Communications for Statistical Applications and Methods
2023, Vol. 30, No. 6, 589–603

https://doi.org/10.29220/CSAM.2023.30.6.589
Print ISSN 2287-7843 / Online ISSN 2383-4757

Counterfactual image generation by disentangling data attributes with deep generative models

Jieon Lim[a], Weonyoung Joo[1,a]

[a]Department of Statistics, EWHA Womans University, Korea

Abstract

Deep generative models target to infer the underlying true data distribution, and it leads to a huge success in generating fake-but-realistic data. Regarding such a perspective, the data attributes can be a crucial factor in the data generation process since non-existent counterfactual samples can be generated by altering certain factors. For example, we can generate new portrait images by flipping the gender attribute or altering the hair color attributes. This paper proposes counterfactual disentangled variational autoencoder generative adversarial networks (CDVAE-GAN), specialized for data attribute level counterfactual data generation. The structure of the proposed CDVAE-GAN consists of variational autoencoders and generative adversarial networks. Specifically, we adopt a Gaussian variational autoencoder to extract low-dimensional disentangled data features and auxiliary Bernoulli latent variables to model the data attributes separately. Also, we utilize a generative adversarial network to generate data with high fidelity. By enjoying the benefits of the variational autoencoder with the additional Bernoulli latent variables and the generative adversarial network, the proposed CDVAE-GAN can control the data attributes, and it enables producing counterfactual data. Our experimental result on the CelebA dataset qualitatively shows that the generated samples from CDVAE-GAN are realistic. Also, the quantitative results support that the proposed model can produce data that can deceive other machine learning classifiers with the altered data attributes.

Keywords: counterfactual data generation, deep generative model, variational autoencoder, generative adversarial network, disentangled feature extraction

1. Introduction

Traditional generative models aim to model the data distribution and the data generation process by inferring the relationship between the data and their labels or attributes. Recently, deep generative models have arisen for generating realistic data due to the success of deep neural networks. For example, a variational autoencoder (VAE) (Kingma and Welling, 2014) explicitly models the latent distribution of data, and it can extract data features in relatively low dimensions with the encoder network. Afterward, the extracted feature can be utilized in data generation through the decoder network. Meanwhile, a generative adversarial network (GAN) (Goodfellow *et al*., 2014) can generate data from implicit distribution by adversarial training of the generator and discriminator networks. As a result, GANs can generate realistic data from random noise through the generator network.

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2022-00166289).

[1]Corresponding author: Department of Statistics, Ewha Womans University, 52 Ewhayeodae-gil, Seodaemun-Gu, Seoul 03760, Korea. E-mail: weonyoungjoo@ewha.ac.kr

Published 30 November 2023 / journal homepage: http://csam.or.kr

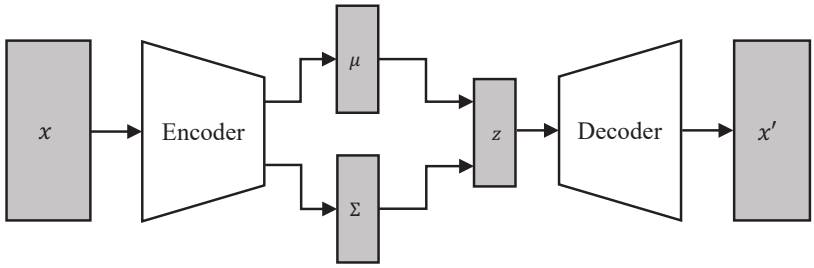


Figure 1: *The neural network architecture of VAE.*

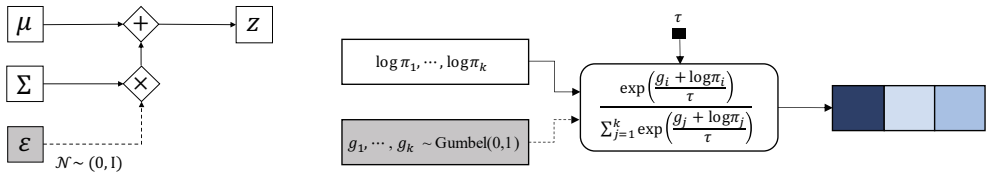
In counterfactual data generation, we can interpret the effectiveness of the alternation or the perturbation by generating the data with an assumption on the situation which has not happened. For example, counterfactual image generation can be utilized in open set recognition to detect unknown classes by resembling real images of known classes and exploring unseen properties in the training dataset (Neal *et al*., 2018). Furthermore, the work of Kusner *et al*. (2017), as known as counterfactual fairness, analyzes if a fair decision has been made with respect to certain factors by comparing the real data and the counterfactually generated fake data.

Recently, several works have been conducted regarding learning disentangled representation in deep generative models. β-VAE (Higgins *et al*., 2016) introduces adjustable hyperparameter β as a scale factor to the KL divergence term in the loss function. Appropriately tuned $\beta > 1$ results in disentangled representation learning and discovers more factors compared to the previous works. JointVAE (Dupont, 2018) utilizes both continuous and discrete priors to generate diverse and realistic images. By augmenting the discrete priors, JointVAE is able to discover categorical features automatically. However, those works did not fully utilize the data attributes, which are the auxiliary data information.

This paper proposes counterfactual disentangled variational autoencoder generative adversarial networks (CDVAE-GAN), specialized for generating counterfactual data by controlling the data attributes, consists of Gaussian VAE with additional Bernoulli latent variables, and GAN. Our contributions are as follows.

- CDVAE-GAN can capture data features and data attributes separately in the latent space by utilizing Gaussian VAE with auxiliary Bernoulli random variables.
- CDVAE-GAN is able to generate counterfactual data by altering the extracted attributes in the latent space.
- High-quality data can be produced due to the nature of GAN.
- The experiment results on the CelebA dataset demonstrate the effectiveness of counterfactual data generation of CDVAE-GAN.

The remainder of this paper is organized as follows. In Section 2, we introduce basic concepts such as VAEs, GAN, and VAE-GAN. The proposed CDVAE-GAN is presented in Section 3, and the empirical results of CDVAE-GAN are presented in Section 4. Finally, Section 5 concludes our work.



(a) *Gaussian VAE* (b) *Categorical VAE*

Figure 2: *Reparameterization trick of VAEs. The shaded nodes indicate the auxiliary random variables for the reparameterization. (a) Gaussian VAE introduces auxiliary standard Gaussian random variable* $\epsilon \sim \mathcal{N}(0, I)$. *(b) Categorical VAE introduces auxiliary Gumbel random variable* $g \sim Gumbel(0, 1)$ *for the repaarameterization, and temperature* $\tau > 0$ *for the continuous relaxation.*

2. Preliminaries

2.1. Variational autoencoders

Variational autoencoder (VAE) (Kingma and Welling, 2014), a subtype of deep generative models, assumes that (1) a latent variable z follows prior distribution $p(z) = \mathcal{N}(0, I)$; and (2) the observation x is generated from the latent variable z with generative neural network $p_\theta(x|z)$. However, due to the intractability of true posterior distribution $p(z|x) = (p_\theta(x|z)p(z))/\int p_\theta(x|z)p(z)dz$ where $p_\theta(x|z)$ is a neural network, we introduce an approximate posterior distribution $q_\phi(z|x) = \mathcal{N}(\mu, \Sigma)$, called inference neural network, to infer the latent variable z. Here, we derive the Gaussian parameters μ and Σ with neural networks from input x, and we call this type of inference an amortized inference. We often refer $q_\phi(z|x)$ and $p_\theta(x|z)$ as encoder and decoder, respectively, since $q_\phi(z|x)$ extracts latent representation z from input x, and $p_\theta(x|z)$ reconstructs the original input x as x'. Figure 1 illustrates the neural network architecture of VAE.

The objective function of VAE is called evidence lower bound (ELBO), which is a lower bound of log-likelihood $\log p(x)$, followed by the intractability of the true posterior distribution $p(z|x)$. Equation (2.2) presents the negative ELBO of VAE, which becomes a loss function.

$$L_{VAE} = L_{Rec} + L_{Prior} \tag{2.1}$$

$$= -\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta\left(x \mid z\right)\right] + D_{KL}\left(q_\phi\left(z \mid x\right) \| p(z)\right). \tag{2.2}$$

The loss function of VAE consists of two terms: (1) reconstruction loss term $L_{Rec} = -\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$; and (2) KL divergence term $L_{Prior} = D_{KL}(q_\phi(z|x)\|p(z))$. Firstly, the reconstruction term enables the encoder and the decoder to produce output $\hat{x}$ as close as the input x. Regarding that, mean squared error is used for the real-valued input, and cross-entropy loss is utilized in 0-1-scaled data. Secondly, the KL divergence term acts as a regularizer for the approximate posterior distribution $q_\phi(z|x)$ from the prior distribution $p(z)$.

Training VAE requires the estimation of the stochastic gradients, which can be derived by either the score function method or the reparameterization trick (Joo *et al.*, 2020b). Especially, VAE utilizes a reparameterization trick in the training phase, i.e., instead of directly sampling z from $\mathcal{N}(\mu_\phi, \Sigma_\phi)$, we alternatively sample ϵ from $\mathcal{N}(0, I)$ and then compute $z = \mu_\phi + \epsilon \times \Sigma_\phi$. Due to the alternative sampling, we can compute the gradients $\partial z/\partial \mu_\phi$ and $\partial z/\partial \Sigma_\phi$. Hence, the gradient signal can flow to the front-end of the neural network. It should be noted that different reparameterization tricks should be applied regarding the selection of prior distributions (Nalisnick and Smyth, 2017; Joo *et al.*, 2020a). Figure 2(a) presents the reparameterization trick of the Gaussian VAE.

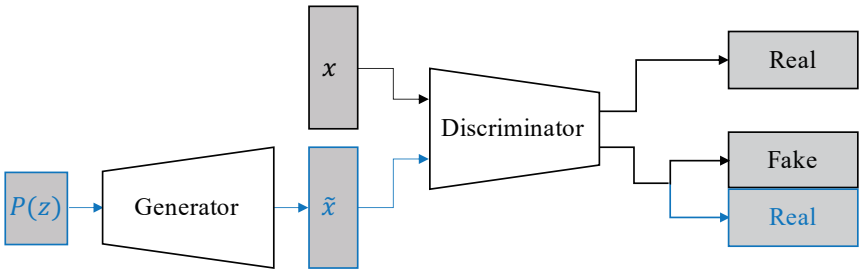


Figure 3: *The neural network architecture of GAN.*

Categorical VAE (Jang *et al.*, 2016) assumes categorical distribution as a prior distribution of the latent variable instead of the Gaussian in the usual VAEs, which we name Gaussian VAE in this paper. Similar to the Gaussian VAE, categorical VAE also requires a reparameterization trick, which is tailored to the categorical distribution. The authors of Jang *et al.* (2017) proposed the Gumbel-Softmax trick, which approximates the categorical distribution with a special form of a continuous distribution. The Gumbel-Softmax trick utilizes (1) auxiliary Gumbel samples $g_1, \ldots, g_k$ from Gumbel distribution Gumbel(0, 1); (2) softmax function to transform real-value space into sum-to-one simplex; and (3) a hyperparameter $\tau > 0$, called temperature, to approximate one-hot values as $\tau \to 0$. Equation (2.3) presents the Gumbel-Softmax transformation given Gumbel samples $g_1, \ldots, g_k$, temperature τ, and the categorical parameter $\pi = (\pi_1, \ldots, \pi_k)$:

$$y_i = \frac{\exp\left(\left(g_i + \log \pi_i\right)/\tau\right)}{\sum_{j=1}^{k} \exp\left(\left(g_j + \log \pi_j\right)/\tau\right)} \quad \text{for } i = 1, \ldots, k, \tag{2.3}$$

and the transformed value $y = (y_1, \ldots, y_k)$ in the sum-to-one simplex has continuously relaxed one-hot value. Finally, Figure 2(b) illustrates the Gumbel-Softmax reparameterization trick.

2.2. Generative adversarial networks

A generative adversarial network (Goodfellow *et al.*, 2014) consists of two sub-networks: Generator G and discriminator D. Generator G takes random input from the standard Gaussian distribution or the standard uniform distribution, and generates fake data by neural networks. The goal of the generator is to fool the discriminator into classifying the generated data as real, and the corresponding objective function of the generator is as follows:

$$\max_G \mathbb{E}_{z\sim p(z)} \left[\log\left(D\left(G(z)\right)\right)\right]. \tag{2.4}$$

Meanwhile, discriminator D classifies whether the input data is real or generated by the generator, i.e., fake. The objective function of the discriminator is as follows:

$$\max_D \mathbb{E}_{x\sim p_{data}} \left[\log\left(D(x)\right)\right] + \mathbb{E}_{z\sim p(z)} \left[\log\left(1 - D\left(G(z)\right)\right)\right]. \tag{2.5}$$

Summing up, the final objective function of GAN is the following:

$$\min_G \max_D L_{GAN} = \min_G \max_D \mathbb{E}_{x\sim p_{data}} \left[\log\left(D(x)\right)\right] + \mathbb{E}_{z\sim p(z)} \left[\log\left(1 - D\left(G(z)\right)\right)\right], \tag{2.6}$$

and Figure 3 illustrates the neural network architecture of GAN.

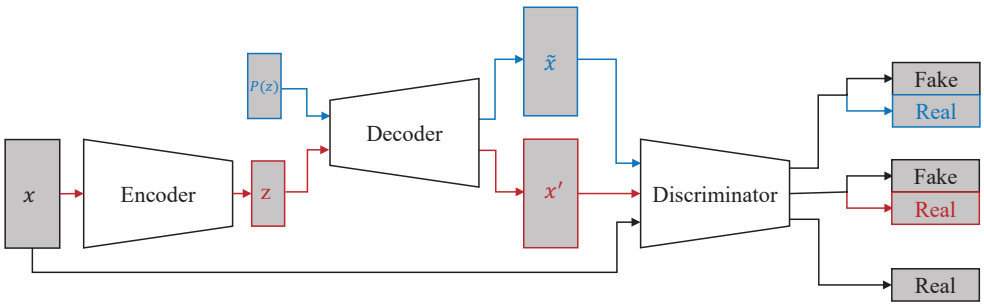


Figure 4: *The neural network structure of VAE-GAN. The discriminator classifies if the data is real or fake (reconstructed or generated by the decoder/generator). Meanwhile, the decoder/generator fools the discriminator into classifying the reconstructed or generated data as real data.*

The generator and the discriminator alternately train with the min-max game, which can be interpreted as finding a Nash equilibrium of a non-cooperative game between two adversarial players. It is well-known that the global optimum of the generator becomes the true data distribution (Goodfellow *et al.*, 2014), i.e., $p_G = p_{data}$ where p_G and p_{data} indicate the generator distribution and the true data distribution, respectively. Following the work of Goodfellow *et al.* (2014), the optimal discriminator D is

$$D_G^*(x) = \arg\max_D L_{GAN} \tag{2.7}$$

$$= \arg\max_D \mathbb{E}_{x\sim p_{data}}\left[\log\left(D(x)\right)\right] + \mathbb{E}_{z\sim p(z)}\left[\log\left(1 - D(G(z))\right)\right] \tag{2.8}$$

$$= \frac{p_{data}(x)}{p_{data}(x) + p_G(x)} \tag{2.9}$$

for any given fixed generator G. This consequently results in

$$\min_G \mathbb{E}_{x\sim p_{data}}\left[\log\left(D_G^*(x)\right)\right] + \mathbb{E}_{z\sim p(z)}\left[\log\left(1 - D_G^*\left(G(z)\right)\right)\right] \tag{2.10}$$

$$= \min_G \mathbb{E}_{x\sim p_{data}}\left[\log\left(D_G^*(x)\right)\right] + \mathbb{E}_{x\sim p_G(x)}\left[\log\left(1 - D_G^*(x)\right)\right] \tag{2.11}$$

$$= \min_G \mathbb{E}_{x\sim p_{data}}\left[\log\frac{p_{data}(x)}{p_{data}(x) + p_G(x)}\right] + \mathbb{E}_{x\sim p_G(x)}\left[\log\frac{p_G(x)}{p_{data}(x) + p_G(x)}\right] \tag{2.12}$$

$$= -\log 4 + 2 \cdot JS\left(p_{data} \| p_G\right) \tag{2.13}$$

$$= -\log 4, \tag{2.14}$$

where $JS(\cdot\|\cdot)$ indicates the Jensen–Shannon divergence between two distributions, and the minimum value is achieved when $p_G = p_{data}$ holds. Hence, the well-optimized generator is able to produce realistic data. However, due to the difficulty of the adversarial training, the generator often fails to achieve the optimum. Consequently, a line of work has been studied to overcome the problem of difficult training in GAN (Radford *et al.*, 2016; Salimans *et al.*, 2016).

2.3. VAE-GAN

The authors of Larsen *et al.* (2016) introduce VAE-GAN, which combines VAE and GAN. Specifically, VAE-GAN consists of three sub-networks: (1) an encoder of VAE, (2) a decoder of VAE,

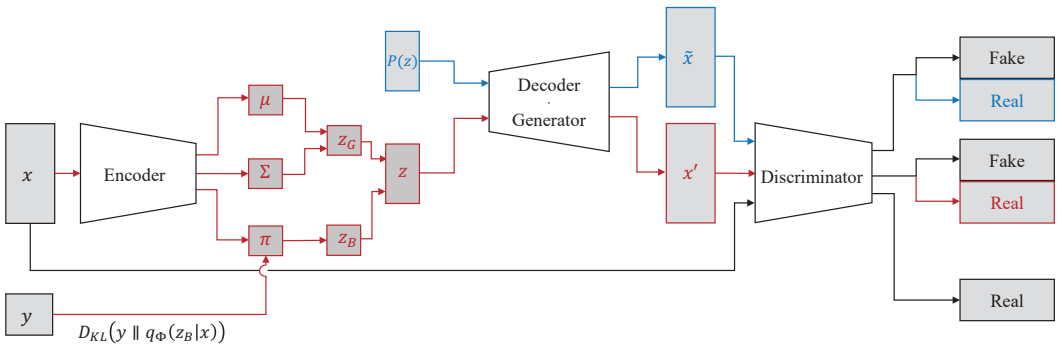


Figure 5: *The neural network architecture of CDVAE-GAN. Compared to VAE-GAN (Larsen et al., 2016), the proposed CDVAE-GAN introduces Bernoulli random variable z_B to model the data attributes. Further, the attribute information is disentangled from the latent feature z_G by the additional regularization term $D_{KL}(y\|q_\phi(z_B|x))$ in the objective function of CDVAE.*

which also work as a generator of GAN, and (3) a discriminator of GAN. Firstly, the encoder acts as the typical VAE encoder, and it extracts the latent representation of input data. Secondly, the decoder/generator plays the role of the VAE decoder as well as the GAN generator. By taking the encoded latent representation from the VAE encoder, it decodes the latent values to the original data space. Meanwhile, it also generates data from random noise as a regular GAN generator. Finally, the discriminator classifies if the data is real or fake. Here, the reconstructed data in the VAE perspective and the generated data in the view of the GAN generator are treated as fake data for the discriminator. The model architecture of VAE-GAN is illustrated in Figure 4. Since our proposed model is an extension of VAE-GAN, we introduce the detailed model structure and the loss function of VAE-GAN in the next section.

3. Methodology

In this section, we introduce counterfactual disentanlged variational autoencoder generative adversarial networks (CDVAE-GAN), an application of VAE-GAN (Larsen *et al.*, 2016) in counterfactual data generation. Figure 5 illustrates the overview of the neural network architecture of CDVAE-GAN, which consists of an encoder, a decoder/generator, and a discriminator, as VAE-GAN. It should be noted that while the proposed CDVAE-GAN has a similar model structure to CVAE-GAN (Bao *et al.*, 2017), the way of using the auxiliary information is different. In detail, CVAE-GAN utilizes a single category of celebrity, for example, Hathaway, Leonardo, Cheryl Hines, Jet Li, etc. However, the proposed CDVAE-GAN utilizes multiple attributes of the portrait, such as bangs, black hair, blond hair, gender, mustache, mouth slightly open, smiling, etc. Consequently, this difference results in a different model purpose, where CVAE-GAN cannot produce counterfactual generation on the attributes. Hence, the two works cannot be directly compared in terms of the counterfactual generation.

3.1. Model description

3.1.1. Encoder

Encoder E of CDVAE-GAN is designed for capturing the latent representation of the existing (real) data. Compared to the typical Gaussian VAE, we additionally utilize discrete latent variables to pro-

duce disentangled latent representations following JointVAE (Dupont, 2018). Particularly, we adopt multiple Bernoulli random variables to (1) directly model the data attributes; (2) separate the attribute information from extracted features of the Gaussian VAE; and (3) control the attributes in the latent space to create the counterfactual data. By taking the real data as input, the CDVAE-GAN encoder outputs (1) low-dimensional latent representation $z_G \in \mathbb{R}^m$ sampled from Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, which is the approximate posterior distribution $q_\phi(z_G|x)$ that captures the feature of the data; and (2) latent attribute $z_B \in \{0, 1\}^k$ sampled from Bernoulli distribution $\text{Ber}(\pi) = q_\phi(z_B|x)$, which reflects the data attributes.

$$z_G \sim \mathcal{N}(\mu, \Sigma), \quad z_B \sim \text{Ber}(\pi) \quad \text{where} \quad \mu, \Sigma, \pi = E_\phi(x). \tag{3.1}$$

To extract data features, which are distinguished from the data attributes, we utilize the known data attributes $y \in \{0, 1\}^k$ of a value zero or one, which are given together with input x. Afterward, we add the forward KL divergence term $D_{KL}(y||q_\phi(z_B|x))$ in the loss function as a regularizer.

The loss function from the encoder, which needs to be minimized, is the following:

$$\begin{aligned}
L_{Div} &= L_{Prior} + \lambda_B L_B && (3.2)\\
&= D_{KL}\left(q_\phi\left(z_G \mid x\right) \| p(z_G)\right) + \lambda_B \cdot D_{KL}\left(y \| q_\phi\left(z_B \mid x\right)\right) && (3.3)\\
&= -\frac{1}{2}\sum_{i=1}^{m}\left(\mu_i^2 + \sigma_i^2 - \log\left(\sigma_i^2\right) - 1\right) + \lambda_B \cdot \sum_{j=1}^{k}\left(y_j \log\frac{y_j}{\pi_j} + \left(1 - y_j\right)\log\frac{1 - y_j}{1 - \pi_j}\right) && (3.4)\\
&\to -\frac{1}{2}\sum_{i=1}^{m}\left(\mu_i^2 + \sigma_i^2 - \log\left(\sigma_i^2\right) - 1\right) - \lambda_B \cdot \sum_{j=1}^{k}\left(y_j \log \pi_j + \left(1 - y_j\right)\log\left(1 - \pi_j\right)\right) && (3.5)\\
&\quad \text{as } y \to 0 \text{ or } 1\ .
\end{aligned}$$

The approximation in equation (3.5) is due to $\lim_{a\to 0^+} a \log a = 0$. While we can utilize the reverse KL divergence instead of the forward KL, the reverse KL divergence is intractable in the proposed Bernoulli case. Hence, we utilize the forward KL divergence instead as the attribute deviating regularizer. It is well-known that optimizing forward and reverse KL divergences results in mode-covering property and mode-seeking property, respectively (Kim *et al.*, 2021). However, when the target distribution and the learnable distribution are both simple, such as in the Bernoulli case, optimizing the forward and reverse KL divergence results in the same optimal learnable distribution. In our experiment, we take $\lambda_B = 1.0$ as a hyperparameter.

Without the additional Bernoulli latent variable and the corresponding forward KL divergence regularizer L_B, the loss function becomes L_{Prior}, and the encoder structure reduces down to the original VAE-GAN encoder. However, since the additional regularizer L_B enforces π to capture the effect of data attribute y, μ and Σ are discriminated from the data attribute information. Therefore, the sampled latent value z_G from $\mathcal{N}(\mu, \Sigma)$ can have a disentangled representation, which is independent of data attributes y.

3.1.2. Decoder/generator

As a decoder of VAE. Decoder G of CDVAE-GAN takes the latent variable z as an input, which consists of the latent feature z_G as well as the latent attribute z_B. From the encode latent value $z = (z_G, z_B)$, the decoder reconstructs original input as $p_\theta(x|z_G, z_B)$. Since the CDVAE-GAN encoder is designed to have the disentangled latent representation between the data feature and the data attribute, the decoder generates (1) general data structure from latent feature z_G, and (2) specific structure from latent

attribute z_B where the data attribute is embedded. Here, the latent attribute z_B can act as conditional information for the VAE decoder. Hence, the decoder part resembles the decoder of conditional VAE (Sohn *et al.*, 2015). In other words, the attribute factor and the output data are in a causal relationship in the generative model view. Without the latent attribute variable z_B, the decoder/generator structure reduces to the original VAE-GAN decoder/generator. The objective of the decoder is minimizing the reconstruction loss, which can be written as follows:

$$L_{Rec} = -\mathbb{E}_{z_G, z_B \sim q_\phi(z_G, z_B | x)} \left[\log p_\theta \left(x \mid z_G, z_B \right) \right]. \tag{3.6}$$

As a generator of GAN. At the same time, the CDVAE-GAN decoder also acts as the generator so that it takes (1) random Gaussian noise sampled from $\mathcal{N}(0, I) = p(z_G)$, which is a prior distribution of the latent variable z_G; and (2) random Bernoulli sample from $\text{Ber}(\eta) = p(z_B)$. In our experiment, we simply take the Bernoulli prior as $\eta = 0.5$. Such generated data, as well as the reconstructed data, do not exist in the training dataset. Hence, they are considered fake data, which are trying to fool discriminator D.

3.1.3. Discriminator

Discriminator D of CDVAE-GAN classifies whether the data is real or fake. In the discriminator's view, all data decoded or generated by decoder/generator G are treated as fake data. Hence, discriminator D and the VAE, which consists of encoder E and decoder/generator G, are adversarially optimized through a min-max game with the following objective functions:

$$\min_{E,G} \max_{D} L_{GAN} \quad \text{where} \tag{3.7}$$

$$L_{GAN} = \mathbb{E}_{x \sim p_{data}} \left[\log D(x) \right] \tag{3.8}$$

$$+ \mathbb{E}_{z_G, z_B \sim p(z_G, z_B)} \left[\log \left(1 - D(G(z_G, z_B)) \right) \right] \tag{3.9}$$

$$+ \mathbb{E}_{x \sim p_{data}} \left[\log \left(1 - D(G(E(x))) \right) \right]. \tag{3.10}$$

Here, equations (3.8), (3.9), and (3.10) results from true data x, the generated data $\tilde{x}$, and the reconstructed data x', respectively.

3.2. Training process

Throughout Section 3.1, the final objective function of CDVAE-GAN is as follows:

$$L_{\text{CDVAE-GAN}} = L_{Div} + L_{Rec} + L_{GAN}. \tag{3.11}$$

When training the neural network parameter of the proposed CDVAE-GAN model, we alternately update the encoder, the decoder/generator, and the discriminator since each network's parameters are linked to different loss components. Among three terms, which consist of the total objective $L_{CDVAE-GAN}$, (1) encoder E minimizes L_{Div} and L_{Rec}, (2) decoder/generator G minimizes L_{Rec} and L_{GAN}, and (3) discriminator D maximizes L_{GAN}. While training decoder/generator G, we additionally utilize a scale factor γ to produce more accurate reconstructions of the data patterns in the training dataset for deceiving the discriminator (Gatys *et al.*, 2016). A greater value of $\gamma > 1$ strengthens the reconstruction quality; therefore, the reconstructed data resembles the original input more. We set $\gamma = 15$ as a hyperparameter in our experiment. The detailed training procedure is shown in Algorithm 1.

Algorithm 1 : CDVAE-GAN model training process.

$\theta_E, \theta_G, \theta_D$ ▷ Neural network parameters
repeat
  $z \leftarrow E(x)$ ▷ Encode input x
  $L_{Div} = D_{KL}(q_\phi(z_G|x)\|p(z_G)) + \lambda_B \cdot D_{KL}(y\|q_\phi(z_B|x))$ ▷ Data attribute $y \in \{0,1\}^k$
  $x' \leftarrow G(z)$ ▷ Decode z, i.e., reconstruct x'
  $L_{Rec} = -\mathbb{E}_{z_G, z_B \sim q_\phi(z_G, z_B|x)}\Big[\log p_\theta(x|z_G, z_B)\Big]$ ▷ Minimize reconstruction loss
  $\tilde{z} = \{\tilde{z}_G, \tilde{z}_B\}$ ▷ Draw random noise $\tilde{z}_G \sim \mathcal{N}(0, I)$, $\tilde{z}_B \sim$ Ber(0.5)
  $\tilde{x} \leftarrow G(\tilde{z})$ ▷ Generate $\tilde{x}$
  $L_{GAN} = \mathbb{E}_{x \sim p_{data}}\Big[\log D(x)\Big]$ ▷ Loss from real data
    $+\mathbb{E}_{z_G, z_B \sim p(z_G, z_B)}\Big[\log\left(1 - D(G(z_G, z_B))\right)\Big]$ ▷ Loss from random noise
    $+\mathbb{E}_{x \sim p_{data}}\Big[\log\left(1 - D(G(E(x)))\right)\Big]$ ▷ Loss from reconstruction
  Update parameters according to gradients.
  $\theta_E \mathrel{+}= -\nabla_{\theta_E}(L_{Div} + L_{Rec})$ ▷ Minimize $L_{CDVAE-GAN}$ with respect to E
  $\theta_G \mathrel{+}= -\nabla_{\theta_G}(\gamma L_{Rec} + L_{GAN})$ ▷ Minimize $L_{CDVAE-GAN}$ with respect to G
  $\theta_D \mathrel{+}= \nabla_{\theta_D}(L_{GAN})$ ▷ Maximize $L_{CDVAE-GAN}$ with respect to D
until convergence

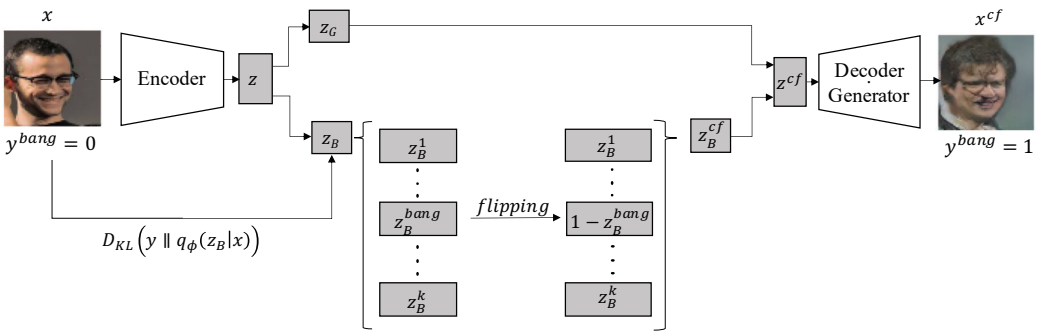


Figure 6: *Counterfactual data generation process of CDVAE-GAN.*

3.3. Counterfactual data generation

From the input data, the proposed CDVAE-GAN is able to generate counterfactual data with respect to a certain data attribute. In the CDVAE-GAN model structure, the latent Bernoulli variable z_B plays a key role in counterfactual data generation. The latent Bernoulli variable z_B models attribute information due to the regularizer term $D_{KL}(y\|q_\phi(z_B|x))$ contained in the loss function, so it is possible to handle the characteristics of specific attributes by controlling the corresponding latent Bernoulli variables. For example, portrait x without bang hair ($y^{bang} = 0$) is encoded as the latent value $z = (z_G, z_B)$. Because of the regularizer term $D_{KL}(y\|q_\phi(z_B|x))$, which enforces the auxiliary Bernoulli variables to capture the data attributes, the latent attribute corresponding to the bang hair tends to have zero value ($z_B^{bang} = 0$). Then, we can generate the counterfactual image x^{cf} by flipping the value of z_B into one ($1 - z_B^{bang} = 1$). Afterward, the well-trained decoder/generator will activate the effect of bang hair while generating the portrait x^{cf} through the decoder/generator G. Figure 6 illustrates the counterfactual image generation process when an input image is given.

Table 1: The detailed network structures and the hyperparameter values of CDVAE-GAN used in our experim ents

Encoder E			Decoder/Generator G	Discriminator D	Hyperparameters
5×5 64 Conv2d			$8 \times 8 \times 512$ FC	5×5 64 Conv2d	learning rate of E: 3e-4
BatchNorm2d			BatchNorm2d	BatchNorm2d	learning rate of G: 3e-4
ReLU			ReLU	LeakyReLU	learning rate of D: 3e-5
5×5 128 Conv2d			5×5 512 ConvTr2d	5×5 128 Conv2d	Optimizer: RMSProp
BatchNorm2d			BatchNorm2d	BatchNorm2d	batch size: 32
ReLU			ReLU	LeakyReLU	$\lambda_B = 1.0$
5×5 256 Conv2d			5×5 256 ConvTr2d	5×5 256 Conv2d	$\gamma = 15.0$
BatchNorm2d			BatchNorm2d	BatchNorm2d	$\tau = 0.1$
ReLU			ReLU	LeakyReLU	
5×5 512 Conv2d			5×5 128 ConvTr2d	5×5 512 Conv2d	
BatchNorm2d			BatchNorm2d	BatchNorm2d	
ReLU			ReLU	LeakyReLU	
2048 FC			5×5 64 ConvTr2d	5×5 512 Conv2d	
BatchNorm1d			BatchNorm2d	BatchNorm2d	
ReLU			ReLU	LeakyReLU	
1024 FC			5×5 3 ConvTr2d	2048 FC	
1024 FC			Tanh	LeakyReLU	
BatchNorm1d				1 FC	
ReLU				Sigmoid	
μ	Σ	π			
100 FC	100 FC	38 FC			
	Exp	Sigmoid			

4. Experiment

4.1. Experiment setting

We conduct experiments on the CelebA dataset (Liu *et al*., 2015) to verify the performance of the proposed CDVAE-GAN. CelebA dataset consists of 202, 599 colored images of celebrities with 40 different binary attribute annotations. Among 40 data attributes, we utilized 38 attributes in our experiment where wearing necktie and young attributes are excluded. To generate counterfactual images, we choose the following attributes: Bangs, black hair, blond hair, male (gender), mustache, mouth slightly open, and smiling. The original CelebA dataset has 178×218 pixel size; however, we scale down to 64×64 in our experiment for fair comparisons against VAE-GAN (Larsen *et al*., 2016) experiments. We randomly divide the dataset into $1 : 1$ ratios for the train/test data split.

We compare the proposed CDVAE-GAN against VAE and VAE-GAN (Larsen *et al*., 2016). The VAE baseline model has the same VAE structure as the proposed CDVAE-GAN, which utilizes the Gaussian VAE and additional Bernoulli random variable. In other words, the difference between VAE and CDVAE-GAN is whether the GAN part exists or not. Hence, the comparison between VAE and CDVAE-GAN can show the efficacy of the GAN module. Meanwhile, the other baseline, the VAE-GAN, which is introduced in Section 2.3, has both the VAE part and the GAN part in the model structure. However, there are no Bernoulli random variables and the corresponding regularizer in the loss function. Hence, the VAE-GAN cannot model the data attribute directly in the latent space of the VAE. Instead, the VAE-GAN calculates and utilizes the mean vector of latent values from true data that contains specific attributes or not, for counterfactual data generation. Table 1 presents the detailed neural network structure of CDVAE-GAN that we utilized in the experiment.

Figure 7: *Counterfactual image generation of the proposed CDVAE-GAN and the baseline models with CelebA dataset. The first row shows the input images of the encoder modules, and the second row presents the reconstructed images from the decoder modules. The rest of the rows exhibit counterfactual images for the selected attributes of CelebA dataset.*

4.2. Qualitative result

We generate counterfactual images from the existing portrait with VAE, VAE-GAN, and the proposed CDVAE-GAN. Figure 7 lists the generated counterfactual images by controlling the 7 attributes bangs, black hair, blond hair, male (gender), mustache, mouth slightly open, and smiling of the test instances. In other words, once we obtain the latent representation z_G and the latent attribute z_B from test data x, the counterfactual image is generated by flipping the latent attribute $1 - z_B$, as we discussed in Section 3.3. For the VAE, the existence of the Bernoulli attribute random variables enables altering the corresponding attribute in the image. However, blurry images are generated in general due to the nature of VAE. Meanwhile, as in the VAE-GAN (Larsen *et al*., 2016), we can obtain clear images with high fidelity by adding the GAN discriminator after the VAE decoder. However, due to the indirect computing process in the counterfactual generation, it fails to reflect data attributes in general, and it shows more correlations among attributes, which is undesirable. Compared to the baselines, the proposed CDVAE-GAN successfully generates counterfactual images with high fidelity. It is because of the effect of the regularizer $D_{KL}(y||q_\phi(z_B|x))$ in the loss function, which (1) disentangles the attribute information z_B from the latent feature z_G; and (2) z_B is controllable as a latent variable in binary. We also provide qualitative results of the proposed CDVAE-GAN with high-resolution images of pixel size 128×128 in Figure 8.

Figure 8: *Counterfactual high-resolution image generation of the proposed CDVAE-GAN with (left) test data input for encoder E, and (right) random sample input for decoder/generator G.*

4.3. Quantitative result

We use two measures to compare the performance of CDVAE-GAN against the baselines quantitatively. The first measure is the Frechet inception distance (FID) score (Heusel *et al.*, 2017) which measures the quality and diversity of generated images. The FID score evaluates the divergence between the real and generated data distribution to measure how realistic the generated data are. Hence, a lower FID score is better. The detailed equation of computing the FID score is as follows:

$$d\left(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\mu',\Sigma')\right)^2 = \left\|\mu-\mu'\right\|_2^2 + \mathrm{tr}\left(\Sigma+\Sigma'-2\left(\Sigma^{\frac{1}{2}}\Sigma'\Sigma^{\frac{1}{2}}\right)^{\frac{1}{2}}\right), \tag{4.1}$$

where Gaussian distributions $\mathcal{N}(\mu,\Sigma)$ and $\mathcal{N}(\mu',\Sigma')$ are fitted from the real and generated datasets. Especially, we utilize pre-trained Inception-v3 (Szegedy *et al.*, 2016) to extract image feature vectors, and then compute the FID score by following the work of Heusel *et al.* (2017). Hence, the FID score is computed in the Inception-v3 image feature space. The proposed CDVAE-GAN marked the lowest FID score in the comparison, as listed in Table 2. The second measure is classification accuracy with

Table 2: FID score from counterfactually generated samples for the proposed CDVAE-GAN and the baselines

VAE	VAE-GAN	CDVAE-GAN
122.86	48.46	**34.24**

The lower is, the better the FID score. The best results are marked in **bold**.

Table 3: Attribute classification accuracy for the generated images from the proposed CDVAE-GAN and the baselines

Attribute	Train	Test	VAE	VAE-GAN	CDVAE-GAN
Bang	92.9%	87.2%	52.4%	66.6%	**73.79%**
Black hair	79.5%	78.9%	52.4%	68.2%	**70.03%**
Blond hair	85.9%	85.1%	60.1%	62.4%	**71.37%**
Male	88.3%	87.5%	52.2%	51.3%	**72.37%**
Mustache	90.4%	86.0%	81.9%	81.3%	**83.08%**
Mouth slightly open	77.9%	76.0%	55.8%	70.3%	**75.47%**
Smiling	89.9%	88.4%	72.9%	75.0%	**81.62%**

We also provide the accuracies from the train/test dataset for reference to the generated image quality. The best results are marked in **bold**.

an auxiliary neural network classifier to evaluate whether the generated image successfully reflects the counterfactual attribute or not. We choose to use neural network classifiers since (1) we can train end-to-end multiple attribute-wise classifiers simultaneously, and (2) the neural network classifiers are simple and powerful. The classification result is listed in Table 3. The classifier is trained with the training data, and we provide the test accuracy together to show its quality. CDVAE-GAN also outperforms the baselines in accuracy. Particularly, it marked similar accuracy compared to the test dataset in several attributes such as mustache or mouth slightly open, and it means that the generated images are enough to deceive the machine learning classifiers.

5. Conclusion

This paper proposes CDVAE-GAN for counterfactual data generation with disentangled representation learning. The proposed CDVAE-GAN consists of Gaussian VAE with additional Bernoulli latent variables and GAN, and the VAE and the GAN are competitively optimized through adversarial training. While the Gaussian VAE produces low-dimensional representation in the latent space, the Bernoulli latent variables disentangly capture the data attributes. Since each data attribute is separately propagated to the Bernoulli latent variables, we can manipulate the latent values by flipping zeros and ones. In that manner, we can generate non-existing counterfactual samples with the VAE decoder, i.e., the CDVAE-GAN generator. We conducted the experiment on the CelebA dataset, and the following attributes are utilized for the counterfactual sample generation: Bangs, black hair, blond hair, male (gender), mustache, mouth slightly open, and smiling. The visualized generated counterfactual samples are qualitatively realistic, and the additional neural network classifier also qualitatively supports that the generated samples successfully mimic the opposite attributes. Hence, the proposed model can be widely utilized for generating unseen data while in the training phase or non-existing data in the real world.

While the proposed CDVAE-GAN can generate counterfactual data based on the inputs, several limitations exist. First, only limited attributes can be accessed in the real world, even though there are numerous factors in observations. For example, while the CelebA dataset has multiple attributes that explain the data instances well, there can be more un-tagged attributes such as clothing style, clothing color, background color, etc. Second, hidden causal relationships exist between attributes, such as

gender → mustache, while the proposed CDVAE-GAN assumes all attributes are independent. Since these are beyond the scope of the proposed CDVAE-GAN, it results in an undesired phenomenon in the counterfactual generation, changing non-attribute factors such as clothing color or background color. Hence, in future work, we can further consider embedding the detailed causality factors among data attributes (Yang *et al*., 2021) in counterfactual data generation.

References

Bao J, Chen D, Wen F, Li H, and Hua G (2017). CVAE-GAN: Fine-grained image generation through asymmetric training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Venice, Italy, 2745–2754.

Dupont E (2018). Learning disentangled joint continuous and discrete representations, *Advances in Neural Information Processing Systems (NeurIPS)*, **31**, 710–720.

Gatys LA, Ecker AS, and Bethge M (2016). Image style transfer using convolutional neural networks, In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2414–2423.

Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, and Bengio Y (2014). Generative adversarial nets, *Advances in Neural Information Processing Systems (NeurIPS)*, **27**, 2672–2680.

Heusel M, Ramsauer H, Unterthiner T, Nessler B, and Hochreiter S (2017). GANs trained by a two time-scale update rule converge to a local nash equilibrium, *Advances in Neural Information Processing Systems (NeurIPS)*, **30**, 6626–6637.

Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, Mohamed S, and Lerchner A (2017). Beta-VAE: Learning basic visual concepts with a constrained variational framework, In *Proceedings of The International Conference on Learning Representations (ICLR)*, Toulon, France.

Jang E, Gu S, and Poole B (2017). Categorical reparameterization with Gumbel-Softmax. In *Proceedings of The International Conference on Learning Representations (ICLR)*, Toulon, France.

Joo W, Lee W, Park S, and Moon IC (2020a). Dirichlet variational autoencoder, *Pattern Recognition*, **107**, 107514.

Joo W, Kim D, Shin S, and Moon IC (2020b). Generalized Gumbel-Softmax gradient estimator for various discrete random variables, Available from: `https://arxiv.org/abs/2003.01847`

Kim D, Song K, Shin S, Kang W, Moon IC, and Joo W (2021). Neural posterior regularization for likelihood-free inference, Available from: `https://arxiv.org/abs/2102.07770`

Kingma DP and Welling M (2014). Auto-Encoding variational Bayes, In *Proceedings of The International Conference on Learning Representations (ICLR)*, Banff, Canada.

Kusner MJ, Loftus J, Russell C, and Silva R (2017). Counterfactual fairness, *Advances in Neural Information Processing Systems (NeurIPS)*, **30**, 4066–4076.

Larsen ABL, Sonderby SK, Larochelle H, and Winther O (2016). Autoencoding beyond pixels using a learned similarity metric, *In International Conference on Machine Learning (PMLR)*, **48**, 1558–1566.

Liu Z, Luo P, Wang X, and Tang X (2015). Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 3730–3738.

Nalisnick E and Smyth P (2017). Stick-Breaking variational autoencoders, *The International Conference on Learning Representations (ICLR)*.

Neal L, Olson M, Fern X, Wong WK, and Li F (2018). Open set learning with counterfactual images, In *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany,

613–628.

Radford A, Metz L, and Chintala S (2016). Unsupervised representation learning with deep convolutional generative adversarial networks, In *Proceedings of The International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico.

Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, and Chen X (2016). Improved techniques for training GANs, *Advances in Neural Information Processing Systems (NeurIPS)*, **29**, 2234–2242.

Sohn K, Lee H, and Yan X (2015). Learning structured output representation using deep conditional generative models, *Advances in Neural Information Processing Systems (NeurIPS)*, **28**, 3483–3491.

Szegedy C, Vanhoucke V, Ioffe S, Shlens J, and Wojna Z (2016). Rethinking the inception architecture for computer vision, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2818–2826.

Yang M, Liu F, Chen Z, Shen X, Hao J, and Wang J (2021). CausalVAE: Disentangled representation learning via neural structural causal models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 9588–9597.

Received February 19, 2023; Revised June 10, 2023; Accepted September 23, 2023