# Learning joint latent representations based on information maximization

Fei Ye and Adrian. G. Bors
Department of Computer Science, University of York, York YO10 5GH, UK

**Abstract**

Learning disentangled and interpretable representations is an important aspect of information understanding. In this paper, we propose a novel deep learning model representing both discrete and continuous latent variable spaces which can be used in either supervised or unsupervised learning. The proposed model is trained using an optimization function employing the mutual information maximization criterion. For the unsupervised learning setting we define a lower bound to the mutual information between the joint distribution of the latent variables corresponding to the real data and those generated by the model. The maximization of this lower bound during the training induces the learning of disentangled and interpretable data representations. Such representations can be used for attribute manipulation and image editing tasks.

*Keywords:* Disentangled learning, Variational Autoencoders (VAE), Generative Adversarial Nets (GAN), Representation learning, Mutual Information.

## 1. Introduction

Learning and using interpretable data representations is an area at the forefront of research in data analysis using deep learning. Learning meaningful variations of data is referred as learning disentangled representations. These are useful not only for exploring signal representations but also for understanding data generation processes. A disentangled representation, according to the definition from [1], is the effect obtained in the multidimensional data representation when changing one of the latent variables. Such actions lead to performing changes in the data corresponding to a specific factor of variation of the target data, while being relatively invariant to changes in the direction of the other factors. By controlling disentangled representations one can explore the hidden aspects of the latent space properties corresponding to multidimensional data.

Disentangled representations [2] have been obtained by adapting classification tasks or through reinforcement learning [3]. Meanwhile, the Variational Autoencoder (VAE) [4] aims to estimate a variable latent space representing the data. The latent space is modelled by employing an optimization function which simultaneously maximizes the log-likelihood of data reconstruction and minimizes the Kullback-Leibler (KL) divergence between the prior representing the data and their corresponding variational distribution. $\beta-$VAE [5] achieves unsupervised disentanglement by modulating the contribution of the KL divergence term in the objective function with a factor $\beta$. This encourages modifying the gap between the variational distributions and their priors leading to disentanglement in the generated data. However, image modelling based on VAEs results in poor, rather blurred image reconstructions. Moreover, VAE-based models would consider only representing continuous variables which has limitations when modelling discrete modes of data variation.

Other well known generative deep learning methods are the Generative Adversarial Networks (GAN) [6]. GANs generate data based on a min-max optimization game. In InfoGAN [7] interpretable representations are achieved by increasing the mutual information (MI) between a subset of latent codes and the observed data. However, InfoGAN is lacking an accurate inference mechanism for the observed data. In addition to unsupervised learning, there are many attempts to explore supervised or semi-supervised learning of disentangled representations [8, 9, 10, 11]. Although some of these approaches can achieve good results, they

would usually require specific *a priori* knowledge about the underlying disentangled representations they are learning. In this paper, we study how to learn a joint latent representation that can capture both discrete and continuous variations of data, which was not attempted by most existing methods. Furthermore, existing disentangled representation methods are based on VAEs and therefore lead to a poor quality of generated results. This motivates us to propose a novel hybrid approach utilizing the inference mechanism of VAEs for learning disentangled representations and the powerful generation ability of GANs.

The following contributions are brought by this research study:

- We propose a novel approach for learning disentangled representations which incorporates additional inference capabilities in order to learn meaningful discrete and continuous variables. We also introduce an adversarial component in the loss function of the proposed generative model in order to encourage the modelling of the joint distribution of data and latent variables.
- We introduce a lower bound, based on the mutual information between the distributions of discrete and continuous latent variables, which is maximized in order to enforce disentanglement in the latent space. By enabling disentanglement in such spaces we are able to edit images and model a variety of specific variations by manipulating interpolations in the latent space. These capabilities can lead to generating images defined by specific properties, influencing their appearance by simulating physical and geometric factors among others.
- We show, through experiments on several databases, that the proposed approach, called InfoVAEGAN, is able to learn meaningful discrete and continuous latent representations under the supervised and unsupervised learning settings.

The rest of the paper is organised as follows. In Section 2 we provide an overview of the relevant literature. Supervised and unsupervised disentangled learning are discussed in Sections 3 and 4, respectively. The experimental results for the proposed frameworks are provided in Section 5 and the conclusions are drawn in Section 6.

## 2. Background and related work

In this section, we review the background of deep generative learning models and provide a brief overview of the disentangled representation learning field.

### *2.1. Variational Autoencoder (VAE)*

Variational autoencoders (VAEs) represent one of the most popular deep generative models. A VAE consists of two component networks: encoder and decoder. Let us consider a given database $\{\mathcal{X} \mid \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \in \mathcal{X}\}$. A latent space $\mathcal{Z}$ is defined by the stochastic latent vectors $\{\mathcal{Z} \mid \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N \in \mathcal{Z}\}$. $\mathcal{Z}$ can be the result of maximizing the average marginal log-likelihood $\sum_{i=1}^{N} \log p(\mathbf{x}_i)/N$. However, such an optimization procedure is intractable. Variational inference was introduced, aiming to obtain through optimization a lower bound on the marginal likelihood. While the optimization loss function used for training a VAE, is represented by maximizing the reconstruction error, at the same time it aims to minimize the Kullback-Leibler (KL) divergence between the prior $p(\mathbf{z})$ and its approximation $q_\theta(\mathbf{z}|\mathbf{x})$, defined by:

$$L(\phi, \theta) = \mathbb{E}_{q_\theta(\mathbf{z}|\mathbf{x})}[\log p_\phi(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\theta(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \leq \log p(\mathbf{x}), \tag{1}$$

where $q_\theta(\mathbf{z}|\mathbf{x})$ is the variational distribution, approximating $p(\mathbf{z})$, implemented by a neural network defined by parameters $\theta$. Meanwhile, $p_\phi(\mathbf{x}|\mathbf{z})$ represents the probability of data reconstruction by the decoder network which is parameterized by a neural network of parameters $\phi$. One efficient approach for optimizing this lower bound consists of using the reparameterization trick [4] for sampling from the variational distribution.

*2.2. Generative Adversarial Network (GAN)*

The GAN model [6], also consists of two component networks: generator and discriminator. The former learns to generate data matching the real data distribution while the second is the discriminator which distinguishes real data samples from fake (the data produced by the generator). These two networks are trained alternatively, which can be seen as playing a min-max game, where the discriminator network is trained to receive both fake and real data samples while trying to distinguish one from another. The GAN loss function is defined as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim pd(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})} \log[1 - D(G(\mathbf{z}))]] \tag{2}$$

where $G$ denotes the generator and $D$ is the discriminator network.

*2.3. Supervised learning of disentangled representations*

Supervised learning is based on the notion of latent variable predictability when labels are available [12]. Sohn *et al.* [13] employed conditional VAEs to learn interpretable representations. In this approach, the generator distribution is conditioned on the probability $p(\mathbf{z}, \mathbf{y})$ characterizing the joint distribution of latent variables $\mathbf{z}$ and data' labels $\mathbf{y}$. Ganin et al. [14] introduced an additional classifier, predicting $\mathbf{y}$ from the latent variables $\mathbf{z}$ inferred by the encoder of a VAE. Meanwhile, the classifier is trained to remove the information correlated with the class label $\mathbf{y}$, from the probabilistic representation of the latent variables, $p(\mathbf{z})$. This would encourage $p(\mathbf{z})$ to capture the properties of the data which are not related to the class label $\mathbf{y}$. Klys *et al.* [15] employed the same idea in order to enforce the independence between the class labels $\mathbf{y}$ and the representation variables $\mathbf{z}$.

*2.4. Unsupervised disentangled learning*

Learning a disentangled representation in an unsupervised manner is a challenging task. Early research work would focus on factorial coding. Given the statistical properties of the inputs from the environment, the approach from [16] sought to find invertible internal representations such that the occurrence of a symbol for a given code is independent of all others. The method from [17] employs the Spike-and-Slab Boltzmann Machines to disentangle multiplicative factors of variation via inference in a generative model. Lately, generative methods such as GANs and VAEs, based on latent learning models, have achieved better results in obtaining disentangled representations. $\beta$-VAE [5] is an unsupervised disentangled learning method, which modifies the objective function of the classical VAE (1), by setting a penalty $\beta \geq 1$ on the KL divergence term as:

$$L(\phi, \theta) = \mathbb{E}_{q_\theta(\mathbf{z}|\mathbf{x})}[\log p_\phi(\mathbf{x}|\mathbf{z})] - \beta D_{KL}(q_\theta(\mathbf{z}|\mathbf{x})||p(\mathbf{z})), \tag{3}$$

where $\beta$ controls the balance between reconstruction fidelity and the degree of disengagement among the latent variables. To understand why increasing $\beta$ would help disengagements, some studies [18, 19] have decomposed the KL divergence term into two components:

$$\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[D_{KL}(q_\theta(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] = I(\mathbf{x}, \mathbf{z}) + D_{KL}(q_\theta(\mathbf{z})||p(\mathbf{z})), \tag{4}$$

where $I(\mathbf{x}, \mathbf{z})$ is the mutual information (MI) between latent variables $\mathbf{z}$ and observed data $\mathbf{x}$. Minimizing the latter term from equation (4) encourages disengagements in the latent variable space. However, this would also lead to the reduction of the mutual information $I(\mathbf{x}, \mathbf{z})$. In practice, this can lead to larger reconstruction errors during the training.

Recently, various methods have been developed to induce disentangled representations without sacrificing the reconstruction quality [20, 21]. For instance, Burgess *et al.* [22] provides the analysis on the emergence of disentangled representations when training $\beta$-VAEs, and proposed a new training procedure which progressively increases the penalty $\beta$ for the KL divergence term (see equation (3)), in order to balance the data disengagement and reconstruction accuracy. Gao *et al.* [23], propose to use multivariate mutual information in order to help find interpretable representations in VAEs. Kim *et al.* [18], proposed a VAE-based disentangled learning model, namely FactorVAE, which encourages the independence in the distribution of

latent representations which is also factorial over all dimensions, by penalizing the total correlation in the VAE objective function. A similar idea was also employed in [24], where the authors demonstrated that the total correlation is the most important penalty in the VAE objective function for inducing disentangled representations. Dupont *et al.* [25], proposed to learn a joint distribution of continuous and discrete latent variables using VAEs in order to discover disentangled and interpretable representations. This joint distribution VAE model would gradually increase the penalty of two different KL divergence terms from the VAE objective function in order to balance the disentanglement and reconstruction quality. A GAN model, called Adversarial Learned Inference (ALI) was used to jointly model the distribution of data and latent variables in [26]. InfoGAN [7] is another unsupervised disentangled learning method, which estimates a subset of codes in order to capture meaningful representations. InfoGAN derives a lower bound on the MI between the latent codes and the generated data which is added as an additional term in the objective function.

### *2.5. Hybrid VAE-GAN models*

Hybrid VAE-GAN models are a promising generative modelling methodology. The main idea of the hybrid model is to learn an inference model in GANs in order to provide inference mechanisms. Adversarial learning is performed onto the hybrid model to match either the data distribution [27], the latent distribution [28] or their joint distributions [26, 29]. VAEGAN [27] combines the adversarial loss and VAEs loss into a unified objective function in order to align better the generated data and real data distributions. The Lifelong VAEGAN was used for learning successively several databases in [30].

This paper is the first to propose a novel hybrid model for learning disentangled representations.

## 3. Supervised disentangled learning using the mutual information criterion

In this section, we describe a new supervised learning framework for disentangled representations based on the mutual information criterion (MI). MI is used for associating unspecified characteristic factors from the latent spaces with class labels in the case of supervised learning.

### *3.1. Learning continuous and discrete latent spaces*

Let us consider a dataset consisting of $N$ labelled data samples $\{\mathcal{X} \mid \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \in \mathcal{X}\}$. Each data sample belongs to one of $K$ classes, defined by labels $\{\mathcal{Y} \mid y_1, \ldots, y_K \in \mathcal{Y}\}$. The goal of the supervised learning framework is to learn a generative image model in which the discrete and continuous representations can be separated into distinct latent spaces. Such a setting can allow us to easily generate specific variations in images by manipulating the learnt latent space. Specifically, the continuous latent variables capture underlying factors across all discrete variations, characterized by smooth transitions in the latent space from one generated image to another.

In the following we consider the VAE component of a hybrid learning model. Learning in the VAE model requires maximizing a variational lower bound on the marginal log-likelihood for the data as follows:

$$\log p(\mathbf{x}) = \log \int\int p(\mathbf{x}, \mathbf{z}, \mathbf{s})\, d\mathbf{z} d\mathbf{s} \geqslant \mathbb{E}_{q(\mathbf{z},\mathbf{s}|\mathbf{x})} \log \left[ \frac{p(\mathbf{x}, \mathbf{z}, \mathbf{s})}{q(\mathbf{z}, \mathbf{s}|\mathbf{x})} \right] = L_{\text{ELBO}}, \tag{5}$$

where the right hand side is called the evidence lower bound (ELBO), which is maximized in our optimization. $q(\mathbf{z}, \mathbf{s}|\mathbf{x})$ is the variational posterior for the joint continuous and discrete spaces, represented by the latent variables $\mathbf{z}$ and $\mathbf{s}$, respectively, aiming to approximate the true posterior $p(\mathbf{z}, \mathbf{s}|\mathbf{x})$. $L_{\text{ELBO}}$ is the loss associated with the evidence lower bound, which is rewritten as :

$$\begin{aligned} L_{\text{ELBO}}(\delta, \theta) &= \mathbb{E}_{q_\delta(\mathbf{z},\mathbf{s}|\mathbf{x})} \left[ \log \frac{p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{s}) p(\mathbf{z}) p(\mathbf{s})}{q_\delta(\mathbf{z}, \mathbf{s}|\mathbf{x})} \right] = \mathbb{E}_{q_\delta(\mathbf{z},\mathbf{s}|\mathbf{x})} \left[ \log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{s}) \right] + \mathbb{E}_{q_\delta(\mathbf{z},\mathbf{s}|\mathbf{x})} \left[ \log \frac{p(\mathbf{z}) p(\mathbf{s})}{q_\delta(\mathbf{z}|\mathbf{x}) q_\delta(\mathbf{s}|\mathbf{x})} \right] \\ &= \mathbb{E}_{q_\delta(\mathbf{z},\mathbf{s}|\mathbf{x})} \left[ \log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{s}) \right] + \mathbb{E}_{q_\delta(\mathbf{z},\mathbf{s}|\mathbf{x})} \left[ \log \frac{p(\mathbf{z})}{q_\delta(\mathbf{z}|\mathbf{x})} \right] + \mathbb{E}_{q_\delta(\mathbf{z},\mathbf{s}|\mathbf{x})} \left[ \log \frac{p(\mathbf{s})}{q_\delta(\mathbf{s}|\mathbf{x})} \right] \\ &= \mathbb{E}_{q_\delta(\mathbf{z},\mathbf{s}|\mathbf{x})} \left[ \log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{s}) \right] - D_{KL} \left( q_\omega(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}) \right) - D_{KL} \left( q_\delta(\mathbf{s}|\mathbf{x}) || p(\mathbf{s}) \right), \end{aligned} \tag{6}$$

where we consider the independence between the latent variables $\mathbf{z}$ and $\mathbf{s}$, as $q_\delta(\mathbf{z},\mathbf{s}|\mathbf{x}) = q_\delta(\mathbf{z}|\mathbf{x})q_\delta(\mathbf{s}|\mathbf{x})$. The first term from the right side of equation (6) corresponds to the reconstruction error of the given training data space, the second term represents the KL divergence between the parameterized and empirical distributions associated with continuous variables $\mathbf{z}$, and the third represents the KL divergence between the categorical distribution and the empirical representation of the discrete variables $\mathbf{s}$, respectively. The approximate posteriors $q_\omega(\mathbf{z}|\mathbf{x})$ and $q_\delta(\mathbf{s}|\mathbf{x})$, defined for continuous and discrete variables, are modelled by separate encoders implemented by neural networks defined by the parameters $\omega$ and $\delta$, respectively. The posterior of the data is considered as a Gaussian distribution with diagonal covariance matrix for defining the probability density function (pdf) of the continuous data. The latent variables $\mathbf{z}$ are sampled using the reparameterisation trick as :

$$\mathbf{z} = \mu + \xi \odot \sigma, \;\; \xi \sim \mathcal{N}(0, \mathbf{I}), \tag{7}$$

where $\mu$ and $\sigma$ are latent variables, characterizing the mean and the absolute deviation, provided by the encoder, and $\xi$ is sampled from the Normal distribution. In practice, we can obtain a discrete representation by implementing a Softmax layer on the top of the encoder. Instead, we choose to use two distinct encoders to represent the variational distributions $q_\delta(\mathbf{s}|\mathbf{x})$ and $q_\omega(\mathbf{z}|\mathbf{x})$, respectively, while a single decoder is used for embedding both continuous and discrete latent variables.

### *3.2. Using the Gumble-Softmax distribution for sampling discrete data*

A Softmax layer normally considered for a Convolution Neural Network (CNN) cannot generate discrete variables, and consequently cannot approximate the true posterior for the discrete latent variables $\mathbf{s}$. In the following, we employ the Gumble-Softmax distribution in order to ensure the diferentiability of the generator for producing discrete variables, thus ensuring an end-to-end training in the proposed deep learning framework. We can draw the discrete representation $\mathbf{y}' = \{y'_1, \ldots, y'_K\}$ of a sample $\mathbf{s}$ by using the Gumble-Softmax trick, [31, 32] :

$$y'_t = \frac{\exp((\log s_t + g_t)/T)}{\sum_i^K \exp((\log s_i + g_i)/T)}, \tag{8}$$

where $g_i$ is a sample drawn from Gumble$(0, 1)$. Our encoder actually outputs a probability vector $\mathbf{s} = (s_1, \ldots, s_K)$. The Gumble-Softmax trick, represents a continuous, diferentiable approximation to the arg max expression, replacing the Softmax function, [33]. $T$ is the temperature parameter and $t = 1, \ldots, K$. If $T$ is large, then the variables generated by the Gumble-Softmax distribution would become uniformly distributed and would no longer characterize one-hot vectors. In contrast, if the temperature parameter approaches zero, the Gumble-Softmax distribution becomes a one-hot encoded categorical distribution.

### *3.3. Employing the mutual information criterion for disentanglement*

In the following we propose to use the Mutual Information (MI) criterion in order to enforce the learning of class-specific attributes and to embed characteristic factors into distinct latent spaces. This would enable the ability to manipulate the generation of data defined by certain characteristics.

First, we consider minimizing the cross entropy $\mathrm{L_C}$ between the class label $\mathbf{y}$ and the discrete latent variables $\mathbf{s}$ in order to obtain an accurate posterior $q_\delta(\mathbf{s}|\mathbf{x})$ for the given data $\mathbf{x}$. This will increase the mutual information between the class label $\mathbf{y}$ and the discrete latent variables $\mathbf{s}$ inferred by the encoder. Furthermore, we consider that the mutual information $\mathrm{I}(\mathcal{Y}, \mathcal{Z})$ should be minimized in order to enforce the independence between $\mathbf{y}$ and $\mathbf{z}$. This optimization procedure would emphasize specific properties in the generated data, which do not depend explicitly on the class label $\mathbf{y}$ to be embedded into the continuous latent spaces. We can rewrite the MI depending on the data and conditional entropies, as in [15]:

$$\mathrm{MI}(\mathcal{Y}, \mathcal{Z}) = H(\mathcal{Y}) - H(\mathcal{Y}|\mathcal{Z}) = H(\mathcal{Y}) + \int\int\int p(\mathbf{x})p(\mathbf{z}|\mathbf{x})p(\mathbf{y}|\mathbf{z}) \log p(\mathbf{y}|\mathbf{z})\, d\mathbf{y}d\mathbf{z}d\mathbf{x} \tag{9}$$

where the first term represents the entropy of $\mathcal{Y}$ and can be seen as a constant, while the second term is the conditional entropy which is rewritten as the integral from the last expression of equation (9). Similarly to the approach from [15], we propose to use an approximate posterior $q_\phi(\mathbf{y}|\mathbf{z})$, modelled by a variational

encoder of parameters $\phi$, aiming to approximate $p(\mathbf{y}|\mathbf{z})$, and $q_\delta(\mathbf{z}|\mathbf{x})$ which is implemented by a variational encoder of parameters $\delta$, for approximating the conditional entropy:

$$-H(\mathcal{Y}|\mathcal{Z}) = \mathbb{E}_{\mathbf{x}\sim\mathcal{X}}\left[\int\int q_\delta(\mathbf{z}|\mathbf{x}) q_\phi(\mathbf{y}|\mathbf{z}) \log q_\phi(\mathbf{y}|\mathbf{z})\, d\mathbf{y} d\mathbf{z}\right] \tag{10}$$

We rewrite this expression by considering the expectation of $q_\delta(\mathbf{z}|\mathbf{x})$ :

$$-H(\mathcal{Y}|\mathcal{Z}) = \mathbb{E}_{q_\delta(\mathbf{z}|\mathbf{x}),\mathbf{x}\sim\mathcal{X}}\left[\int q_\phi(\mathbf{y}|\mathbf{z}) \log q_\phi(\mathbf{y}|\mathbf{z})\, d\mathbf{y}\right] = \mathrm{L_{MI}}, \tag{11}$$

where the expectation is taken over the entire dataset $\mathbf{x} \sim \mathcal{X}$ and over the approximate class posterior $q_\delta(\mathbf{z}|\mathbf{x})$. In order to solve (11) we need to know the exact approximate posterior $q_\phi(\mathbf{y}|\mathbf{z})$ which is optimized by :

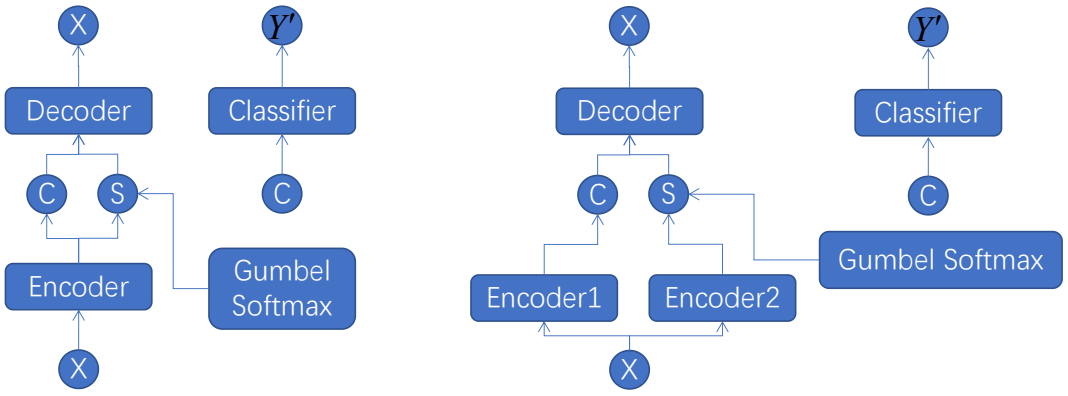
$$\mathrm{L}_C = \mathbb{E}_{q_\delta(\mathbf{z}|\mathbf{x}),\mathbf{x}\sim\mathcal{X},\mathbf{y}\sim\mathcal{Y}}[q_\phi(\mathbf{y}|\mathbf{z})]. \tag{12}$$

Consequently, our main training procedure consists of simultaneously optimizing two loss functions :

$$\min_{\delta,\phi}[\rho_1 \mathrm{L_C}(\delta,\phi) + \rho_2 \mathrm{L_{MI}}(\delta,\phi)] \tag{13}$$

$$\max_{\delta}[\mathrm{L_C}(\delta)] \tag{14}$$

where $\rho_1$, $\rho_2$ are the hyperparameters which control the relative strength of each entropy term. Optimizing $\mathrm{L_{MI}}$ and $\mathrm{L_C}$ from equations (13) and (14), respectively, can be seen as a min-max adversarial learning procedure, where the classifier $q_\phi(\mathbf{y}|\mathbf{z})$ tries to predict the class label $\mathbf{y}$ for the given latent variable $\mathbf{z}$, while the encoder $q_\delta(\mathbf{z}|\mathbf{x})$ aims to yield the latent variable $\mathbf{z}$ which decreases the accuracy of the classifier. Two different structures for the supervized learning framework are shown in Figure 1. In the first structure, from Figure 1(a), the encoder distributions $q_\delta(\mathbf{z}|\mathbf{x})$ and $q_\delta(\mathbf{y}|\mathbf{x})$ share the same parameters $\delta$. In the second structure, from Figure 1(b), we consider two separate encoders to model the approximate posteriors $q_\delta(\mathbf{z}|\mathbf{x})$ and $q_\phi(\mathbf{y}|\mathbf{x})$.




(a) Structure with single encoder.

(b) Structure with dual encoder, for continuous and discrete variables, respectively.

Figure 1: Supervised learning structures, where $\mathbf{c}$ and $\mathbf{s}$ refer to the continuous and discrete variables, respectively, while $\mathbf{y}'$ denotes the predicted class label.

## 4. Unsupervised disentangled learning

In this section, we propose a novel unsupervised learning framework which can discover disentangled representations by using an accurate inference network.

*4.1. Sampling from the prior distributions*

In the following we consider two prior distributions for a given empirical data distribution: one characterizing continuous random variables and another for discrete (one-hot) random vectors. The distribution characterizing continuous variables $\mathbf{c}$ is Gaussian, $p(\mathbf{c}) = \mathcal{N}(\mu, \sigma^2)$, while for categorical variables $\mathbf{s}$, characterizing one-hot vectors, is considered as Multinomial, $p(\mathbf{s}) = Cat(k = K, p = 1/K)$, where $K$ is the dimension of the discrete vector, which is equal to the number of classes in the given dataset. Knowing prior information about the data, certainly has many advantages. For instance, the prior information can be used for driving the generation process such that we can easily measure the amount of information correlated with the prior distribution when observing generative results. This procedure can lead to developing mechanisms for generating images with attributes specific to the prior.

Let us consider the following generation process :

$$\mathbf{x} \sim q_\theta(\mathbf{x}|\mathbf{z}, \mathbf{s}, \mathbf{c}) \tag{15}$$

where $\mathbf{c} \sim p(\mathbf{c})$, $\mathbf{s} \sim p(\mathbf{s})$, represent the continuous and discrete variables, while $\mathbf{z} \sim p(\mathbf{z})$ is treated as random noise with its distribution $p(\mathbf{z})$, defined as a Gaussian with diagonal covariance matrix. The distribution $q_\theta(\mathbf{x}|\mathbf{z}, \mathbf{s}, \mathbf{c})$ is modelled by a decoder defined by the network of parameters $\theta$. We also introduce an inference network $p_\phi(\mathbf{c}, \mathbf{s}|\mathbf{x})$ defined by the parameters $\phi$, which can be seen as a partial inverse mapping of the decoder. We consider the adversarial learning procedure, similar to that used for GAN networks [6], in order to train both the encoder and decoder and this choice is justified in the next section. The adversarial loss is then defined as that for a GAN network :

$$\begin{aligned}\min_{G,E}\max_{D} V(G,E,D) =& \mathbb{E}_{\mathbf{x}\sim p_d(\mathbf{x})}[\mathbb{E}_{\mathbf{c}\sim p_\phi(\mathbf{c}|\mathbf{x}),\mathbf{s}\sim p_\phi(\mathbf{s}|\mathbf{x})}[\log D(\mathbf{x},\mathbf{c},\mathbf{s})]] + \\ & \mathbb{E}_{\mathbf{c}\sim p(\mathbf{c}),\mathbf{s}\sim p(\mathbf{s}),\mathbf{z}\sim p(\mathbf{z})}[\mathbb{E}_{\mathbf{x}\sim q_\theta(\mathbf{x}|\mathbf{z},\mathbf{s},\mathbf{c})}[\log(1 - D(\mathbf{x},\mathbf{c},\mathbf{s}))]] = \mathrm{L}_{\mathrm{GAN}}\end{aligned} \tag{16}$$

where $p_d(\mathbf{x})$ represents the empirical probability of the data, $G$, $E$ and $D$ are the generator, encoder and discriminator component networks, respectively.

Equation (16) is similar to those used in unsupervised adversarial approaches such as BiGAN [34] and ALI [26]. In our approach we introduce an auxiliary classifier in order to output binary predictions for distinguishing between two different joint distributions. For the first approach we consider the joint distribution of real images together with generated (fake) images defined by the latent variables inferred by the encoder. The learning structure is presented in Figure 2(a).

**Proposition 1.** For the encoder $E$ and generator $G$, the optimal discriminator $D$ is given by

$$D^*_{EG}(\mathbf{x}, \mathbf{c}, \mathbf{s}) = \frac{p(\mathbf{x}, \mathbf{c}, \mathbf{s})}{p(\mathbf{x}, \mathbf{c}, \mathbf{s}) + q(\mathbf{x}, \mathbf{c}, \mathbf{s})} \tag{17}$$

where the probabilities of data $p(\mathbf{x})$, continuous variables $p(\mathbf{c})$, and discrete variables $p(\mathbf{s})$, are considered as independent from each other :

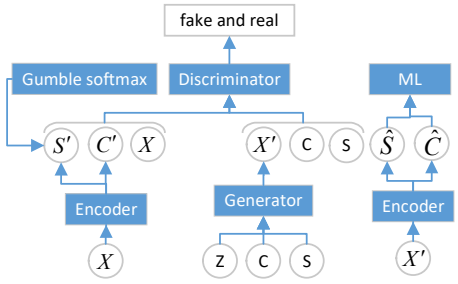
$$p(\mathbf{x}, \mathbf{c}, \mathbf{s}) = p(\mathbf{x})p(\mathbf{c})p(\mathbf{s}), \tag{18}$$

while the generator is characterized by :

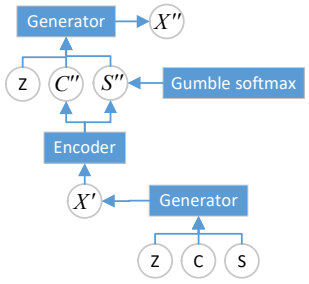
$$q(\mathbf{x}, \mathbf{c}, \mathbf{s}) = \int q(\mathbf{x}|\mathbf{c}, \mathbf{s}, \mathbf{z})p(\mathbf{c}, \mathbf{s})p(\mathbf{z})d\mathbf{z}. \tag{19}$$

*Proof.* The discriminator is used to maximize $V(G, E, D)$ and from (16) we have:

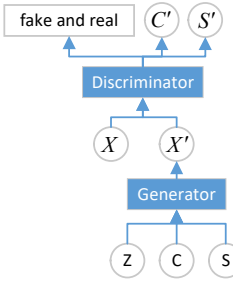
$$\begin{aligned} V(G,E,D) &= \int\int\int p(\mathbf{x})p(\mathbf{c})p(\mathbf{s})\log(D(\mathbf{x},\mathbf{c},\mathbf{s}))d\mathbf{x}d\mathbf{c}d\mathbf{s} \\ &+ \int\int\int p(\mathbf{z})p(\mathbf{c})p(\mathbf{s})q(\mathbf{x}|\mathbf{c},\mathbf{s},\mathbf{z})\log(1 - D(G(\mathbf{z},\mathbf{c},\mathbf{s}),\mathbf{c},\mathbf{s}))d\mathbf{x}d\mathbf{z}d\mathbf{c}d\mathbf{s} \\ &= \int\int\int p(\mathbf{x},\mathbf{c},\mathbf{s})\log(D(\mathbf{x},\mathbf{c},\mathbf{s}))d\mathbf{x}d\mathbf{c}d\mathbf{s} \\ &+ \int\int\int q(\mathbf{x},\mathbf{c},\mathbf{s})\log(1 - D(\mathbf{x},\mathbf{c},\mathbf{s}))d\mathbf{x}d\mathbf{c}d\mathbf{s} \end{aligned} \tag{20}$$



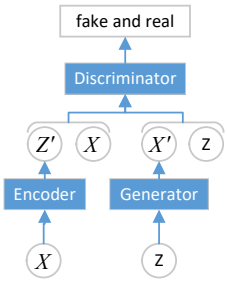

(a) The proposed model for GAN learning




(b) The proposed model for VAE learning




(c) InfoGAN [7]




(d) ALI [26]

Figure 2: The proposed unsupervised learning structure, together with the presentation of the diagrams for InfoGAN [7] and the Adversarial Learned Inference (ALI) [26], where $\mathbf{c}$ and $\mathbf{s}$ refer to the continuous and discrete variables, respectively, and $\mathbf{z}$ is random Gaussian noise.

where we use equations (18) and (19) and we used $\mathbf{x} \sim G(\mathbf{z}, \mathbf{c}, \mathbf{s})$.

We consider the following observation. For any $(w, r) \in \mathbb{R}^2$, the function

$$f(y) = w \log(y) + r \log(1 - y) \tag{21}$$

achieves the maximum for $y = \frac{w}{w+r}$ which when applied to (20) results in the expression from equation (17), which proves Proposition 1.

**Proposition 2.** The objective function $V(G, E, D)$ for the optimal discriminator $D^*_{EG}$ can be redefined considering the Jensen-Shannon divergence between $p(\mathbf{x}, \mathbf{c}, \mathbf{s})$ and $q(\mathbf{x}, \mathbf{c}, \mathbf{s})$.

*Proof.* We can reformulate the objective function $V(G, E, D)$ from (16) as :

$$\begin{aligned} C(E, G) = \max_D V(G, E, D) &= E_{p(\mathbf{x},\mathbf{c},\mathbf{s})}[\log D^*_{EG}(\mathbf{x}, \mathbf{c}, \mathbf{s})] + E_{q(\mathbf{x},\mathbf{c},\mathbf{s})p(\mathbf{z})}[1 - \log D^*_{EG}(G(\mathbf{c}, \mathbf{s}, \mathbf{z}), \mathbf{c}, \mathbf{s})] \\ &= E_{p(\mathbf{x},\mathbf{c},\mathbf{s})}[\log D^*_{EG}(\mathbf{x}, \mathbf{c}, \mathbf{s})] + E_{q(\mathbf{x},\mathbf{c},\mathbf{s})}[1 - \log D^*_{EG}(\mathbf{x}, \mathbf{c}, \mathbf{s})] \\ &= E_{p(\mathbf{x},\mathbf{c},\mathbf{s})}\left[\log \frac{p(\mathbf{x}, \mathbf{c}, \mathbf{s})}{p(\mathbf{x}, \mathbf{c}, \mathbf{s}) + q(\mathbf{x}, \mathbf{c}, \mathbf{s})}\right] + E_{q(\mathbf{x},\mathbf{c},\mathbf{s})}\left[\log \frac{q(\mathbf{x}, \mathbf{c}, \mathbf{s})}{p(\mathbf{x}, \mathbf{c}, \mathbf{s}) + q(\mathbf{x}, \mathbf{c}, \mathbf{s})}\right] \end{aligned} \tag{22}$$

where we apply the consequences of Proposition 1 and expression (17). Then the above equation can be rewritten considering the Jensen-Shannon (JS) divergence $D_{JS}$ between the two distributions :

$$C(E, G) = 2\mathrm{D}_{\mathrm{JS}}[(\mathrm{p}(\mathbf{x}, \mathbf{c}, \mathbf{s})||\mathrm{q}(\mathbf{x}, \mathbf{c}, \mathbf{s}))]. \tag{23}$$

**Theorem 1.** The global minimum for $C(E, G)$ is obtained if and only if $p(\mathbf{x}, \mathbf{c}, \mathbf{s}) = q(\mathbf{x}, \mathbf{c}, \mathbf{s})$ and the optimal minimum is achieved for $C(E, G) = -\log 4$.

**Proof.** For $p(\mathbf{x}, \mathbf{c}, \mathbf{s}) = q(\mathbf{x}, \mathbf{c}, \mathbf{s})$ from Proposition 1, $D^*_{EG}(\mathbf{x}, \mathbf{c}, \mathbf{s}) = 1/2$, and considering Proposition 2 and equation (22), we have:

$$\mathbb{E}_{p(\mathbf{x},\mathbf{c},\mathbf{s})}[-\log 2] + \mathbb{E}_{q(\mathbf{x},\mathbf{c},\mathbf{s})}[-\log 2] = -\log 4. \tag{24}$$

*4.2. Mutual information (MI) maximization for disentanglement*

In information theory, mutual information is used for measuring the amount of information inferred by one random variable when observing another one. In the proposed model we aim to preserve the information represented by both continuous and discrete latent variables during the encoder inference and generation processes.

Let us denote $(\mathbf{s}, \mathbf{c})$ as the joint discrete and continuous variable, which captures meaningful representations of generated images. During the decoding, we treat the variable $\mathbf{z}$ as the noise and we desire to allow the discrete variable $\mathbf{s}$ and the continuous variable $\mathbf{c}$ to capture the discrete and continuous variations of generated images. In order to achieve this, we introduce to maximize the mutual information between the joint latent variable $\mathbf{u}$ and the data generated by the decoder, $\mathrm{I}(\mathbf{u}, \mathrm{G}(\mathbf{x}, \mathbf{u}))$. Similar mutual information objectives have been successfully adopted by other research studies [7, 35]. However, the mutual information is hard to optimize directly since it requires the true posterior $p(\mathbf{u}|\mathbf{x})$ which is not available. In order to address this problem we define an auxiliary distribution $W(\mathbf{u}|\mathbf{x})$ to approximate the true posterior and derive a lower bound on the mutual information $\mathrm{I}(\mathbf{u}, \mathrm{G}(\mathbf{z}, \mathbf{u}))$, expressed as depending on the entropy of latent variables $H(\mathbf{u})$ and on the conditional entropy, $H(\mathbf{u}|G(\mathbf{z}, \mathbf{u}))$ :

$$\begin{aligned}
\mathrm{I}(\mathbf{u}, \mathrm{G}(\mathbf{z}, \mathbf{u})) &= H(\mathbf{u}) - H(\mathbf{u}|G(\mathbf{z}, \mathbf{u})) = \int\int G(\mathbf{z}, \mathbf{u}) p(\mathbf{u}|\mathbf{x}) \log p(\mathbf{u}|\mathbf{x}) \frac{W(\mathbf{u}|\mathbf{x})}{W(\mathbf{u}|\mathbf{x})} \, d\mathbf{x} d\mathbf{u} + H(\mathbf{u}) \\
&= \int\int G(\mathbf{z}, \mathbf{u}) p(\mathbf{u}|\mathbf{x}) \log \frac{p(\mathbf{u}|\mathbf{x})}{W(\mathbf{u}|\mathbf{x})} \, d\mathbf{x} d\mathbf{u} + \int\int G(\mathbf{z}, \mathbf{u}) p(\mathbf{u}|\mathbf{x}) \log W(\mathbf{u}|\mathbf{x}) d\mathbf{x} d\mathbf{u} + H(\mathbf{u}) \\
&= \mathbb{E}_{\mathbf{x}\sim G(\mathbf{z},\mathbf{u})} D_{KL}[p(\mathbf{u}|\mathbf{x})||W(\mathbf{u}|\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim G(\mathbf{z},\mathbf{u})}[\mathbb{E}_{\mathbf{u}\sim p(\mathbf{u},\mathbf{x})}[\log W(\mathbf{u}|\mathbf{x})]] + H(\mathbf{u}) \\
&\geqslant \mathbb{E}_{\mathbf{x}\sim G(\mathbf{z},\mathbf{u})}[\mathbb{E}_{\mathbf{u}\sim p(\mathbf{u},\mathbf{x})}[\log W(\mathbf{u}|\mathbf{x})]] + H(\mathbf{u}) = \mathrm{L}_{\mathrm{MI}},
\end{aligned} \tag{25}$$

where the auxiliary distribution $W(\mathbf{u}|\mathbf{x})$ is implemented by the encoder, and we consider that the KL divergence is positive or at least equal to 0, and where $\mathrm{L}_{\mathrm{MI}}$ denotes the objective function based on the mutual information loss. We consider $H(\mathbf{u})$ as a constant for the sake of simplicity. While on one hand the maximization of the mutual information can help learn interpretable representations, on the other hand it can not ensure the derivation of an accurate inference network.

*4.3. Learning by means of an accurate inference network*

Possessing an inference network provides many advantages for a data generative model. For instance, an inference network would allow our model to generate data with certain specific characteristics, or to transfer meaningful variations of an image to another without using any supervision mechanism. In this paper, we propose to change the lower bound on the marginal log-likelihood of the generator distribution in order to derive an accurate inference network.

Let us consider the generator as a probabilistic machine defined by the distribution $q(\mathbf{x}) = G(\mathbf{x}|\mathbf{z}, \mathbf{s}, \mathbf{c})$. We also assume that the encoder is a partial inverse mapping of the generator and the random variable $\mathbf{z}$ is sampled from a fixed distribution and is viewed as a random representation, while the inference network is designed to infer only meaningful latent representations. We can derive an invariant inference network to yield continuous and discrete latent variables which would define the properties of the images produced by the generator. For instance when generating images of human faces, the random noise can, after passing through the network, define human identities while the other latent variables can capture various meaningful variations such as the hair style, lighting, pose or human emotions among others. Such meaningful variations would be shown across all human identities in the generated images, through association.

We derive the following lower bound on the generator distribution:

$$\log q(\mathbf{x}) = \log \mathbb{E}_{p(\mathbf{z})p_\phi(\mathbf{c},\mathbf{s}|\mathbf{x})} \left[ \frac{q_\theta(\mathbf{x}, \mathbf{c}, \mathbf{s}, \mathbf{z})}{p(\mathbf{z}) p_\phi(\mathbf{c}, \mathbf{s}|\mathbf{x})} \right] \geq \mathbb{E}_{p(\mathbf{z})p_\phi(\mathbf{c},\mathbf{s}|\mathbf{x})} \left[ \log \left( \frac{q_\theta(\mathbf{x}, \mathbf{c}, \mathbf{s}, \mathbf{z})}{p(\mathbf{z}) p_\phi(\mathbf{c}, \mathbf{s}|\mathbf{x})} \right) \right], \tag{26}$$

where we used Jensen's inequality. We consider that the latent variables $\mathbf{c}$, $\mathbf{s}$, $\mathbf{z}$, are independent from each

other, and then we rewrite (26) as :

$$\begin{aligned}\log q(\mathbf{x}) &\geq \mathbb{E}_{p(\mathbf{z})p_\phi(\mathbf{c},\mathbf{s}|\mathbf{x})}\left[\log\left(\frac{q_\theta(\mathbf{x}|\mathbf{c},\mathbf{s},\mathbf{z})p(\mathbf{c})p(\mathbf{s})p(\mathbf{z})}{p(\mathbf{z})p_\phi(\mathbf{c}|\mathbf{x})p_\phi(\mathbf{s}|\mathbf{x})}\right)\right] \\ &\geq \mathbb{E}_{p(\mathbf{z})p_\phi(\mathbf{c},\mathbf{s}|\mathbf{x})}\left[\log q_\theta(\mathbf{x}|\mathbf{c},\mathbf{s},\mathbf{z})\right] + \mathbb{E}_{p(\mathbf{z})p_\phi(\mathbf{c},\mathbf{s}|\mathbf{x})}\left[\log\left(\frac{p(\mathbf{c})}{p_\phi(\mathbf{c}|\mathbf{x})}\right)\right] + \mathbb{E}_{p(\mathbf{z})p_\phi(\mathbf{c},\mathbf{s}|\mathbf{x})}\left[\log\left(\frac{p(\mathbf{s})}{p_\phi(\mathbf{s}|\mathbf{x})}\right)\right] \\ &\geq \mathbb{E}_{p(\mathbf{z})p_\phi(\mathbf{c},\mathbf{s}|\mathbf{x})}[\log q_\theta(\mathbf{x}|\mathbf{c},\mathbf{s},\mathbf{z})] - D_{KL}(p_\phi(\mathbf{c}|\mathbf{x})||p(\mathbf{c})) - D_{KL}(p_\phi(\mathbf{s}|\mathbf{x})||p(\mathbf{s})) = \mathrm{L_{VAE}}\end{aligned} \tag{27}$$

where $\mathrm{L_{VAE}}$ represents the VAE component of the loss function for the proposed model. The last two terms correspond to the KL divergence on the Gaussian and the Gumble-softmax (8) distributions, used for inferring continuous and categorical data, respectively. This VAE learning framework is different from most other VAE frameworks which would aim to model empirical data distributions. The proposed VAE learning is used to model the generator network distribution by generating images with known prior information, where the latent variables are sampled from the prior distributions. Learning to generate images in this case is easier than by using training data samples and can lead to an accurate inference model which typically works better on predicting discrete variables. The VAE loss is only used to optimize the parameters of the inference network. The model's structure and its training procedure are illustrated in the diagram from Figure 2(b).

For the final objective function, we include the adversarial loss $\mathrm{L_D}$, which is considered separately for the discriminator network, while we have the loss $\mathrm{L_{EG}}$ for the encoder and generator, VAE loss $\mathrm{L_{VAE}}$, and the Mutual Information (MI) maximization $\mathrm{L_{MI}}$. The gradients of the loss functions, used during the training, are calculated as :

$$\begin{aligned}\mathrm{L_D} = \bigtriangledown_\varphi \frac{1}{m}\sum_{i=1}^{m}[&\log D(\mathbf{x}_i, E(\mathbf{x}_i)) \\ &+ \log(1 - D(G(\mathbf{z}_i, \mathbf{c}_i, \mathbf{s}_i), \mathbf{c}_i, \mathbf{s}_i)],\end{aligned} \tag{28}$$

$$\mathrm{L_{EG}} = \bigtriangledown_{\theta,\phi} \frac{1}{\mathrm{m}}\sum_{\mathrm{i=1}}^{\mathrm{m}}[\log(1 - \mathrm{D}(\mathrm{G}(\mathbf{z}_\mathrm{i}, \mathbf{c}_\mathrm{i}, \mathbf{s}_\mathrm{i}), \mathbf{c}_\mathrm{i}, \mathbf{s}_\mathrm{i}))], \tag{29}$$

$$\mathrm{L_{MI}} = \bigtriangledown_{\varphi,\theta,\phi} \frac{1}{m}\sum_{i=1}^{m}\left[\mathbb{E}_{\mathbf{x}_i\sim G(\mathbf{z},\mathbf{u})}[\mathbb{E}_{\mathbf{s}_i\sim p(\mathbf{s},\mathbf{x})}[\log p_\phi(\mathbf{s}_i|\mathbf{x}_i)]] + \mathbb{E}_{\mathbf{x}_i\sim G(\mathbf{z},\mathbf{u})}[\mathbb{E}_{\mathbf{c}_i\sim p(\mathbf{c},\mathbf{x})}[\log p_\phi(\mathbf{c}_i|\mathbf{x}_i)]]\right], \tag{30}$$

$$\mathrm{L_{VAE}} = \bigtriangledown_{\phi} \frac{1}{m}\sum_{i=1}^{m}\left[\mathbb{E}_{p(\mathbf{z})p_\phi(\mathbf{c},\mathbf{s}|\mathbf{x}_i)}[\log q_\theta(\mathbf{x}_i|\mathbf{c}_i,\mathbf{s}_i,\mathbf{z}_i)] - D_{KL}(p_\phi(\mathbf{c}_i|\mathbf{x}_i)||p(\mathbf{c})) - D_{KL}(p_\phi(\mathbf{s}_i|\mathbf{x}_i)||p(\mathbf{s}))\right], \tag{31}$$

where we consider a batch size of $m$ data samples used during the training. We optimize the parameters of all components in our model by maximizing the mutual information $\mathrm{L_{MI}}$ and update the parameters of the inference model by means of the VAE loss, $\mathrm{L_{VAE}}$. The algorithm used for training the proposed model is named InfoVAEGAN, integrating jointly the inference capabilities, characteristic of VAEs with the generative abilities of GANs. The pseudocode of InfoVAEGAN is provided in Algorithm 1.

The schemes for the unsupervised learning for the GAN and VAE structures from InfoVAEGAN are shown in Figures 2(a) and 2(b), respectively. These processing structures are compared with other schemes used for deep unsupervised learning structures such as InfoGAN [7] and Adversarially Learned Inference (ALI) [26], shown in Figures 2(c) and 2(d), respectively. The proposed unsupervised approach uses discrete variables $\mathbf{s}$ aiming to capture meaningful variations that cannot be learnt by the continuous variables $\mathbf{c}$. The attributes learnt by the discrete variables $\mathbf{s}$ would benefit many down-stream applications such as the ability to easily manipulate image attributes. In the approach proposed in this paper, unlike in [25], the VAE learning procedure is used to model the generator distribution instead of empirical data distributions. The objective function used in the proposed model is also different from that used in InfoGAN [7], given that it is used for the inference of discrete and continuous variables.

**Algorithm 1:** The InfoVAEGAN training algorithm.

Algorithm : Training procedure
1:Sample $X^r = \{x^1, x^2, ...., x^N\}$ from training dataset
2:**While** $epoch < epoch^{\max}$ **do**
3: **While** $bath < bath^{\max}$ **do** minibatch procedure
4: $x_{batch} = Select(epoch, X^r)$ batch samples
5: $z \leftarrow p(z)$ Sample from Gaussian distribution
6: $c \leftarrow p(c)$ Sample from Gaussian distribution
7: $s \leftarrow p(s)$ Sample from Categorical prior
8: $x_g = G(z, c, s)$ Generate fack images
9: $\mu, \sigma^2, \overline{s} = E(x_g)$ Infer latent representations
10: $c' \leftarrow N(\mu, \sigma^2)$
11: $s' \leftarrow GumbleSoftmax(\overline{s})$
12: $x_r = G(z, c', s')$ Reconstruct image
13: Update discriminator network by $L_D$
14: Update encoder and decoder by $L_{EG}$
15: Update all components by $L_{mul}$
16: Update encoder by $L_{VAE}$
17: **End**
17:**End**

### *4.4. Training the Generator using real images*

Similarly to InfoGAN [7], the inference network in the proposed InfoVAEGAN model never sees real images, and consequently this defines the limitations of the proposed approach. In the following we consider training the generator network using real images, aiming to provide more accurate data reconstructions which can be seen as an inverse mapping of the inference network. We consider a similar generator to the one used so far in the proposed InfoVAEGAN with the difference that the new generator receives as inputs only two latent vectors inferred by the inference network, while attempting to yield a reconstruction that approximates the input of the inference network as much as possible. In this case we use the following loss function $\mathrm{L_{VAE2}}$ in order to train the generator :

$$\log p(\mathbf{x}) \geq \mathbb{E}_{p_\phi(\mathbf{c},\mathbf{s}|\mathbf{x})}[\log q_\gamma(\mathbf{x}|\mathbf{c},\mathbf{s})] - D_{KL}(p_\phi(\mathbf{c}|\mathbf{x})||p(\mathbf{c})) - D_{KL}(p_\phi(\mathbf{s}|\mathbf{x})||p(\mathbf{s})) = \mathrm{L_{VAE2}} \tag{32}$$

where $q_\gamma(\mathbf{x}|\mathbf{c},\mathbf{s})$ is parameterized by a deep neural network, defined by parameters $\gamma$. Then, we produce a new model, named InfoVAEGAN2, by replacing $\mathrm{I_{VAE}}$ from (31) with $\mathrm{I_{VAE2}}$, considering the generator training derived from (32). Under these conditions, during the training, the inference network receives a mini-batch of real images and outputs the parameters for the Gumble-softmax and Gaussian distributions. The discrete and continuous latent variables, sampled from the Gumble-softmax and Gaussian distributions, are concatenated into a vector, which is then fed into the generator aiming to recover the images from the training set.

## 5. Experimental results

In this section, we evaluate the ability of the proposed joint latent representation model, InfoVAEGAN, to learn disentangled representations, under both supervised and unsupervised learning frameworks.

*5.1. Supervised learning*

The encoder and decoder are implemented by fully connected networks with two layers of 500 hidden nodes each. When using the first supervised learning structure, whose diagram is shown in Figure 1(a), the encoder typically outputs 10-dimensional latent space vectors, representing the mean and variance vectors, for modelling the underlying Gaussian distributions. Then the probabilistic vector is sampled from this distribution and transformed into a discrete latent vector by using the Gumble-softmax distribution, (8). For the second supervised learning structure, implemented using the diagram shown in Figure 1(b), we use separate encoders in order to model the continuous and discrete latent variables.

In the following, we evaluate the performance of the proposed supervised joint latent models on two datasets, MNIST [36] and Fashion-MNIST [37]. Specifically, we consider a variation of the datasets where the digital images are rotated. In order to construct such datasets, we select 5,000 handwritten versions of a specific digit from MNIST, or of a clothing item from Fashion-MNIST, and then rotate these images nine times, with angle values at equal intervals between 0 and 80 degrees, resulting in a total of 45,000 training and testing samples. Then we select randomly 90 image samples from among both original and rotated images for training, while the rest is used for testing. The information about the angle of the rotation is treated as a one-hot vector during the training, resulting in 10 classes, each corresponding to the rotated images with a particular angle.

**MNIST dataset.** MNIST dataset contains images of handwritten digits covering many variations in their appearance. We train the supervised model with the two joint discrete and continuous latent models on MNIST dataset for 100 epochs. The results are shown in Figures 3(a) and 3(b), for using the learning structures with one encoder and two encoders, from Figures 1(a) and 1(b), respectively. In the first two rows from Figure 3, we show the real images for the handwritten digit '3', displaying the characteristics of variation for discrete (defined by the rotation angle) and continuous latent variables, respectively. After the training, we randomly select two sets of images from the testing data, as shown in the first two rows from Figure 3. We want to generate new images that would preserve the handwritten style of the digit from one image from the first row and adopt the angle information of the digit from the second image, shown on the second row of images from Figure 3. The information corresponding to the continuous and discrete latent variables is inferred by the encoder and is used as the input for the decoder. In this example, the discrete latent variables would capture the angle information, while the other underlying factors are embedded into the continuous latent spaces. It can be observed that our model can accurately infer discrete latent variables, as shown in the results from the last two rows from Figures 3(a) and 3(b), where the generated images capture well the orientation of the images of the digit '3' from the second row and the continuous characteristics of the images from the first row. We also generate images characterized by different angle orientations while displaying a certain discrete latent variable, and the examples are shown in Figure 4. Meanwhile, embedding variations in the continuous variables is illustrated in the generated images shown in Figure 5.

**Fashion-MNIST dataset.** Fashion-MNIST consists of greylevel images showing clothing items. In the following, we repeat the experiments described above with the MNIST database, by considering rotation variations as discrete variables, as well as modelling the continuous variables, for the Fashion-MNIST dataset. Selected original images are shown in the first two rows from Figure 6, while the generated results are shown in the last two rows. We observe that the generated images preserve specific characteristics such as the greylevel variation and the shape of the clothing items from the first row of images while they are rotated according to the angle information inferred from the second row of images. In Figure 7 we show the results on an image when varying the orientation angle while keeping the other variables fixed. In Figure 8 we show the results when fixing a discrete variable, represented by the rotation angle, while transferring the continuous variable information from the images shown in Figure 8(a).

In the following, we want to see whether the continuous latent variables can learn disentangled representations. We vary a single variable in the continuous latent space while fixing the others. We show the results in Figures 9 and 10 for MNIST and Fashion-MNIST database, respectively. We can observe that the continuous latent variables capture various attributes for the given data, such as the handwritten style, lighting variation and shape variation without changing the angle information. These results demonstrate that

(a) Results for the structure from Figure 1(a).

(b) Results for the structure from Figure 1(b).

Figure 3: The first two rows are showing testing images from the MNIST database, while the other two rows display the images generated by transferring the writing styles of the images of handwritten digits from the first row while using the angle information of the images from the second row.

(a) Results for the structure from Figure 1(a).

(b) Results for the structure from Figure 1(b).

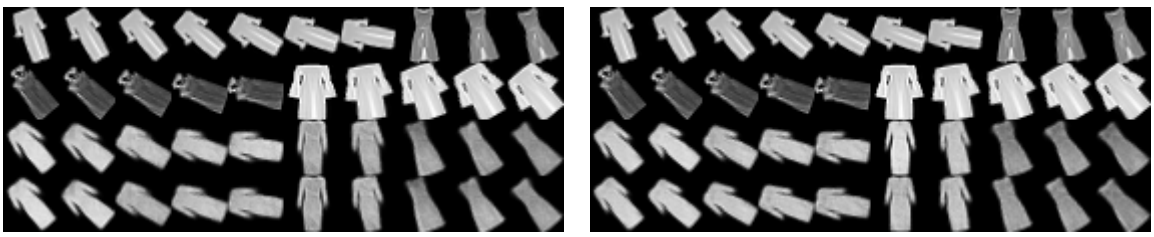
Figure 4: Generated images of the digit '3' when changing the angle from 0 to 80 degrees, while fixing other latent variables.

(a) Random images from MNIST dataset.

(b) Generated images by the structure from Figure 1(a).

(c) Generated images by the structure from Figure 1(b).

Figure 5: Generated images of the digit '3' when fixing the angle information as 0 and transferring the continuous characteristics of the original images from (a).



(a) Results for the structure from Figure 1(a).

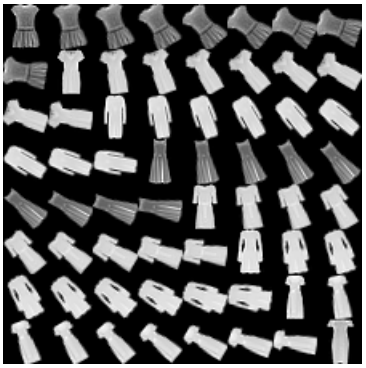
(b) Results for the structure from Figure 1(b).

Figure 6: The first and second rows represent images from Fashion-MNIST dataset, while the images from the other two rows contain generated images, defined by the style transferred from the images on the first row and using the angle information transferred from the images from the second row.

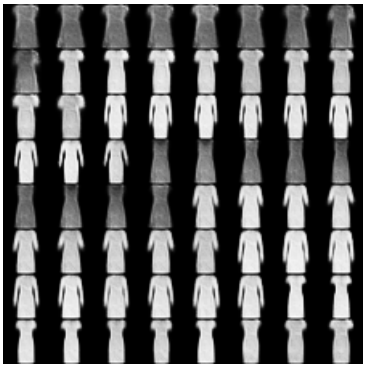
(a) Results for the structure from Figure 1(a).

(b) Results for the structure from Figure 1(b).

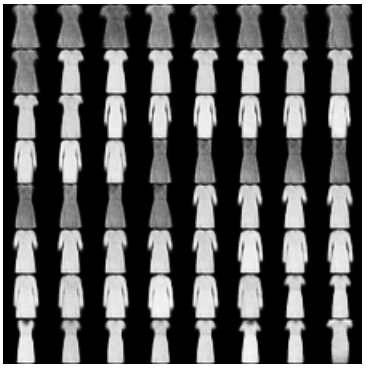
Figure 7: Results when changing the angle information from 0 to 80 degrees while fixing the other variables.



(a) Random images from Fashion-MNIST dataset.



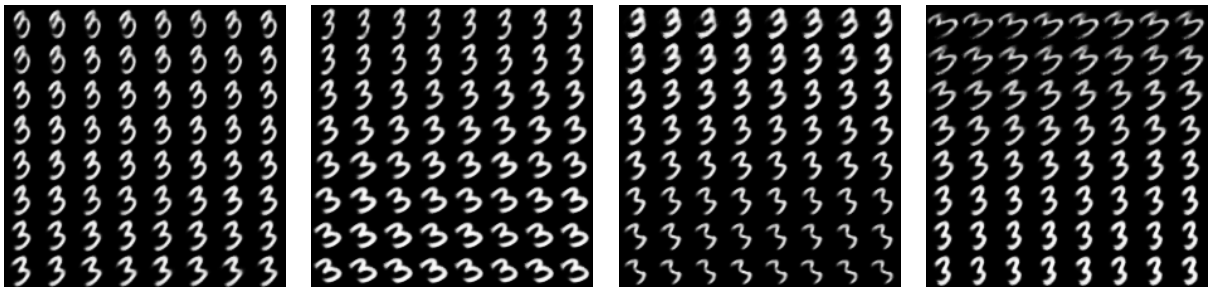
(b) Results for the structure from Figure 1(a).



(c) Results for the structure from Figure 1(b).

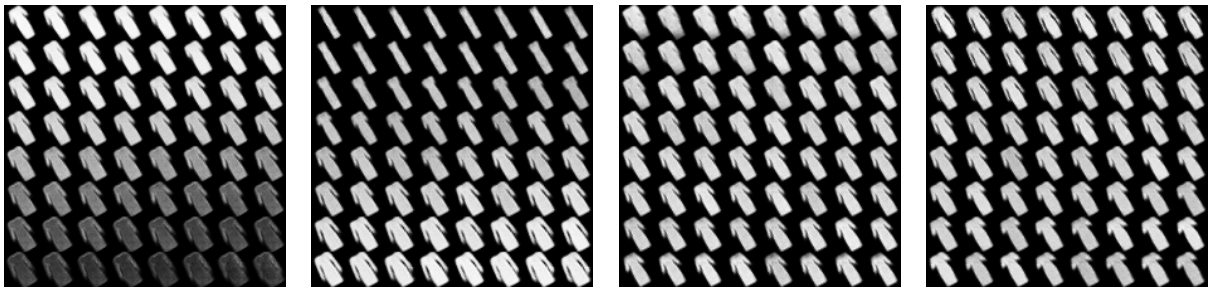
Figure 8: Reconstructed results on Fashion-MNIST, where the discrete variable, represented by the vertical orientation is fixed, while transferring the continuous variable properties from the images shown in (a).

InfoVAEGAN can model meaningful data representations after learning separate discrete and continuous latent variables.



(a) Variable 1. (b) Variable 7. (c) Variable 9. (d) Variable 6.

Figure 9: Image generation results on the MNIST database when fixing the angle information and changing a single variable from -3 to 3.



(a) Variable 2. (b) Variable 4. (c) Variable 1. (d) Variable 3.

Figure 10: Image generation results on Fashion-MNIST database when fixing the angle information and changing a single variable from -3 to 3.

### *5.2. Unsupervised learning*

In the general case we do not have class labels for the given images and we would have to adopt unsupervised learning. In the following experiments we consider for comparison InfoGAN [7] and the Adversarially Learned Inference (ALI) [26], which are deep learning models related to those proposed in this study. We use the same network architecture and hyperparameter configurations for all methods for a fair comparison.

**MNIST dataset.** In order to allow the discrete latent variable to capture the characteristics of the handwritten digit shape, we choose one categorical vector sampled from the categorical distribution $\mathbf{s} \sim Cat(K = 10, p = 0.1)$ and two continuous codes sampled from the uniform distribution $U(-1, 1)$. For the proposed InfoVAEGAN model we first infer the latent variables from the testing data samples and then use them as inputs for the generator. The discrete latent variables are sampled from the Gumble-softmax distribution (8), while the continuous latent variables are sampled from the Gaussian distribution whose mean and diagonal covariance are parameterized by the encoder. Selected images of handwritten digits from MNIST database are shown in Figure 11(a). The results for the proposed InfoVAEGAN and InfoVAEGAN2 models, whose training is described in Sections 4.3 and 4.4, are shown in Figures 11(d) and 11(e), respectively, while the reconstruction results for ALI [26] and InfoGAN [7] are shown in Figures 11(b) Figures 11(c). The results indicate that InfoVAEGAN and InfoGAN can generate accurate images of digits, while ALI cannot do that. Meanwhile, InfoVAEGAN2 can provide more accurate reconstructions which demonstrates that the additional generator is an exact inverse mapping of the inference network.

In the following experiments, we explore the latent space of InfoVAEGAN by varying the discrete latent variable $\mathbf{s}$ from 0 to 9 and sample the continuous latent variables from the uniform distribution $U(-1, 1)$.

(a) Real testing images.

(b) ALI [26] reconstructions.

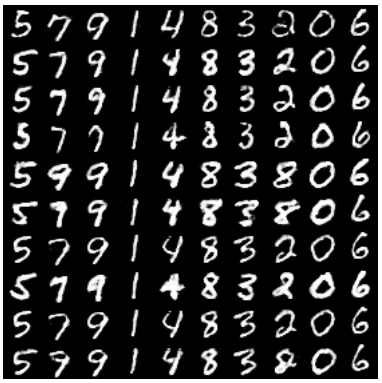
(c) InfoGAN [7] reconstructions.

(d) InfoVAEGAN reconstructions.

(e) InfoVAEGAN2 reconstructions.

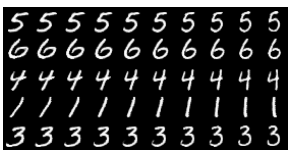
Figure 11: The reconstruction results on MNIST images.

(a) InfoVAEGAN.

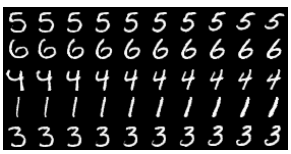


(b) InfoGAN.

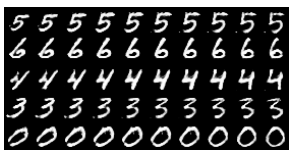
Figure 12: Generation results across all classes.



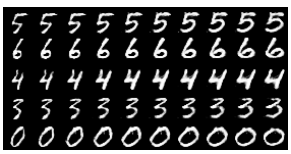
(a) InfoVAEGAN, when changing $\mathbf{c}_1$.



(b) InfoVAEGAN, when changing $\mathbf{c}_2$.



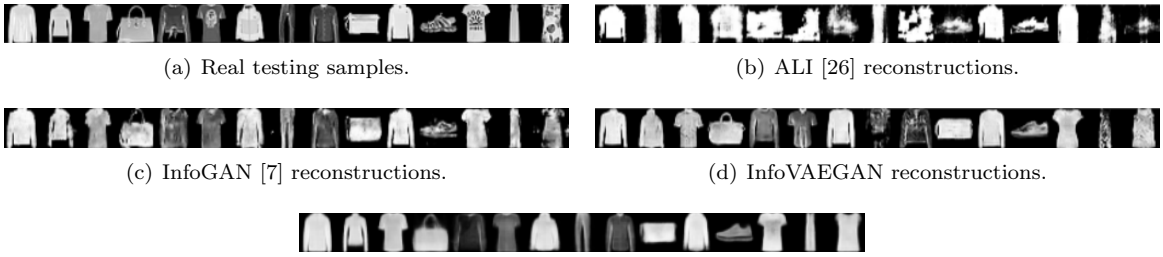
(c) InfoGAN, when changing $\mathbf{c}_1$.



(d) InfoGAN, when changing $\mathbf{c}_2$.

Figure 13: Generation results when changing the continuous variables 1 and 2 from -1 to 1, respectively.

The images of handwritten digits generated by InfoVAEGAN in the style of those from the MNIST database are shown in Figure 12(a), where each column represents the generated images corresponding to a specific discrete variable and each row is characterized by a certain variation in the continuous latent variables. For comparison the images generated by InfoGAN [7] are shown in Figure 12(b). It can be observed that some of the results produced by InfoGAN, such as the images for the digit '2' from the 8th column are rather corresponding to images of the digit '8' in 3 cases. Also many of the images generated by InfoGAN are not that sharp and clear as those generated by the proposed InfoVAEGAN, as it can be observed in the results from Figure 12. Next, we change the continuous codes $c_1$, $c_2$ from -1 to 1, and keep fixed the other latent variables. The results when varying $c_1$ are shown in Figures 13(a) and 13(c), for InfoVAEGAN and InfoGAN, respectively, while those when varying $c_2$ are shown in Figures 13(b) and 13(d), for InfoVAEGAN and InfoGAN, respectively. From these results it can be observed that InfoVAEGAN captures well specific rotation angles as well as the characteristics of the handwriting styles of the digits.



(a) Real testing samples.

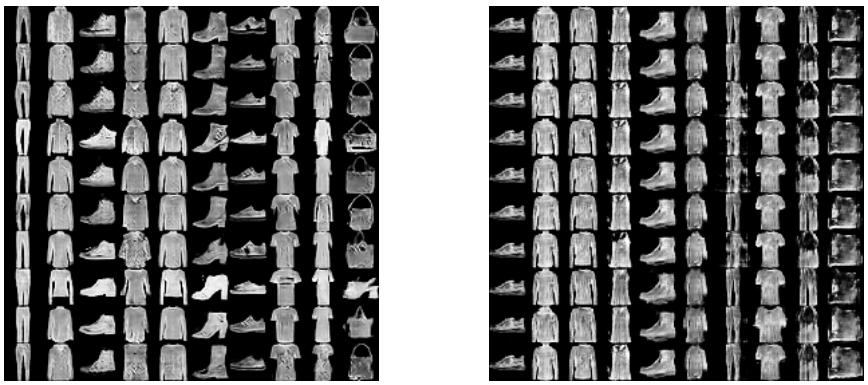
(b) ALI [26] reconstructions.

(c) InfoGAN [7] reconstructions.

(d) InfoVAEGAN reconstructions.

(e) InfoVAEGAN2 reconstructions.

Figure 14: The reconstruction results on Fashion-MNIST.

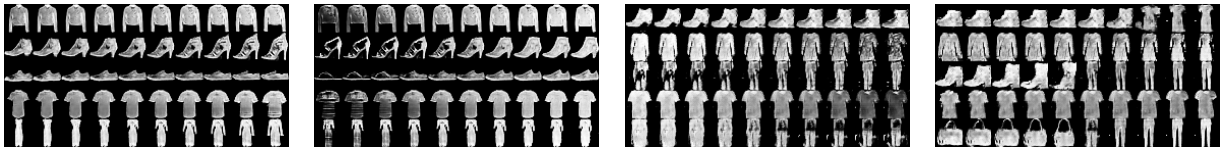


(a) InfoVAEGAN.

(b) InfoGAN.

Figure 15: Generated image results across all MNIST classes.

**Fashion-MNIST dataset.** The Fashion-MNIST dataset contains grey-level images of clothing items which display more complex information than the images from the MNIST dataset. The same hyperparameter configuration is used for Fashion-MNIST as for the MNIST dataset when testing InfoVAEGAN and InfoGAN. Selected images from Fashion-MNIST database are shown in Figure 14(a). The results for the proposed models InfoVAEGAN and InfoVAEGAN2 are provided in Figures 14(d) and 14(e), while the results for ALI [26] and InfoGAN [7] are shown in Figures 14(b) Figures 14(c). From these results it can be observed that InfoVAEGAN and InfoVAEGAN2 can reconstruct the images from this database better than ALI or InfoGAN. Generated images across all Fashion-MNIST classes are shown in Figure 15(a) for InfoVAEGAN and in Figure 15(b) for InfoGAN, when changing the discrete code from 0 to 9. From these images we observe that InfoGAN tends to produce images of a similar style, while InfoVAEGAN is able to yield a variety of



(a) InfoVAEGAN, when changing $\mathbf{c}_1$.

(b) InfoVAEGAN, when changing $\mathbf{c}_2$.

(c) InfoGAN, when changing $\mathbf{c}_1$.

(d) InfoGAN, when changing $\mathbf{c}_2$.

Figure 16: Generation results when changing the continuous variables 1 and 2 from -1 to 1, respectively.

image styles, defined by changes in shape and texture. Then, we change individually two continuous latent variables $\mathbf{c}_1$, $\mathbf{c}_2$ from -1 to 1 and the results are presented in Figures 16(a) and 16(b) for InfoVAEGAN and for comparison in Figures 16(c) and 16(d) for InfoGAN. We observe that the continuous latent variables in the proposed approach can capture well the variations in dimension and lighting for all items, according to the results from Figures 16(a) and 16(b), respectively. Meanwhile, InfoGAN may generate unexpected categories of images when the latent variable is other than 0, as it can be observed in Figure 16(d).

In the following, we consider testing the unsupervised classification ability of the proposed InfoVAEGAN on both MNIST and Fashion-MNIST datasets. The results on the testing set are provided in Table 1. Most existing unsupervised learning methods consider mixture models, so we refer to $K$ as the number of components in the mixture, in the third column of Table 1. In addition, we can extend the proposed approach to infer the training set in order to improve the classification performance in a CNN network. We only consider data samples generated by the inference network with the confidence larger than 0.9 and do not consider other data. Then we train a simple CNN network which considers the inferred training samples as inputs. The loss is calculated by using the class label inferred by the inference network. From Table 1 we observe that the proposed InfoVAEGAN achieves higher accuracy than InfoGAN and than most other models considered for comparison and listed in the table. The CNN trained on the selected training set obtained by the inference network achieves the best accuracy. This demonstrates that the proposed approach can be used as an inference network generating features which can be used as inputs for other deep networks.

Table 1: Unsupervised classification accuracy on MNIST dataset.

| Dataset | Method | K | Classification |
|---|---|---|---|
| MNIST | InfoVAEGAN | 1 | 95.65 |
| | Proposed CNN | 1 | 96.45 |
| | InfoGAN [7] | 1 | 93.35 |
| | GMVAE[38] | 30 | 89.27 |
| | GMVAE[38] | 16 | 87.82 |
| | AAE[28] | 16 | 90.45 |
| | CatGAN[39] | 30 | 95.73 |
| | DEC[40] | 10 | 84.30 |
| | PixelGAN[19] | 30 | 94.73 |
| Fashion-MNIST | InfoVAEGAN | 1 | 51.24 |
| | InfoGAN | 1 | 39.80 |
| | Proposed CNN | 1 | 51.46 |

**3D-chairs dataset.** We consider a 10-dimensional vector for modelling the discrete $\mathbf{s}$ and continuous $\mathbf{c}$ latent variables in order to represent the information from the 3D-chairs dataset [41]. We set the dimension of the random vector $\mathbf{z}$ as 100. After the training, we change a single latent variable from the latent space, while fixing the others. The results produced by InfoVAEGAN are shown in Figures 17(f)-(j), while those produced by $\beta$-TCVAE [24], which is known for producing disentangled representations, are shown in

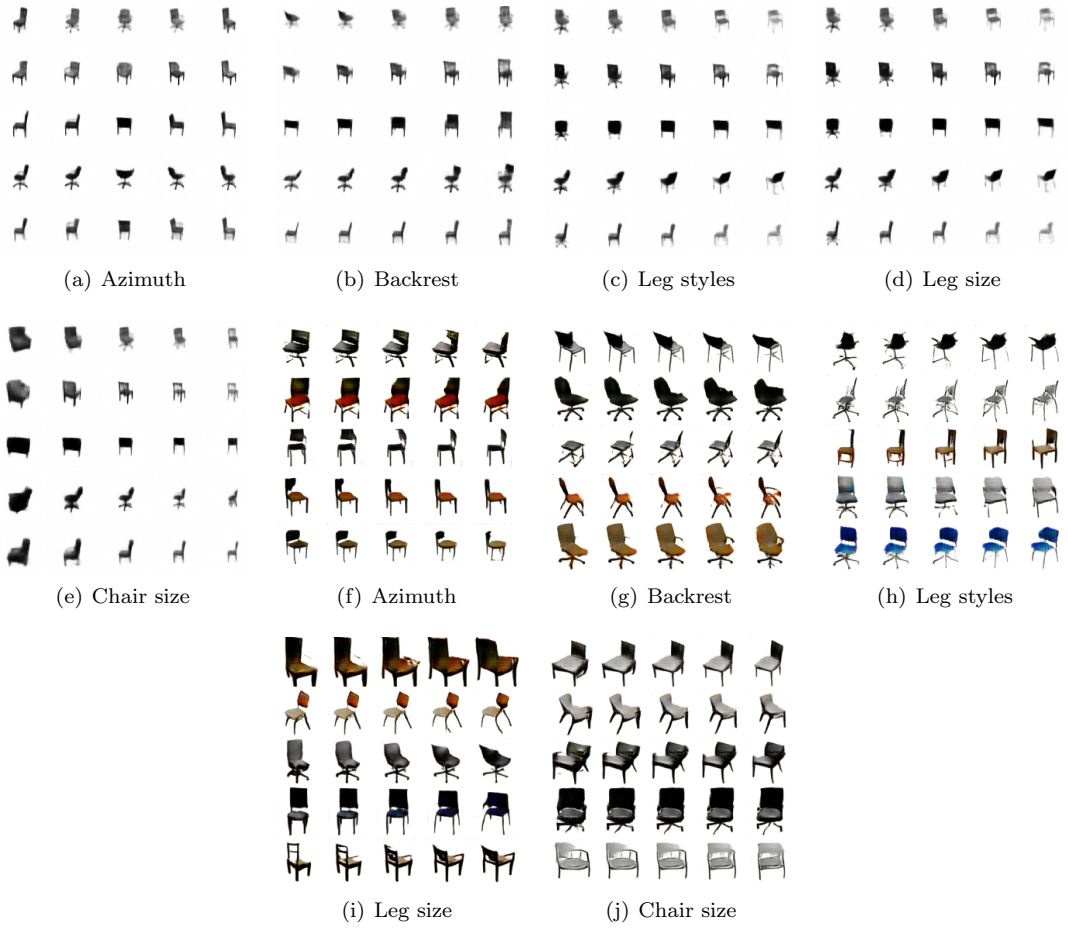



Figure 17: Results when manipulating latent codes on the 3D chairs dataset. Each time we change a single latent variable in the latent space from -1 to 1 while fixing the others . The results in (a)-(e) are produced by $\beta$-TCVAE [24], while those from (f)-(j) are produced by InfoVAEGAN.

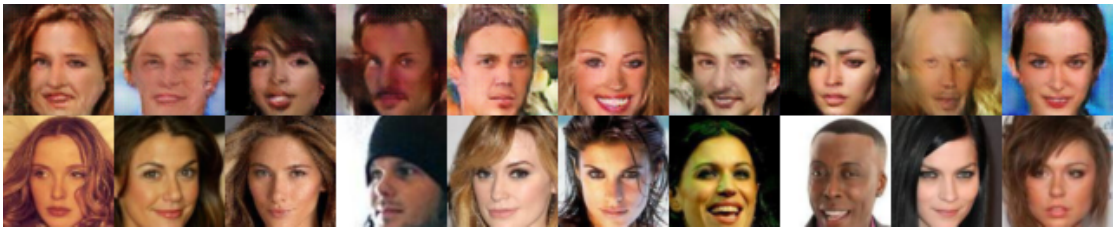
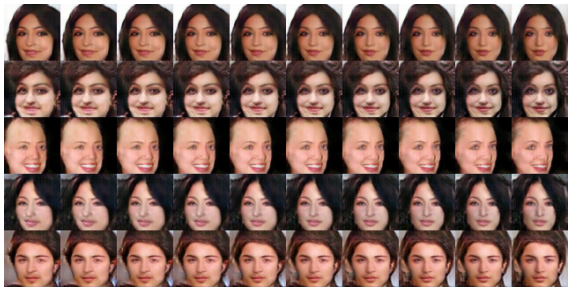


Figure 18: Face images generated by InfoVAEGAN when training on CelebA dataset. The first row of images represents generated face images, while the second row represent real images used for testing. The discrete and continuous latent variables are inferred by the encoder which are then used as inputs for the generator network.

(a) Changing the gender

(b) Skin color

(c) Varying azimuth (pose)



(d) Changing emotion

(e) Changing hair style

(f) Applying moustache

(g) Changing face width

(h) Applying bangs on face appearance

Figure 19: Manipulating latent codes on CelebA dataset. We change a single latent variable in the latent space between -1 to 1, while fixing the others.

Figures 17(a)-(e). From these results we can observe that the proposed approach can discover five distinct types of variations in the data, which is better than what InfoGAN can do, [7]. The proposed approach can generate results which are better while showing more diverse variations in the data, invariant to the discrete variable, when compared to $\beta$-TCVAE. For instance, for the results when the chair size is increasing in Figure 17(j), other features such as the backrest of the chair are changing proportionally. Proportional changes can also be observed in Figure 17(i) when changing the chairs' leg size in the 3D Chairs images.

**CelebA dataset.** In the following we evaluate the proposed InfoVAEGAN approach on CelebA dataset [42]. This dataset contains almost 200,000 images of human faces with 40 different attributes such as pose change, gender, lighting change and so on. All face images are cropped and resized to $64 \times 64$ pixels. We model the latent space of these face images and then we explore their latent space using interpolations and by manipulating their underlying characteristics. We randomly choose 90% of data as the training set while the remaining data are used for testing. We consider a 10-dimensional vector for modelling the discrete $\mathbf{s}$ and continuous $\mathbf{c}$ latent variables. The generated images are shown in Figure 18, where we infer the continuous and discrete latent variables from the testing image samples and use the resulting variables as inputs for the generator. Although the proposed approach may not always accurately reconstruct the test images, we can see that the generated human faces preserve important data characteristics such as the azimuth indicating face orientation. We also change a single latent variable while fixing the others. The ability of the proposed methodology to manipulate specific latent variables in face images is shown in the examples from Figures 19(a)-(h), when changing the gender in the face image, skin color, varying face azimuth, the appearance of emotion, hair style, applying/removing moustache to the face, changing the ratio between the face's width and height, and changing the style of bangs in face images, respectively. We observe that the proposed InfoVAEGAN approach is able to discover at least eight different disengaged representations, and model continuous variations within each of these, which represent better results than those achieved by InfoGAN, [7]. The manipulated generative results of InfoVAEGAN are also better than those of $\beta$-VAE [5], which generates rather blurred images.

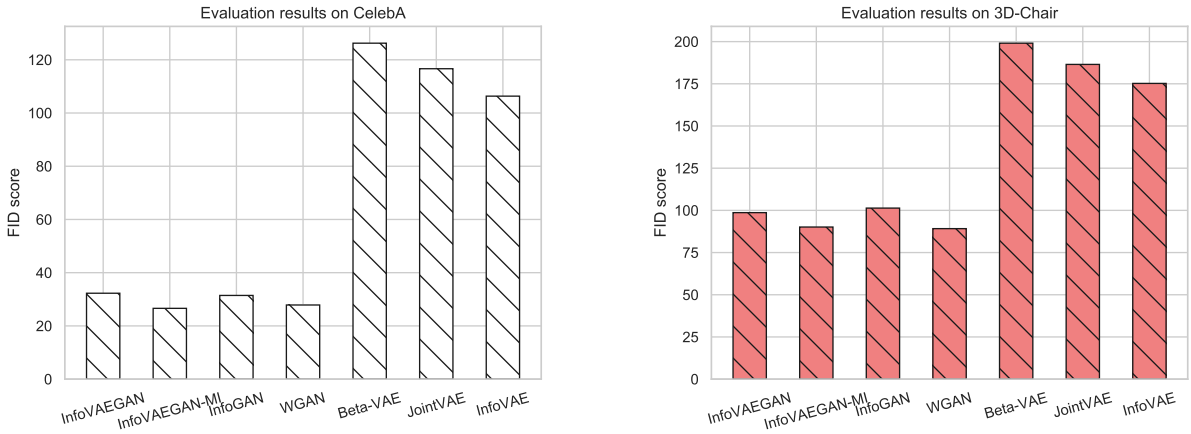
### *5.3. Assessing numerically the quality of the generated images*

In this section, we assess numerically the quality of the generated images and compare the results with those obtained by other methods. Firstly, we investigate the disentanglement ability of the proposed approach by using the metric from [18] and the dataset dSprites [43]. The results are reported in Table 2, where all other results are cited from [25], except for InforVAE [44]. It can be observed that the proposed InfoVAEGAN approach achieves a competitive disentanglement score when compared to those of the other methods.

Table 2: Disentanglement scores on the dSprites dataset [43].

| Method | Variable | Score |
|---|---|---|
| InfoVAEGAN | 10 | 0.81 |
| $\beta$-VAE [5] | 10 | 0.73 |
| FactorVAE [18] | 10 | 0.82 |
| JointVAE [25] | 10 | 0.69 |
| InfoVAE [44] | 10 | 0.72 |
| GUIDED-VAE [45] | 10 | 0.66 |
| GUIDED--TCVAE [45] | 10 | 0.72 |

In the following, we use the Frechet Inception Distance (FID) [46], to evaluate the quality of the generated images in CelebA and 3D-Chairs databases. The FID results calculated on CelebA and 3D-chairs are provided in the bar-plots from Figures 20(a) and 20(b), respectively, for the generated images, where InfoVAEGAN-MI denotes that the proposed approach does not use the MI loss, $\mathrm{L_{MI}}$ from (30). The results show that InfoVAEGAN-MI, unlike InfoVAEGAN, has a better effect on the performance of the generator network. It can be observed that all GAN based methods outperform the VAE based methods, including




(a) CelebA database (b) 3D-Chairs database

Figure 20: Assessment of the image quality using Frechet Inception Distance (FID).

$\beta$-VAE [5] and JointVAE [25]. These results indicate that the proposed InfoVAEGAN approach can mitigate well both the disentanglement performance as well as the quality of the generated images. The proposed InfoVAEGAN model generates images of higher quality than InfoVAE.

## 6. Conclusions

In this paper, we introduce two new deep learning generative frameworks, combining the latent space inference abilities of VAEs with the generative capabilities of GANs for supervised and unsupervised disentangled representation learning. The results provided show that the proposed frameworks can learn meaningful representations. Under the supervised setting, we show how mutual information regularization can be used to induce disentangled representations in continuous and discrete latent variable spaces. Under the unsupervised setting we propose to maximize the mutual information between the joint latent variables and the generated data in order to encourage the learning of interpretable and disentangled representations in datasets containing complex images. We propose a novel algorithm, namely InfoVAEGAN, to train the joint latent variable model while the adversarial loss is used to encourage matching the two joint distributions. InfoVAEGAN can be seen as a novel hybrid method based on GANs and VAEs, which is able not only to produce high quality images but can also discover interpretable representations in the data. In addition, the learned representations can be applied in many down-stream tasks such as image editing through attribute manipulation as well as for exploring the latent space through interpolations between data characterized by different attributes. The experimental results also show that the proposed approach outperforms the state of the art methods when assessing either the disentanglement results or the unsupervised classification capabilities.

These research results can be expanded by adapting other disentangled methods, such as by employing the total correlation [23] in the framework described in this paper in order to induce better-disentangled representations. We are currently working on developing novel algorithms able to learn meaningful and informative latent representation across multiple tasks under the lifelong learning setting.

## References

[1] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Trans. on Pattern Analysis and Machine Intelligence 35 (8) (2013) 1798–1828.

[2] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, S. J. Gershman, Building machines that learn and think like people, Behavioral and brain sciences 40, E253 (2017).

[3] I. Higgins, A. Pal, A. Rusu, L. Matthey, C. Burgess, A. Pritzel, M. Botvinick, C. Blundell, A. Lerchner, Darla: Improving zero-shot transfer in reinforcement learning, in: Proc. of Int. Conf. on Machine Learning (ICML), vol. PMLR 70, 2017, pp. 1480–1490.
[4] D. P. Kingma, M. Welling, Auto-encoding variational Bayes (2013).
URL https://arxiv.org/abs/1312.6114
[5] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, $\beta$-VAE: Learning basic visual concepts with a constrained variational framework, in: Proc. Int. Conf. on Learning Representations (ICLR), 2017.
[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Inf. Proc. Systems (NIPS), 2014, pp. 2672–2680.
[7] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, P. Abbeel, InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets, in: Advances in Neural Inf. Proc. Systems (NIPS), 2016, pp. 2172–2180.
[8] T. D. Kulkarni, W. F. Whitney, P. Kohli, J. Tenenbaum, Deep convolutional inverse graphics network, in: Advances in Neural Inf. Proc. Systems (NIPS), 2015, pp. 2539–2547.
[9] S. Reed, K. Sohn, Y. Zhang, H. Lee, Learning to disentangle factors of variation with manifold interaction, in: Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 32(2), 2014, pp. 1431–1439.
[10] W. F. Whitney, M. Chang, T. Kulkarni, J. B. Tenenbaum, Understanding visual concepts with continuation learning, in: Proc. ICLR Workshop, 2016.
URL https://arxiv.org/abs/1602.06822
[11] J. Yang, S. E. Reed, M.-H. Yang, H. Lee, Weakly-supervised disentangling with recurrent transformations for 3D view synthesis, in: Advances in Neural Inf. Proc. Systems (NIPS), 2015, pp. 1099–1107.
[12] F. Locatello, M. Tschannen, S. Bauer, G. Rätsch, B. Schölkopf, O. Bachem, Disentangling factors of variation using few labels, in: Proc. Int. Conf. on Learning Representations (ICLR), 2020.
URL https://arxiv.org/abs/1905.01258
[13] K. Sohn, H. Lee, X. Yan, Learning structured output representation using deep conditional generative models, in: Advances in Neural Inf. Proc. Systems (NIPS), 2015, pp. 3483–3491.
[14] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, Jour. of Machine Learning Research 17 (1) (2016) 2096–2030.
[15] J. Klys, J. Snell, R. Zemel, Learning latent subspaces in variational autoencoders, in: Advances in Neural Inf. Proc. Systems (NIPS), 2018, pp. 6445–6455.
[16] J. Schmidhuber, Learning factorial codes by predictability minimization, Neural Computation 4 (6) (1992) 863–879.
[17] G. Desjardins, A. Courville, Y. Bengio, Disentangling factors of variation via generative entangling (2012).
URL https://arxiv.org/abs/1210.5474
[18] H. Kim, A. Mnih, Disentangling by factorising, in: Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 80, 2018, pp. 2649–2658.
[19] A. Makhzani, B. J. Frey, Pixelgan autoencoders, in: Advances in Neural Inf. Proc. Systems (NIPS), 2017, pp. 1975–1985.
[20] Y. Jeong, H. O. Song, Learning discrete and continuous factors of data via alternating disentanglement, in: Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 97, 2019, pp. 3091–3099.
[21] E. Mathieu, T. Rainforth, N. Siddharth, Y. W. Teh, Disentangling disentanglement in variational autoencoders, in: Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 97, 2019, pp. 4402–4412.
[22] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, A. Lerchner, Understanding disentangling in $\beta$-VAE, in: Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 97, 2019, pp. 4402–4412.
[23] S. Gao, R. Brekelmans, G. V. Steeg, A. Galstyan, Auto-encoding total correlation explanation, in: Proc. Int. Conf. on Artificial Intelligence and Statistics (AISTATS), vol. PMLR 89, 2019, pp. 1157–1166.
[24] T. Q. Chen, X. Li, R. B. Grosse, D. K. Duvenaud, Isolating sources of disentanglement in variational autoencoders, in: Advances in Neural Inf. Proc. Systems (NIPS), 2018, pp. 2615–2625.
[25] E. Dupont, Learning disentangled joint continuous and discrete representations, in: Advances in Neural Inf. Proc. Systems (NIPS), 2018, pp. 708–718.
[26] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, A. Courville, Adversarially learned inference, in: Proc. Int. Conf. on Learning Representations (ICLR), 2017.
URL https://arxiv.org/abs/1606.00704
[27] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, O. Winther, Autoencoding beyond pixels using a learned similarity metric, in: Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 48, 2016, pp. 1558–1566.
[28] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey, Adversarial autoencoders, in: Proc. Int. Conf. on Learning Representations (ICLR), 2016.
URL https://arxiv.org/abs/1511.05644
[29] C. Li, H. Liu, C. Chen, Y. Pu, L. Chen, R. Henao, L. Carin, Alice: Towards understanding adversarial learning for joint distribution matching, in: Advances in Neural Inf. Proc. Systems (NIPS), 2017, pp. 5495–5503.
[30] Y. Fei, A. G. Bors, Learning latent representations across multiple data domains using Lifelong VAEGAN, in: Proc. of European Conf. on Computer Vision (ECCV), vol. LNCS 12365, 2020, pp. 777–795.
[31] B. E. J. Gumbel, Statistical theory of extreme values and some practical applications:a series of lectures, 1954.
[32] C. Maddison, D. Tarlow, T. Minka, A* sampling, in: Advances in Neural Inf. Processing Systems (NIPS), 2014, pp. 3086–3094.
[33] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, in: Proc. Int. Conf. on Learning Representations (ICLR), 2018.

URL https://arxiv.org/abs/1611.01144
[34] J. Donahue, P. Krähenbühl, T. Darrell, Adversarial feature learning, in: Proc. Int. Conf. on Learning Representations (ICLR), 2017.
URL https://arxiv.org/abs/1605.09782
[35] D. Barber, F. Agakov, The IM algorithm: a variational approach to information maximization, in: Advances in Neural Inf. Proc. Systems (NIPS), 2003, pp. 201–208.
[36] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. of the IEEE 86 (11) (1998) 2278–2324.
[37] H. Xiao, K. Rasul, R. Vollgraf, Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms (2017).
URL https://arxiv.org/abs/1708.07747
[38] N. Dilokthanakul, P. A. Mediano, M. Garnelo, M. C. Lee, H. Salimbeni, K. Arulkumaran, M. Shanahan, Deep unsupervised clustering with gaussian mixture variational autoencoders, in: Proc. Int. Conf. on Learning Representations (ICLR), 2017.
URL https://arxiv.org/abs/1611.02648
[39] J. T. Springenberg, Unsupervised and semi-supervised learning with categorical generative adversarial networks, in: Proc. Int. Conf. on Learning Representations (ICLR), 2015.
URL https://arxiv.org/abs/1511.06390
[40] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: Proc. Int. Conf. on Machine Learning (ICML), vol. PMLR 48, 2016, pp. 478–487.
[41] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, J. Sivic, Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models, in: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 3762–3769.
[42] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proc. of IEEE Int. Conf. on Computer Vision (ICCV), 2015, pp. 3730–3738.
[43] dSprites - Disentanglement testing Sprites dataset, 2017.
URL https://deepmind.com/research/open-source/dsprites-disentanglement-testing-sprites-dataset
[44] S. Zhao, J. Song, S. Ermon, InfoVAE: Balancing learning and inference in variational autoencoders, in: Proc. of the AAAI Conf. on Artificial Intelligence, Vol. 33, 2019, pp. 5885–5892.
[45] Z. Ding, Y. Xu, W. Xu, G. Parmar, Y. Yang, M. Welling, Z. Tu, Guided variational autoencoder for disentanglement learning, in: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, 2020, pp. 7920–7929.
[46] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, GANs trained by a two time-scale update rule converge to a local Nash equilibrium, in: Advances in Neural Inf. Proc. Systems, 2017, pp. 6626–6637.