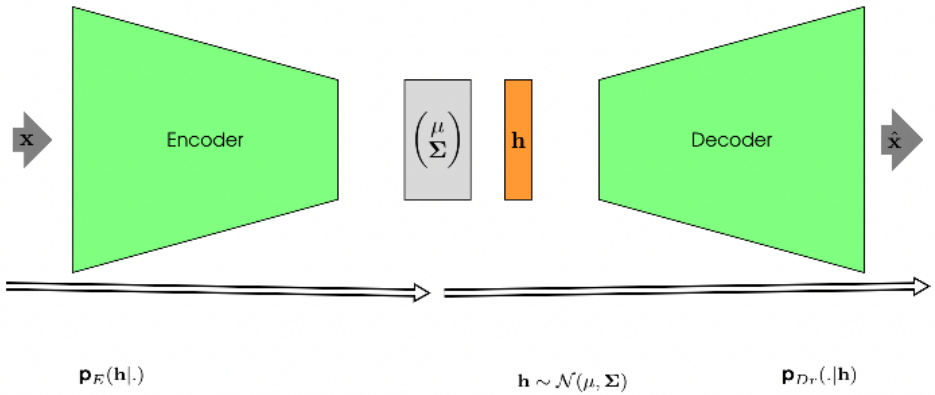


VARIATIONAL AUTOENCODERS

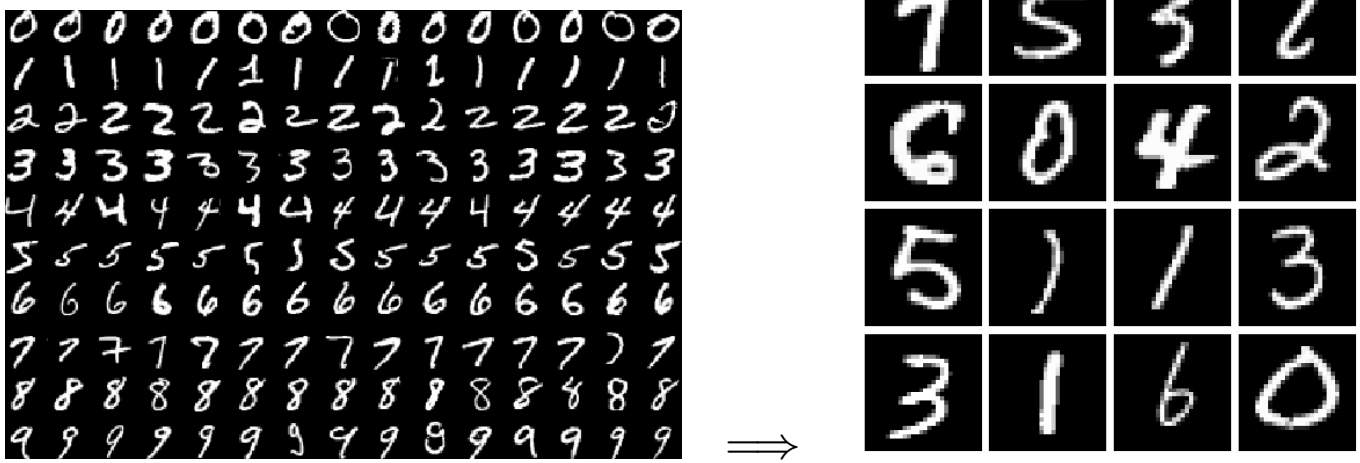
Vincent Barra
LIMOS, UMR 6158 CNRS, Université Clermont Auvergne

GENERATIVE MODELS

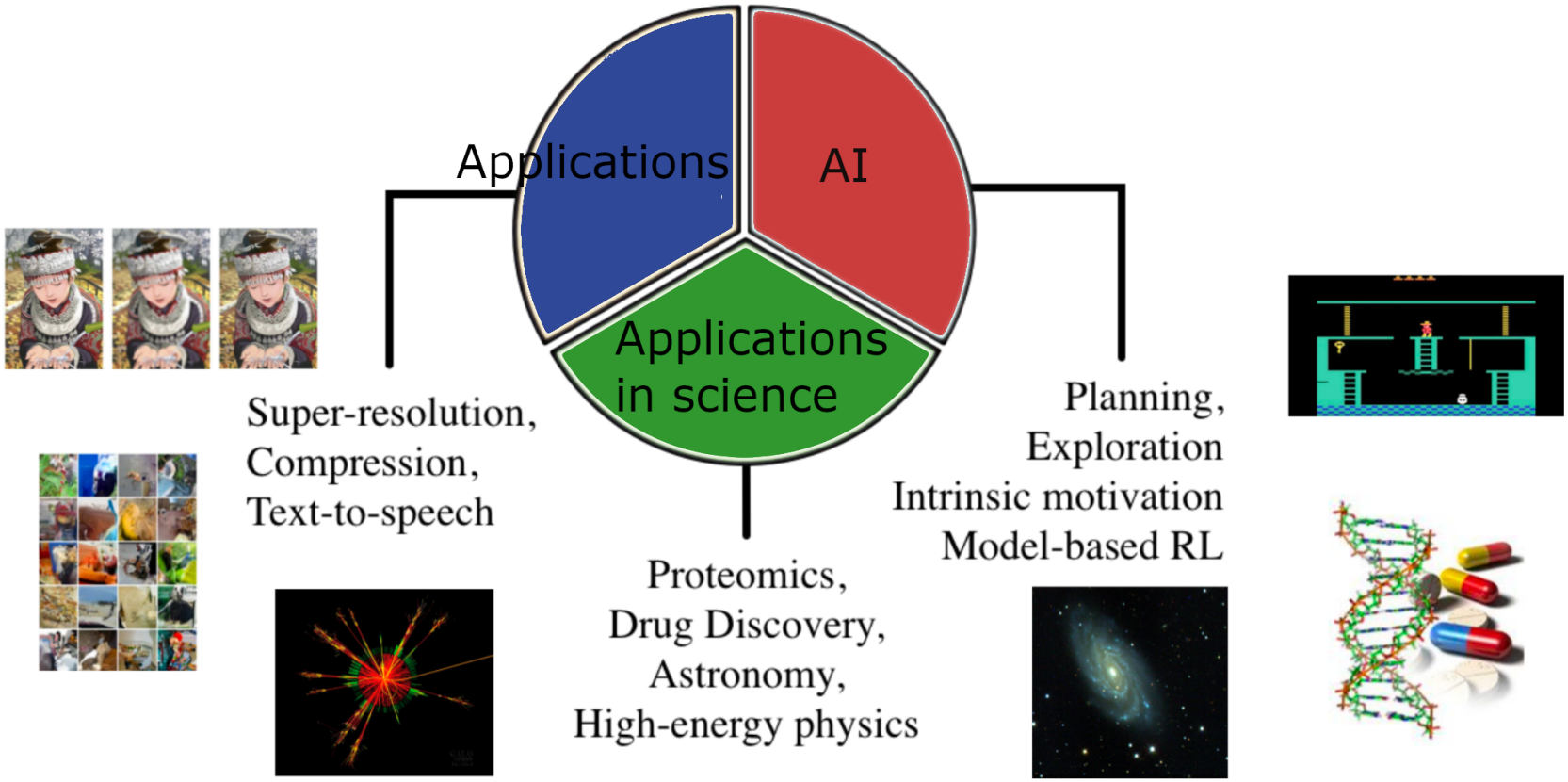
Objective

Learning a model that represents the distribution of data, with input training samples

$$\mathcal{P}_{model}(x) \sim \mathcal{P}_{data}(x)$$

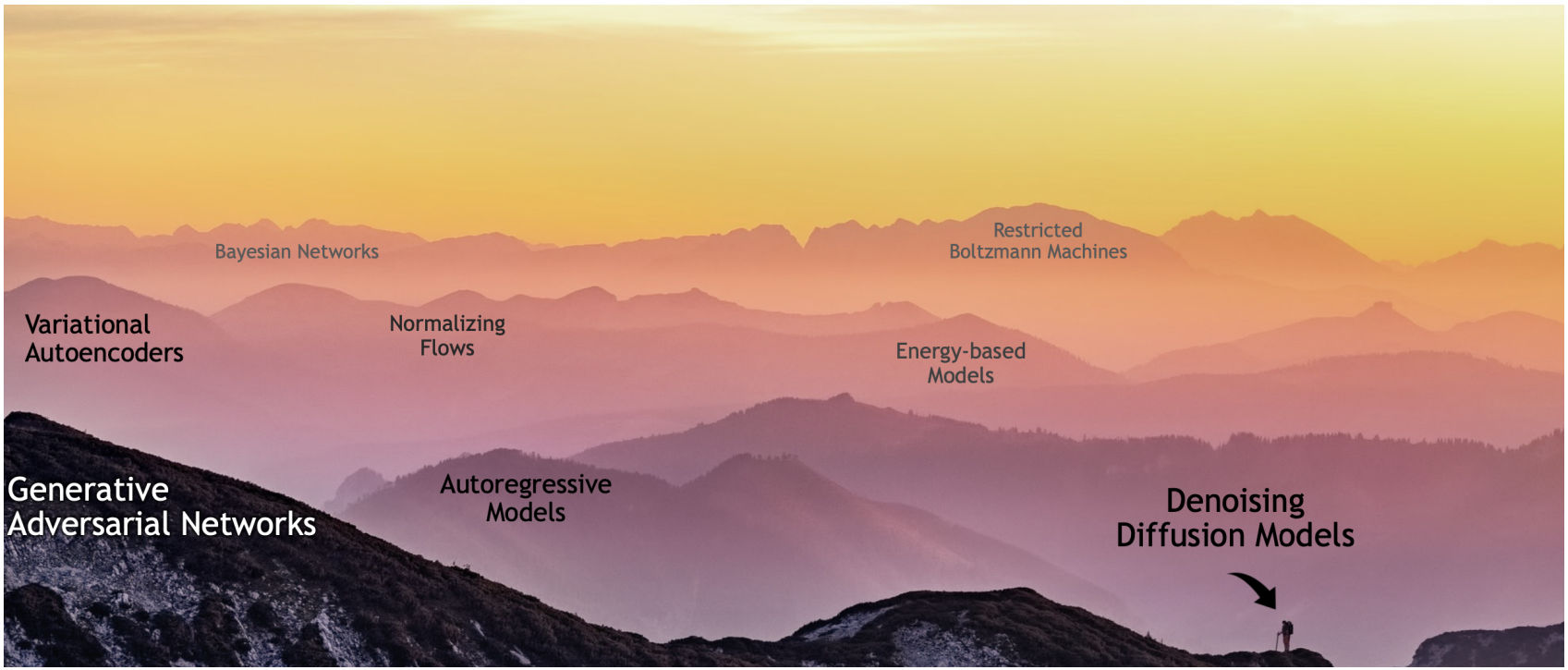


GENERATIVE MODELS



Source: G Louppe

THE LANDSCAPE OF GENERATIVE MODELS



Source: Song et al., CVPR 2023.

Latent variable models We'll see

- Variational Autoencoders
- Generative Adversarial Networks

LIMOS LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES
cnrs
UCA UNIVERSITÉ Clermont Auvergne

LATENT MODELS

A latent variable model relates a set of observable variables $\boldsymbol{x} \in X$ to a set of latent variables $\boldsymbol{h} \in H$

$$\mathbf{p}(\boldsymbol{x}, \boldsymbol{h}) = \mathbf{p}(\boldsymbol{x}|\boldsymbol{h})\mathbf{p}(\boldsymbol{h})$$

if $\boldsymbol{h}$ are causal factors for $\boldsymbol{x} \Rightarrow$ sampling from $\mathbf{p}(\boldsymbol{x}|\boldsymbol{h})$ = generative process from H to X.

Inference

Inference: given $\mathbf{p}(\boldsymbol{x}, \boldsymbol{h})$, compute

$$\mathbf{p}(\boldsymbol{h}|\boldsymbol{x}) = \frac{\mathbf{p}(\boldsymbol{x}|\boldsymbol{h})\mathbf{p}(\boldsymbol{h})}{\mathbf{p}(\boldsymbol{x})}$$

Intractable

LATENT MODELS

Variational Inference

- $\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})$: family of distributions approximating $\mathbf{p}(\boldsymbol{h}|\boldsymbol{x})$
- ϕ is optimized to minimize the "distance" between both distributions.
- Among all similarity measures: Kullback Leibler divergence

$$
\begin{aligned}
KL(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})||\mathbf{p}(\boldsymbol{h}|\boldsymbol{x})) &= \mathbb{E}_{\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})}\left[log\frac{\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})}{\mathbf{p}(\boldsymbol{h}|\boldsymbol{x})}\right] \\
&= \mathbb{E}_{\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})}\left[log(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})) - log(\mathbf{p}(\boldsymbol{x},\boldsymbol{h}))\right] + log(\mathbf{p}(\boldsymbol{x}))
\end{aligned}
$$

Still intractable

LATENT MODELS

But...

$$\begin{aligned}\min_{\phi} KL(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})||\mathbf{p}(\boldsymbol{h}|\boldsymbol{x})) &= \min_{\phi} log(\mathbf{p}(\boldsymbol{x})) - \mathbb{E}_{\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})}\left[log(\mathbf{p}(\boldsymbol{x},\boldsymbol{h})) - log(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x}))\right] \\ &= \max_{\phi} \underbrace{\mathbb{E}_{\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})}\left[log(\mathbf{p}(\boldsymbol{x},\boldsymbol{h})) - log(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x}))\right]}_{ELBO(\boldsymbol{x},\phi)}\end{aligned}$$

LATENT MODELS

But...

$$\begin{aligned}\min_{\phi} KL(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})||\mathbf{p}(\boldsymbol{h}|\boldsymbol{x})) &= \min_{\phi} log(\mathbf{p}(\boldsymbol{x})) - \mathbb{E}_{\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})}\left[log(\mathbf{p}(\boldsymbol{x},\boldsymbol{h})) - log(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x}))\right] \\ &= \max_{\phi} \underbrace{\mathbb{E}_{\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})}\left[log(\mathbf{p}(\boldsymbol{x},\boldsymbol{h})) - log(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x}))\right]}_{ELBO(\boldsymbol{x},\phi)}\end{aligned}$$

ELBO

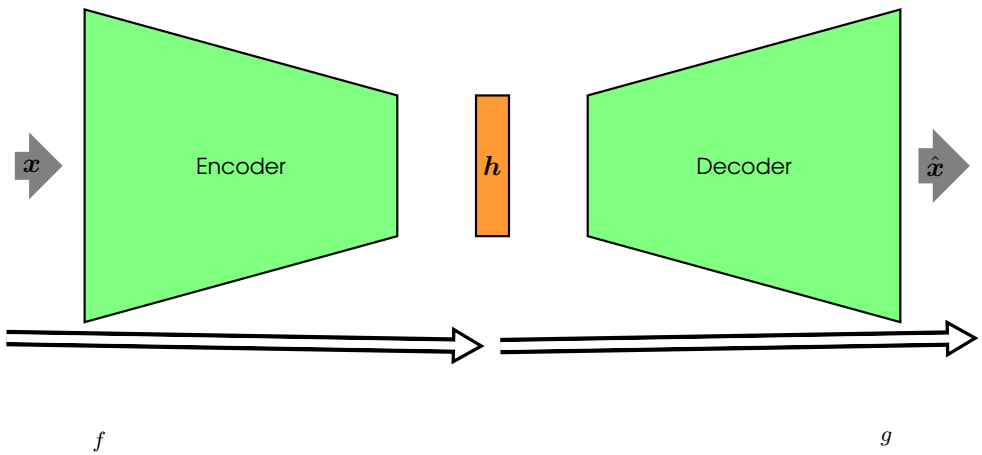
ELBO: Evidence Lower Bound Objective

$$\begin{aligned}ELBO(\boldsymbol{x},\phi) &= \mathbb{E}_{\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})}\left[log(\mathbf{p}(\boldsymbol{x},\boldsymbol{h})) - log(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x}))\right] \\ &= \mathbb{E}_{\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})}\left[log(\mathbf{p}(\boldsymbol{x}|\boldsymbol{h}))\mathbf{p}(\boldsymbol{h}) - log(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x}))\right] \\ &= \mathbb{E}_{\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})}\left[log(\mathbf{p}(\boldsymbol{x}|\boldsymbol{h}))\right] - KL(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})||\mathbf{p}(\boldsymbol{h}))\end{aligned}$$

Maximizing $ELBO(\boldsymbol{x},\phi)$:

- The first term encourages distributions to be centered on configurations of latent variables $\boldsymbol{h}$ explaining the observed data
- The second term enforces distributions to be close to the prior.

AUTOENCODERS

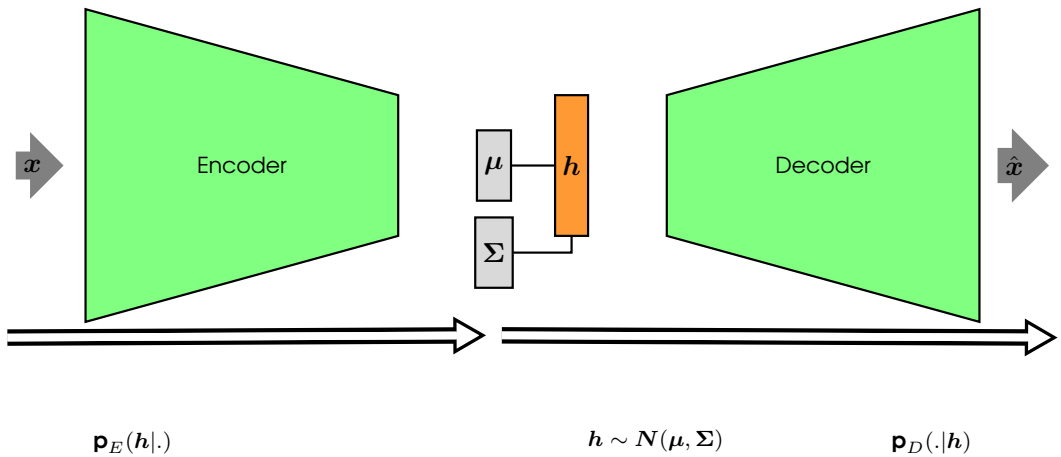


Key ideas

- A neural network trained using unsupervised learning
- Trained to copy its input to its output
- Learns an embedding h

$$\hat{\boldsymbol{x}} = g[f(\boldsymbol{x})] \quad h = f(\boldsymbol{x})$$

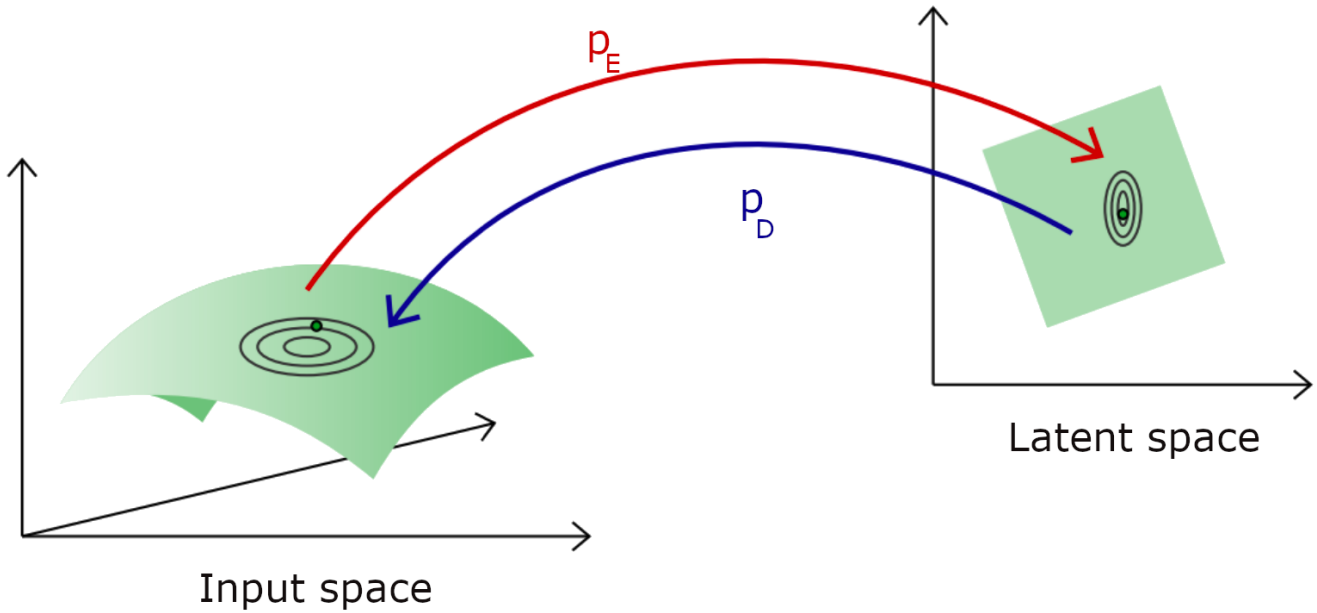
VARIATIONAL AUTOENCODERS



Probabilistic twist on AE

- $f \Longrightarrow \mathbf{p}_E(\boldsymbol{h}|\boldsymbol{x})$
- $g \Longrightarrow \mathbf{p}_D(\boldsymbol{x}|\boldsymbol{h})$
- Sample from μ and Σ to compute latent samples

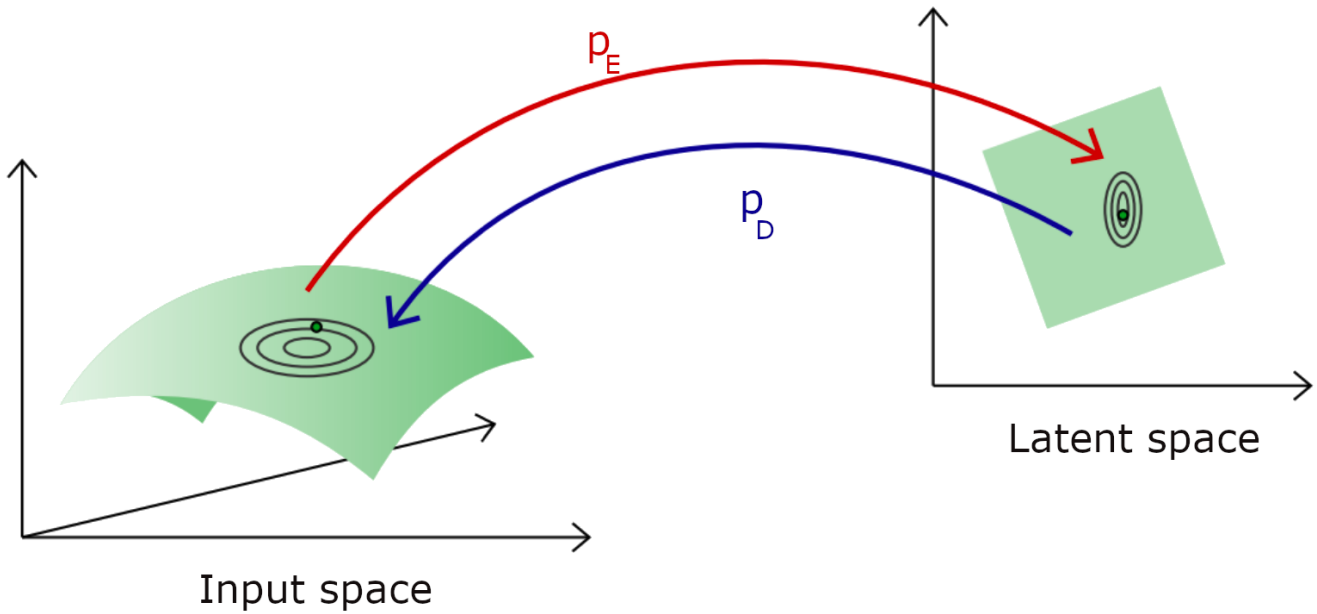
VARIATIONAL AUTOENCODERS



Probabilistic twist on AE

- $f \Longrightarrow \mathbf{p}_E(\boldsymbol{h}|\boldsymbol{x})$
- $g \Longrightarrow \mathbf{p}_D(\boldsymbol{x}|\boldsymbol{h})$
- Sample from μ and Σ to compute latent samples

VARIATIONAL AUTOENCODERS



Probabilistic twist on AE

- $\mathbf{p}_E(\boldsymbol{h}|\boldsymbol{x}) = \mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})$
- $\mathbf{p}_D(\boldsymbol{x}|\boldsymbol{h}) \Longrightarrow \mathbf{p}_\theta(\boldsymbol{x}|\boldsymbol{h})$
- $\mathbf{p}(\boldsymbol{h})$: Fixed prior on the latent distribution

Loss:

$$\min_{\phi,\theta}(-ELBO(\boldsymbol{x},\phi)) = -\mathbb{E}_{\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})}\left[log(\mathbf{p}_\theta(\boldsymbol{x}|\boldsymbol{h}))\right] + KL(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})||\mathbf{p}(\boldsymbol{h}))$$

ELBO

Loss

$$\min_{\phi,\theta}(-ELBO(\boldsymbol{x},\phi)) = -\mathbb{E}_{\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})}\left[log(\mathbf{p}_\theta(\boldsymbol{x}|\boldsymbol{h}))\right] + KL(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})||\mathbf{p}(\boldsymbol{h}))$$

- Given θ, optimize ϕ so that latent variable distribution explains the observed data, while remaining close to the prior
- Given ϕ, optimize θ so that observed data is well explained by the latent variables.

VARIATIONAL AUTOENCODERS

Encoder/decoder

Both encoder and decoder map to a Gaussian with diagonal covariance.

- $\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}^f(\boldsymbol{X}), diag(\boldsymbol{\sigma}^f(\boldsymbol{X}))$: Inference network (encoder)
- $\mathbf{p}_\theta(\boldsymbol{x}|\boldsymbol{h}) = \mathcal{N}(\boldsymbol{\mu}^g(\boldsymbol{h}), diag(\boldsymbol{\sigma}^g(\boldsymbol{h}))$: Generative network (decoder)

Variational Autoencoders

Encoder/decoder

Both encoder and decoder map to a Gaussian with diagonal covariance.

- $\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}^f(\boldsymbol{X}), diag(\boldsymbol{\sigma}^f(\boldsymbol{X}))$: Inference network (encoder)
- $\mathbf{p}_\theta(\boldsymbol{x}|\boldsymbol{h}) = \mathcal{N}(\boldsymbol{\mu}^g(\boldsymbol{h}), diag(\boldsymbol{\sigma}^g(\boldsymbol{h}))$: Generative network (decoder)

Prior $\mathbf{p}(\boldsymbol{h})$

- Impose a distribution for the random latent variable.
- Train the decoder so that its output matches the training data.
- Common choice: $\boldsymbol{h} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}^2)$

VARIATIONAL AUTOENCODERS

Let $\boldsymbol{x}_1 \cdots \boldsymbol{x}_n$ be the training inputs.

$$KL(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})||\mathbf{p}(\boldsymbol{h})) = \frac{1}{n}\left[-\frac{1}{2}\sum_{i=1}^{d}\left(1 + log\sigma_i^f(\boldsymbol{x}) - (\mu_i^f(\boldsymbol{x}))^2 - \sigma_i^f(x)\right)\right]$$

VARIATIONAL AUTOENCODERS

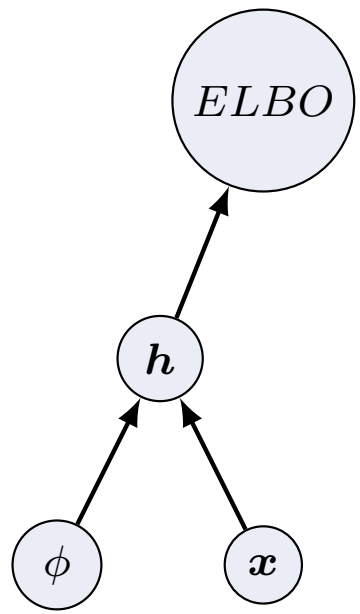
Let $\boldsymbol{x}_1 \cdots \boldsymbol{x}_n$ be the training inputs.

$$KL(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})||\mathbf{p}(\boldsymbol{h})) = \frac{1}{n}\left[-\frac{1}{2}\sum_{i=1}^{d}\left(1 + log\sigma_i^f(\boldsymbol{x}) - (\mu_i^f(\boldsymbol{x}))^2 - \sigma_i^f(x)\right)\right]$$

$$\min_{\phi,\theta}(-ELBO(\boldsymbol{x},\phi)) = -\mathbb{E}_{\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})}\left[log(\mathbf{p}_\theta(\boldsymbol{x}|\boldsymbol{h}))\right] + KL(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})||\mathbf{p}(\boldsymbol{h}))$$

Optimization

- Gradients w.r.t $\theta \Longrightarrow$ easy to obtain
- Gradients w.r.t $\phi \Longrightarrow$ more difficult.



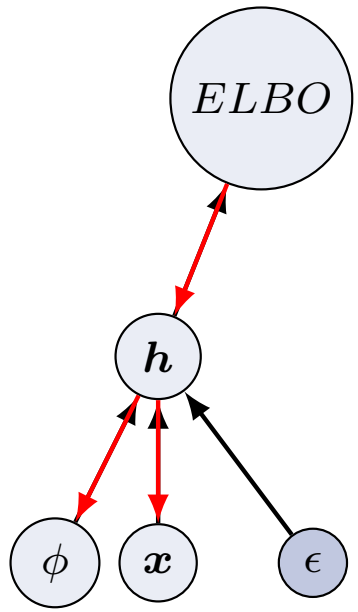
Cannot backpropagate through $\boldsymbol{h}$ to compute $\nabla_\phi(ELBO)$

REPARAMETERIZATION TRICK

Trick

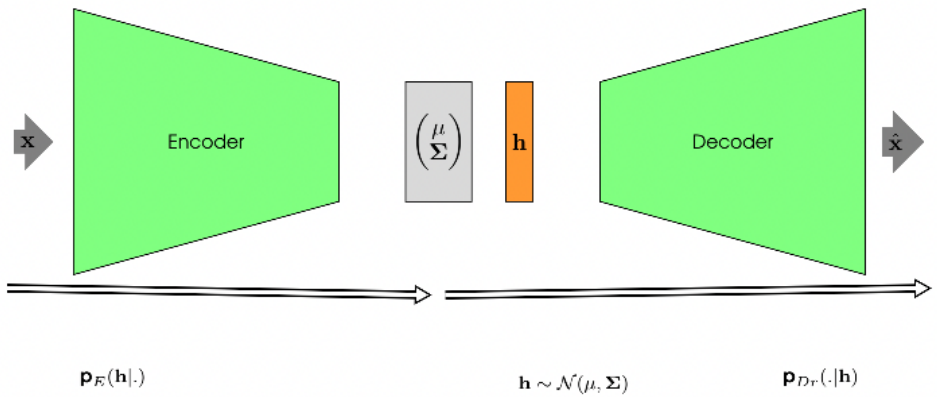
Expressing $\boldsymbol{h}$ as some differentiable and invertible transformation of another random variable ϵ given $\boldsymbol{x}$ and ϕ.

$$\boldsymbol{h} = \mu(\boldsymbol{x}, \phi) + \sigma(\boldsymbol{x}, \phi) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$



Gradient can flow out of any random variable $\Rightarrow$ backpropagation is possible.

SUMMARY

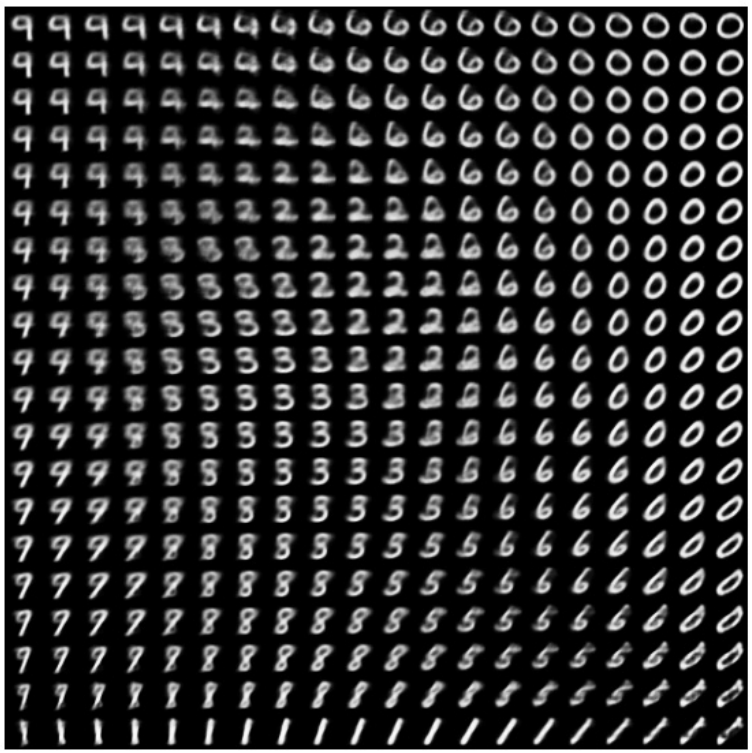


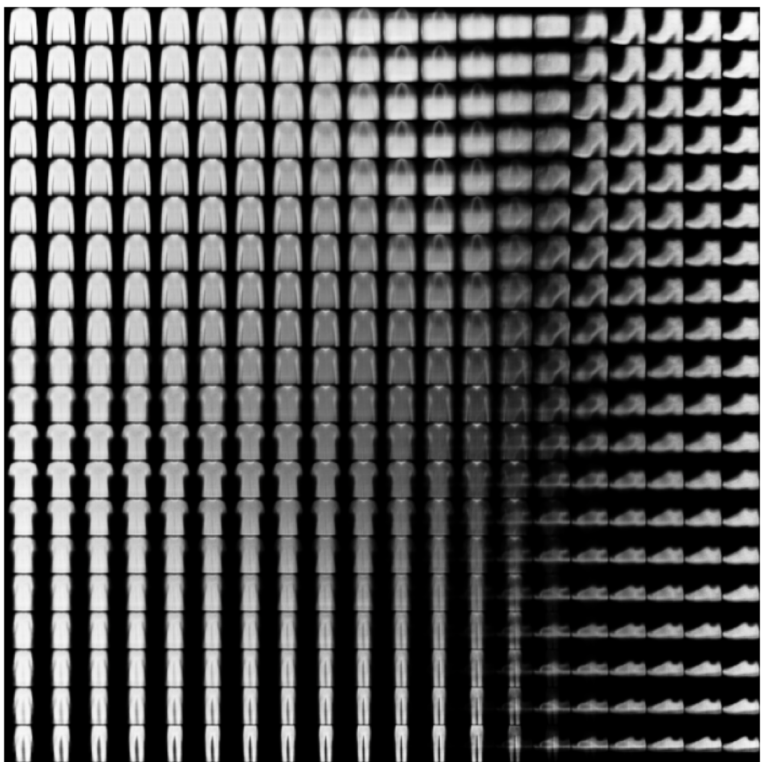
VAE

1. Define Encoder and Decoder
2. Define the latent space distribution using the reparametrization trick
3. Define the ELBO loss
4. Train and play with VAE !

EXPLORATION

- Changing one single variable h_i and keep all other h_j fixed.
- Dimensions of $\boldsymbol{h}$ encode different interpretable latent features.





DISENTANGLEMENT

Kingma et al, 2014

DISENTANGLEMENT

VS

Homogeneous skin color, pose

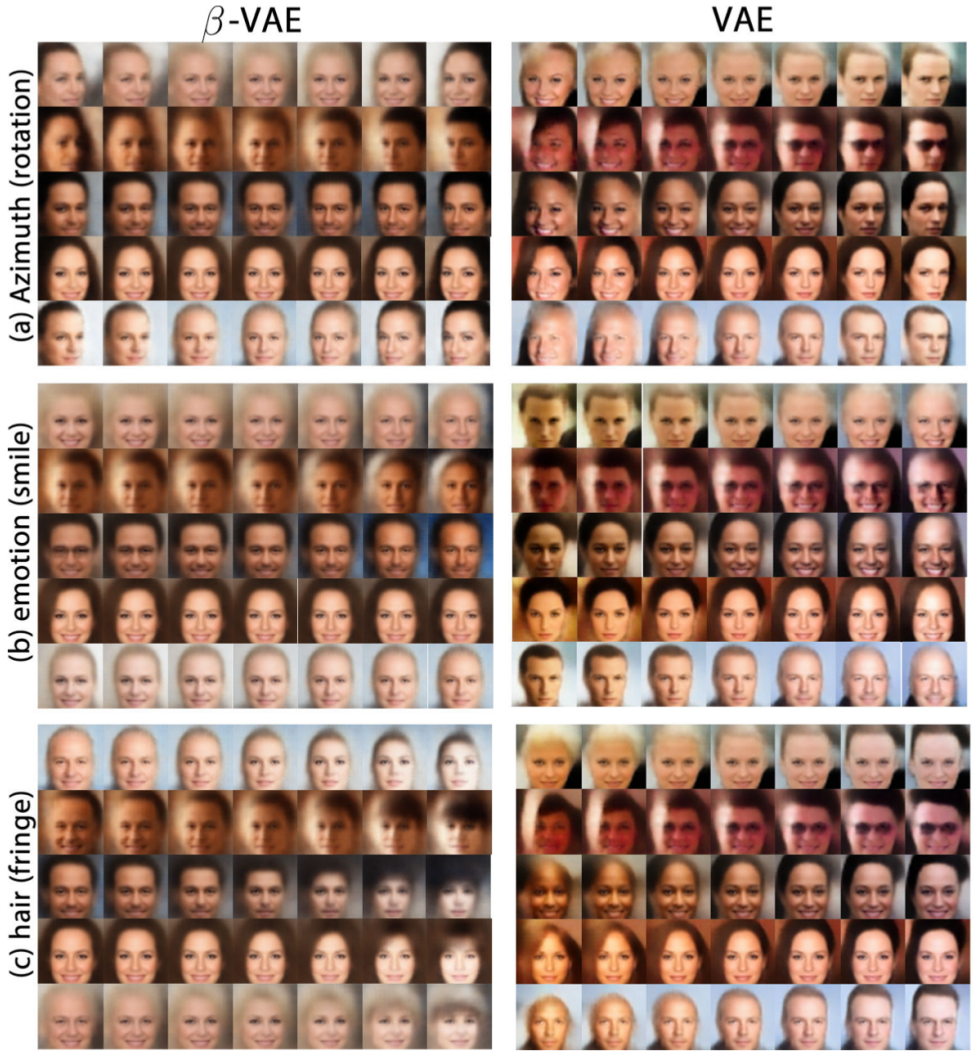
Diverse skin color, pose, illumination

Amini and Soleimany, 2019

DISENTANGLEMENT

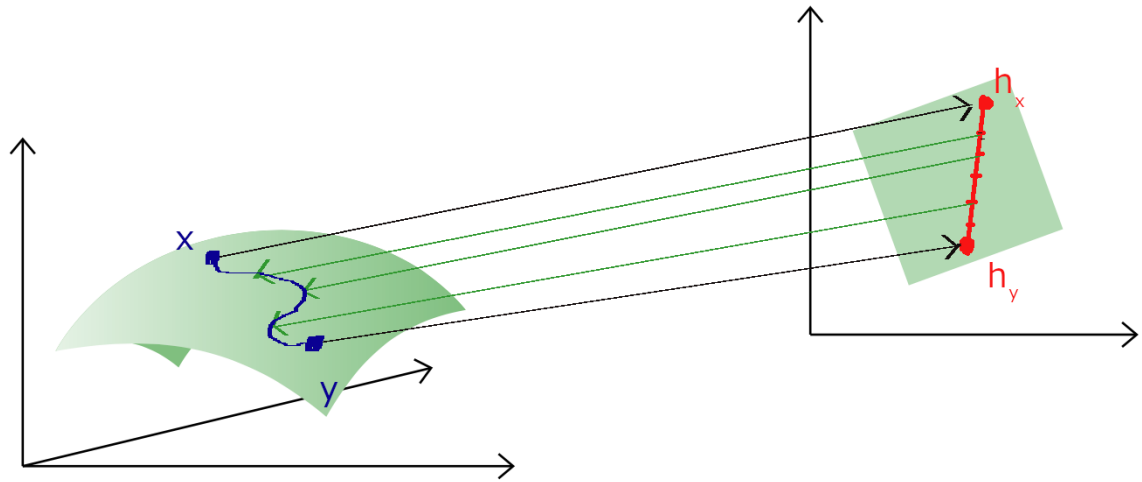
$\beta - VAE$: enforces disentanglement

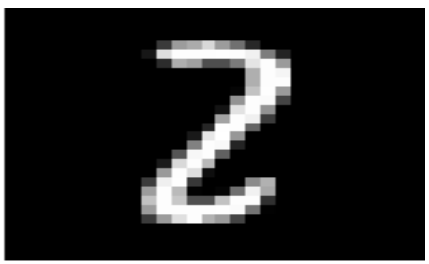
$$\min_{\phi,\theta}(-ELBO(\boldsymbol{x},\phi)) = -\mathbb{E}_{\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})}\left[log(\mathbf{p}_\theta(\boldsymbol{x}|\boldsymbol{h}))\right] + \beta KL(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})||\mathbf{p}(\boldsymbol{h}))$$



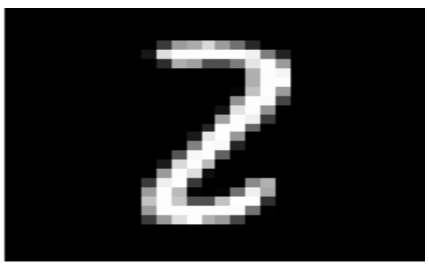
Higgins et al., 2017

EXPLORATION





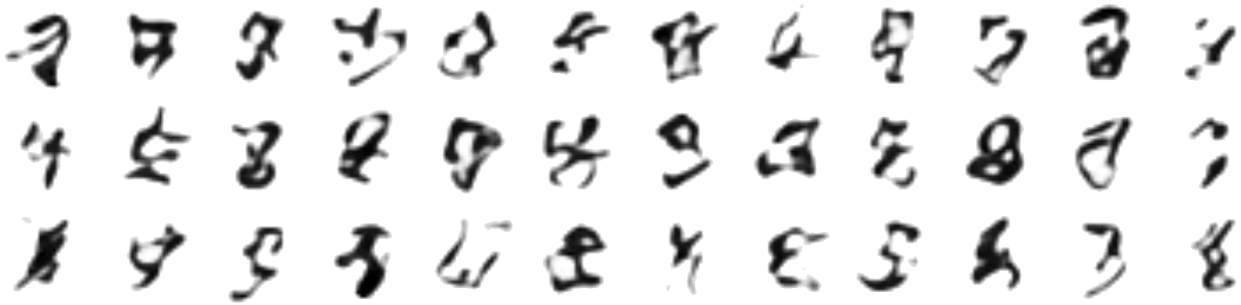
Interpolation in the image space



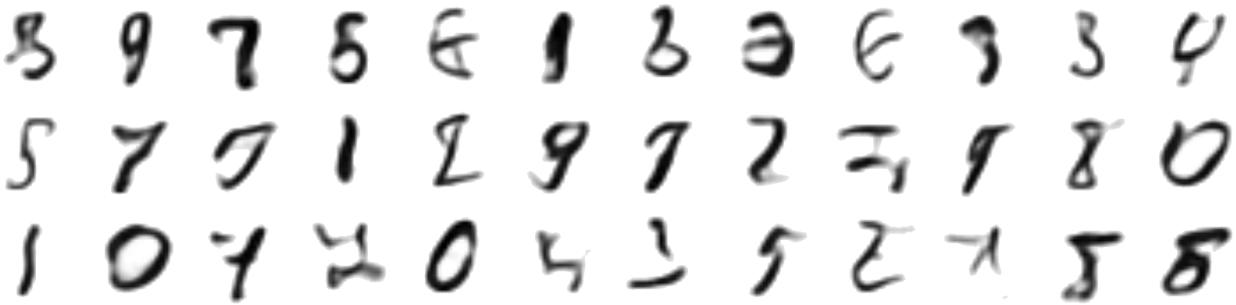
Interpolation in the latent space

Sampling

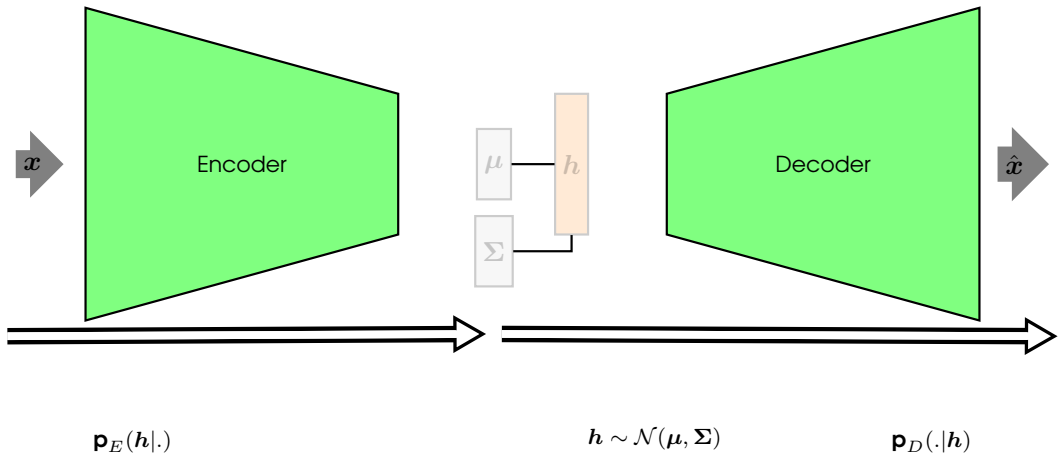
AE (d=32)



VAE (d=32)



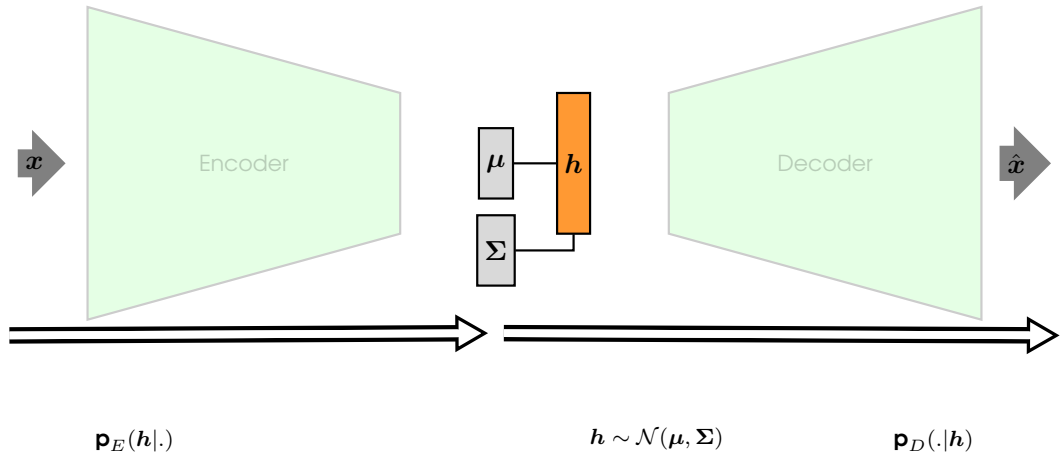
ENCODER-DECODER



Encoder-Decoder

- MLP
- CNN
- RNN

REPARAMETERIZATION TRICK



$$h = \mu(x, \phi) + \sigma(x, \phi) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

```
z_mean, z_log_var = inputs
batch_size = K.shape(z_mean)[0]
epsilon = K.random_normal(shape=(batch_size, latent_dim),mean=0., stddev=1.)
```

KL DIVERGENCE

$$KL(\mathbf{p}_\phi(\boldsymbol{h}|\boldsymbol{x})||\mathbf{p}(\boldsymbol{h})) = \frac{1}{n}\left[-\frac{1}{2}\sum_{i=1}^{d}\left(1 + log\sigma_i^f(\boldsymbol{x}) - (\mu_i^f(\boldsymbol{x}))^2 - \sigma_i^f(x)\right)\right]$$

```
z_mean, z_log_var = encoder(x) of
kl_loss = - 0.5 * K.sum(1 + z_log_var - K.square(z_mean) - K.exp(z_log_var), axis=-1)
```

AND NOW GATHER ALL THE STUFF

```python
def VAE(input_shape, encoder, decoder, sampling_layer,beta):

    x = Input(shape=input_shape, name="input")
    z_mean, z_log_var = encoder(x)
    z = sampling_layer([z_mean, z_log_var])
    x_decoded_mean = decoder(z)
    vae = Model(x, x_decoded_mean)

    # Loss function : Evidence Lower Bound (ELBO)
    xent_loss = dim * metrics.binary_crossentropy(K.flatten(x), K.flatten(x_decoded_mean))
    kl_loss = - 0.5 * K.sum(1 + z_log_var - K.square(z_mean) - K.exp(z_log_var), axis=-1)
    vae_loss = K.mean(xent_loss + beta*kl_loss)

    vae.add_loss(vae_loss)
    vae.compile(optimizer='adam')
    return vae
```