Disentangling high-level factors and their features with Conditional Vector Quantized VAEs

Kaifeng Zou, Sylvain Faisan, Fabrice Heitz, Sébastien Valette

Disentangling high-level factors and their features with Conditional Vector Quantized VAEs

Kaifeng Zou[a], Sylvain Faisan[a], Fabrice Heitz[a], Sébastien Valette[b]

[a]*ICube Laboratory, University of Strasbourg, CNRS, Strasbourg, France*
[b]*CREATIS, INSA-Lyon, CNRS, INSERM, Lyon, France*

Abstract

Two recent works have shown the benefit of modeling both high-level factors and their related features to learn disentangled representations with variational autoencoders (VAE). We propose here a novel VAE-based approach that follows this principle. Inspired by conditional VAE, the features are no longer treated as random variables over which integration must be performed. Instead, they are deterministically computed from the input data using a neural network whose parameters can be estimated jointly with those of the decoder and of the encoder. Moreover, the quality of the generated images has been improved by using discrete latent variables and a two-step learning procedure, which makes it possible to increase the size of the latent space without altering the disentanglement properties of the model. Results obtained on two different datasets validate the proposed approach that achieves better performance than the two aforementioned works in terms of disentanglement, while providing higher quality images.

Keywords:
variational autoencoder, disentangled representation learning, generative models

1. Introduction

There is a key challenge to learn disentangled representations where high-level factors would be independently and explicitly encoded [1]. Disentangled representations allow us to manipulate data by modifying high level factors, thus paving the way to easier interpretation of the influence of these factors [2]. It has also been shown that these representations may be more sample-efficient, less sensitive to nuisance variables, and better in terms of generalization [3]. They are thus used in many applications such as face attribute manipulation [4], action generation [5] and image-to-image translation [6].

There is a substantial literature on disentangled representation learning [7]. Since better disentangled models can be obtained under supervision [8], we are only interested in the (semi)-supervised case, and specifically in Variational Autoencoder methods (VAE). VAEs [9] are versatile models of choice to learn such representations in the semi-supervised case [10, 11]. To achieve disentanglement with a VAE, the latent representation is generally divided into two parts [12, 10, 11]: the non-interpretable part and the disentangled part corresponding to variables that explicitly model the factors of interest. But these variables only represent the labels associated with the factors of interest and not the features that can be related to these factors. Consequently, these features are either lost, or entangled in the other latent variables. The works of [13, 14] clearly show that modeling both labels and their associated features improves the model's disentanglement properties. In [13], a feature is associated with each high level factor. The latent space is composed of two different sets of random variables: the first one is composed of features associated with the labels, and the second one models information not directly associated with any of the labels. This implies that the latent space no longer contains the labels, but each label is used to condition its associated feature (in the latent space). Subsequently, this method will be denoted CCVAE (characteristic capturing VAE).

In [14], we proposed a novel conditional dependency structure where both the labels and their features belong to the latent space. In this model, the conditional priors of the features given the label have to be set properly to ensure the desired disentanglement properties. Moreover, the loss function is then composed of two Kullback-Leibler divergences (KLD), that have to be weighted differently, so as to achieve satisfactory results. This makes the approach [14] difficult to use. This second method will be denoted JDVAE (Joint disentanglement of labels and their features with VAE) in the following.

In this article, we propose, as in [13, 14], a VAE-based approach that models explicitly both the high-level factors and their associated features. The proposed model will be denoted CVQVAE (Conditional Vector Quantized VAE), and can be considered as an extension of the work of [14]. To overcome the limitations of [14], the features are no longer considered as random variables over which integration has to be performed. Instead, each feature is here (deterministically) computed from the input data using a neural network whose parameters can be estimated jointly with those of the decoder and of the encoder. These features (as well as the labels and the latent variables)

Email addresses: kaifeng.zou@unistra.fr (Kaifeng Zou), faisan@unistra.fr (Sylvain Faisan), fabrice.heitz@unistra.fr (Fabrice Heitz), sebastien.valette@creatis.insa-lyon.fr (Sébastien Valette)

are then used by the decoder to reconstruct the data. This approach is inspired by conditional VAE (CVAE) [15, 16, 17], except that the conditioning variable is known for CVAE, and computed in CVQVAE. We thus obtain a simplified model (free of conditional priors for the features, and a single KLD loss). Moreover, to improve the quality of the generated images and in particular to generate less blurry images, the Gaussian prior on the latent representation has been replaced by a categorical distribution [18]. The resulting model is more difficult to optimize, but we circumvent this problem with an efficient two-step learning procedure. The proposed model outperforms the two approaches mentioned above on two different datasets.

2. Conditional Vector Quantized Variational AutoEncoder

2.1. Architecture of the model

Without loss of generality, we consider for the presentation of the model that there is one binary high-level factor (label). Note that the extension to several high-level factors is straightforward. The architecture of CVQVAE is illustrated in Fig. 1. The underlying latent representation of the image x is composed of the label y, along with the (other) latent variables z. Finally, c denotes the (continuous) features related to y. As an example, for face images, the "glasses" label y is equal to 1 if the subject is wearing glasses, 0 otherwise. c represents the (continuous) features of the glasses (shape/size/color) and z models the intrinsic properties of the face.

As shown in Fig. 1, the proposed model is composed of an encoder (E_ϕ and C_ϕ), a decoder (D_θ), an embedding space and tokens (ϕ and θ refer to the parameters of the encoder and of the decoder). It relies on the estimation of distributions $q_\phi(y|x)$, $q_\phi(z|x,y)$ and $p_\theta(x|y,z,c)$. Sec. 2.2 explains the reasoning behind this choice and how the distributions are defined. Finally, all the parameters of the model are jointly estimated (Sec. 2.3).

The encoder: It is composed of two neural networks E_ϕ and C_ϕ: C_ϕ takes as input x and outputs the features c and the label distribution $q_\phi(y|x)$. Then, y is set to the most likely label for testing. When training (semi-supervised case), it is set to its true value (if y is known), or sampled from $q_\phi(y|x)$. Finally, E_ϕ takes as input x and y (as tokens) and outputs z_e which is used in conjunction with the embedding space to compute $q_\phi(z|x,y)$. z is either sampled from this distribution during training (See Sec. 2.3) or set to the most likely value during testing.

The embedding space: As in [18], an embedding space, composed of K vectors of R^D, is used to model the categorical distribution $q_\phi(z|x,y)$ (see Sec. 2.2). Moreover, the indices of z are replaced with the vectors of the embedding space (of the same indices) to obtain z_q.

The decoder: The decoder D_θ outputs the distribution $p_\theta(x|y,z,c)$. Under the Gaussian assumption of Eq. 3, this is achieved by outputting the mean of this distribution. As shown in Fig. 1, D_θ is not directly fed with z, y and c. A new variable z_q is computed from z (previous paragraph), tokens are used for representing y (next paragraph) and c and y are combined deterministically to feed D_θ (last paragraph).

The tokens: The label y is not directly fed into E_ϕ and D_θ. As in [14, 5], the label information y is encoded through the use of learnable parameters. They are used here to transfer the y label information to each input of the convolution blocks of E_ϕ and D_θ. As in [14, 5], these parameters are called tokens. We have two sets of learnable tokens for E_ϕ that each consist of five images (each image is associated with a residual block of the encoder). The set is selected according to the value of y. For each convolutional residual block, we concatenate the token and the input of the block along the channel dimension. The same strategy is used for the tokens of D_θ. Additionally to the five images, the two sets related to D_θ have another token that is a scalar one: it is concatenated to z_q (z_q is flattened).

Finally, c is not directly fed into the decoder D_θ. D_θ takes as input a feature vector generated by combining y and c deterministically. To enhance model flexibility, the components of this vector only encode information related to one label ($y = 0$ or $y = 1$): components encoding a property for $y = 0$ are zero if $y = 1$ and vice versa. This procedure is also adapted to the meaning of the high-level factor. As an example, the two high-level factors, "smile" and "glasses", differ from the fact that the features associated with the "smile" label have a meaning whether the person smiles (y=1) or not (y=0), whereas the features associated with the "glasses" label encode the shape/size/color of the glasses, thus having only a meaning in the case $y = 1$ (for $y = 0$, there is nothing more to encode than the fact that $y = 0$). Considering the "glasses" label, c is multiplied by y. It enables us to constrain the obtained vector to be a null vector if y is 0, and to be equal to c otherwise ($y = 1$). For the "smile" label, each label ($y = 0$ and $y = 1$) has its own features. Consequently, the components of c are divided into two equal parts. The first and the second parts represent respectively features for $y = 0$ (neutral face) and for $y = 1$ (smiling face). The components of the first part and of the second part are multiplied by $1 - y$ and y, respectively, so that the first part's components are zero if $y = 1$ and the second part's components are zero if $y = 0$. In the following, N_c denotes the number of components of c that are related to a label: c is of size N_c for the "glasses" label and of size $2N_c$ for the "smile" label.

2.2. Conditional dependency structure

The generative process of CVQVAE is inspired by the work of [11], except that no feature c is defined in [11], and by the CVAE approach [15, 16, 17]. It writes:

$$p_\theta(x, y, z|c) = p_\theta(x|y, z, c)p(y)p(z), \tag{1}$$

where θ represents the parameters of the decoder. Following the idea of CVAE, our purpose should be to approximate the posterior $p_\theta(z, y|x, c)$. However, contrary to [15, 16, 17], the value of c is actually not given, but is computed from x with C_ϕ. Since c is deterministically obtained from x, we have: $p_\theta(z, y|x, c) = p_\theta(z, y|x)$. Consequently, we approximate the posterior $p_\theta(z, y|x)$ by $q_\phi(z, y|x)$ where ϕ represents the parameters of the encoder. It writes:

$$q_\phi(z, y|x) = q_\phi(z|x, y)q_\phi(y|x). \tag{2}$$

The distributions in Eq. 1 and 2 are modeled as follows: y follows a uniform discrete distribution. In accordance with [9],

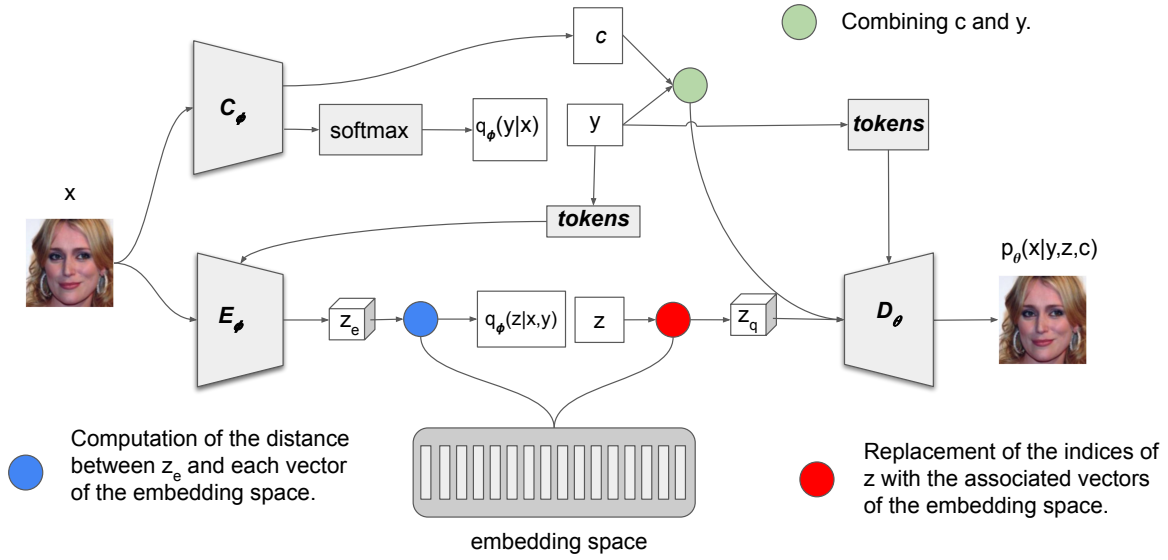


Figure 1: Architecture of CVQVAE. E_ϕ consists of 5 residual blocks. C_ϕ consists of 5 residual blocks followed by one single fully connected layer. D_θ is composed of one fully connected layer followed by 5 residual blocks.

$p_\theta(x|y,z,c)$ is modelled as a Gaussian distribution: its mean is computed by a neural network (the decoder D_θ of parameter θ) that takes as input y, z and c. We have:

$$p_\theta(x|y,z,c) = \mathcal{N}(x; D_\theta(y,z,c), vI), \tag{3}$$

where v is a hyperparameter. As in [11], $q_\phi(y|x)$ is a discrete distribution whose probabilities are provided by a softmax layer. Instead of using the traditional Gaussian assumption, we follow the idea of [18] to model the prior on z and the distribution $q_\phi(z|x,y)$ so as to improve the quality of the generated images.

In [18], z is a map (of size $N_z \times N_z$) and each component of z is a categorical variable that represents the index of a vector of a shared embedding space (this space is composed of K vectors of R^D). Each component of z is independent and identically distributed and follows a uniform discrete distribution. Moreover, $q_\phi(z_{i,j} = k|x)$ (there is no variable y in [18]) is set to 1 for $k = \arg\min_k \|E_{\phi_{i,j}}(x) - e_k\|$, 0 otherwise, where $E_\phi(x)$ is the continuous output of the encoder (and $E_{\phi_{i,j}}(x)$ its value at coordinate (i,j)), and where e_k is the k-th vector of the shared embedding space. We propose to set the posterior $q_\phi(z_{i,j} = k|x,y)$ as a function of $\|E_{\phi_{i,j}}(x,y) - e_k\|$. The smaller $\|E_{\phi_{i,j}}(x,y) - e_k\|$, the larger the probability $q_\phi(z_{i,j} = k|x,y)$ should be. It is defined as:

$$q_\phi(z_{i,j} = k|x,y) = \frac{e^{-\|E_{\phi_{i,j}}(x,y) - e_k\|}}{\sum_{k=1}^{K} e^{-\|E_{\phi_{i,j}}(x,y) - e_k\|}}. \tag{4}$$

Sec. 2.3 explains the relevance of this modeling based on the loss function to be optimized.

2.3. Parameter optimization

If y is known, the optimization of $\log p_\theta(x,y|c)$ can be achieved by maximizing the Evidence Lower BOund (ELBO). Under the reasonable assumption that z and c are conditionally independent given x and y, it writes:

$$\log p_\theta(x,y|c) \geq E_{z\sim q_\phi(z|x,y)} log(p_\theta(x,y,z|c)/q_\phi(z|x,y)). \tag{5}$$

Note that Eq.5 (and Eq. 1) are defined for arbitrary values of c. In the proposed approach, c is set as a function of x but Eq. 5 remains valid because x is a "constant" in Eq. 5. Optimizing Eq. 5 therefore allows to also estimate c. By using Eq. 1, the ELBO term writes (we neglect the constant term $\log p(y)$):

$$E_{z\sim q_\phi(z|x,y)} \log(p_\theta(x|y,z,c)) - KL(q_\phi(z|x,y)\|p(z)), \tag{6}$$

where KL is the Kullback–Leibler divergence. The first term is approximated by a Monte Carlo estimate: we propose to use the Straight-Through Gumbel-Softmax estimator [19] to sample from $q_\phi(z|x,y)$. Moreover, without loss of generality, the term $\log(p_\theta(x|y,z,c))$ in Eq. 6 can be replaced by the mean squared error between x and $D_\theta(y,z,c)$ provided that the second term of Eq. 6 is weighted by a factor β.

The second term can be computed analytically since both distributions $q_\phi(z|x,y)$ and $p(z)$ are discrete. This term acts as a regularization term that constrains the latent space to have good properties: close samples in the latent space should have similar reconstructions. In [18], this term cannot play its role because the choice of the distribution $q_\phi(z_{i,j} = k|x)$ leads to a constant KL divergence. Hence we propose a distribution $q_\phi(z_{i,j} = k|x)$ that allows to obtain such a regularization. Under our hypothesis, the term $-KL(q_\phi(z|x,y)\|p(z))$ can be obtained by summing over (i,j) the entropy of $q_\phi(z_{i,j}|x,y)$ (up to a constant).

If y is unknown (semi-supervised case), y is sampled from $q_\phi(y|x)$ as in [10] using a Gumbel-softmax relaxation and the same loss function is used.

Finally, in both cases, three additional terms are added to the loss function. As in [11], we add a classification loss $\alpha \log q_\phi(y|x)$ to the ELBO term (α is set to 1) because the term

$q_\phi(y|x)$ does not contribute to the loss function if y is known. Moreover, since a Gumbel-softmax relaxation is used to sample z, the gradients are simply copied from z_q to z_e, similarly to straight-through gradient estimation in [18]. Consequently, the parameters of the embedding space do not receive gradients from the loss and we use the additional term presented in [18] to learn the embedding space. Finally, a commitment loss presented in [18] is also used (its weight is set to 0.25 as in [18]).

2.4. Architecture and training variations

To obtain a more detailed evaluation of our contributions, we suggest a range of alternatives, labeled A through F, with our current CVQVAE method denoted as E. Approaches A through D employ standard initialization strategies and the relevant loss function to train the models' parameters, while for approaches E and F, a two-step learning procedure is implemented.

Approaches A and B are based on the proposed CVQVAE except that no feature is associated with y (i.e. c is removed from the model). The resulting models have also the same conditional dependency structure as the model M2 in [11]. The distribution $q_\phi(z|x, y)$ is modeled as proposed in [18] for approach A and as proposed in Sec. 2.2 (Eq. 4) for approach B.

Approach C corresponds to the proposed CVQVAE with standard training. Approach D is based on the CVQVAE with two differences: instead of using a discrete latent representation for z, z follows a zero-centered multivariate normal distribution with unit variance ($p(z) = \mathcal{N}(z; 0, I)$) and the distribution $q_\phi(z|x, y)$ is defined as a Gaussian distribution whose parameters are given by the encoder [9]. Moreover, as in [14], we use AdaIN [20] as a normalization method. AdaIN injects the latent variable z to each layer of the decoder through a fully connected layer. Using AdaIN causes the decoder to attach greater importance to z. The model associated with approach D is denoted as CGVAE (conditional Gaussian VAE).

Approach E is similar to approach C, relying on the proposed CVQVAE method. Approach F employs a model named CGVAE2, which is similar to CGVAE but without the use of AdaIN. Both approaches use a two-step learning procedure. The rationale behind two-step learning is that the optimization problem would be easier to solve if c was known: to this end, we start to train a simplified model (approach D with a small latent space) that also has the C_ϕ network (that enables us to compute y and c) as well as the tokens. Then, for the estimation of the parameters of CVQVAE (approach E) or of CGVAE2 (approach F), the parameters of the C_ϕ network and the tokens are initialized with those obtained by approach D. Note that these parameters are frozen during the first iterations of the optimization procedure.

3. Experiments

Implementation details: We experiment on the CelebA [21] and CheXpert [22] datasets each containing more than 200000 images (80% is used for training) of size 128^2. The first dataset is composed of labeled face images, on which we conduct quantitative experiments (for the "glasses" and "smile" labels), as well as qualitative experiments (for the "beard" and "makeup" labels). The second dataset is composed of labeled X-ray chest images on which three experiments are conducted.

Hyperparameters (N_z, N_c, K, D, β) have been tuned using a cross-validation strategy in the experiment relative to the "glasses" label with CVQVAE. Other experiments use the same tuned hyperparameters: N_c has been set to 16 and β to 1e-4 (see text under Eq. 6). When modeling z as a categorical variable, the size of the latent space z has been set to $S_z = N_z \times N_z$ with $N_z = 8$, and the embedding space is composed of $K = 512$ vectors of dimension $D = 16$. For CGVAE2 (approach F), S_z has been set to 1024 which is equal to the number of components of z_q for CVQVAE (1024=8×8×16). This allows for a fair comparison between CVQVAE and CGVAE2. For CGVAE (approach D), we set S_z to a small value (100) to obtain a simplified model with better convergence properties. Note that approach D is mainly useful to initialize CVQVAE and CGVAE2.

The models have been trained independently for each experiment. We used the Adam optimizer with a learning rate equal to 10^{-4}, a batch size of 32 and a supervision rate set to 0.2. The experiments were conducted using PyTorch 1.9 and CUDA 10.2, leveraging a Nvidia 1080Ti graphics card.

Evaluation metrics: We consider two different tasks: the classification task, and the exchange of high level factors and their related features between two images (so as to measure the disentangled properties of the model). The classification task is assessed using the Balanced Classification Accuracy (BCA).

The disentangled ability of the model is evaluated by computing the success rate of swapping the attributes. In order to distinguish between classification errors and disentanglement errors, the true labels are used to perform this task: we select random pairs of images composed of one image of both classes denoted $x_{y=1}$ and $x_{y=0}$. Their values of c and y are then exchanged to create two fake images. They are generated by feeding the decoder with z_0, c_1, $y = 1$ (for the first one), and with z_1, c_0, $y = 0$ (for the second one), where z_0, c_0, and z_1, c_1 denote the latent variables and the features computed from $x_{y=0}$, and $x_{y=1}$, respectively. As an example, for the "glasses" label, the first fake image should exhibit the face of $x_{y=0}$ with the glasses of $x_{y=1}$ and the second fake image should show the face of $x_{y=1}$ without glasses. We consider that the swap ("from 0 to 1" or "from 1 to 0") is successful when the associated generated image is well-classified by an independent classifier based on ResNet 50 [23]. We denote by SR(+) (resp. SR(-)) the success rates for going from "0 to 1" (resp. "1 to 0").

Note that SR(+) and SR(-) are not perfect evaluation criteria for measuring disentanglement properties of the models. As an example, for the "glasses" label, SR(+) does not check that the glasses added to $x_{y=0}$ are those of $x_{y=1}$. Consequently, some swapping results will be presented to check whether the features are well-transferred or not. Moreover, in order to obtain a quantitative criterion, we propose to compute the Classification Feature Distance (CFD) as the L2 norm between two outputs of the last layer of the independent classifier. These two outputs are obtained by feeding the classifier once with the original image $x_{y=1}$ ($x_{y=0}$, resp.) and once with the fake image that has the same values of c and y as the original image: the fake image

Table 1: Results for the "glasses" label in terms of (*i*) success rates of swapping SR(-) and SR(+), (*ii*) CFD, and (*ii*) FID that compares the distribution of fake images with the one of real images. The models are described in the text.

Model	SR(-)	SR(+)	CFD	FID
A	91.07%	72.26%	0.200	20.38
B:A+KLD	93.46%	77.46%	0.142	20.03
C:B+c	99.61%	76.91%	0.112	20.78
D:CGVAE	99.99%	62.96%	0.114	21.27
E:CVQVAE	**100%**	**79.13%**	**0.093**	**20.05**
F:CGVAE2	99.85%	72.83%	0.097	20.51

Figure 2: Attribute swapping ("glasses label") using C, D and E approaches. The second, third and fourth images should be $x_{y=0}$ with the glasses of $x_{y=1}$.

is generated by the decoder with z_0, c_1, $y = 1$ (z_1, c_0, $y = 0$, resp.). As a reminder, z_0, c_0, and z_1, c_1 denote the latent variables and the features computed from $x_{y=0}$, and $x_{y=1}$. The CFD is based on the assumption that an ideal attribute swap should not change the features extracted by the classifier. We also compute one Fréchet Inception Distance [24] (FID), that compares the distribution of fake images with the one of real images.

3.1. Comparison of approaches A to F

Results obtained with approaches A to F are provided in Tab 1 for the "glasses" label. A and B perform well, but they cannot transfer the features of the glasses to another image since glasses are not explicitly modeled. Moreover, the regularization over the latent space, induced by the proposed modeling of $q_\phi(z|x, y)$ (Eq. 4), improves the disentanglement properties of the model: SR(+) and SR(-) obtained with B are larger than those obtained with A. Thanks to the modeling of c, approach C obtains better results in terms of SR(-) and CFD. However, visual inspection of the results show that c not only carries information about the glasses but also about the face, as illustrated in Fig. 2 (C): the glasses are well transferred from $x_{y=1}$ to $x_{y=0}$ but some features of the faces are also transferred. The use of AdaIN in approach D results in a slightly deterioration of the model's disentanglement properties ($SR(+)$ decreases), and the modeling of z (the latent space is only 100) leads to a reduction of image quality. However, the modification of c does not change the face anymore (see Fig. 2 (D)), thus showing that c is free of any information about the face.

Results obtained with Approach E (CVQVAE) enable to obtain the best results in terms of quantitative criteria (Tab. 1). Moreover, visual inspection of the results (Fig. 2(E)) shows that the properties of the glasses are relatively well transferred, while preserving the main features of the face. Finally, as in [14, 13], these results clearly illustrate the interest of modeling the features related to the high-level factors. Indeed, as shown by the values of SR(+) and SR(-), CVQVAE yields better disentanglement representations than methods A and B for which the properties of the glasses are not modelled. Note also that AdaIN is not used in the CVQVAE approach. AdaIN was shown in [14] to improve the reconstruction of the images. However, it is no longer worth using AdaIN when the size of the latent space is increased. Furthermore, the use of AdaIN slightly weakens the disentanglement properties of the model.

Finally, while CGVAE2 yields very satisfactory results, CVQVAE provides better results than CGVAE2, both in terms of disentanglement and image quality. Moreover, increasing β for CGVAE2 produces disentanglement properties similar to CVQVAE, but at the expense of image quality (they are blurry, data not shown). These results illustrate the relevance of using a discrete latent representation.

3.2. Comparison with state-of-the-arts methods

The proposed approach is compared with two VAE-based approaches that also model the features related to the high-level factors: CCVAE [13] and JDVAE [14] and with the model M2 of [11] with a Gaussian prior for z (the features are not modelled). Finally, for all methods, the architectures of the encoder and of the decoder are similar to those of JDVAE[14]. For these approaches, the size of the latent space S_z has been set to 100: larger latent space results in a model that is more difficult to optimize and leads to a reduction in the performance model.

Results obtained with the "glasses" and with the "smile" labels are provided in Tab. 2 and in Fig. 3. First, all methods obtain good classification accuracy (BCA) despite a supervision rate equal to 0.2. Note that the accuracy of Resnet 50 is 98.69% and 93.07% for the "glasses" and the "smile" labels, respectively. With respect to the quality of the fake images (FID), CVQVAE provides images of better quality, thus justifying the use of a larger latent space. Regarding the success rates of swapping (SR(-) and SR(+)), results obtained with M2 are less satisfactory than those obtained with the other methods, showing once again the interest of modeling both the high-level factors and their features. An analysis of the results obtained by CCVAE, JDVAE, and CVQVAE for SR(+) and SR(-) requires to consider the labels separately. For the "glasses" label, results obtained with CCVAE are relatively satisfactory but the features of glasses are not well-transferred (CFD values in Tab. 2 and results in Fig. 3). Results are more satisfactory with JDVAE [14]. However, CVQVAE obtains the best success rates for adding and removing glasses. Additionally, our method correctly extracts most of the features of the glasses from the image $x_{y=1}$ and reconstructs them reasonably well on $x_{y=0}$ (Fig. 3), which shows that the label and features of the glasses have been properly disentangled from the attributes of the faces.

For the "smile" label, visual inspection of the reconstructed images (without attribute swapping, data not shown) shows that JDVAE and CCVAE have difficulties in extracting the features related to the smile. As an example, for a neutral face with open mouth, its reconstruction shows a closed mouth. Similarly, for a smiling face with wide open mouth, the mouths of the reconstructed images are less open. On the opposite, CVQVAE provides better reconstructions. Our hypothesis is that the problem

Table 2: Results for the "glasses" and the "smile" labels in terms of CFD, success rates of swapping SR(-) and SR(+), FID, and in terms of BCA.

	glasses					smile				
Model	CFD	SR(-)	SR(+)	FID	BCA	CFD	SR(-)	SR(+)	FID	BCA
CCVAE	0.145	95.52%	47.89%	21.10	96.26%	0.052	**96.97%**	74.75%	16.14	**90.80%**
JDVAE	0.098	94.98%	64.25%	21.66	**97.09%**	0.059	81.80%	79.34%	16.36	90.52%
M2	0.274	80.34%	33.02%	22.27	96.13%	0.069	44.07%	50.72%	16.75	90.48%
CVQVAE	**0.093**	**100%**	**79.13%**	**20.05**	96.67%	**0.049**	89.25%	**90.25%**	**14.54**	90.09%

Table 3: Results for three different pathologies in terms of CFD, success rates of swapping (SR = (SR(-)+SR(+))/2), FID, and in terms of BCA.

	cardiomegaly				atelectasis				consolidation			
Model	CFD	SR	FID	BCA	CFD	SR	FID	BCA	CFD	SR	FID	BCA
CCVAE	0.298	57.51%	8.01	**80.33%**	0.137	49.38%	7.67	71.92%	0.215	62.16%	7.04	**80.67%**
JDVAE	0.261	60.27%	7.82	79.44%	0.138	50.92%	7.62	71.11%	0.250	62.23%	7.07	80.07%
M2	0.347	47.36%	8.99	79.58%	0.233	41.99%	7.99	70.99%	0.446	36.93%	8.43	80.58%
CVQVAE	**0.169**	**69.97%**	**7.13**	79.92%	**0.134**	**64.55%**	**6.87**	**72.86%**	**0.117**	**72.36%**	**6.91**	80.46%

is made easier with CVQVAE because the components of c that represent the neutral face are not the same than those representing the smiling face. Regarding the success rates of swapping (SR(-) and SR(+)), results obtained with CCVAE look satisfactory, especially for SR(-) but this number is biased. SR(-) is actually greater than the accuracy of ResNet 50 (when classifying neutral face). This shows that it is easier for the classifier to classify neutral fake images than real neutral images. This is due to the fact that the neutral images obtained with CCVAE are actually too neutral. Indeed, we can observe that the features related to the smile are not properly transferred to other images (see Fig. 3). As we saw previously, this is not only a feature transfer problem, but also a feature extraction problem. Results are actually slightly improved with JDVAE [14], but the best results are undoubtedly obtained with CVQVAE.

Fig. 4 shows results obtained with other labels, which further illustrate the versatility of the model.

In addition, we show the effectiveness of our model on the CheXpert dataset (Tab. 3). Three different experiments have been conducted. In these experiments, $y = 1$ is associated to a pathology ("cardiomegaly", "atelectasis" or "consolidation") and $y = 0$ is related to the "non finding" label (no pathology). Quantitative results show once again that CVQVAE outperforms the other methods. Since no feature has been related to the label $y = 0$ (it was also the case for the "glasses" label), it is possible to reconstruct an image with a pathology as an image without pathologies. The difference between its reconstruction and its reconstruction as a "free of pathology" image reveals the influence of the pathology (in green on Fig. 5).

3.3. Exploration in the feature space c

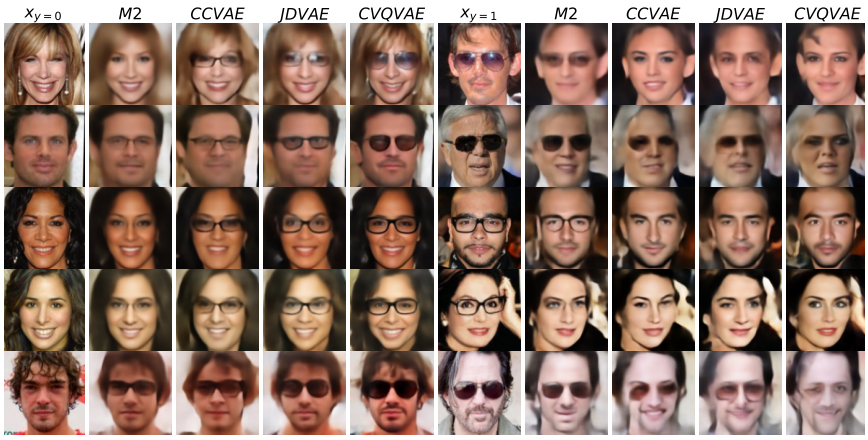
We have carried out several experiments on the information encoded by the variable c. In Fig. 6, fake images are generated by feeding the decoder with z_1, y_1 and $c = c1 + \alpha(c2 - c1)$ ($\alpha \in [0, 1]$), where z_i, y_i, and c_i denote the variables related to image x_i (i=1 or 2) with y_1=y_2. As an example, for the "glasses" label, if $y_1 = 1$, the generated glasses should be similar to those of x_1 (if α is close to 0), of x_2 (if α is close to 1), or in-between (for other values of α). Moreover, in all cases, the generated face should be the one of x_1. Results shown in Fig. 6 are consistent with our expectations: interpolation in the feature space c results in a smooth transition between smiles (top), neutral faces (middle), or types of glasses (bottom).

In Fig. 7, the influence of the magnitude of c is shown: images are generated by feeding the decoder with z, y and λc ($\lambda \in [0, 2]$), where z, y, and c are computed from x. Results are shown for the "smile" label for $y = 1$ (Fig. 7, top) and for the "glasses" label for $y = 1$ (Fig. 7, bottom).

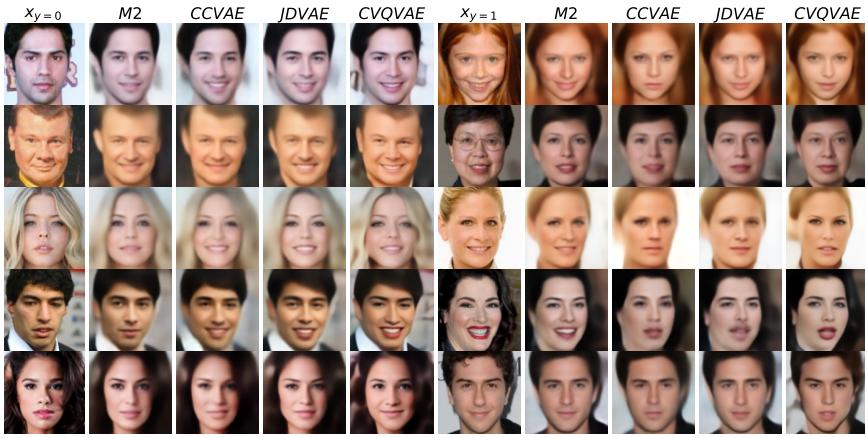
Increasing or decreasing the magnitude of c leads to amplifying or reducing the related features in the generated images. For example, with $\lambda = 2$, the frames of glasses become very dark and wide, and the way of smiling is also exaggerated (the mouth is notably more open). Moreover, even if $y = 1$, a null value for c ($\lambda = 0$) prevents glasses from being generated.

3.4. Multiple attribute disentanglement

Our approach can easily be extended to the multiple attribute case. Two high-level factors are considered hereafter: y_1 and y_2 denote the labels, and c_1 and c_2 denote the related features. Equations of Sec. 2 remain valid by setting y to (y_1, y_2), and c to (c_1, c_2). We use the following assumption: $p(y) = p(y_1)p(y_2)$ and $q_\phi(y|x) = q_\phi(y_1|x).q_\phi(y_2|x)$. The architecture of the model can easily be extended to the two high-level factor cases. This has been achieved by modifying the last layer of the C_ϕ network. Results obtained are shown in Fig.8 where the purpose is to transfer the glasses and the smile of $x_{y_1=1,y_2=1}$ to $x_{y_1=0,y_2=0}$.



(a)



(b)

Figure 3: Attribute swapping ("glasses" label (a) and "smile" label (b)) with M2, CCVAE, JDVAE and CVQVAE. For each row, the second, third, fourth and fifth images should be $x_{y=0}$ with the glasses (a) or smile (b) of $x_{y=1}$. The four rightmost images should be $x_{y=1}$, but without glasses (a) or with the neutral attitude of $x_{y=0}$ (b).

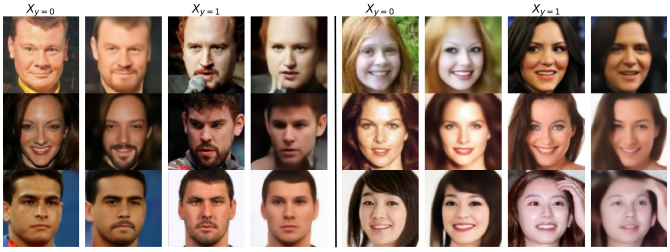


Figure 4: Attribute swapping for "beard" (left) and "makeup" labels (right). Presentation is similar to Fig. 3 except only results of CVQVAE are shown.

4. Conclusion

Our CVQVAE approach clearly outperforms the state-of-the-art approaches, both in terms of disentanglement and in terms of generated image quality. Future works could adapt CVQVAE to the architecture of a hierarchical VQ-VAE (such as the one proposed in VQ-VAE2 [25]) and GAN (such as VQGAN[26]) so as to further improve the quality of generated images.

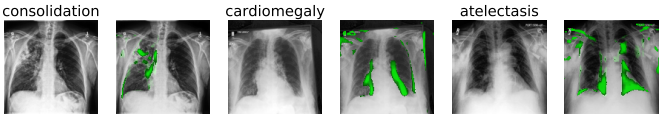


Figure 5: Results obtained with three different pathologies on the CheXpert dataset. For each pathology, the original image (with the name of the pathology at its top) is on the left, and the regions in green (at the right of the original image) represent regions that differ the most between the reconstruction and the "pathology-free" reconstruction.

Acknowledgement

This work was funded by the TOPACS ANR-19-CE45-0015 project of the French National Research Agency (ANR).

References

[1] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE transactions on pattern analysis and machine intelligence 35 (2013) 1798–1828.

[2] K. Zou, S. Faisan, F. Heitz, M. Epain, P. Croisille, L. Fanton, S. Valette, Disentangled representations: towards interpretation of sex determination from hip bone, The Visual Computer (2023).

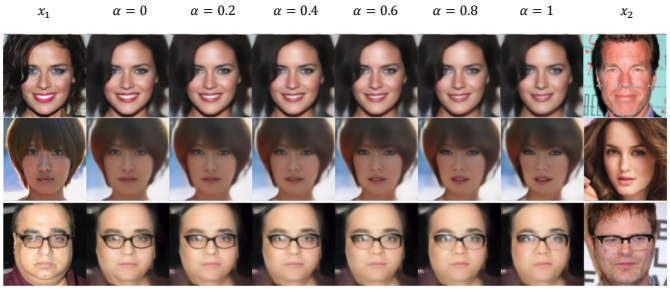


Figure 6: Interpolation in the feature space with different values of α using x_1 and x_2 (see text for details). Each column corresponds to the generated results for α given at its top, with the exception of the left and right columns that correspond to x_1 and x_2. The bottom row is related to the "glasses" label (with $y_1 = 1$) while the two other rows correspond to the "smile" label ($y_1 = 1$ for the top row and $y_1 = 0$ for the middle one).

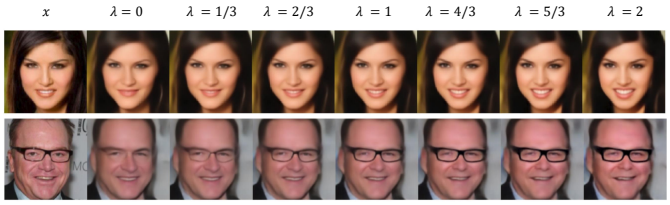


Figure 7: Increasing or decreasing the magnitude of c from x with different values of λ (see text for details). Each column corresponds to the generated results for λ given at its top, with the exception of the left column that corresponds to x. The top row is related to the "smile" label (with $y = 1$) while the bottom row corresponds to the "glasses" label (with $y = 1$).

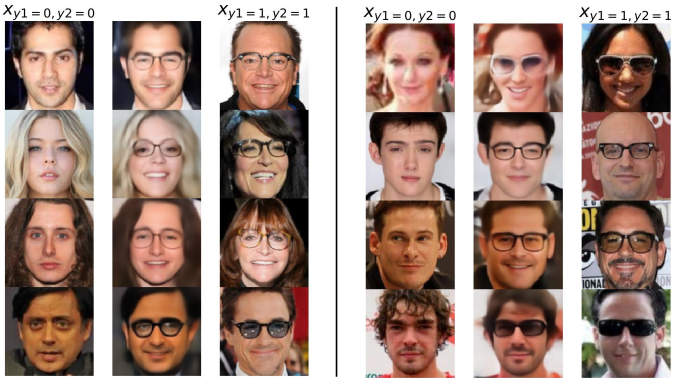


Figure 8: Multiple attributes transfer with CVQVAE. There are 8 different examples, 2 per row, so that there are 4 on the left and 4 on the right. For each example, glasses and smile from $x_{y_1=1,y_2=1}$ are transfered to $x_{y_1=0,y_2=0}$, and the resulting image is located between $x_{y_1=0,y_2=0}$ and $x_{y_1=1,y_2=1}$.

[3] S. Van Steenkiste, F. Locatello, J. Schmidhuber, O. Bachem, Are disentangled representations helpful for abstract visual reasoning?, Advances in Neural Information Processing Systems 32 (2019).

[4] Z. He, W. Zuo, M. Kan, S. Shan, X. Chen, Attgan: Facial attribute editing by only changing what you want, IEEE transactions on image processing 28 (2019) 5464–5478.

[5] M. Petrovich, M. J. Black, G. Varol, Action-conditioned 3D human motion synthesis with transformer VAE, in: International Conference on Computer Vision (ICCV), 2021.

[6] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, M.-H. Yang, Diverse image-to-image translation via disentangled representations, in: Proceedings of the European conference on computer vision (ECCV), 2018.

[7] X. Liu, P. Sanchez, S. Thermos, A. O'Neil, S. Tsaftaris, Learning disentangled representations in the imaging domain, Medical Image Analysis 80 (2022).

[8] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, O. Bachem, Challenging common assumptions in the unsupervised learning of disentangled representations, in: International Conference on Machine Learning, PMLR, 2019, pp. 4114–4124.

[9] D. P. Kingma, M. Welling, Auto-Encoding Variational Bayes, in: International Conference on Learning Representations, (ICLR), 2014.

[10] N. Siddharth, B. Paige, J.-W. van de Meent, A. Desmaison, N. D. Goodman, P. Kohli, F. Wood, P. H. S. Torr, Learning disentangled representations with semi-supervised deep generative models, in: Advances in Neural Information Processing Systems, volume 30, 2017.

[11] D. Kingma, D. Rezende, S. Mohamed, M. Welling, Semi-supervised learning with deep generative models, in: Advances in Neural Information Processing Systems, 2014.

[12] L. Maaløe, C. K. Sønderby, S. K. Sønderby, O. Winther, Auxiliary deep generative models, in: Proceedings of The 33rd International Conference on Machine Learning, 2016, pp. 1445–1453.

[13] T. Joy, S. Schmon, P. Torr, N. Siddharth, T. Rainforth, Capturing label characteristics in VAEs, in: International Conference on Learning Representations, (ICLR), 2020.

[14] K. Zou, S. Faisan, F. Heitz, , S. Valette, Joint disentanglement of labels and their features with VAE., in: IEEE International Conference on Image Processing (ICIP), 2022.

[15] K. Sohn, H. Lee, X. Yan, Learning structured output representation using deep conditional generative models, Advances in neural information processing systems 28 (2015).

[16] X. Yan, J. Yang, K. Sohn, H. Lee, Attribute2image: Conditional image generation from visual attributes, in: European conference on computer vision (ECCV), 2016, pp. 776–791.

[17] Y.-C. Cheng, H.-Y. Lee, M. Sun, M.-H. Yang, Controllable image synthesis via SegVAE, in: European conference on computer vision (ECCV), 2020, pp. 159–174.

[18] A. Van Den Oord, O. Vinyals, et al., Neural discrete representation learning, Advances in neural information processing systems 30 (2017).

[19] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, arXiv preprint arXiv:1611.01144 (2016).

[20] X. Huang, S. Belongie, Arbitrary style transfer in real-time with adaptive instance normalization, in: Proceedings of International Conference on Computer Vision (ICCV), 2017, pp. 1501–1510.

[21] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of International Conference on Computer Vision (ICCV), 2015.

[22] J. A. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. L. Ball, K. S. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. Langlotz, B. N. Patel, M. P. Lungren, A. Ng, Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison, AAAI Conference on Artificial Intelligence, 2019.

[23] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the conference on computer vision and pattern recognition (CVPR), 2016, pp. 770–778.

[24] 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, Advances in neural information processing systems 30 (2017).

[25] A. Razavi, A. van den Oord, O. Vinyals, Generating diverse high-fidelity images with VQ-VAE-2, in: Advances in Neural Information Processing Systems, 2019.

[26] P. Esser, R. Rombach, B. Ommer, Taming transformers for high-resolution image synthesis, in: Proceedings of the conference on computer vision and pattern recognition (CVPR), 2021, pp. 12873–12883.