भारतीय प्रौद्योगिकी संस्थान दिल्ली
Indian Institute of Technology Delhi

COV877
Special Module on Visual Computing

Generative AI for Visual Content Creation: Image, Video, and 3D

VAE & VQVAE

Instructor:

Dr. Lokender Tiwari

Research Scientist

Variational Autoencoder (VAE)

Variational Autoencoders

- Probabilistic Generative Model (like normalizing flows)
- Learn distribution $Pr(\mathbf{x})$ over data
- Post training we can sample from this distribution

Latent Variable Model

- An indirect approach to represent probability distribution of data $Pr(\mathbf{x})$

$$Pr(\mathbf{x}) = \underbrace{\int \overbrace{\boxed{Pr(\mathbf{x}, \mathbf{z})}}^{\text{joint distribution of data \& unobserved latent variable}} d\mathbf{z}}_{\text{marginalization}}$$

conditional probability

$$Pr(\mathbf{x}) = \int \underbrace{Pr(\mathbf{x}|\mathbf{z})}_{\text{likelihood}} \overbrace{Pr(\mathbf{z})}^{\text{prior}} d\mathbf{z}$$

complex distributions $Pr(\mathbf{x})$ can be represented using simple $Pr(\mathbf{x}|\mathbf{z})$ and $Pr(\boldsymbol{z})$

e.g. Mixture of Gaussians

Latent Variable Model - Mixture of Gaussian Example

- $\mathbf{z}$ is a discrete latent variable with categorial prior distribution $Pr(\mathbf{z})$

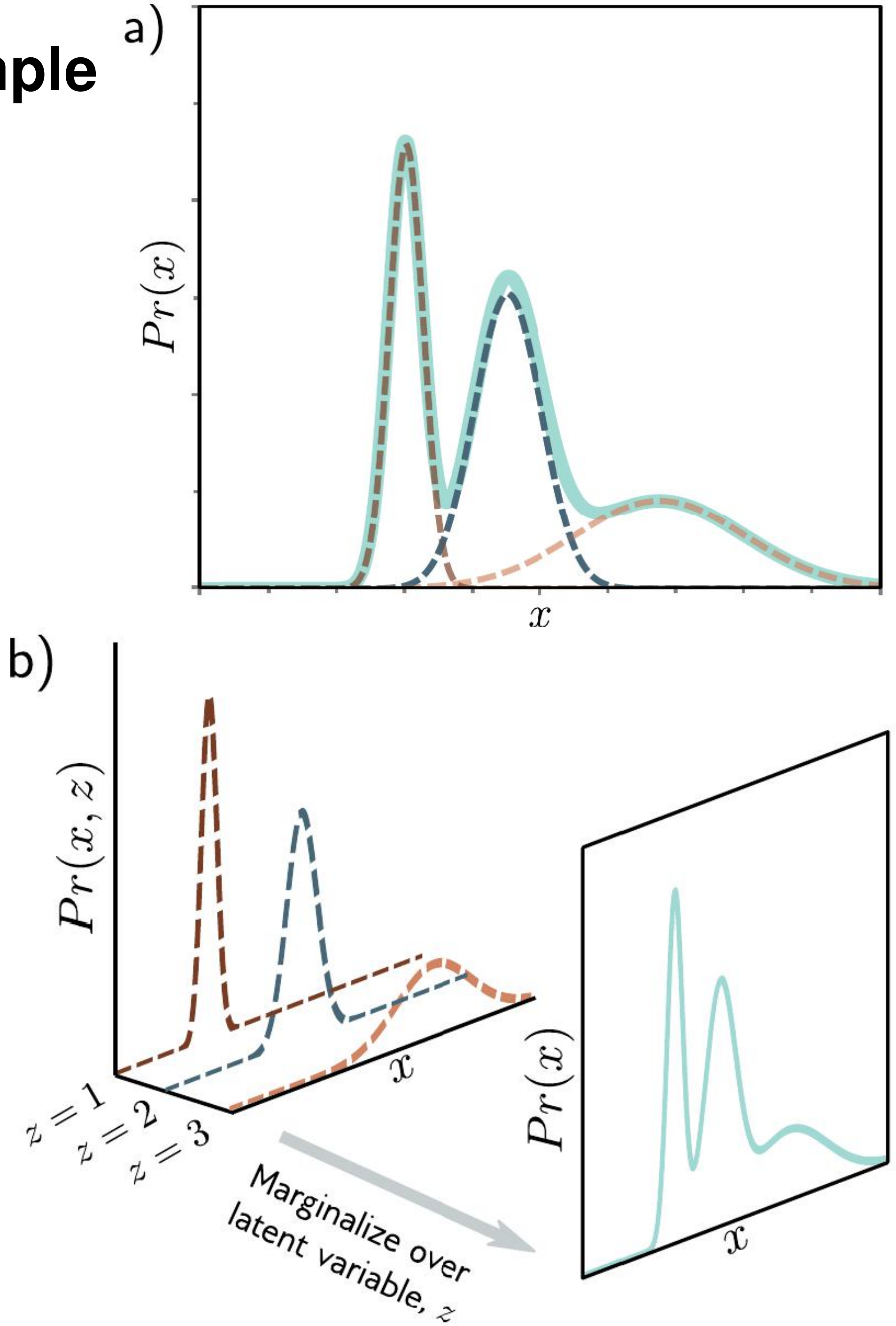
$$Pr(z=n) \quad = \quad \lambda_n$$

- Likelihood $Pr(\mathbf{x}|\mathbf{z}=\mathbf{n})$ of data $\mathbf{x}$ given $\mathbf{z}=\mathbf{n}$ is normally distributed with mean μ_n and variance σ_n^2

$$Pr(x|z=n) \quad = \quad \text{Norm}_x\left[\mu_n, \sigma_n^2\right]$$

$$\begin{aligned} Pr(x) &= \sum_{n=1}^{N} Pr(x, z=n) \\ &= \sum_{n=1}^{N} Pr(x|z=n) \cdot Pr(z=n) \\ &= \sum_{n=1}^{N} \lambda_n \cdot \text{Norm}_x\left[\mu_n, \sigma_n^2\right]. \end{aligned}$$

Marginalization



Nonlinear Latent Variable Model

Both data $\mathbf{x}$ and latent variable $\mathbf{z}$ are continous and multivariate

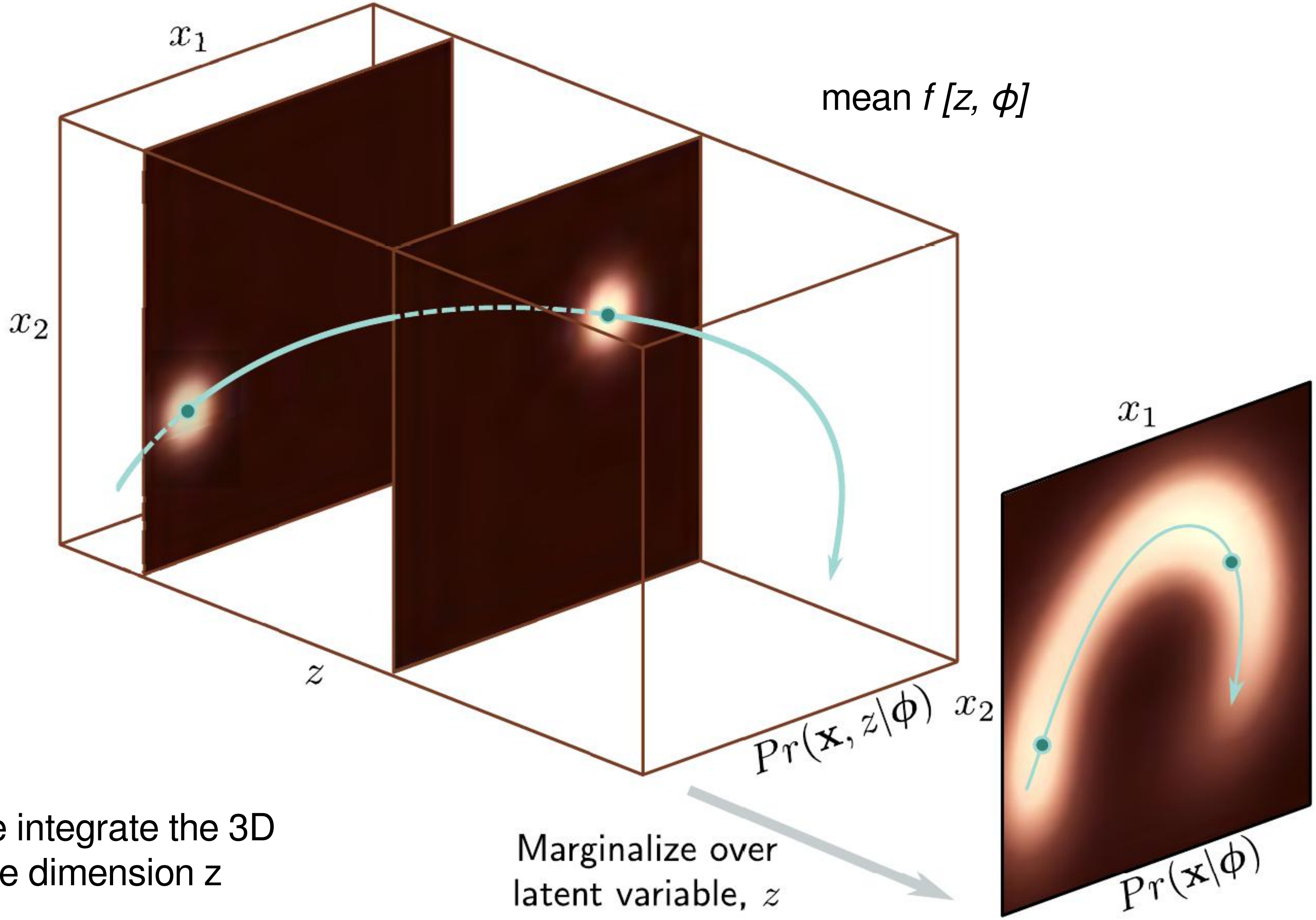
$$Pr(\mathbf{z}) = \text{Norm}_{\mathbf{z}}[\mathbf{0}, \mathbf{I}]$$

likelihood $$Pr(\mathbf{x}|\mathbf{z}, \phi) = \text{Norm}_{\mathbf{x}}\left[\mathbf{f}[\mathbf{z}, \phi], \sigma^2\mathbf{I}\right]$$ mean is a nonlinear function of the latent variable

$$\begin{aligned}
Pr(\mathbf{x}|\phi) &= \int Pr(\mathbf{x}, \mathbf{z}|\phi) d\mathbf{z} \\
&= \int Pr(\mathbf{x}|\mathbf{z}, \phi) \cdot Pr(\mathbf{z}) d\mathbf{z} \\
&= \int \text{Norm}_{\mathbf{x}}\left[\mathbf{f}[\mathbf{z}, \phi], \sigma^2\mathbf{I}\right] \cdot \text{Norm}_{\mathbf{z}}\left[\mathbf{0}, \mathbf{I}\right] d\mathbf{z}
\end{aligned}$$

Nonlinear Latent Variable Model

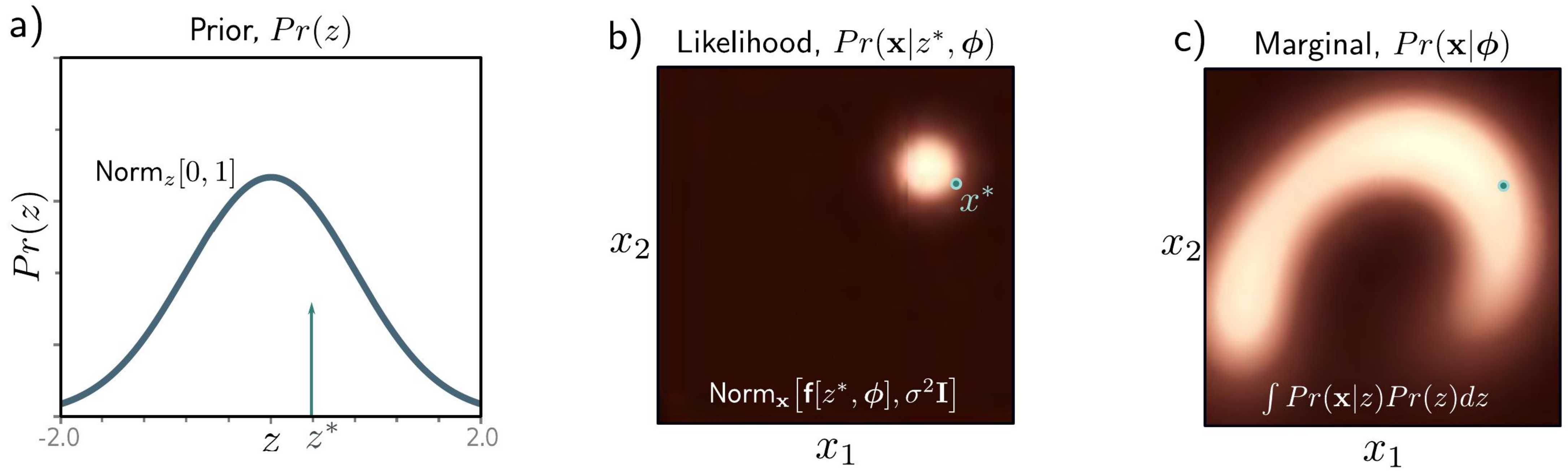
- Both data $\mathbf{x}$ and latent variable $\mathbf{z}$ are continous and multivariate



to get $Pr(x)$, we integrate the 3D volume over the dimension z

Nonlinear Latent Variable Model - Generation from nonlinear latent variable model

- Both data $\mathbf{x}$ and latent variable $\mathbf{z}$ are continous and multivariate



Nonlinear Latent Variable Model - Training

- Maximize the log-likelihood w.r.t the model parameters over training set

- For simplicity assume the variance term is known

- Goal is to learn Φ

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \left[\sum_{i=1}^{I} \log \left[Pr(\mathbf{x}_i | \boldsymbol{\phi}) \right] \right]$$

$$Pr(\mathbf{x}_i | \boldsymbol{\phi}) = \int \operatorname{Norm}_{\mathbf{x}_i} [\mathbf{f}[\mathbf{z}, \boldsymbol{\phi}], \sigma^2 \mathbf{I}] \cdot \operatorname{Norm}_{\mathbf{z}} [\mathbf{0}, \mathbf{I}] d\mathbf{z}$$

- Unfortunately, this is intractable.
- There is no closed-form expression for the integral and no easy way to evaluate it for a particular value of x

Nonlinear Latent Variable Model - ELBO

$$\hat{\phi} = \underset{\phi}{\operatorname{argmax}} \left[\sum_{i=1}^{I} \log\left[Pr(\mathbf{x}_i|\phi) \right] \right]$$

- Define lower bound on the log-likelihood
- a function which is always less than or equal to the log-likelihood for a given Φ and other parameters
- ELBO (Evidence Lower Bound)

Nonlinear Latent Variable Model - ELBO

- ELBO (Evidence Lower Bound) - Derivation

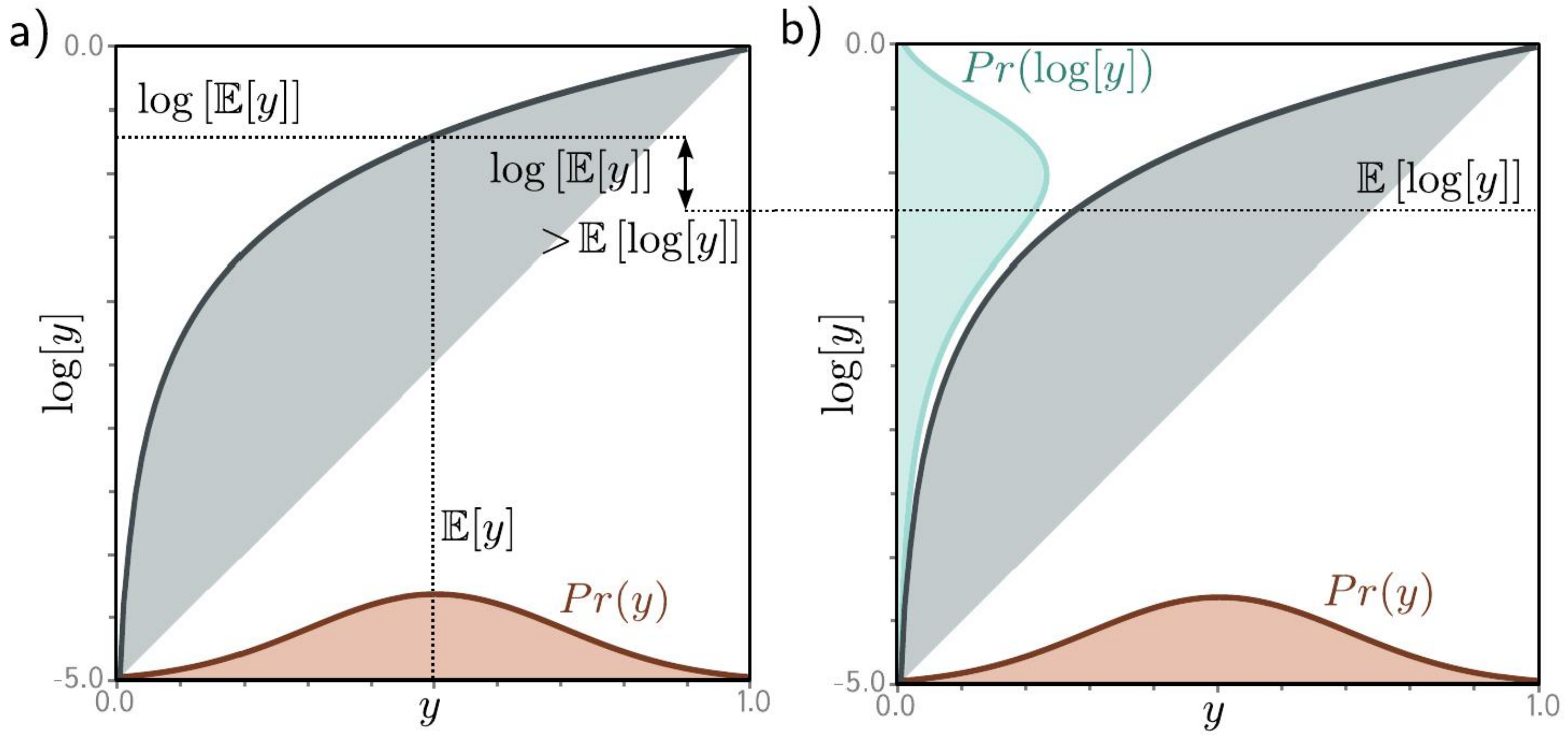
$$g[\mathbb{E}[y]] \geq \mathbb{E}\big[g[y]\big]$$

$$\log\big[\mathbb{E}[y]\big] \geq \mathbb{E}\big[\log[y]\big]$$

Jensen Inequality
(Concave Function)

$$\log\left[\int Pr(y)h[y]dy\right] \geq \int Pr(y)\log[h[y]]dy$$

Expansion



Nonlinear Latent Variable Model - ELBO

- ELBO (Evidence Lower Bound) - Derivation

$$\begin{aligned}\log[Pr(\mathbf{x}|\boldsymbol{\phi})] &= \log\left[\int Pr(\mathbf{x},\mathbf{z}|\boldsymbol{\phi})d\mathbf{z}\right] \\ &= \log\left[\int q(\mathbf{z})\frac{Pr(\mathbf{x},\mathbf{z}|\boldsymbol{\phi})}{q(\mathbf{z})}d\mathbf{z}\right]\end{aligned}$$

multiply & divide by arbitrary probability distribution over latent i.e., *q*(z)

$$\log\left[\int q(\mathbf{z})\frac{Pr(\mathbf{x},\mathbf{z}|\boldsymbol{\phi})}{q(\mathbf{z})}d\mathbf{z}\right] \geq \underbrace{\int q(\mathbf{z})\log\left[\frac{Pr(\mathbf{x},\mathbf{z}|\boldsymbol{\phi})}{q(\mathbf{z})}\right]d\mathbf{z}}_{\text{Evidence lower bound}}$$

using Jensen's inequality

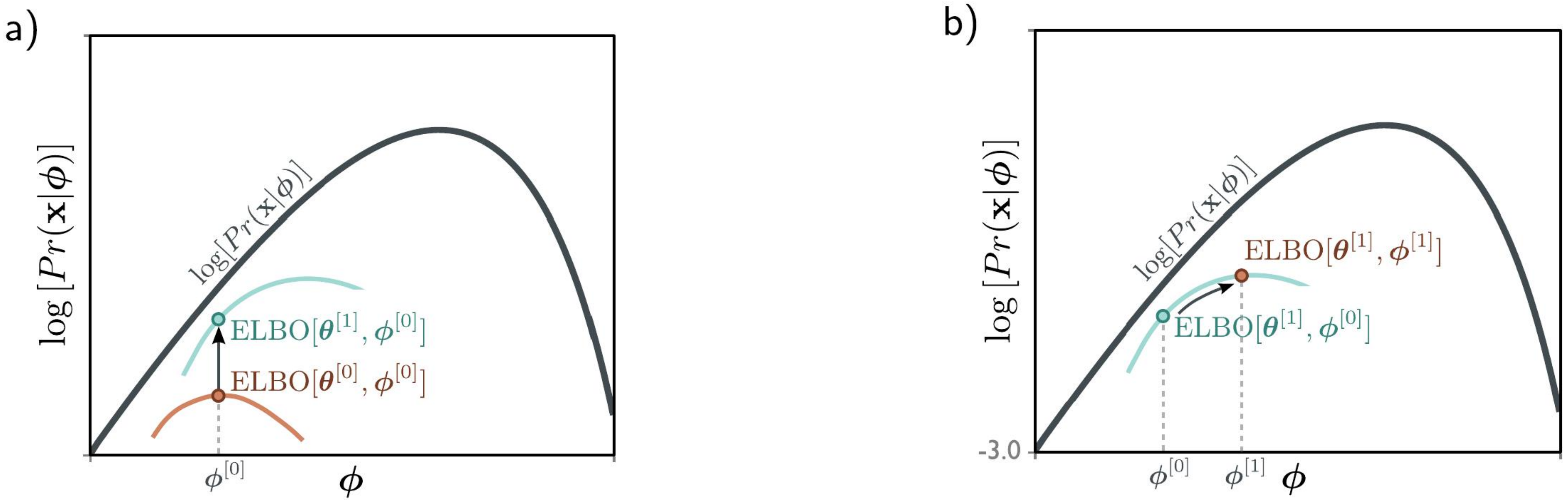
- In practice $q(\mathbf{z})$ is parameterized by $\boldsymbol{\theta}$

$$\text{ELBO}[\boldsymbol{\theta},\boldsymbol{\phi}] = \int q(\mathbf{z}|\boldsymbol{\theta})\log\left[\frac{Pr(\mathbf{x},\mathbf{z}|\boldsymbol{\phi})}{q(\mathbf{z}|\boldsymbol{\theta})}\right]d\mathbf{z}$$

maximize w.r.t both $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$

Nonlinear Latent Variable Model - **ELBO**

$$\hat{\phi} = \underset{\phi}{\operatorname{argmax}} \left[\sum_{i=1}^{I} \log\left[Pr(\mathbf{x}_i|\phi) \right] \right]$$



- **Objective** : maximize the log-likelihood (black curve) with respect to the parameters ϕ
- For fixed θ, we get a function of ϕ (two colored curves for different values of θ)
- The log-likelihood can be increased by either ways

Nonlinear Latent Variable Model - ELBO

- **Tight ELBO** - ELBO concide with the likelihood function
  - for a fixed $\boldsymbol{\Phi}$

$$\begin{aligned}
\text{ELBO}[\boldsymbol{\theta}, \phi] &= \int q(\mathbf{z}|\boldsymbol{\theta}) \log\left[\frac{Pr(\mathbf{x}, \mathbf{z}|\phi)}{q(\mathbf{z}|\boldsymbol{\theta})}\right] d\mathbf{z} \\
&= \int q(\mathbf{z}|\boldsymbol{\theta}) \log\left[\frac{Pr(\mathbf{z}|\mathbf{x}, \phi)Pr(\mathbf{x}|\phi)}{q(\mathbf{z}|\boldsymbol{\theta})}\right] d\mathbf{z} \\
&= \int q(\mathbf{z}|\boldsymbol{\theta}) \log\big[Pr(\mathbf{x}|\phi)\big] d\mathbf{z} + \int q(\mathbf{z}|\boldsymbol{\theta}) \log\left[\frac{Pr(\mathbf{z}|\mathbf{x}, \phi)}{q(\mathbf{z}|\boldsymbol{\theta})}\right] d\mathbf{z} \\
&= \log\big[Pr(\mathbf{x}|\phi)\big] + \int q(\mathbf{z}|\boldsymbol{\theta}) \log\left[\frac{Pr(\mathbf{z}|\mathbf{x}, \phi)}{q(\mathbf{z}|\boldsymbol{\theta})}\right] d\mathbf{z} \\
&= \log\big[Pr(\mathbf{x}|\phi)\big] - \mathrm{D}_{KL}\Big[q(\mathbf{z}|\boldsymbol{\theta}) \Big|\Big| \boxed{Pr(\mathbf{z}|\mathbf{x}, \phi)}\Big].
\end{aligned}$$

conditional probability

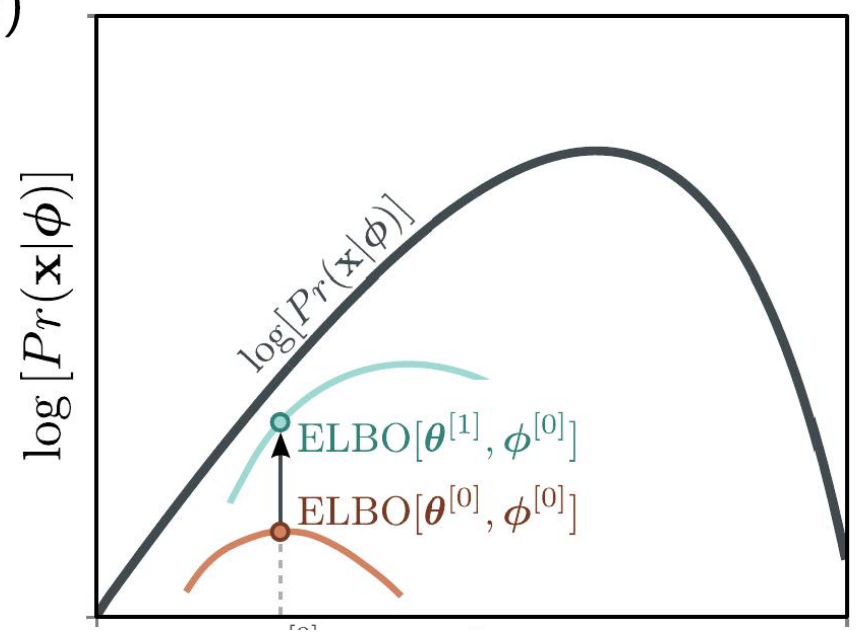
posterior distribution

Kullback-Leibler (KL) Divergence (Measures the distance between two distributions)

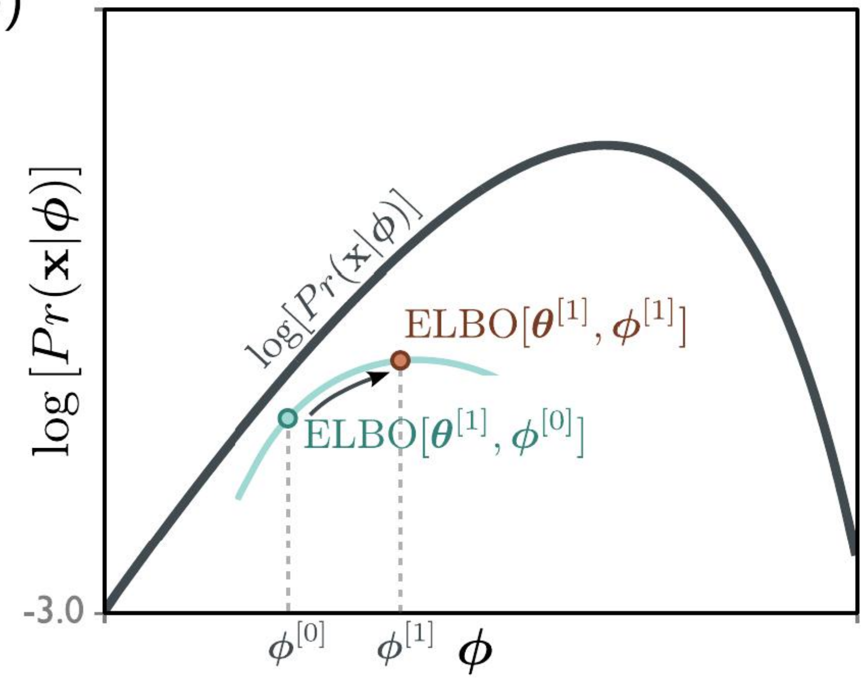
the bound is tight, when KL distance is zero i.e.,

$$q(\mathbf{z}|\boldsymbol{\theta}) = Pr(\mathbf{z}|\mathbf{x}, \phi)$$

a)



b)



Nonlinear Latent Variable Model - **ELBO**

- ELBO as reconstruction loss - KL distance to prior

$$
\begin{aligned}
\mathrm{ELBO}[\boldsymbol{\theta}, \boldsymbol{\phi}] &= \int q(\mathbf{z}|\boldsymbol{\theta}) \log\left[\frac{Pr(\mathbf{x},\mathbf{z}|\boldsymbol{\phi})}{q(\mathbf{z}|\boldsymbol{\theta})}\right] d\mathbf{z} \\
&= \int q(\mathbf{z}|\boldsymbol{\theta}) \log\left[\frac{Pr(\mathbf{x}|\mathbf{z},\boldsymbol{\phi})Pr(\mathbf{z})}{q(\mathbf{z}|\boldsymbol{\theta})}\right] d\mathbf{z} \\
&= \int q(\mathbf{z}|\boldsymbol{\theta}) \log\left[Pr(\mathbf{x}|\mathbf{z},\boldsymbol{\phi})\right] d\mathbf{z} + \int q(\mathbf{z}|\boldsymbol{\theta}) \log\left[\frac{Pr(\mathbf{z})}{q(\mathbf{z}|\boldsymbol{\theta})}\right] d\mathbf{z} \\
&= \int q(\mathbf{z}|\boldsymbol{\theta}) \log\left[\boxed{Pr(\mathbf{x}|\mathbf{z},\boldsymbol{\phi})}\right] d\mathbf{z} - \underbrace{\mathrm{D}_{KL}\left[q(\mathbf{z}|\boldsymbol{\theta}) \middle|\middle| Pr(\mathbf{z})\right],}
\end{aligned}
$$

conditional probability

reconstruction accuracy

degree of agreement between auxiliary distribution and prior

This formulation is generally used in the **variational autoencoder**

Nonlinear Latent Variable Model - ELBO

- Tight ELBO - ELBO concide with the likelihood function
  - for a fixed ***Φ***

$$q(\mathbf{z}|\boldsymbol{\theta}) = Pr(\mathbf{z}|\mathbf{x}, \phi)$$

$$Pr(\mathbf{z}|\mathbf{x}, \phi) = \frac{Pr(\mathbf{x}|\mathbf{z}, \phi)Pr(\mathbf{z})}{\boxed{Pr(\mathbf{x}|\phi)}}$$

Baye’s rule

intractable, we can’t evaluate this

Solution ?

multivariate normal distribution mean *μ* and diagonal variance *Σ*

- **Solution** : make a variational approximation
- Choose a **simple auxiliary distribution** $\boxed{q(\mathbf{z} \mid \theta)}$ to approximate **true posterior**

during training the goal is to find values of *μ* and *Σ*
such that the normal distrubution is close to the true posterior *Pr(***z** | **x***)*

= minimize the KL divergence

$$q(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) = \mathrm{Norm}_{\mathbf{z}}\left[\underbrace{\mathbf{g}_{\boldsymbol{\mu}}[\mathbf{x}, \boldsymbol{\theta}], \mathbf{g}_{\boldsymbol{\Sigma}}[\mathbf{x}, \boldsymbol{\theta}]}\right]$$

Neural network with params **θ**

The Variational Autoencoder

$$\mathrm{ELBO}[\boldsymbol{\theta}, \boldsymbol{\phi}] = \underbrace{\int q(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) \log\left[Pr(\mathbf{x}|\mathbf{z}, \boldsymbol{\phi})\right] d\mathbf{z}}_{\text{Intractable Integral}} - \mathrm{D}_{KL}\left[q(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) \middle|\middle| Pr(\mathbf{z})\right]$$

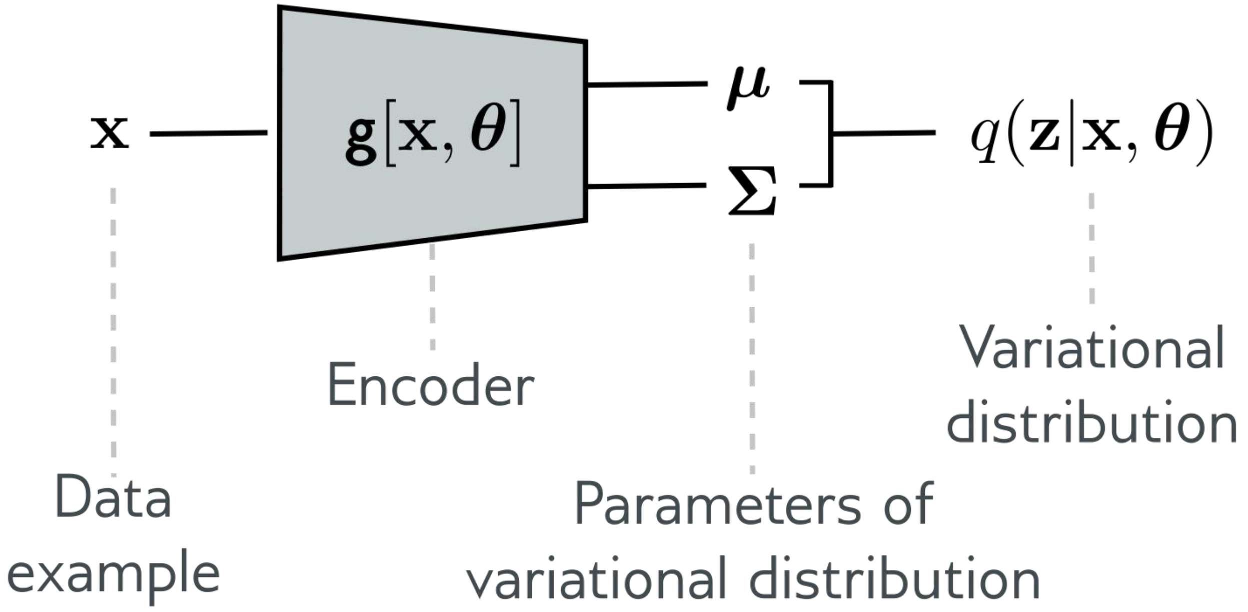
Solution : approximate it by sampling i.e., take a Monte Carlo estimate

$$\mathrm{ELBO}[\boldsymbol{\theta}, \boldsymbol{\phi}] \approx \log\left[Pr(\mathbf{x}|\mathbf{z}^*, \boldsymbol{\phi})\right] - \mathrm{D}_{KL}\left[q(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) \middle|\middle| Pr(\mathbf{z})\right]$$

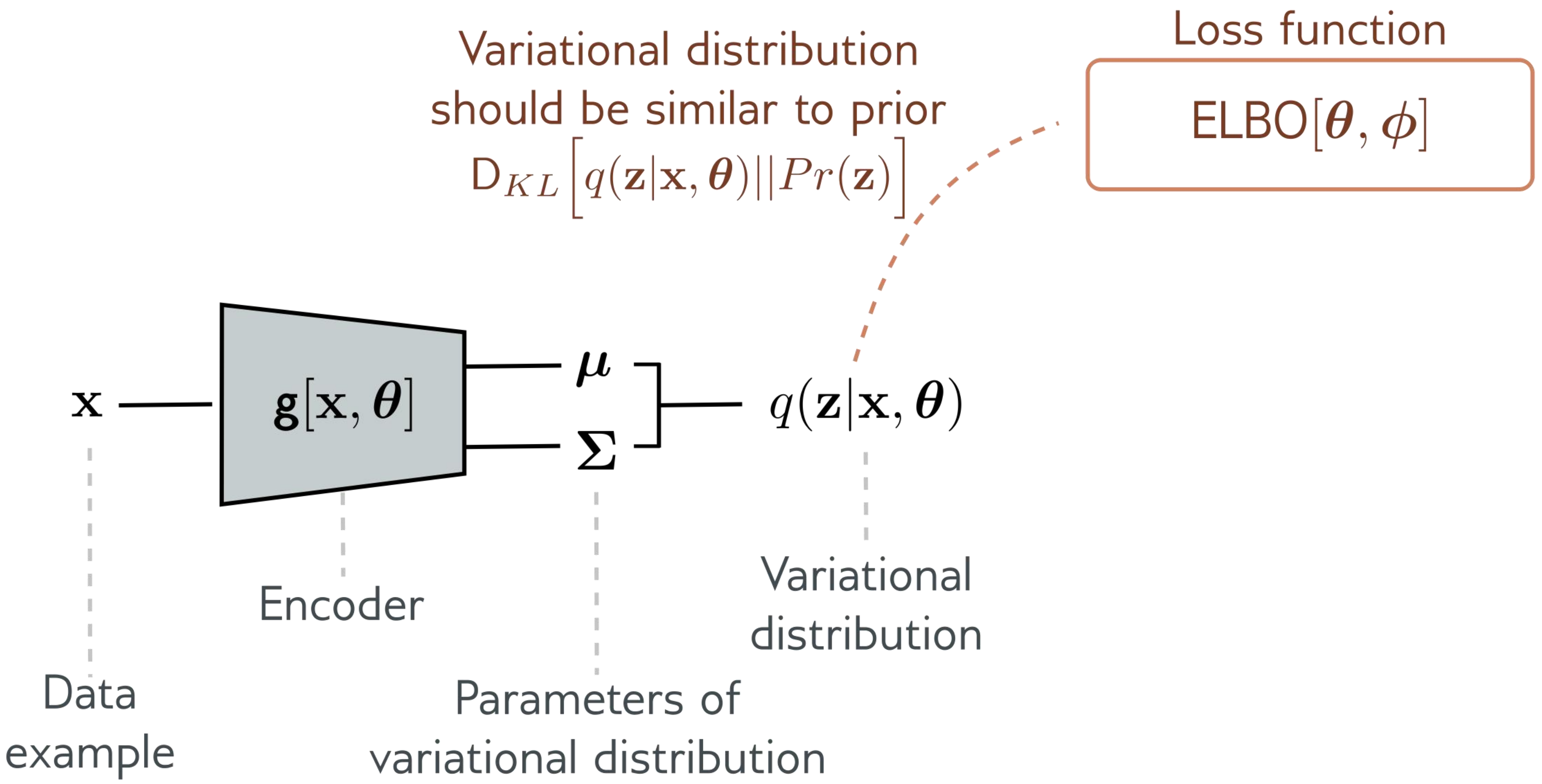
$$\mathrm{D}_{KL}\left[q(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) \middle|\middle| Pr(\mathbf{z})\right] = \frac{1}{2}\left(\mathrm{Tr}[\boldsymbol{\Sigma}] + \boldsymbol{\mu}^T\boldsymbol{\mu} - \boxed{D_{\mathbf{z}}} - \log\left[\det[\boldsymbol{\Sigma}]\right]\right)$$

Dimensionality of the latent space

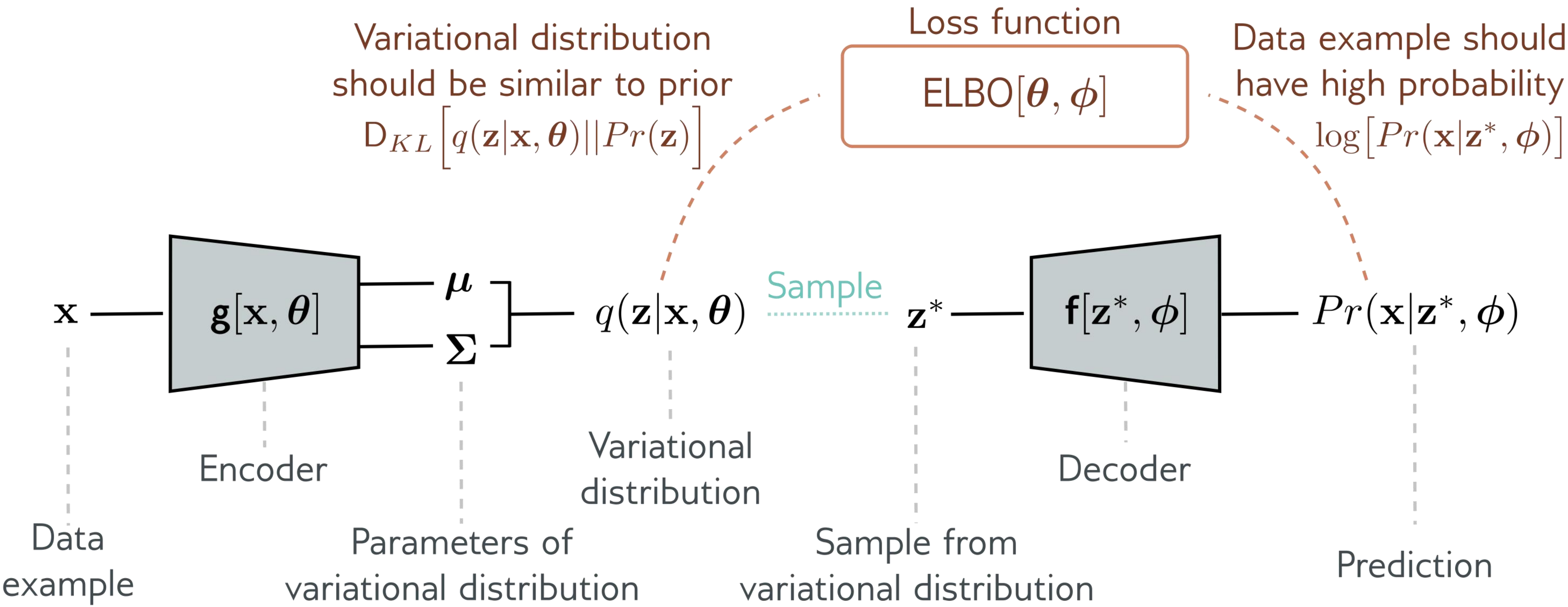
The Variational Autoencoder



The Variational Autoencoder



The Variational Autoencoder



Vector Quantised
Variational Autoencoder (VAE)

The VQVAE - Vector Quantized VAE

The Vector Quantised- Variational AutoEncoder (VQ-VAE), differs from VAEs in two key ways:

- the encoder network outputs discrete, rather than continuous codes
- the prior is learnt rather than static

Neural Discrete Representation Learning

Aaron van den Oord
DeepMind
avdnoord@google.com

Oriol Vinyals
DeepMind
vinyals@google.com

Koray Kavukcuoglu
DeepMind
korayk@google.com

Van Den Oord, Aaron, and Oriol Vinyals. "Neural discrete representation learning." Advances in neural information processing systems 30 (2017).

The VQVAE - Vector Quantized VAE

Key Idea :

- Incorporate ideas from vector quantisation (VQ)
- VQ method allows model to circumvent issues of “posterior collapse”
- where the latents are ignored when they are paired with a powerful autoregressive decoder =
- typically observed in the VAE framework.

VAE setup

- Posterior Distribution $q(z|x)$
- Prior Distribution $p(z)$
- Decoder with distribution $p(x|z)$

Posterior and prior are assumed normally distributed

Van Den Oord, Aaron, and Oriol Vinyals. "Neural discrete representation learning." Advances in neural information processing systems 30 (2017).

The VQVAE - Vector Quantized VAE

VQ-VAE:

- Discrete latent variables
- New way of training leveraging the concepts from vector quantisation (VQ)
- Posterior and prior distributions are categorical

Van Den Oord, Aaron, and Oriol Vinyals. "Neural discrete representation learning." Advances in neural information processing systems 30 (2017).

The VQVAE - Formulation

Discrete Latent Variables

Define a latent embedding space $e \in R^{K \times D}$
K : Size of the discrete latent space (K-way categorical)

D is the dimensionality of each latent embedding vector

$$e_i \in R^D, i \in 1, 2, ..., K$$

The encoder takes an input and produce output $z_e(x)$

Discrete latent variable is then calculated by a nearest neighbour look-up using a shared embedding space

$$q(z = k|x) = \begin{cases} 1 & \text{for } \mathrm{k} = \mathrm{argmin}_j \|z_e(x) - e_j\|_2 \\ 0 & \text{otherwise} \end{cases}$$

Van Den Oord, Aaron, and Oriol Vinyals. "Neural discrete representation learning." Advances in neural information processing systems 30 (2017).

The VQVAE - Formulation

the posterior is a categorical distribution as one-hot

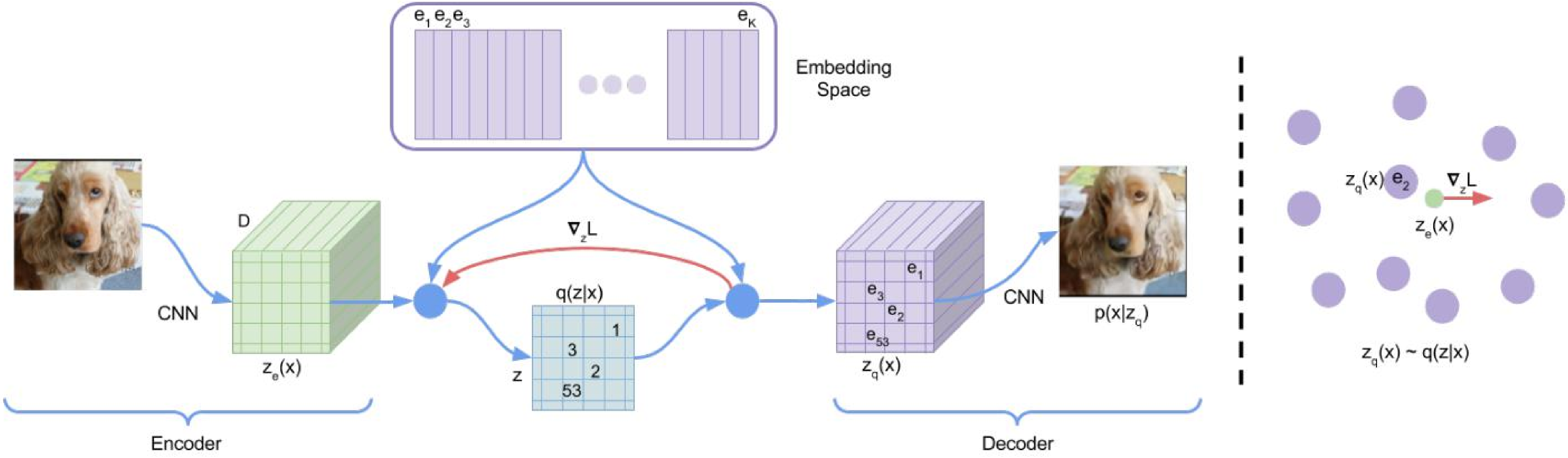
$$q(z = k|x) = \begin{cases} 1 & \text{for } \ \mathrm{k} = \mathrm{argmin}_j \|z_e(x) - e_j\|_2 \\ 0 & \text{otherwise} \end{cases}$$

Decoder input is the embedding vector calculated from NN search

$$z_q(x) = e_k, \quad \text{where} \quad k = \mathrm{argmin}_j \|z_e(x) - e_j\|_2$$

Van Den Oord, Aaron, and Oriol Vinyals. "Neural discrete representation learning." Advances in neural information processing systems 30 (2017).

The VQVAE - Training



Van Den Oord, Aaron, and Oriol Vinyals. "Neural discrete representation learning." Advances in neural information processing systems 30 (2017).

The VQVAE - Training

There is no real gradient defined for the below equation,

$$z_q(x) = e_k, \quad \text{where} \quad k = \text{argmin}_j \|z_e(x) - e_j\|_2$$

however we approximate the gradient similar to the straight-through estimator

- copy gradients from decoder input to encoder output .

$$z_q(x) \qquad\qquad z_e(x)$$

- During forward computation the nearest embedding is passed to the decoder,

- During the backwards pass the gradient is passed unaltered to the encoder

Will this work ? Why?

- the output representation of the encoder and the input to the decoder share the same D dimensional space,

- the gradients contain useful information for how the encoder has to change its output to lower the reconstruction loss.

Van Den Oord, Aaron, and Oriol Vinyals. "Neural discrete representation learning." Advances in neural information processing systems 30 (2017).

The VQVAE - Training

- the gradient must push the encoder's output to be discretised differently in the next forward pass, because the assignment will be different

- Total Loss

$$L = \log p(x|z_q(x)) + \|\text{sg}[z_e(x)] - e\|_2^2 + \beta\|z_e(x) - \text{sg}[e]\|_2^2$$

Van Den Oord, Aaron, and Oriol Vinyals. "Neural discrete representation learning." Advances in neural information processing systems 30 (2017).

The VQVAE - Training

- the gradient must push the encoder's output to be discretised differently in the next forward pass, because the assignment will be different

- Total Loss

reconstruction loss
Due to the straight-through gradient estimation of mapping from $z_e(x)$ to $z_q(x)$, the embeddings e_i receive no gradients from the reconstruction loss

$$L = \boxed{\log p(x|z_q(x))} + \boxed{\|\text{sg}[z_e(x)] - e\|_2^2} + \boxed{\beta \|z_e(x) - \text{sg}[e]\|_2^2}$$

VQ objective uses the l_2 error to move the embedding vectors e_i towards the encoder outputs $z_e(x)$

Commitment Loss
Volumne of the embedding space can grow arbitrarily. To make sure the encoder commits to an embedding and its output does not grow.

Van Den Oord, Aaron, and Oriol Vinyals. "Neural discrete representation learning." Advances in neural information processing systems 30 (2017).

The VQVAE - Results

Left: ImageNet 128x128x3 images, right: reconstructions from a VQ-VAE with a 32x32x1 latent space, with K=512.

Van Den Oord, Aaron, and Oriol Vinyals. "Neural discrete representation learning." Advances in neural information processing systems 30 (2017).