $\mathbf{V}$ *is a set of* ***endogenous variables*** *determined to be included in the model, so the complete set of variables in* $\mathcal{M}$ *is* $U \cup V$. *Since we include the stochasticity from* $P(u)$ *in* $\mathcal{M}$*, we can impose a probability distribution of* V *as* $P(v) := P(V = v) = \sum_{\{u|f_V(V,u)=v\}} P(u)$ *under the condition of including* U *in determining* V.
4. $\mathbf{F}$ *is a set of* ***deterministic functions****. We define* $f_{V_i} \in F$ *by* $V_i = f_{V_i}(Pa_{V_i}, U_{V_i})$*. Here,* Pa_{V_i} *is a parent set of* V_i*;* U_{V_i} *is a set of exogenous variables influencing on* V_i*; and* f_{V_i} *is a mapping function from the causes (*Pa_{V_i}*) to the effect (*V_i*).*

The advantage of a causal graph comes from measuring causal effects, which requires *do-calculus* on a submodel [15], an intervention, and a potential response.

Definition .2 *(Submodel). Let* $\mathcal{M}$ *be a probabilistic causal model;* X*, a set of variable in* V*; and* x*, a particular realization of* X*. A submodel* $\mathcal{M}_x$ *of* $\mathcal{M}$ *is the probabilistic causal model* $\mathcal{M}_x =< \mathbf{U}, \mathbf{V}, \mathbf{F}_x, P(\mathbf{u}) >$ *where* $\mathbf{F}_x = \{f_{V_i} : V_i \notin X\} \cup \{X = x\}$.

Definition .3 *(Intervention). Given the definitions up to this point, the effect of action, or intervention,* $do(X = x)$ *on* $\mathcal{M}$ *is given by the submodel* $\mathcal{M}_x$.

Definition .4 *(Potential Response). Assuming* $\mathbf{X}, \mathbf{Y} \subset \mathbf{V}$*, the potential response in the probabistic causal model is the probability distribution of* Y *as* $Y_x = Y_x(u) = P(Y_x = y) = P(Y_x = y|do(X = x))$*, and we simplfy* $P(Y_x = y)$ *as* $P_x(y)$.

When we introduce a set of conditional variables, Z, which is exclusive to X and Y, we can define a conditional potential response as $P(Y_x = y|Z_x = z)$ as $P_x(y|z)$. We clarify that $P(Y = y|do(X = x))$ is the probability of observing $Y = y$ when we intervene $\mathcal{M}$ with $X = x$. On the contrary, $P(Y = y|X = x)$ means the probability of observing $Y = y$ out of the observations with $X = x$.

Pearl introduce counterfactual inference in three steps. This is specified in following Theorem.

Theorem 1 *(Inference on Counterfactual effect [15].) Given model* $\mathcal{M}$*, the conditional probability* $P(Y_x|o)$ *means a counterfactual sentence of "If it were x then Y," given observation o.* $P(Y_x|o)$ *can be evaluated by the following three steps.*

1. ***Abduction*** *Update* $P(u)$ *by the observation o to obtain* $P(u|o)$
2. ***Action*** *Modify* $\mathcal{M}$ *by the action* $do(X = x)$*, where* X *is the antecedent of the counterfactual, to obtain the submodel* $\mathcal{M}_x$.
3. ***Prediction*** *Use the modified model* $< \mathcal{M}_x, P(u|o) >$ *to compute the probability of* Y*, the consequence of the counterfactual.*

3 Loss Derivation, Algorithm, and Complexity

3.1 Evidence Lower Bound ($\mathcal{L}_{ELBO}$)

The detail derivation of the variational lower bound of DCEVAE is described as follows:

$$
\begin{aligned}
\log p_\theta(x_d, x_r, y|a) &\overset{(a)}{=} \log \mathbb{E}_{q_\phi(u_d|a,x_d,y)q_\phi(u_r|a,x_r,y)} \frac{p_\theta(x_d, x_r, y, u_d, u_r|a)}{q_\phi(u_d|a, x_d, y)q_\phi(u_r|a, x_r, y)} \\
&\overset{(b)}{\geq} \mathbb{E}_{q_\phi(u_d|a,x_d,y)q_\phi(u_r|a,x_r,y)} \log \frac{p_\theta(x_d, x_r, y, u_d, u_r|a)}{q_\phi(u_d|a, x_d, y)q_\phi(u_r|a, x_r, y)} \\
&\overset{(c)}{=} \mathbb{E}_{q_\phi(u_d|a,x_d,y)q_\phi(u_r|a,x_r,y)} \log \frac{p_\theta(y|a, u_d, u_r)p_\theta(x_d|a, u_d)p_\theta(x_r|u_r)p(u_d, u_r)}{q_\phi(u_d|a, x_d)q_\phi(u_r|a, x_r)} \\
&= \mathbb{E}_{q_\phi(u_d|a,x_d,y)q_\phi(u_r|a,x_r,y)}[\log p_\theta(y|a, u_d, u_r)] + \mathbb{E}_{q_\phi(u_d|a,x_d)}[\log p_\theta(x_d|a, u_d)] \\
&+ \mathbb{E}_{q_\phi(u_r|a,x_r,y)}[\log p_\theta(x_r|u_r)] + KL(q_\phi(u_d|a, x_d, y)||p(u_d)) + KL(q_\phi(u_r|a, x_r, y)||p(u_r)) \\
&:= \mathcal{L}_{ELBO}.
\end{aligned}
$$

Here, (a) the law of total probability, (b) Jensen's inequality, and (c) factorization under our assumption that $x_d|a.u_d$ is independent to u_r, $x_r|u_r$ is independent to u_d and a, and the generation of x_d and y are conditioned on a.

3.2 Total Correlation ($\mathcal{L}_{TC}$)

For a given set of n random variables $X_1, X_2, ..., X_n$, the total correlation $TC(X_1, X_2, ..., X_n)$ is defined as the KL divergence from the join distribution $p(X_1, ..., X_n)$ to the independent distribution of $p(X_1)p(X_2)...p(X_n)$,

$$
TC(X_1, ..., X_n) = KL[p(X_1, ..., X_n)||p(X_1)p(X_2)...p(X_n)].
$$

The total correlation (TC) is derived based on the *density ratio trick* [17]. Here, we suppose that N data points are drawn from the distribution $q(a, u_d, u_r)$ and assigned a label $y = +1$. The remaining N data points are drawn from distribution $q(a, u_r)q(u_d)$ and assigned label $y = -1$. By this construction, we can write the probabilities $q(a, u_d, u_r)$ and $q(a, u_r)q(u_d)$ in a conditional form as $p(a, u_d, u_r|y = 1)$ and $p(a, u_d, u_r|y = -1)$. The detail derivation for the approximation form of TC loss is as follows:

$$
\begin{aligned}
\mathcal{L}_{TC} &= KL\left(q(a, u_d, u_r) \| q(a, u_r)q(u_d)\right) \\
&\overset{(a)}{=} E_{q(a,u_d,u_r)}\left[\log \frac{q(a, u_d, u_r)}{q(a, u_r)q(u_d)}\right] \\
&\overset{(b)}{=} \mathbb{E}_{q(a,u_d,u_r)}\left[\log \frac{p(a, u_d, u_r|y = 1)}{p(a, u_d, u_r|y = -1)}\right] \\
&\overset{(c)}{=} \mathbb{E}_{q(a,u_d,u_r)}\left[\log \frac{p(y = 1|a, u_d, u_r)p(a, u_d, u_r)}{p(y = 1)} \Big/ \frac{p(y = -1|a, u_d, u_r)p(a, u_d, u_r)}{p(y = -1)}\right] \\
&\overset{(d)}{=} \mathbb{E}_{q(a,u_d,u_r)}\left[\log \frac{p(y = 1|a, u_d, u_r)}{p(y = -1|a, u_d, u_r)}\right] \\
&= \mathbb{E}_{q(a,u_d,u_r)}\left[\log \frac{p(y = 1|a, u_d, u_r)}{1 - p(y = 1|a, u_d, u_r)}\right] \approx \mathbb{E}_{q(a,u_d,u_r)}\left[\log \frac{D_\psi(a, u_d, u_r)}{1 - D_\psi(a, u_d, u_r)}\right].
\end{aligned}
$$

Here, (a) the definition of KL divergence, (b) the aforementioned assumption, (c) the Bayes' theorem, and (d) the assumption that $p(y = 1) = p(y = -1)$.

3.3 Whole Algorithm

Algorithm 1: The learning algorithm of the DCEVAE

Input: observations $\left(a^{(i)}, x_d^{(i)}, x_r^{(i)}, y^{(i)}\right)_{i=1}^{N}$, batch size m, DCEVAE and Discriminator optimizers are g and g_D
Output: DCEVAE decoder parameters $\theta = (\theta_d, \theta_r, \theta_y)$, encoder parameters $\phi = (\phi_d, \phi_r)$, discriminator parameter ψ

while *objective is not converged* **do**
  for *DCEVAE training iterations* **do**
    Randomly selected batch $\left(a^{(i)}, x_d^{(i)}, x_r^{(i)}, y^{(i)}\right)_{i\in\mathcal{B}}$ of size m
    Sample $u_d^{(i)} \sim q_{\phi_d}(u_d|a^{(i)}, x_d^{(i)}, y^{(i)})$ and $u_r^{(i)} \sim q_{\phi_r}(u_r|a^{(i)}, x_r^{(i)}, y^{(i)})$ for $\forall i \in \mathcal{B}$
    $\theta \leftarrow g\left(\nabla_\theta \frac{1}{m}\sum_{i\in\mathcal{B}}\left[\log \frac{p_{\theta_y}(y^{(i)}|a^{(i)},u_d^{(i)},u_r^{(i)})p_{\theta_d}(x_d^{(i)}|a^{(i)},u_d^{(i)})p_{\theta_r}(x_r^{(i)}|u_r^{(i)})}{q_{\phi_d}(u_d^{(i)}|a^{(i)},x_d^{(i)},y^{(i)})q_{\phi_r}(u_r^{(i)}|a^{(i)},x_r^{(i)},y^{(i)})} - \beta_{tc}\log\frac{D_\psi(a^{(i)},u_d^{(i)},u_r^{(i)})}{1-D_\psi(a^{(i)},u_d^{(i)},u_r^{(i)})}\right)$
    Randomly selected batch $\left(a^{(i)}, x_d^{(i)}, x_r^{(i)}, y^{(i)}\right)_{i\in\mathcal{B}}$ of size m
    Sample ${u'_d}^{(i)} \sim q_{\phi_d}(u_d|a^{(i)}, x_d^{(i)}, y^{(i)})$ and ${u'_r}^{(i)} \sim q_{\phi_r}(u_r|a^{(i)}, x_r^{(i)}, y^{(i)})$ for $\forall i \in \mathcal{B}'$
    for z *in* $\left[(a, u'_r), u'_d\right]$ **do**
      $\pi \leftarrow$ random permutation on $\{1, ..., m\}$
      $(z_{perm}^{(i)})_{i=1}^m \leftarrow (z_{perm}^{\pi(i)})_{i=1}^m$
    end for
    $\theta \leftarrow g_D\left(\nabla_\psi \frac{1}{2m}\left[\sum_{i\in\mathcal{B}}\log\left(D_\psi([a^{(i)}, u_d^{(i)}, u_r^{(i)}])\right) + \sum_{i\in\mathcal{B}}\log\left(1 - D_\psi([a_{perm}^{(i)}, u_{dperm}^{(i)}, u_{rperm}^{(i)}])\right)\right]\right)$
  end for
end while

3.4 Complexity Analysis

We compare the time complexity for baselines and our model in the causal estimation tasks. The time complexity of each model is expressed in the following table.

Model	Training	Testing
CEVAE	$\mathcal{O}(ND_x + N(D_x + D_a)D_y + NDU + NU)$	$\mathcal{O}(ND_x + N(D_x + D_a)D_y + NDU)$
mCEVAE	$\mathcal{O}(NDU + S^2U)$	$\mathcal{O}(NDU)$
Causal GAN	$\mathcal{O}(ND(Z + C) + NDH)$	$\mathcal{O}(ND(Z + C))$
DCEVAE (ours)	$\mathcal{O}(NDU + NU + N(D_a + U)H)$	$\mathcal{O}(NDU)$

Table 1: The time complexity for baselines and DCEVAE.

In Table 1, N is the numbers of the data instances; $D = D_x + D_a + D_y$ is the sum of the dimension of each feature vector x, a, and y; U is the dimension of the latent vectors; H is the dimension of hidden layer in discriminator D_ψ in DCEVAE; S is the number of samples used for computing MMD in mCEVAE; Z is the dimension of noise for Generator in Causal GAN; and C is the value depends on the assumed causal graph structure. The more complex causal graph is, the higher value C has.

4 Experimental Settings

4.1 Dataset Detail

Causal Estimation and Fair Classification We use the UCI Adult income dataset [1] for causal estimation and fair classification tasks. The dataset contains 65,123 samples with 11 variables for causal estimations. We binarize each attribute, and we only use seven variables following CFGAN [18]. We categorized attributes based on the causal graph which is usually assumed for UCI Adult [19, 18].

- a: *gender*
- y: *income*
- x_d: *marital status*, *education level*, *occupation*, *hours per week*, *workclass*, and *relationship*
- x_r: *race*, *age*, and *native country*.

Counterfactual Image Generation We use the CelebA dataset [12] for the counterfactual image generation. The dataset contains 202,599 face images and 40 binary annotations per image. Following the setting in CausalGAN, we intervene on a variable, *Mustache*, which is altered in the counterfactual image. Then attributes are categorized as follows:

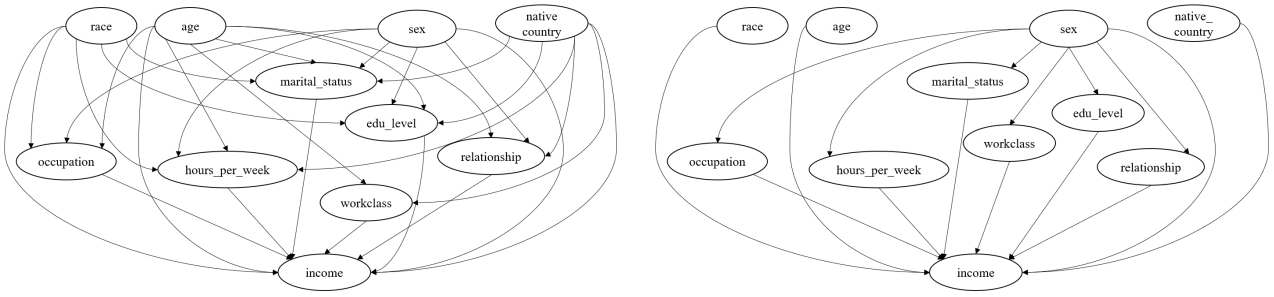
- a: *Mustache*
- y: *Image*
- x_r: *Young*, *Male*, *Eye glasses*, *Bald*, *Smiling*, *Wearing Lipstick*, *Mouth Slightly Open*, and *Narrow Eyes*.

It should be noted that CausalGAN used *Mustache* without any intervention descendants. We expanded the candidate variables of intervention with descendants, *Smiling*. Then attributes are categorized as follows:

- a: *Mustache*
- y: *Image*
- x_d: *Mouth Slightly Open* and *Narrow Eyes*
- x_r: *Young*, *Male*, *Eye glasses*, *Bald*, *Smiling*, *Wearing Lipstick*, and *Mustache*.

4.2 The Causal Graph Structure

Causal Estimation and Fair Classification Figure 2a shows the causal graph used in CausalGAN, CF-AN [10], and CFGAN for UCI Adult dataset. Figure 2b shows the incomplete graph structure used in CFGAN-IC. An incomplete graph specifies the descendant attributes of intervention attribute, *sex*. DCEVAE also specifies the descendants of intervention variables, so we used the incomplete causal graph structure on the baseline for a fair comparison.



(a) Complete Causal Graph

(b) Incomplete Causal Graph

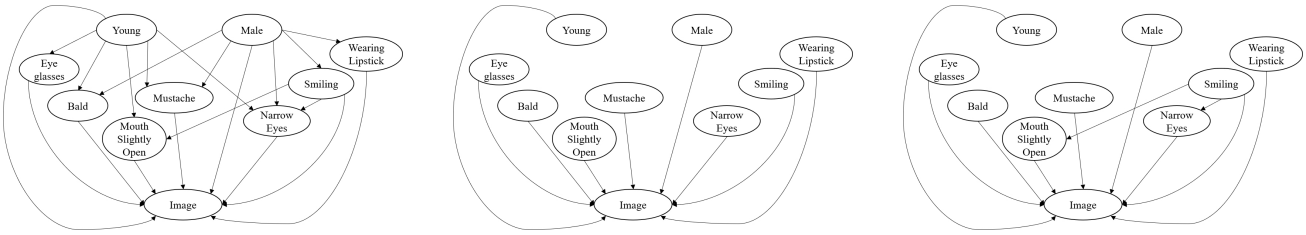
Figure 2: The causal graph structure for UCI Adult dataset, (a) complete and (b) incomplete.

Counterfactual Image Generation Figure 3a shows the causal graph used in CausalGAN (DCGAN and BEGAN). Figure 3b and 3c show the incomplete graph structure for intervention *Mustache* and *Smiling* used in CFGAN-IC.

4.3 Evaluation Metrics

Causal Estimation and Fair Classification We evaluate the performance on the fairness task with following metrics:

- **Causal Effect Evaluation.** *Total effect* is $TE(a_1, a_0) = P(y_{a_1}) - P(y_{a_0})$ from two interventions of a_0 and a_1 on the intervention, a. While the total effect assumes the total change of interventions, we may consider a partial change by keeping, o, the subset of x, so *Counterfactual effect* becomes $CE(a_1, a_0|o) = P(y_{a_1}|o) - P(y_{a_0}|o)$.



(a) Complete Causal Graph (b) Incomplete Causal Graph for *Mustache* (c) Incomplete Causal Graph for *Smiling*

Figure 3: (a) The causal graph used in CausalGAN [9] for CelebA dataset. (b,c) The incomplete graph structure for intervention on *Mustache* and *Smiling*.

- **Data Utility Evaluation.** We evaluate the counterfactual data utility by testing a real dataset with trained classifiers on the counterfactual dataset. Support Vector Machine (SVM) and Logistic Regression (LR) are trained based on the generated dataset. Then, we test the original dataset with the well-trained classifiers. Also, we compute the chi-square distance (χ^2) [3] which indicates the similarity between the generated dataset and the real dataset. Smaller χ^2 indicates better data utility.

Counterfactual Image Generation We evaluate the quality of counterfactual generated images by the metrics used in MaskGAN [11]:

- **Semantic-level Evaluation.** We examined the classification accuracy of rest variables, x_r, on counterfactually generated images to evaluate whether our generated images preserve the information of x_r. We trained binary facial label classifiers for corresponding sepecific labels on the CelebA dataset by using the ResNet-18 [6] architecture. The classifier ignored the minority class *Mustache* $= 1$, so we balanced the train and test dataset 50:50 for attribute *Mustache*. Also, classification accuracy of *Mustache* on counterfactual images is unreliable because the classifier only trained on the real image includes *Mustache* with *Male*, not *Mustache* with *Female*.
- **Distribution-level Evaluation.** We measure the quality and the diversity of generated images with the Frechet Inception Distance (FID) [2] which is most often used for evaluating samples of Generative Adversarial Networks. FID is calculated by computing the Frechet distance between two Gaussian fitted representations. We used PyTorch [14] to compute FID score.[1]
- **Identity Preserving Evaluation.** We conducted the face verification experiment by ArcFace [4] to evaluate the identity preservation ability. Arcface is the pre-trained model with LFW Face dataset [7]. We used PyTorch to compute IP score.[2] In the experimental setting, we selected 400 pairs of a face form testing set in CelebA, and each pair contains an original face and its counterfactual face. Besides, each face was resized to 112×112 in the testing stage. Both images are evaluated as identical when the output of ArcFace, the probability indicates how both images are the same, over the threshold.

4.4 Network Architecture and Optimization

We used PyTorch for the implementation of our model. We repeat all experiments for 5 times. The network architecture and the optimization detail for each dataset, UCI Adult and CelebA, are described in the following section.

UCI Adult Dataset We train our model for 500 epochs using Adam optimizer with *betas* $= (0.9, 0.999)$, *eps* $= 10^{-8}$ and initial $lr = 10^{-4}$. We use minibatches of size 64. If the validation loss is not decreased for consecutive 30 epochs, we apply the early stop.

For UCI Adult dataset, Our architecture consists of fully connected layers, Batch Normalization, and ReLU activation. The discriminator consists of fully connected layers and LeakyReLU. The detailed architecture is described in Table 2.

For tuning the hyper-parameters in the loss function Eq. 10, we trained the model with $\beta_{TC} = [0.2, 0.4, 0.8, 1.6, 3.2, 6.4]$, and $\beta_f = [0, 0.1, 0.2, ..., 1.0]$. we use the values $\beta_{tc} = 0.4$, and $\beta_f = 0$. For fair task, we use $\beta_f = 0.2$.

$$\mathcal{L} = \mathbb{E}_{q_\phi(u_d|a,x_d,y)q_\phi(u_r|a,x_r,y)}[\log p_\theta(y|a,u_d,u_r)] + \mathbb{E}_{q_\phi(u_d|a,x_d,y)}[\log p_\theta(x_d|a,u_d)] + \mathbb{E}_{q_\phi(u_r|a,x_r,y)}[\log p_\theta(x_r|u_r)] + KL(q_\phi(u_d|a,x_d,y)||p(u_d)) + KL(q_\phi(u_r|a,x_r,y)||p(u_r)) + \beta_{tc}\mathcal{L}_{TC} + \beta_f\mathcal{L}_f. \quad (10)$$

CelebA Dataset We follow the experimental settings in *Learning latent subspaces in Variational Autoencoder* (CSVAE) [8] for training our model. We train our model for 2300 epochs using Adam optimizer with *betas* $= (0.9, 0.999)$, *eps* $= 10^{-8}$ and initial $lr = 10^{-3/2}$. We use PyTorchs learning rate scheduler MultiStepLR with *milestones* $= \{3^i | i = 0, ..., 6\}$ and *gamma*

[1] https://github.com/mseitzer/pytorch-fid

[2] https://github.com/ronghuaiyang/arcface-pytorch

Encoder q_d (or q_r)	**Decoder p_y**	**Decoder p_d**	**Decoder p_r**	**Discriminator D_ψ**
Input $\in \mathbb{R}^{d_{x_d}\ (or\ d_{x_r})+d_a+d_y}$	Input $\in \mathbb{R}^{d_{u_d}+d_{u_r}+d_a}$	Input $\in \mathbb{R}^{d_{u_d}+d_a}$	Input $\in \mathbb{R}^{d_{u_r}}$	Input $\in \mathbb{R}^{d_{all}}$
FC. d_h - ReLU.	FC. d_h - ReLU.	FC. d_h - ReLU.	FC. d_h - ReLU.	FC. d_h - LeakyReLU.
FC. d_h - ReLU.	FC. d_y	FC. d_{x_d}	FC. d_{x_r}	FC. d_h - LeakyReLU.
FC. d_{u_d} (or d_{u_r})				FC. d_h - LeakyReLU.
				FC. 2

Table 2: Encoder and Decoder architecture for UCI Adult dataset. d_{u_d}, d_{u_r}, d_{x_d}, d_{x_r}, d_a, and d_h indicate the dimension of u_d, u_r, x_d, x_r, a, and hidden dimension. We use $d_h = 100$ for UCI Adult. In the table, FC. is fully connected layer.

$= 0.1^{1/7}$. We use minibatches of size 64. The different setting between our model and CSVAE is that we used early stop if the validation loss is not decreased for consecutive 30 epochs.

Our architecture consists of convolutional layers, Dropout, Batch Normalization, and ReLU activation. The discriminator consists of fully connected layer and LeakyReLU. The detailed architecture is described in Table 3.

Encoder q	**Decoder p_y**	**Decoder p_d**	**Decoder p_r**	**Discriminator D_ψ**
Input $64 \times 64 \times 3$ RGB images	Input $\in \mathbb{R}^{d_{u_d}+d_{u_r}}$	Input $\in \mathbb{R}^{d_{u_d}+d_a}$	Input $\in \mathbb{R}^{d_{u_r}}$	Input $\in \mathbb{R}^{d_{u_d}+d_{u_r}+d_a}$
4×4 conv. 32 - Dropout(0.04) - BN. - ReLU.	FC. $d_{u_d} + d_a$ - ReLU.	FC. d_{u_r} - ReLU.	FC. 256 - ReLU.	FC. 1000 - LeakyReLU.
4×4 conv. 32 - Dropout(0.04) - BN. - ReLU. stride 2	FC. $4 \times 4 \times 64$ - ReLU.	FC. d_{x_d}	FC. d_{x_r}	FC. 1000 - LeakyReLU.
4×4 conv. 32 - Dropout(0.04) - BN. - ReLU.	4×4 upconv. - 64 ReLU.			FC. 1000 - LeakyReLU.
4×4 conv. 32 - Dropout(0.04) - BN. - ReLU.	4×4 upconv. - 32 ReLU.			FC. 1000 - LeakyReLU.
FC. 256 - FC. $2 \times (d_{x_r} + d_{x_d})$	4×4 upconv. - 32 ReLU.			FC. 2
	4×4 upconv. 1. stride 2			

Table 3: Encoder and Decoder architecture for CelebA dataset. d_{u_d}, d_{u_r}, d_{x_d}, d_{x_r}, and d_a indicate the dimension of u_d, u_r, x_d, x_r, and a. We use $(d_{u_d}, d_{u_r}) = (5, 5)$ for *Mustache* and *Smiling*. In the table, conv. is convolutional network, BN. is Batch normalization, and FC. is fully connected layer.

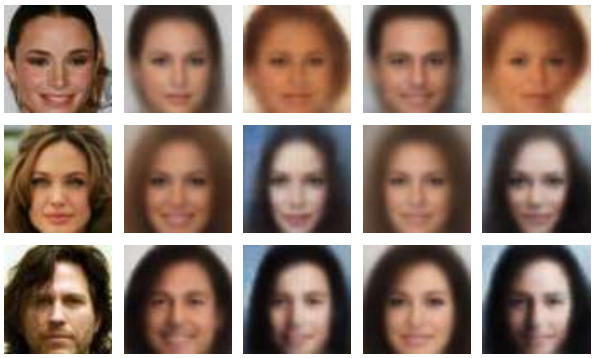
For tuning the hyper-parameters in the loss function Eq. 11, we trained the model with $\beta_{tc} = [0.2, 0.4, 0.8, 1.6, 3.2, 6.4]$. we use the values $\beta_{tc} = 3.2$ for *Mustache* and $\beta_{tc} = 0.4$ for *Smiling*.

$$\begin{aligned}\mathcal{L} = &\mathbb{E}_{q_\phi(u_d|a,y)q_\phi(u_r|a,y)}[\log p_\theta(y|a,u_d,u_r)] + \mathbb{E}_{q_\phi(u_d|a,y)}[\log p_\theta(x_d|a,u_d)] + \mathbb{E}_{q_\phi(u_r|a,y)}[\log p_\theta(x_r|u_r)] \\ &+ KL(q_\phi(u_d|a,y)||p(u_d)) + KL(q_\phi(u_r|a,y)||p(u_r)) + \beta_{tc}\mathcal{L}_{TC}. \end{aligned} \tag{11}$$

5 Additional Experimental Results

5.1 Counterfactual Images Generation from DCEVAE with *Smiling*

Figure 4 shows the real and counterfactual images from DCEVAE, CVAE, CEVAE, and mCEVAE with intervention on *Smiling*. DCEVAE preserves the identity of the original image, but other models fail to preserve originality. CEVAE alters the gender since each decoder is trained on a dataset with different ratios of gender. For example, decoder of *Smiling* $= 0$ is trained with data mostly composed of *Male* images, so when real image is *Female* with *Smiling*, the counterfactual image is endangered to suffer gender alteration. In mCEVAE, the latent value is changed in the overlapping process between latent with $a = 0$ and $a = 1$, so it significantly loses the originality.



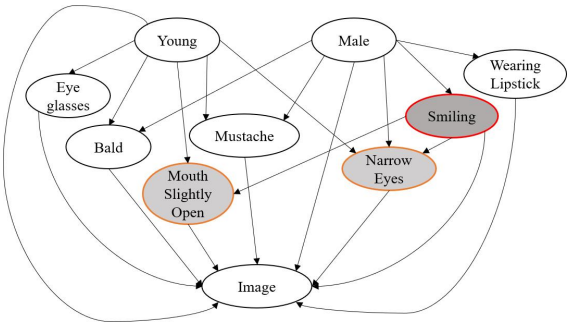


Figure 4: (Left) The pairs of images from VAE intervention on $Smiling$ with CelebA. (Right) The causal graph of CelebA.

Table 4 shows the quantitative results of counterfactual image generation with *Smiling*. For real image, DCEVAE has the best performance on generating counterfactual image reflecting flipped intervention, see *Target* accuracy on *Real*. Also, DCEVAE has the highest IP and the lowest FID among VAE based models. Even compared with GAN based models, DCEVAE maintains x_r well.

	Model	Target	Attribute clssification accuracy (%)					IP	FID
			WL	ML	E	B	Y		
Real	CVAE	$51.26_{\pm 0.81}$	$73.49_{\pm 0.68}$	$72.06_{\pm 2.16}$	$\mathbf{63.96_{\pm 5.39}}$	$86.69_{\pm 4.27}$	$64.40_{\pm 8.64}$	$0.33_{\pm 0.10}$	$169.62_{\pm 11.06}$
	CEVAE	$48.33_{\pm 0.61}$	$73.16_{\pm 0.05}$	$65.96_{\pm 0.24}$	$61.12_{\pm 0.31}$	$88.34_{\pm 0.14}$	$69.63_{\pm 0.15}$	$0.28_{\pm 0.01}$	$163.79_{\pm 1.87}$
	mCEVAE	$50.54_{\pm 1.37}$	$73.22_{\pm 0.03}$	$68.06_{\pm 4.05}$	$56.52_{\pm 3.61}$	$86.68_{\pm 1.43}$	$62.78_{\pm 2.83}$	$0.26_{\pm 0.03}$	$165.13_{\pm 1.33}$
	DCEVAE (ours)	$\mathbf{62.14_{\pm 1.52}}$	$\mathbf{73.55_{\pm 0.07}}$	$\mathbf{76.22_{\pm 0.55}}$	$63.54_{\pm 0.71}$	$\mathbf{89.34_{\pm 0.17}}$	$\mathbf{72.48_{\pm 0.42}}$	$\mathbf{0.34_{\pm 0.01}}$	$\mathbf{162.72_{\pm 1.75}}$
Pair	CWGAN	$54.93_{\pm 0.52}$	$93.30_{\pm 0.59}$	$81.02_{\pm 0.31}$	$84.89_{\pm 0.98}$	$90.92_{\pm 0.31}$	$77.34_{\pm 0.65}$	$0.32_{\pm 0.01}$	$\mathbf{100.85_{\pm 1.58}}$
	DCGAN	$56.83_{\pm 3.93}$	$77.44_{\pm 7.16}$	$61.50_{\pm 3.46}$	$79.70_{\pm 4.07}$	$88.56_{\pm 3.24}$	$66.82_{\pm 4.10}$	$0.08_{\pm 0.02}$	$142.04_{\pm 4.07}$
	BEGAN	$53.48_{\pm 1.79}$	$91.96_{\pm 4.24}$	$68.90_{\pm 8.37}$	$80.93_{\pm 2.22}$	$89.18_{\pm 5.61}$	$71.27_{\pm 7.12}$	$0.30_{\pm 0.35}$	$190.75_{\pm 94.59}$
	DCGAN-IC	$56.37_{\pm 3.55}$	$83.93_{\pm 7.27}$	$65.50_{\pm 0.88}$	$80.00_{\pm 4.19}$	$87.71_{\pm 4.99}$	$66.50_{\pm 4.83}$	$0.06_{\pm 0.01}$	$140.05_{\pm 4.55}$
	BEGAN-IC	$53.97_{\pm 1.58}$	$88.77_{\pm 1.25}$	$62.57_{\pm 1.65}$	$77.73_{\pm 2.56}$	$85.21_{\pm 2.68}$	$65.75_{\pm 1.31}$	$0.06_{\pm 0.01}$	$148.69_{\pm 2.94}$
	CVAE	$56.22_{\pm 0.92}$	$95.20_{\pm 2.49}$	$82.40_{\pm 3.39}$	$76.13_{\pm 8.08}$	$90.10_{\pm 5.74}$	$71.99_{\pm 13.82}$	$0.36_{\pm 0.25}$	$169.62_{\pm 11.06}$
	CEVAE	$50.15_{\pm 0.43}$	$\mathbf{98.96_{\pm 0.12}}$	$\mathbf{90.96_{\pm 0.29}}$	$\mathbf{87.15_{\pm 0.50}}$	$\mathbf{98.78_{\pm 0.22}}$	$\mathbf{93.62_{\pm 0.39}}$	$\mathbf{1.00_{\pm 0.00}}$	$163.79_{\pm 1.87}$
	mCEVAE	$54.99_{\pm 2.05}$	$96.18_{\pm 1.91}$	$72.71_{\pm 8.14}$	$63.46_{\pm 5.82}$	$89.20_{\pm 2.13}$	$67.30_{\pm 5.21}$	$0.12_{\pm 0.03}$	$165.13_{\pm 1.33}$
	DCEVAE (ours)	$\mathbf{63.49_{\pm 1.64}}$	$94.35_{\pm 0.74}$	$90.04_{\pm 1.26}$	$78.20_{\pm 0.76}$	$95.89_{\pm 0.47}$	$89.28_{\pm 0.38}$	$0.92_{\pm 0.01}$	$162.72_{\pm 1.75}$

Table 4: Attribute classification accuracy, FID Score (FID), and Identity Preserving (IP) score (threshold$= 0.2$) of images intervened on $a =$ *Smiling*. Here, WL:*Wearing Lipstick*, ML: *Male*, E: *Eyeglasses*, B: *Bald*, Y: *Young*. The numbers in bold indicate the best performance among baselines, and the underlined numbers indicate that the second best performance among baselines.

5.2 Counterfactual Images Generation from GAN

Baselines using GAN, Conditional GAN with Wasserstein distance, CausalGAN, and CausalGAN-IC, generated counterfactual images for *Mustache*, in Figure 5, and *Smiling*, in Figure 6. For Generating images from GAN, pairs of fake images have different intervention values, 0 and 1, with the same noise. Figure 5 and 6 show that Causal GAN and Causal GAN-IC generate

the low quality of images. On the contrary, CWGAN, causal BEGAN, and causal BEGAN-IC generate clear images compared to VAE based models.

Figure 5 shows that some counterfactual images from Causal GAN, Causal BEGAN, Causal GAN-IC, and Causal BEGAN-IC fail to make *Mustache* or alter the gender. CWGAN hardly generate *Mustache* for *Female* since CWGAN only leans the conditional distribution, not intervention distribution.

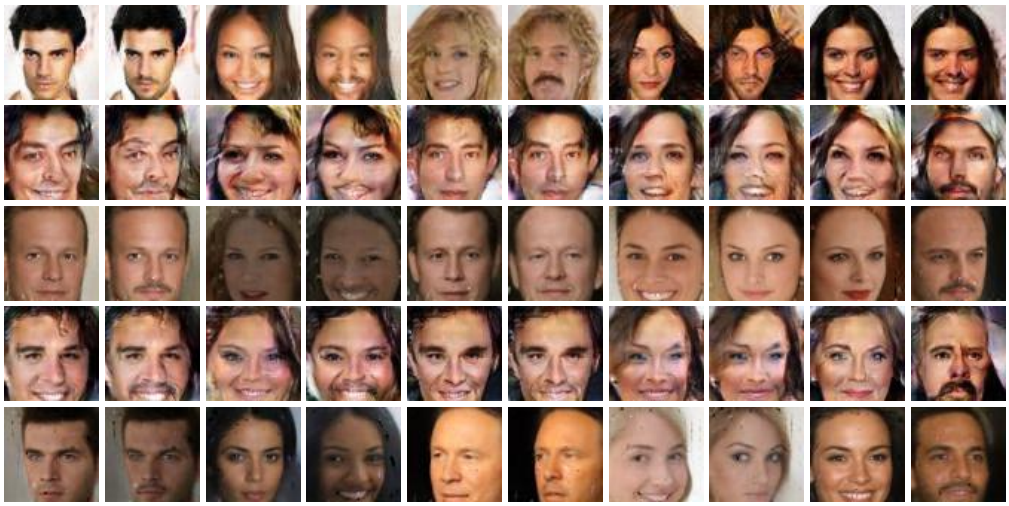


Figure 5: The pairs of images from GAN intervention on $Smiling$. The odd (even) columns show the images intervention on *Mustache*$= 0$ (*Mustache*$= 1$). The images from row 1 to row 5 are Conditional GAN with Wasserstein distance, Causal GAN, Causal BEGAN, Causal GAN-IC, and Causal BEGAN-IC. We report successful images from column 1 to 4, and failure images from column 5 to 10.

Figure 6 shows that GAN based models generate some failure images which lose the identity of its paired images. For example, models except for CWGAN in Figure 6, gender attribute is altered with flipped intervention on failure images. CWGAN does not lose the identity, but it fails to flipped the intervention.

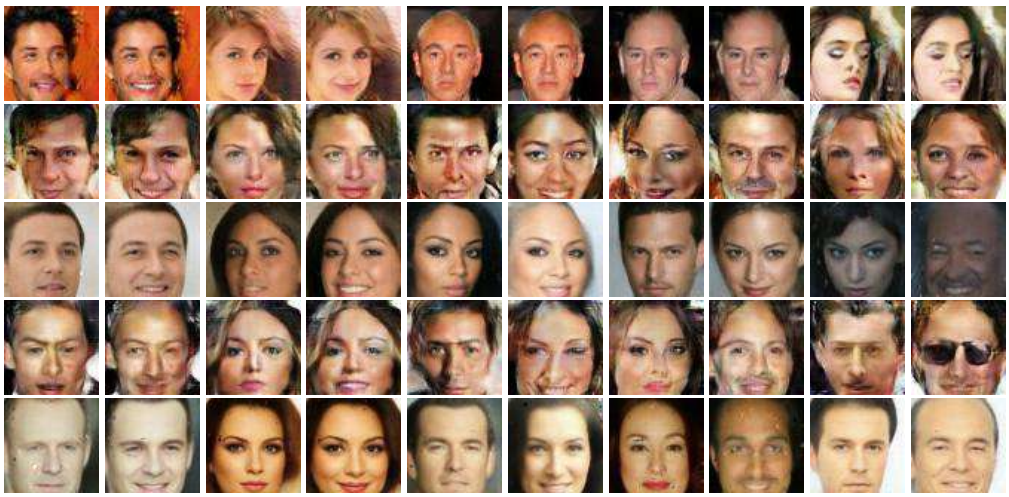


Figure 6: The pairs of images from GAN intervention on $Smiling$. The odd (even) columns show the images intervention on *Smiling*$= 0$ (*Smiling*$= 1$). The images from row 1 to row 5 are Conditional GAN with Wasserstein distance, Causal GAN, Causal BEGAN, Causal GAN-IC, and Causal BEGAN-IC. We report successful images from column 1 to 4, and failure images from column 5 to 10.

5.3 Identity Preserving Evaluation for all Thresholds

We reported Identity Preserving (IP) value with threshold 0.2 for *Real*, and 0.6 for *Pair*. Here, Table 5 and 6 show the IP value for all thresholds from 0.1 to 0.9 for *Mustache* and *Smiling*. Each IP-thr means the probability of how both sets are identity for given threshold, thr. In *Real* case, both sets are real images and counterfactual images from VAE based model. In *Pair* cases, both sets are generated from the model, reconstructed images and counterfactual images in VAE, and images of different interventions with the same noise in GAN.

	Model	IP-0.1	IP-0.2	IP-0.3	IP-0.4	IP-0.5	IP-0.6	IP-0.7	IP-0.8	IP-0.9
Real	CVAE	$0.66_{\pm 0.04}$	$0.26_{\pm 0.04}$	$0.04_{\pm 0.01}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
	CEVAE	$0.68_{\pm 0.03}$	$0.29_{\pm 0.04}$	$0.06_{\pm 0.01}$	$\mathbf{0.01_{\pm 0.0}}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
	mCEVAE	$0.64_{\pm 0.03}$	$0.23_{\pm 0.02}$	$0.04_{\pm 0.01}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
	DCEVAE (ours)	$\mathbf{0.71_{\pm 0.02}}$	$\mathbf{0.33_{\pm 0.02}}$	$\mathbf{0.08_{\pm 0.01}}$	$\mathbf{0.01_{\pm 0.0}}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
Pair	CWGAN	$\mathbf{1.00_{\pm 0.00}}$	$0.97_{\pm 0.00}$	$0.84_{\pm 0.01}$	$0.58_{\pm 0.01}$	$0.30_{\pm 0.02}$	$0.11_{\pm 0.01}$	$0.02_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
	DCGAN	$0.96_{\pm 0.02}$	$0.83_{\pm 0.04}$	$0.62_{\pm 0.05}$	$0.42_{\pm 0.04}$	$0.25_{\pm 0.03}$	$0.13_{\pm 0.02}$	$0.05_{\pm 0.01}$	$0.01_{\pm 0.0}$	$0.0_{\pm 0.0}$
	BEGAN	$0.97_{\pm 0.02}$	$0.86_{\pm 0.07}$	$0.67_{\pm 0.16}$	$0.51_{\pm 0.25}$	$0.39_{\pm 0.30}$	$0.3_{\pm 0.35}$	$0.23_{\pm 0.38}$	$0.15_{\pm 0.28}$	$0.0_{\pm 0.0}$
	DCGAN-IC	$0.95_{\pm 0.01}$	$0.80_{\pm 0.02}$	$0.57_{\pm 0.01}$	$0.36_{\pm 0.02}$	$0.20_{\pm 0.01}$	$0.09_{\pm 0.01}$	$0.04_{\pm 0.01}$	$0.01_{\pm 0.0}$	$0.0_{\pm 0.0}$
	BEGAN-IC	$0.93_{\pm 0.02}$	$0.75_{\pm 0.05}$	$0.51_{\pm 0.07}$	$0.31_{\pm 0.06}$	$0.16_{\pm 0.04}$	$0.08_{\pm 0.02}$	$0.04_{\pm 0.01}$	$0.02_{\pm 0.01}$	$\mathbf{0.01_{\pm 0.01}}$
	CVAE	$\mathbf{1.00_{\pm 0.00}}$	$0.99_{\pm 0.03}$	$0.95_{\pm 0.08}$	$0.86_{\pm 0.13}$	$0.68_{\pm 0.17}$	$0.42_{\pm 0.17}$	$0.11_{\pm 0.07}$	$0.01_{\pm 0.01}$	$0.0_{\pm 0.0}$
	CEVAE	$\mathbf{1.00_{\pm 0.00}}$	$\mathbf{1.00_{\pm 0.00}}$	$0.98_{\pm 0.01}$	$0.94_{\pm 0.04}$	$0.86_{\pm 0.06}$	$0.75_{\pm 0.07}$	$0.56_{\pm 0.07}$	$0.19_{\pm 0.05}$	$0.0_{\pm 0.0}$
	mCEVAE	$\mathbf{1.00_{\pm 0.00}}$	$0.98_{\pm 0.03}$	$0.93_{\pm 0.09}$	$0.85_{\pm 0.16}$	$0.7_{\pm 0.19}$	$0.43_{\pm 0.21}$	$0.16_{\pm 0.18}$	$0.03_{\pm 0.04}$	$0.0_{\pm 0.0}$
	DCEVAE (ours)	$\mathbf{1.00_{\pm 0.00}}$	$\mathbf{1.00_{\pm 0.00}}$	$\mathbf{1.00_{\pm 0.00}}$	$\mathbf{1.00_{\pm 0.00}}$	$\mathbf{1.00_{\pm 0.00}}$	$\mathbf{0.98_{\pm 0.01}}$	$\mathbf{0.8_{\pm 0.06}}$	$\mathbf{0.27_{\pm 0.05}}$	$0.0_{\pm 0.0}$

Table 5: The identity preserving values for threshold from 0.1 to 0.9 for $a =$ *Mustache*.

	Model	IP-0.1	IP-0.2	IP-0.3	IP-0.4	IP-0.5	IP-0.6	IP-0.7	IP-0.8	IP-0.9
Real	CVAE	$0.72_{\pm 0.07}$	$0.33_{\pm 0.1}$	$\mathbf{0.08_{\pm 0.05}}$	$\mathbf{0.01_{\pm 0.01}}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
	CEVAE	$0.67_{\pm 0.01}$	$0.28_{\pm 0.01}$	$0.06_{\pm 0.0}$	$\mathbf{0.01_{\pm 0.0}}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
	mCEVAE	$0.68_{\pm 0.03}$	$0.26_{\pm 0.03}$	$0.05_{\pm 0.01}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
	DCEVAE (ours)	$\mathbf{0.73_{\pm 0.01}}$	$\mathbf{0.34_{\pm 0.01}}$	$\mathbf{0.08_{\pm 0.0}}$	$\mathbf{0.01_{\pm 0.0}}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
Pair	CWGAN	$\mathbf{1.0_{\pm 0.0}}$	$\mathbf{1.0_{\pm 0.0}}$	$0.99_{\pm 0.0}$	$0.92_{\pm 0.0}$	$0.7_{\pm 0.01}$	$0.32_{\pm 0.01}$	$0.05_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
	DCGAN	$0.96_{\pm 0.02}$	$0.81_{\pm 0.04}$	$0.59_{\pm 0.05}$	$0.37_{\pm 0.05}$	$0.2_{\pm 0.04}$	$0.08_{\pm 0.02}$	$0.02_{\pm 0.01}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
	BEGAN	$0.97_{\pm 0.02}$	$0.86_{\pm 0.07}$	$0.68_{\pm 0.16}$	$0.51_{\pm 0.24}$	$0.4_{\pm 0.3}$	$0.3_{\pm 0.35}$	$0.22_{\pm 0.38}$	$0.13_{\pm 0.27}$	$0.0_{\pm 0.0}$
	DCGAN-IC	$0.94_{\pm 0.01}$	$0.77_{\pm 0.02}$	$0.53_{\pm 0.03}$	$0.31_{\pm 0.03}$	$0.15_{\pm 0.02}$	$0.06_{\pm 0.01}$	$0.01_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
	BEGAN-IC	$0.93_{\pm 0.02}$	$0.75_{\pm 0.04}$	$0.52_{\pm 0.04}$	$0.3_{\pm 0.03}$	$0.15_{\pm 0.02}$	$0.06_{\pm 0.01}$	$0.01_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
	CVAE	$\mathbf{1.0_{\pm 0.0}}$	$0.99_{\pm 0.01}$	$0.95_{\pm 0.08}$	$0.83_{\pm 0.2}$	$0.63_{\pm 0.28}$	$0.36_{\pm 0.25}$	$0.11_{\pm 0.11}$	$0.0_{\pm 0.01}$	$0.0_{\pm 0.0}$
	CEVAE	$\mathbf{1.0_{\pm 0.0}}$	$\mathbf{1.0_{\pm 0.0}}$	$\mathbf{1.0_{\pm 0.0}}$	$\mathbf{1.0_{\pm 0.0}}$	$\mathbf{1.0_{\pm 0.0}}$	$\mathbf{1.0_{\pm 0.0}}$	$\mathbf{0.97_{\pm 0.0}}$	$\mathbf{0.68_{\pm 0.02}}$	$\mathbf{0.02_{\pm 0.0}}$
	mCEVAE	$\mathbf{1.0_{\pm 0.0}}$	$0.98_{\pm 0.02}$	$0.9_{\pm 0.08}$	$0.67_{\pm 0.1}$	$0.37_{\pm 0.07}$	$0.12_{\pm 0.03}$	$0.01_{\pm 0.01}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
	DCEVAE (ours)	$\mathbf{1.0_{\pm 0.0}}$	$\mathbf{1.0_{\pm 0.0}}$	$\mathbf{1.0_{\pm 0.0}}$	$\mathbf{1.0_{\pm 0.0}}$	$\mathbf{1.0_{\pm 0.0}}$	$0.92_{\pm 0.01}$	$0.49_{\pm 0.02}$	$0.05_{\pm 0.01}$	$0.0_{\pm 0.0}$

Table 6: The identity preserving values for threshold from 0.1 to 0.9 for $a =$ *Smiling*.

References

[1] Asuncion, A.; and Newman, D. 2007. UCI machine learning repository.

[2] Cao, C.; Weng, Y.; Zhou, S.; Tong, Y.; and Zhou, K. 2013. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics* 20(3): 413–425.

[3] Daliri, M. R. 2013. Chi-square distance kernel of the gaits for the diagnosis of Parkinson's disease. *Biomedical Signal Processing and Control* 8(1): 66–70.

[4] Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4690–4699.

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

[6] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

[7] Huang, G. B.; Mattar, M.; Berg, T.; and Learned-Miller, E. 2008. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments.

[8] Klys, J.; Snell, J.; and Zemel, R. 2018. Learning latent subspaces in variational autoencoders. In *Advances in Neural Information Processing Systems*, 6444–6454.

[9] Kocaoglu, M.; Snyder, C.; Dimakis, A. G.; and Vishwanath, S. 2017. Causalgan: Learning causal implicit generative models with adversarial training. *arXiv preprint arXiv:1709.02023* .

[10] Kusner, M. J.; Loftus, J.; Russell, C.; and Silva, R. 2017. Counterfactual fairness. In *Advances in Neural Information Processing Systems*, 4066–4076.

[11] Lee, C.-H.; Liu, Z.; Wu, L.; and Luo, P. 2019. MaskGAN: towards diverse and interactive facial image manipulation. *arXiv preprint arXiv:1907.11922* .

[12] Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2018. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August* 15: 2018.

[13] Lucas, J.; Tucker, G.; Grosse, R. B.; and Norouzi, M. 2019. Don't Blame the ELBO! A Linear VAE Perspective on Posterior Collapse. In *Advances in Neural Information Processing Systems*, 9408–9418.

[14] Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in PyTorch. In *NIPS-W*.

[15] Pearl, J. 2009. *Causality*. Cambridge university press.

[16] Rustamov, R. M. 2019. Closed-form Expressions for Maximum Mean Discrepancy with Applications to Wasserstein Auto-Encoders. *arXiv preprint arXiv:1901.03227* .

[17] Takahashi, H.; Iwata, T.; Yamanaka, Y.; Yamada, M.; and Yagi, S. 2019. Variational autoencoder with implicit optimal priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 5066–5073.

[18] Xu, D.; Wu, Y.; Yuan, S.; Zhang, L.; and Wu, X. 2019. Achieving causal fairness through generative adversarial networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*.

[19] Zhang, L.; Wu, Y.; and Wu, X. 2016. A causal framework for discovering and removing direct and indirect discrimination. *arXiv preprint arXiv:1611.07509* .