Conditional Subspace VAE

Alexander Lyzhov, Artem Shafarostov,
Marina Pominova, Elizaveta Lazareva

CSVAE architecture

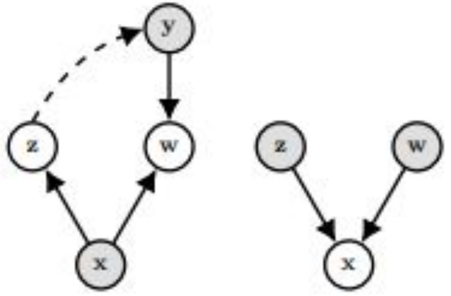
Idea: minimize MI between label and subspace
Can find features correlated with labels for manipulation

Alternative to VAE, CondVAE, CondVAE-info

Introduced in “Learning Latent Subspaces in Variational Autoencoders” (NIPS’18)

CSVAE architecture

$$\log p_{\theta,\gamma}(\mathbf{x},\mathbf{y},\mathbf{w},\mathbf{z}) = \log p_{\theta}(\mathbf{x}|\mathbf{w},\mathbf{z}) + \log p(\mathbf{z}) + \log p_{\gamma}(\mathbf{w}|\mathbf{y}) + \log p(\mathbf{y})$$



$$\begin{aligned}
&- \beta_1 \mathbb{E}_{q_\phi(\mathbf{z},\mathbf{w}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{w},\mathbf{z})\right] + \beta_2 D_{KL}\left(q_\phi(\mathbf{w} \mid \mathbf{x},\mathbf{y}) \parallel \log p(\mathbf{w} \mid \mathbf{y})\right) \\
&+ \beta_3 D_{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x},\mathbf{y}) \parallel p(\mathbf{z})\right) + \beta_4 \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})\mathcal{D}(\mathbf{x})}\left[\int_Y q_\delta(\mathbf{y} \mid \mathbf{z}) \log q_\delta(\mathbf{y} \mid \mathbf{z})\, d\mathbf{y}\right] \\
&- \log p(\mathbf{y})
\end{aligned}$$

$$- \beta_5 \mathbb{E}_{q(\mathbf{z}|\mathbf{x})\mathcal{D}(\mathbf{x},\mathbf{y})}\left[\log q_\delta(\mathbf{y} \mid \mathbf{z})\right].$$

CSVAE architecture

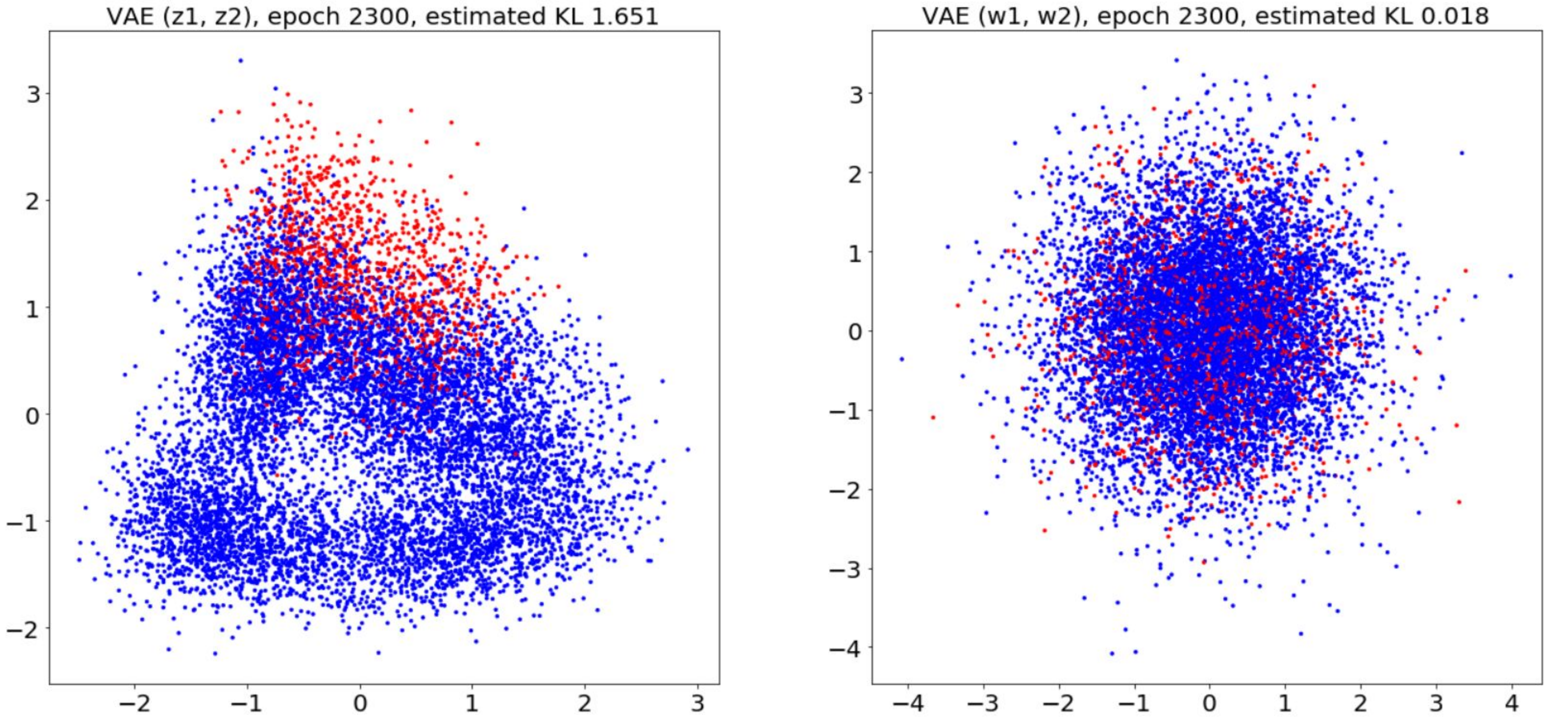
Idea: minimize MI between label and subspace
Can find features correlated with labels for manipulation

Encoders: x->z, xy->w
Decoders: zw->x, w->y (unusual)
Optimizer 1: for decoder w->y
Optimizer 2: for all other encoders/decoders

Label Y = 0/1. Latent space is z1, z2, w1, w2.

w1, w2 shouldn’t have information about Y, but z1, z2 shouldn’t have it.

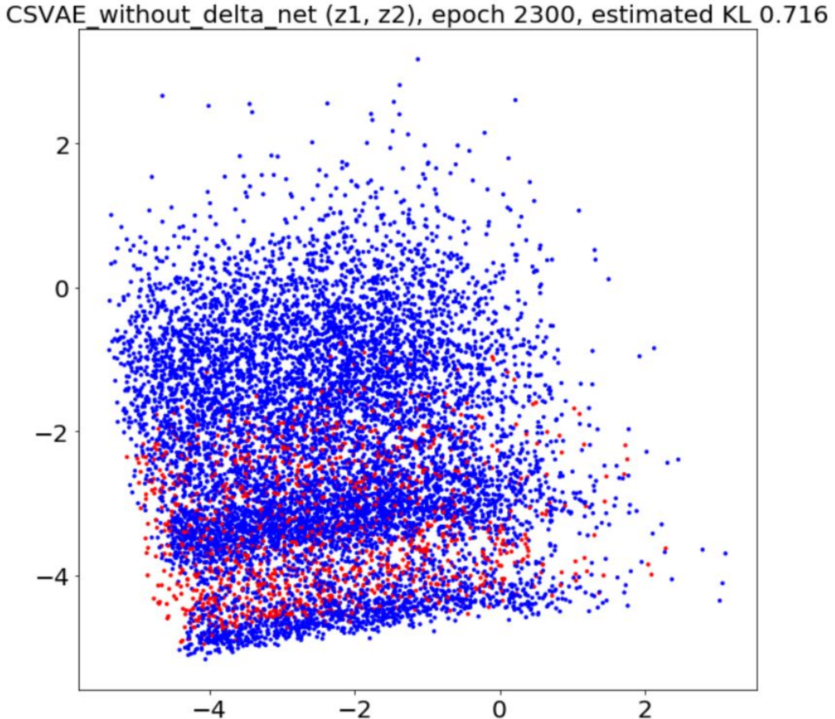
Vanilla VAE

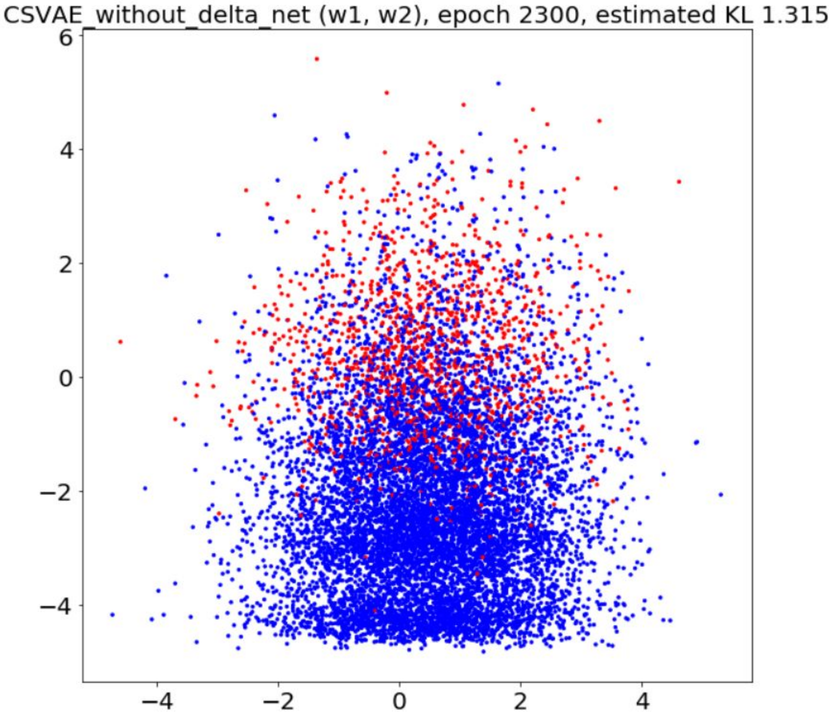


Label Y = 0/1. Latent space is z1, z2, w1, w2.

w1, w2 shouldn't have information about Y, but z1, z2 shouldn't have it.

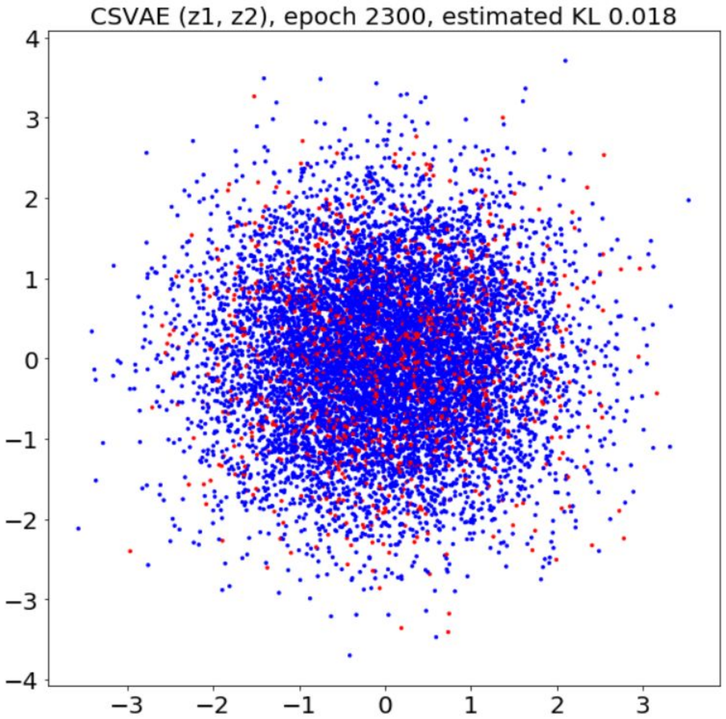
CSVAE without adversarial component

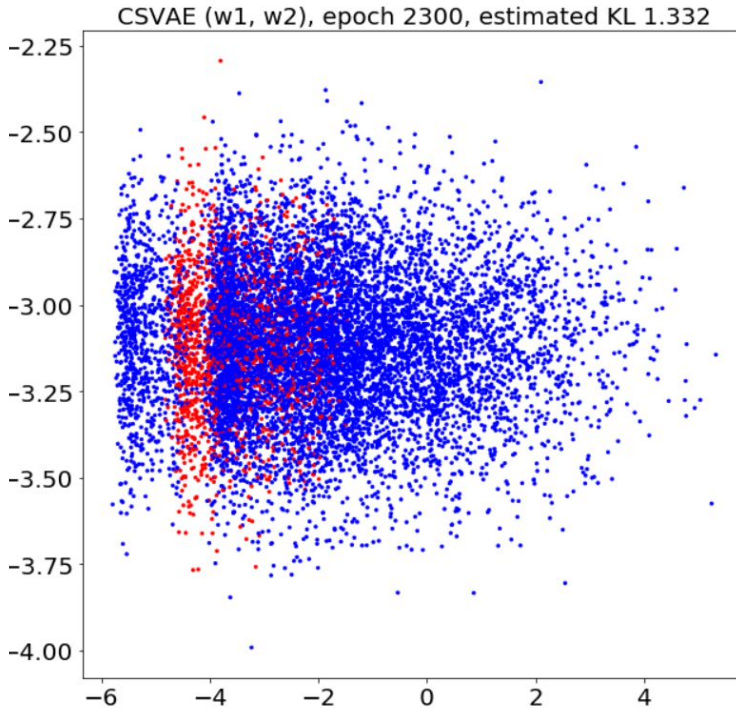




Label Y = 0/1. Latent space is z1, z2, w1, w2.
w1, w2 should have information about Y, but z1, z2 shouldn't have it.

CSVAE





CelebA

- **200,000** images of celebrity faces
- **40** labelled attributes

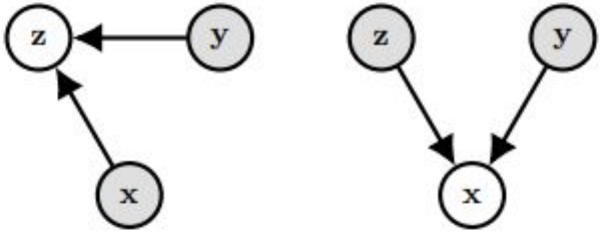
Experiments details

- First trained basic VAE architecture
- Used **same encoder-decoder architecture** and **pretrained weights** for conv layers in all experiments

CelebA

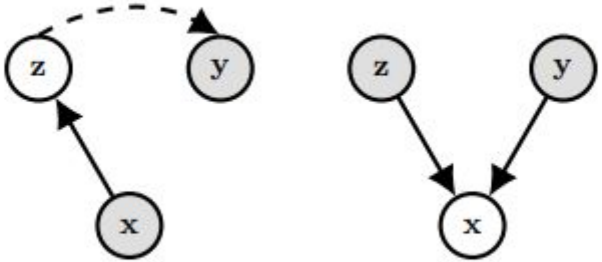
Comparison of CondVAE with other approaches:

CondVAE



- Generated samples are conditioned on label y

CondVAE-info

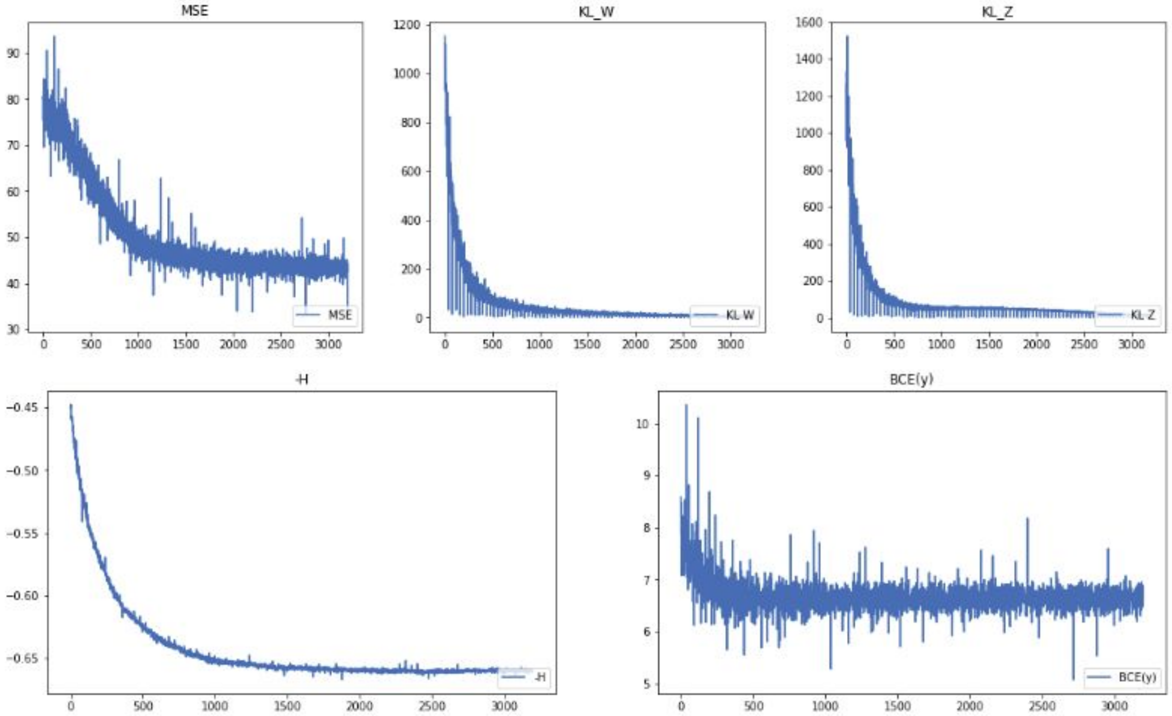


- Additional network predicts label y from latent vector
- Adversarially trained to make latent representation z independent from label

CelebA

- 200,000 images
  of celebrity faces
- 40 labelled attributes

- First trained base VAE
  architecture
- Used same architecture
  and pretrained weights
  for CSVAE and
  All other experiments



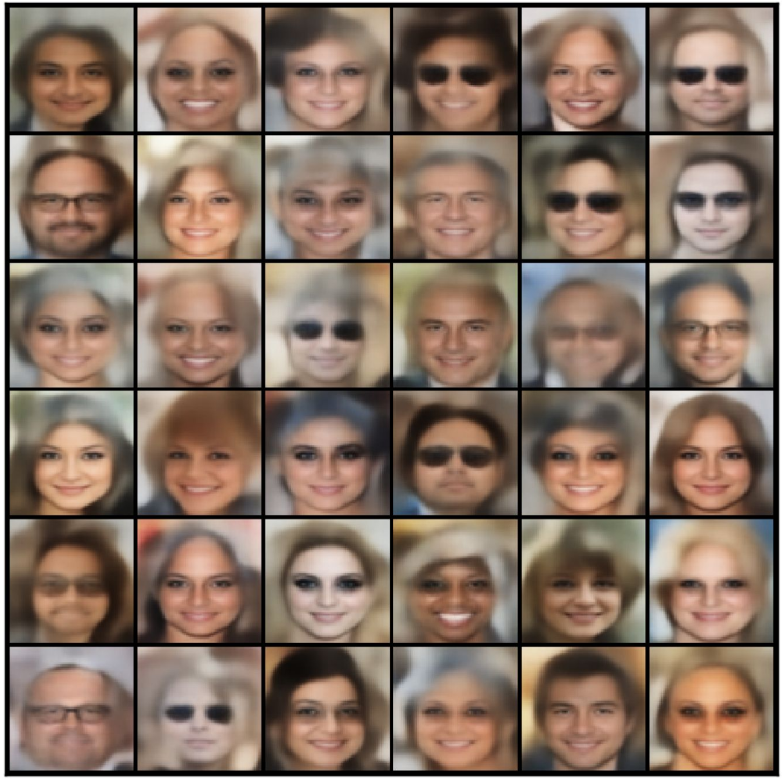
CSVAE loss components

Attributes: “Eyeglasses”

CSVAE vs. CondVAE

Attributes: “Eyeglasses” and “Smiling”

CSVAE vs. CondVAE



Attributes: “Male” without “Heavy Makeup”, "Wearing Lipstick"

CSVAE vs. CondVAE

Attributes: “Heavy Makeup” without 'Male', 'Receding_Hairline', 'Eyeglasses', 'Mustache'

CSVAE vs. CondVAE

Interpolation: No eyeglasses ⇢ Eyeglasses

CondVAE

vs.

CSVAE

Conclusions

CondVAE

- More accurate samples
- Simpler and easier to train model

CSVAE:

- Many networks including competing components
- Many hyperparameters to tune
  => more difficult to train properly on complex tasks (CelebA)
- Richer controllable attributes and structured latent space