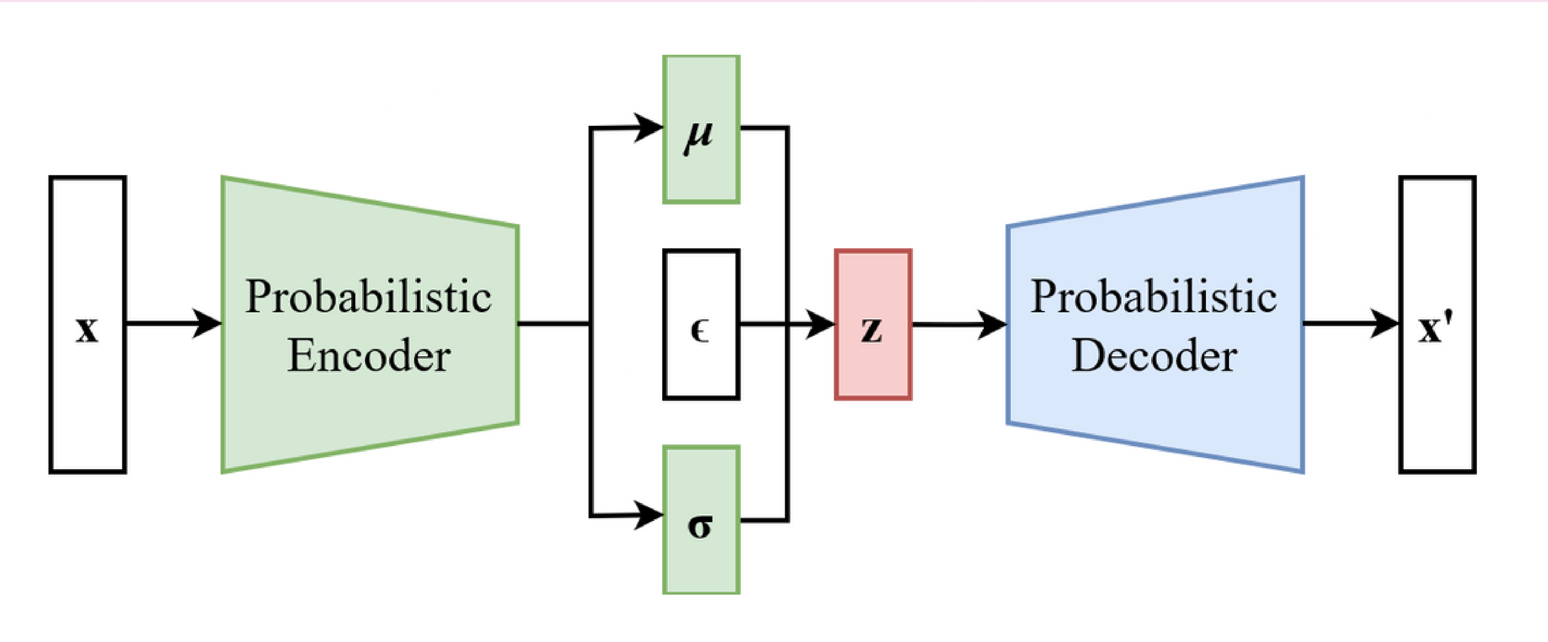
Capturing Label Characteristics in VAEs

A new method for including specific features within latent variables when dealing with supervised generation.

Quentin Bourbon **Vincent Bardusco**

1

Use of Variational Autoencoders

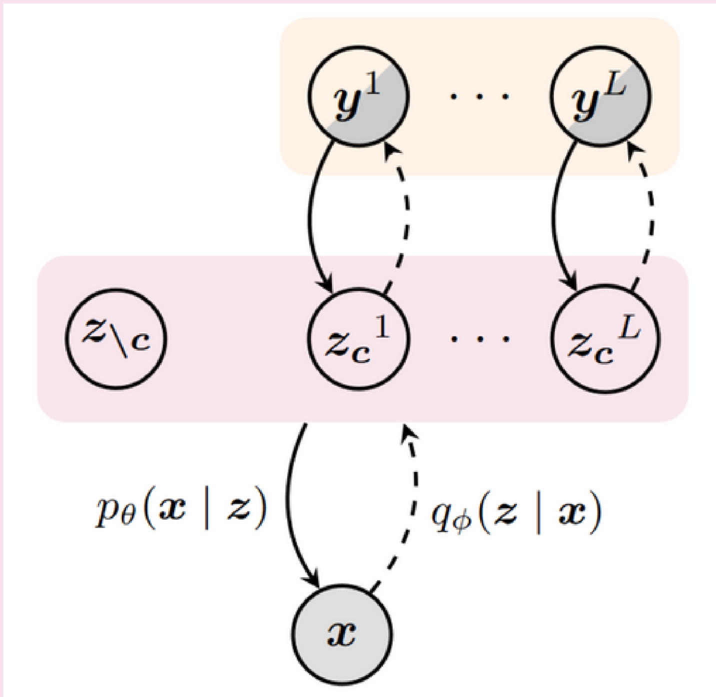


More than just an autoencoder, the VAE allows us to control the shape of our latent space. This allows you to simply walk around the latent space and control the output.

However, it could be a real challenge to change one or two features of the output image without changing everything, as the features are encoded in all the latent variables.

2

Graphical model of CCVAE



One solution would be to include the labels of the images during generation, so that the labels could be encoded in the latent variables. Some work has placed labels in feature space, but this destroys the representation.

We also want to capture more information than just the label itself.

Then you can decide to put the representation of a label into a single latent variable, whose independence is guaranteed by a classifier and a conditional prior that learn in parallel which latent goes with which label.

Thus, we split the latent space into a space of latent characteristic variables and a space of latent style variables.

3

Implementation

$$\mathcal{L}_{\mathrm{CCVAE}}(\boldsymbol{x}, \boldsymbol{y}) = E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\frac{q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c)}{q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x})} \log\left(\frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p_\psi(\boldsymbol{z} \mid \boldsymbol{y})}{q_\varphi(\boldsymbol{y} \mid \boldsymbol{z}_c) q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}\right)\right] + \log q_{\varphi,\phi}(\boldsymbol{y} \mid \boldsymbol{x}) + \log p(\boldsymbol{y})$$

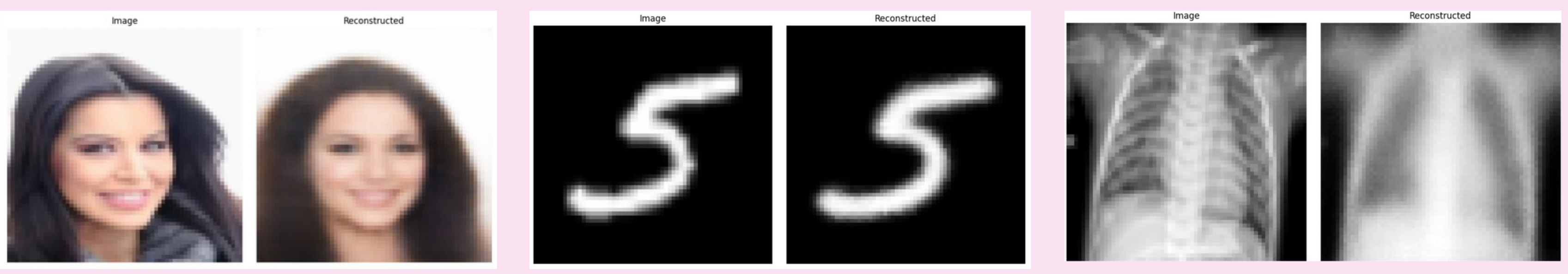
Encoder	Decoder
Input 32 x 32 x 3 channel image	Input $\in \mathbb{R}^m$
$32 \times 3 \times 4 \times 4$ Conv2d stride 2 & ReLU	$m \times 256$ Linear layer
$32 \times 32 \times 4 \times 4$ Conv2d stride 2 & ReLU	$128 \times 256 \times 4 \times 4$ ConvTranspose2d stride 1 & ReLU
$64 \times 32 \times 4 \times 4$ Conv2d stride 2 & ReLU	$64 \times 128 \times 4 \times 4$ ConvTranspose2d stride 2 & ReLU
$128 \times 64 \times 4 \times 4$ Conv2d stride 2 & ReLU	$32 \times 64 \times 4 \times 4$ ConvTranspose2d stride 2 & ReLU
$256 \times 128 \times 4 \times 4$ Conv2d stride 1 & ReLU	$32 \times 32 \times 4 \times 4$ ConvTranspose2d stride 2 & ReLU
$256 \times (2\times m)$ Linear layer	$3 \times 32 \times 4 \times 4$ ConvTranspose2d stride 2 & Sigmoid

Classifier	Conditional Prior
Input $\in \mathbb{R}^{m_c}$	Input $\in \mathbb{R}^{m_c}$
$m_c \times m_c$ Diagonal layer	$m_c \times m_c$ Diagonal layer

4

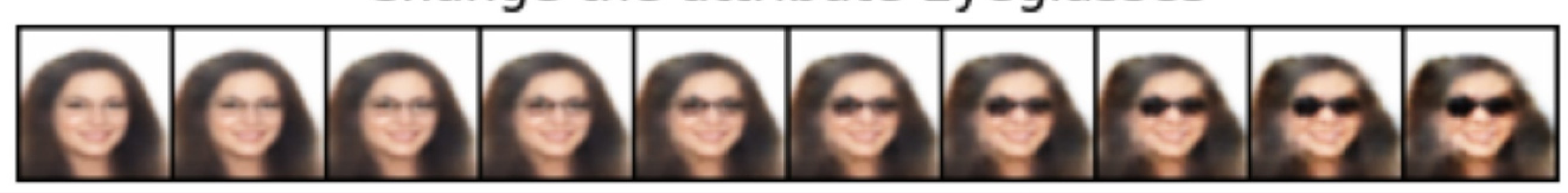
Results with CCVAE

Reconstruction

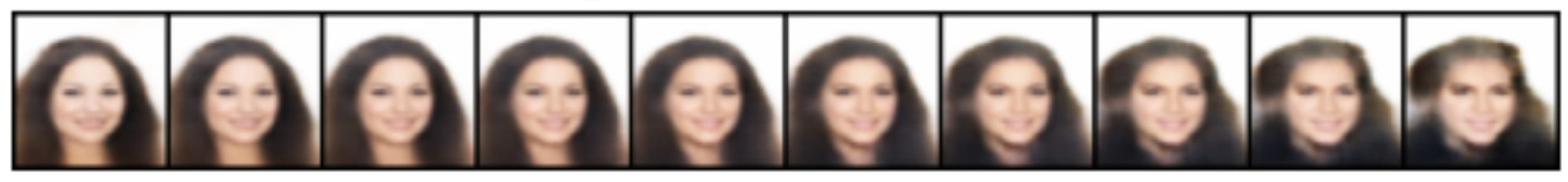


Latent walk

Change the attribute Eyeglasses



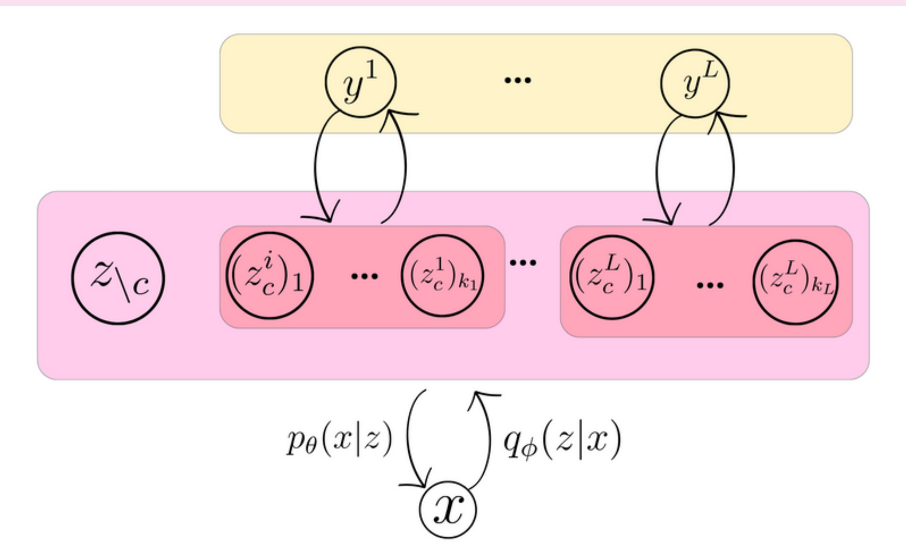
Change the attribute Male



5

Graphical model of MCCVAE

The problem is that sometimes it's not enough to learn a complex label with only 1 latent variable, so we use the same mechanism that allows a label to be described by multiple latent variables. We called it MCCVAE.



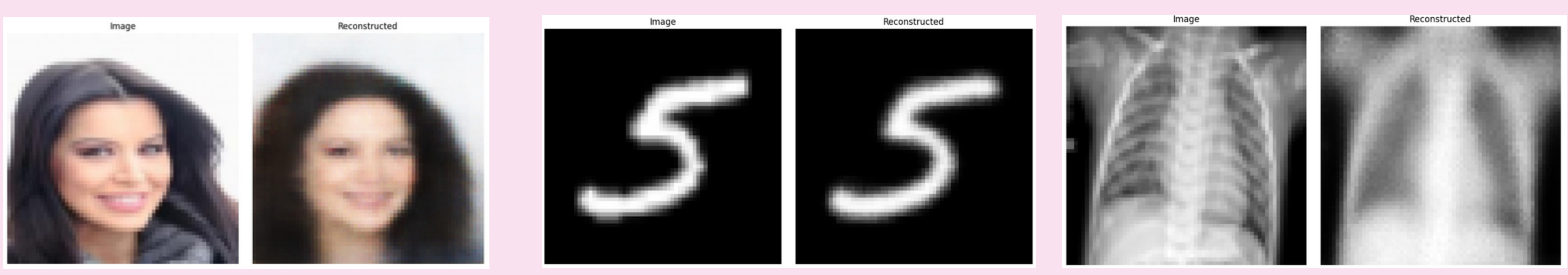
label = several features

Dedicating multiple latent variables to describe a single label could be a way to:

- improve the generation
- identify the components that make up the label

MMCVAE appears to be better suited for generating complex data.

Reconstruction

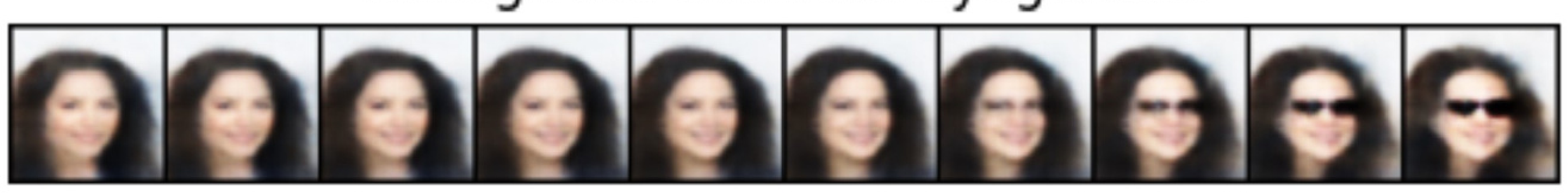


6

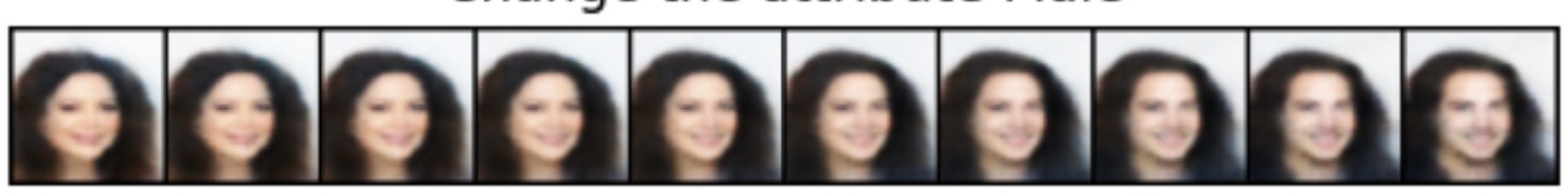
Results with MCCVAE

When changing all the set of a characteristic variables

Change the attribute Eyeglasses



Change the attribute Male



When changing characteristic variables 1 by 1

Change the attribute Male

Change the attribute Male