Variational AutoEncoder

Dr. Mohamed Reda Bouadjenek

School of Information Technology,
Faculty of Sci Eng & Built Env

reda.bouadjenek@deakin.edu.au

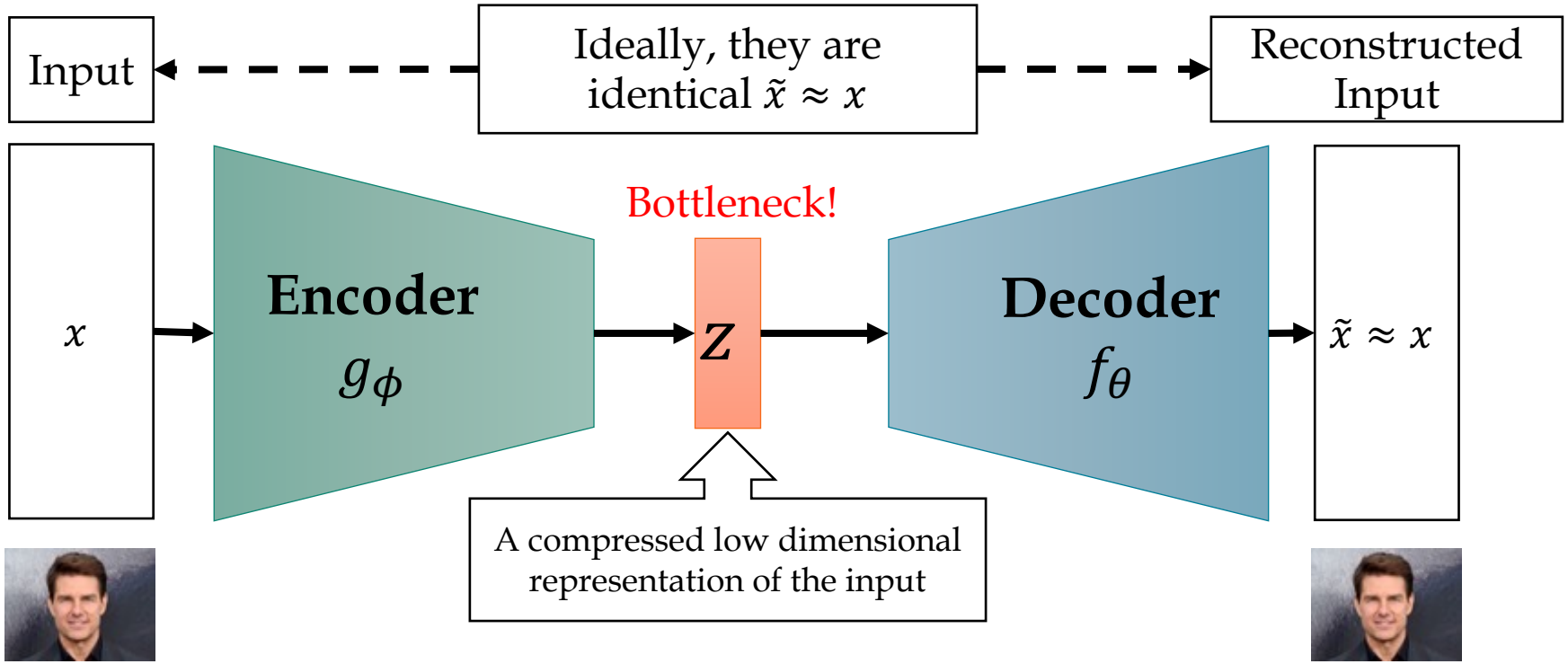
June 7, 2023

Outline

- Intuition and introduction to Variational AutoEncoders (VAE)
- KL Divergence and its significance
- Working details of Variational AutoEncoder
- Derivation of Loss function for Variational AutoEncoder
- Optimization and Reparametrization Trick

Introduction

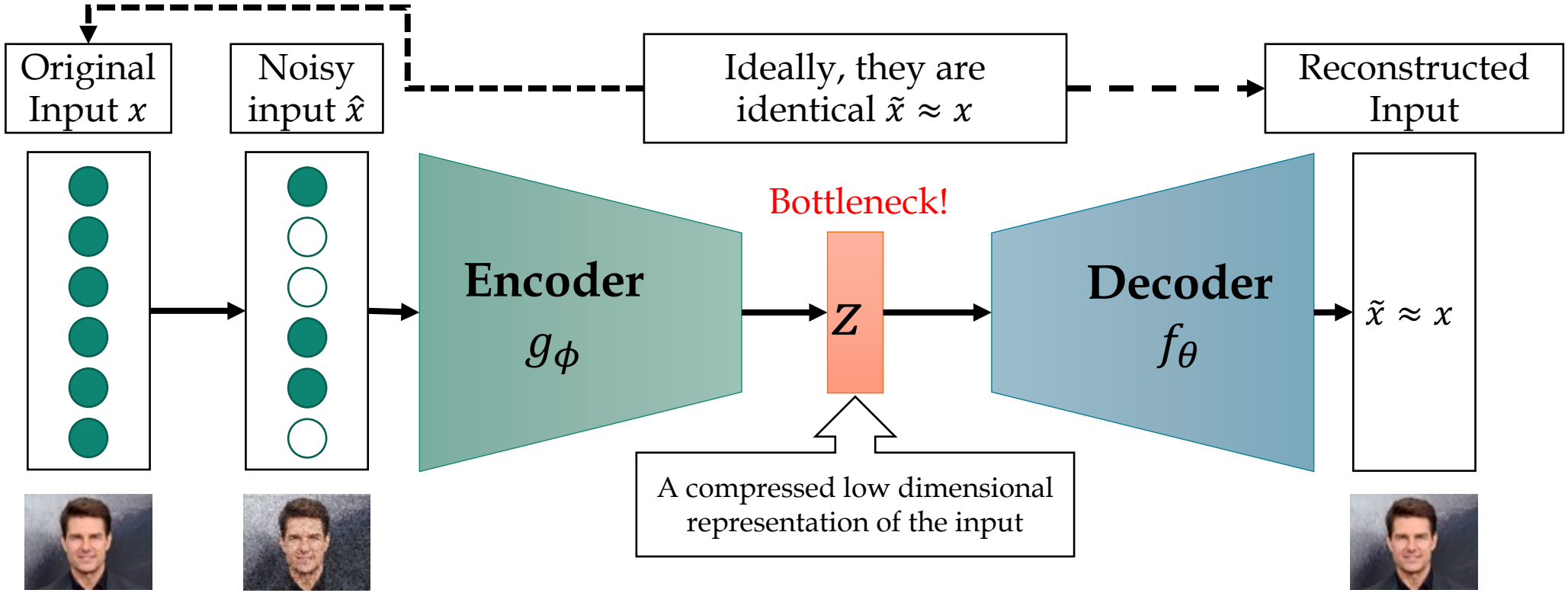
Stacked AutoEncoders for image reconstruction



Cost function:

$$\mathcal{L}(\theta, \phi) = \frac{1}{n}\sum_{i=0}^{n}\left(x^{(i)} - f_\theta\left(g_\phi\left(x^{(i)}\right)\right)\right)^2$$

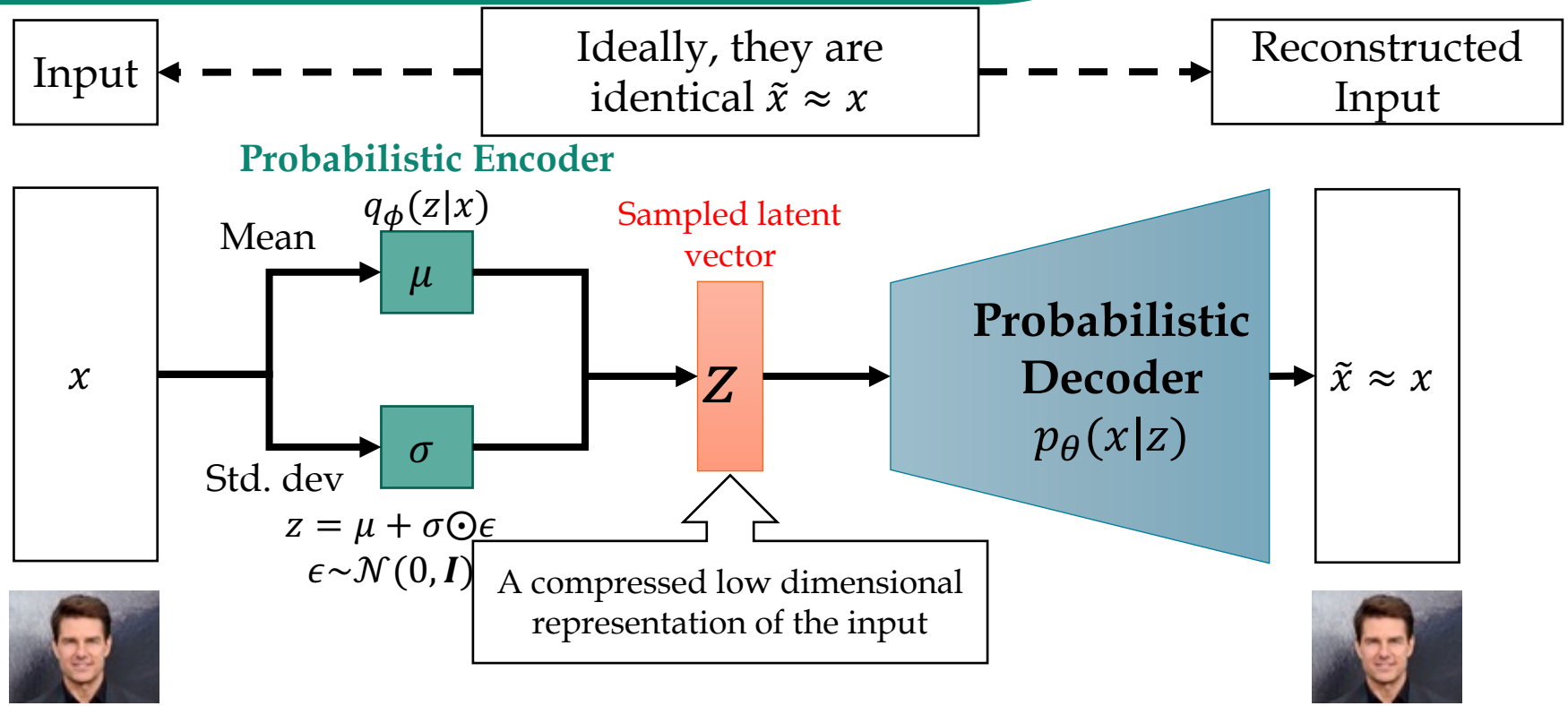
Denoising AutoEncoders for image reconstruction



Cost function:

$$\mathcal{L}(\theta, \phi) = \frac{1}{n}\sum_{i=0}^{n}\left(x^{(i)} - f_\theta\left(g_\phi\left(\hat{x}^{(i)}\right)\right)\right)^2$$

Variational AutoEncoders

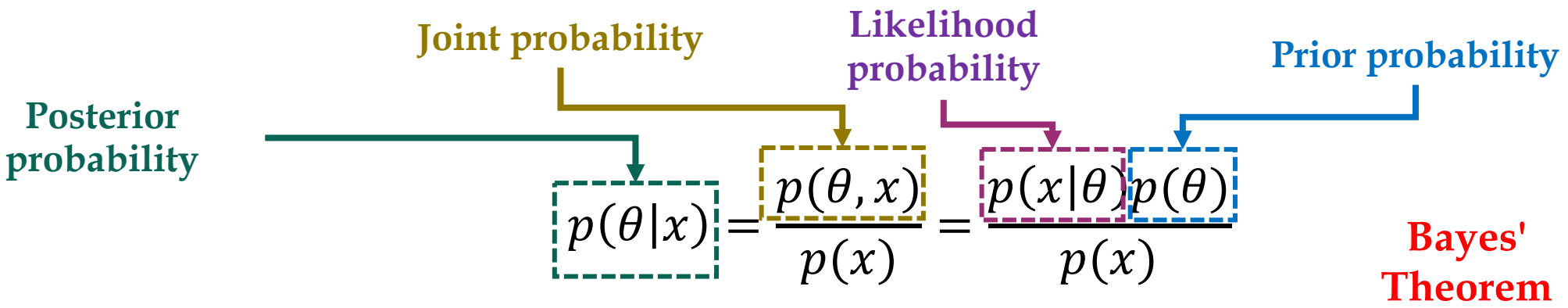


Cost function:

$$\mathcal{L}(\theta, \phi) = -\mathbb{E}_{z \sim q_\phi(z|x)}\left[\log\left(p_\theta(x|z)\right)\right] + D_{\boldsymbol{KL}}\left[q_\phi(z|x) \parallel p_\theta(z)\right]$$

Pre-requisite to understand VAE

- Probability
  - $p(x)$: defines the probability of a random variable x
  - $p(x|y)$: defines the probability of a random variable x given that y ahs happened. Also called conditional probability
  - $\mathbb{E}[x]$
  - KL Divergence



- **Theorem of Total Probability**
  - Let $\theta_1, \theta_2 \cdots \theta_n$ be a set of mutually exclusive events (i.e., $\theta_i \cap \theta_j = 0$) and x is the union of N mutually exclusive events, then:

$$p(x) = \sum_{i=0}^{n} p(x|\,\theta_i)p(\theta_i)$$

- By substitution we get:

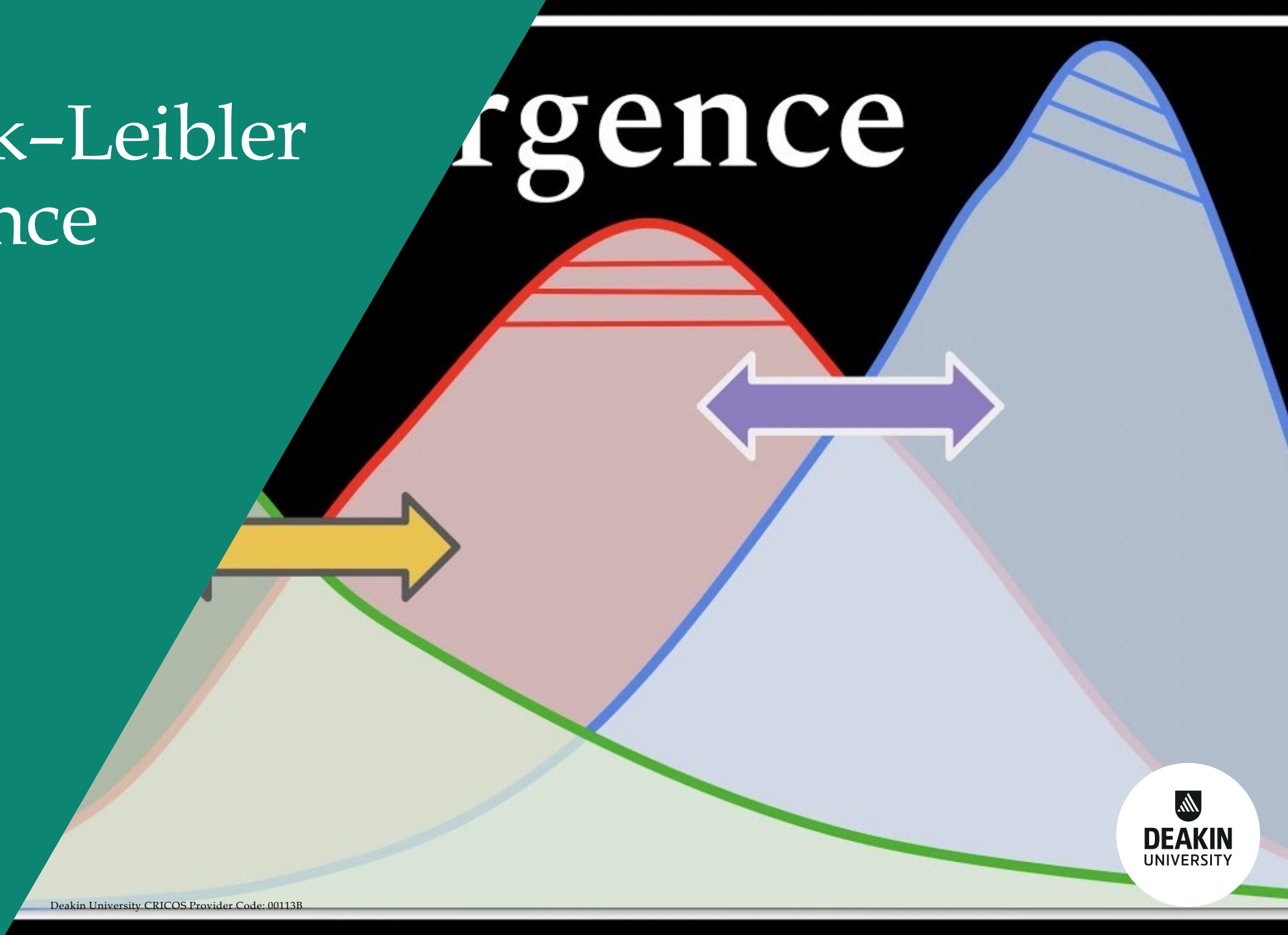
$$p(\theta|x) = \frac{p(\theta,x)}{p(x)} = \frac{p(x|\theta)p(\theta)}{\sum_{i=0}^{n} p(x|\,\theta_i)p(\theta_i)}$$

Expectation of a random variable x

- The expected value of a random variable x is a weighted average of the possible values that x can take. It is defined as:

$$\mathbb{E}[x] = \sum_{i=0}^{n} x_i \; p(x = x_i) = \mathbb{E}_p[x]$$

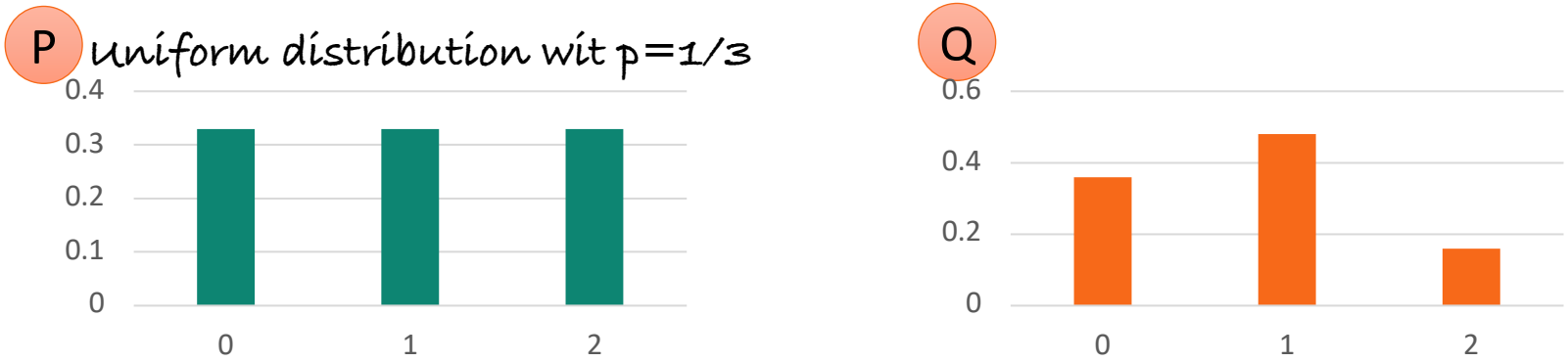
Kullback–Leibler divergence



- The Kullback-Leibler divergence (KL) is a measure of how one probability distribution is different from the second
- Given two discrete probability distributions p and q define, the KL divergence is defined as

$$D_{KL}(p \parallel q) = \sum_x p(\mathrm{x}) \log\left(\frac{p(x)}{q(x)}\right)$$

- Example



$$D_{KL}([p \parallel q) = \frac{1}{3}\log\left(\frac{0.33}{0.36}\right) + \frac{1}{3}\log\left(\frac{0.33}{0.48}\right) + \frac{1}{3}\log\left(\frac{0.33}{0.16}\right)$$

$$= 0.09637$$

- Properties:
  - $D_{KL}(p \parallel q)$ or $D_{KL}(q \parallel p) \geq 0$
  - $D_{KL}(p \parallel q) \neq D_{KL}(q \parallel p) \geq 0$ (Not symmetric)
- Suppose we have two multivariate normal distributions defined as:

$$p(x) = \mathcal{N}(x; \mu_1, \Sigma_1)$$

$$q(x) = \mathcal{N}(x; \mu_2, \Sigma_2)$$

  - Where μ_1 and μ_2 are the means and Σ_1 and Σ_2 are the covariance matrices
  - And the multivariate density is defined as:

$$\mathcal{N}(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

  - If the two distributions have the same dimension k, then:

$$D_{KL}(p \parallel q) = \frac{1}{2}\left[\log\frac{\Sigma_2}{\Sigma_1} - k + tr(\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)^T \Sigma_2^{-1}(\mu_2 - \mu_1)\right]$$

- Proof:
  - We know:

$$D_{KL}(p \parallel q) = \sum_{x} p(\mathrm{x}) \log\left(\frac{p(x)}{q(x)}\right) \quad (1)$$

  - We also know that:

$$\mathrm{p}(x) = \frac{1}{\sqrt{(2\pi)^k |\Sigma_1|}} \exp\left(-\frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1} (x-\mu_1)\right)$$

  - If we take the logarithm, we get:

$$\log\big(p(x)\big) = -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_1| - \frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1} (x-\mu_1) \quad (2)$$

  - Similarly:

$$\log\big(q(x)\big) = -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_2| - \frac{1}{2}(x-\mu_2)^T \Sigma_2^{-1} (x-\mu_2) \quad (3)$$

- Equation 1 can rewritten as:

$$D_{KL}(p \parallel q) = \sum_{x} p(\mathrm{x}) \left(\log\big(p(x)\big) - \log(q(x))\right)$$

- Substituting 2 and 3 in 1 results in:

$$D_{KL}(p \parallel q) = \sum_{x} p(\mathrm{x}) \left(-\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_1| - \frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right.$$

$$\left. + \frac{k}{2}\log(2\pi) + \frac{1}{2}\log|\Sigma_2| + \frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2)\right)$$

- Which can be simplified to:

$$D_{KL}(p \parallel q) = \sum_{x} p(\mathrm{x}) \left(\frac{1}{2} \log \frac{|\Sigma_2|}{|\Sigma_1|} + \frac{1}{2} (x - \mu_2)^T \Sigma_2^{-1} (x - \mu_2) - \frac{1}{2} (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1) \right)$$

- Now, let's consider part by part:

$$\sum_{x} p(\mathrm{x}) \left(\frac{1}{2} (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1) \right) = \mathbb{E}_p \left[\frac{1}{2} (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1) \right]$$

- Let's rewrite again:

Scalar

$$\mathbb{E}_p\left[\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right]$$

$$\mathbb{E}_p\left[tr\left(\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right)\right] \quad \text{A}$$

$$\mathbb{E}_p\left[tr\left(\frac{1}{2}(x-\mu_1)(x-\mu_1)^T\Sigma_1^{-1}\right)\right] \quad \text{D}$$

Covariance matrix

$$tr\left(\mathbb{E}_p[(x-\mu_1)(x-\mu_1)^T]\frac{1}{2}\Sigma_1^{-1}\right) \quad \text{E}$$

$$tr\left(\Sigma_1\frac{1}{2}\Sigma_1^{-1}\right)$$

$$tr(I_k)=\frac{k}{2}$$

- Trace and expectation trick
  - If x is a scalar, then $\mathbb{E}[x]=\mathbb{E}[tr(x)]$ A
  - $tr(AB)=tr(BA)$ B
  - $tr(ABC)=tr(BCA)$ C
    $=tr(CAB)$ D
  - $tr(ABC)\neq tr(ACB)$
  - $\mathbb{E}[tr(x)]=tr(\mathbb{E}[x])$ E

$$D_{KL}(p \parallel q) = \sum_x p(\mathrm{x}) \left(\frac{1}{2} \log \frac{|\Sigma_2|}{|\Sigma_1|} + \frac{1}{2} (x - \mu_2)^T \Sigma_2^{-1} (x - \mu_2) - \frac{1}{2} (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1) \right)$$

- Now, let's consider part by part:

$$\sum_x p(\mathrm{x}) \left(\frac{1}{2} (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1) \right) = \mathbb{E}_p \left[\frac{1}{2} (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1) \right]$$

$$\frac{k}{2}$$

$$D_{KL}(p \parallel q) = \sum_{x} p(\mathrm{x}) \left(\frac{1}{2} \log \frac{|\Sigma_2|}{|\Sigma_1|} + \frac{1}{2} (x - \mu_2)^T \Sigma_2^{-1} (x - \mu_2) - \frac{1}{2} (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1) \right)$$

- Now, let's consider the second part:

$$\sum_{x} p(\mathrm{x}) \left(\frac{1}{2} (x - \mu_2)^T \Sigma_2^{-1} (x - \mu_2) \right)$$

$$\sum_x p(\mathrm{x}) \left(\frac{1}{2} (x - \mu_2)^T \Sigma_2^{-1} (x - \mu_2) \right)$$

$$\sum_x p(\mathrm{x}) \left(\frac{1}{2} [(x - \mu_1) + (\mu_1 - \mu_2)]^T \Sigma_2^{-1} [(x - \mu_1) + (\mu_1 - \mu_2)] \right)$$

$$\sum_x p(\mathrm{x}) \left(\frac{1}{2} [(x - \mu_1)^T + (\mu_1 - \mu_2)^T] \Sigma_2^{-1} [(x - \mu_1) + (\mu_1 - \mu_2)] \right)$$

$$\sum_x p(\mathrm{x}) \left(\frac{1}{2} [(x - \mu_1)^T \Sigma_2^{-1} + (\mu_1 - \mu_2)^T \Sigma_2^{-1}][(x - \mu_1) + (\mu_1 - \mu_2)] \right)$$

If A is a symmetric matrix, then $x^T Ay = y^T Ax$

$$\sum_x p(\mathrm{x}) \left(\frac{1}{2} [(x - \mu_1)^T \Sigma_2^{-1} (x - \mu_1) + (x - \mu_1)^T \Sigma_2^{-1} (\mu_1 - \mu_2) + (\mu_1 - \mu_2)^T \Sigma_2^{-1} (x - \mu_1) + (\mu_1 - \mu_2)^T \Sigma_2^{-1} (\mu_1 - \mu_2)] \right)$$

$$\sum_x p(\mathrm{x}) \left(\frac{1}{2} [(x - \mu_1)^T \Sigma_2^{-1} (x - \mu_1) + 2(x - \mu_1)^T \Sigma_2^{-1} (\mu_1 - \mu_2) + (\mu_1 - \mu_2)^T \Sigma_2^{-1} (\mu_1 - \mu_2)] \right)$$

$$\mathbb{E}_p \left[\frac{1}{2} (x - \mu_1)^T \Sigma_2^{-1} (x - \mu_1) + (x - \mu_1)^T \Sigma_2^{-1} (\mu_1 - \mu_2) + \frac{1}{2} (\mu_1 - \mu_2)^T \Sigma_2^{-1} (\mu_1 - \mu_2) \right]$$

$$\mathbb{E}_p\left[\frac{1}{2}(x-\mu_1)^T\Sigma_2^{-1}(x-\mu_1)+(x-\mu_1)^T\Sigma_2^{-1}(\mu_1-\mu_2)+\frac{1}{2}(\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2)\right]$$

- Expanding, we get

$$\mathbb{E}_p\left[\frac{1}{2}(x-\mu_1)^T\Sigma_2^{-1}(x-\mu_1)\right]+\mathbb{E}_p\left[(x-\mu_1)^T\Sigma_2^{-1}(\mu_1-\mu_2)\right]+\mathbb{E}_p\left[\frac{1}{2}(\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2)\right]$$

$$=tr\left(\frac{\Sigma_1\Sigma_2^{-1}}{2}\right)+0+\frac{1}{2}(\mu_1-\mu_2)^T\Sigma_2^{-1}(\mu_1-\mu_2)$$

Similar to previous derivative

Proof on next slide

$\mathbb{E}$[constant]= constant

$$\mathbb{E}_p[(x-\mu_1)^T\Sigma_2^{-1}(\mu_1-\mu_2)] = \left(\mathbb{E}_p[x]-\mu_1\right)^T\Sigma_2^{-1}(\mu_1-\mu_2)$$

$$= (\mu_1-\mu_1)^T\Sigma_2^{-1}(\mu_1-\mu_2)$$

$$= 0$$

In summary

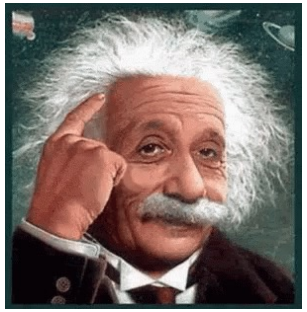
$$D_{KL}(p \parallel q) = \sum_x p(\mathrm{x}) \left(\frac{1}{2} \log \frac{|\Sigma_2|}{|\Sigma_1|} + \frac{1}{2}(x-\mu_2)^T \Sigma_2^{-1} (x-\mu_2) - \frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1} (x-\mu_1) \right) \quad 1$$

$$\sum_x p(\mathrm{x}) \left(\frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1} (x-\mu_1) \right) = \frac{k}{2} \quad 2$$

$$\sum_x p(\mathrm{x}) \left(\frac{1}{2}(x-\mu_2)^T \Sigma_2^{-1} (x-\mu_2) \right) = \frac{1}{2} tr(\Sigma_1 \Sigma_2^{-1}) + \frac{1}{2}(\mu_1-\mu_2)^T \Sigma_2^{-1} (\mu_1-\mu_2) \quad 3$$

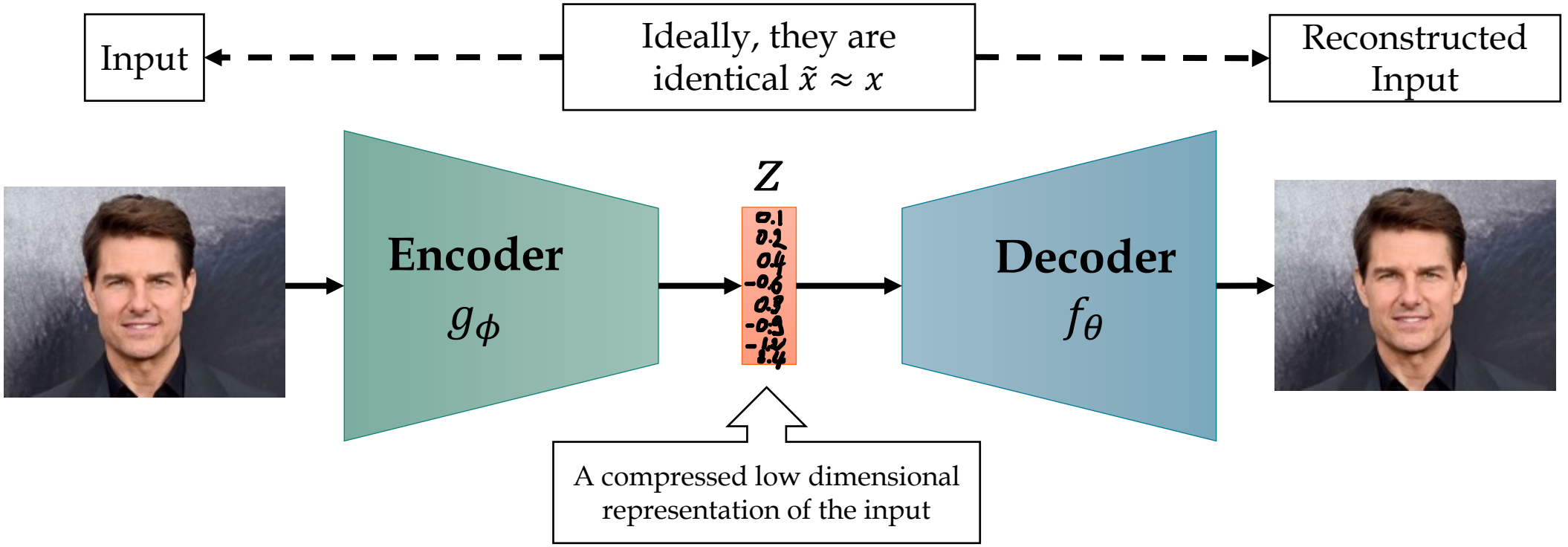
- Substituting 2 and 3 in 1 we obtain:

$$D_{KL}(p \parallel q) = \frac{1}{2} \left[\log \frac{|\Sigma_2|}{|\Sigma_1|} - k + tr(\Sigma_2^{-1}\Sigma_1) + (\mu_2-\mu_1)^T \Sigma_2^{-1} (\mu_2-\mu_1) \right]$$



Working details of Variational AutoEncoder

Stacked AutoEncoders for image reconstruction

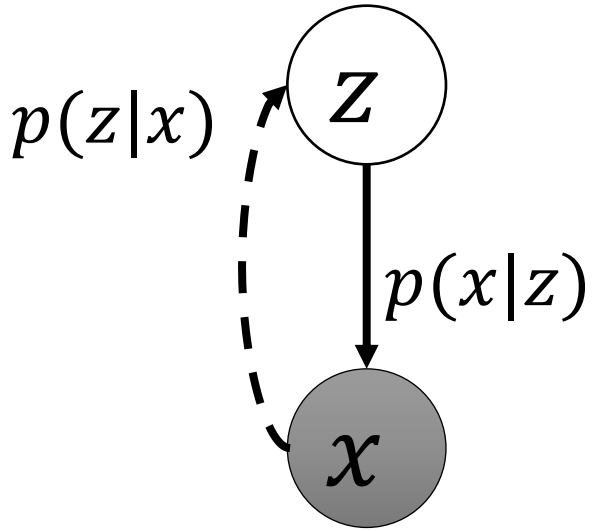


Cost function:

$$\mathcal{L}(\theta, \phi) = \frac{1}{n}\sum_{i=0}^{n}\left(x^{(i)} - f_\theta\left(g_\phi\left(x^{(i)}\right)\right)\right)^2$$

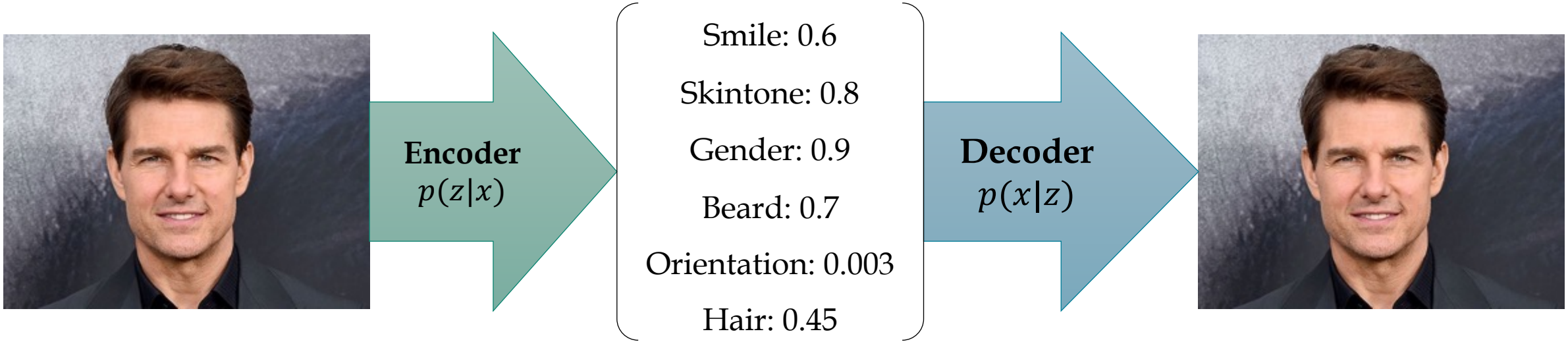
Latent variables

- We model the system as a collection of random variables
  - The edge "→" draw from z to x is the conditional distribution $p(x|z)$



What are the latent variables?

- Latent variables z correspond to real features or characteristics of the object

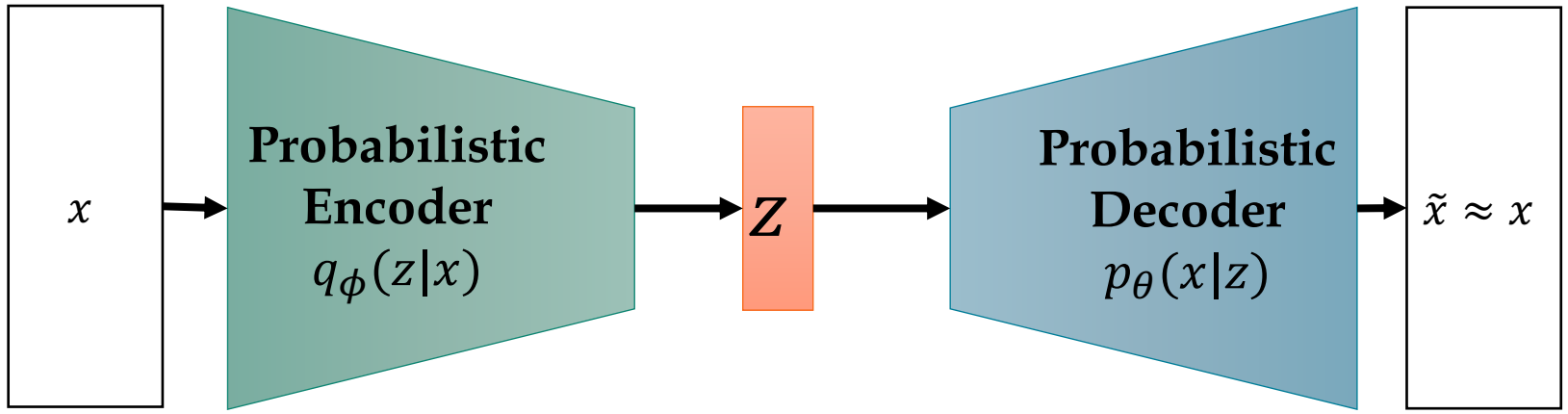


What are the latent variables?

- In the above example, we trained autoencoder on a large dataset of faces with encoding dimension of 6
  - An ideal autoencoder will learn the descriptive attributes of faces such as skin color, smile, etc… in order to describe an observation in some compressed form.
- In the above example, we have described the input image in terms of latent variables using single value to describe each attribute.
  - For instance, what single value will assign for photo of Monalisa?

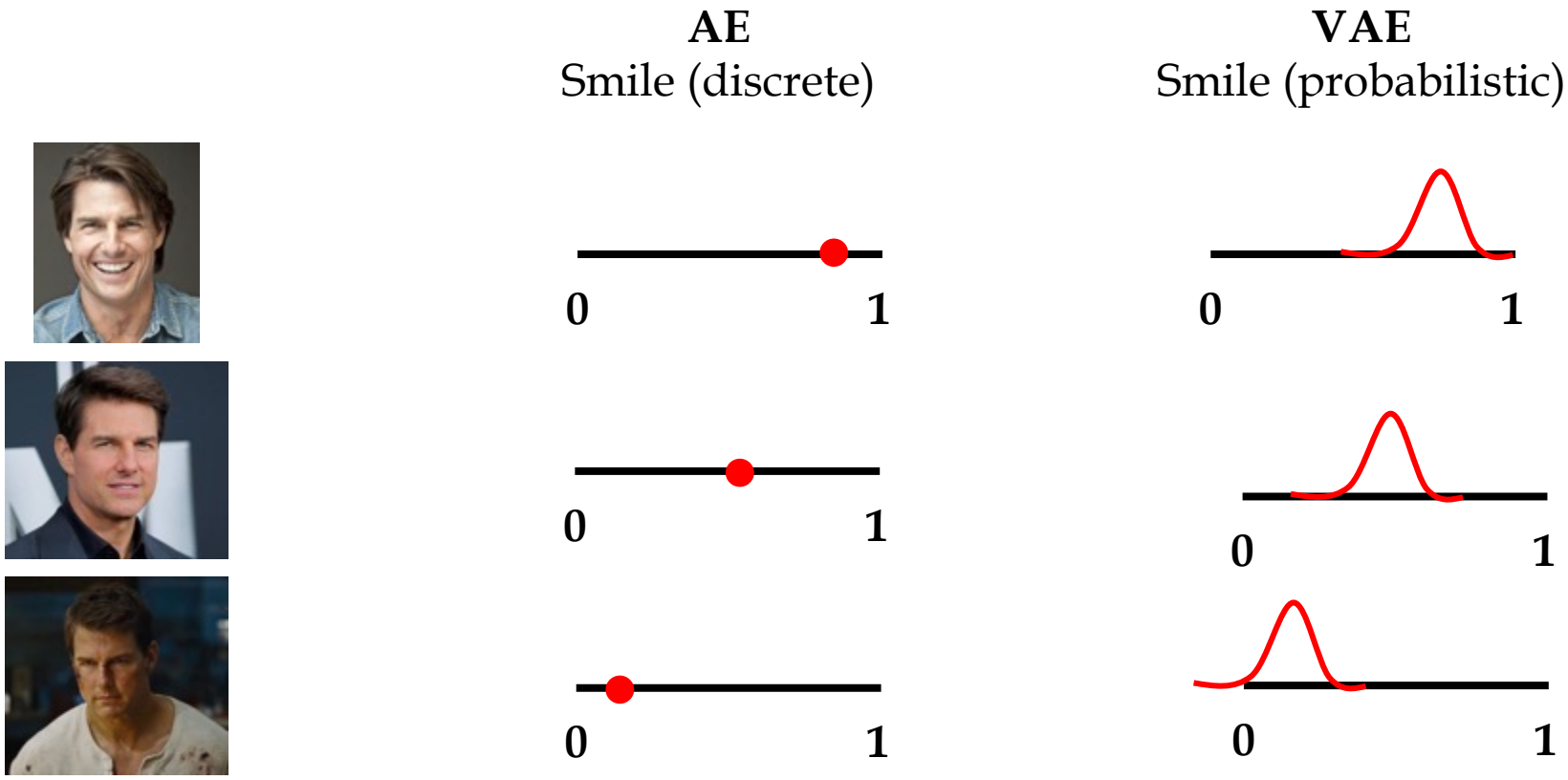
The goal of a VAE

- The goal of a VAE is to find a distribution $q_\phi(z|x)$ of some latent variables, which we can sample from $z \sim q_\phi(z|x)$, to generate new samples $\tilde{x}$ from $p_\theta(x|z)$



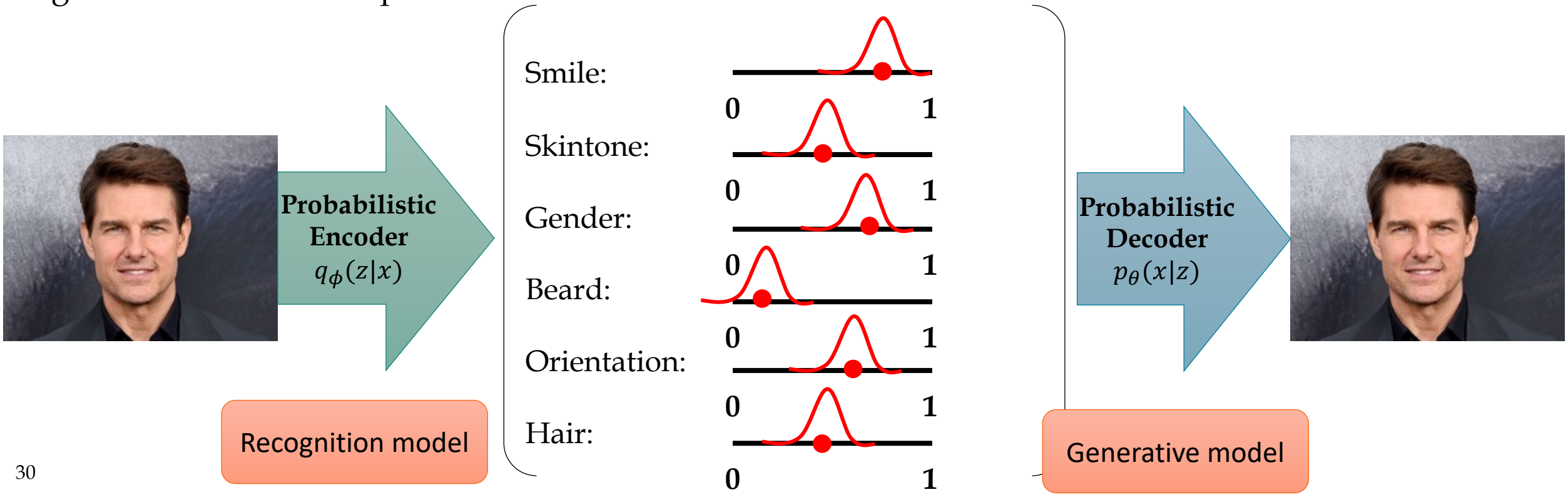
AutoEncoders vs. VAE

- Using VAE, we define latent attributes in probabilistic terms



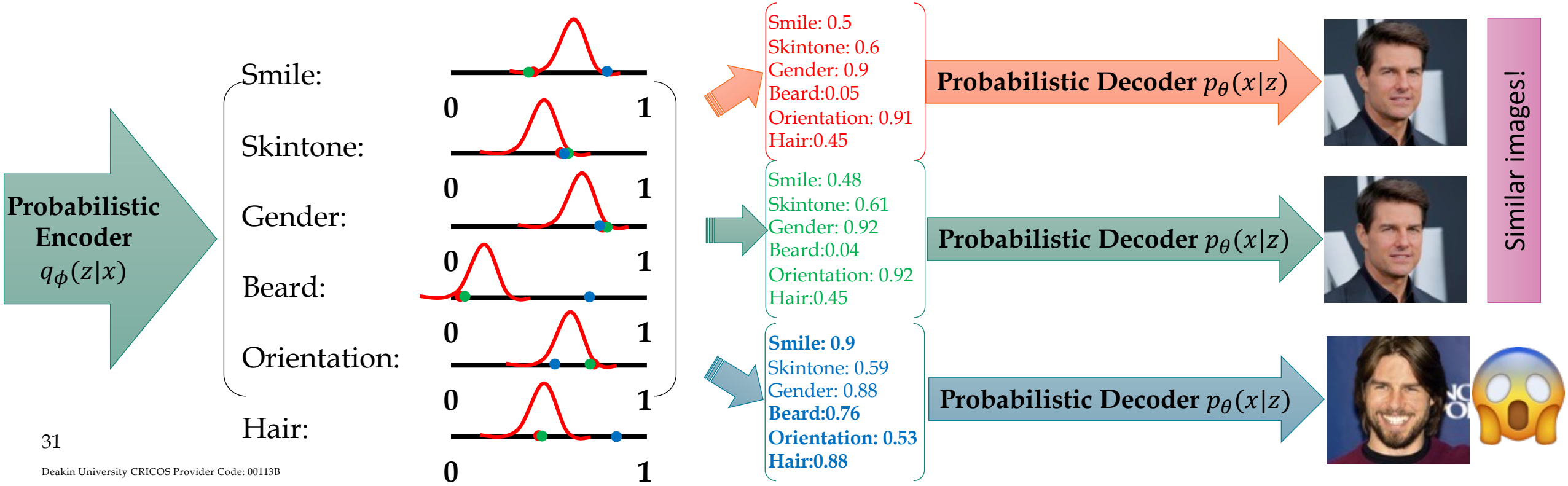
What are the latent variables in VAE?

- With this approach, we now represent each latent attribute for a given input as a probability distribution. When decoding, we will randomly sample from each latent state distribution to generate a vector as input for the decoder

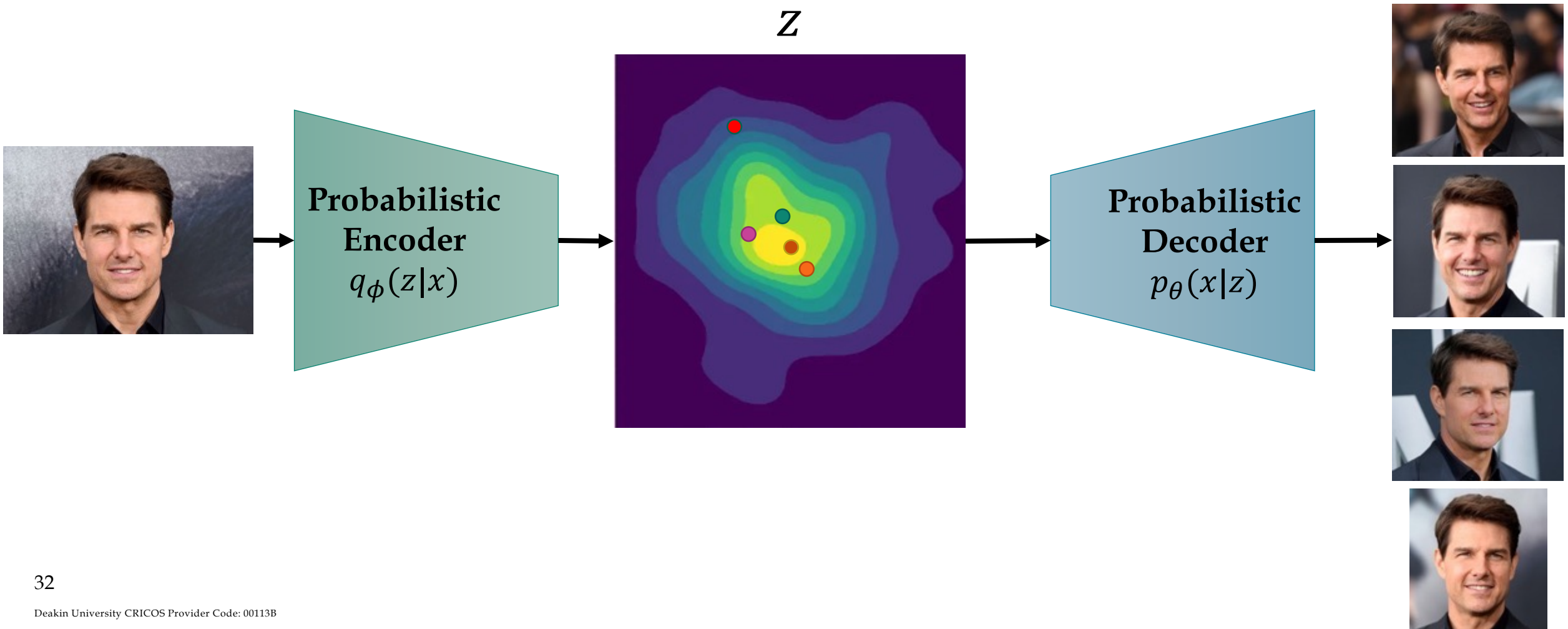


What are the latent variables in VAE?

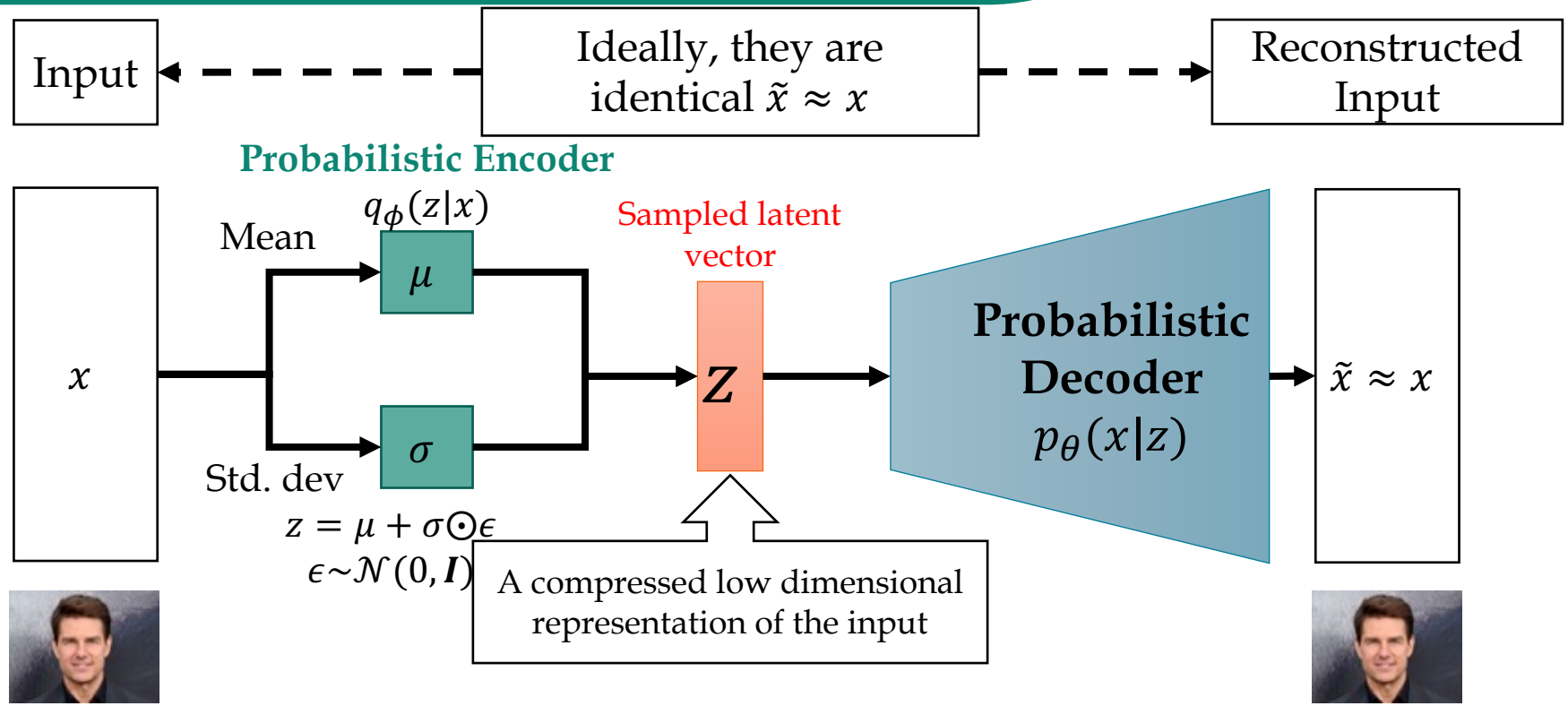
- If we design our encoder model to produce a statistical distribution of potential values, we can randomly select from that distribution and input those values into our decoder model. This means that values that are located close together in latent space will result in similar reconstructions.



VAE example



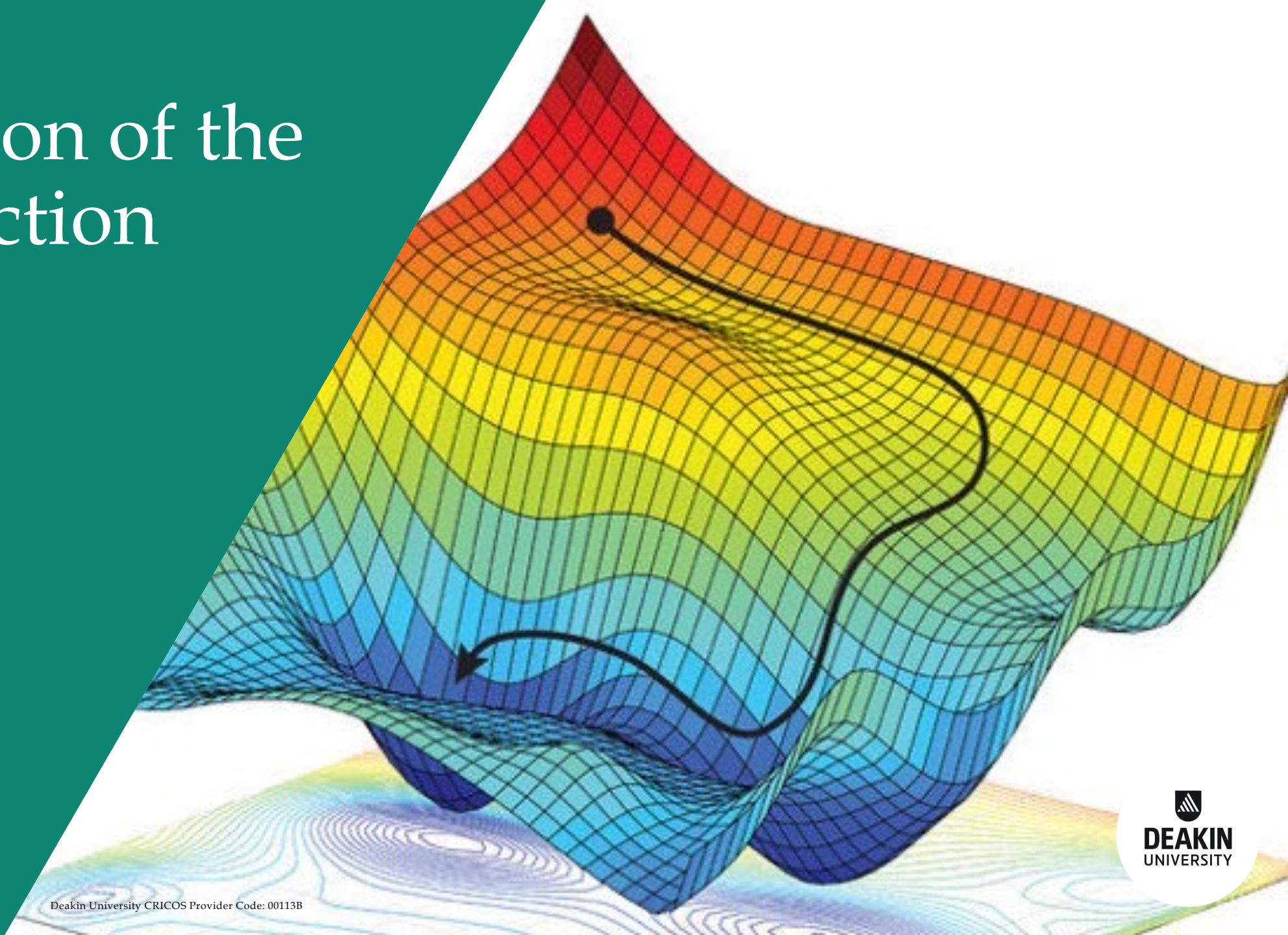
Variational AutoEncoders



Cost function:

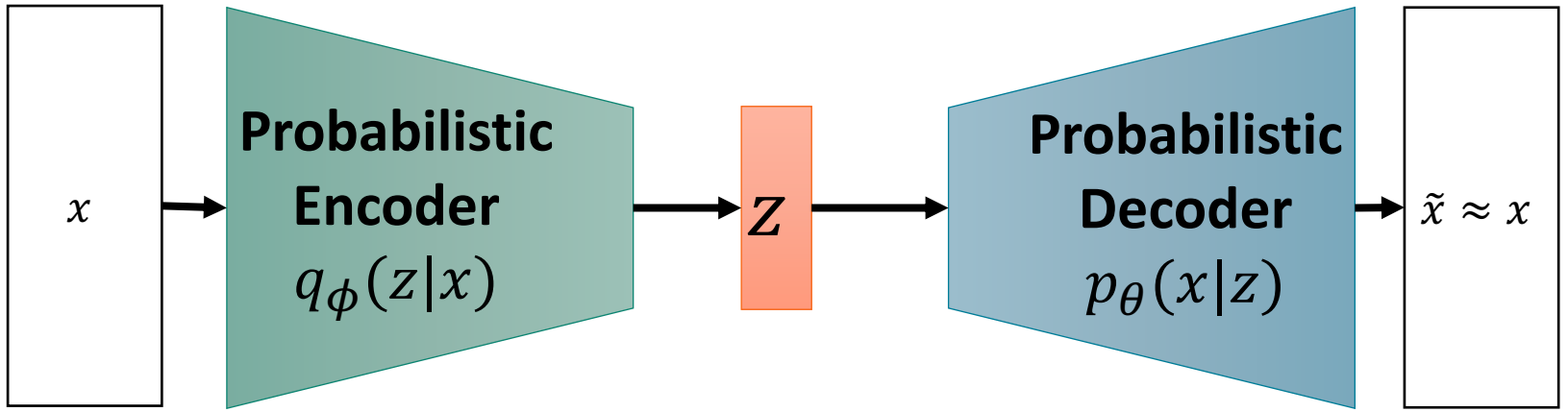
$$\mathcal{L}(\theta, \phi) = -\mathbb{E}_{z \sim q_\phi(z|x)}\left[\log\left(p_\theta(x|z)\right)\right] + D_{\boldsymbol{KL}}\left[q_\phi(z|x) \parallel p_\theta(z)\right]$$

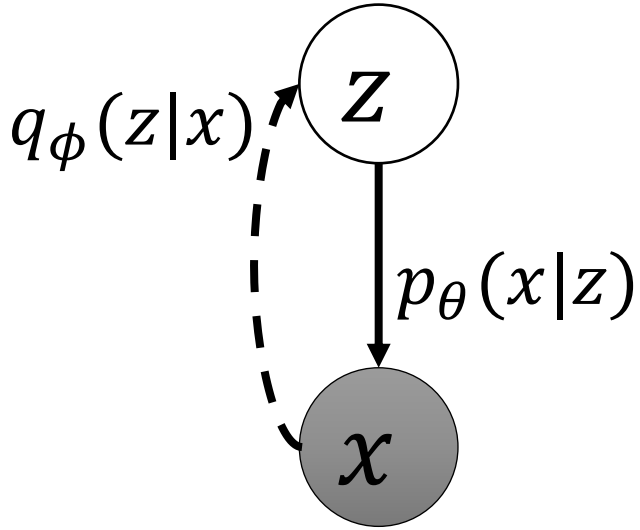
Derivation of the loss function



The goal of a VAE

- The goal of a VAE is to find a distribution $q_\phi(z|x)$ of some latent variables, which we can sample from $z \sim q_\phi(z|x)$, to generate new samples $\tilde{x}$ from $p_\theta(x|z)$



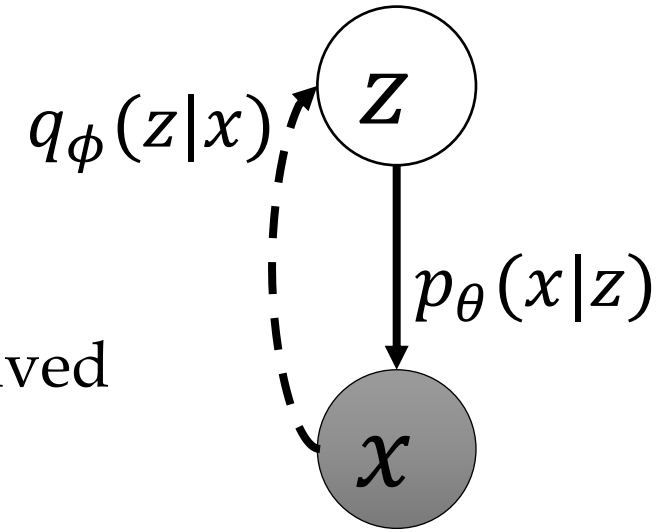


The problem of Approximate Inference

- Let x be a set of observed variables and let z be the set of latent variables with joint distribution $p(z, x)$. Then, the inference problem is to compute the conditional distribution of the latent variables given the observations, i.e., $p(z|x)$. We can write it as:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

A



- Evaluating this equation is difficult because $p(x)$ cannot be solved

Reason:

$$p(x) = \int_z p(x,z)dz = \iiint_z p(x|z)p(z)dz_i$$

- The integral is not available in closed form or is intractable (i.e., requires exponential time to compute) due to multiple integrals involved for the latent variable vector z
- **Alternative?**
  - The alternative is to approximate $p(z|x)$ by another distribution $q(z|x)$ which is defined in such a way that it has tractable solution. This is done using Variational Inference (VI).

Variational Inference (VI)

- The main idea of Variational Inference (VI) is to pose the inference problem as an optimization problem.
- How?
  - By modeling $p(z|x)$ using $q(z|x)$ where $q(z|x)$ has a simple distribution such as Gaussian.
  - Let's calculate KL divergence between $p(z|x)$ and $q(z|x)$:

$$D_{KL}\left(q_\phi(z|x) \parallel p_\theta(z|x)\right) = \sum_z q_\phi(z|x)\log\left(\frac{q_\phi(z|x)}{p_\theta(z|x)}\right)$$

$$= \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log\left(\frac{q_\phi(z|x)}{p_\theta(z|x)}\right)\right]$$

$$= \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log\left(q_\phi(z|x)\right) - \log\left(p_\theta(z|x)\right)\right]$$

B

Variational Inference (VI)

- Substituting (A) in (B) results in:

$$D_{KL}\left(q_\phi(z|x) \parallel p_\theta(z|x)\right) = \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log\left(q_\phi(z|x)\right) - \log\left(\frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)}\right)\right]$$

$$= \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log\left(q_\phi(z|x)\right) - \log\left(p_\theta(x|z)\right) - \log\left(p_\theta(z)\right) + \log\left(p_\theta(x)\right)\right]$$

- Since the expectation is over z and $p_\theta(x)$ does not involve z, it can be moved out

$$D_{KL}\left(q_\phi(z|x) \parallel p_\theta(z|x)\right) - \log\left(p_\theta(x)\right) = \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log\left(q_\phi(z|x)\right) - \log\left(p_\theta(x|z)\right) - \log\left(p_\theta(z)\right)\right]$$

Variational Inference (VI)

- Rearranging the equation, we obtain:

$\boldsymbol{z}: \boldsymbol{z} \sim \boldsymbol{q_\phi(z|x)}$

$$D_{KL}\big(q_\phi(z|x) \parallel p_\theta(z|x)\big) - \log\big(p_\theta(x)\big) = -\mathbb{E}_{\boldsymbol{z}}\big[\log\big(p_\theta(x|z)\big)\big] + \mathbb{E}_{\boldsymbol{z}}\left[\log\left(q_\phi(z|x)\right) - \log\big(p_\theta(z)\big)\right]$$

$$\mathcal{L}(\theta, \phi) = -\mathbb{E}_{\boldsymbol{z}}\big[\log\big(p_\theta(x|z)\big)\big] + D_{\boldsymbol{KL}}\big[q_\phi(z|x) \parallel p_\theta(z)\big]$$

- This is the VAE loss function, where the first term represents the reconstruction likelihood, and the second term ensures that our learned distributions $\boldsymbol{q}$ is similar to the prior distribution $\boldsymbol{p}$
- Also, we have:

$$\mathcal{L}(\theta, \phi) = D_{KL}\big(q_\phi(z|x) \parallel p_\theta(z|x)\big) - \log\big(p_\theta(x)\big)$$

VAE loss function

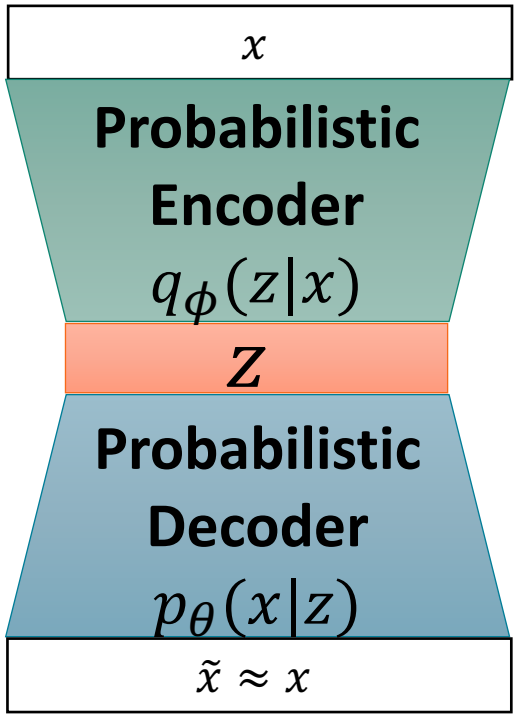
$$\mathcal{L}(\theta, \phi) = \underbrace{-\mathbb{E}_{z \sim q_\phi(z|x)}\left[\log\left(p_\theta(x|z)\right)\right]}_{\text{Reconstruction term}} + \underbrace{D_{KL}\left[q_\phi(z|x) \parallel p_\theta(z)\right]}_{\text{Regularizer term}}$$

- So, our target is to find optimal θ, ϕ such that

$$\theta^*, \phi^* = \underset{\theta,\phi}{\operatorname{argmin}}\, \mathcal{L}(\theta, \phi)$$

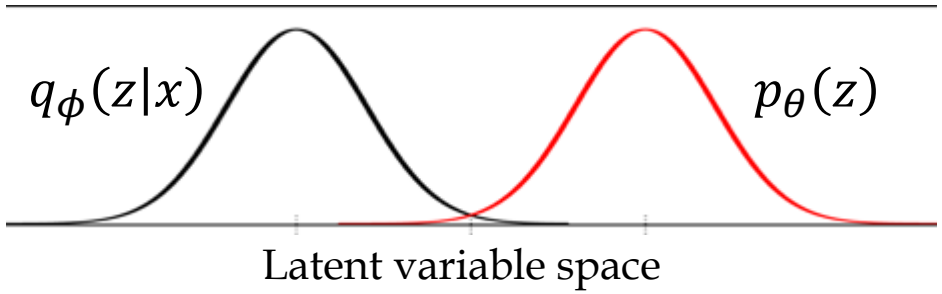
Intuition about the loss function

$$\mathcal{L}(\theta, \phi) = \underbrace{-\mathbb{E}_{z\sim q_\phi(z|x)}\left[\log\left(p_\theta(x|z)\right)\right]}_{\text{Reconstruction term}} + \underbrace{D_{KL}\left[q_\phi(z|x) \parallel p_\theta(z)\right]}_{\text{Regularizer term}}$$



$$p_\theta(x|z) = \mathcal{N}\left(\mu_\theta(z), \Sigma_\theta(z)\right)$$

So, we take log of Gaussian we get a square error between the data sample x and mean of the Gaussian distribution



KL divergence ensures that the pdf of latent variables $q_\phi(z|x)$ does not collapse with zero variance but penalizes if deviates from $\mathcal{N}(0,1) = p_\theta(z)$

VAE loss function: reconstruction term 1/2

- We assume that the probabilistic decoder is is modeled as a Gaussian for regression (multivariate Bernoulli for classification):

$$p_\theta(x|z) = \mathcal{N}\left(\mu_\theta(z), \Sigma_\theta(z)\right)$$

- Furthermore, we assume that the components of the Gaussian are independent. Hence, we have:

$$p_\theta(x|z) = \mathcal{N}\left(\mu_\theta(z), \Sigma_\theta(z)\right) = \prod_{i=0}^{k} \mathcal{N}\left(\mu_i(z), \sigma_i(z)\right) = \prod_{i=0}^{k} \frac{1}{\sqrt{2\pi\sigma_i}} \exp\left(-\frac{\left(x_i - \mu_i(z)\right)^2}{2{\sigma_i}^2}\right)$$

VAE loss function: reconstruction term 2/2

- Taking the logarithm, we get:

$$\log\left(p_\theta(x|z)\right) = \log \prod_{i=0}^{k} \frac{1}{\sqrt{2\pi\sigma_i}} \exp\left(-\frac{\left(x_i - \mu_i(z)\right)^2}{2{\sigma_i}^2}\right)$$

$$= \sum_{i=0}^{k} \log \frac{1}{\sqrt{2\pi\sigma_i}} \exp\left(-\frac{\left(x_i - \mu_i(z)\right)^2}{2{\sigma_i}^2}\right)$$

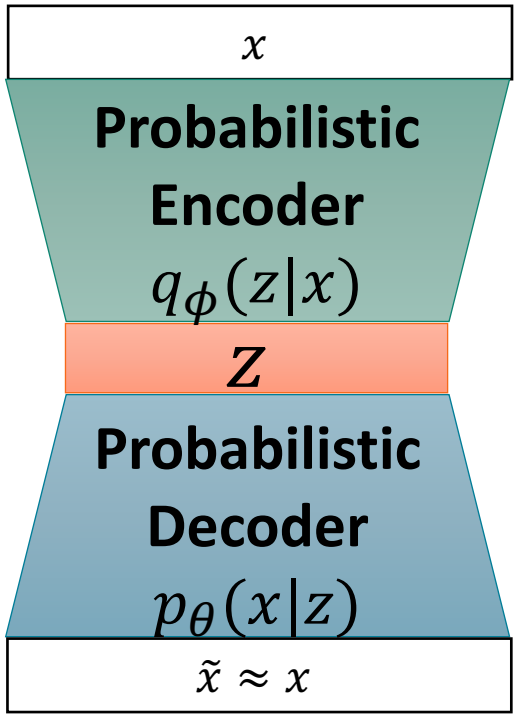
$$= \sum_{i=0}^{k} \log \frac{1}{\sqrt{2\pi\sigma_i}} - \sum_{i=0}^{k} \frac{1}{2{\sigma_i}^2}\left(x_i - \mu_i(z)\right)^2$$

$$\propto -\sum_{i=0}^{k} \frac{1}{2}\left(x_i - \mu_i(z)\right)^2 = -\frac{1}{2}\left(x - \mu_\theta(z)\right)^2$$

Intuition about the loss function

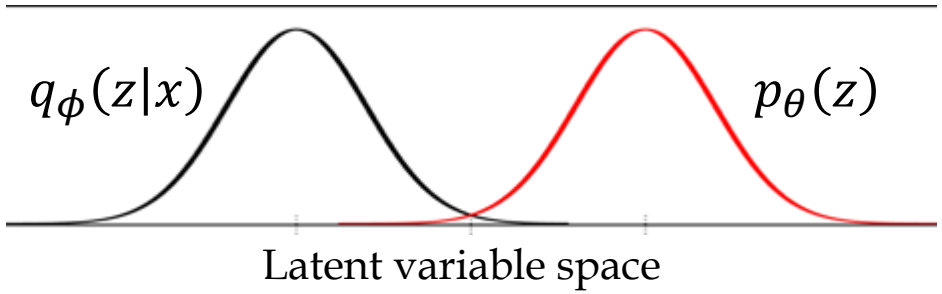
$$\mathcal{L}(\theta, \phi) = \underbrace{-\mathbb{E}_{z\sim q_\phi(z|x)}\left[\log\left(p_\theta(x|z)\right)\right]}_{\text{Reconstruction term}} + \underbrace{D_{KL}\left[q_\phi(z|x) \parallel p_\theta(z)\right]}_{\text{Regularizer term}}$$

$\mathcal{N}(0,1)$



$$p_\theta(x|z) = \mathcal{N}\left(\mu_\theta(z), \Sigma_\theta(z)\right)$$

So, we take log of Gaussian we get a square error between the data sample x and mean of the Gaussian distribution



KL divergence ensures that the pdf of latent variables $q_\phi(z|x)$ does not collapse with zero variance but penalizes if deviates from $\mathcal{N}(0,1) = p_\theta(z)$

VAE loss function: regularizer term

$$D_{KL}\left[q_\phi(z|x) \parallel p_\theta(z)\right]$$ **Regularizer term**

- Here, $p_\theta(z)$ is the latent variable distribution
  - The easy choice is $\mathcal{N}(0,1)$
- We want $q_\phi(z|x)$ to be as close as possible to $p_\theta(z) = \mathcal{N}(0,1)$ so that we can sample it easily
- Having to $p_\theta(z) = \mathcal{N}(0,1)$ adds another benefit
  - The KL divergence has a closed form!

VAE loss function: regularizer term

- We have previously proved that:

$$D_{KL}(p \parallel q) = \frac{1}{2}\left[\log\frac{|\Sigma_2|}{|\Sigma_1|} - k + tr(\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)^T\Sigma_2^{-1}(\mu_2 - \mu_1)\right]$$

- Given the previous assumption that $p_\theta(z) = \mathcal{N}(0,1)$, we get:

$$D_{\boldsymbol{KL}}\left[q_\phi(z|x) \parallel p_\theta(z)\right] = D_{\boldsymbol{KL}}\left[q_\phi\left(\mu_\phi(x), \Sigma_\phi(x)\right) \parallel \mathcal{N}(0,1)\right]$$

$$= \frac{1}{2}\left[-\log\left(\left|\Sigma_\phi(x)\right|\right) - k + tr\left(\Sigma_\phi(x)\right) + \mu_\phi(x)^T\mu_\phi(x)\right]$$

  - Here k is the dimension of the Gaussian
  - $tr\left(\Sigma_\phi(x)\right)$ is the trace function, which is the sum of diagonal matrix of $\Sigma_\phi(x)$
  - The determinant $\left|\Sigma_\phi(x)\right|$ of a diagonal matrix is product of it's diagonal

VAE loss function: regularizer term

- Hence, we have:

$$D_{KL}\left[q_\phi\left(\mu_\phi(x), \Sigma_\phi(x)\right) \parallel \mathcal{N}(0,1)\right] = \frac{1}{2}\left[-\log\left(\left|\Sigma_\phi(x)\right|\right) - k + tr\left(\Sigma_\phi(x)\right) + \mu_\phi(x)^T \mu_\phi(x)\right]$$

$$= \frac{1}{2}\left[-\log\left(\prod_k \Sigma_\phi(x)\right) - \sum_k 1 + \sum_k \Sigma_\phi(x) + \sum_k \mu_\phi(x)^2\right]$$

$$= \frac{1}{2}\left[-\sum_k \log\left(\Sigma_\phi(x)\right) - \sum_k 1 + \sum_k \Sigma_\phi(x) + \sum_k \mu_\phi(x)^2\right]$$

$$= \frac{1}{2}\sum_k \left[-\log\left(\Sigma_\phi(x)\right) - 1 + \Sigma_\phi(x) + \mu_\phi(x)^2\right]$$

VAE loss function

- So, the final loss function of VAE becomes:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{z \sim q_\phi(z|x)} \left[\frac{1}{2} \left(x - \mu_\theta(z) \right)^2 \right] - \frac{1}{2} \sum_k \left[1 + \log \left(\Sigma_\phi(x) \right) - \mu_\phi(x)^2 - \Sigma_\phi(x) \right]$$

$$= \frac{1}{L} \sum_{l=1}^{L} \frac{1}{2} \left(x - \mu_\theta\left(z^{[l]} \right) \right)^2 - \frac{1}{2} \sum_k \left[1 + \log \left(\Sigma_\phi(x) \right) - \mu_\phi(x)^2 - \Sigma_\phi(x) \right]$$

  - Where $z^{[l]} \sim q_\phi(z|x)$ **(Monte Carlo Estimate)**

Monte Carlo Estimate: A Simple Example

- Monte Carlo estimate provides an approximation of the expectation through random sampling and averaging
- **Example:** Approximating the mean of a biased coin
  - Let X denote a binomial random variable that equals 1 when a coi flip results in heads, and 0 when it lands on tails
  - We can estimate the mean by conducting multiple flips as follows
    - $\mathbb{E}[X] = \frac{1+0+1+1+\cdots+0}{n} = \frac{1}{n}\sum_1^n x_i$

VAE loss function

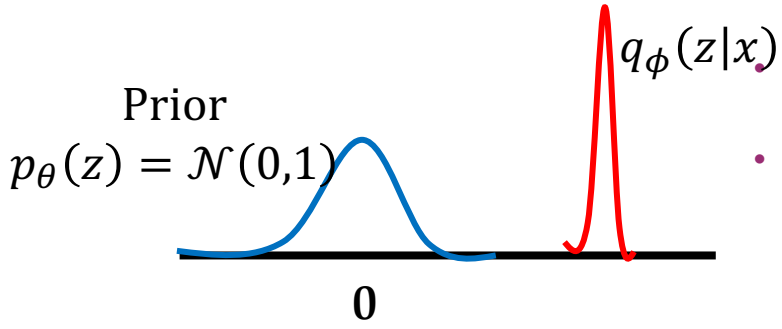
- So, the final loss function of VAE becomes:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{z \sim q_\phi(z|x)} \left[\frac{1}{2} \left(x - \mu_\theta(z) \right)^2 \right] - \frac{1}{2} \sum_k \left[1 + \log \left(\Sigma_\phi(x) \right) - \mu_\phi(x)^2 - \Sigma_\phi(x) \right]$$

$$= \frac{1}{L} \sum_{l=1}^{L} \frac{1}{2} \left(x - \mu_\theta \left(z^{[l]} \right) \right)^2 - \frac{1}{2} \sum_k \left[1 + \log \left(\Sigma_\phi(x) \right) - \mu_\phi(x)^2 - \Sigma_\phi(x) \right]$$

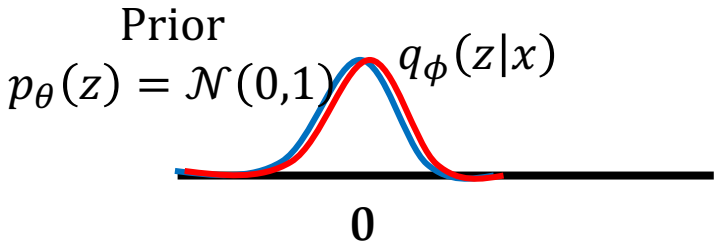
  - Where $z^{[l]} \sim q_\phi(z|x)$ **(Monte Carlo Estimate)**
  - **Most of the time the Monte Carlo Estimate consists into a single draw (the number of samples drawn is a hyperparameter of the VAE model and can vary depending on the complexity of the data and the desired accuracy)**

$\mathcal{L} = data\ fidelity + KL\ Divergence$(Regularizer)

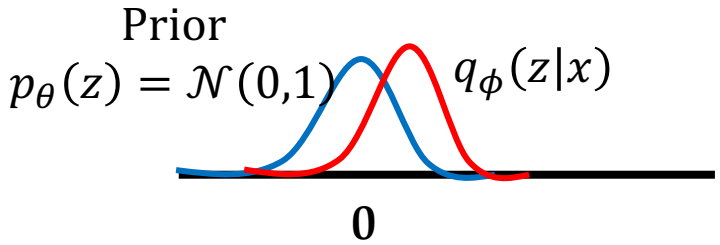


- Without regularization, network cheats by learning narrow distribution: encourage distribution to describe the input (**similar to autoencoder**)
- With small variance, this distribution is representing a single value
- Other issues include overfitting, poor latent space structure, and lack of diversity in generated samples

---

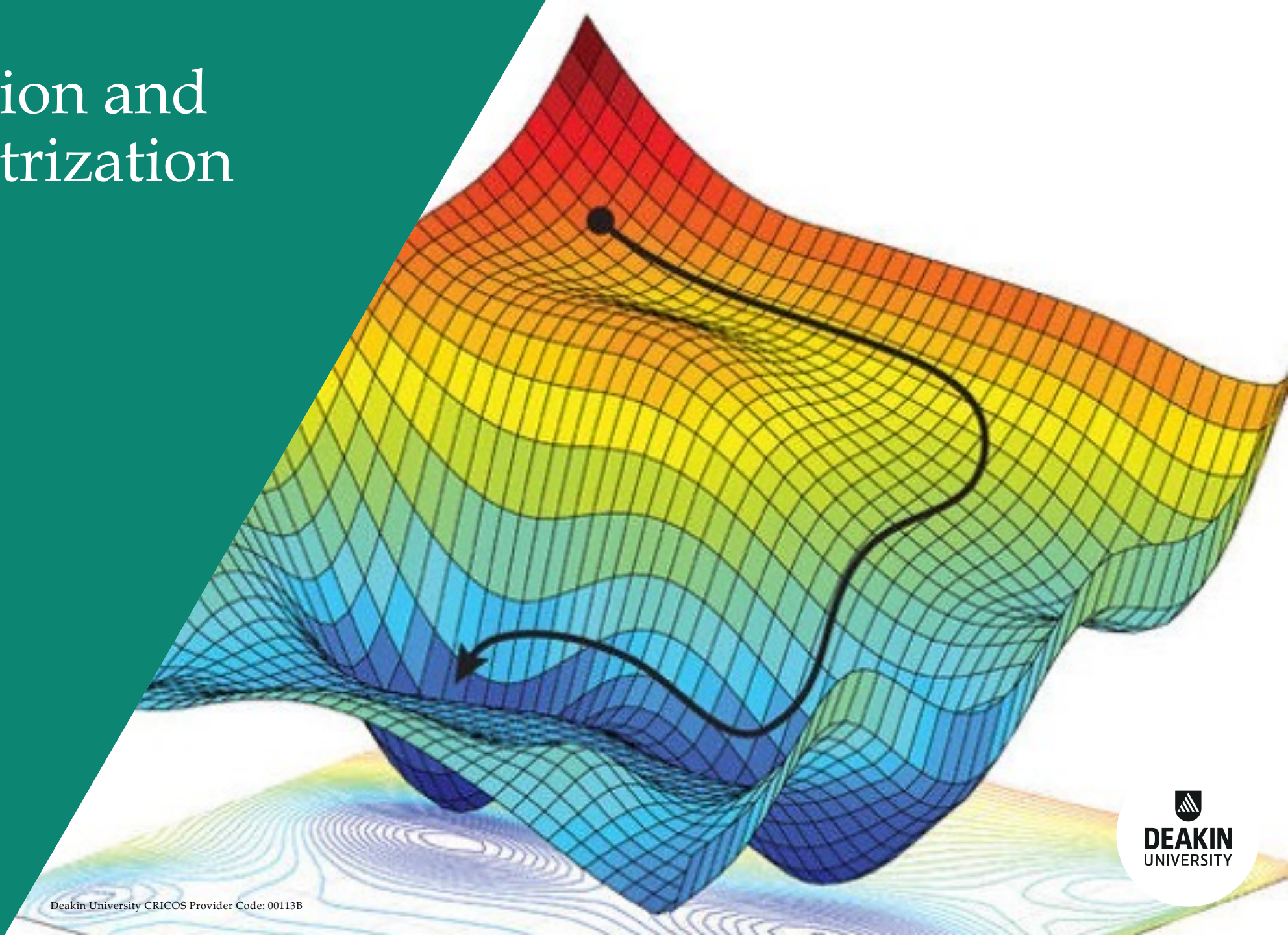


- Without regularization, data fidelity term, the encoder maps any input to a normal distribution and the network does not learn anything (we are not learning any characteristics of the input)



- Attraction between the two distribution is due to the KL div
- Sufficient variance is ensured using the KL div
- Promotes a smooth and well-structured latent space

Optimization and Reparametrization Trick



Optimization of the loss function VAE

- So, our target is to find optimal θ, ϕ such that

$$\theta^*, \phi^* = \underset{\theta,\phi}{\operatorname{argmin}}\, \mathcal{L}(\theta, \phi)$$

$$\mathcal{L}(\theta, \phi) = -\mathbb{E}_{\boldsymbol{z}}\left[\log\left(p_\theta(x|z)\right)\right] + D_{\boldsymbol{KL}}\left[q_\phi(z|x) \parallel p_\theta(z)\right]$$

$$= D_{KL}\left(q_\phi(z|x) \parallel p_\theta(z|x)\right) - \log\left(p_\theta(x)\right)$$

- In variational Bayesian method, this loss function is known as the variational lower bound or *"evidence lower bound (ELBO)"*.
- This "lower bound" part comes from the fact that KL divergence is always non-negative and thus $\mathcal{L}(\theta, \phi)$ is the lower bound of the log-likelihood of the data, i.e., $\log\left(p_\theta(x)\right)$.

$$\log\left(p_\theta(x)\right) = -\mathcal{L}(\theta, \phi) + D_{KL}\left(q_\phi(z|x) \parallel p_\theta(z|x)\right)$$

$$\log\left(p_\theta(x)\right) \geq -\mathcal{L}(\theta, \phi)$$

Evidence lower bound (ELBO)

- Hence, we have

$$\log\left(p_\theta(x)\right) \geq -\mathcal{L}(\theta, \phi)$$

- Therefore, minimizing the loss is equivalent to maximizing the lower bound of the probability of generating real data samples.
- Computing the exact log-likelihood of the data is often intractable due to the integration over all possible latent space values. Instead, by using variational inference techniques, we can derive a lower bound on the log-likelihood that is computationally feasible to optimize.

Optimizing the VAE loss function

- Recall the loss function of VAE:

$$\underset{\theta,\phi}{\operatorname{argmin}}\,\mathcal{L}(\theta,\phi) = \underset{\theta,\phi}{\operatorname{argmin}}\left\{-\mathbb{E}_{z\sim q_\phi(z|x)}\left[\log\left(p_\theta(x|z)\right)\right] + \frac{1}{2}\sum_k\left[\Sigma_\phi(x) + \mu_\phi(x)^2 - \log\left(\Sigma_\phi(x)\right) - 1\right]\right\}$$

- Identify θ^*, ϕ^* using the gradient descent algorithm:

Repeat until convergence {

✔ $\theta = \theta - \alpha\nabla_\theta\mathcal{L}(\theta,\phi)$ *# When calculating this derivative, ϕ is constant*

✖ $\phi = \phi - \alpha\nabla_\phi\mathcal{L}(\theta,\phi)$ *# When calculating this derivative, θ is constant*

} *# Problem occurs with this derivative that we*
 # solve using reparameterization trick

Optimizing the VAE loss function ($\nabla_\theta \mathcal{L}(\theta, \phi)$)

$$\nabla_\theta \mathcal{L}(\theta, \phi) = \nabla_\theta \left\{ -\mathbb{E}_{z \sim q_\phi(z|x)} \left[\log\left(p_\theta(x|z) \right) \right] + \frac{1}{2} \sum_k [\quad] \right\} = 0$$

Optimizing the VAE loss function ($\nabla_\phi \mathcal{L}(\theta, \phi)$)

$$\nabla_\phi \mathcal{L}(\theta, \phi) = \nabla_\phi \left\{ -\mathbb{E}_{z \sim q_\phi(z|x)}\left[\log\left(p_\theta(x|z)\right)\right] + \frac{1}{2}\sum_k \left[\Sigma_\phi(x) + \mu_\phi(x)^2 - \log\left(\Sigma_\phi(x)\right) - 1\right] \right\}$$

- The derivative $\nabla_\phi \mathcal{L}(\theta, \phi)$ **is harder** to estimate because ϕ appears in the distribution with respect to which the expectation is taken

$$\nabla_\phi \mathbb{E}_{q_\phi(z|x)}[f(z)] \neq \mathbb{E}_{q_\phi(z|x)}[\nabla_\phi f(z)]$$

Reparameterization trick

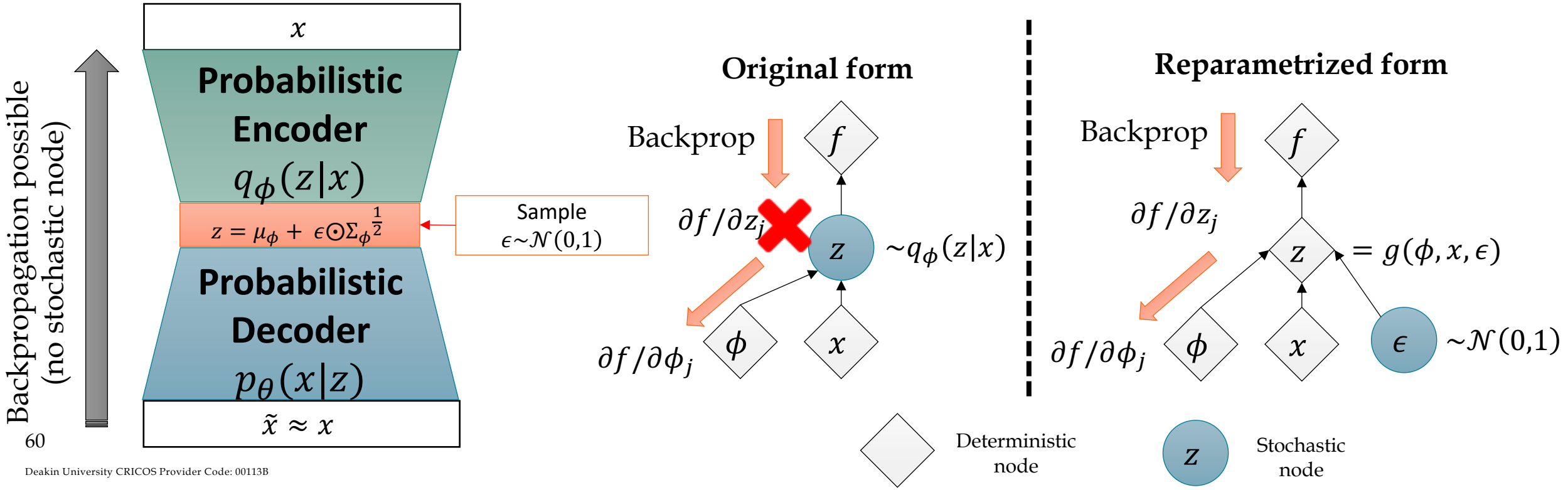
- If rewrite this expectation in such a way the ϕ appears inside the expectation, then we can push the gradient inside the expectation:

$$\mathbb{E}_{z\sim q_\phi(z|x)}[f(z)] = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,1)}\left[f(g_\phi(\epsilon, x))\right]$$

  - Such that z = $g_\phi(\epsilon, x)$, any linear transformation with $\epsilon\sim\mathcal{N}(0,1)$
  - In our case, $g_\phi(\epsilon, x) = \mu_\phi(x) + \epsilon\odot\Sigma_\phi(x)^{\frac{1}{2}} = z\sim\mathcal{N}\left(\mu_\phi(z), \Sigma_\phi(z)\right)$

Reparameterization trick

- Instead of sampling $z \sim q_\phi(z|x)$, we sample from $\epsilon \sim \mathcal{N}(0,1)$ and then we apply the linear transformation using $z = \mu_\phi(x) + \epsilon \odot \Sigma_\phi(x)^{\frac{1}{2}}$



Optimizing the VAE loss function ($\nabla_\phi \mathcal{L}(\theta, \phi)$)

$$\nabla_\phi \mathcal{L}(\theta, \phi) = \nabla_\phi \left\{ -\mathbb{E}_{z \sim q_\phi(z|x)} \left[\log\left(p_\theta(x|z)\right) \right] + \frac{1}{2} \sum_k \left[\Sigma_\phi(x) + \mu_\phi(x)^2 - \log\left(\Sigma_\phi(x)\right) - 1 \right] \right\}$$

$$= \nabla_\phi \left\{ -\mathbb{E}_{\epsilon \sim \mathcal{N}(0,1)} \left[\log\left(p_\theta\left(x \middle| g_\phi(\epsilon, z)\right)\right) \right] + \frac{1}{2} \sum_k \left[\Sigma_\phi(x) + \mu_\phi(x)^2 - \log\left(\Sigma_\phi(x)\right) - 1 \right] \right\}$$

$$= -\mathbb{E}_{\epsilon \sim \mathcal{N}(0,1)} \left[\nabla_\phi \log\left(p_\theta\left(x \middle| g_\phi(\epsilon, z)\right)\right) \right] + \nabla_\phi \left\{ \frac{1}{2} \sum_k \left[\Sigma_\phi(x) + \mu_\phi(x)^2 - \log\left(\Sigma_\phi(x)\right) - 1 \right] \right\}$$

Monte-carlo estimate of expectation

$$= -\frac{1}{S} \sum_{l=1}^{S} \nabla_\phi \log\left(p_\theta\left(x \middle| g_\phi(\epsilon, z)\right)\right) + \cdots$$

- Where $g_\phi(\epsilon, x) = \mu_\phi(x) + \epsilon \odot \Sigma_\phi(x)^{\frac{1}{2}} = z \sim \mathcal{N}\left(\mu_\phi(z), \Sigma_\phi(z)\right)$

VAE loss function

$$\mathcal{L}(\theta, \phi) = \frac{1}{L}\sum_{l=1}^{L}\frac{1}{2}\left(x - \mu_\theta\left(z^{[l]}\right)\right)^2 - \frac{1}{2}\sum_{k}\left[1 + \log\left(\Sigma_\phi(x)\right) - \mu_\phi(x)^2 - \Sigma_\phi(x)\right]$$

- Where $z^{[l]} \sim q_\phi(z|x)$ **(Monte Carlo Estimate)**
- **Most of the time the Monte Carlo Estimate consists into a single draw (the number of samples drawn is a hyperparameter of the VAE model and can vary depending on the complexity of the data and the desired accuracy)**

Demo:

https://keras.io/examples/generative/vae/

Questions?

Slides available on:
https://rbouadjenek.github.io/teaching.html