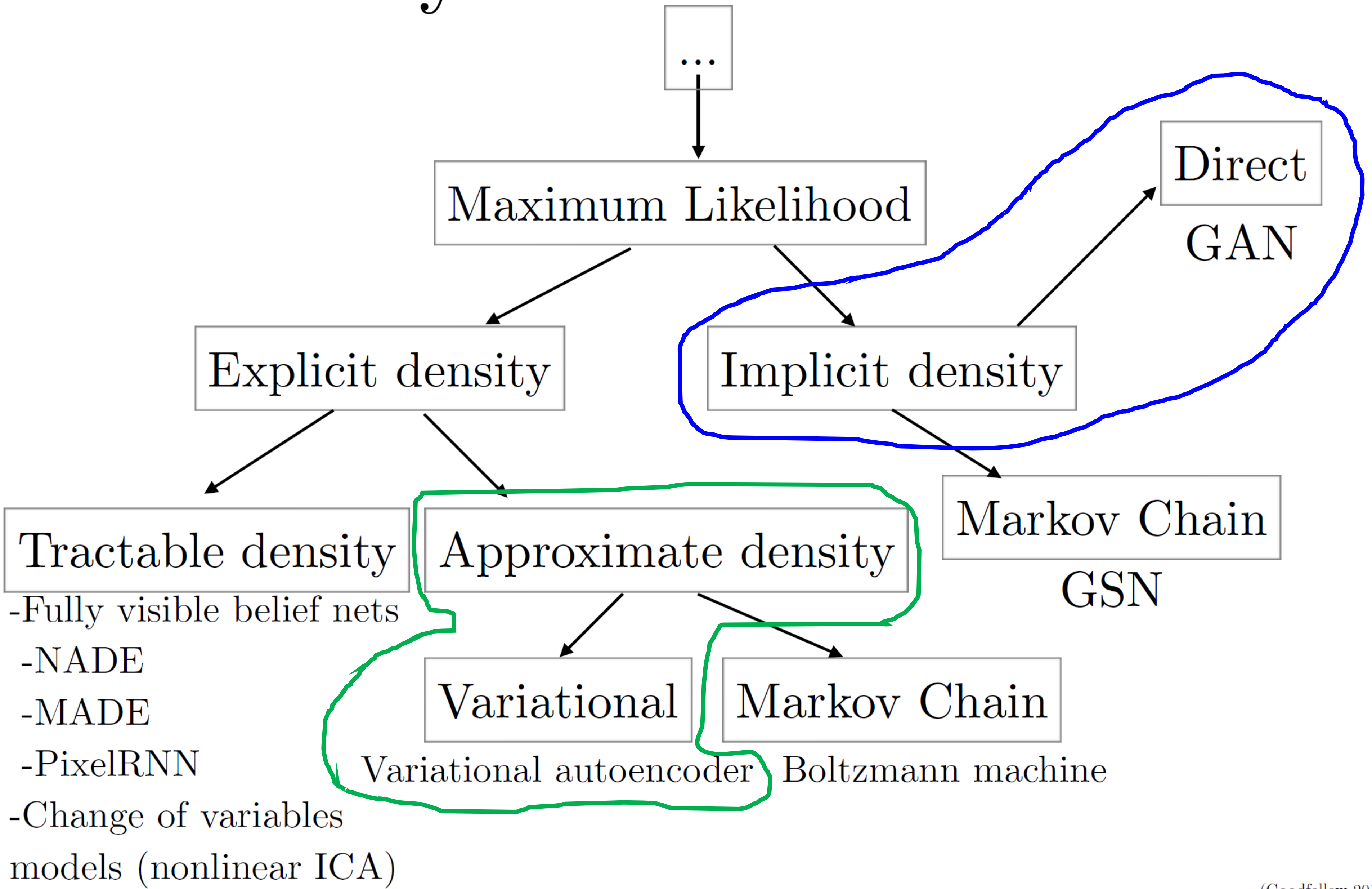
Variational Autoencoder

Slides based on :Deep Learning, MIT Press Book of Ian Goodfellow et al.
Lecture Slides of S. Lazebnik , Illinois University

Generative Models

- **Generative Models**
- **Task:** Given a dataset of images {X1,X2...} can we learn the distribution of X?
- Generative models are typically meant for modelling P(X).
- Often applications seek for models which we can **sample** from.
    - Models that can generate random examples that follow the distribution of P(X).

Taxonomy of Generative Models

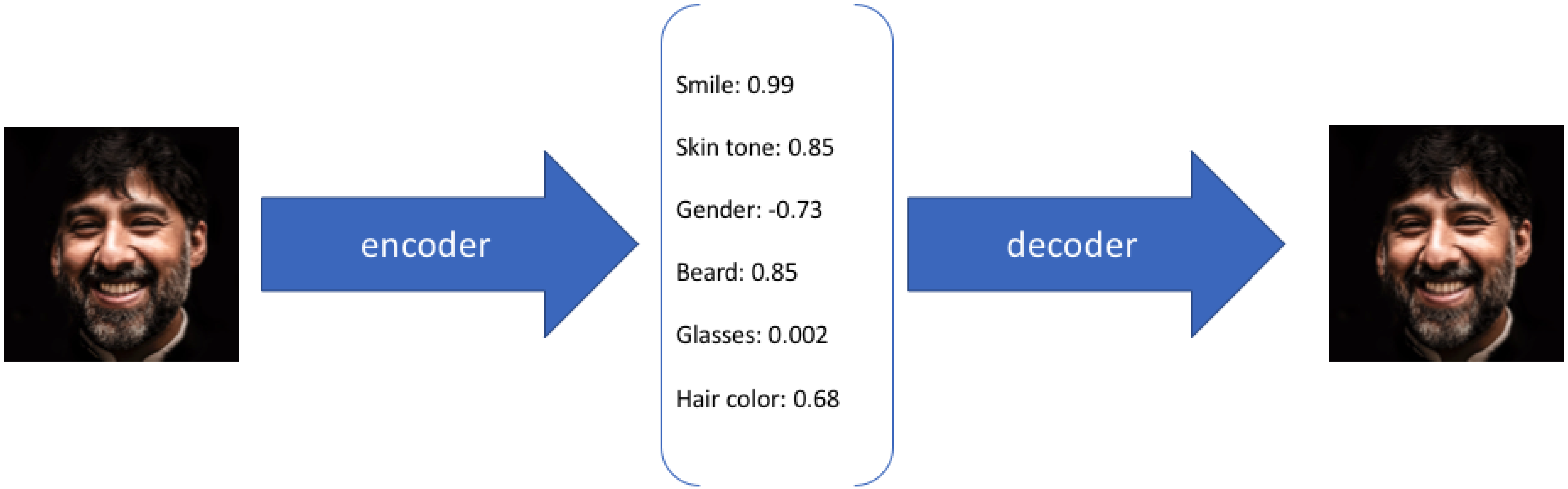


Generative Models: GAN

- Remember that explicit form of $P(X|z)$ is not needed with input z in the latent space (features).

- But it is difficult to map back to the latent variable given a desired form of an output (an image).

Reconstruction from latent space variable is difficult



Latent attributes

Source: https://www.jeremyjordan.me/variational-autoencoders/

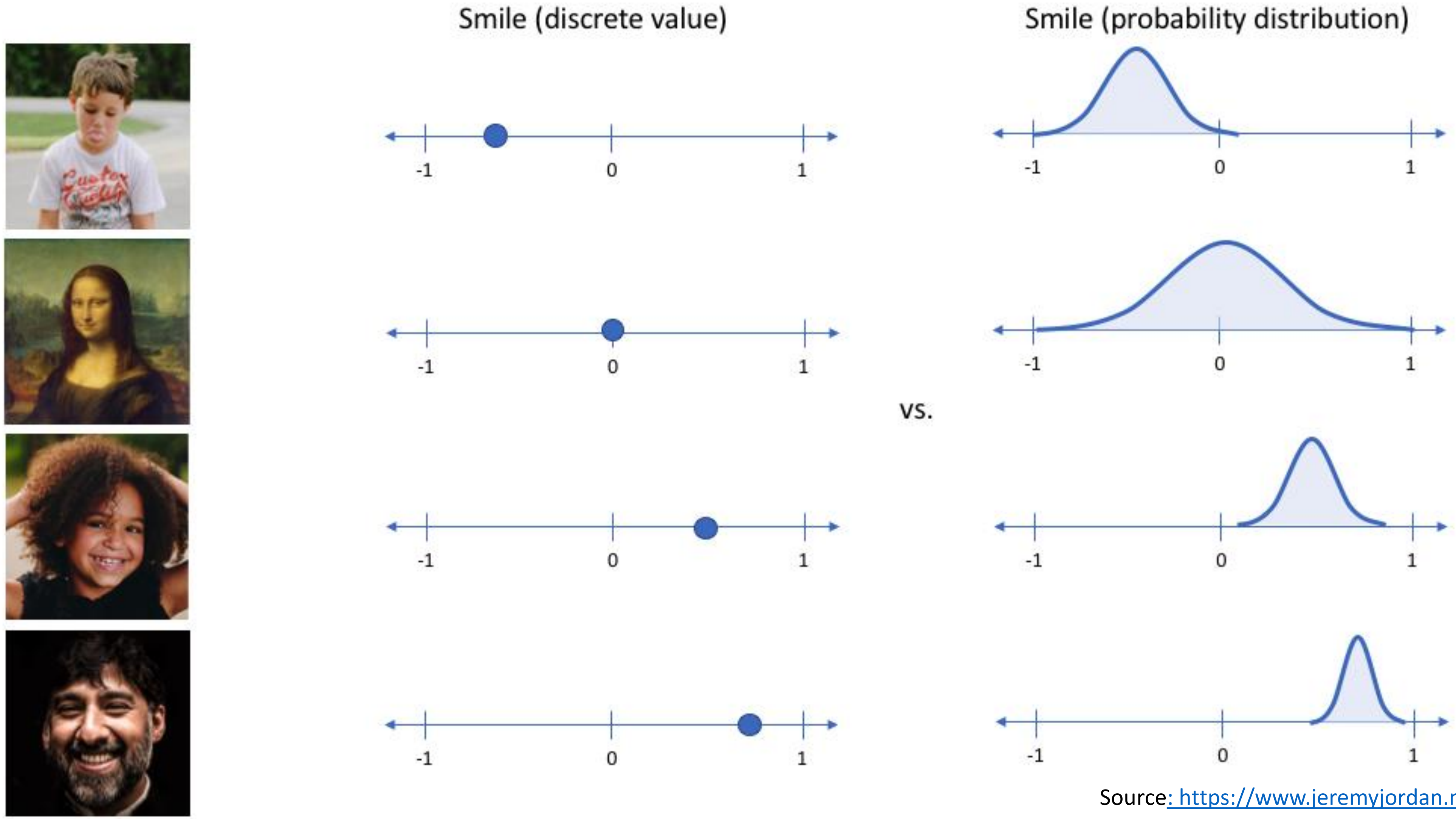
Variational Autoencoders

- Prior to GAN, variational autoencoders (VAEs) were meant for explicit Modelling of *P(X|z)* with the given model parameters.

- z ~ P(z), which can be sampled from Gaussian distribution.

$$P(X) = \int P(X|z;\theta)P(z)dz$$

Can we sample from a Gaussian Distribution to reconstruct the input ?



Source: https://www.jeremyjordan.me/variational-autoencoders/

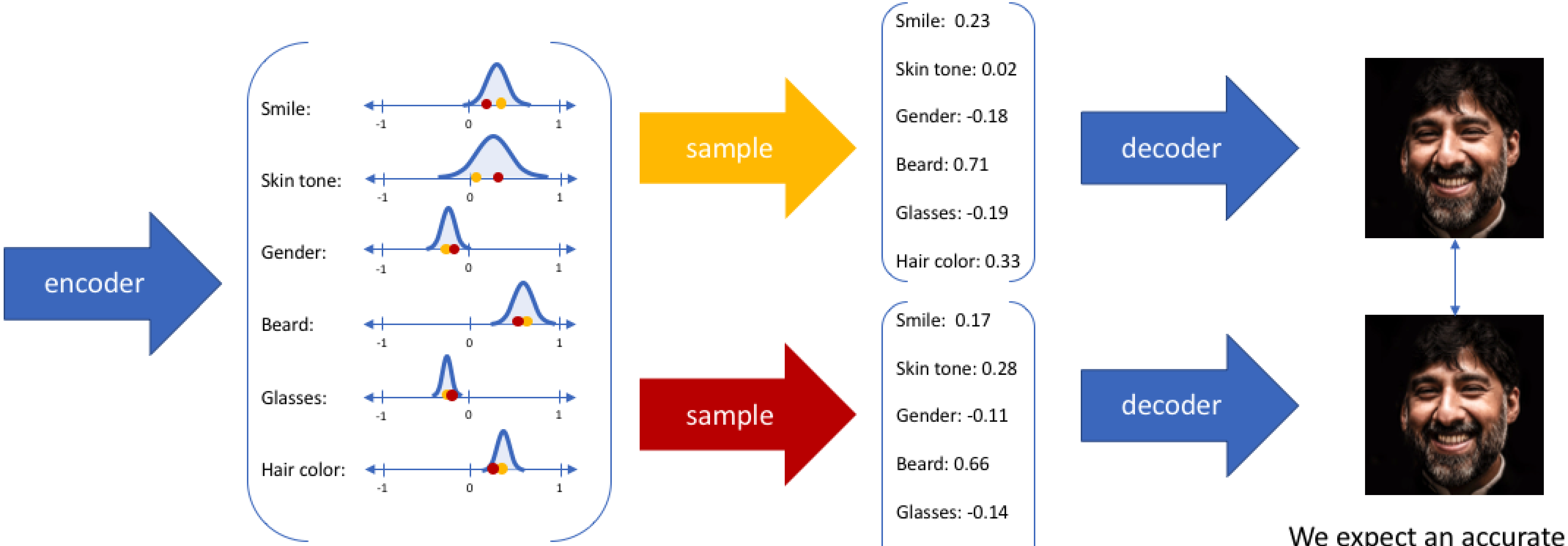
Variational Autoencoders

- Maximum Likelihood --- Find the model parameters to maximize P(X), where X is the data.
- Approximate with samples of z

$$P(X) \approx \frac{1}{n} \sum_{i=0}^{n} P(X|z_i)$$

- So one requires large number of samples of z.
- For most of them $P(X|z) \approx 0$.
- Not practical computationally.

Approximate with samples of z:
We want Accurate Reconstruction



Latent distributions

Sampled latent attributes

We expect an accurate reconstruction for any sample from the latent state distributions

Source: https://www.jeremyjordan.me/variational-autoencoders/

Variational Autoencoders

- Is it possible to know which z will generate $P(X|z) >> 0$?

- It amounts to learning a distribution $Q(z)$, where $z \sim Q(z)$ generates $P(X|z) >> 0$.

Variational Autoencoders

- Suppose we can learn a distribution Q(z), where z ~ Q(z) generates P(X|z) >> 0.

- We want to learn P(X) such that $P(X) = E_{z \sim P(z)} P(X|z)$
  - not so practical. Why ?

- We will compute $E_{z \sim Q(z)} P(X|z)$ which is a more practical approach.

Variational Autoencoders

- How can we relate $E_{z\sim Q(z)}P(X|z)$ with $P(X)$?

- To know the relation, we first define Kullback–Leibler (KL) Divergence also known as relative entropy (measure).

$$D(Q(z)\|P(z|X)) = E_{z\sim Q(z)}[\log Q(z) - \log P(z|X)]$$

Relation Between $E_{z \sim Q(z)} P(X|z)$ and $P(X)$

- By Bayes rule

$$P(z|X) = \frac{P(X|z)P(z)}{P(X)}$$

- Apply logarithm on both sides

$$\log P(z|X) = \log P(X|z) + \log P(z) - \log P(X)$$

Relation Between $E_{z \sim Q(z)} P(X|z)$ and $P(X)$

- By definition of KL divergence

$$D(Q(z) \| P(z|X)) = E_{z \sim Q(z)}[\log Q(z) - \log P(z|X)]$$

- Substitute for $\log P(z|X)$ in the above expression.

$$\log P(z|X) = \log P(X|z) + \log P(z) - \log P(X)$$

Relation Between $E_{z\sim Q(z)}P(X|z)$ and $P(X)$

- KL divergence

$$D(Q(z)\|P(z|X)) = E_{z\sim Q(z)}[\log Q(z) - \log P(X|z) - \log P(z) + \log P(X)]$$

- By the properties of expectation function

$$E_{z\sim Q(z)}[Y + constant] = E_{z\sim Q(z)}[Y] + constant$$

- Therefore

$$D(Q(z)\|P(z|X)) = E_{z\sim Q(z)}[\log Q(z) - \log P(X|z) - \log P(z)] + \log P(X)$$

Relation Between $E_{z\sim Q(z)}P(X|z)$ and $P(X)$

$$D(Q(z)\,||\,P(z|X)) = E_{z\sim Q(z)}[\log Q(z) - \log P(X|z) - \log P(z)] + \log P(X)$$

- Rearrange terms

$$log\,P(X) - D(Q(z)||P(z|X)) = E_{z\sim Q(z)}[\log P(X|z)] - E_{z\sim Q(z)}[\log Q(z) - \log P(z)]$$

$$log\,P(X) - D(Q(z)\,||\,P(z|X)) = E_{z\sim Q(z)}[\log P(X|z)] - D(Q(z)||P(z))$$

Lower Bound on Log Probability

- Recall we want to maximize P(X) with respect to model parameters.
- But we can not maximize P(X) as we have no control \mechanism to maximize this probability from the latent variables.
- Now we have

$$log\, P(X) - D(Q(z) \| P(z|X)) = E_{z\sim Q(z)}[\log P(X|z)] - D(Q(z)||P(z))$$

- KL divergence D in the above expression is always > 0. This means

$$\log P(X) > \log P(X) - D[Q(z) \,||\, P(z|X)].$$

- So we maximize the lower bound on $\log P(X)$.

Lower Bound on Log Probability

- Hence the problem boils down to maximizing the following expression (lower bound on $\log P(X)$)

$$E_{z\sim Q(z)}[\log P(X|z)] \quad - D(Q(z)||P(z))$$

Remember the assumption that

"we can learn a distribution Q(z), where z ~ Q(z) generates P(X|z) >> 0."

How do we get Q(z) ?

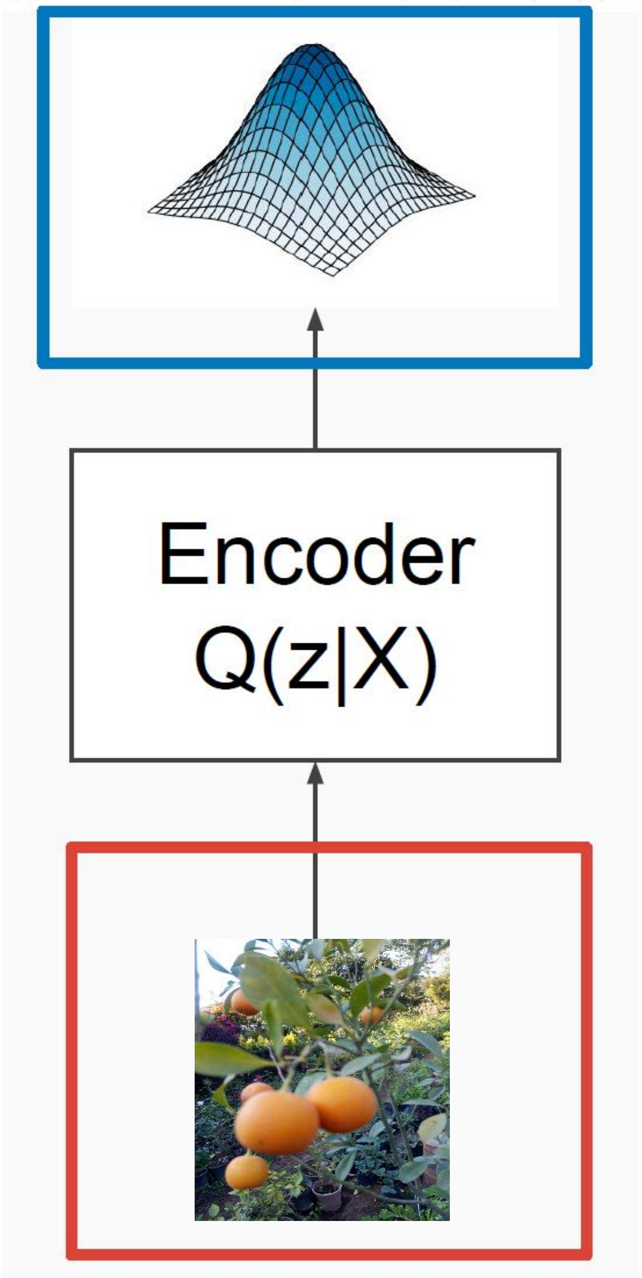
How to get Q(z) ?

- Model Q(z|X) with a neural network.
- Assume Q(z|X) to be a Gaussian

$$N(\mu, c \cdot I)$$

- Neural network **outputs** the **mean μ**, and a diagonal covariance matrix **c** · **I**.
- **Input:** Image
- **Output:** Distribution: Two vectors μ and c.

- Call Q(z|X) the '**Encoder**'.



Variational Autoencoder – Loss Function

$$log\, P(X) - D(Q(z)\,\|P(z|X)) = E_{z \sim Q(z)}[\log P(X|z)] \quad - D(Q(z)||P(z))$$

- Let us convert the lower bound to a loss function:

- Model P(X|z) with a neural network.
- Let f(z) be the network output.
- Assume X to be independent identically distributed random variable in a Gaussian distribution.
  - X = f(z) + η , where η ~ N(0,I) .
- Then the problem is reduced to minimizing the error of regression

$$\|X - f(z)\|^2$$

- Call P(X|z) the Decoder.

Variational Autoencoder – Loss Function …

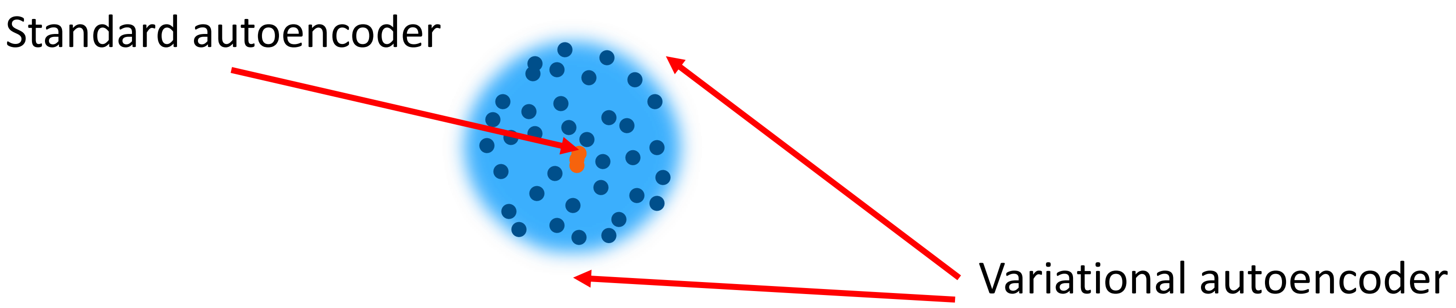
- If $P(z) \sim N(0,1)$ then $D[Q(z|X) \,||\, P(z)]$ has a closed form solution.

- So the loss function can be viewed as

$$L = \|X - f(z)\|^2 - \alpha\, D[Q(z|X) \,||\, P(z)]$$

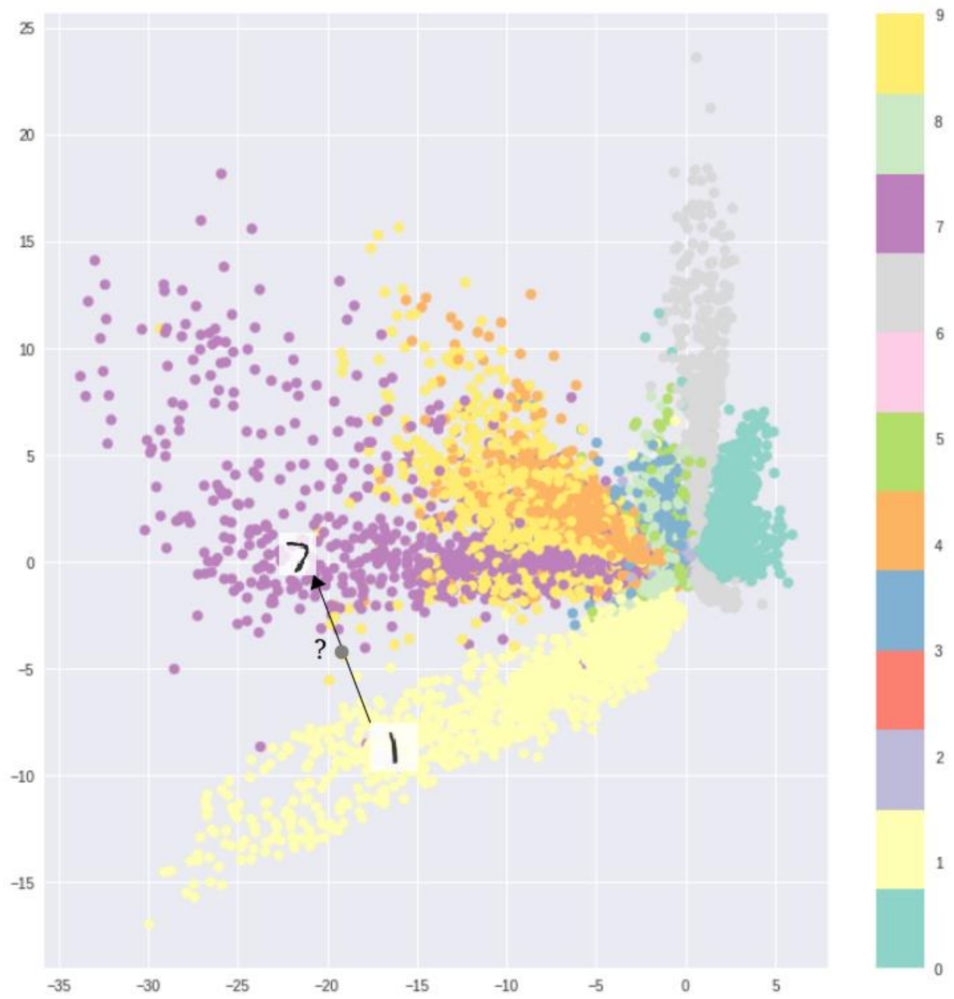
Training

- Note that the model is stochastically generating output which means that even for the same input, mean and standard deviations , the actual encoding may vary slightly on every single pass.
- This is due to random sampling from the encoder's output.



Training

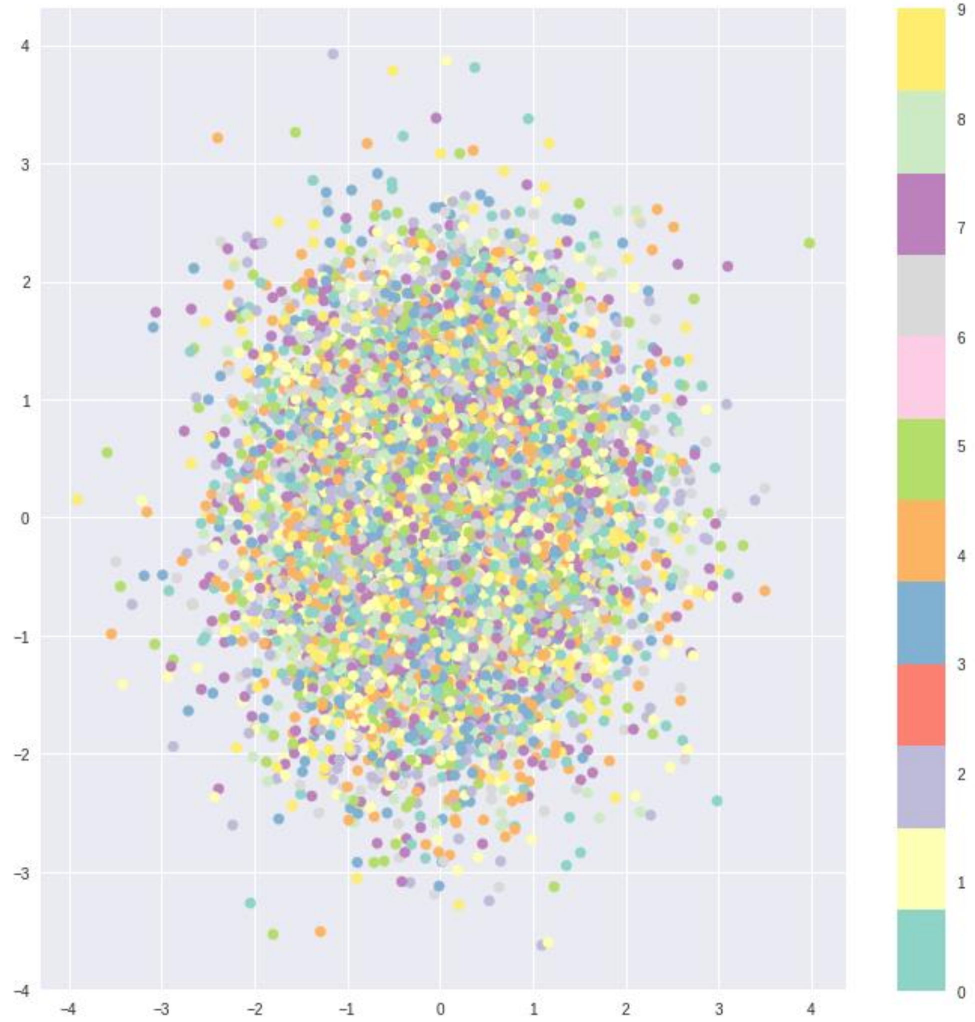
- If only $||X - f(z)||^2$ is chosen for optimization on a dataset, it results in creation of distinct clusters for different image classes.
- Therefore, the decoder is not able to produce any variance from an input image, it learns to replicate the input, as in standard autoencoder.



Source: https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf

Training

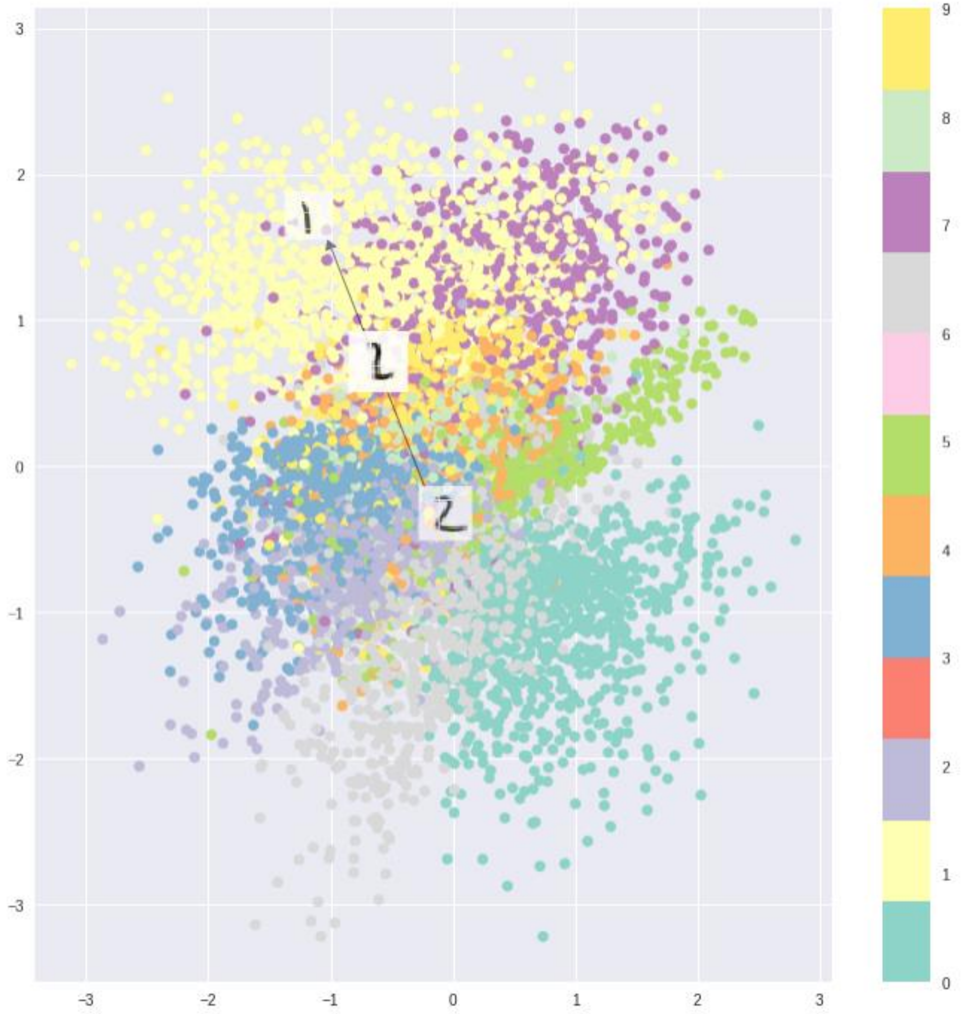
- If only KL loss $D[Q(z|X) \,||\, P(z)]$ is chosen for optimization, it results in encoded vectors that are densely placed near the center of the latent space
- The decoder is not able to infer properly due to denseness in the latent space with not much variation.



Source: https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf

Training

- Optimizing the two together helps in generation of a latent space which maintains the similarity of nearby encoded vectors by clustering them together.
- And is very dense near the latent space origin.
- So the decoder is able to learn enough about variances in encoded vectors.

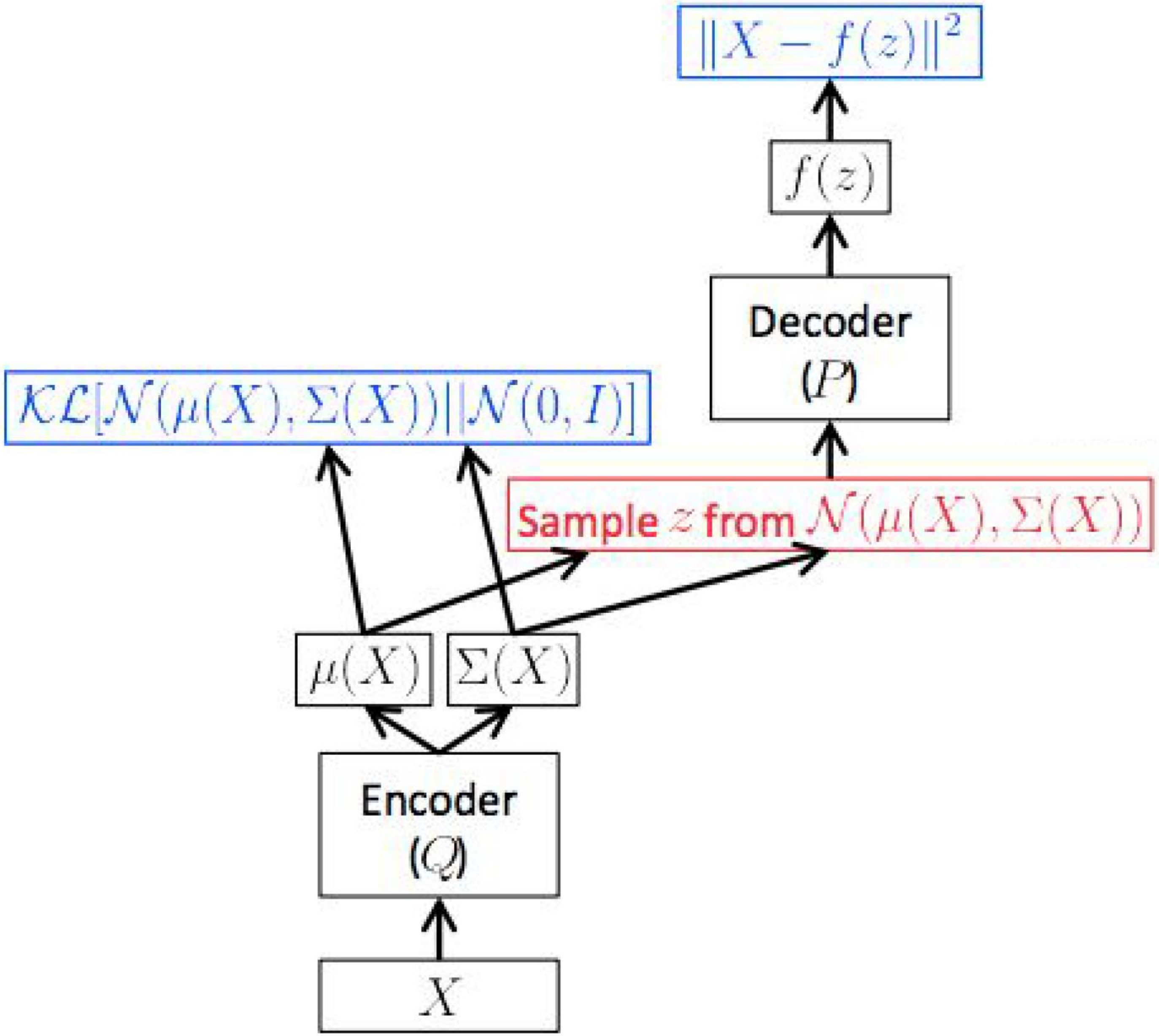


Source: https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf

Training

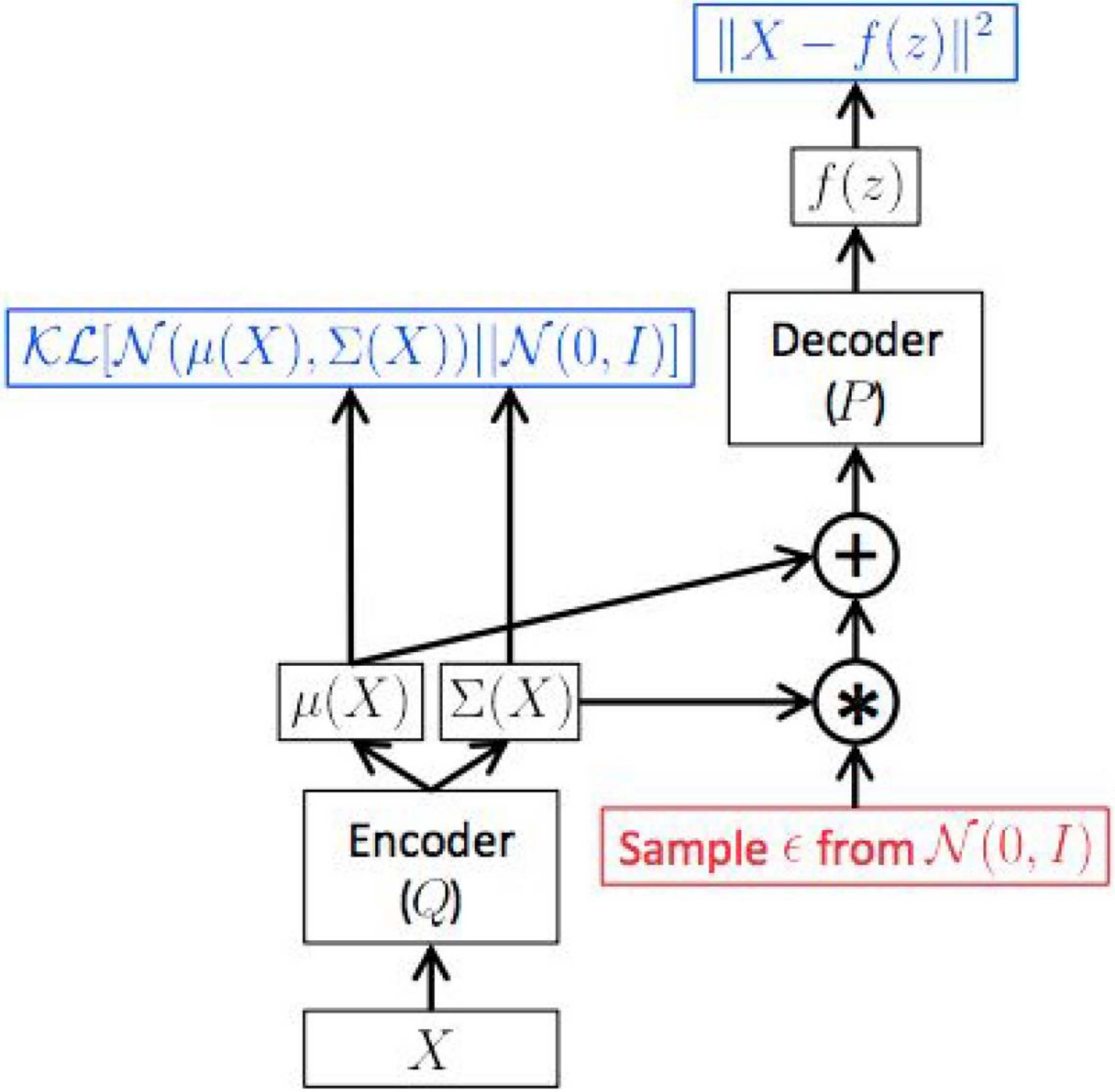
- Training the Decoder is easy using the standard backpropagation.

- Training of encoder requires a better thinking. How to get the distribution ?



Reparameterization

- How to effectively backpropagate through the z samples to the Encoder?
- **Reparametrization Trick**
- z ~ N(μ, σ) is equivalent to μ + σ · ε, where ε ~ N(0, 1)
- Once this is done, we can easily backpropagate the loss to the Encoder.



Training

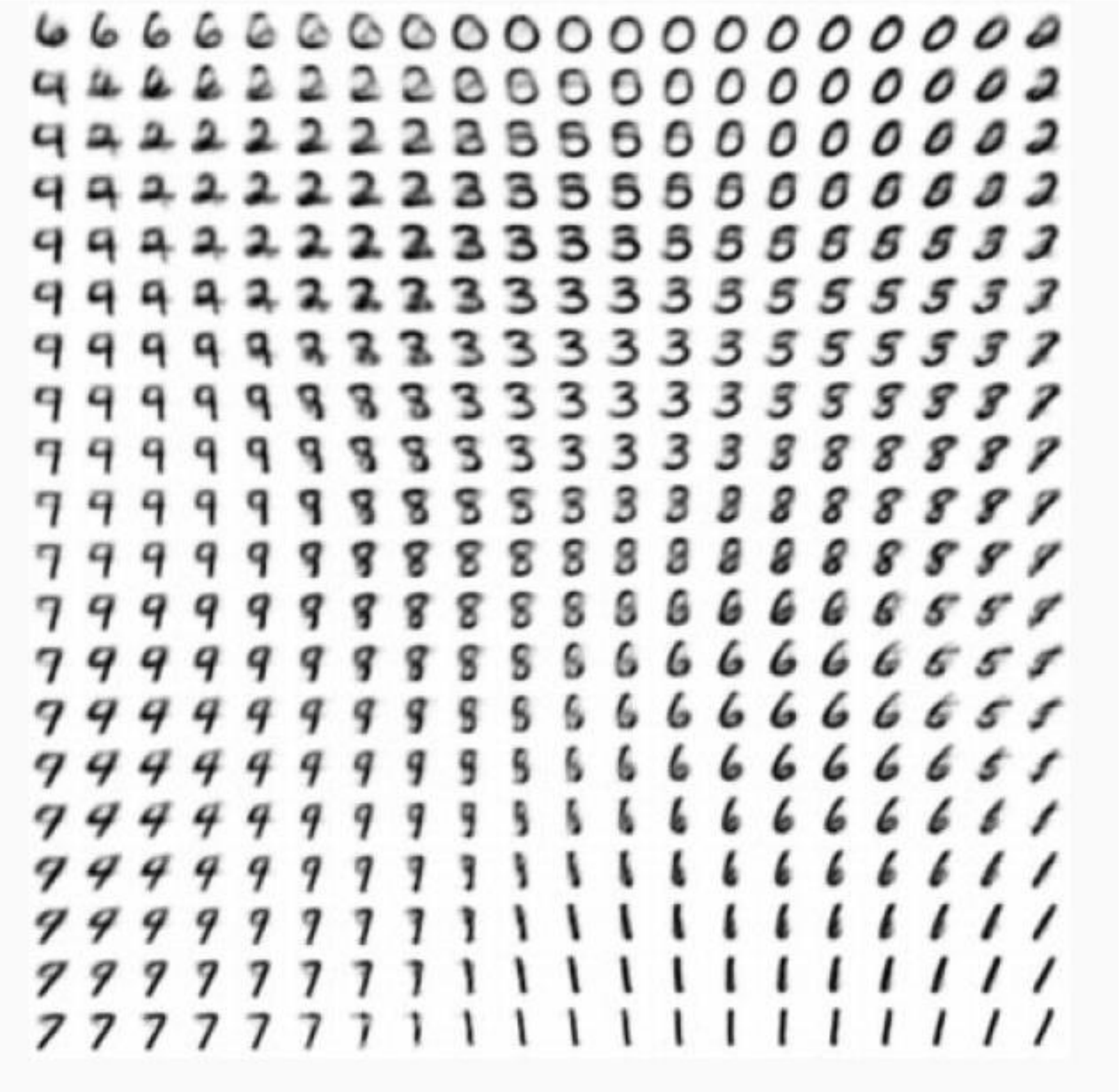
- Given a dataset of examples X = {X1, X2...}
- Initialize parameters for Encoder and Decoder
- **Repeat till convergence:**
- Take a random minibatch *XM* of M examples from X
- ε <-- Sample M noise vectors from N(0, I)
- Compute the loss *L*(*XM*, ε, θ) after a forward pass in the neural network.
- Use gradient descent on *L* to update Encoder and Decoder.

Testing

- To evaluate the performance of VAE on generating a new sample.
- Sample z ~ $N(0,I)$ and pass it through the Decoder.
- No role of encoder as the latent variable itself is passed through the decoder.
- No good measure, relies on visual inspection.

VAE on MNIST Dataset

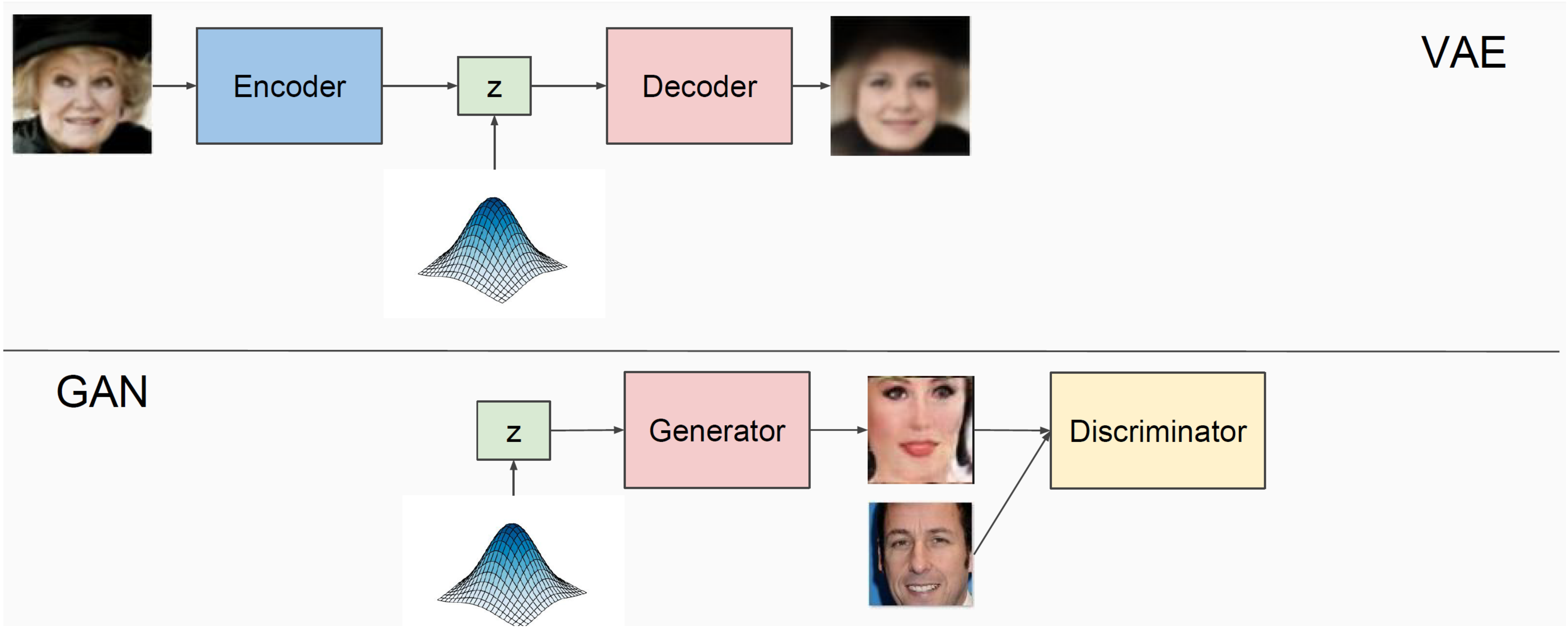
- As you see, distinct digits smoothly transform from one digit to another.
- This smooth transformation is useful when we want to interpolate between two observations, like a smiling face and a laughing face, a face without and with spectacles.



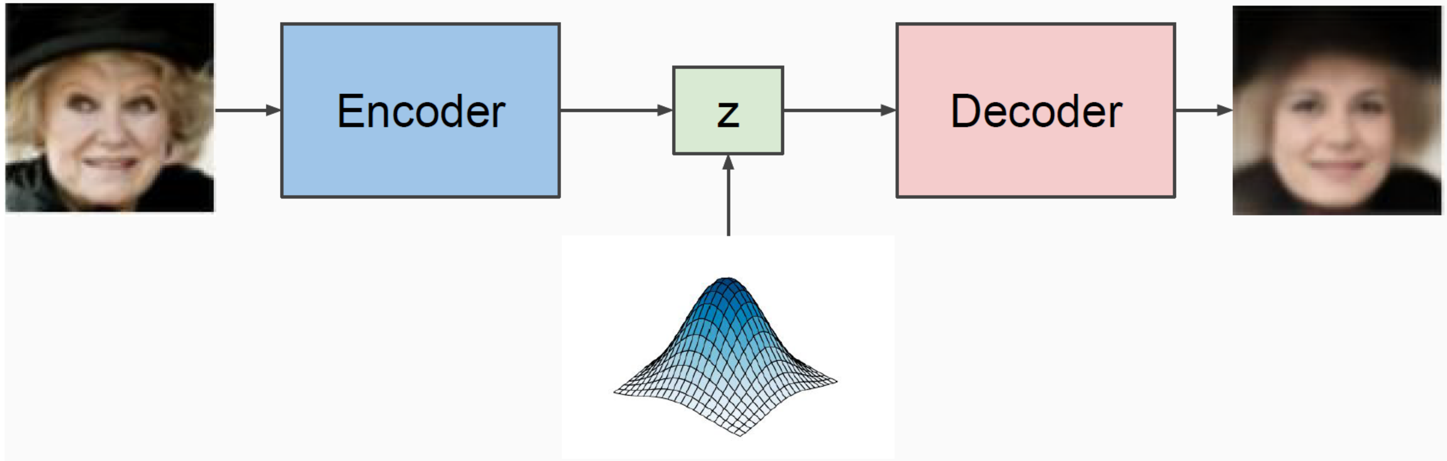
A Comprehensive Repository of Generative Model Codes

- https://github.com/wiseodd/generative-models

VAE and GAN: Comparison



VAE and GAN: Comparison

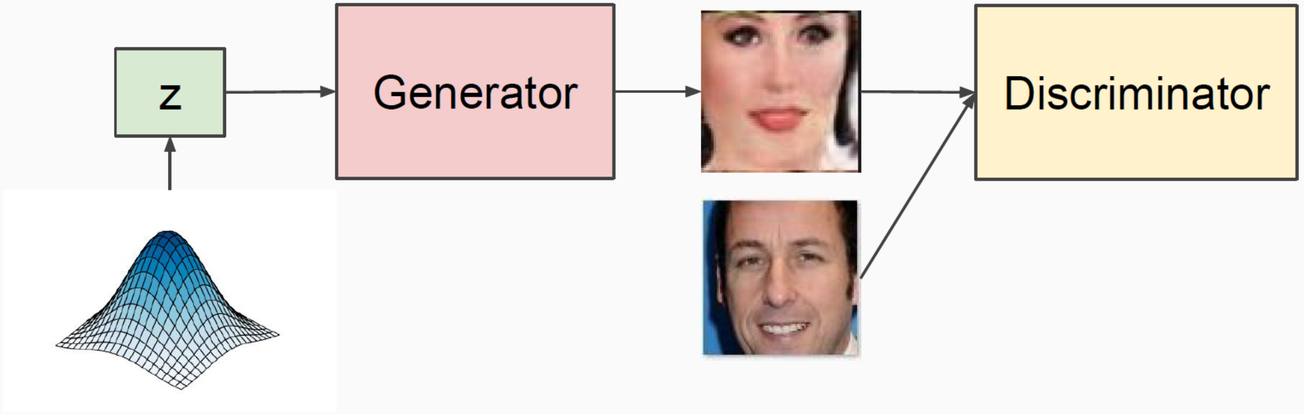


VAE

✓: Given an X **easy** to find z.
✓: Interpretable probability P(X)

X: Usually outputs blurry Images

GAN

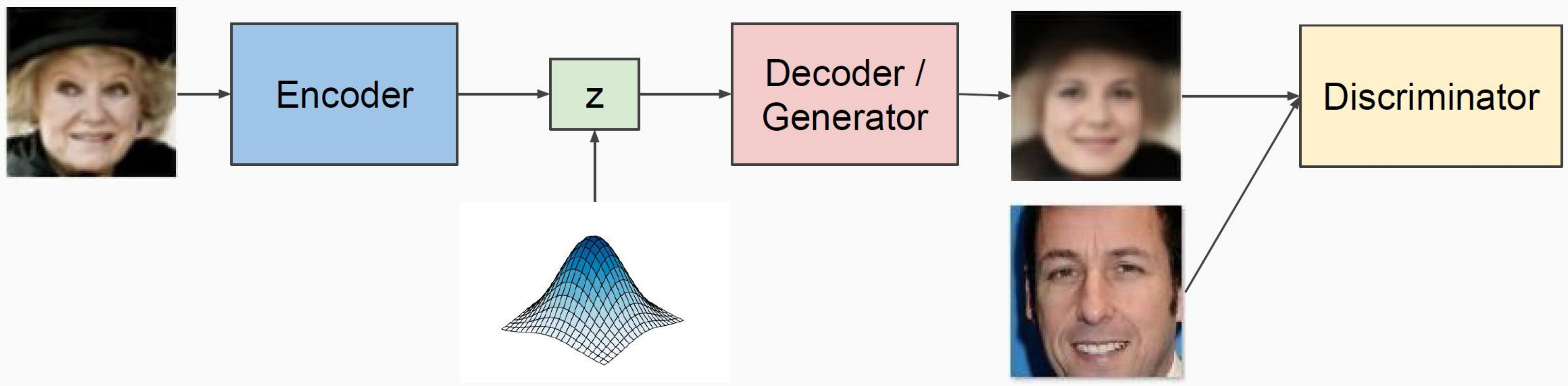


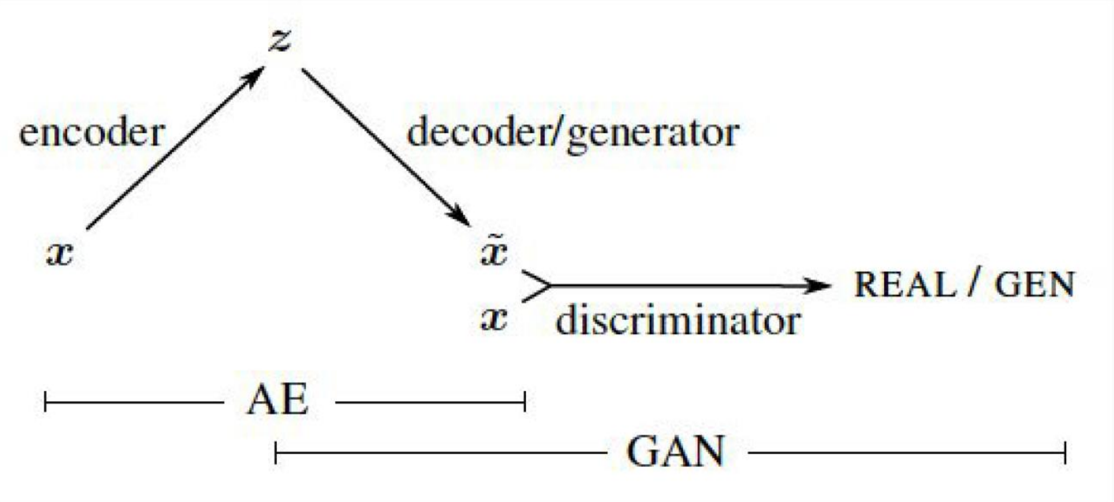
✓: Very sharp images

X: Given an X **difficult** to find z. (Need to backprop.)

✓/X: No explicit P(X).

Combined VAE + GAN

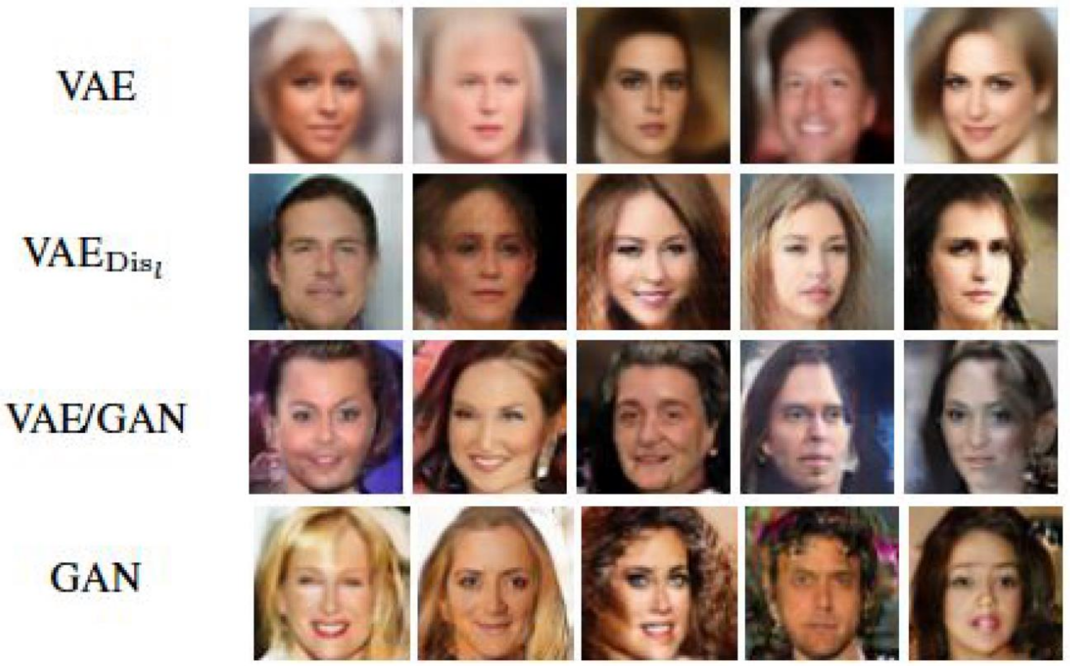




KL Divergence L_2 Difference

$$\mathcal{L} = \mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{llike}}^{\text{Dis}_l} + \mathcal{L}_{\text{GAN}}$$

Results



VAE$_{Dis_l}$: Train a GAN first, then use the discriminator of GAN to train a VAE.

VAE/GAN: GAN and VAE trained together.

Acknowledgement

- Sincere thanks to Prof. S. Lazebnik and her team from Illinois University for permitting to use their course material for lecture slides.