

Lecture 2

Variational Autoencoder

6.S978 Deep Generative Models

Kaiming He
EECS, MIT



Overview

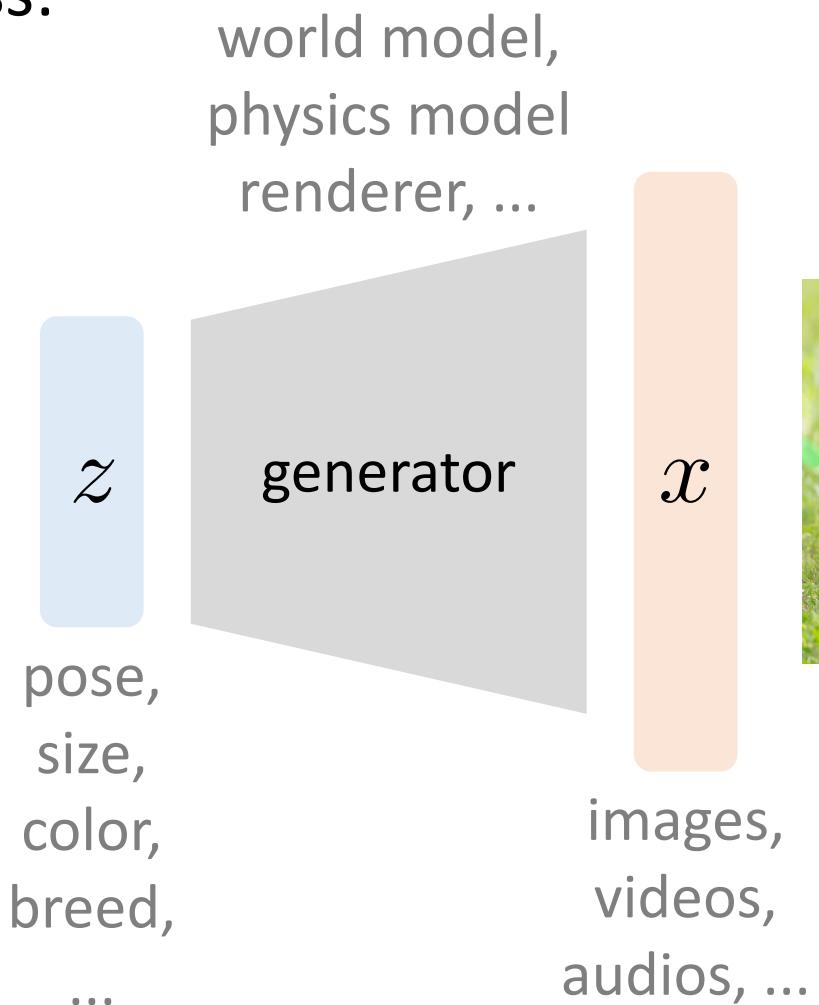
- Variational Autoencoder (VAE)
- Relation to Expectation-Maximization (EM)
- Vector Quantized VAE (VQ-VAE)

Variational Autoencoder (VAE)

Latent Variable Models

Assuming a data generation process:

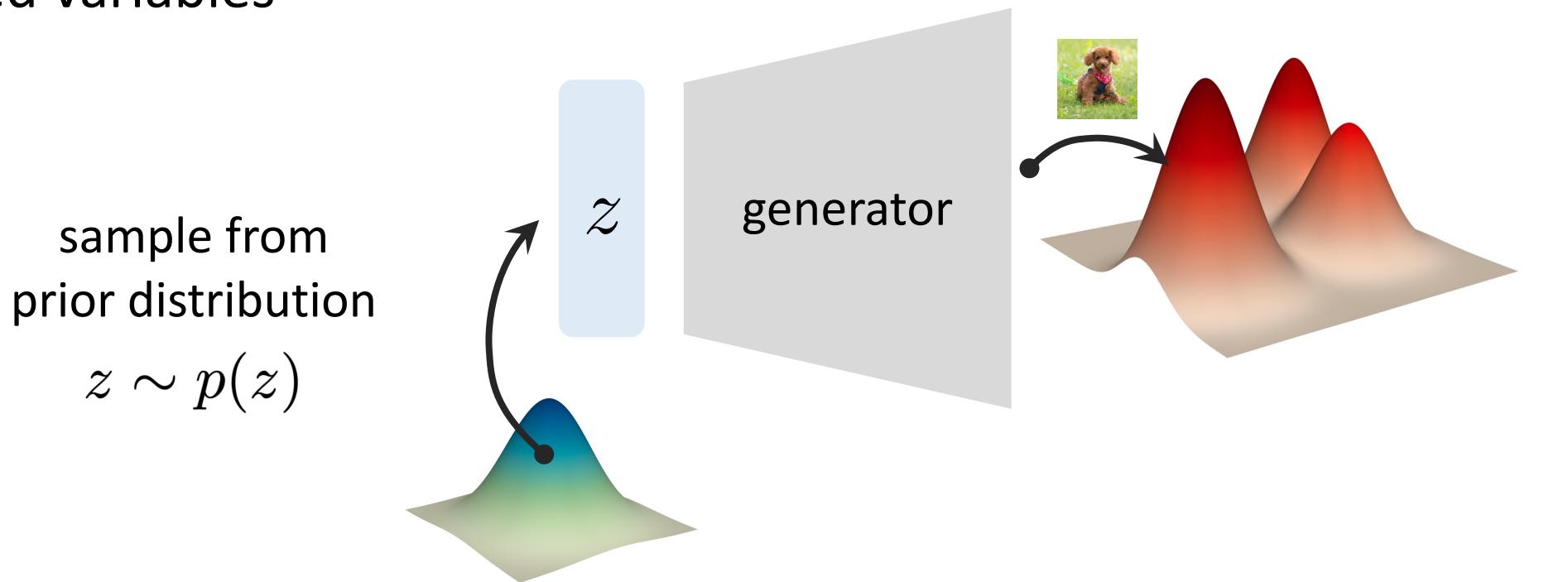
- z - latent variables
- x - observed variables



Latent Variable Models

Assuming a data generation process:

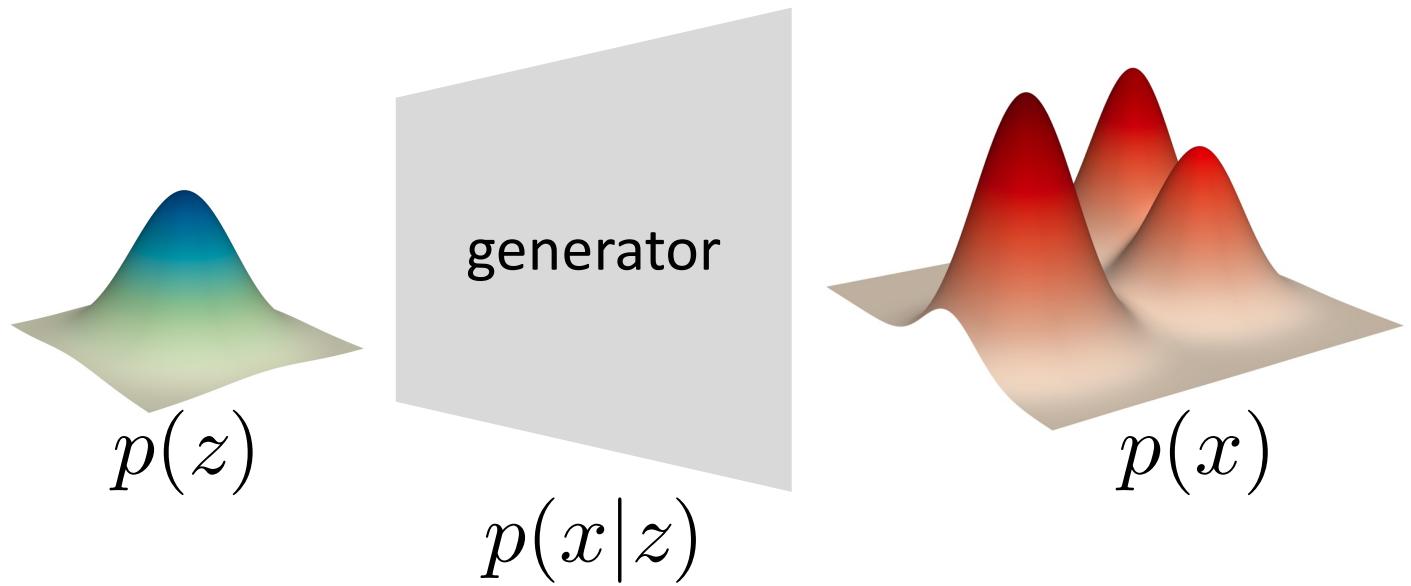
- z - latent variables
- x - observed variables



Latent Variable Models

Assuming a data generation process:

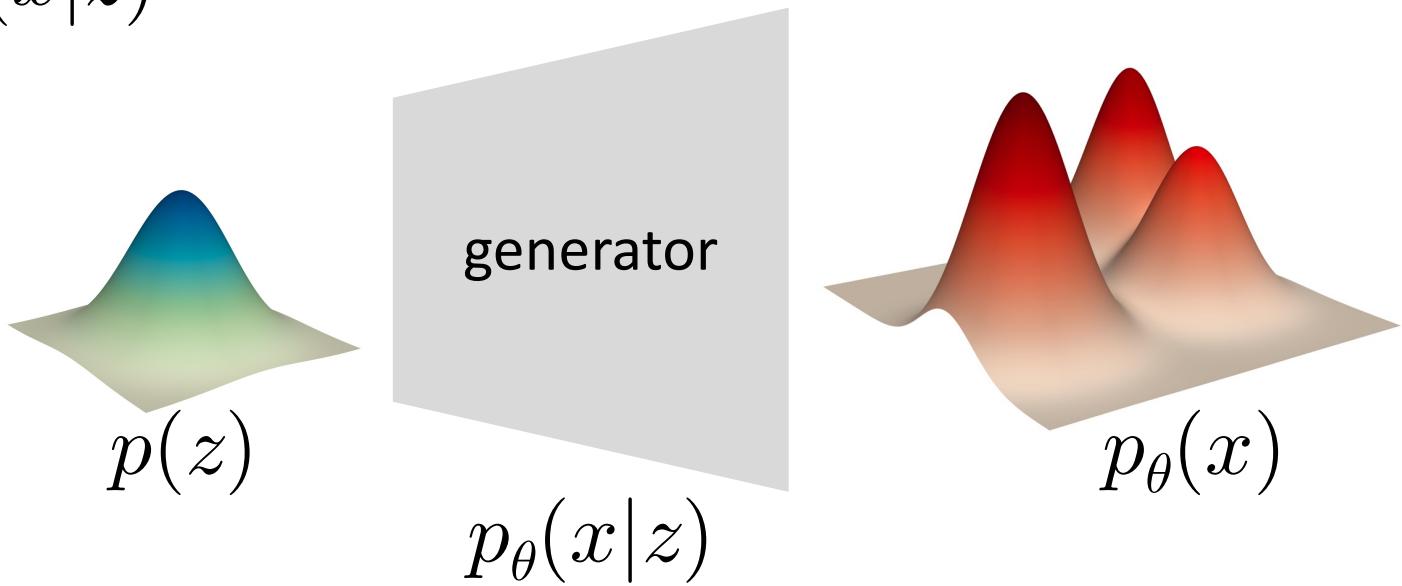
- z - latent variables
- x - observed variables



Latent Variable Models

Represent a distribution by a neural network

- θ - learnable parameters
- represent a function: $p_\theta(x|z)$



Measuring how good a distribution is ...

Minimize Kullback–Leibler (KL) divergence:

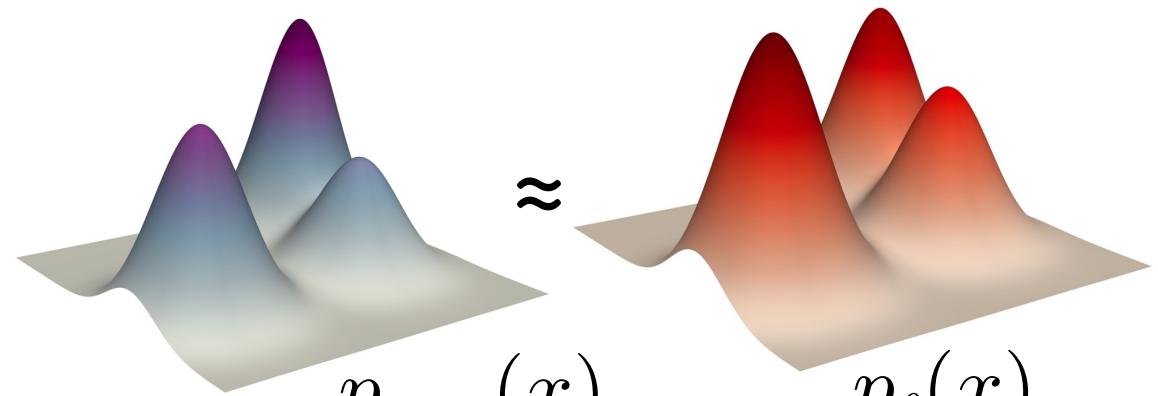
$$\min_{\theta} \mathcal{D}_{\text{KL}}(p_{\text{data}} \parallel p_{\theta})$$

Note: consider other criteria than KL?

⇒ Maximize likelihood:

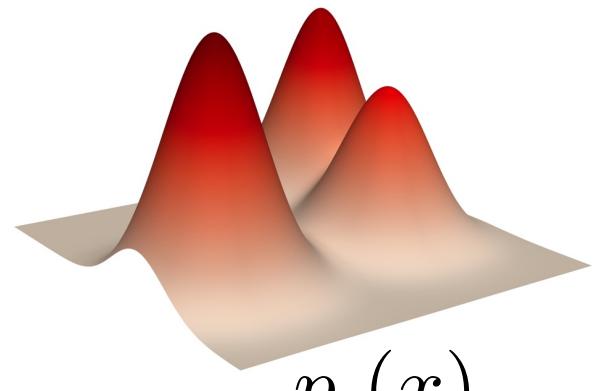
$$\max_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\theta}(x)$$

$$\begin{aligned} & \arg \min_{\theta} \mathcal{D}_{\text{KL}}(p_{\text{data}} \parallel p_{\theta}) \quad \text{tl; dr} \\ = & \arg \min_{\theta} \sum_x p_{\text{data}}(x) \log \frac{p_{\text{data}}(x)}{p_{\theta}(x)} \\ = & \arg \min_{\theta} \sum_x -p_{\text{data}}(x) \log p_{\theta}(x) + \text{const} \\ = & \arg \max_{\theta} \sum_x p_{\text{data}}(x) \log p_{\theta}(x) \\ = & \arg \max_{\theta} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\theta}(x) \end{aligned}$$



Latent Variable Models

We want to maximize $\mathbb{E}_{x \sim p_{data}} \log p_\theta(x)$



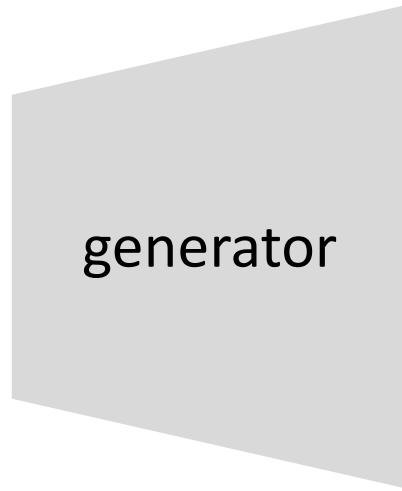
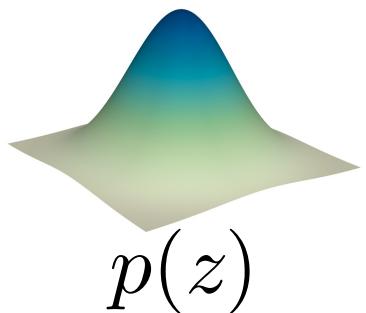
$$p_\theta(x)$$

Latent Variable Models

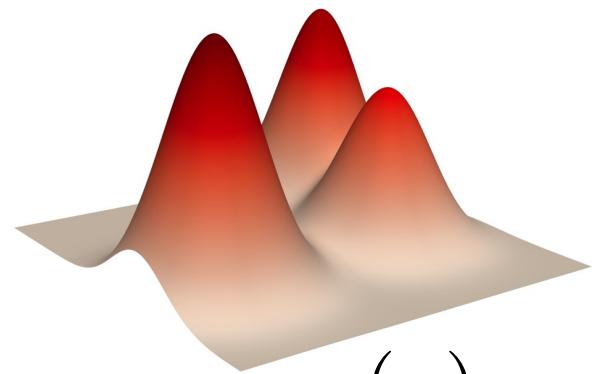
We want to maximize $\mathbb{E}_{x \sim p_{data}} \log p_\theta(x)$

with $p_\theta(x)$ represented as:

$$p_\theta(x) = \int_z p_\theta(x|z)p(z)dz$$



$$p_\theta(x|z)$$



$$p_\theta(x)$$

Latent Variable Models

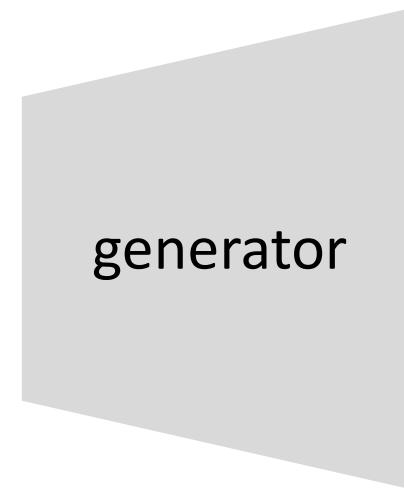
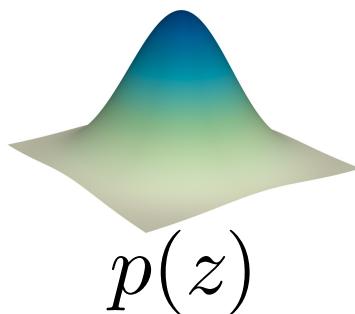
We want to maximize $\mathbb{E}_{x \sim p_{data}} \log p_\theta(x)$

with $p_\theta(x)$ represented as:

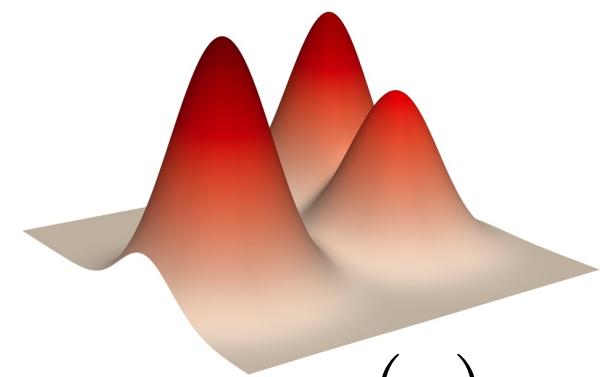
$$p_\theta(x) = \int_z p_\theta(x|z)p(z)dz$$

Two sets of unknowns:

- We need to optimize for θ
- We can't control “**true**” $p(z)$



$$p_\theta(x|z)$$



$$p_\theta(x)$$

Idea: introduce a “controllable” distribution $q(z)$

Latent Variable Models

$$\begin{aligned} \log p_{\theta}(x) &= \int_z q(z) \log p_{\theta}(x) dz \\ &= \int_z q(z) \log \left(\frac{p_{\theta}(x|z)p_{\theta}(z)}{p_{\theta}(z|x)} \right) dz \\ &= \int_z q(z) \log \left(\frac{p_{\theta}(x|z)p_{\theta}(z)}{p_{\theta}(z|x)} \frac{q(z)}{q(z)} \right) dz \\ &= \int_z q(z) \left(\log p_{\theta}(x|z) + \log \frac{p_{\theta}(z)}{q(z)} + \log \frac{q(z)}{p_{\theta}(z|x)} \right) dz \\ &= \mathbb{E}_{z \sim q(z)} \left[\log p_{\theta}(x|z) \right] - \mathcal{D}_{\text{KL}} \left(q(z) \parallel p_{\theta}(z) \right) + \mathcal{D}_{\text{KL}} \left(q(z) \parallel p_{\theta}(z|x) \right) \end{aligned}$$

Rewrite log likelihood by latent z

- for any distribution $q(z)$
- Bayes' rule

Latent Variable Models

$$\begin{aligned} & \log p_{\theta}(x) \\ = & \int_z q(z) \log p_{\theta}(x) dz \\ = & \int_z q(z) \log \left(\frac{p_{\theta}(x|z)p_{\theta}(z)}{p_{\theta}(z|x)} \right) dz \\ = & \int_z q(z) \log \left(\frac{p_{\theta}(x|z)p_{\theta}(z)}{p_{\theta}(z|x)} \frac{q(z)}{q(z)} \right) dz \\ = & \int_z q(z) \left(\log p_{\theta}(x|z) + \log \frac{p_{\theta}(z)}{q(z)} + \log \frac{q(z)}{p_{\theta}(z|x)} \right) dz \\ = & \mathbb{E}_{z \sim q(z)} \left[\log p_{\theta}(x|z) \right] - \mathcal{D}_{\text{KL}}(q(z) || p_{\theta}(z)) + \mathcal{D}_{\text{KL}}(q(z) || p_{\theta}(z|x)) \end{aligned}$$

Rewrite log likelihood by latent z

- for any distribution $q(z)$
- Bayes' rule
- just algebra
- just algebra

Latent Variable Models

$$\begin{aligned} & \log p_{\theta}(x) \\ = & \int_z q(z) \log p_{\theta}(x) dz \\ = & \int_z q(z) \log \left(\frac{p_{\theta}(x|z)p_{\theta}(z)}{p_{\theta}(z|x)} \right) dz \\ = & \int_z q(z) \log \left(\frac{p_{\theta}(x|z)p_{\theta}(z)}{p_{\theta}(z|x)} \frac{q(z)}{q(z)} \right) dz \\ = & \int_z q(z) \left(\log p_{\theta}(x|z) + \log \frac{p_{\theta}(z)}{q(z)} + \log \frac{q(z)}{p_{\theta}(z|x)} \right) dz \\ = & \mathbb{E}_{z \sim q(z)} \left[\log p_{\theta}(x|z) \right] - \mathcal{D}_{\text{KL}}(q(z) || p_{\theta}(z)) + \mathcal{D}_{\text{KL}}(q(z) || p_{\theta}(z|x)) \end{aligned}$$

Rewrite log likelihood by latent z

- for any distribution $q(z)$
- Bayes' rule
 - just algebra
- just algebra

Latent Variable Models

$$\begin{aligned} & \log p_{\theta}(x) \\ = & \int_z q(z) \log p_{\theta}(x) dz \\ = & \int_z q(z) \log \left(\frac{p_{\theta}(x|z)p_{\theta}(z)}{p_{\theta}(z|x)} \right) dz \\ = & \int_z q(z) \log \left(\frac{p_{\theta}(x|z)p_{\theta}(z)}{p_{\theta}(z|x)} \frac{q(z)}{q(z)} \right) dz \\ = & \int_z q(z) \left(\log p_{\theta}(x|z) + \log \frac{p_{\theta}(z)}{q(z)} + \log \frac{q(z)}{p_{\theta}(z|x)} \right) dz \\ = & \mathbb{E}_{z \sim q(z)} [\log p_{\theta}(x|z)] - \mathcal{D}_{\text{KL}}(q(z) || p_{\theta}(z)) + \mathcal{D}_{\text{KL}}(q(z) || p_{\theta}(z|x)) \end{aligned}$$

Rewrite log likelihood by latent z

- for any distribution $q(z)$
- Bayes' rule
- just algebra
- just algebra

Latent Variable Models

$$\begin{aligned} & \log p_{\theta}(x) \\ = & \int_z q(z) \log p_{\theta}(x) dz \\ = & \int_z q(z) \log \left(\frac{p_{\theta}(x|z)p_{\theta}(z)}{p_{\theta}(z|x)} \right) dz \\ = & \int_z q(z) \log \left(\frac{p_{\theta}(x|z)p_{\theta}(z)}{p_{\theta}(z|x)} \frac{q(z)}{q(z)} \right) dz \\ = & \int_z q(z) \left(\log p_{\theta}(x|z) + \log \frac{p_{\theta}(z)}{q(z)} + \log \frac{q(z)}{p_{\theta}(z|x)} \right) dz \\ = & \mathbb{E}_{z \sim q(z)} \left[\log p_{\theta}(x|z) \right] - \mathcal{D}_{\text{KL}}(q(z) || p_{\theta}(z)) + \mathcal{D}_{\text{KL}}(q(z) || p_{\theta}(z|x)) \end{aligned}$$

Rewrite log likelihood by latent z

- for any distribution $q(z)$
- Bayes' rule
- just algebra
- just algebra

Latent Variable Models

$$\begin{aligned} & \log p_{\theta}(x) \\ = & \int_z q(z) \log p_{\theta}(x) dz \\ = & \int_z q(z) \log \left(\frac{p_{\theta}(x|z)p_{\theta}(z)}{p_{\theta}(z|x)} \right) dz \\ = & \int_z q(z) \log \left(\frac{p_{\theta}(x|z)p_{\theta}(z)}{p_{\theta}(z|x)} \frac{q(z)}{q(z)} \right) dz \\ = & \int_z q(z) \left(\log p_{\theta}(x|z) + \log \frac{p_{\theta}(z)}{q(z)} + \log \frac{q(z)}{p_{\theta}(z|x)} \right) dz \\ = & \mathbb{E}_{z \sim q(z)} \left[\log p_{\theta}(x|z) \right] - \mathcal{D}_{\text{KL}} \left(q(z) \parallel p_{\theta}(z) \right) + \mathcal{D}_{\text{KL}} \left(q(z) \parallel p_{\theta}(z|x) \right) \end{aligned}$$

Rewrite log likelihood by latent z

- for any distribution $q(z)$
- Bayes' rule
- just algebra
- just algebra

Latent Variable Models

intractable

$$\log p_{\theta}(x)$$

$$= \int_z q(z) \log p_{\theta}(x) dz$$

$$= \int_z q(z) \log \left(\frac{p_{\theta}(x|z)p_{\theta}(z)}{p_{\theta}(z|x)} \right) dz$$

$$= \int_z q(z) \log \left(\frac{p_{\theta}(x|z)p_{\theta}(z)}{p_{\theta}(z|x)} \frac{q(z)}{q(z)} \right) dz$$

$$= \int_z q(z) \left(\log p_{\theta}(x|z) + \log \frac{p_{\theta}(z)}{q(z)} + \log \frac{q(z)}{p_{\theta}(z|x)} \right) dz$$

$$= \mathbb{E}_{z \sim q(z)} [\log p_{\theta}(x|z)] - \mathcal{D}_{\text{KL}}(q(z) || p_{\theta}(z)) + \mathcal{D}_{\text{KL}}(q(z) || p_{\theta}(z|x))$$

tractable

tractable

intractable

Rewrite log likelihood by latent z

- for any distribution $q(z)$
- Bayes' rule

Latent Variable Models

$$\begin{aligned} & \text{intractable } \boxed{\log p_{\theta}(x)} - \mathcal{D}_{\text{KL}}\left(q(z)||p_{\theta}(z|x)\right) \text{ intractable} \\ &= \boxed{\mathbb{E}_{z \sim q(z)} \left[\log p_{\theta}(x|z) \right]} - \mathcal{D}_{\text{KL}}\left(q(z)||p_{\theta}(z)\right) \\ & \qquad \qquad \qquad \text{tractable} \qquad \qquad \qquad \text{tractable} \end{aligned}$$

Latent Variable Models

$$\begin{aligned} & \text{intractable } \boxed{\log p_\theta(x)} - \mathcal{D}_{\text{KL}}(q(z) || p_\theta(z|x)) \text{ intractable} \\ = & \boxed{\mathbb{E}_{z \sim q(z)} [\log p_\theta(x|z)]} - \mathcal{D}_{\text{KL}}(q(z) || p_\theta(z)) \\ & \quad \text{tractable} \qquad \qquad \text{tractable} \end{aligned}$$


- This is called Evidence Lower Bound (ELBO)
- Lower bound of $\log p_\theta(x)$
- This equation holds for any distribution $q(z)$

Latent Variable Models

- This is called Evidence Lower Bound (ELBO)
 - Lower bound of $\log p_\theta(x)$
 - This equation holds for any distribution $q(z)$
 - Parameterize $q(z)$ by $q_\phi(z|x)$

Latent Variable Models

- This is called Evidence Lower Bound (ELBO)
 - Lower bound of $\log p_\theta(x)$
 - This equation holds for any distribution $q(z)$
 - Parameterize $q(z)$ by $q_\phi(z|x)$
 - let $p_\theta(z)$ be a simple known prior $p(z)$

Variational Autoencoder

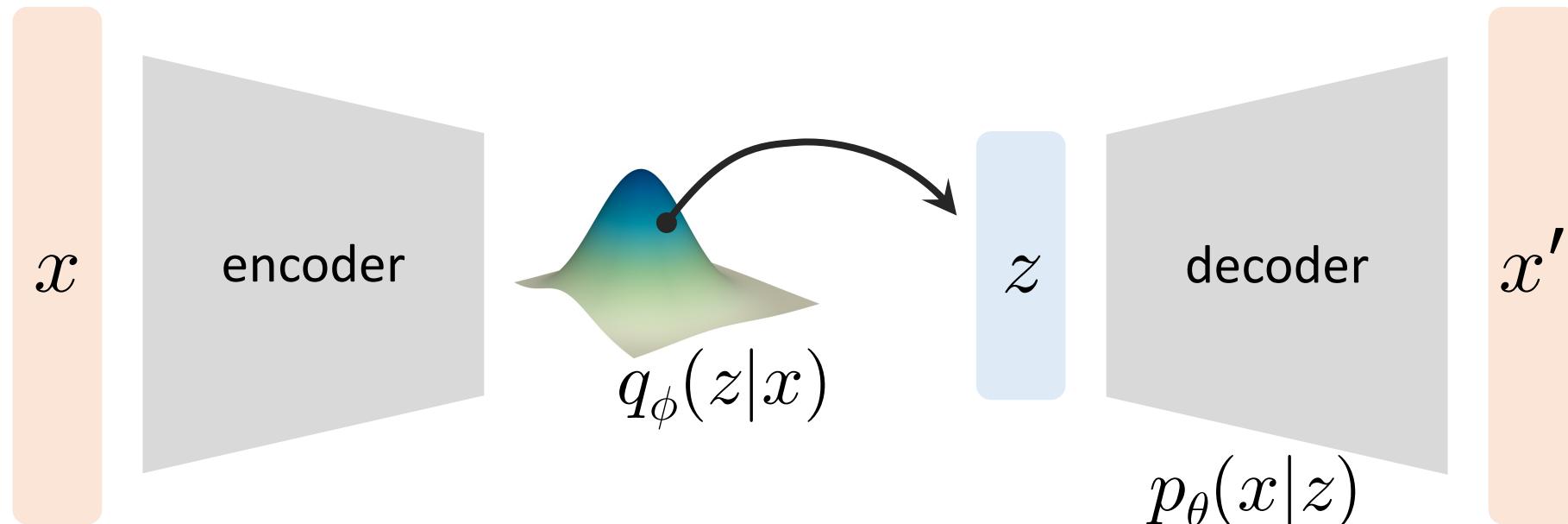
Maximize ELBO \Rightarrow minimize an objective:

$$\mathcal{L}_{\theta,\phi}(x) = -\mathbb{E}_{z \sim q_{\phi}(z|x)} [\log p_{\theta}(x|z)] + \mathcal{D}_{\text{KL}}(q_{\phi}(z|x) || p(z))$$

Variational Autoencoder

Maximize ELBO \Rightarrow minimize an objective:

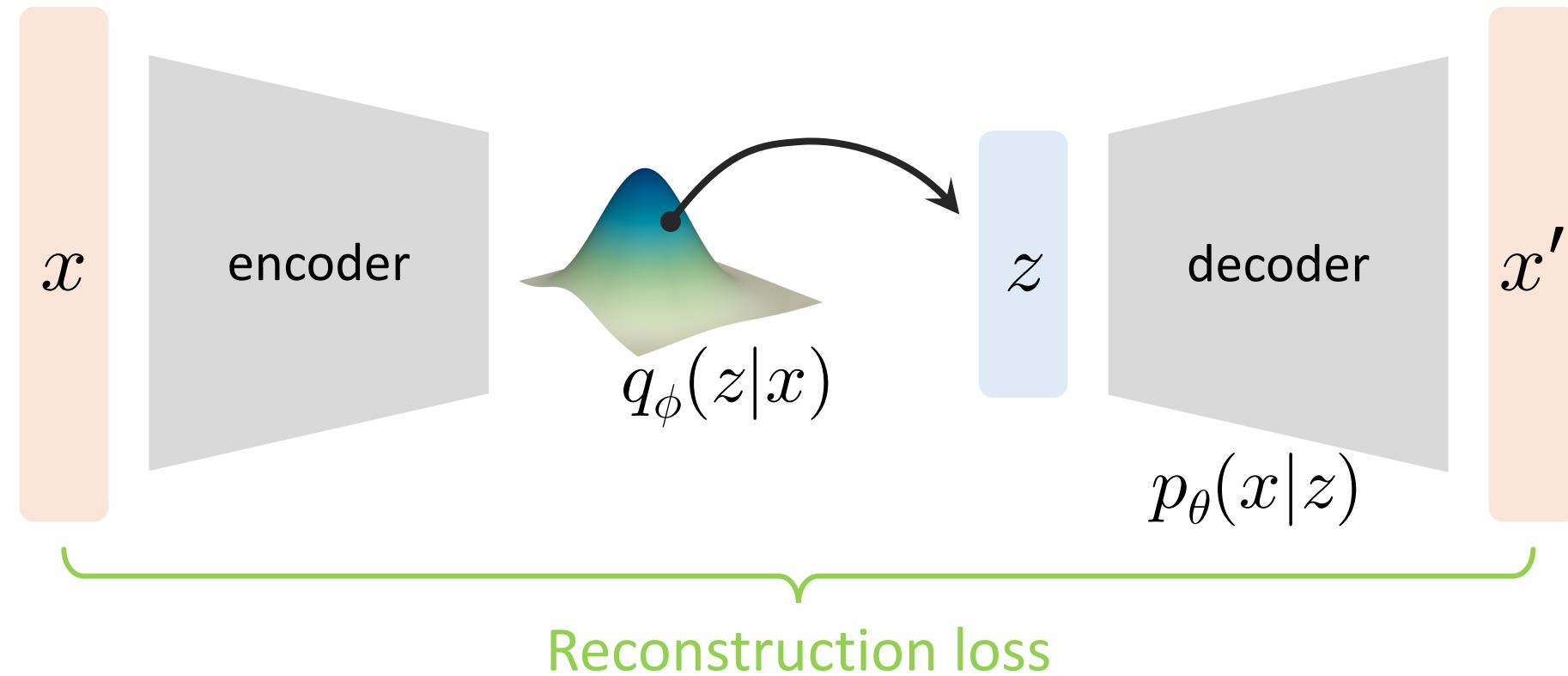
$$\mathcal{L}_{\theta, \phi}(x) = -\mathbb{E}_{z \sim q_{\phi}(z|x)} [\log p_{\theta}(x|z)] + \mathcal{D}_{\text{KL}}(q_{\phi}(z|x) || p(z))$$



Variational Autoencoder

Maximize ELBO \Rightarrow minimize an objective:

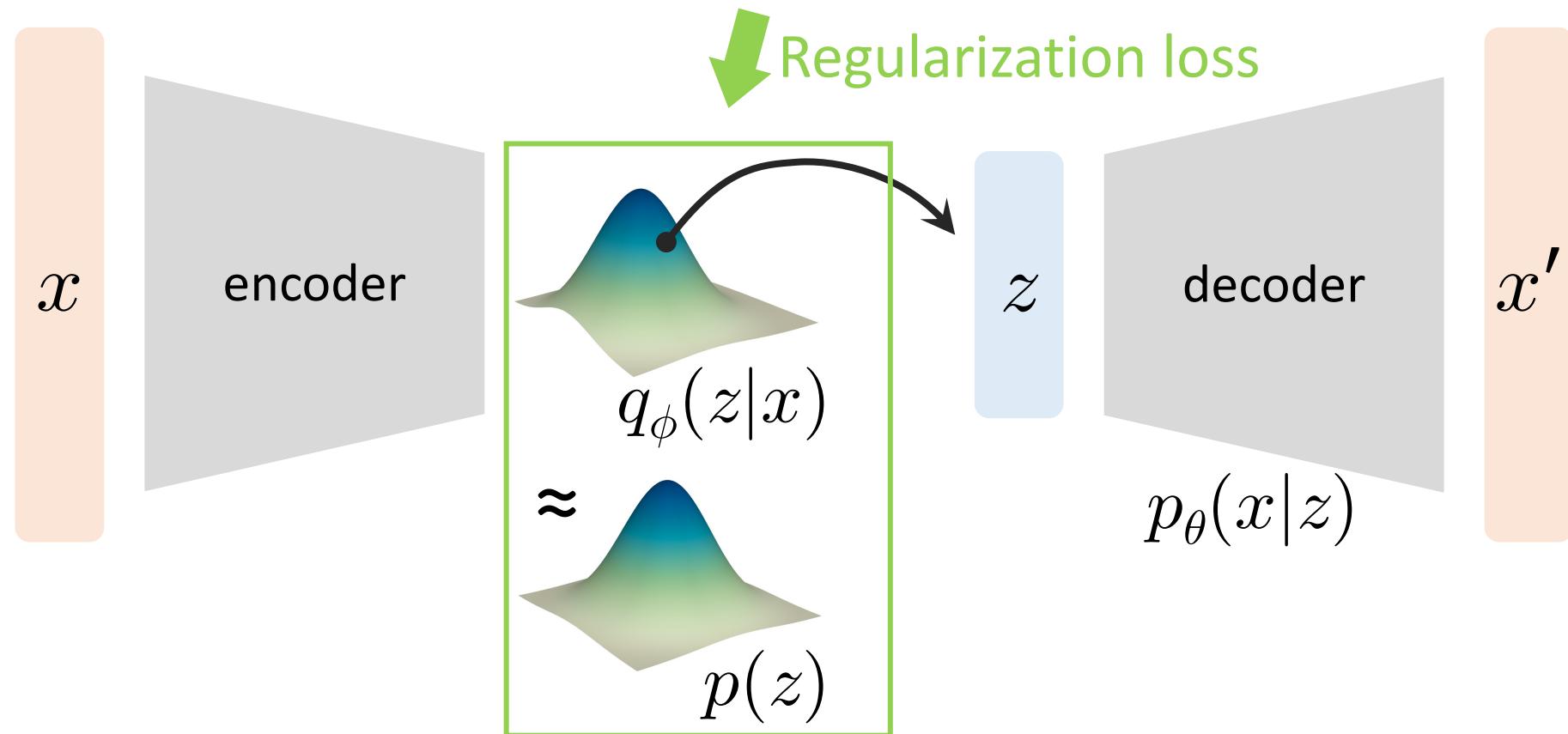
$$\mathcal{L}_{\theta, \phi}(x) = -\mathbb{E}_{z \sim q_{\phi}(z|x)} [\log p_{\theta}(x|z)] + \mathcal{D}_{\text{KL}}(q_{\phi}(z|x) || p(z))$$



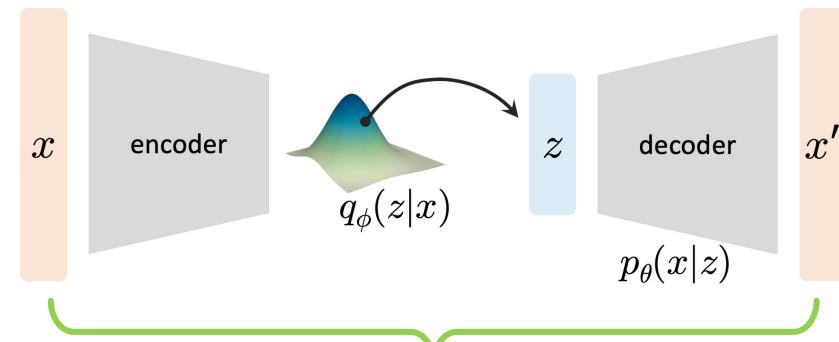
Variational Autoencoder

Maximize ELBO \Rightarrow minimize an objective:

$$\mathcal{L}_{\theta, \phi}(x) = -\mathbb{E}_{z \sim q_{\phi}(z|x)} [\log p_{\theta}(x|z)] + \boxed{\mathcal{D}_{\text{KL}}(q_{\phi}(z|x) || p(z))}$$



Variational Autoencoder



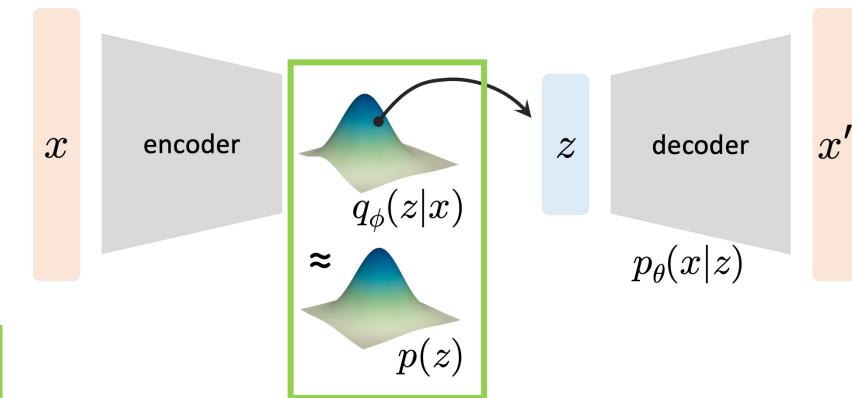
Reconstruction loss

$$\mathcal{L}_{\theta, \phi}(x) = -\mathbb{E}_{z \sim q_\phi(z|x)} [\log p_\theta(x|z)] + \mathcal{D}_{\text{KL}}(q_\phi(z|x) || p(z))$$

Example: L2 loss

- one-step Monte Carlo: $z \sim q_\phi(z|x)$
- map z by decoder net: $g_\theta(z) \rightarrow x'$ network estimates
distribution's parameters
- model $p_\theta(x|z)$ by Gaussian: $p_\theta(x|z) = \mathcal{N}(x | x', \sigma_0^2)$ (assume fixed std)
- negative log likelihood: $\frac{1}{2\sigma_0^2} \|x - x'\|^2 + \text{const}$
- L2 loss \Rightarrow a Gaussian neighborhood around data point x

Variational Autoencoder



Regularization loss

$$\mathcal{L}_{\theta, \phi}(x) = -\mathbb{E}_{z \sim q_\phi(z|x)} [\log p_\theta(x|z)] + \boxed{\mathcal{D}_{\text{KL}}(q_\phi(z|x) || p(z))}$$

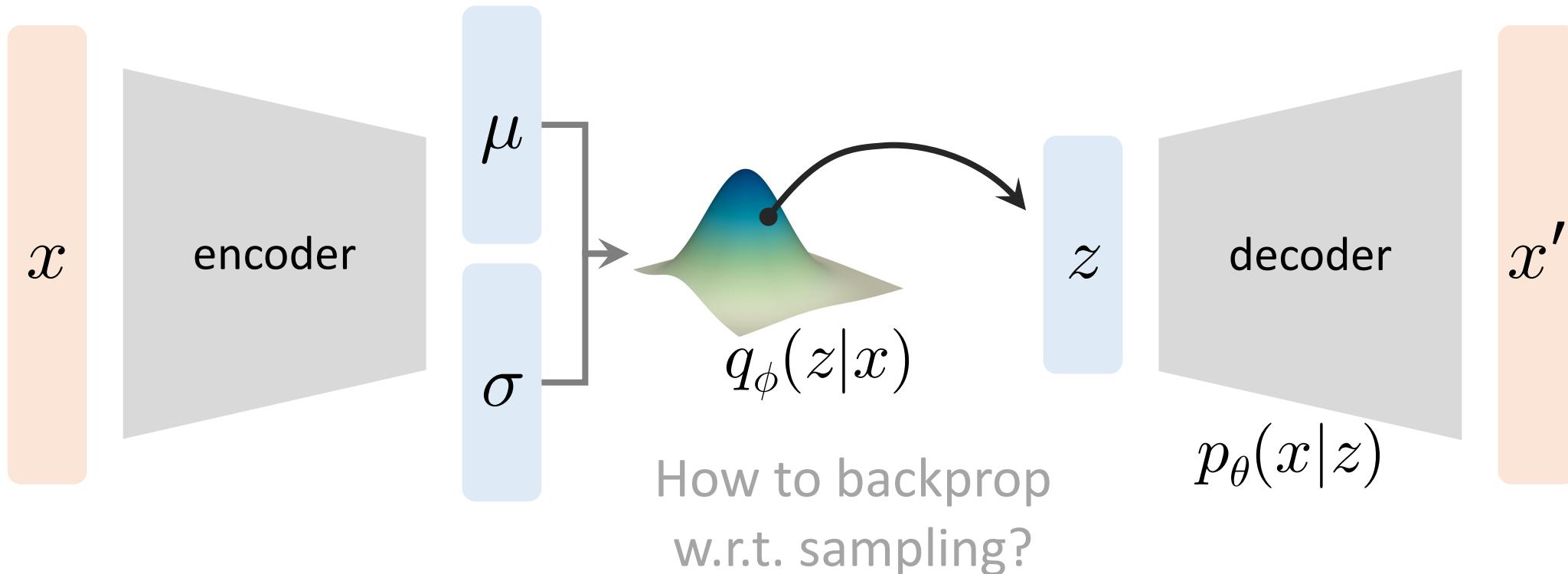
Example: Gaussian prior

- let $p(z) = \mathcal{N}(z | 0, \mathbf{I})$
- model $q_\phi(z|x)$ by Gaussian: $\mathcal{N}(z | \mu, \sigma)$
- map x by encoder net: $f_\phi(x) \rightarrow \mu, \sigma$ again, network estimates distribution's parameters
- compute loss analytically: $\mathcal{D}_{\text{KL}}(\mathcal{N}(z | \mu, \sigma) || \mathcal{N}(z | 0, \mathbf{I}))$ (see pset 1)
- fixed covariance \Rightarrow L2 loss on μ (see pset 1)

Variational Autoencoder

Maximize ELBO \Rightarrow minimize an objective:

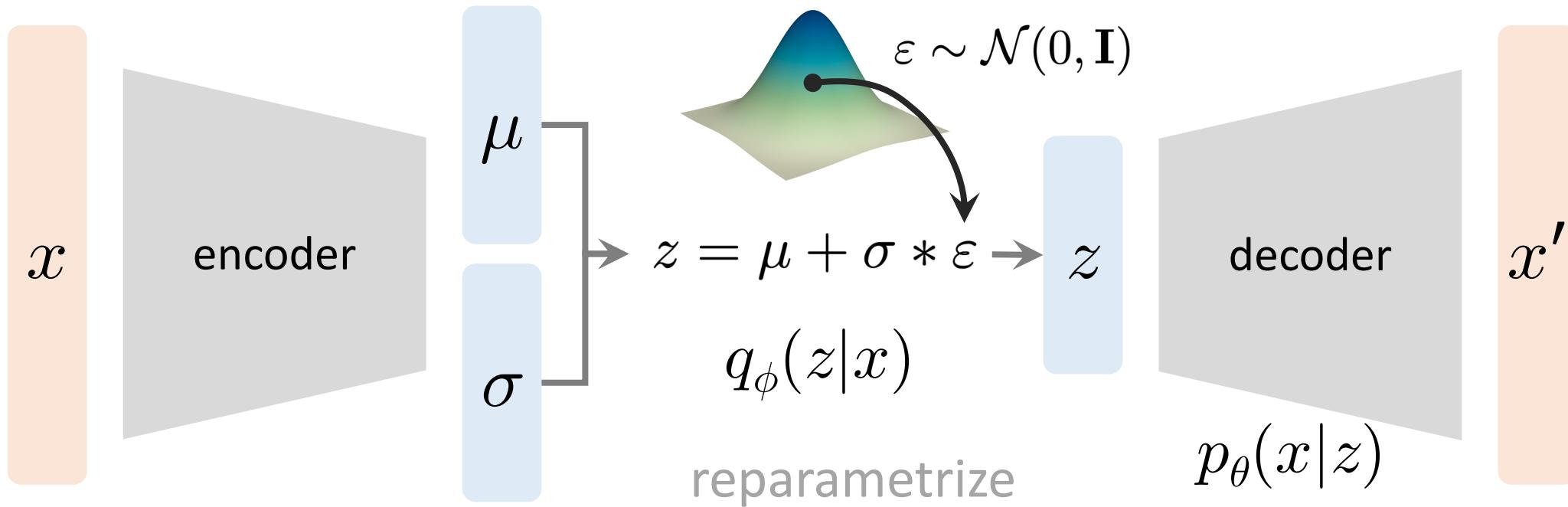
$$\mathcal{L}_{\theta, \phi}(x) = -\mathbb{E}_{z \sim q_{\phi}(z|x)} [\log p_{\theta}(x|z)] + \mathcal{D}_{\text{KL}}(q_{\phi}(z|x) || p(z))$$



Variational Autoencoder

Maximize ELBO \Rightarrow minimize an objective:

$$\mathcal{L}_{\theta, \phi}(x) = -\mathbb{E}_{z \sim q_{\phi}(z|x)} [\log p_{\theta}(x|z)] + \mathcal{D}_{\text{KL}}(q_{\phi}(z|x) || p(z))$$



Variational Autoencoder

... so far, we have discussed an objective on one x :

$$\mathcal{L}_{\theta,\phi}(x) = -\mathbb{E}_{z \sim q_{\phi}(z|x)} [\log p_{\theta}(x|z)] + \mathcal{D}_{\text{KL}}(q_{\phi}(z|x) || p(z))$$

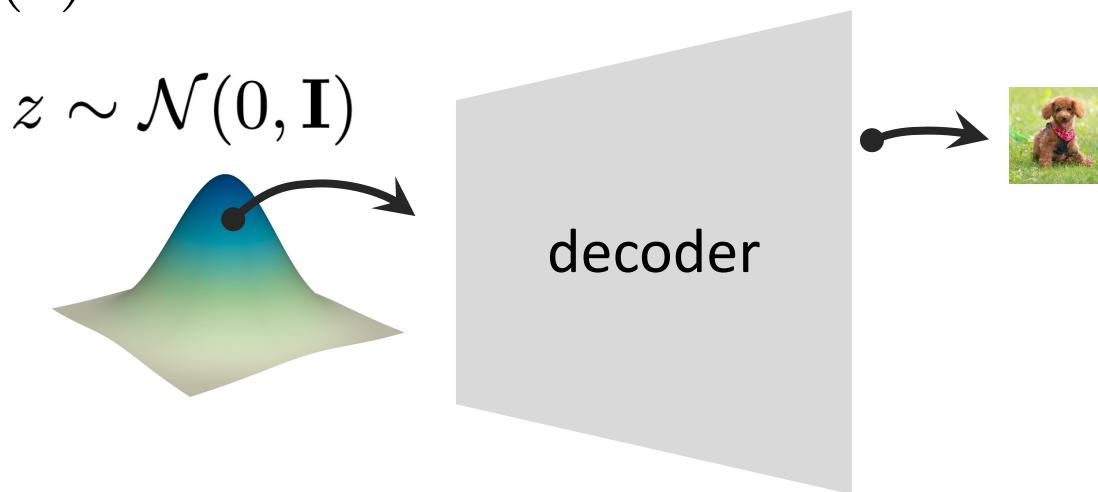
Overall loss is expectation over data:

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

Variational Autoencoder

Inference (generation):

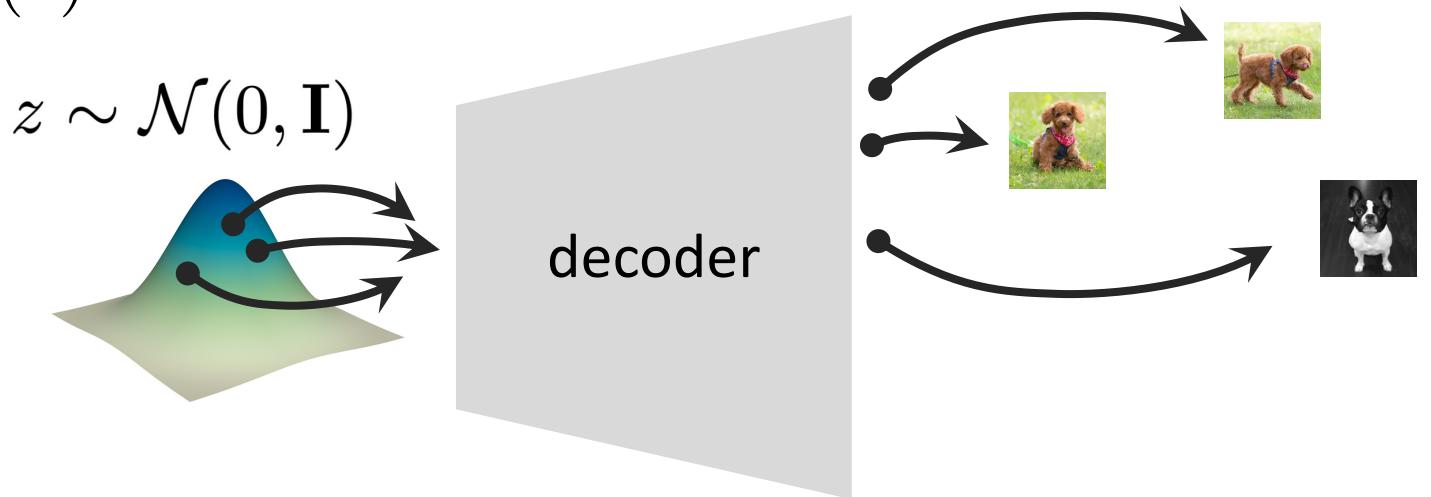
- sample z from: $\mathcal{N}(0, \mathbf{I})$
- map z by decoder net: $g_\theta(z)$



Variational Autoencoder

Inference (generation):

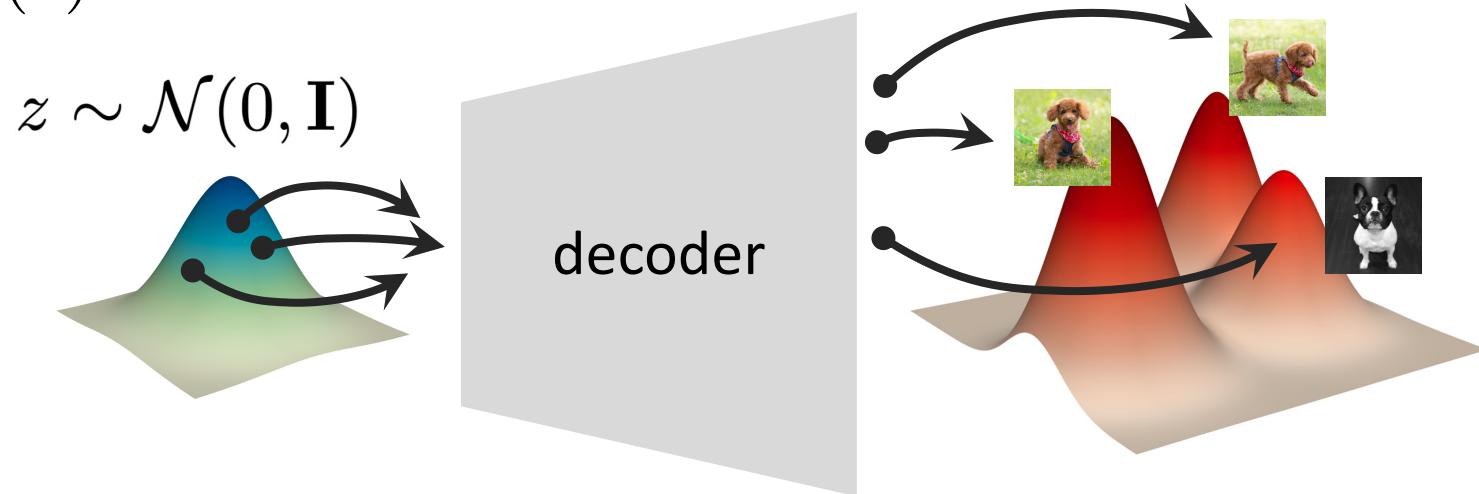
- sample z from: $\mathcal{N}(0, \mathbf{I})$
- map z by decoder net: $g_\theta(z)$



Variational Autoencoder

Inference (generation):

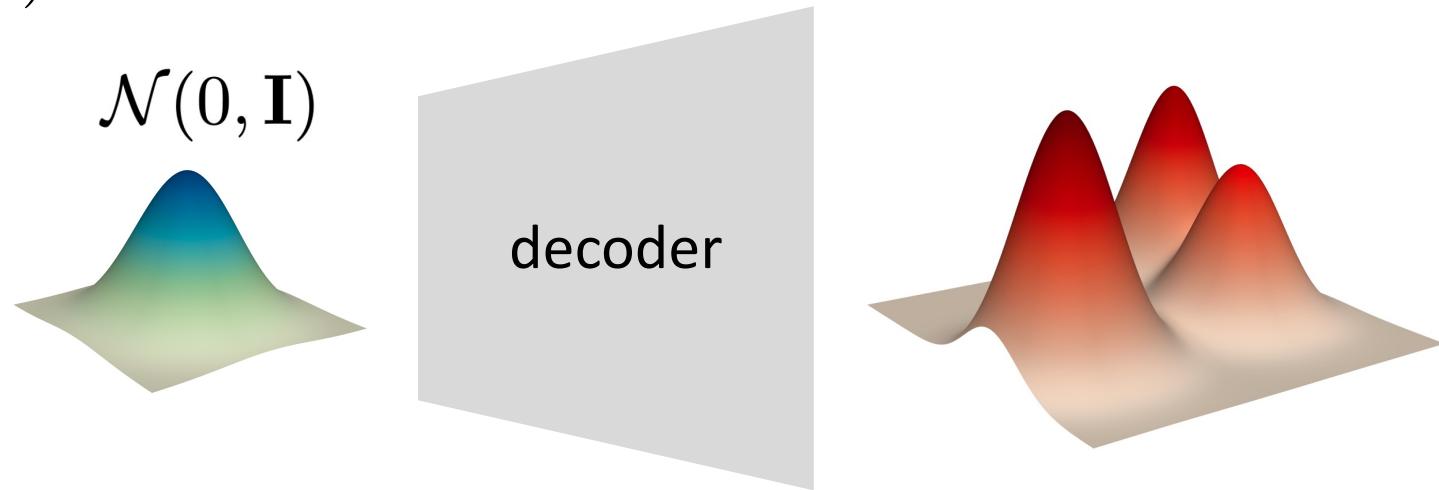
- sample z from: $\mathcal{N}(0, \mathbf{I})$
- map z by decoder net: $g_\theta(z)$



Variational Autoencoder

Inference (generation):

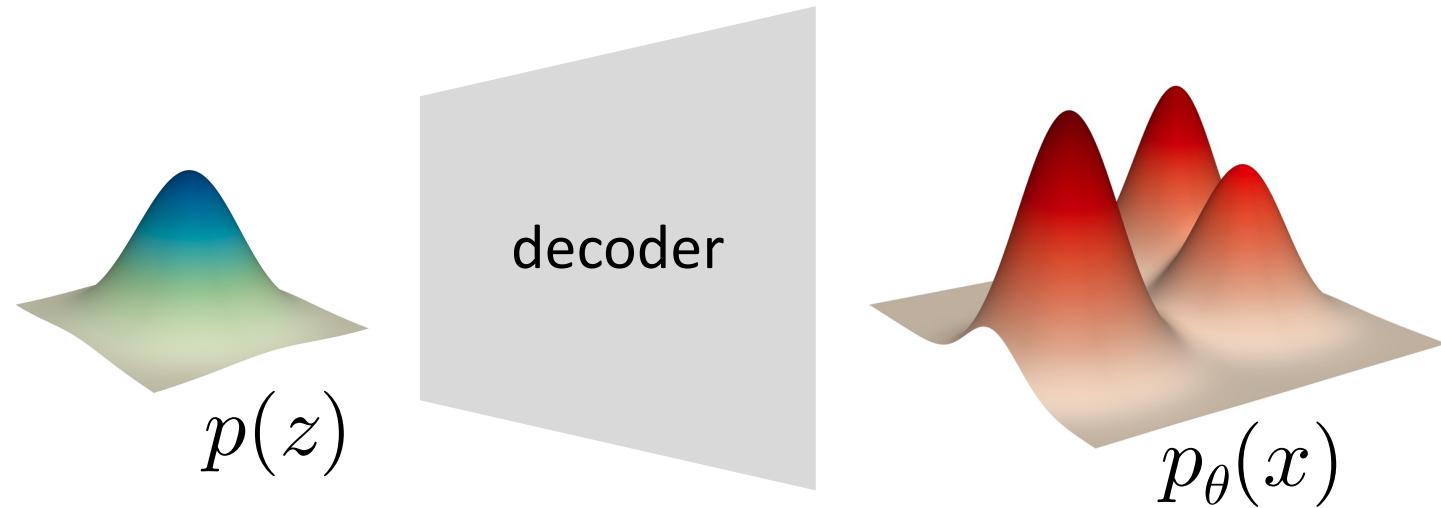
- sample z from: $\mathcal{N}(0, \mathbf{I})$
- map z by decoder net: $g_\theta(z)$



Decoder is a deterministic mapping from one distribution to another.

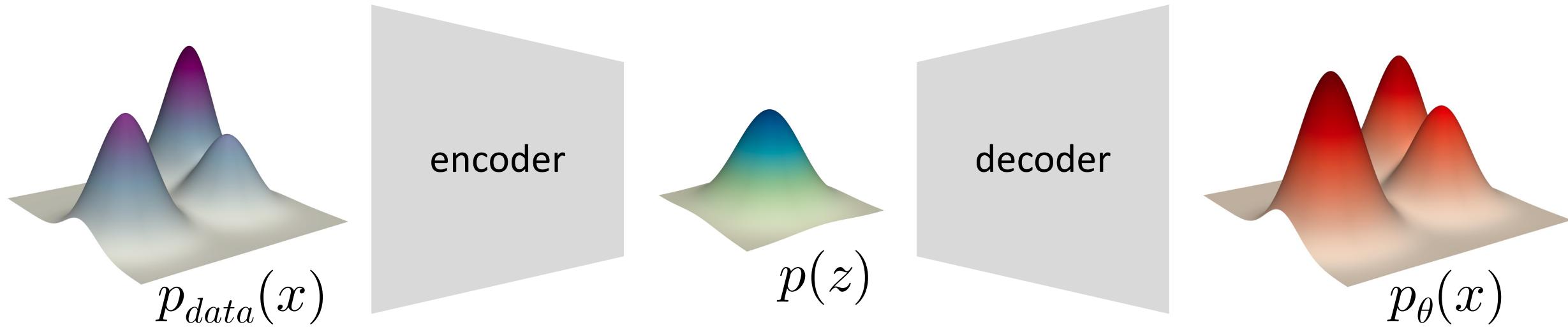
A view of “Autoencoding Distributions”

- decoder: maps latent distribution to data distribution



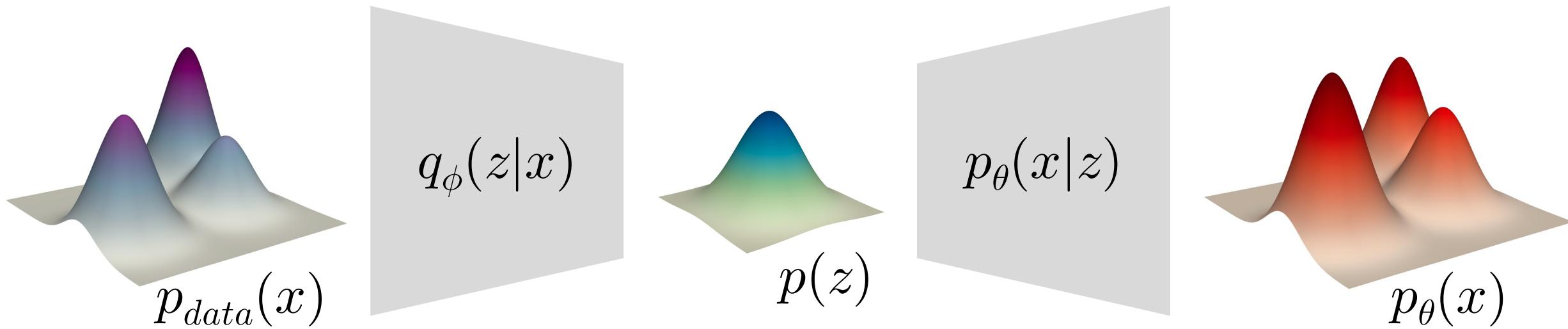
A view of “Autoencoding Distributions”

- encoder: maps data distribution to latent distribution
- decoder: maps latent distribution to data distribution



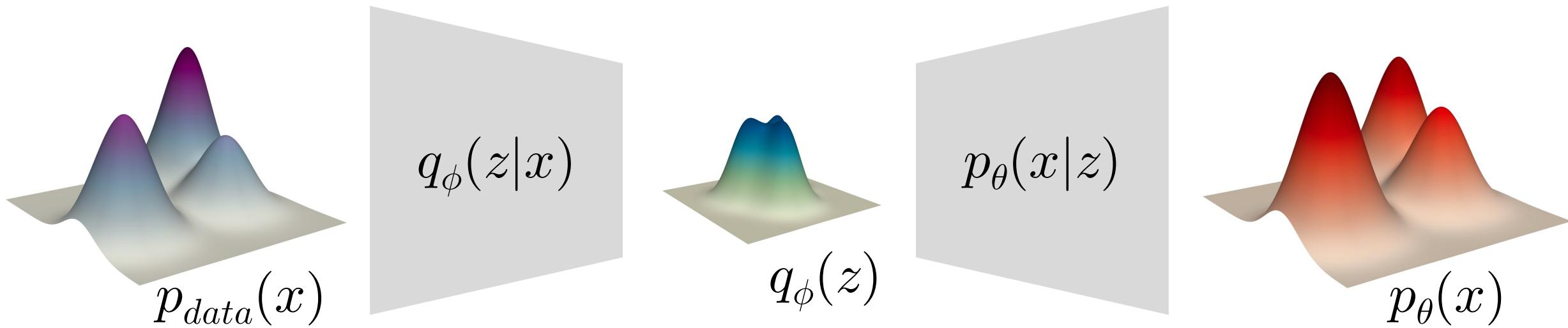
A view of “Autoencoding Distributions”

- encoder: maps data distribution to latent distribution
- decoder: maps latent distribution to data distribution



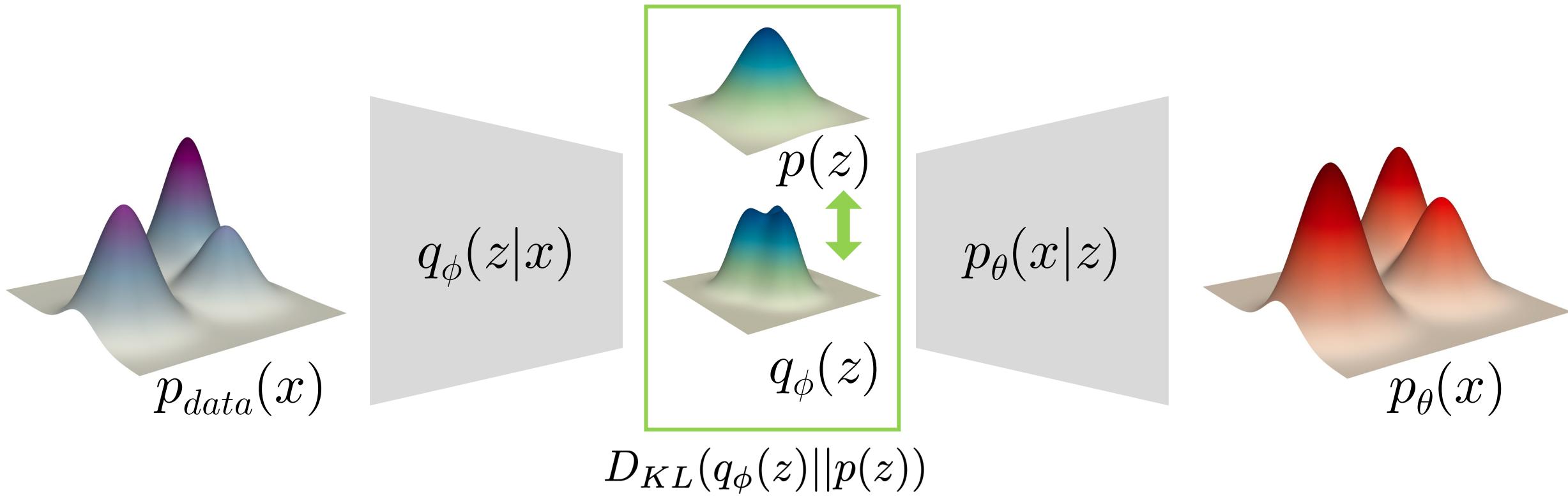
A view of “Autoencoding Distributions”

- encoded latent distribution: $q_\phi(z) = \int_x q_\phi(z|x)p_{data}(x)dx$



A view of “Autoencoding Distributions”

- encoded latent distribution: $q_\phi(z) = \int_x q_\phi(z|x)p_{data}(x)dx$



E.g., see “InfoVAE: Information Maximizing Variational Autoencoders”, 2017

Illustration

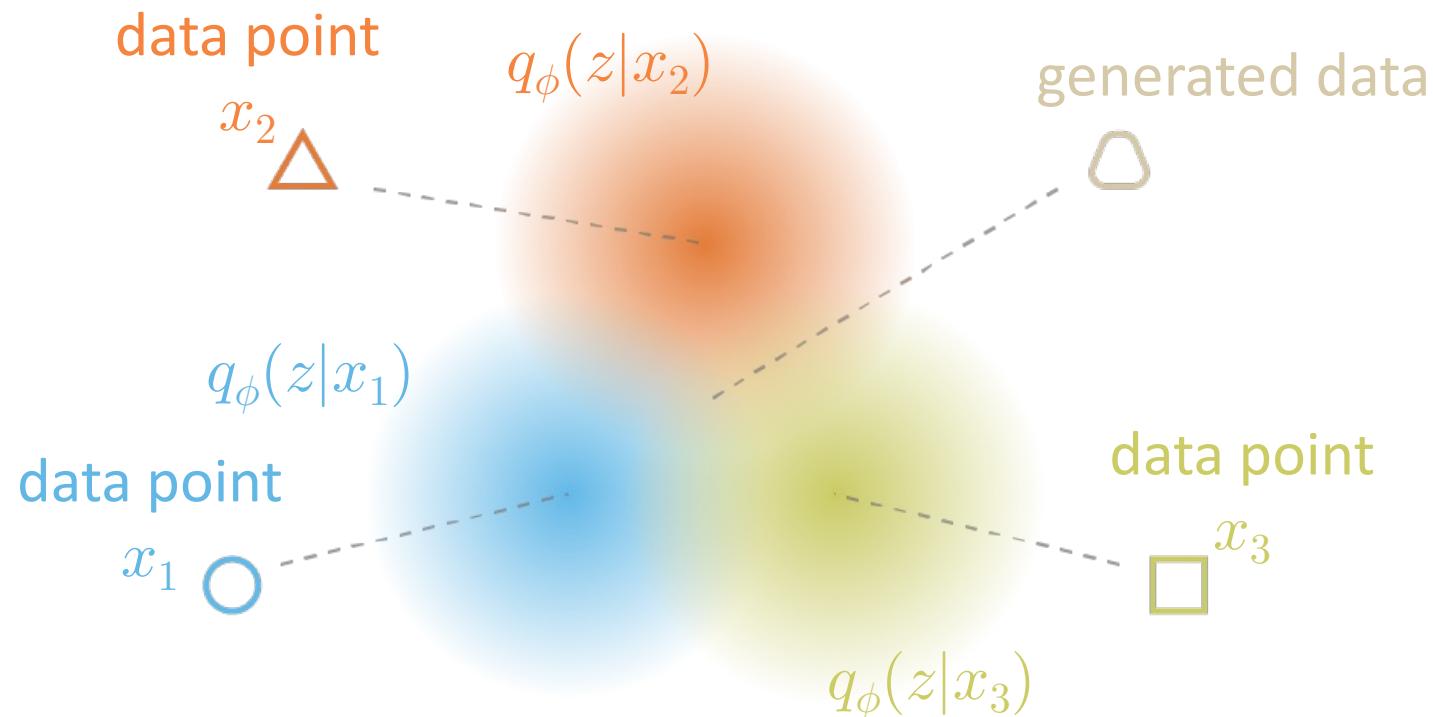


Figure adapted from: Joseph Rocca "Understanding Variational Autoencoders (VAEs)"
<https://towardsdatascience.com/understanding-variational-autoencoders-vae-f70510919f73>

Illustration

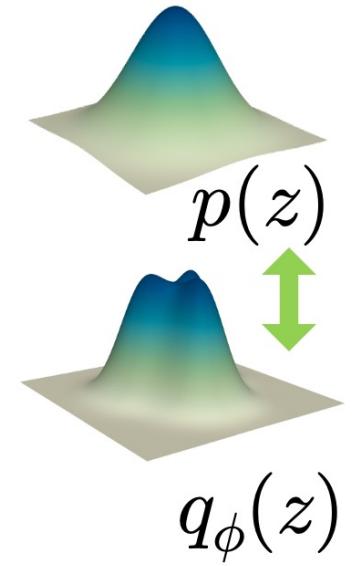
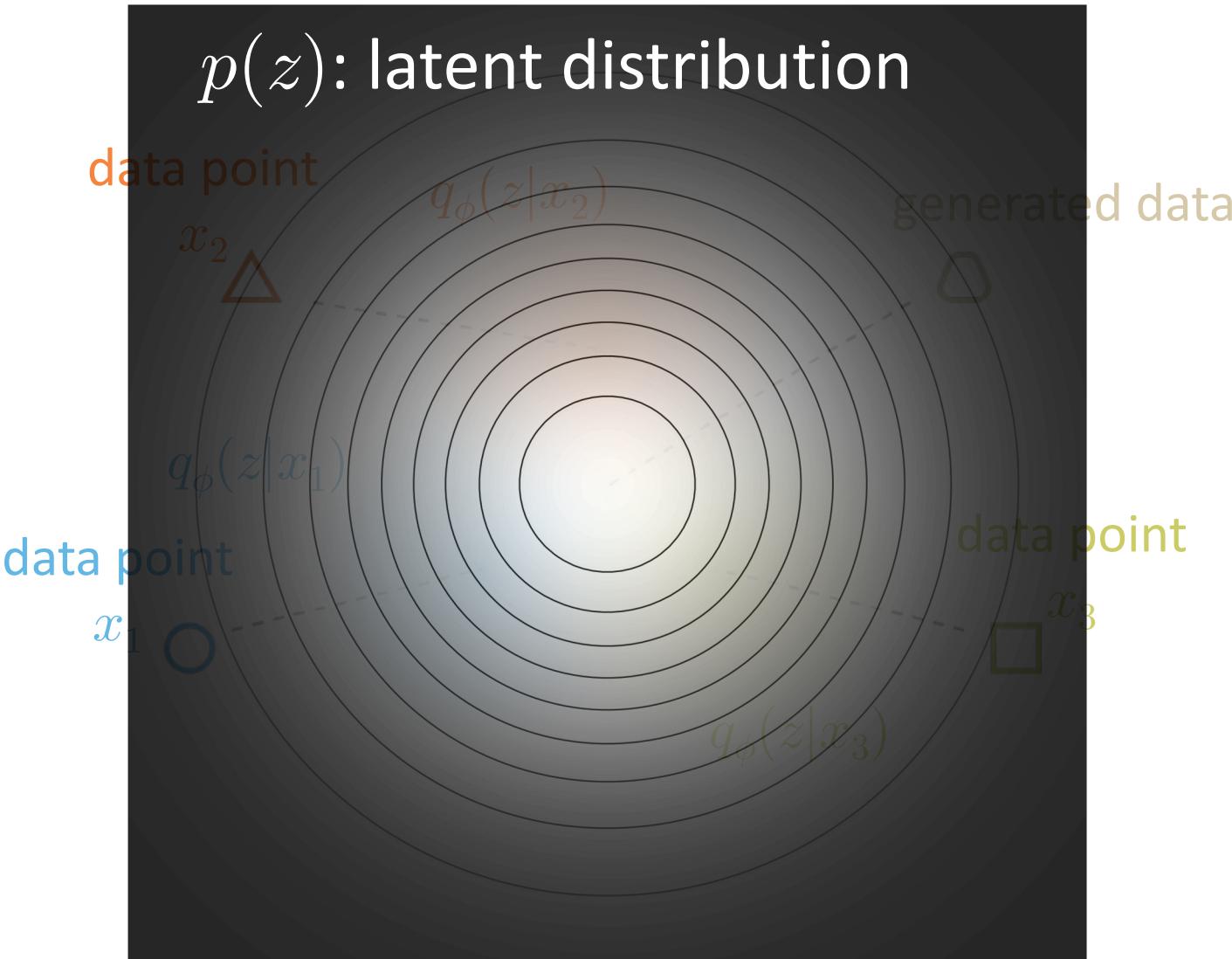
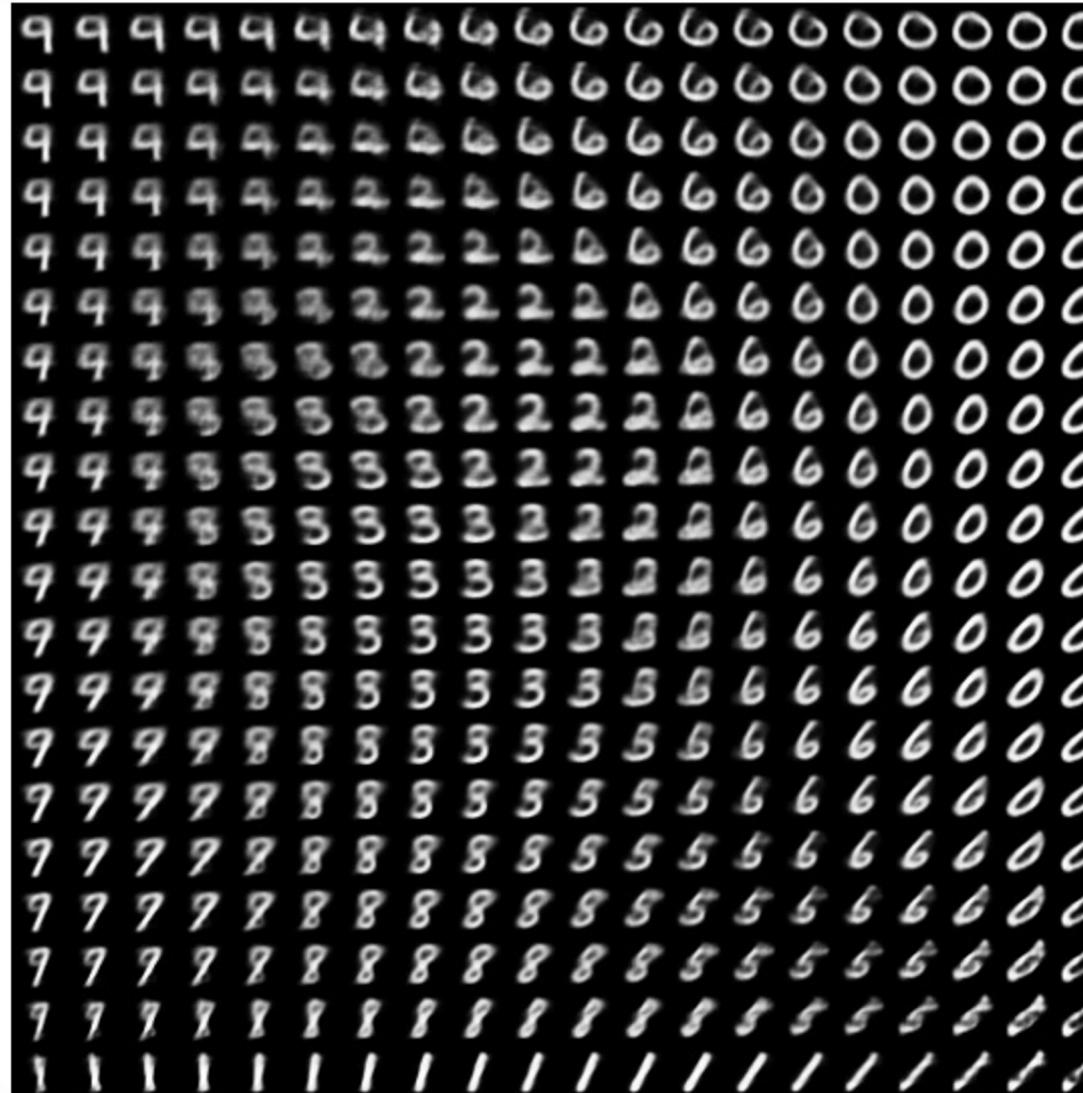


Figure adapted from: Joseph Rocca “Understanding Variational Autoencoders (VAEs)”
<https://towardsdatascience.com/understanding-variational-autoencoders-vae-f70510919f73>

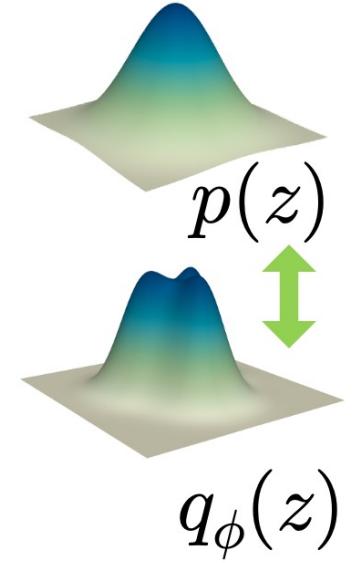
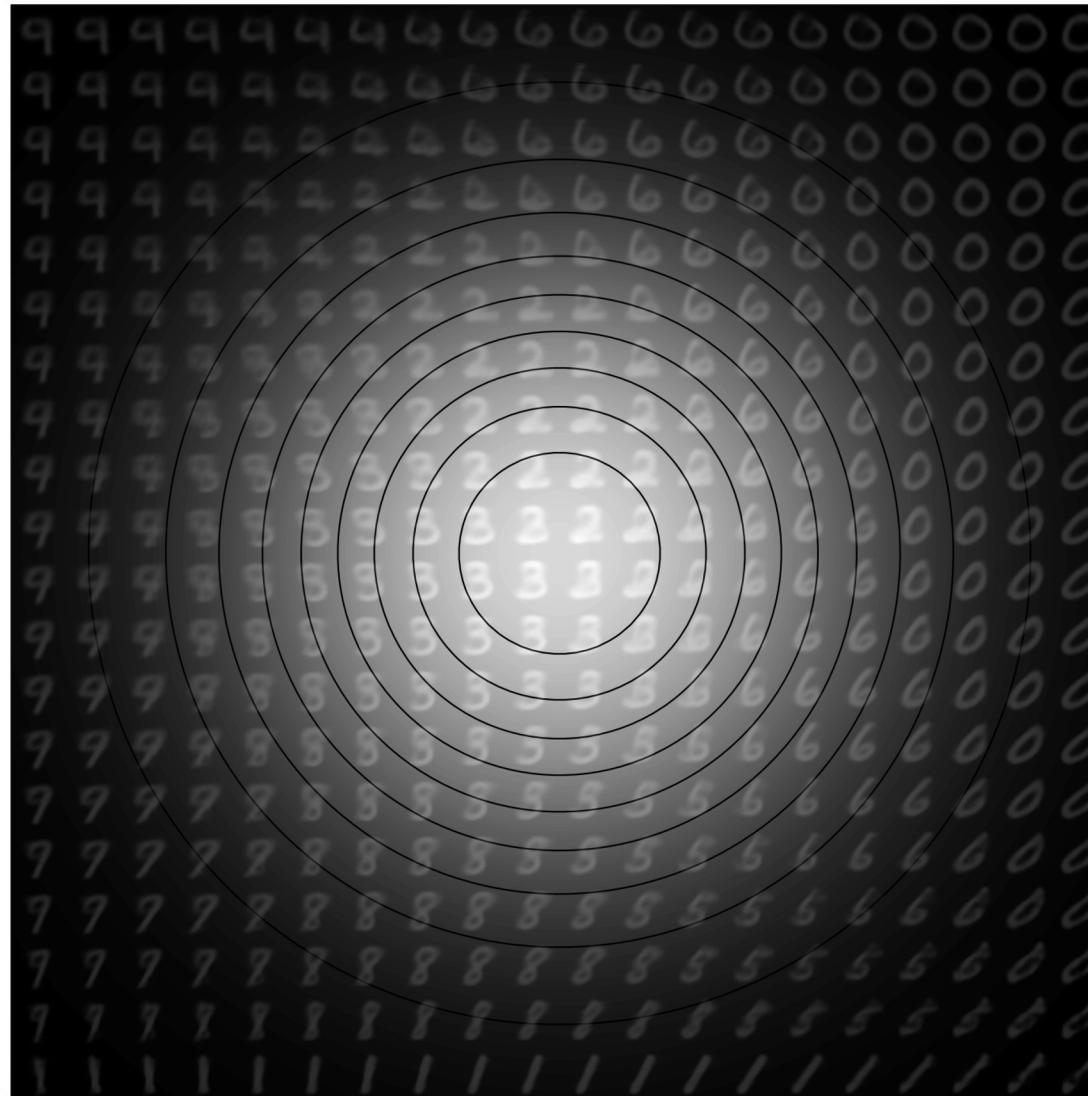
VAE: 2D latent space on MNIST



“Convolutional Variational Autoencoder”

<https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/generative/cvae.ipynb>

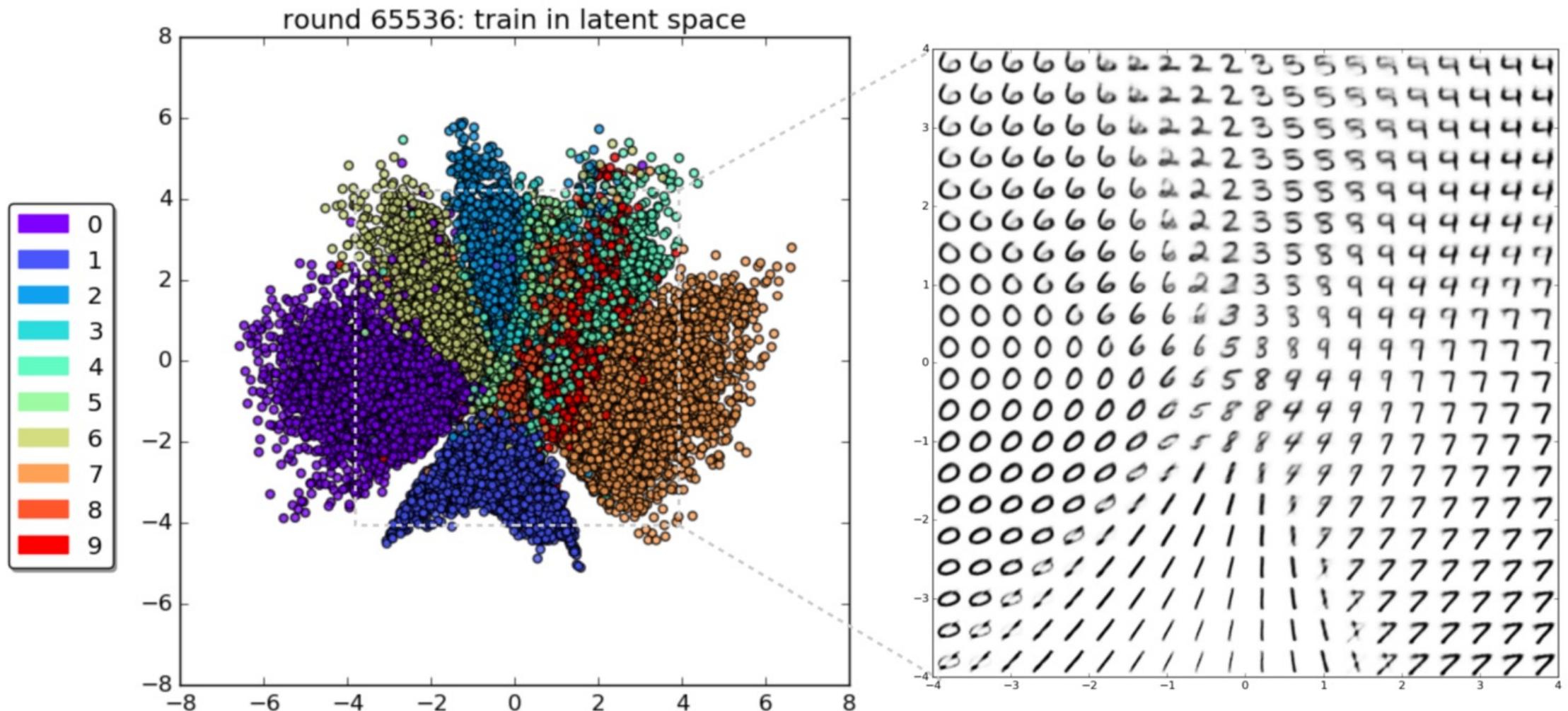
VAE: 2D latent space on MNIST



“Convolutional Variational Autoencoder”

<https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/generative/cvae.ipynb>

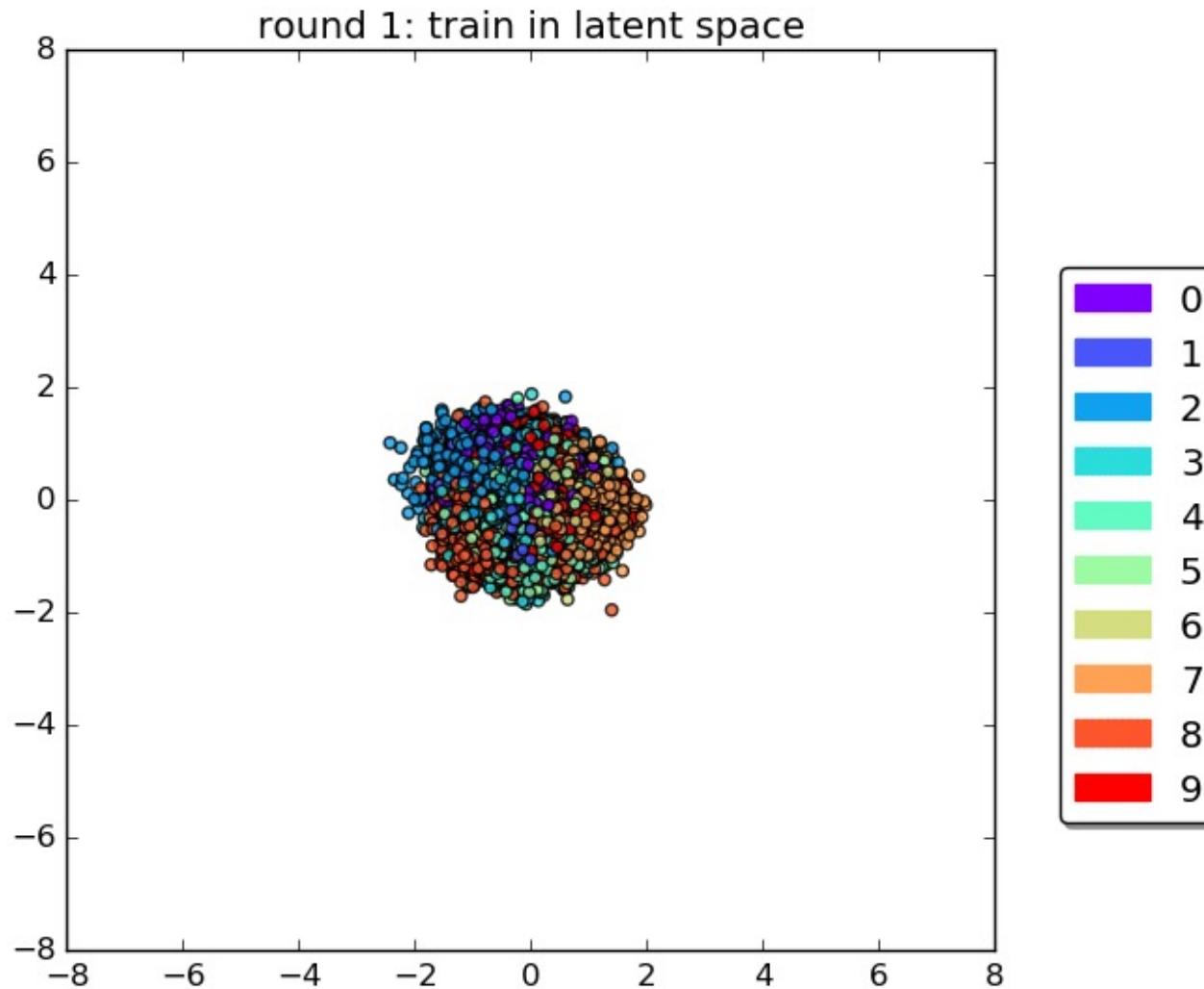
VAE: 2D latent space on MNIST



"Introducing Variational Autoencoders (in Prose and Code)"

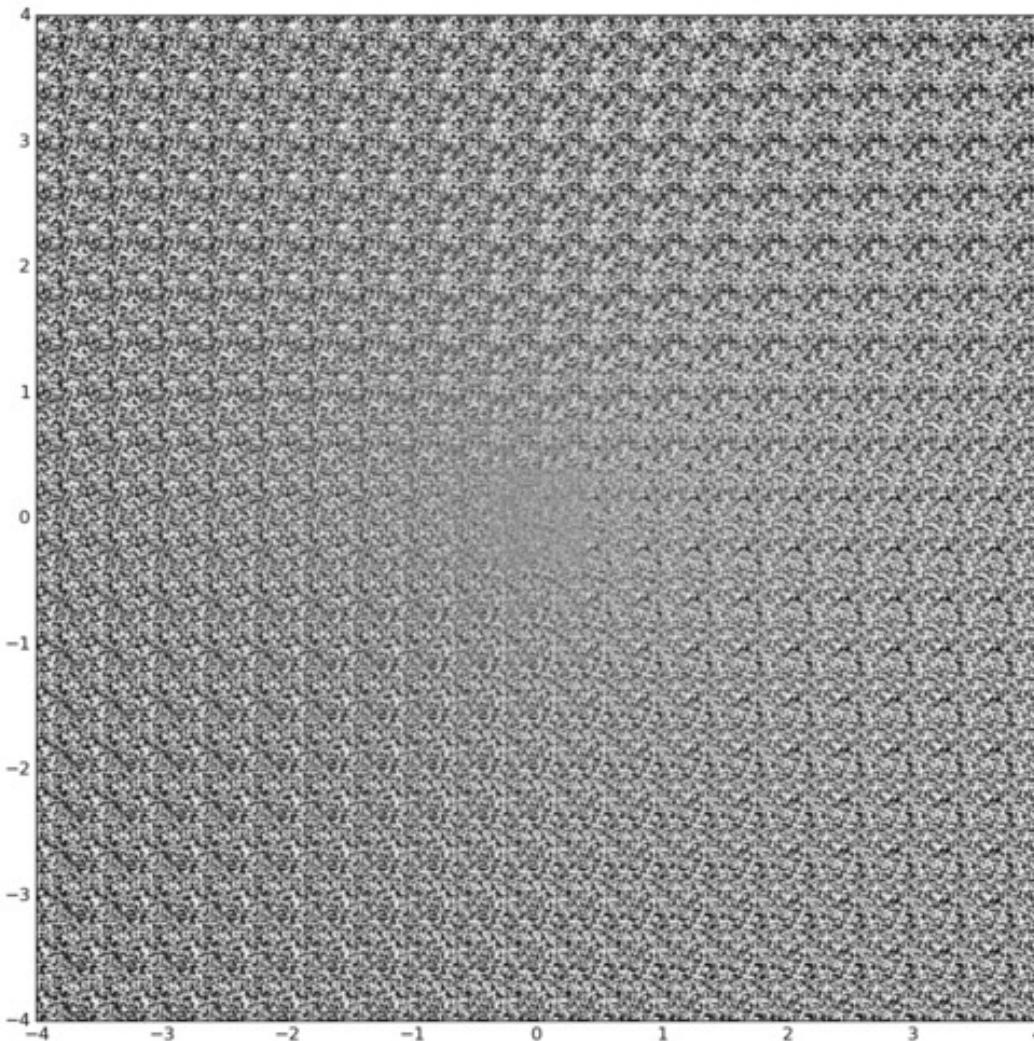
<https://blog.fastforwardlabs.com/2016/08/12/introducing-variational-autoencoders-in-prose-and-code.html>

VAE: 2D latent space on MNIST



To pdf users: this is animation. Check it on: “Introducing Variational Autoencoders (in Prose and Code)”
<https://blog.fastforwardlabs.com/2016/08/12/introducing-variational-autoencoders-in-prose-and-code.html>

VAE: 2D latent space on MNIST



To pdf users: this is animation. Check it on: “Introducing Variational Autoencoders (in Prose and Code)”
<https://blog.fastforwardlabs.com/2016/08/12/introducing-variational-autoencoders-in-prose-and-code.html>

VAE: 2D latent space on “Frey Face” dataset



Relation to Expectation-Maximization (EM)

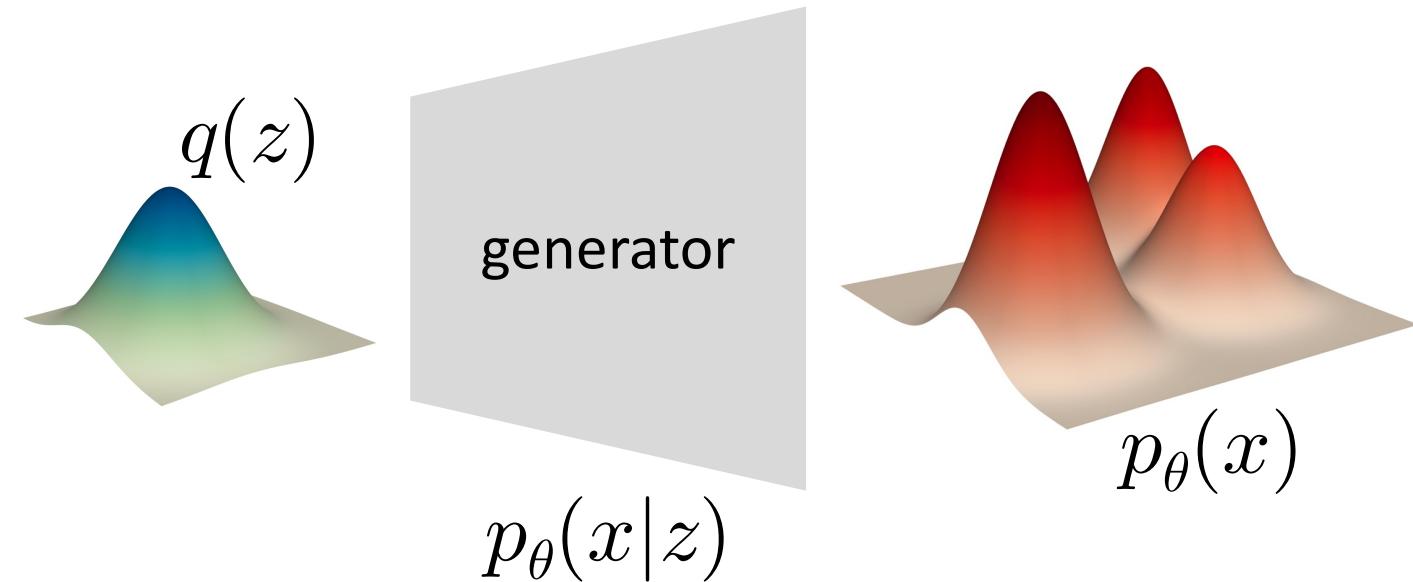
Recap: Latent Variable Models

Two sets of variables:

- q : distribution of latent
- θ : parameters of generator

VAE:

- parametrize q by a network
- stochastic gradient decent

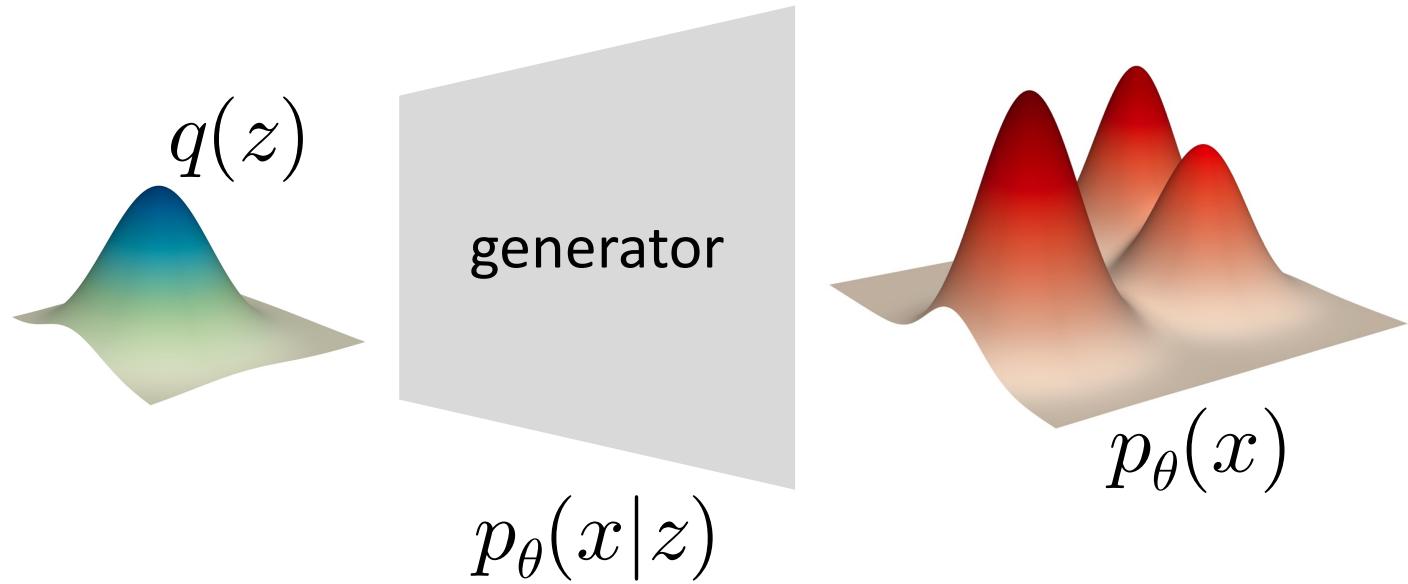


Expectation-Maximization (EM):

- often parametrize q analytically
- coordinate descent (i.e., alternating optimization)

EM as A Max-Max Procedure

$$\text{ELBO} = \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[\mathbb{E}_{z \sim q(z|x)} \left[\log p_{\theta}(x|z) \right] - \mathcal{D}_{\text{KL}}(q(z|x) || p(z)) \right]$$



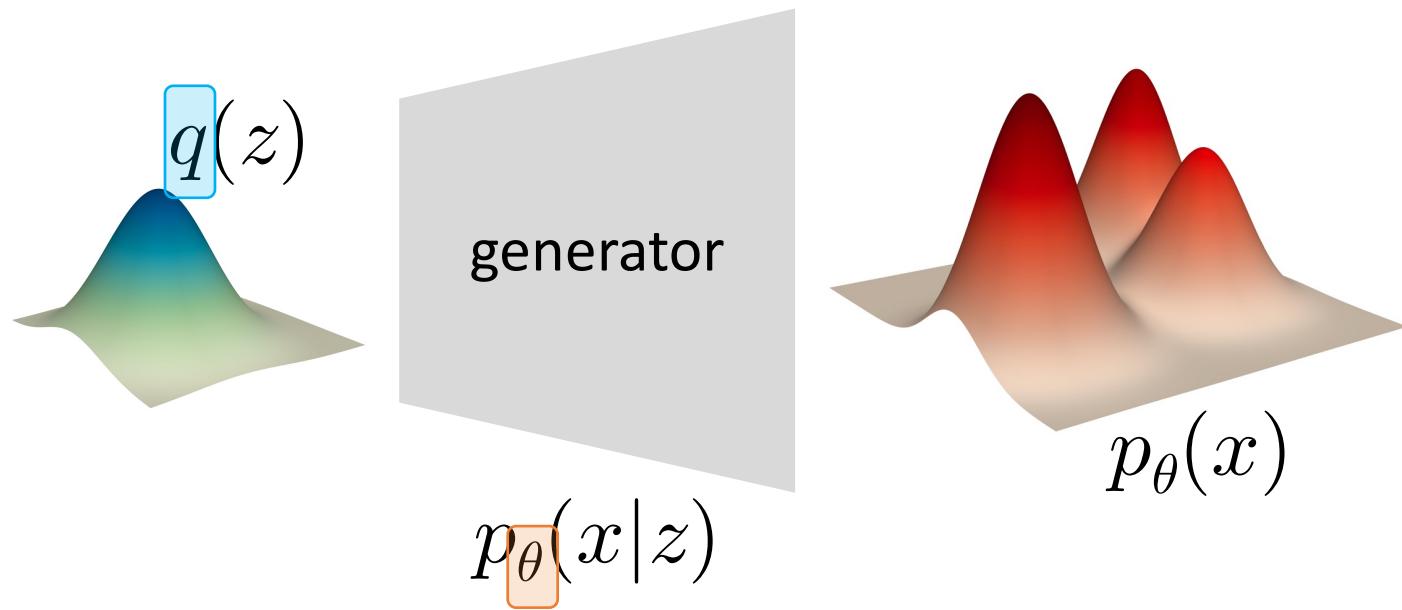
EM as A Max-Max Procedure

$$\text{ELBO} = \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[\mathbb{E}_{z \sim q(z|x)} \left[\log p_{\theta}(x|z) \right] - \mathcal{D}_{\text{KL}}(q(z|x) || p(z)) \right]$$

$$\max_{\theta, q} \text{ELBO}(\theta, q(\cdot))$$

Two sets of variables:

- q - distribution of latent
- θ - parameters of generator



Coordinate descent:

- max-max procedure (GAN: max-min)

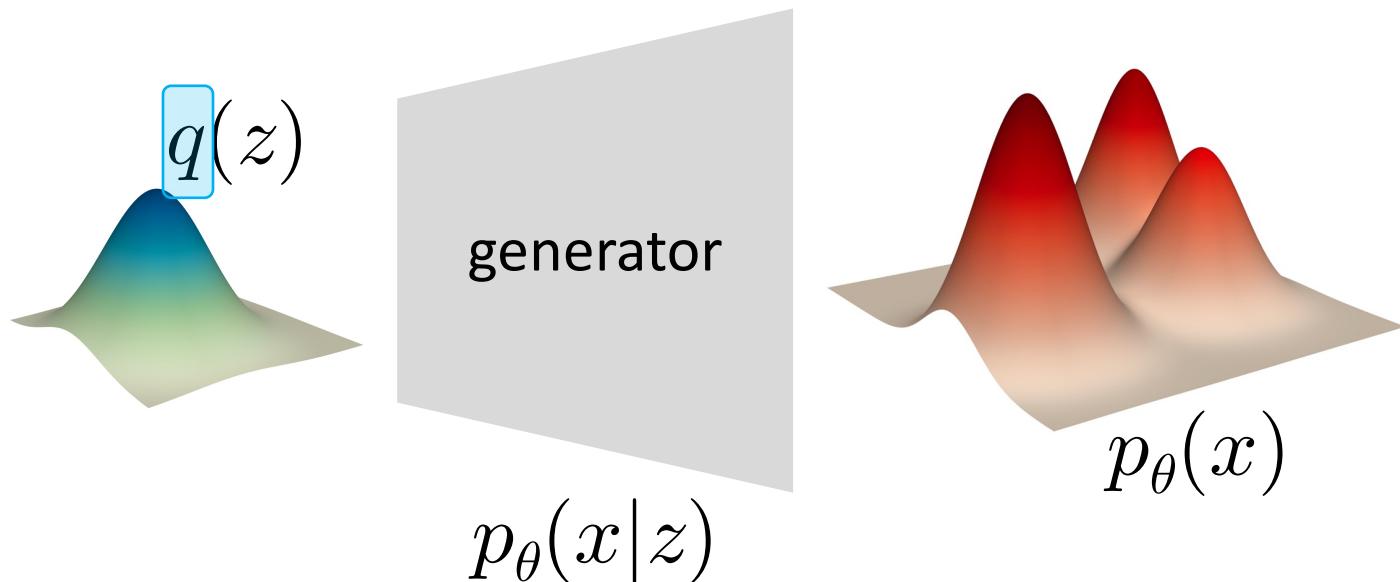
EM as A Max-Max Procedure

$$\text{ELBO} = \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[\mathbb{E}_{z \sim q(z|x)} \left[\log p_{\theta}(x|z) \right] - \mathcal{D}_{\text{KL}}(q(z|x) || p(z)) \right]$$

$$\max_{\theta, q} \text{ELBO}(\theta, q(\cdot))$$

E-step: optimize for q

$$q^{(t)} = p_{\theta^{(t)}}(z|x)$$



EM as A Max-Max Procedure

$$\text{ELBO} = \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[\mathbb{E}_{z \sim q(z|x)} \left[\log p_{\theta}(x|z) \right] - \mathcal{D}_{\text{KL}}(q(z|x) || p(z)) \right]$$

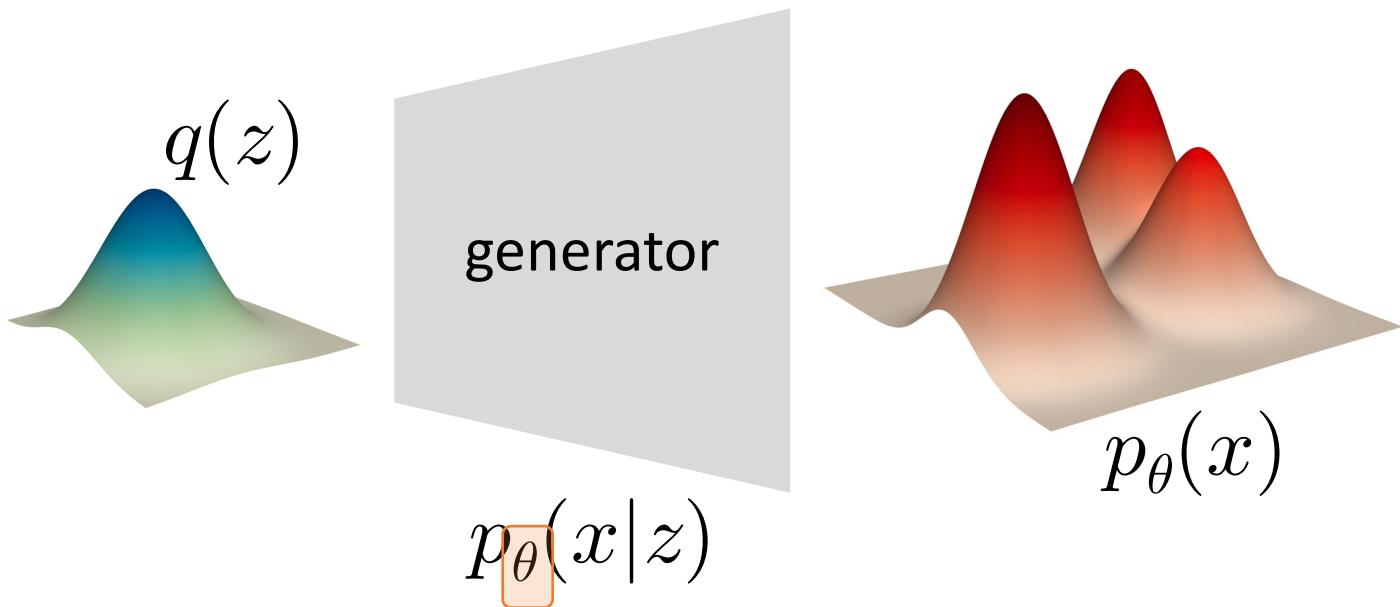
$$\max_{\theta, q} \text{ELBO}(\theta, q(\cdot))$$

E-step: optimize for q

$$q^{(t)} = p_{\theta^{(t)}}(z|x)$$

M-step: optimize for θ

$$\theta^{(t+1)} = \arg \max_{\theta} Q(\theta|\theta^{(t)})$$



with sub-objective defined as: $Q(\theta|\theta^{(t)}) = \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{p_{\theta^{(t)}}(z|x)} [\log p_{\theta}(x, z)]$

EM as A Max-Max Procedure

$$\text{ELBO} = \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[\mathbb{E}_{z \sim q(z|x)} \left[\log p_{\theta} \right] \right]$$

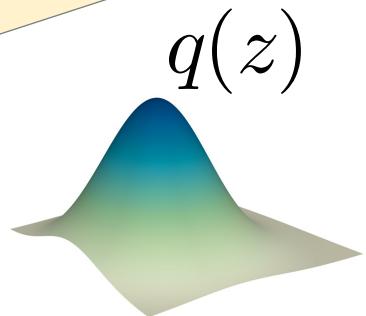
$$\max_{\theta, q} \text{ELBO}(\theta, q(\cdot))$$

q : often in analytical forms

- Gaussian Mixtures
- K-means

E-step: optimize for q

$$q^{(t)} = p_{\theta^{(t)}}(z|x)$$

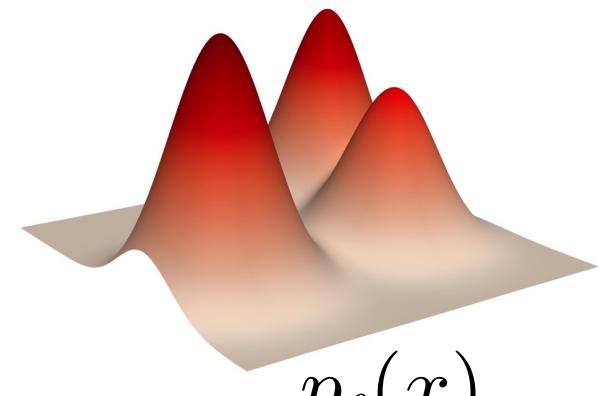


generator

M-step: optimize for θ

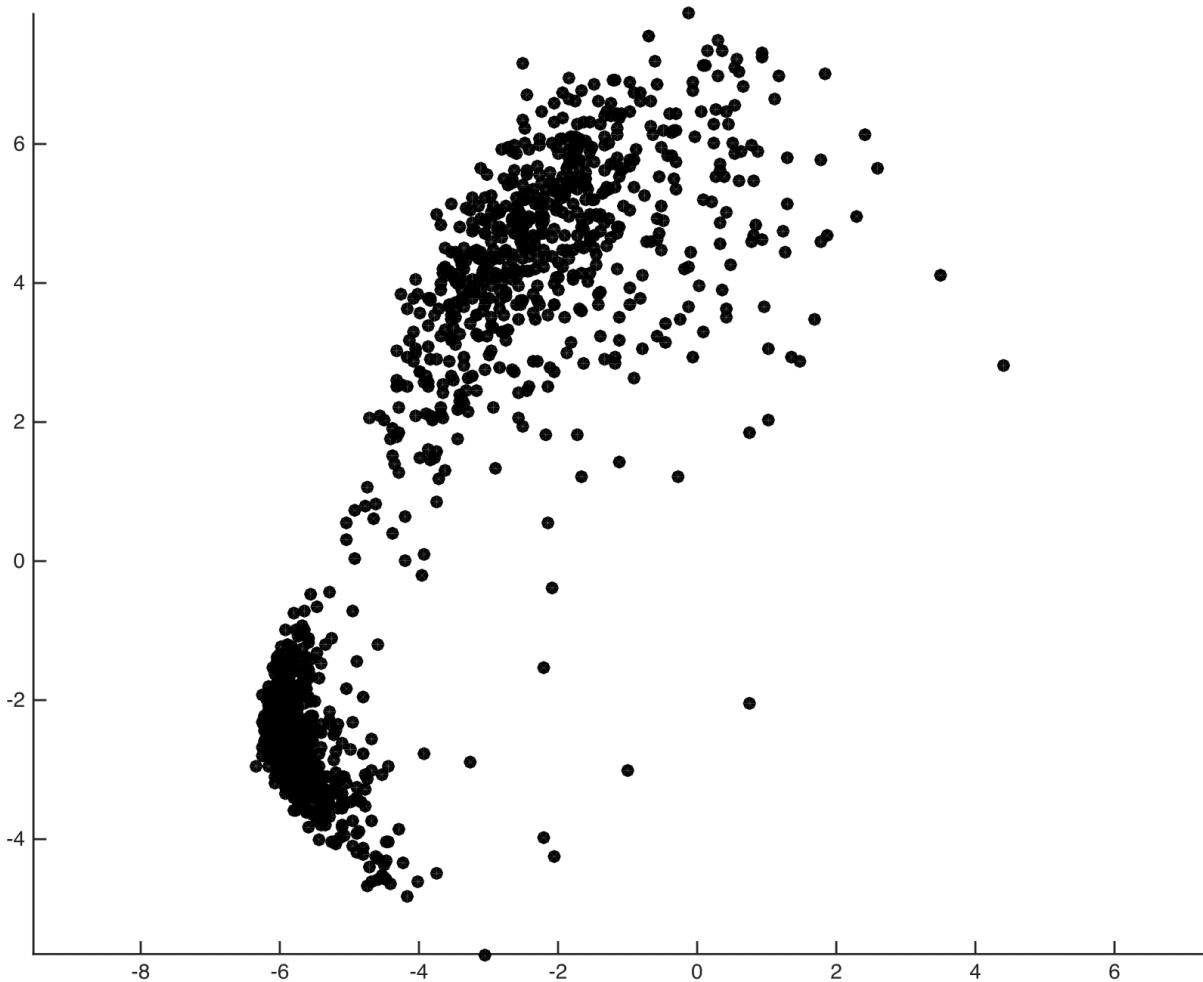
$$\theta^{(t+1)} = \arg \max_{\theta} Q(\theta|\theta^{(t)})$$

$$p_{\theta}(x|z)$$



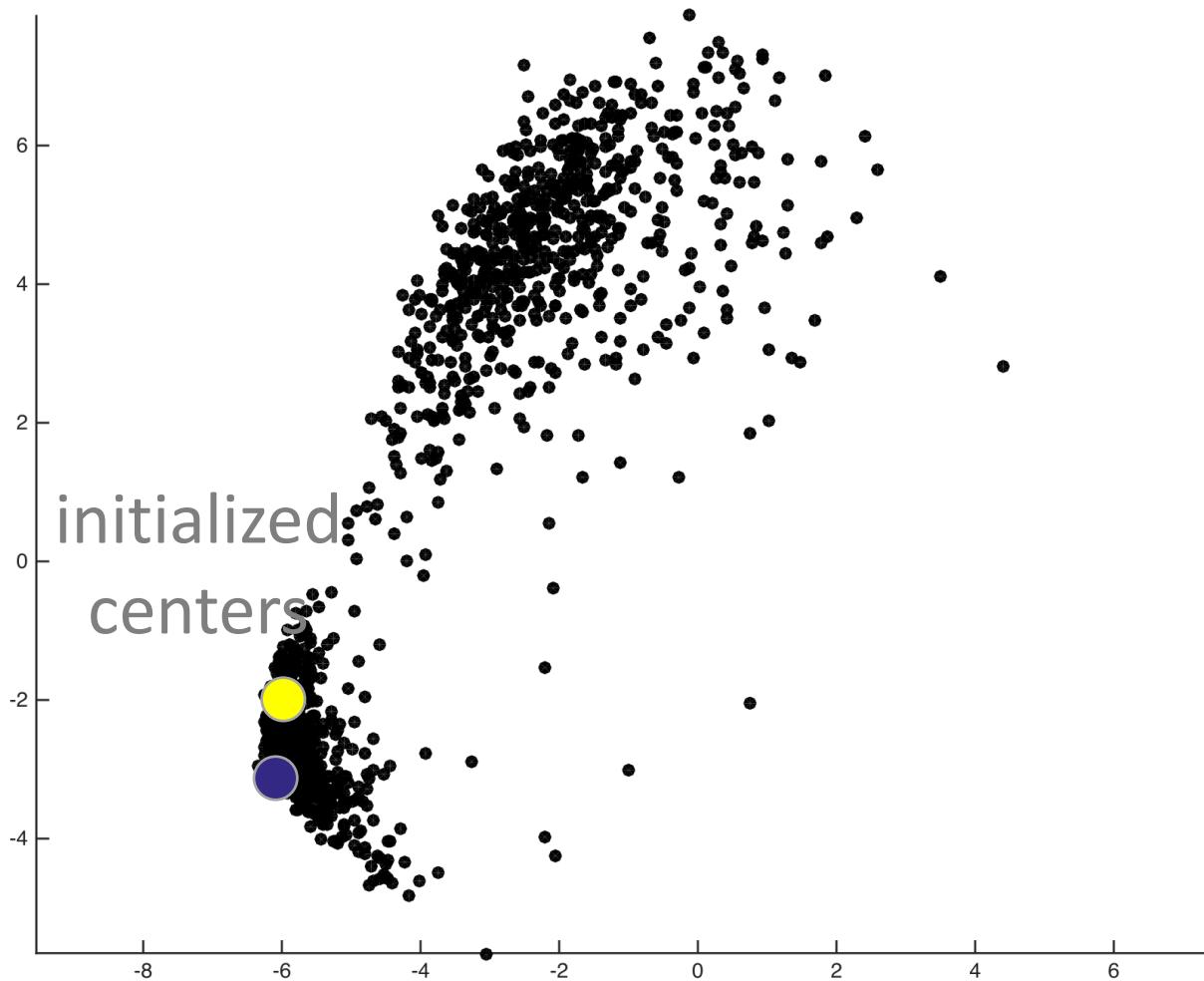
with sub-objective defined as: $Q(\theta|\theta^{(t)}) = \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{p_{\theta^{(t)}}(z|x)} [\log p_{\theta}(x, z)]$

A running example of EM: K-means



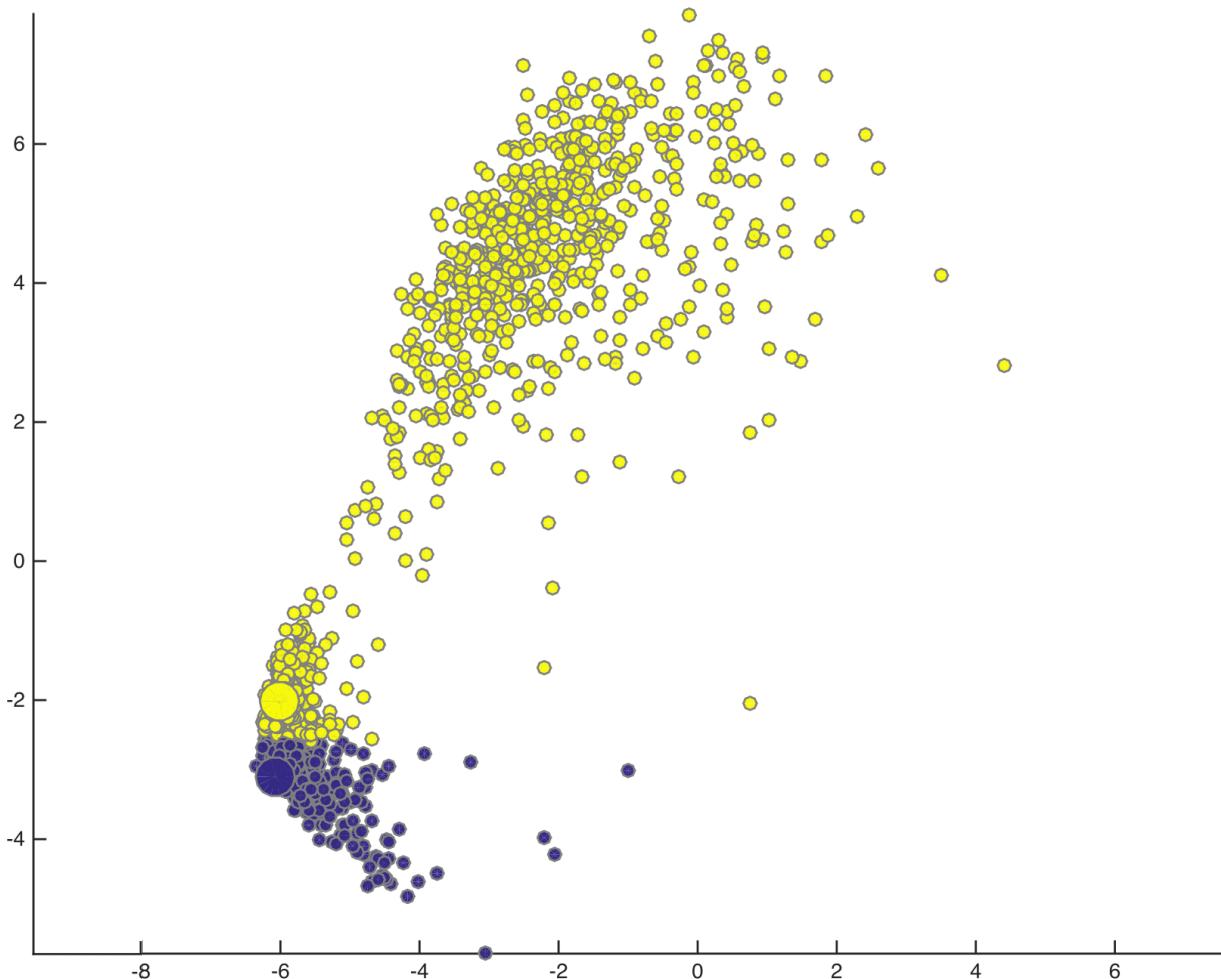
A running example of EM: K-means

- cluster centers: θ



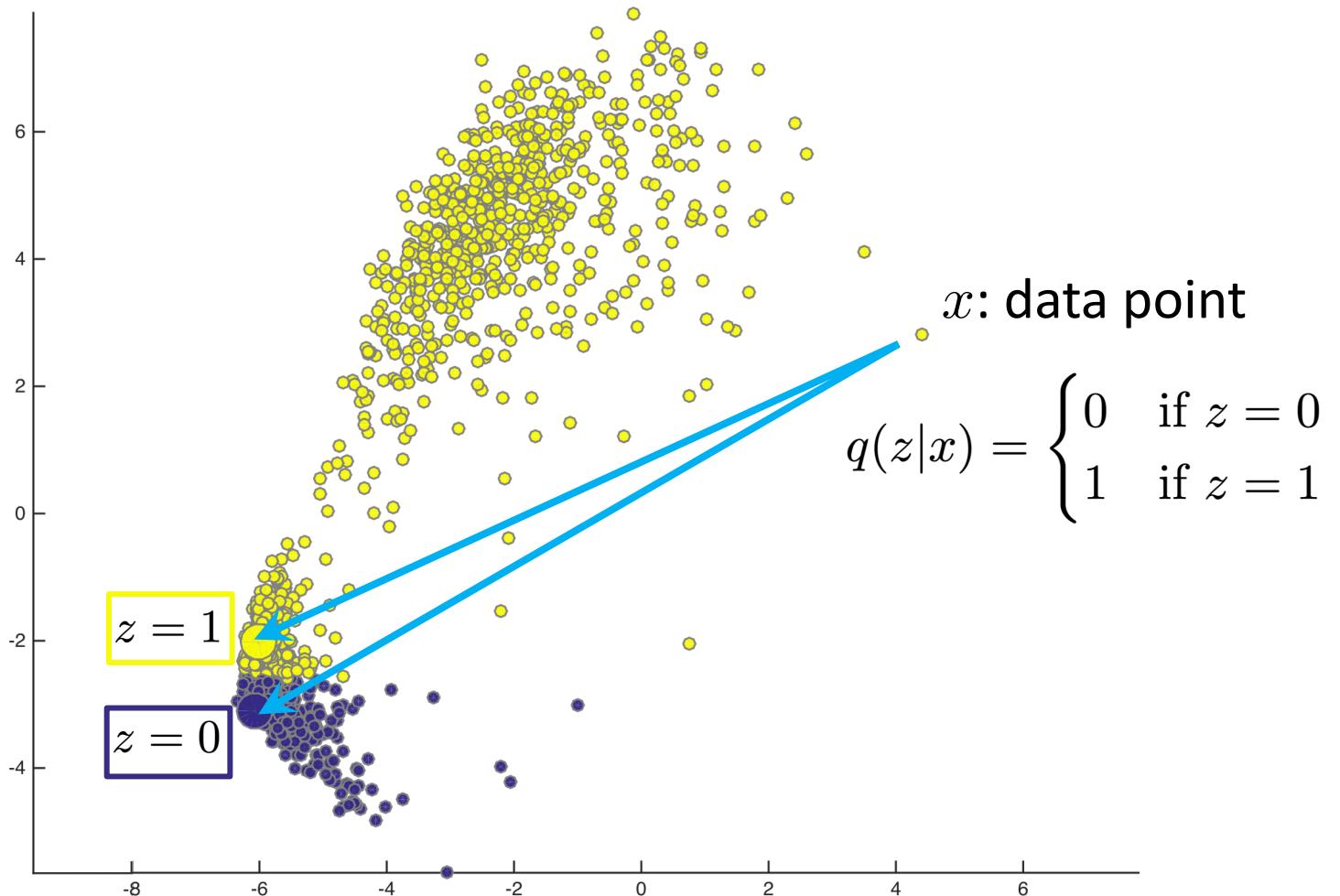
A running example of EM: K-means

- cluster centers: θ
- assignment:



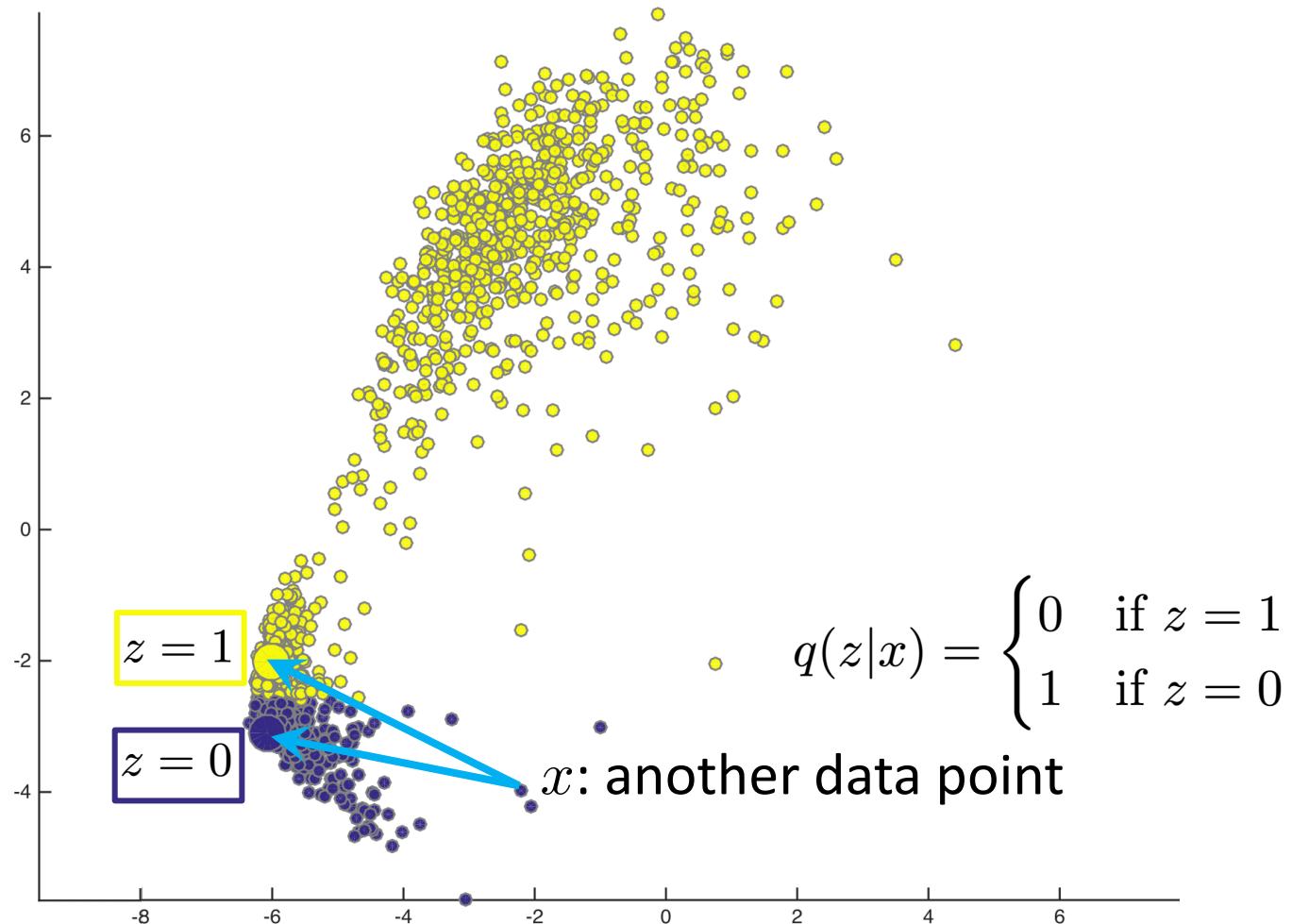
A running example of EM: K-means

- cluster centers: θ
- assignment: E-step



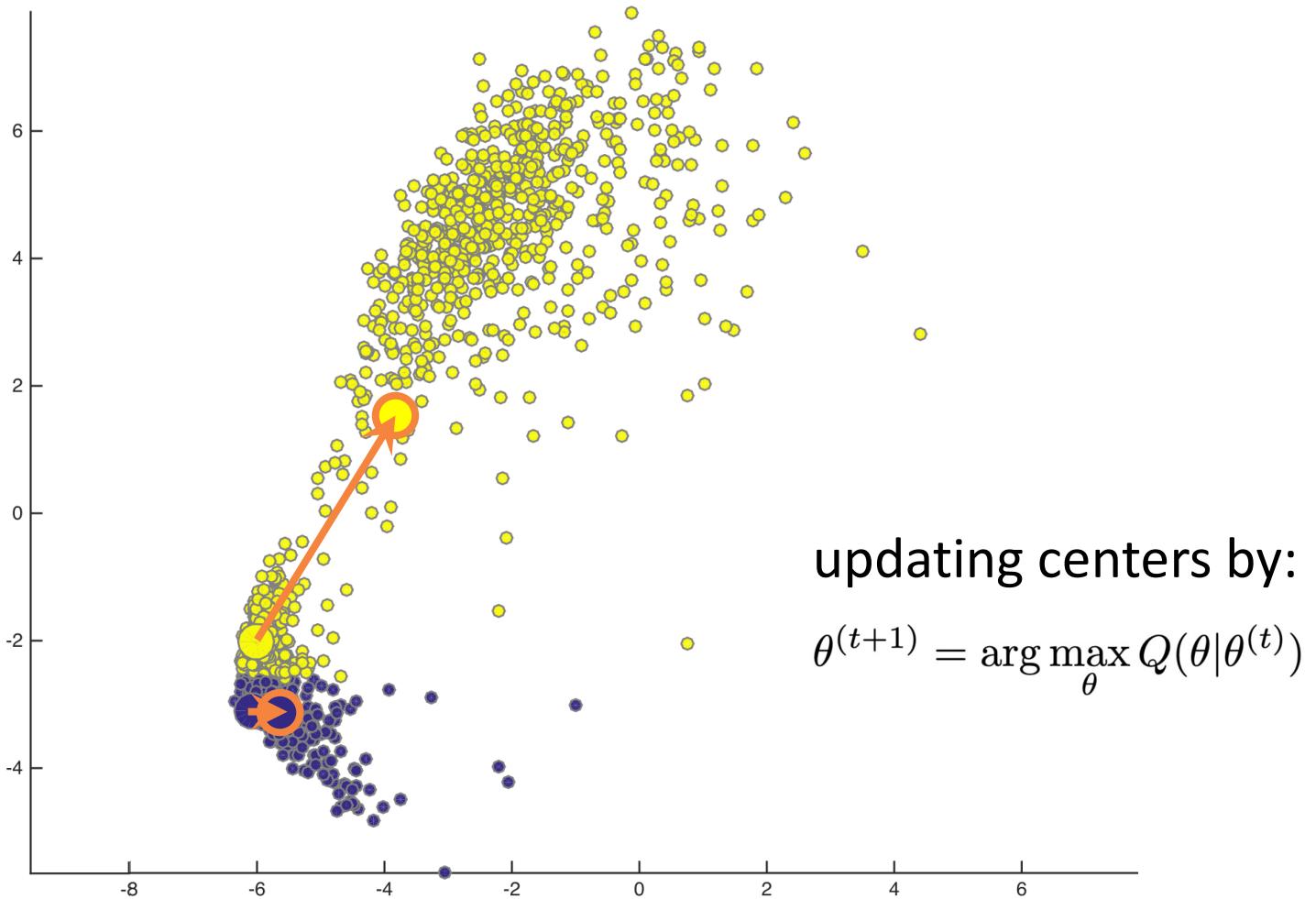
A running example of EM: K-means

- cluster centers: θ
- assignment: E-step



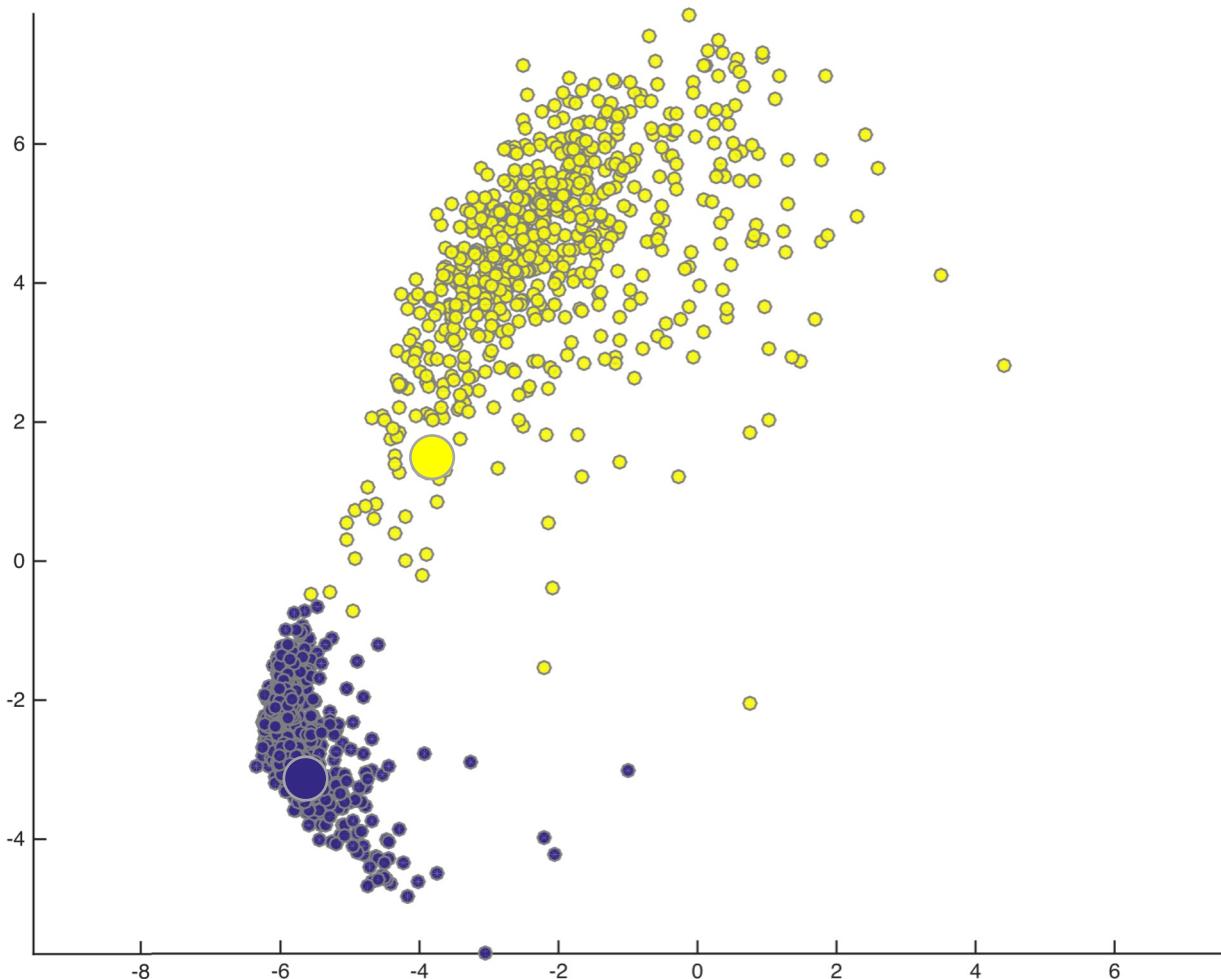
A running example of EM: K-means

- cluster centers: θ
- assignment: E-step
- update: M-step

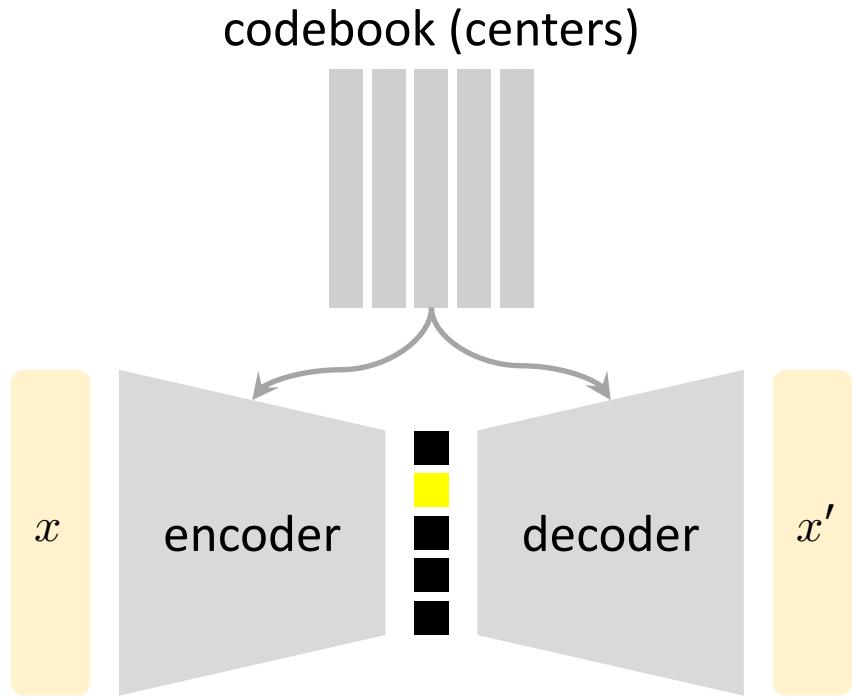


A running example of EM: K-means

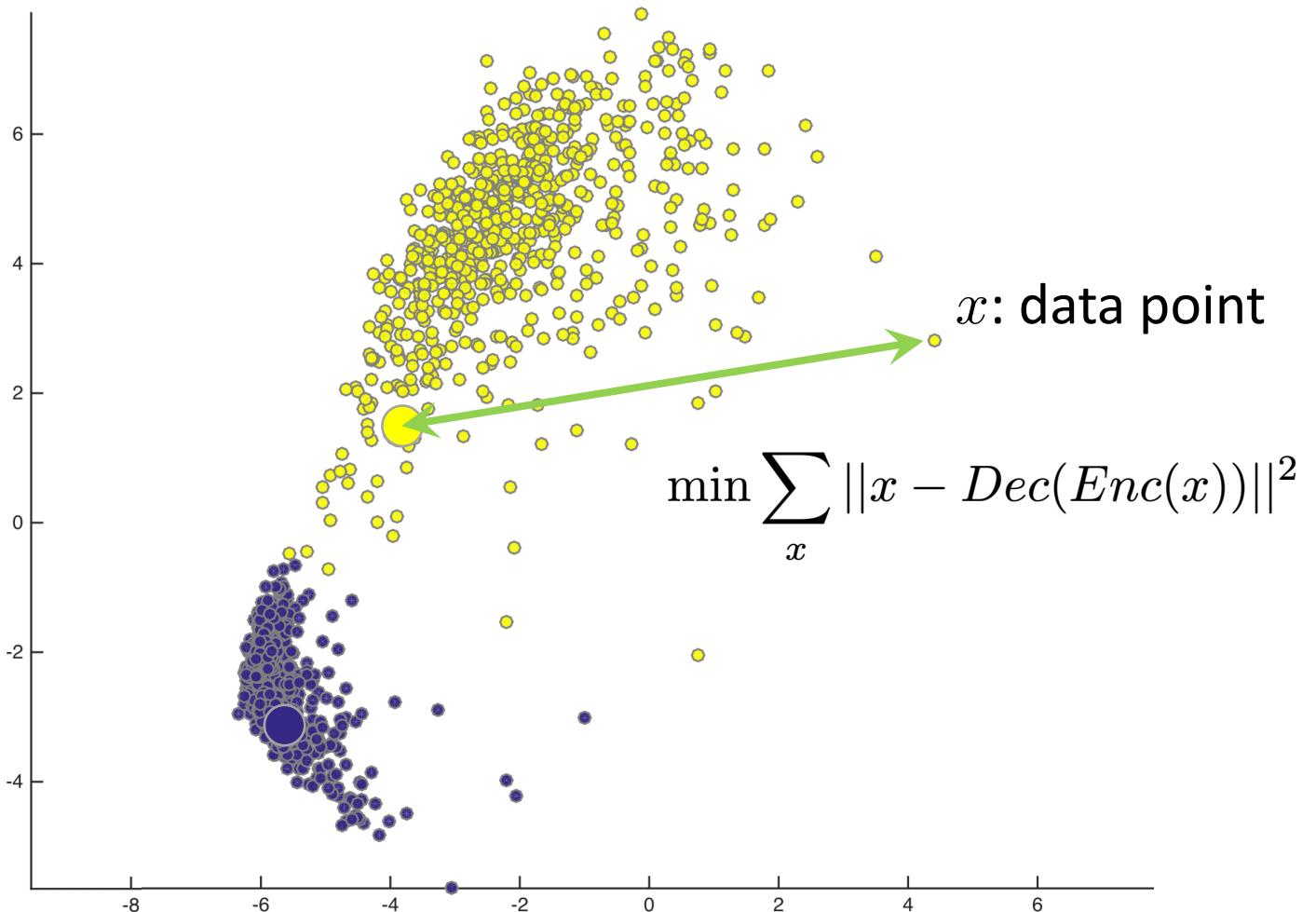
- cluster centers: θ
- assignment: E-step
- update: M-step



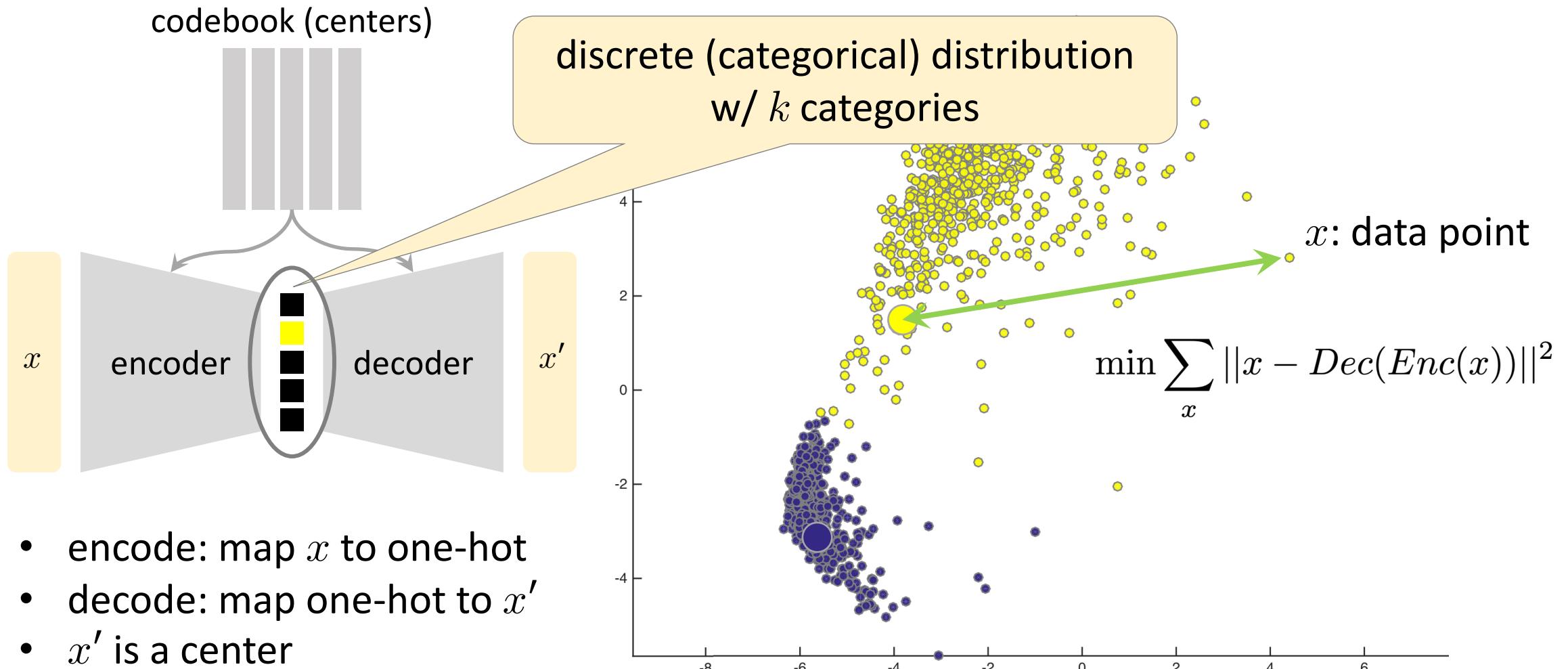
K-means as Autoencoder



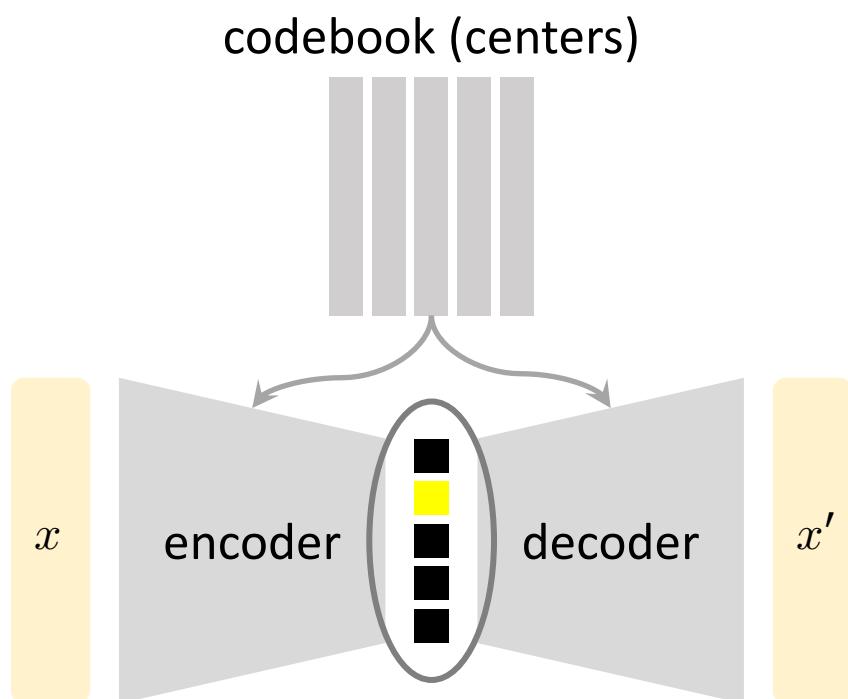
- encode: map x to one-hot
- decode: map one-hot to x'
- x' is a center



K-means as Autoencoder

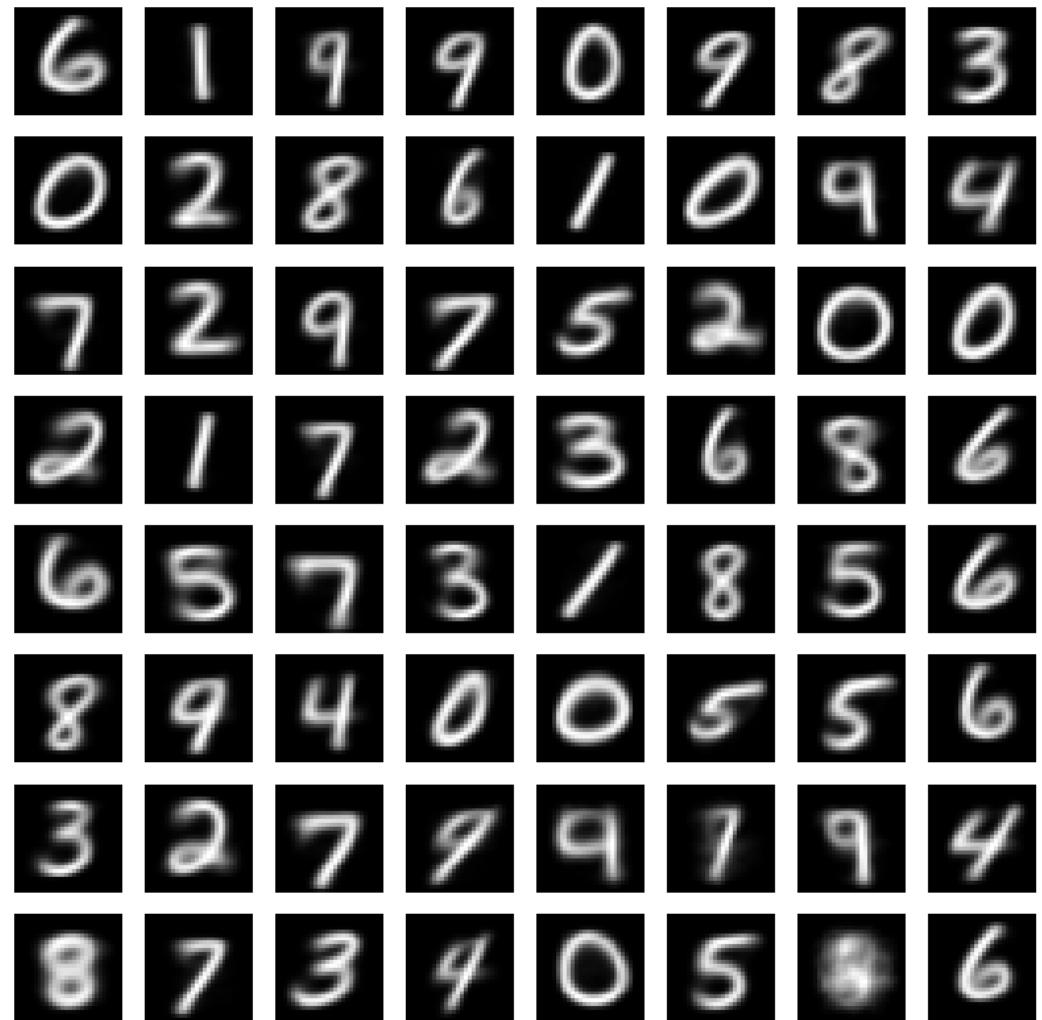


K-means as Autoencoder



- encode: map x to one-hot
- decode: map one-hot to x'
- x' is a center

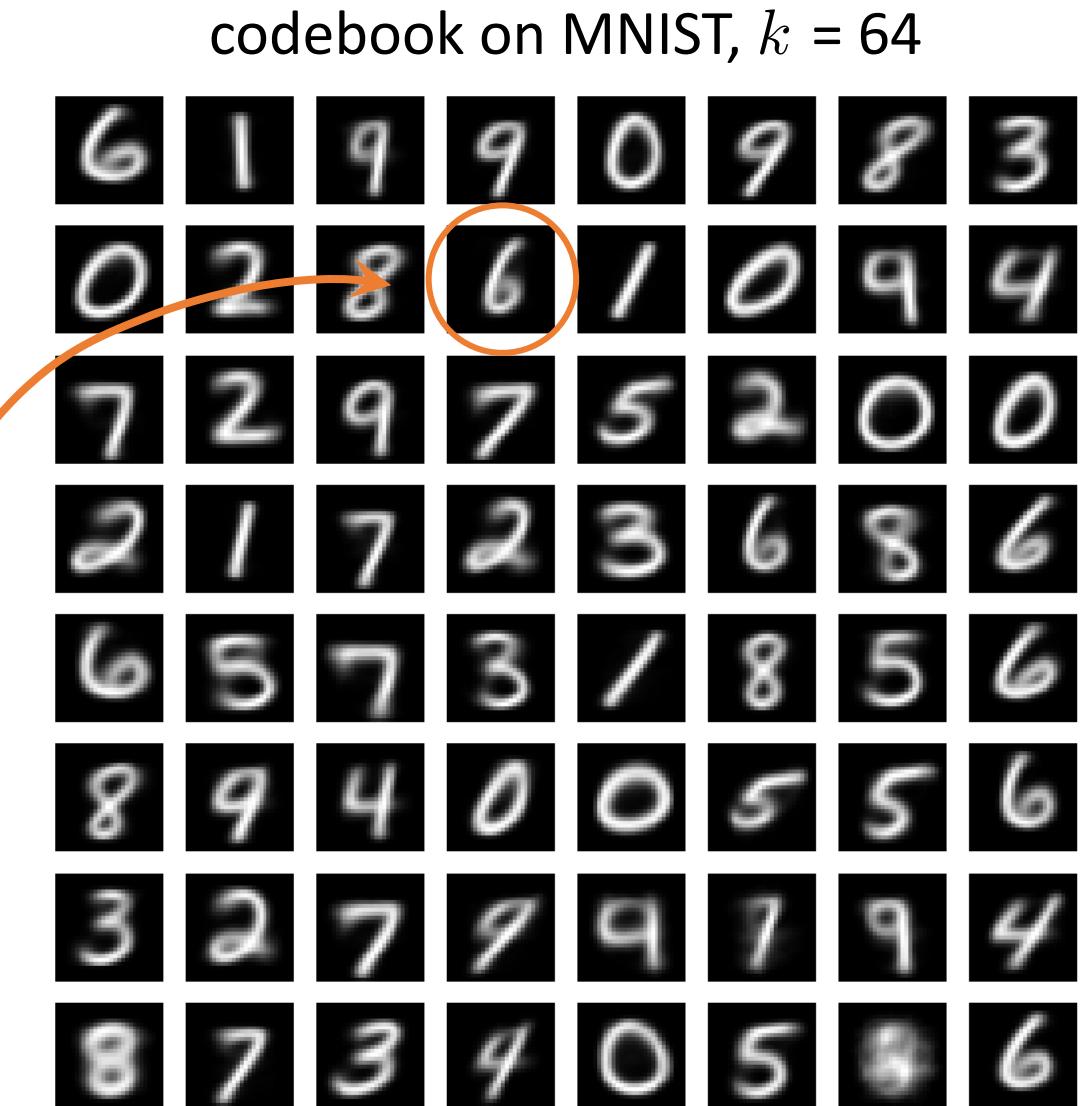
codebook on MNIST, $k = 64$



K-means as Generative Models

- randomly sample: $z \sim \mathcal{U}[0, k)$
- map z by the decoder
- generation result is one codeword

$z = 11$



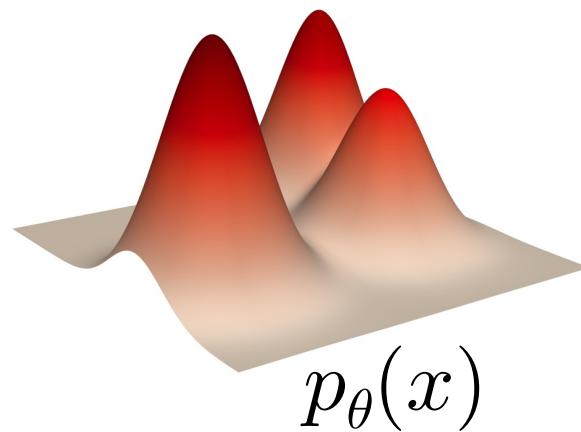
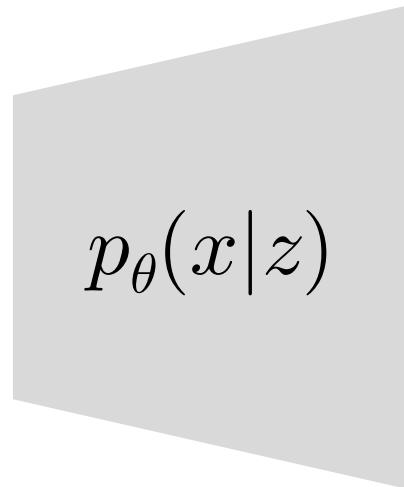
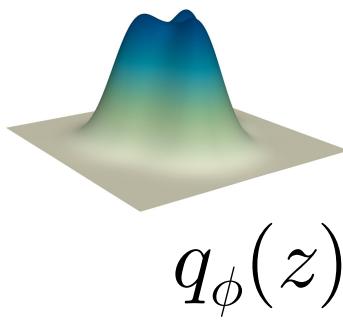
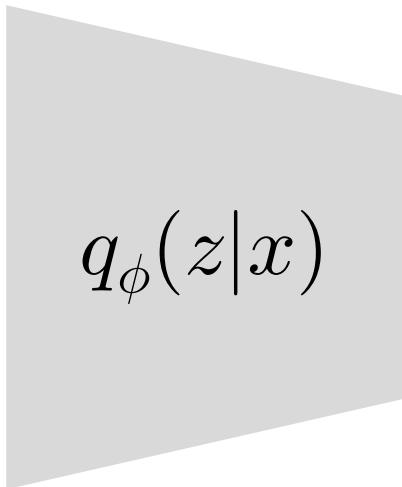
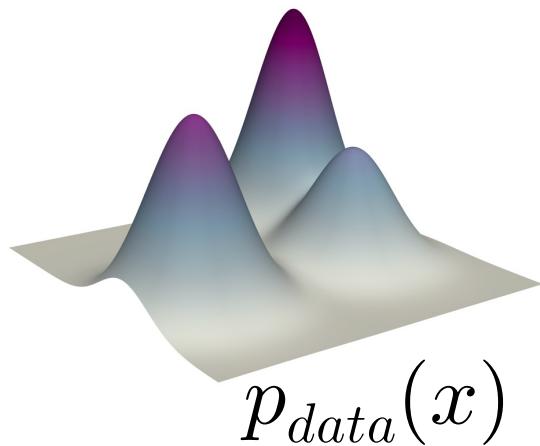
thus far, ...

- **VAE:** maximize ELBO
 - parameterize q by network
 - optimize by Stochastic Gradient Descent
- **EM:** maximize ELBO
 - parameterize q analytically
 - optimize by Coordinate Descent
- **K-means:**
 - special case of EM; special case of AE
 - discrete distribution
- next: **VQ-VAE**

Vector Quantized VAE (VQ-VAE)

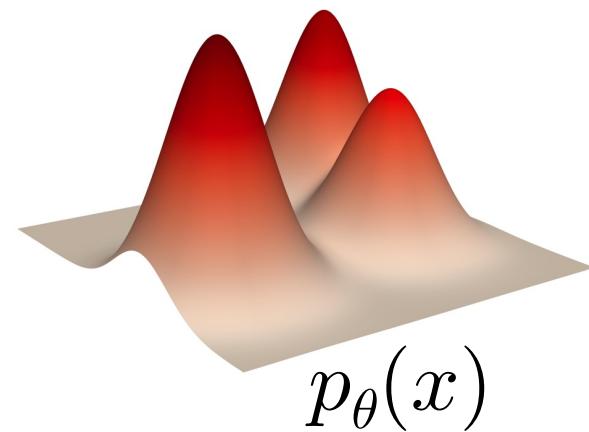
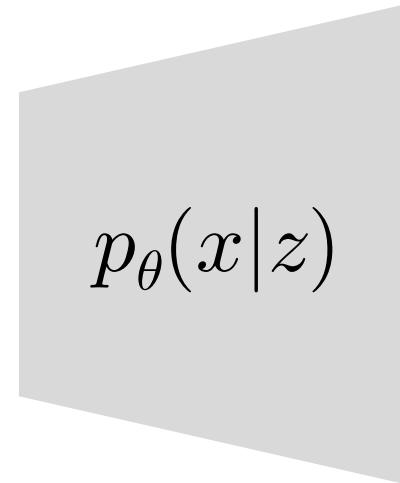
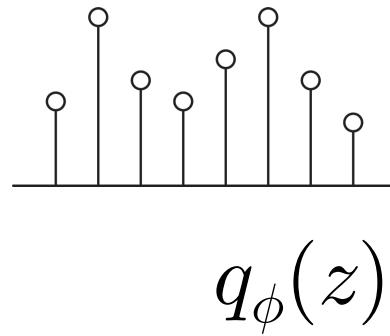
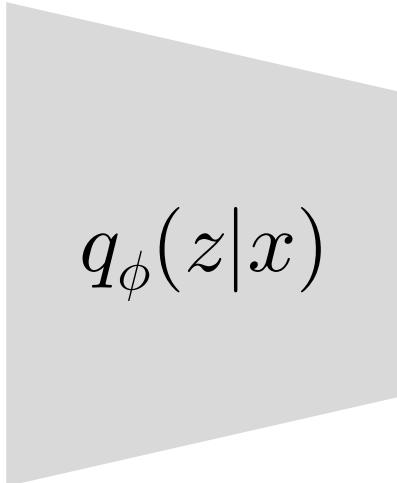
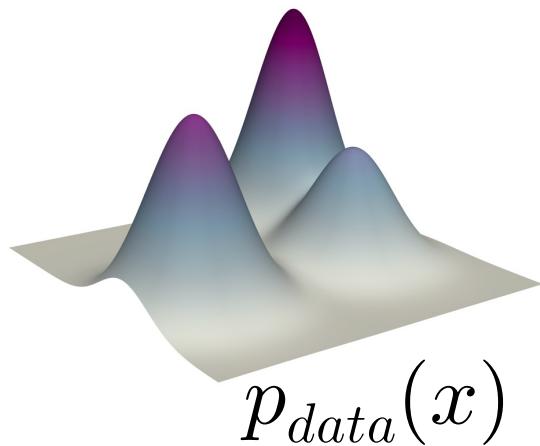
Recap

- Original VAE: latent variables are continuous



Discrete Latent Variables

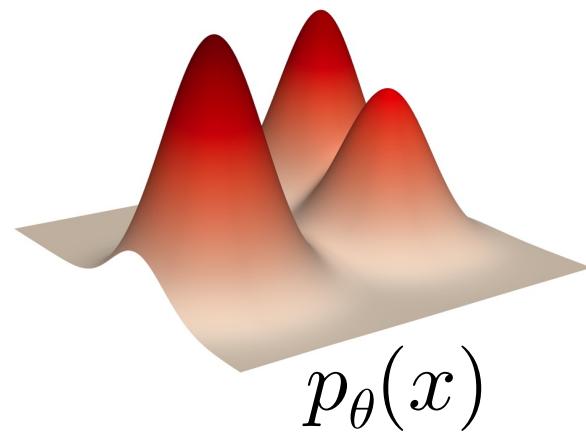
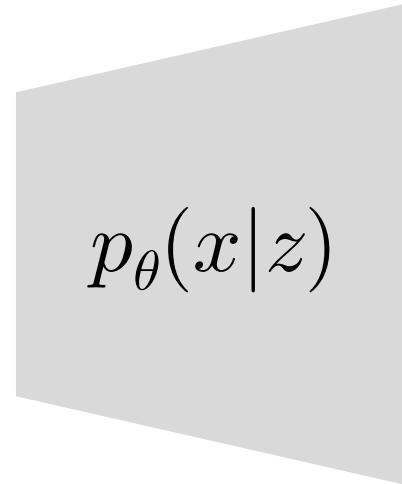
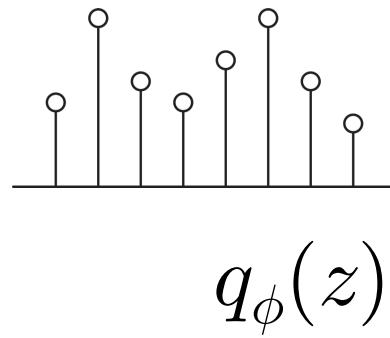
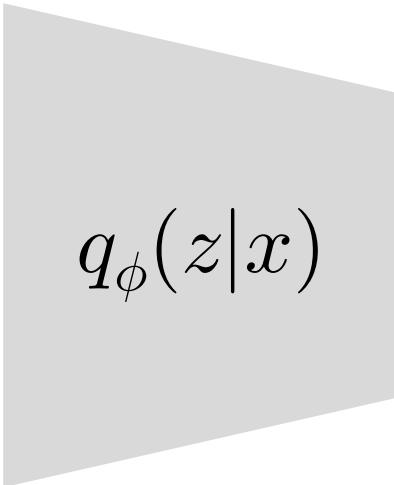
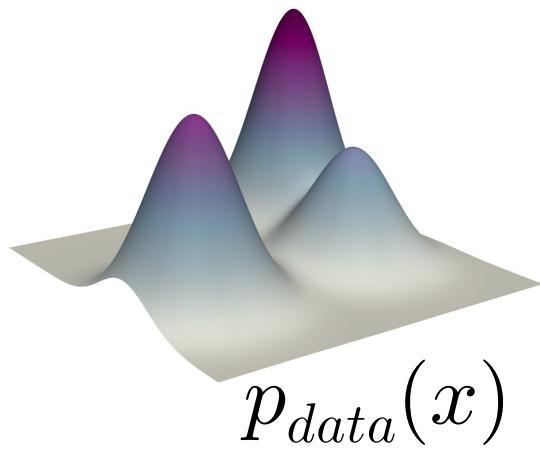
- model **multimodal** distributions
- **categorical**: no particular relation between numbers (SSN, zip code, ...)
- **symbolic**: language, speech, planning, ...



Discrete Latent Variables + VAE

Maximize ELBO

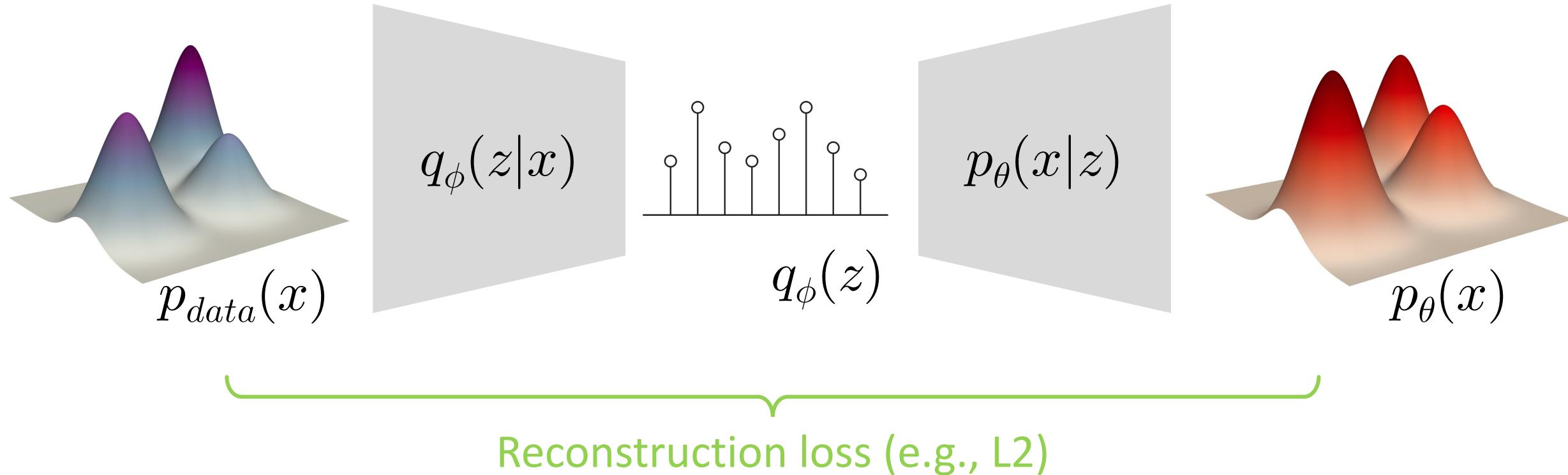
- Reconstruction loss: about x
- Regularization loss: about z (discrete)



Discrete Latent Variables + VAE

Reconstruction loss: about x

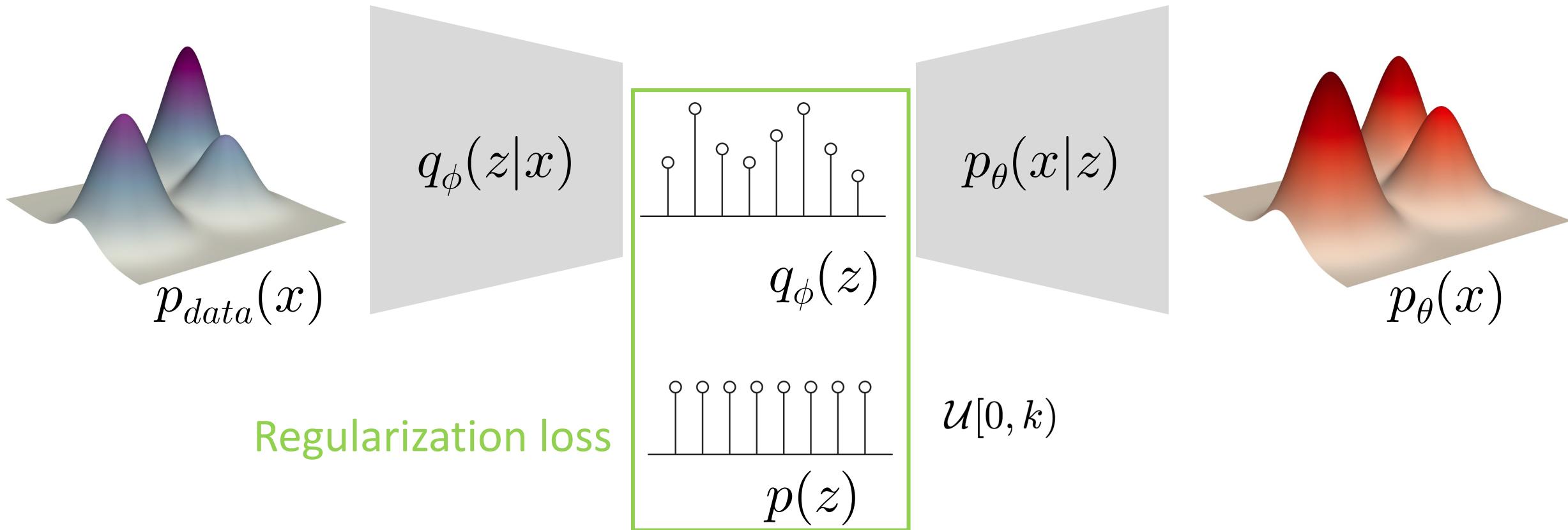
- same as VAE: $-\mathbb{E}_{z \sim q_\phi(z|x)} [\log p_\theta(x|z)]$



Discrete Latent Variables + VAE

Regularization loss: about z

- conceptually, same as VAE: $\mathcal{D}_{\text{KL}}(q_{\phi}(z|x) || p(z))$
- but how can we backprop w.r.t. discrete sampling?



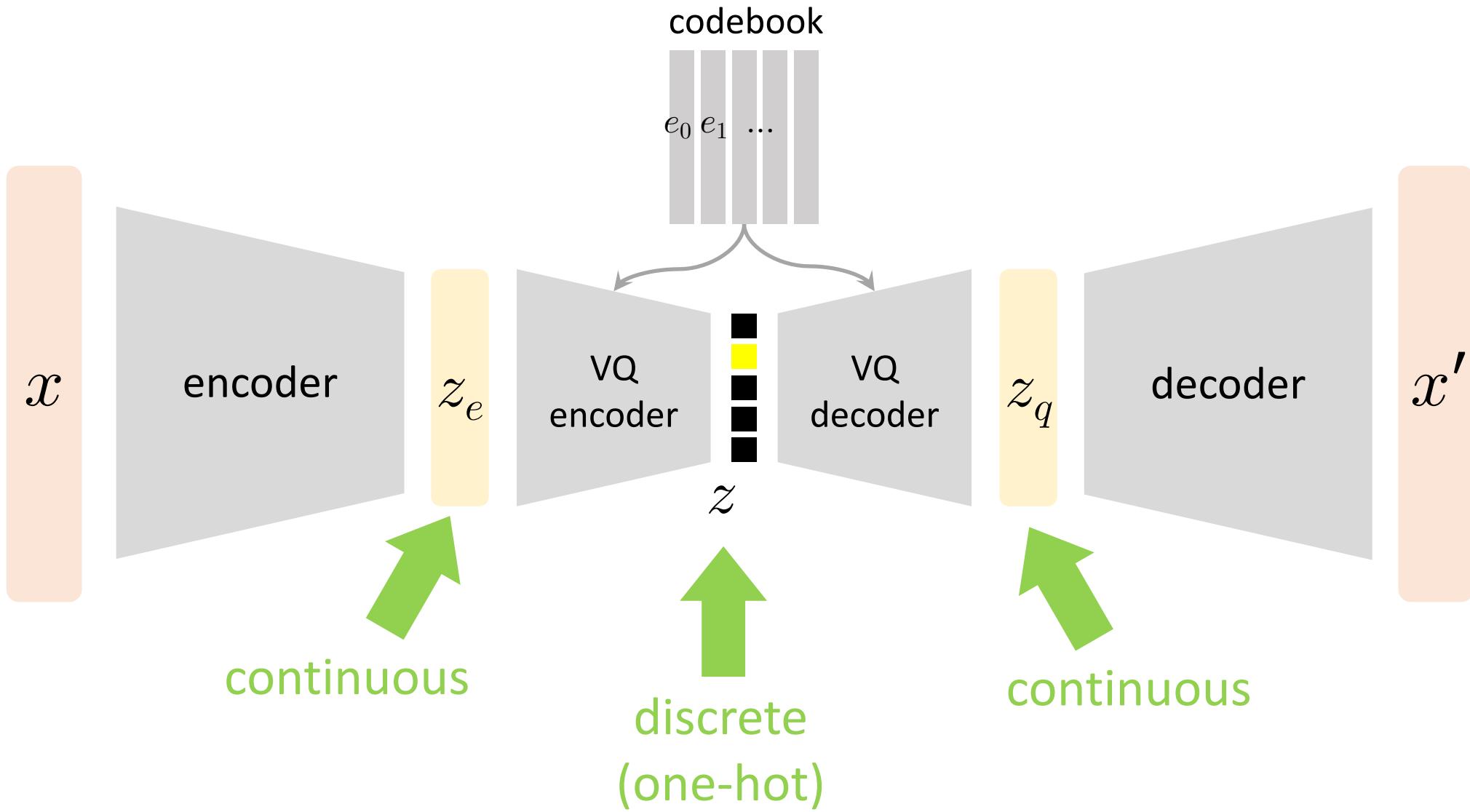
Discrete Latent Variables + VAE

Solution: K-means

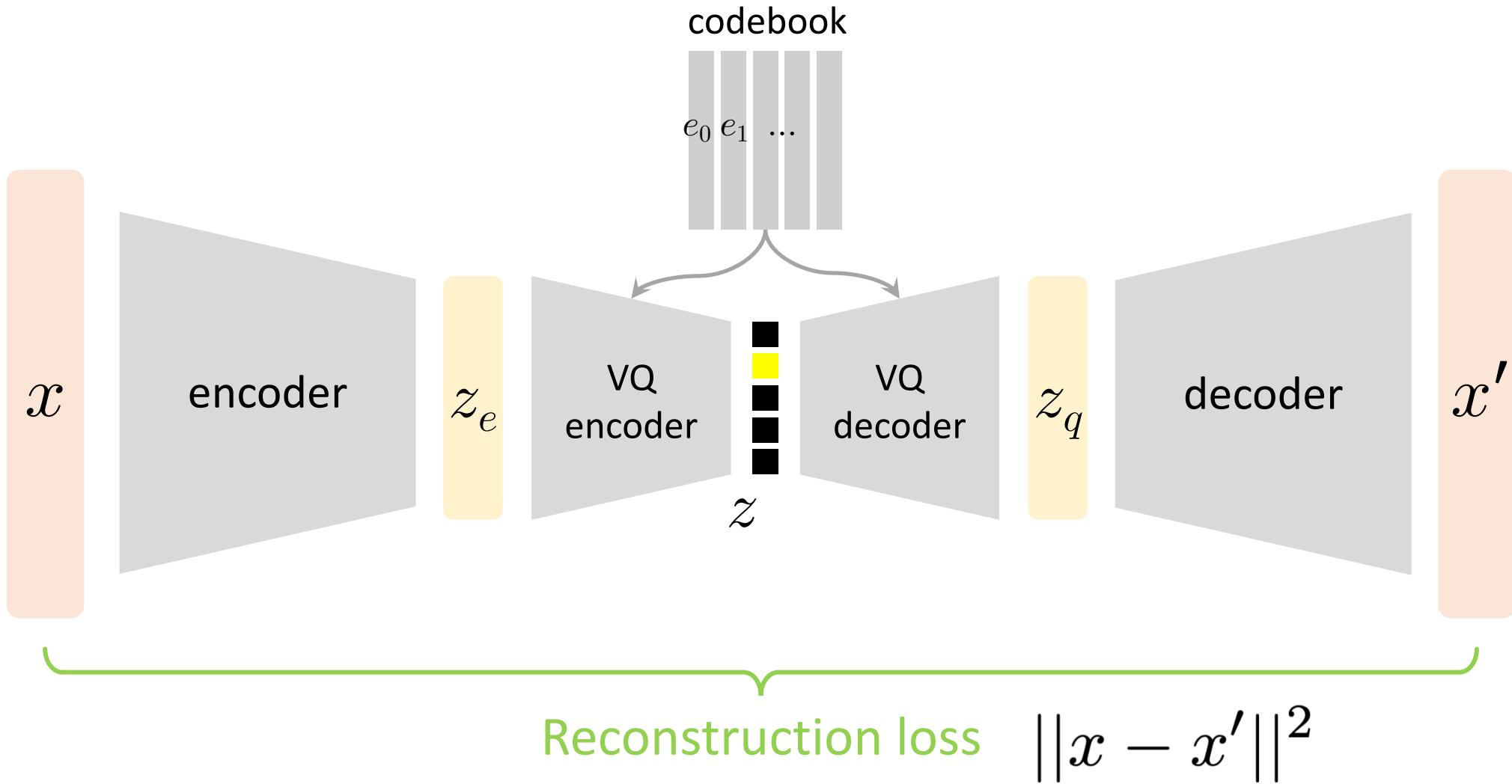
- K-means is autoencoding
- K-means has an objective function (reconstruction loss)
- K-means implicitly encourages codebook uniformity

This leads us to VQ-VAE ...

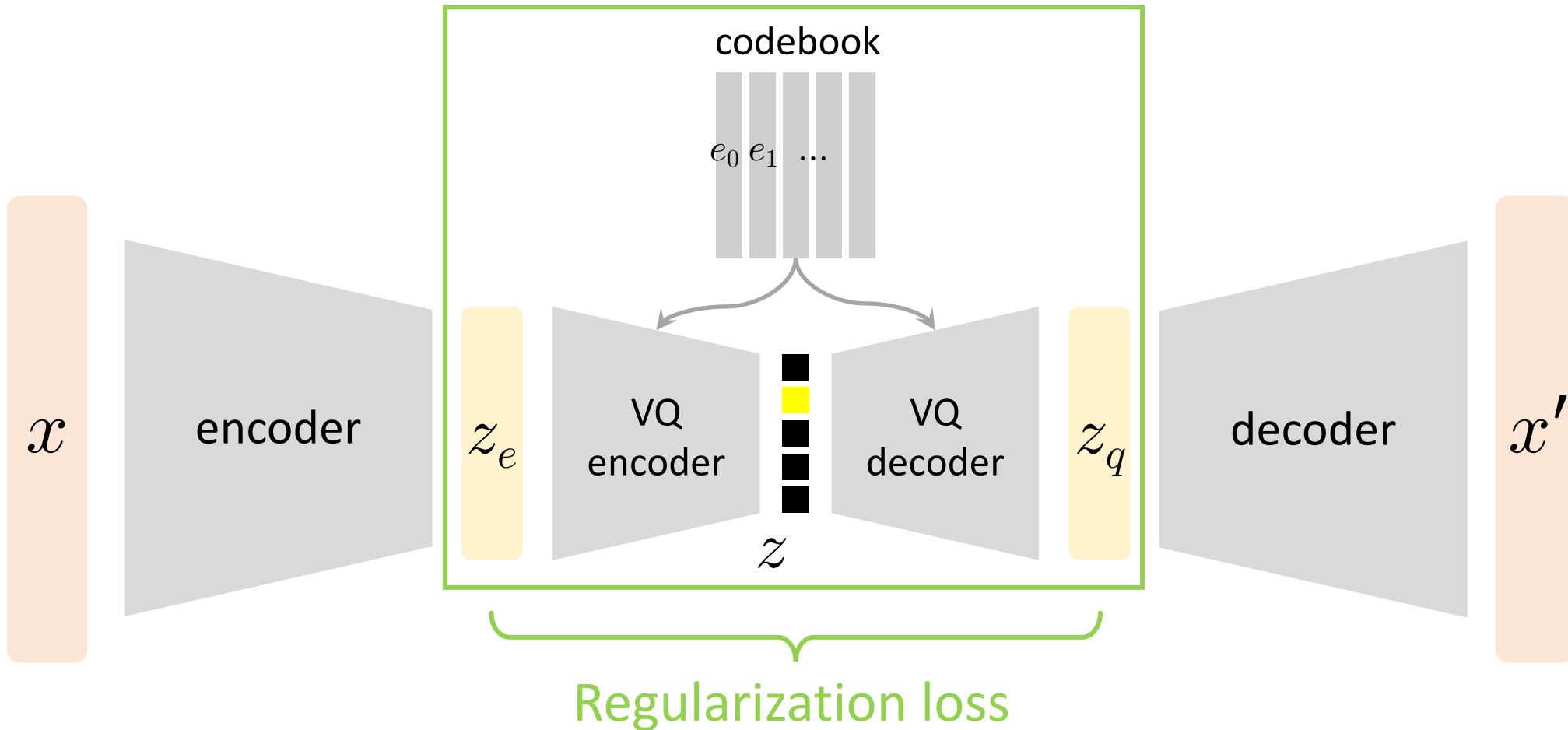
Vector Quantized VAE



Vector Quantized VAE



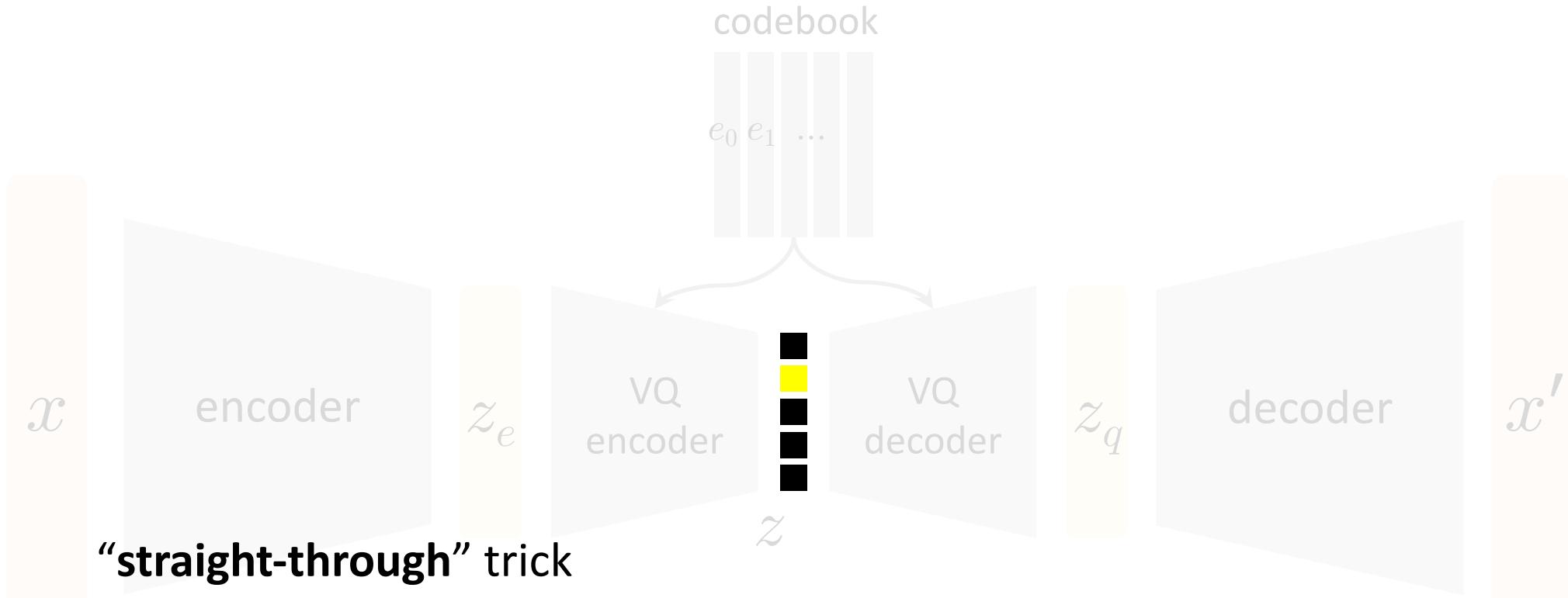
Vector Quantized VAE



conceptually, this is the K-means reconstruction loss: $\|z_e - z_q\|^2$

*The VQ-VAE paper uses $\|\text{sg}[z_e(x)] - e\|_2^2 + \beta \|z_e(x) - \text{sg}[e]\|_2^2$ which weights the gradients differently

How to backprop through one-hot vector?



“straight-through” trick

- forward: hardmax’s output (i.e., argmax and one-hot)
- backward: softmax’s gradient
- in code: `stop_grad(hardmax(y) - softmax(y)) + softmax(y)`

Vector Quantized VAE

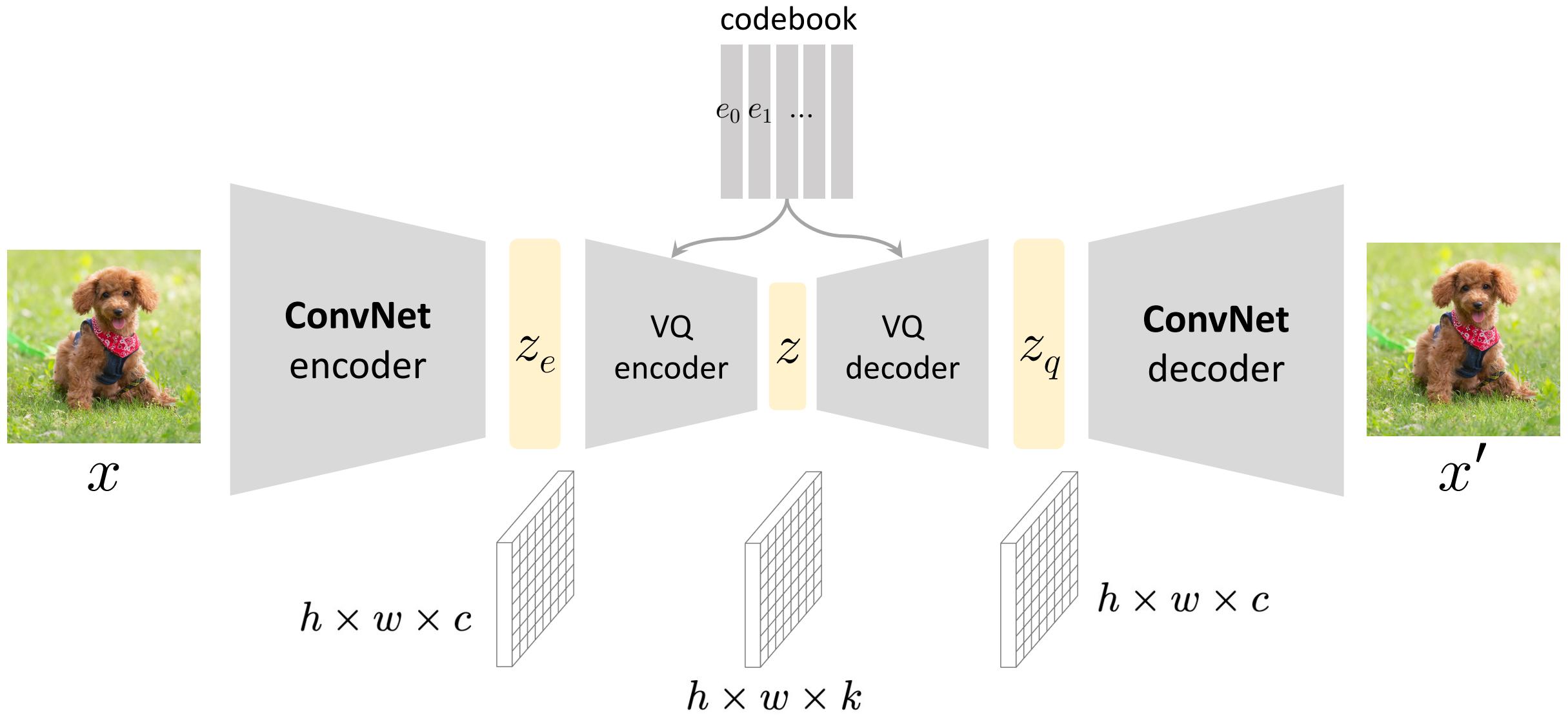
A single one-hot latent is not useful

- it's "deep K-means": with deep encoder/decoder
- a valid generative model; but not a "good" one

VQ-VAE: often used as "**tokenizers**"

- output multiple one-hot vectors
- don't reduce latent spatial/temporal size to 1
- use ConvNet/Transformer as encoder and decoder

VQ-VAE as Tokenizers



Notes

- Both VAE and VQ-VAE can be “tokenizers” (produce spatial latents).

But:

- prior $p(z)$ only models per-token (per-location) distribution
- prior $p(z)$ doesn’t model **joint** distribution across tokens
- spatial tokens are not **independent**
- at inference, we can’t sample from **i.i.d.** prior $p(z)$

Next: modeling joint distribution:

- Autoregressive models
- Masked models
- Diffusion models

This Lecture

- Variational Autoencoder (VAE)
- Relation to Expectation-Maximization (EM)
- Vector Quantized VAE (VQ-VAE)

Main References

- Kingma and Welling. “Auto-Encoding Variational Bayes”, ICLR 2014
- Neal and Hinton. “A view of the EM algorithm that justifies incremental, sparse, and other variants”, 1999
- Hastie, et al. “The Elements of Statistical Learning”, 2001
- van den Oord, et al. “Neural Discrete Representation Learning”, NeurIPS 2017