Lecture 2

Variational Autoencoder

6.S978 Deep Generative Models

Kaiming He
Fall 2024, EECS, MIT

Overview

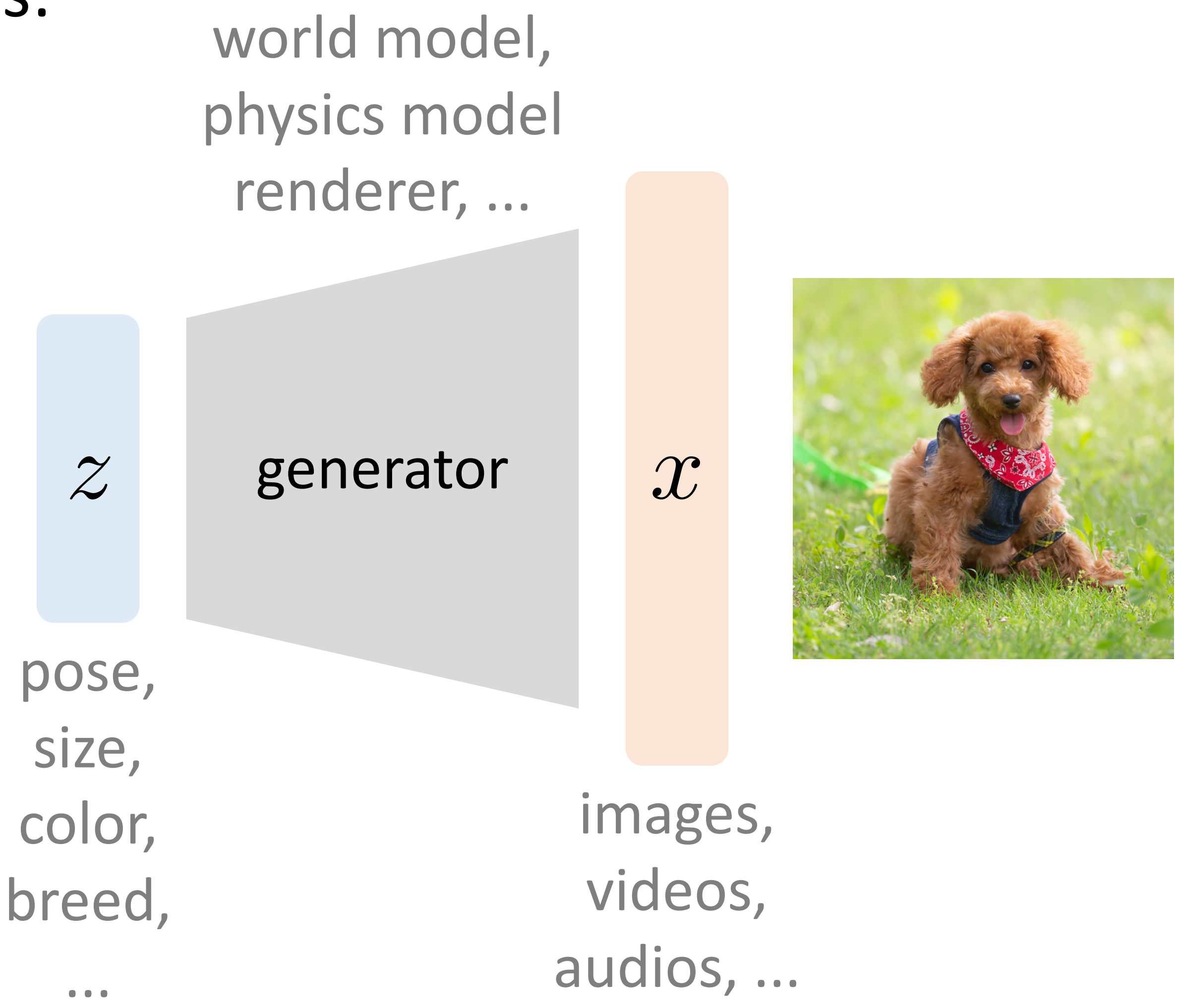
- Variational Autoencoder (VAE)
- Relation to Expectation-Maximization (EM)
- Vector Quantized VAE (VQ-VAE)

Variational Autoencoder (VAE)

Latent Variable Models

Assuming a data generation process:

- z - latent variables
- x - observed variables



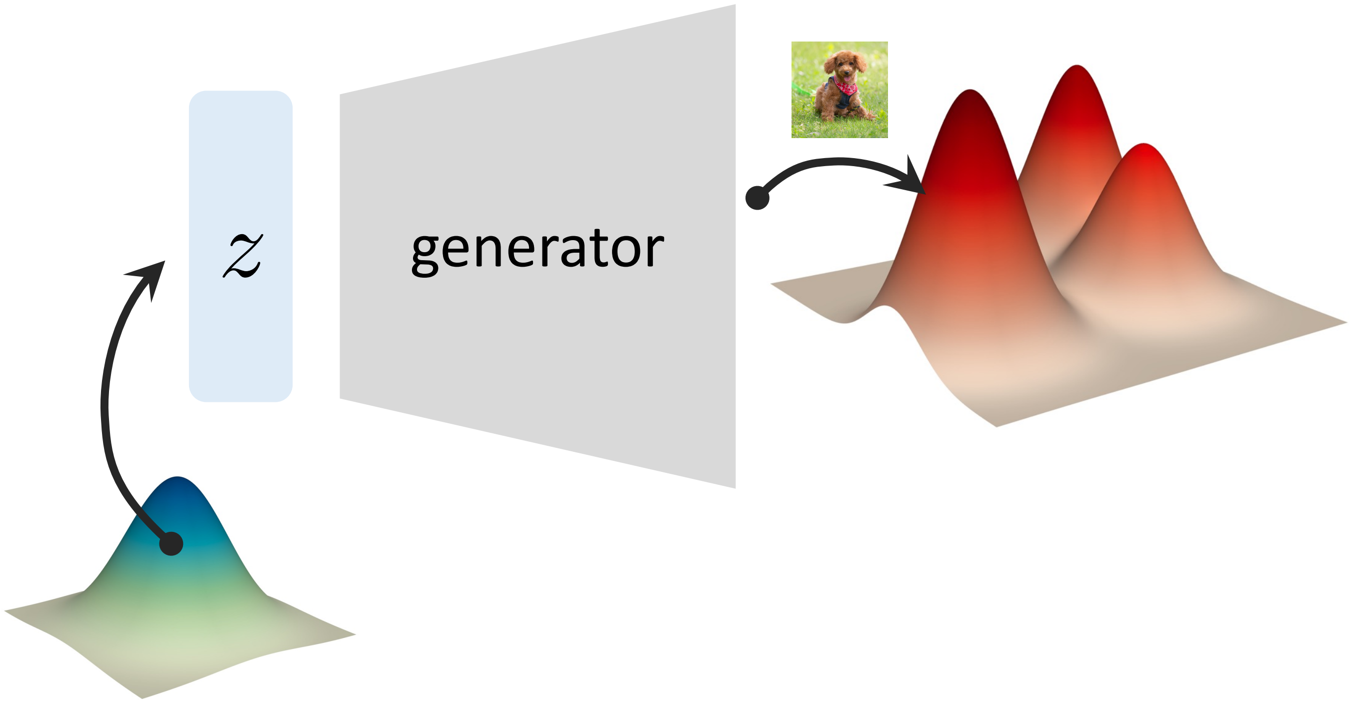
Latent Variable Models

Assuming a data generation process:

- z - latent variables
- x - observed variables

x has a distribution

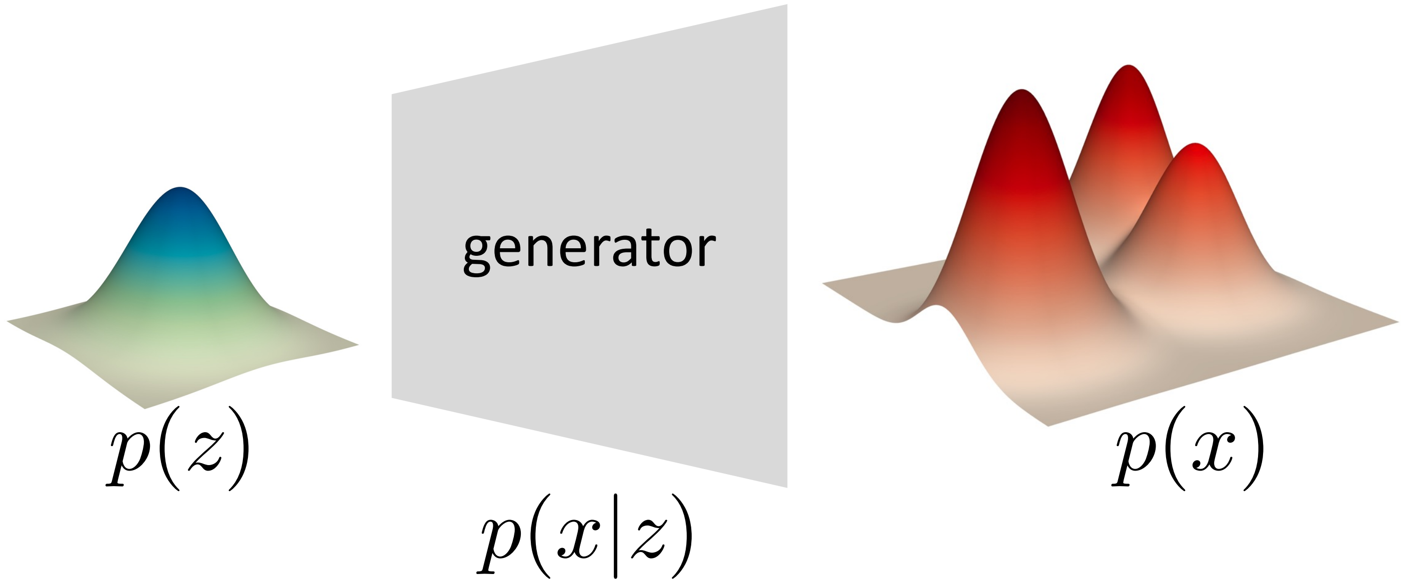
sample from
prior distribution
$z \sim p(z)$



Latent Variable Models

Assuming a data generation process:

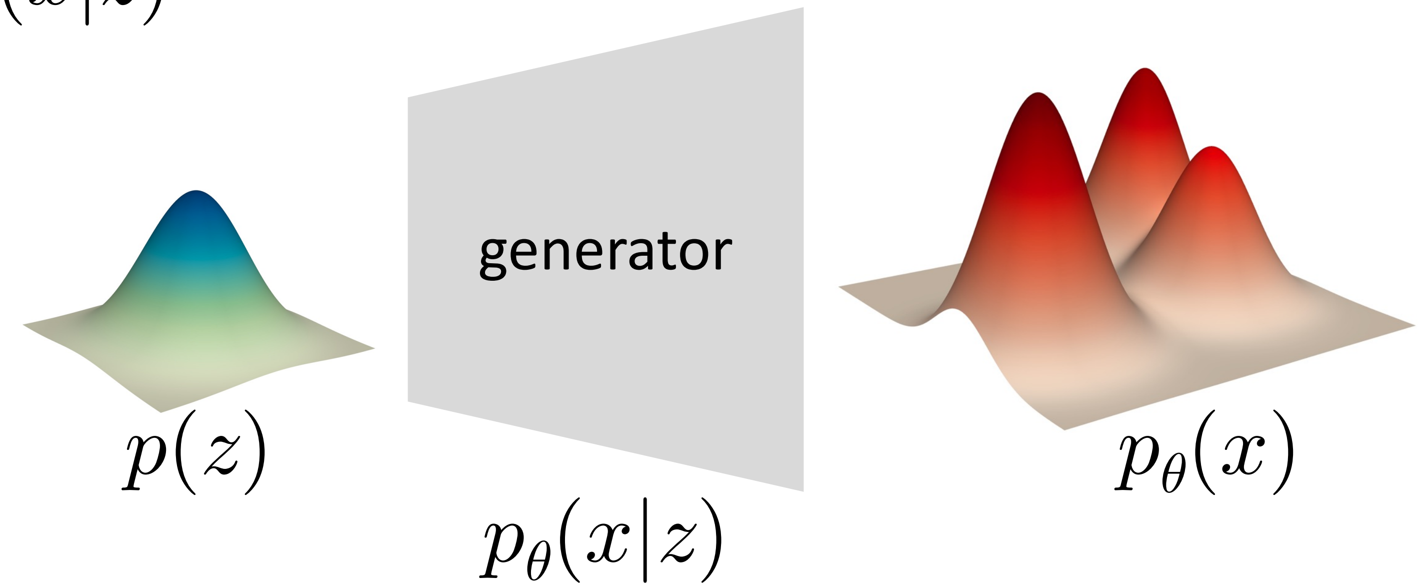
- z - latent variables
- x - observed variables



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:

$$\min_{\theta} \mathcal{D}_{\mathrm{KL}}(\, p_{data} \,||\, p_{\theta} \,)$$

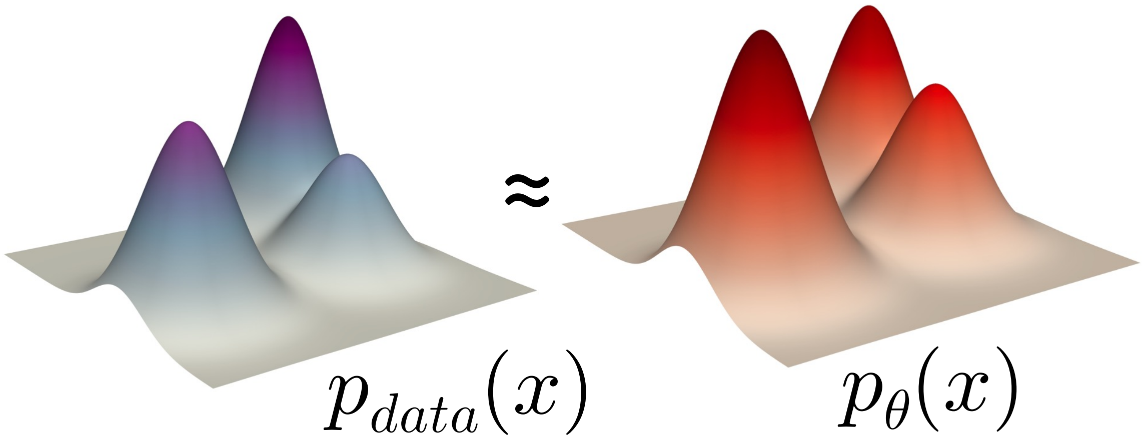
Note: consider other criteria than KL?

⇒ Maximize likelihood:

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

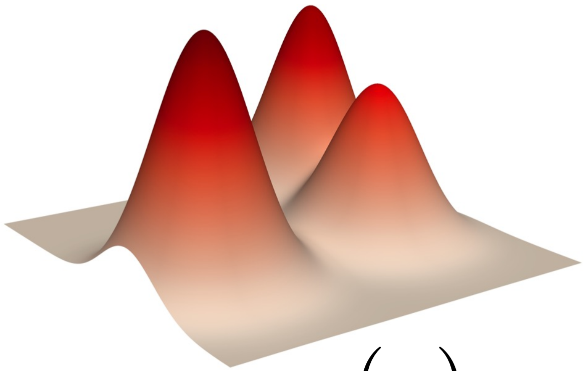
tl; dr

$$\begin{aligned}
& \arg\min_{\theta} \mathcal{D}_{\mathrm{KL}}(\, p_{data} \,||\, p_{\theta} \,) \\
= \; & \arg\min_{\theta} \sum_{x} p_{data}(x) \log \frac{p_{data}(x)}{p_{\theta}(x)} \\
= \; & \arg\min_{\theta} \sum_{x} -p_{data}(x) \log p_{\theta}(x) + const \\
= \; & \arg\max_{\theta} \sum_{x} p_{data}(x) \log p_{\theta}(x) \\
= \; & \arg\max_{\theta} \mathbb{E}_{x \sim p_{data}} \log p_{\theta}(x)
\end{aligned}$$



Latent Variable Models

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



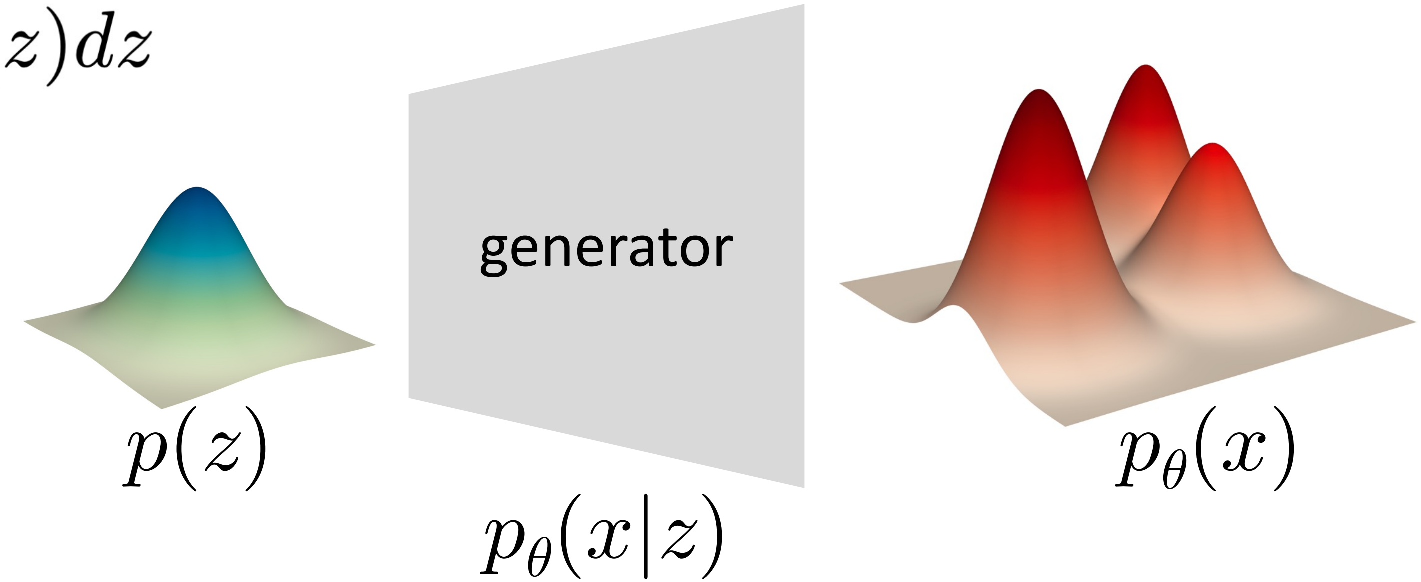
$p_\theta(x)$

Latent Variable Models

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

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

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

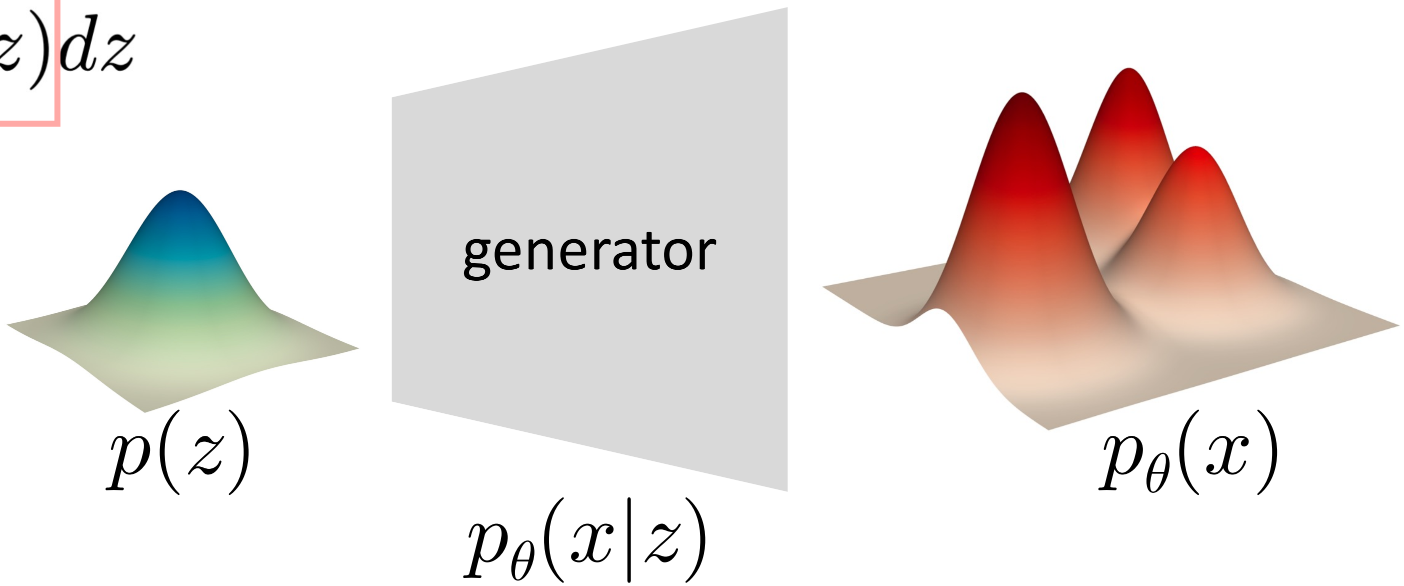


Latent Variable Models

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

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

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



Two sets of unknowns:

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

Idea: introduce a "controllable" distribution $q(z)$

Latent Variable Models

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

Rewrite log likelihood by latent z

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

- for <u>any</u> distribution $q(z)$

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

- Bayes' rule

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

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

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

Latent Variable Models

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

Rewrite log likelihood by latent z

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

Latent Variable Models

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

Rewrite log likelihood by latent z

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

Latent Variable Models

Rewrite log likelihood by latent z

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

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

- for <u>any</u> distribution $q(z)$

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

- Bayes' rule

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

- just algebra

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

- just algebra

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

Latent Variable Models

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

Rewrite log likelihood by latent z

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

- for any distribution $q(z)$

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

- Bayes' rule

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

- just algebra

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

- just algebra

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

Latent Variable Models

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

Rewrite log likelihood by latent z

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

Latent Variable Models

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

Rewrite log likelihood by latent z

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

- for <u>any</u> distribution $q(z)$

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

- Bayes' rule

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

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

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

tractable tractable intractable

Latent Variable Models

intractable $\boxed{\log p_\theta(x)} \boxed{- \mathcal{D}_{\mathrm{KL}}\Big(q(z)||p_\theta(z|x)\Big)}$ intractable

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

Latent Variable Models

$$\text{intractable} \quad \boxed{\log p_\theta(x)} \boxed{- \mathcal{D}_{\text{KL}}\Big(q(z)||p_\theta(z|x)\Big)} \quad \text{intractable}$$

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

- This is called Evidence Lower Bound (ELBO)
- Lower bound of $\log p_\theta(x)$

- This equation holds for <u>any</u> distribution $q(z)$

Latent Variable Models

$$\underbrace{\underbrace{\boxed{\mathbb{E}_{z\sim \overset{q_\phi(z|x)}{\cancel{q(z)}}}\Big[\log p_\theta(x|z)\Big]}}_{\text{tractable}}\ \underbrace{\boxed{- \mathcal{D}_{\mathrm{KL}}\Big(\overset{q_\phi(z|x)}{\cancel{q(z)}}||p_\theta(z)\Big)}}_{\text{tractable}}}$$

- This is called Evidence Lower Bound (ELBO)
- Lower bound of $\log p_\theta(x)$

- This equation holds for <u>any</u> distribution $q(z)$

- Parameterize $q(z)$ by $q_\phi(z|x)$

Latent Variable Models

$$\underbrace{\underset{\text{tractable}}{\boxed{\mathbb{E}_{z\sim \overset{q_\phi(z|x)}{\cancel{q(z)}}}\left[\log p_\theta(x|z)\right]}}\;\underset{\text{tractable}}{\boxed{-\mathcal{D}_{\mathrm{KL}}\left(\overset{q_\phi(z|x)}{\cancel{q(z)}}\,||\,\overset{p(z)}{\cancel{p_\theta(z)}}\right)}}}$$

- This is called Evidence Lower Bound (ELBO)
- Lower bound of $\log p_\theta(x)$

- This equation holds for <u>any</u> distribution $q(z)$

- Parameterize $q(z)$ by $q_\phi(z|x)$
- let $p_\theta(z)$ be a simple known prior $p(z)$

Variational Autoencoder

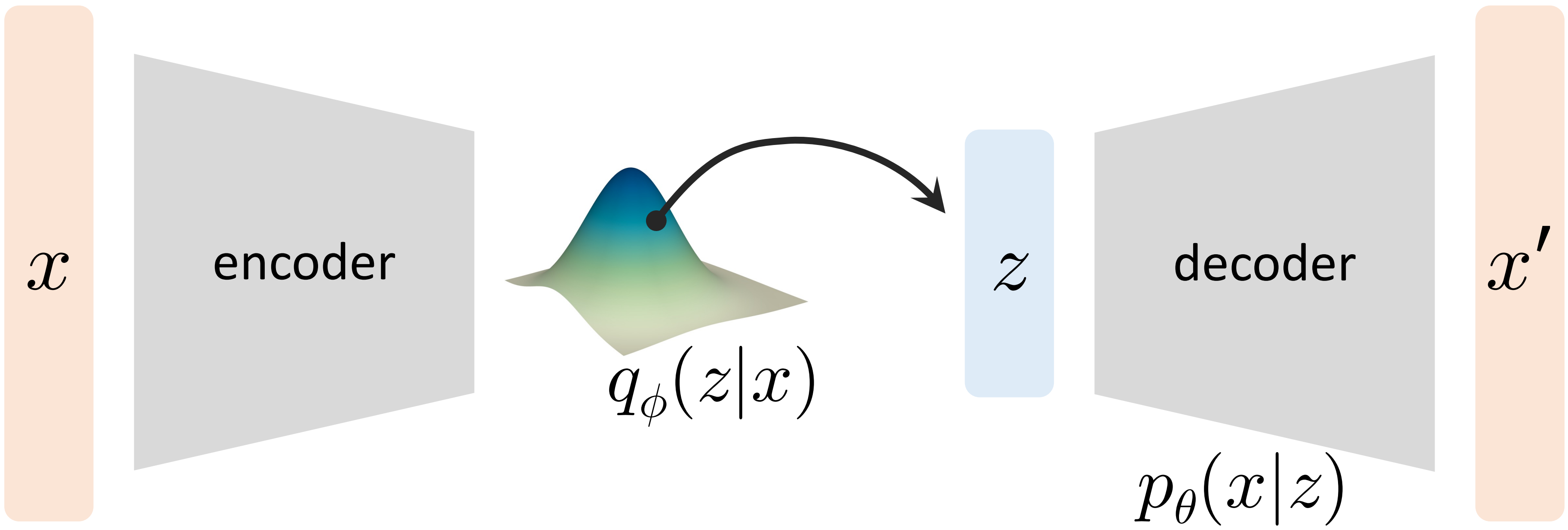
Maximize ELBO ⇒ minimize an objective:

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

Variational Autoencoder

Maximize ELBO ⇒ minimize an objective:

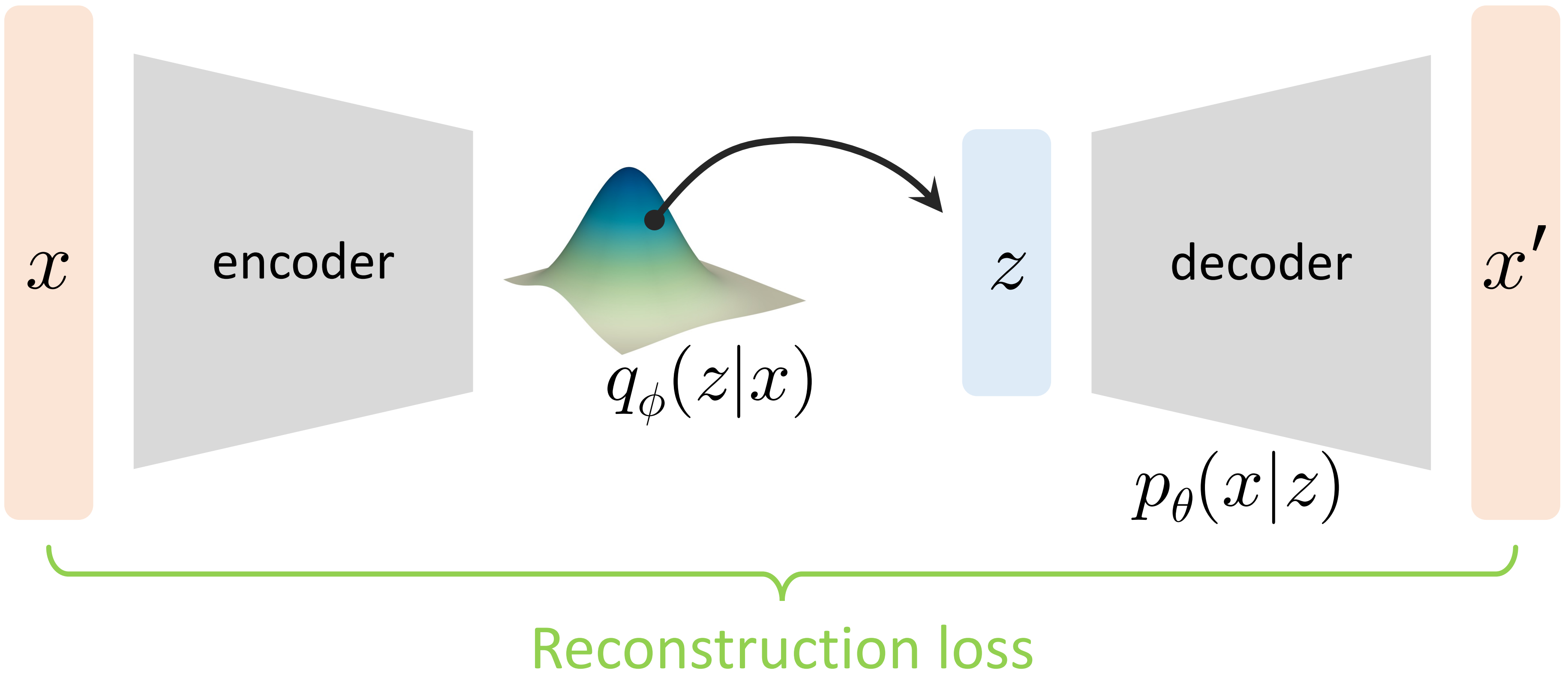
$$\mathcal{L}_{\theta,\phi}(x) = \boxed{-\mathbb{E}_{z\sim q_\phi(z|x)}\Big[\log p_\theta(x|z)\Big]} + \boxed{\mathcal{D}_{\mathrm{KL}}\Big(q_\phi(z|x)||p(z)\Big)}$$



Variational Autoencoder

Maximize ELBO ⇒ minimize an objective:

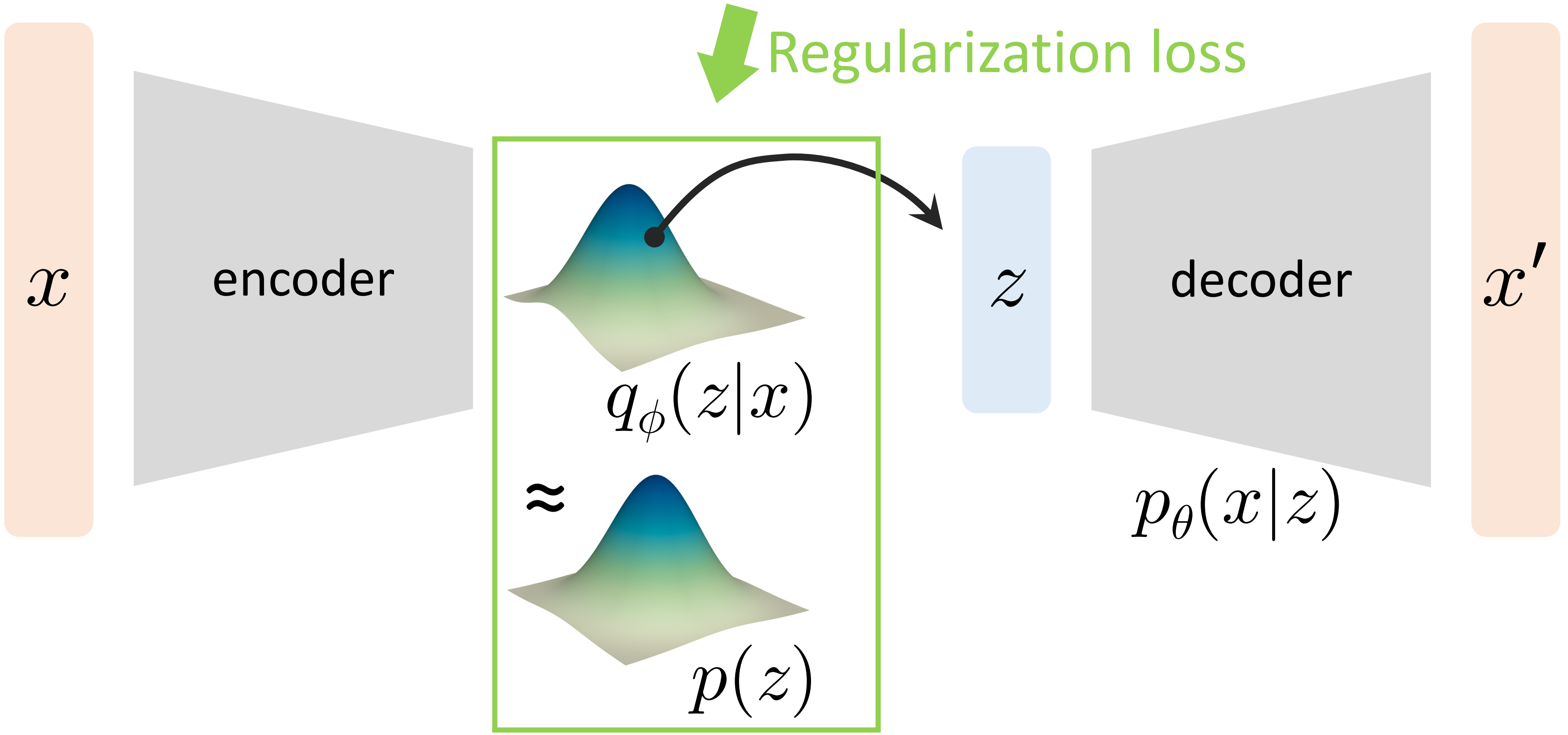
$$\mathcal{L}_{\theta,\phi}(x) = \boxed{-\mathbb{E}_{z\sim q_\phi(z|x)}\Big[\log p_\theta(x|z)\Big]} + \mathcal{D}_{\mathrm{KL}}\Big(q_\phi(z|x)||p(z)\Big)$$



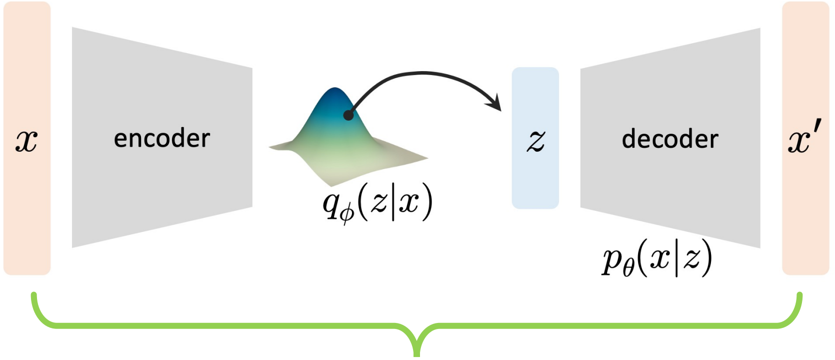
Variational Autoencoder

Maximize ELBO ⇒ minimize an objective:

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



Variational Autoencoder



Reconstruction loss

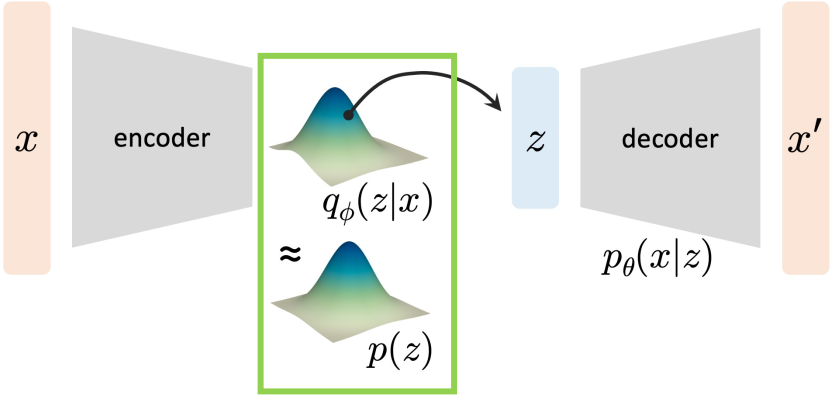
$$\mathcal{L}_{\theta,\phi}(x) = \boxed{-\mathbb{E}_{z\sim q_\phi(z|x)}\Big[\log p_\theta(x|z)\Big]} + \mathcal{D}_{\text{KL}}\Big(q_\phi(z|x)||p(z)\Big)$$

Example: L2 loss

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

- L2 loss ⇒ a Gaussian neighborhood around data point x

Variational Autoencoder



Regularization loss

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

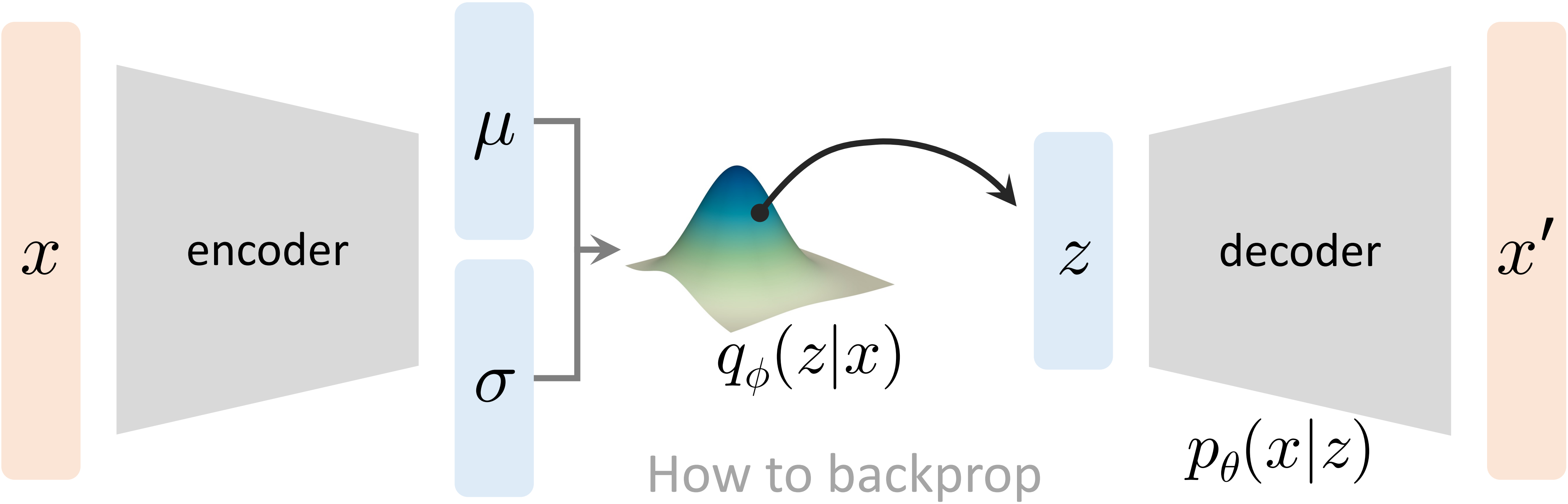
Example: Gaussian prior

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

Variational Autoencoder

Maximize ELBO ⇒ minimize an objective:

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

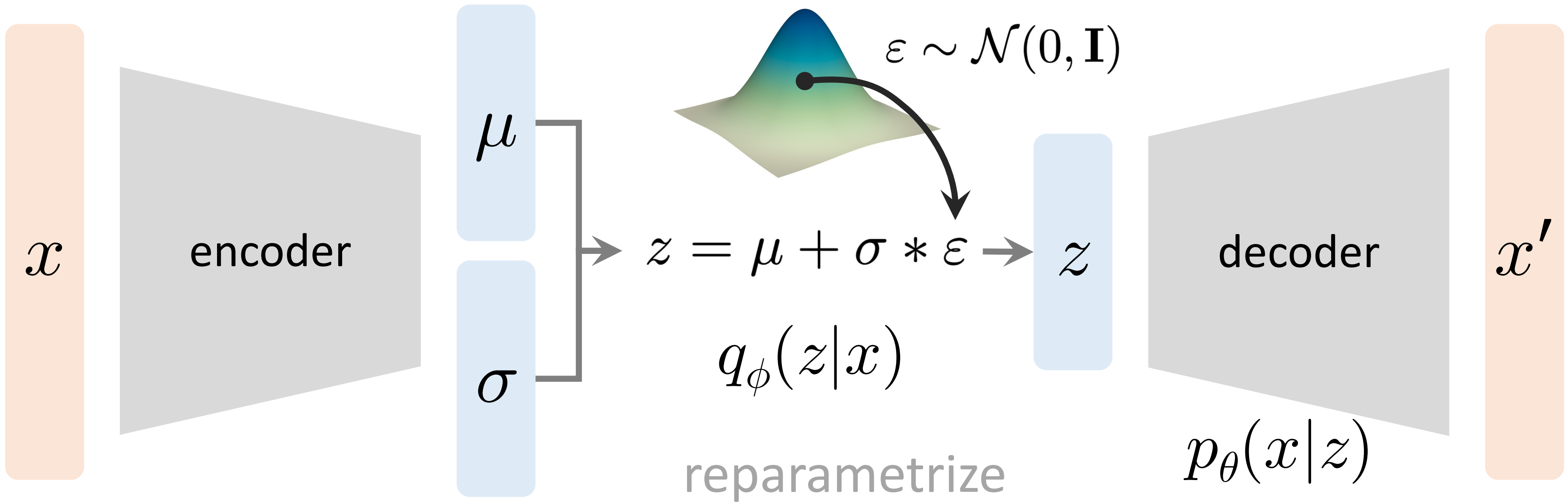


How to backprop w.r.t. sampling?

Variational Autoencoder

Maximize ELBO ⇒ minimize an objective:

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



Variational Autoencoder

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

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

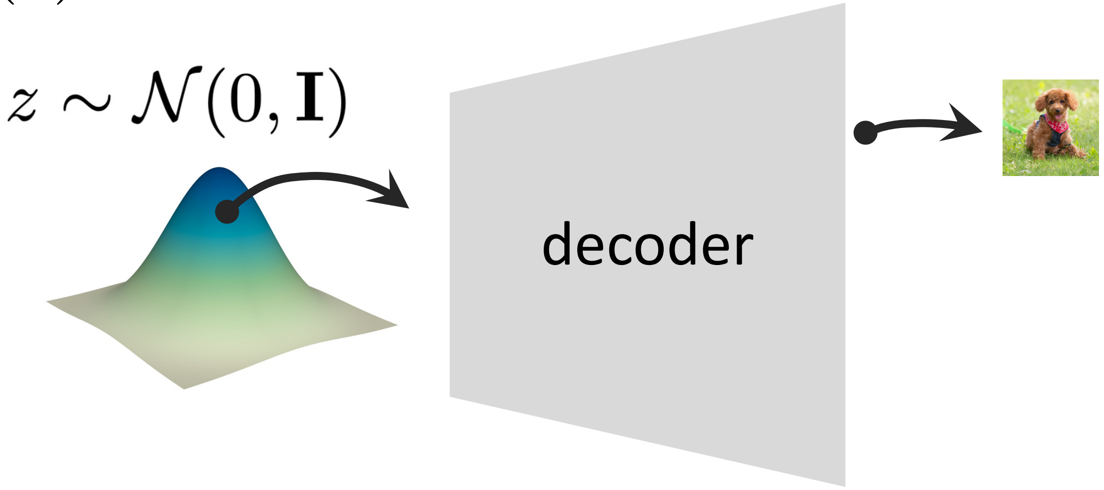
Overall loss is expectation over data:

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

Variational Autoencoder

Inference (generation):

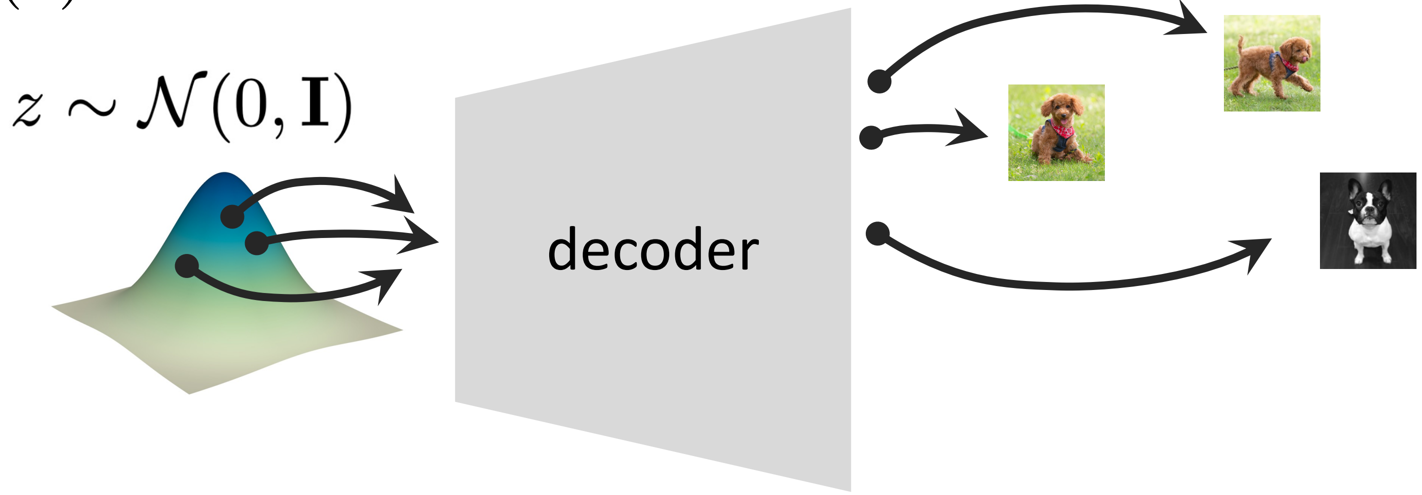
- sample z from: $\mathcal{N}(0, \mathbf{I})$
- map z by decoder net: $g_\theta(z)$



Variational Autoencoder

Inference (generation):

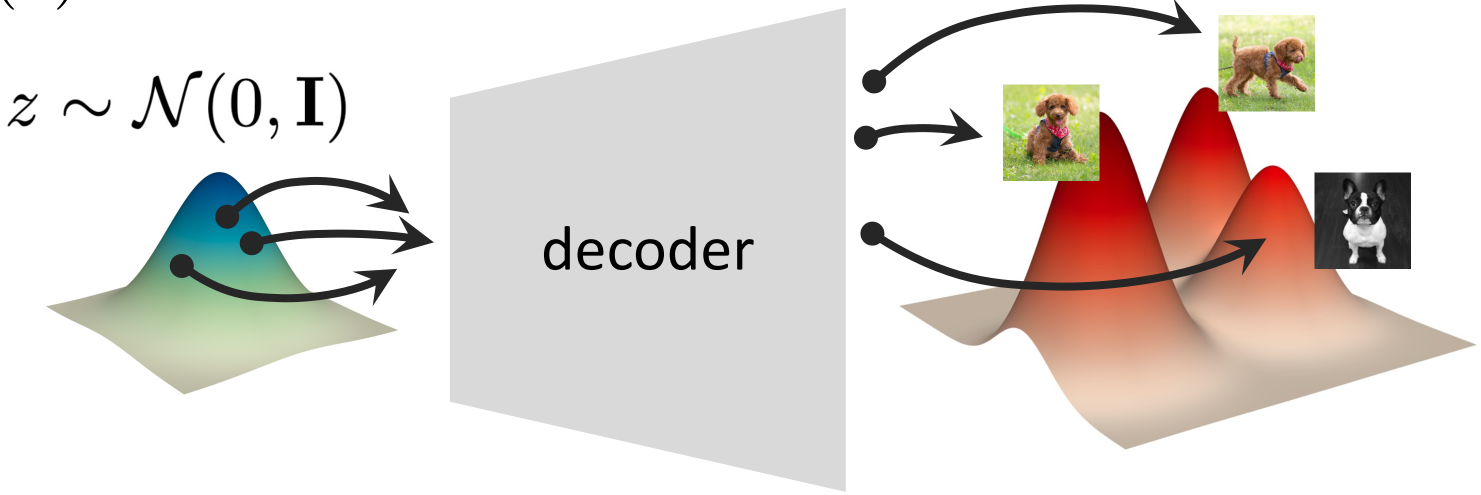
- sample z from: $\mathcal{N}(0, \mathbf{I})$
- map z by decoder net: $g_\theta(z)$



Variational Autoencoder

Inference (generation):

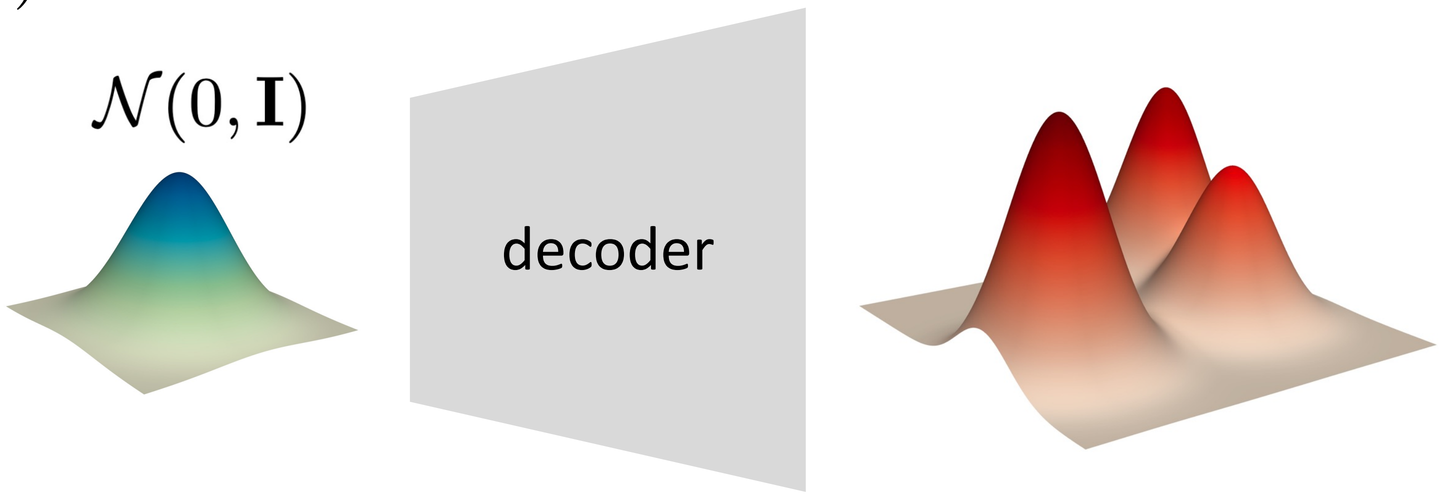
- sample z from: $\mathcal{N}(0, \mathbf{I})$
- map z by decoder net: $g_\theta(z)$



Variational Autoencoder

Inference (generation):

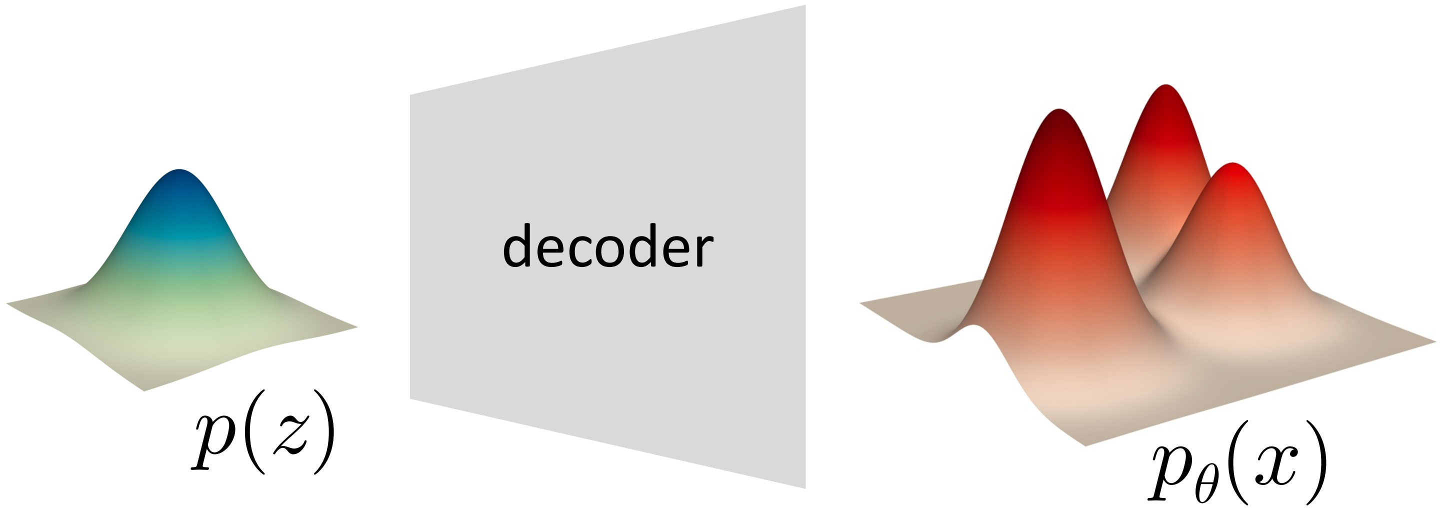
- sample z from: $\mathcal{N}(0, \mathbf{I})$
- map z by decoder net: $g_\theta(z)$



Decoder is a deterministic mapping from one distribution to another.

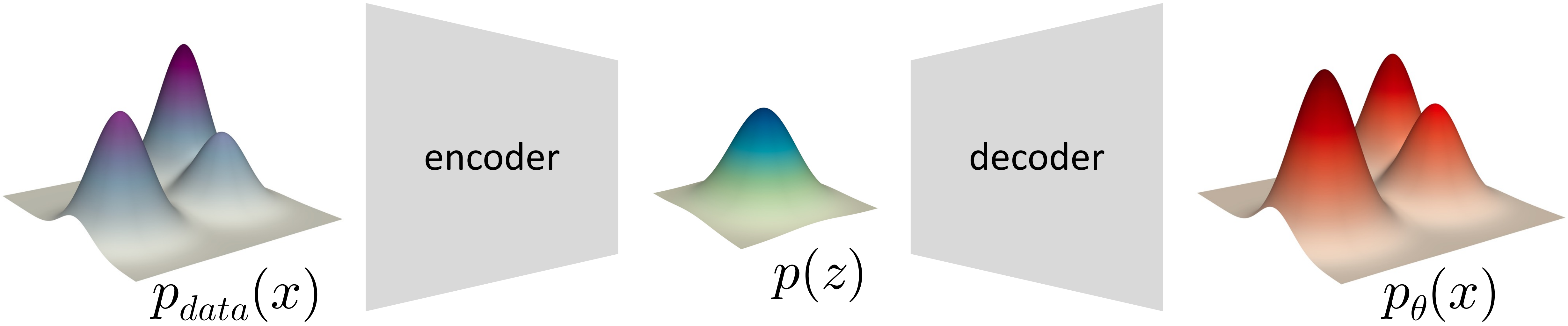
A view of “Autoencoding Distributions”

- decoder: maps latent distribution to data distribution



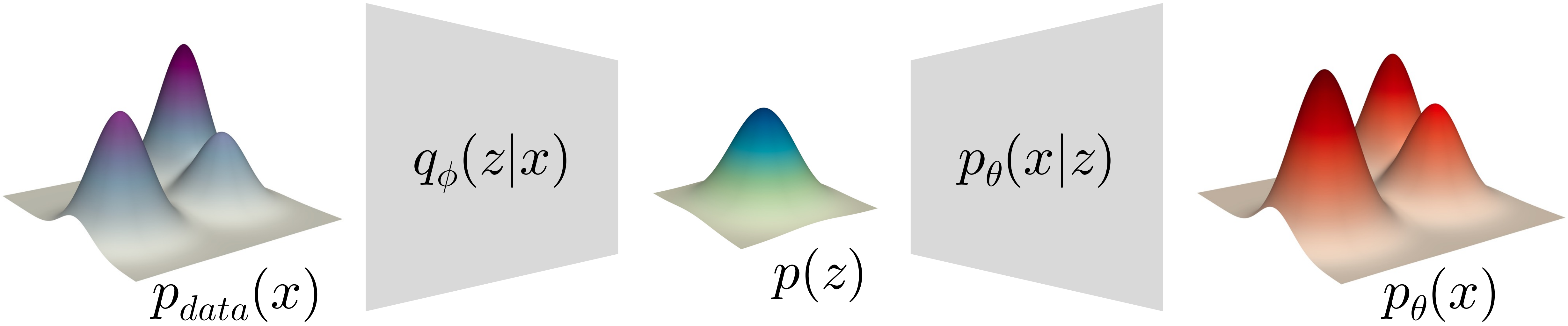
A view of "Autoencoding Distributions"

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



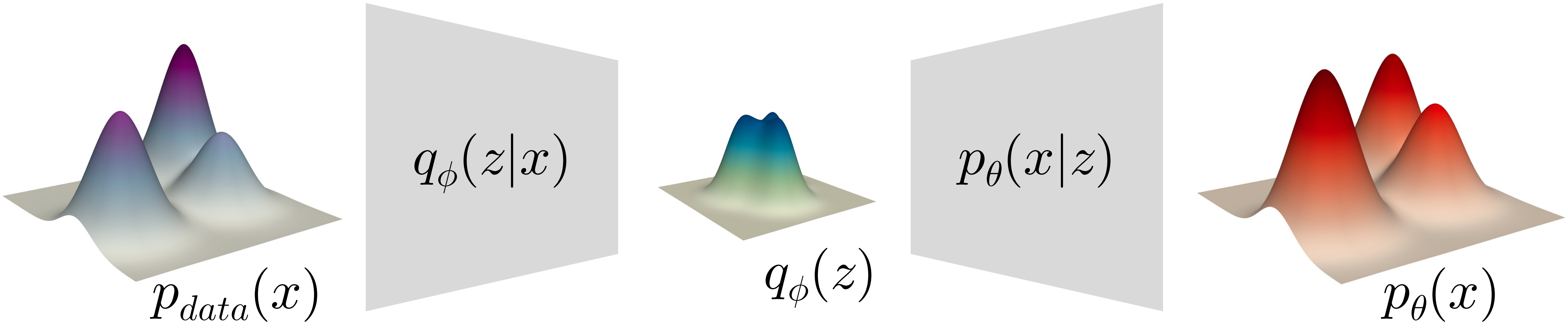
A view of "Autoencoding Distributions"

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



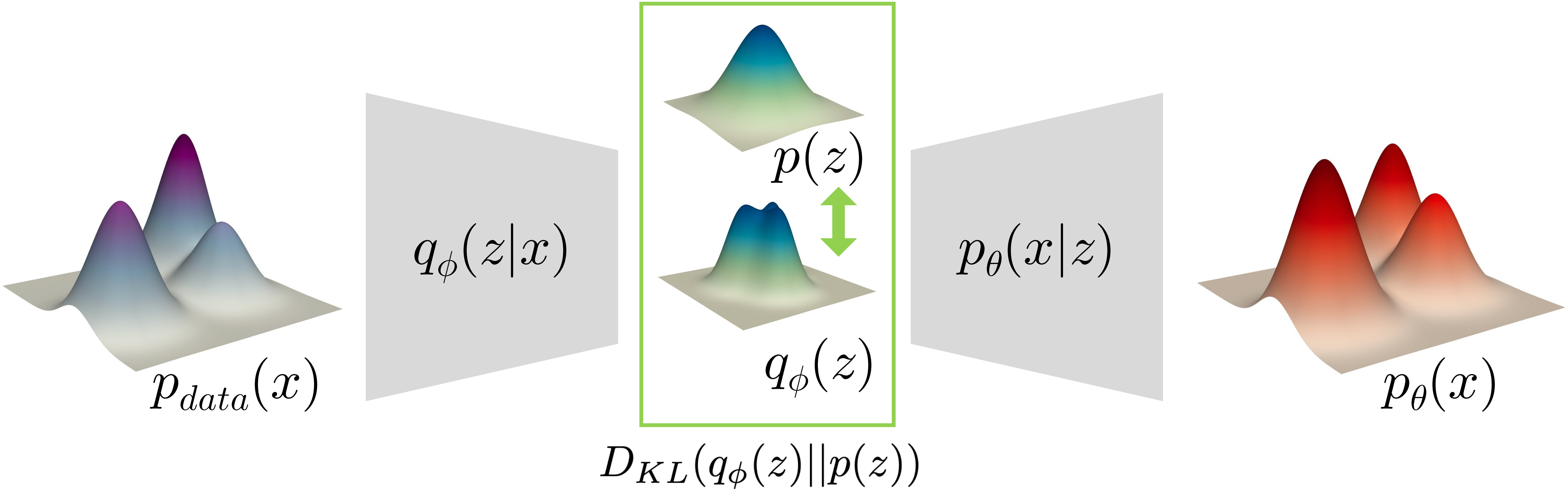
A view of "Autoencoding Distributions"

- 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

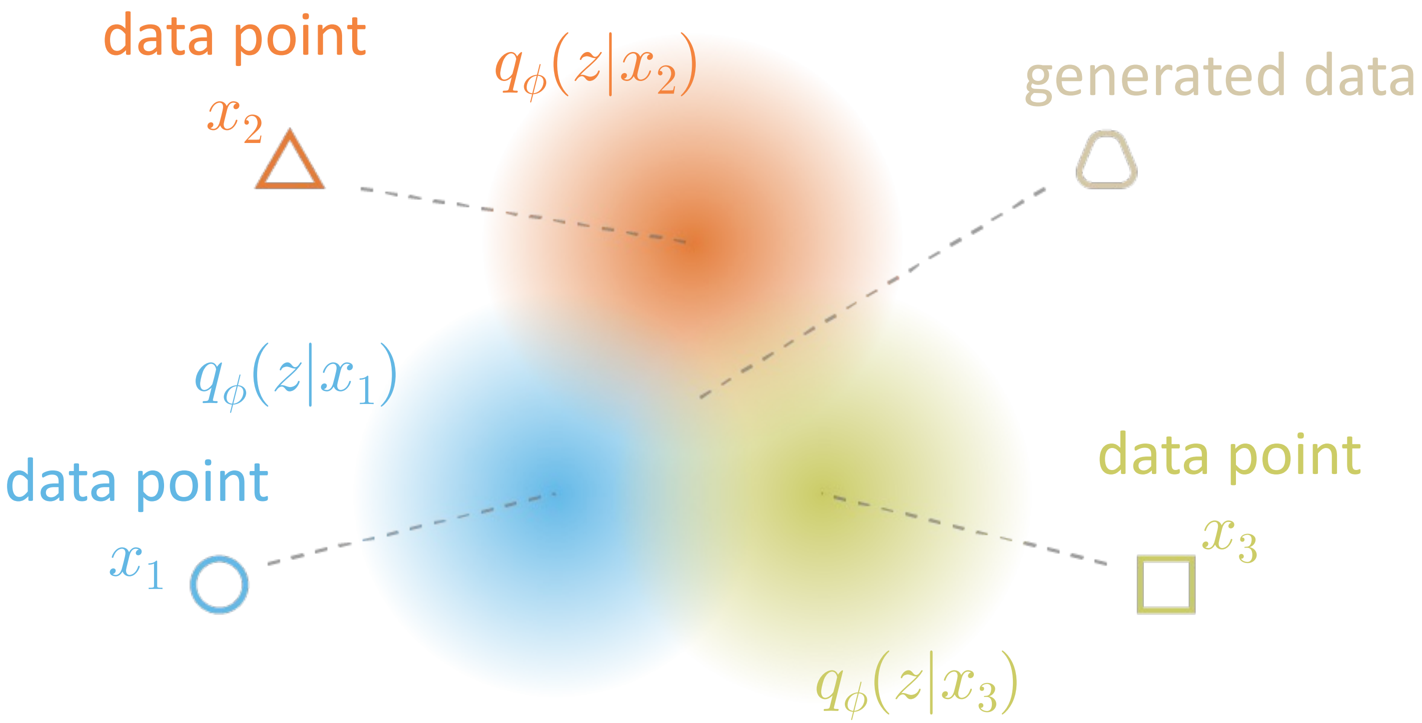
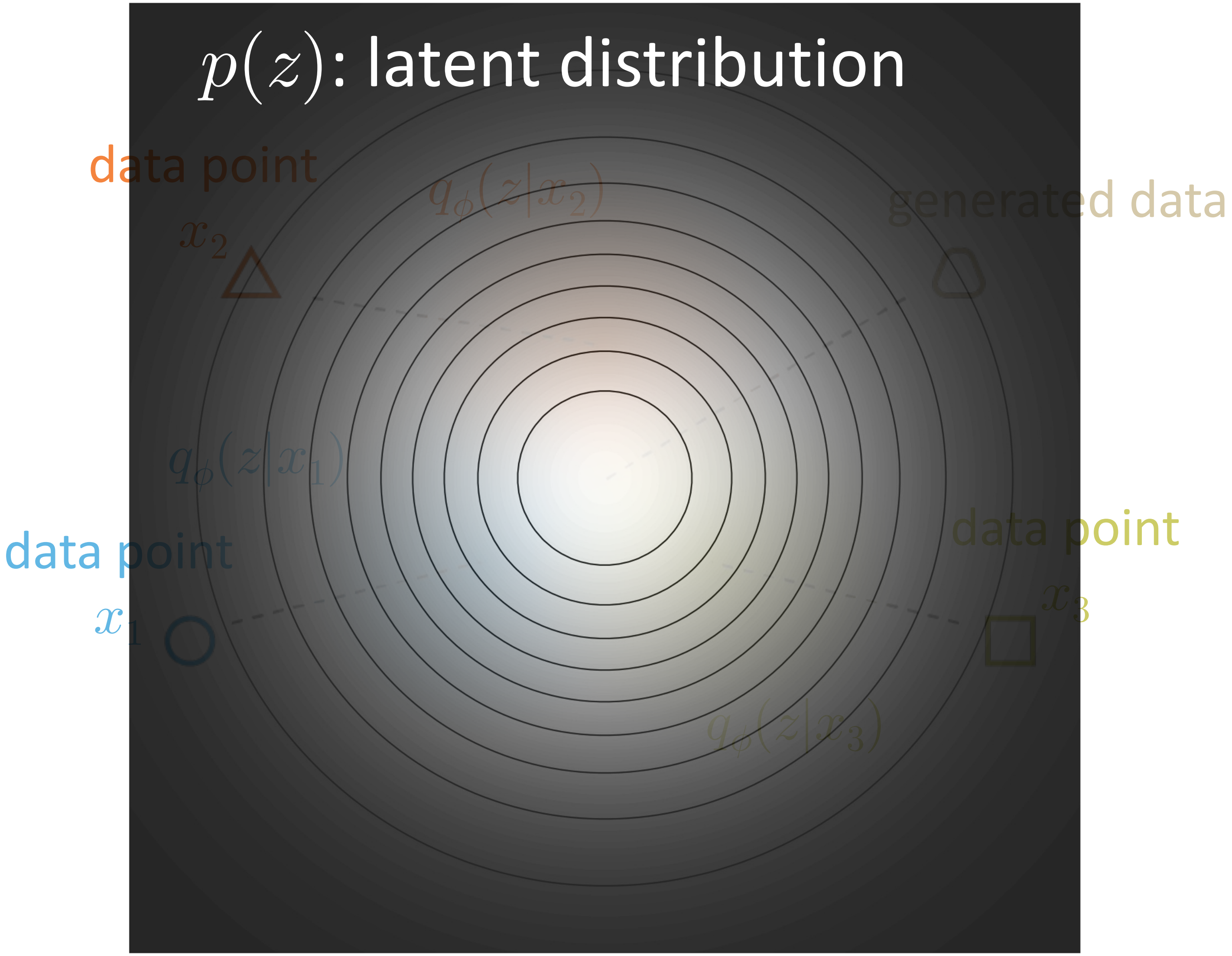


Figure adapted from: Joseph Rocca "Understanding Variational Autoencoders (VAEs)"
https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Illustration



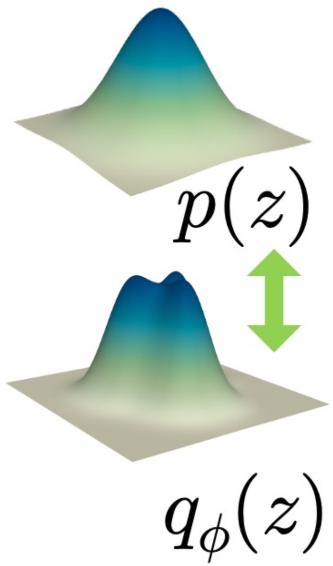
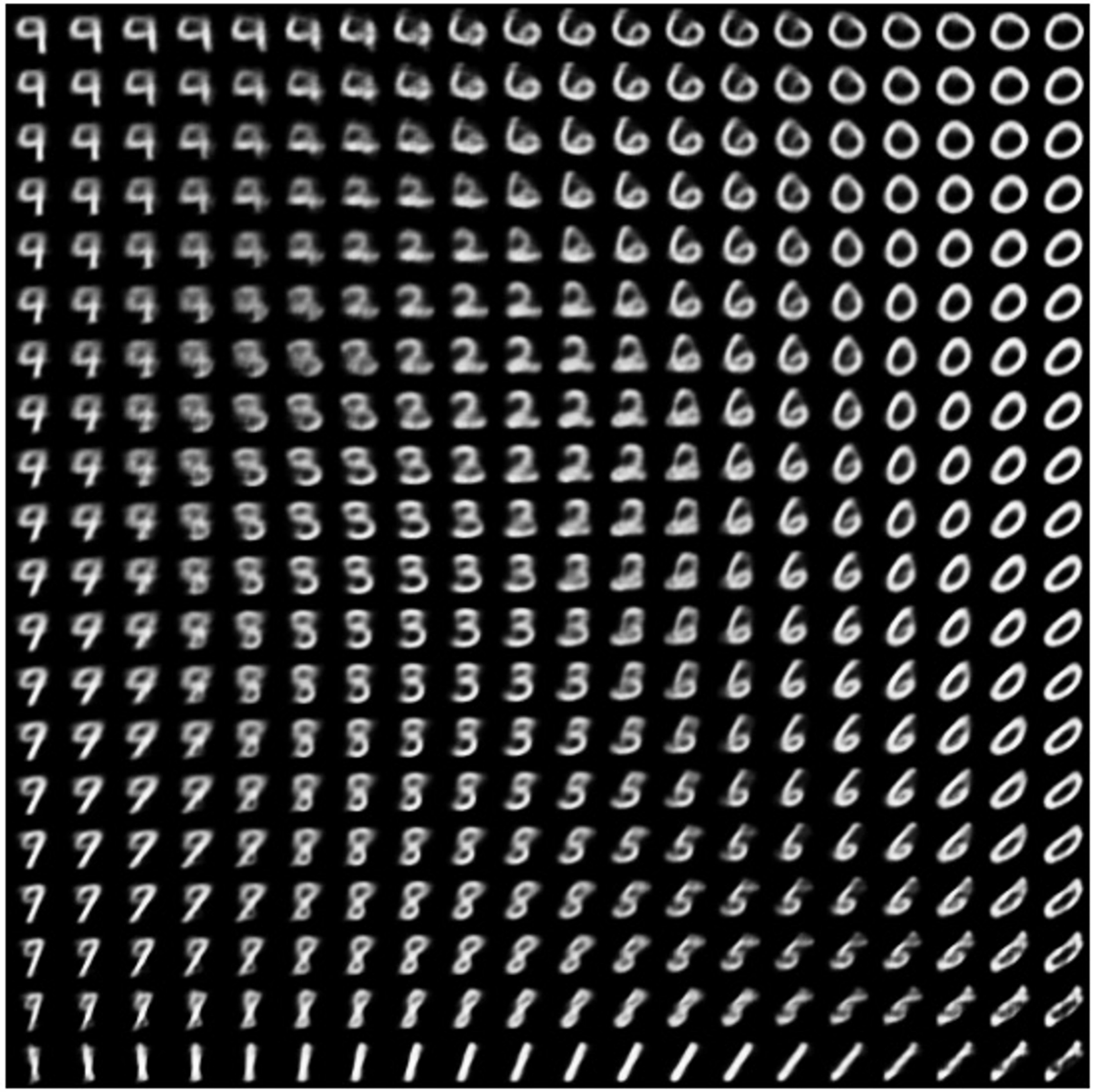


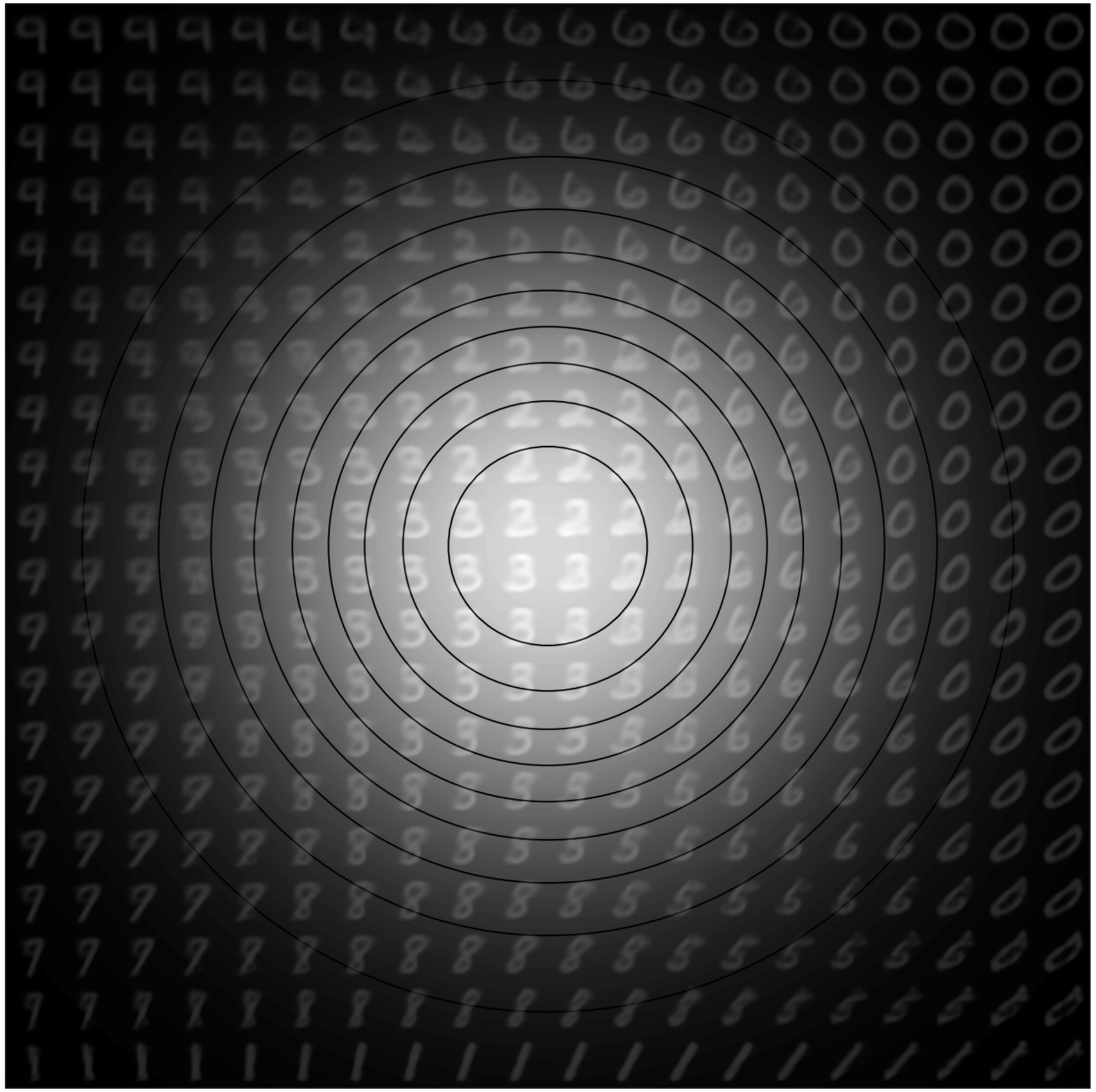
Figure adapted from: Joseph Rocca "Understanding Variational Autoencoders (VAEs)"
https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

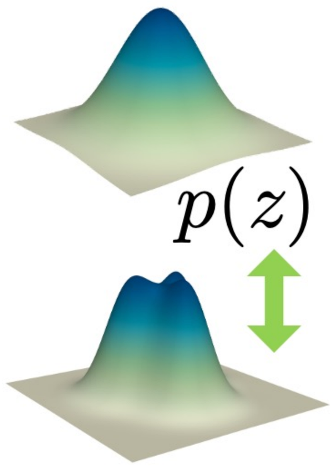
VAE: 2D latent space on MNIST



“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



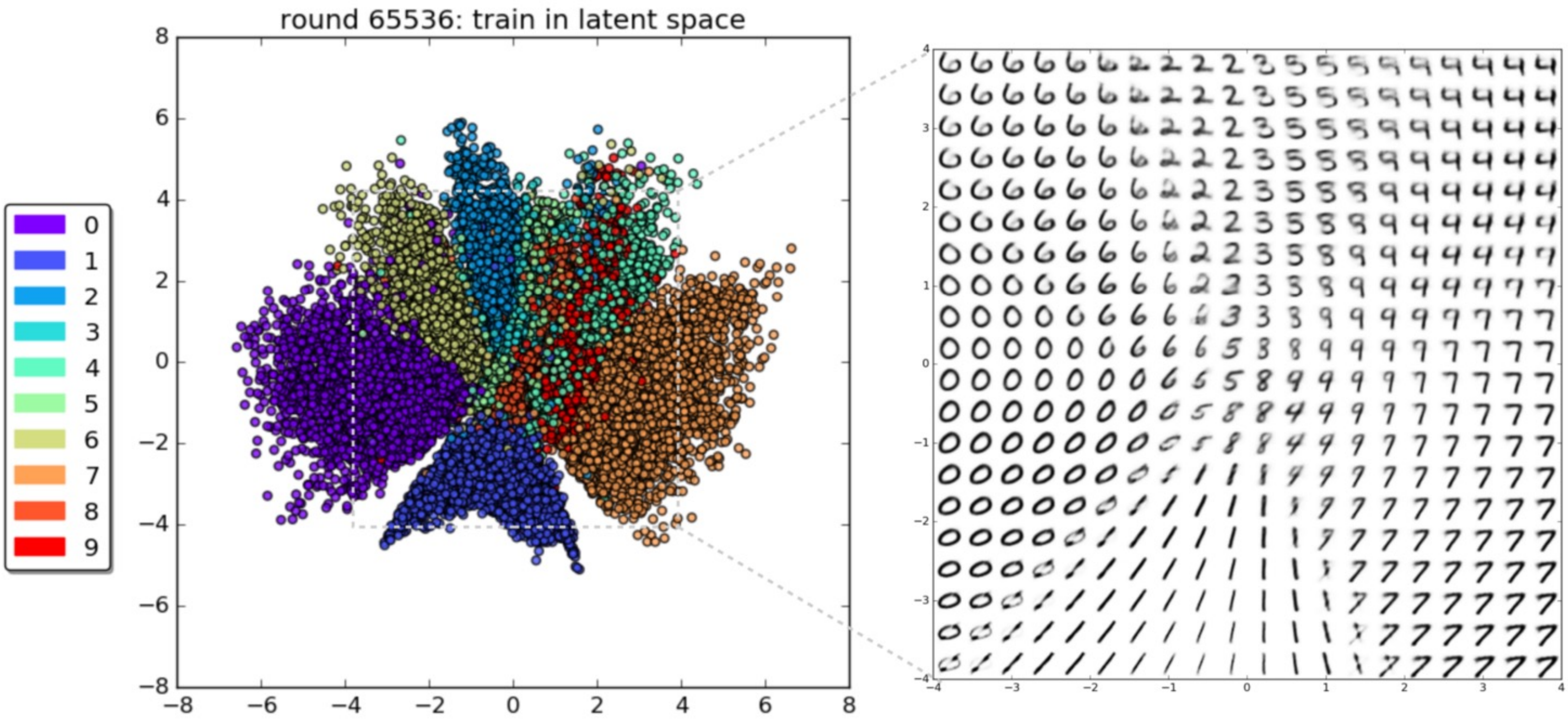


$p(z)$

$q_\phi(z)$

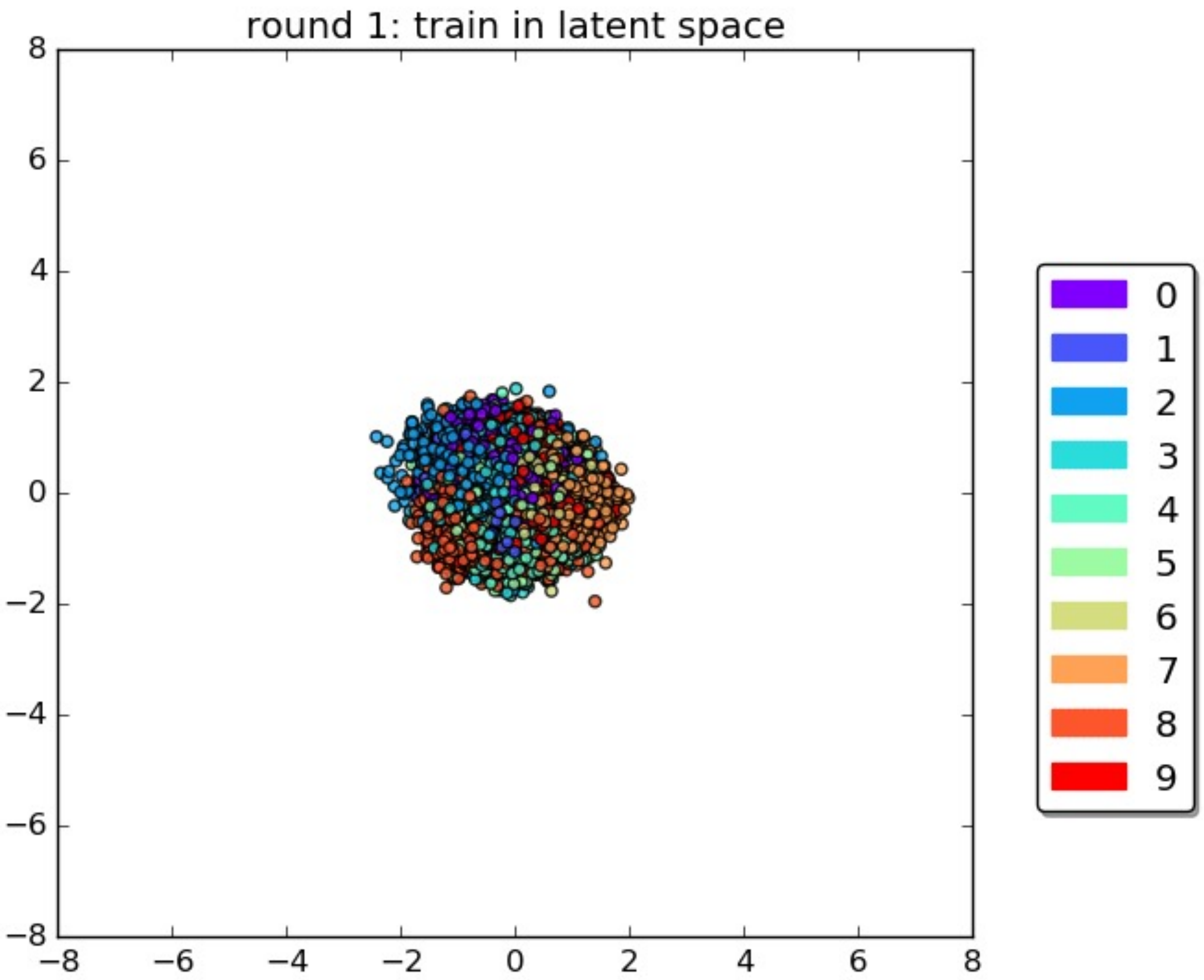
"Convolutional Variational Autoencoder"
https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/generative/cvae.ipynb

VAE: 2D latent space on MNIST



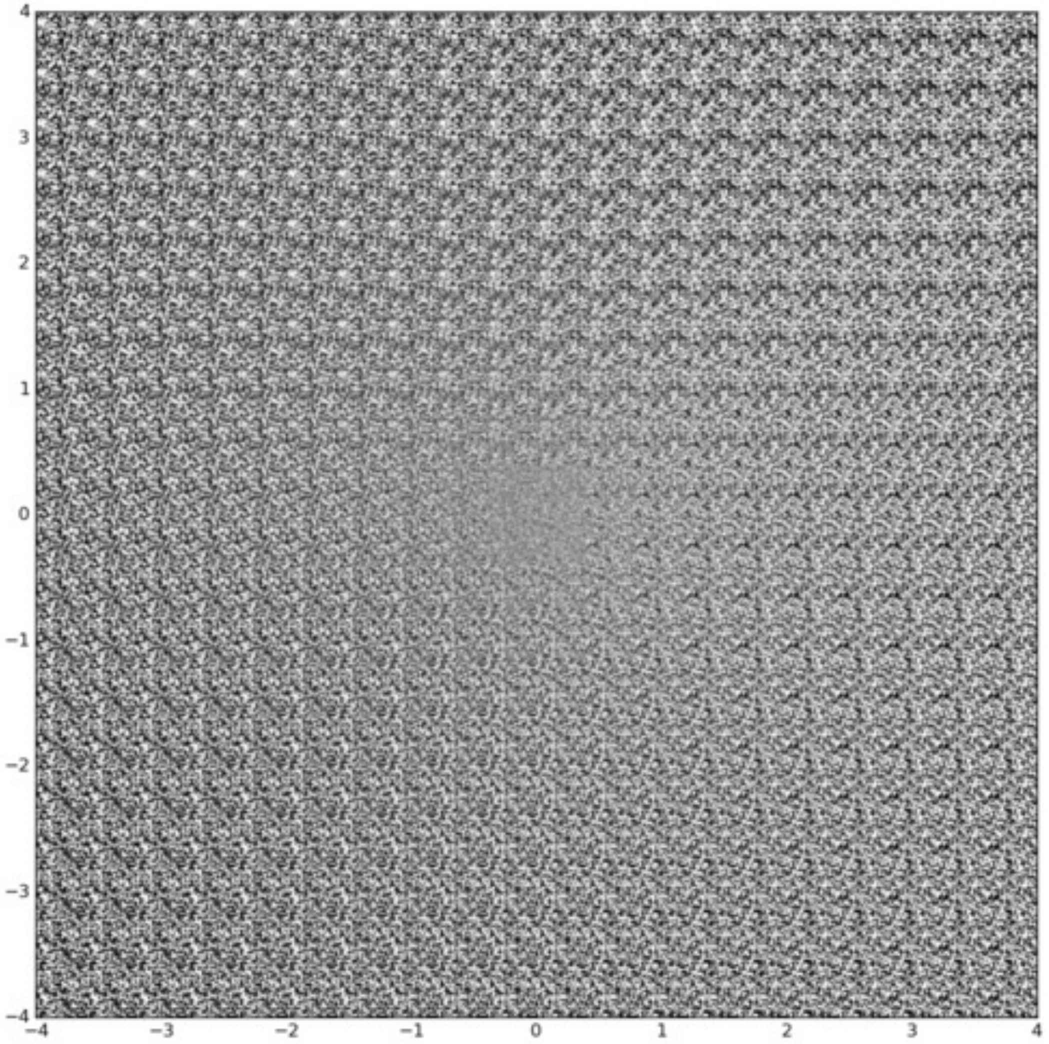
“Introducing Variational Autoencoders (in Prose and Code)”
https://blog.fastforwardlabs.com/2016/08/12/introducing-variational-autoencoders-in-prose-and-code.html

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

Kingma and Welling. Auto-Encoding Variational Bayes, ICLR 2014

Relation to
Expectation-Maximization (EM)

Recap: Latent Variable Models

Two sets of variables:

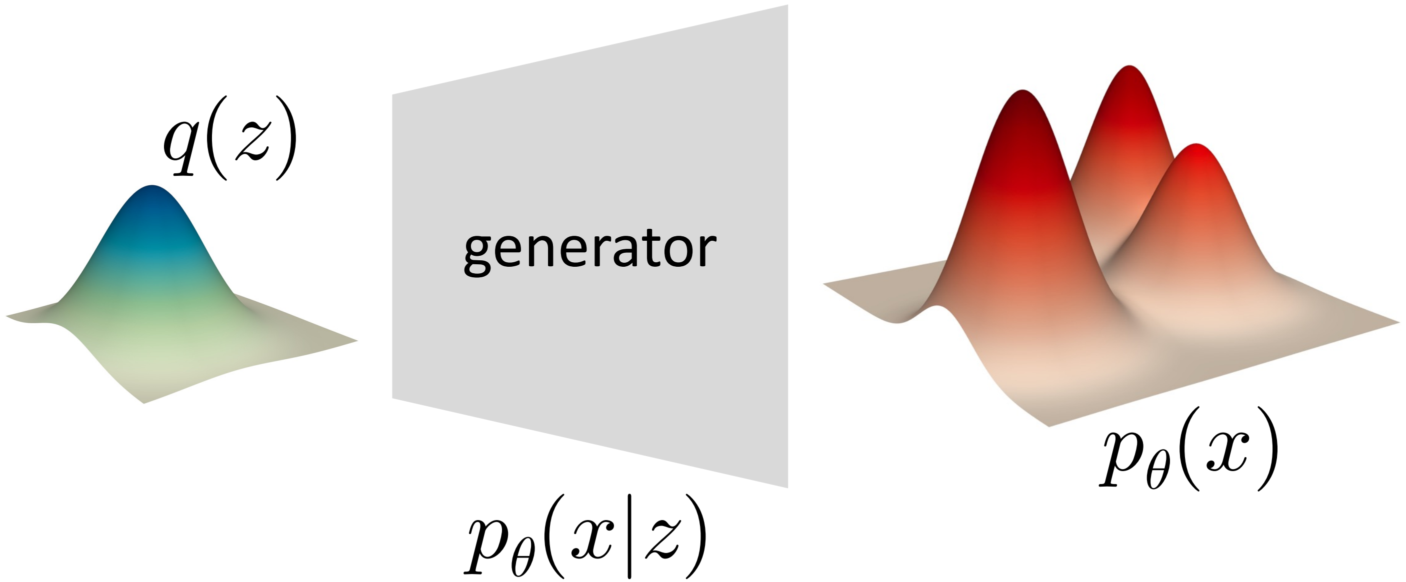
- q: distribution of latent
- θ: parameters of generator

VAE:

- parametrize q by a network
- stochastic gradient decent

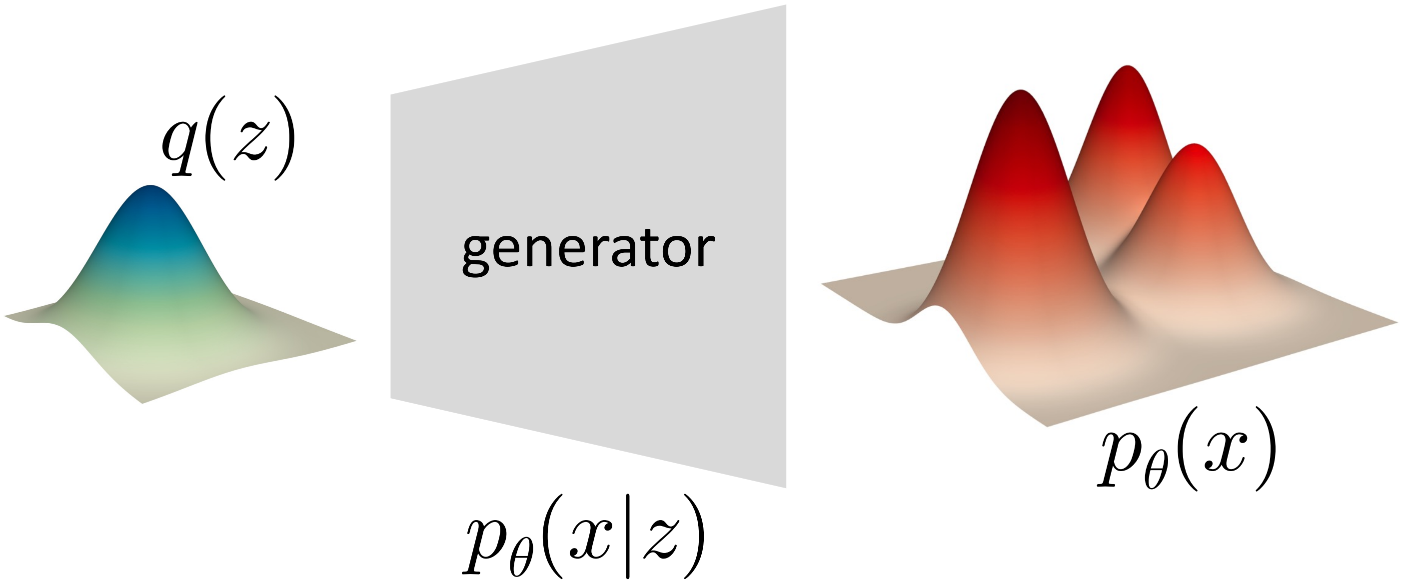
Expectation-Maximization (EM):

- often parametrize q <u>analytically</u>
- <u>coordinate descent</u> (i.e., alternating optimization)



EM as A Max-Max Procedure

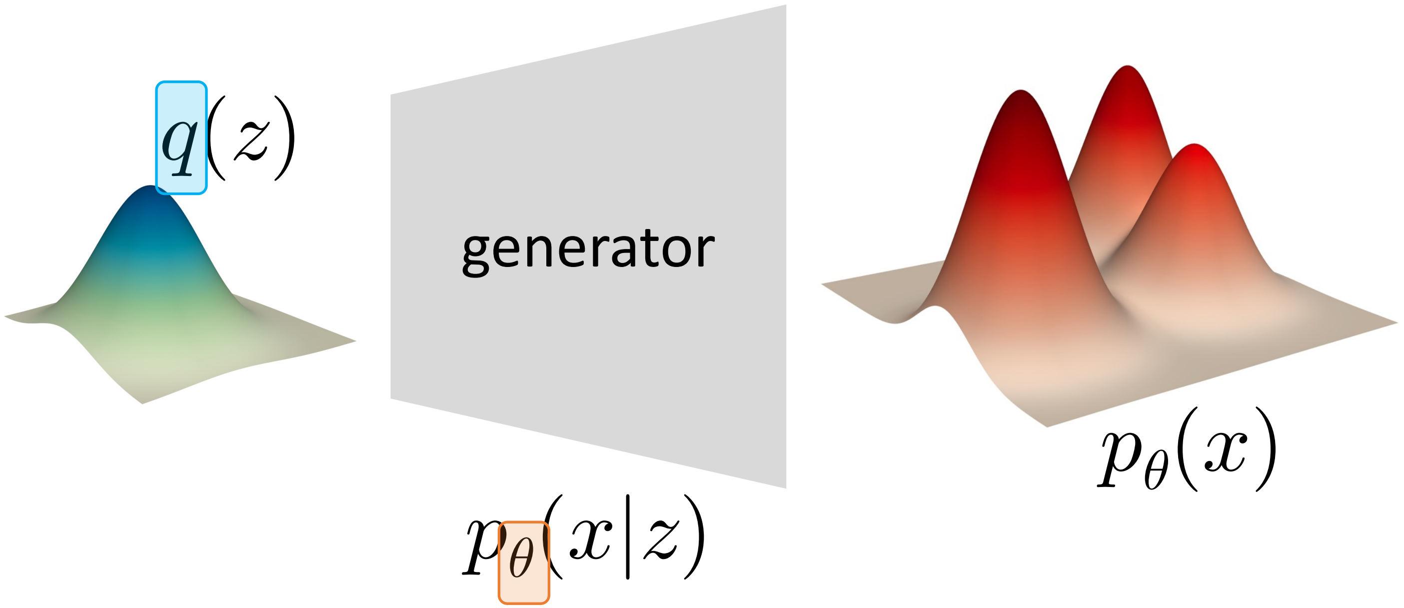
$$\mathrm{ELBO} = \mathbb{E}_{x \sim p_{data}(x)}\left[\mathbb{E}_{z \sim q(z|x)}\left[\log p_\theta(x|z)\right] - \mathcal{D}_{\mathrm{KL}}\Big(q(z|x)||p(z)\Big)\right]$$



EM as A Max-Max Procedure

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

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



Two sets of variables:

- $\boxed{q}$ - distribution of latent
- $\boxed{\theta}$ - parameters of generator

Coordinate descent:

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

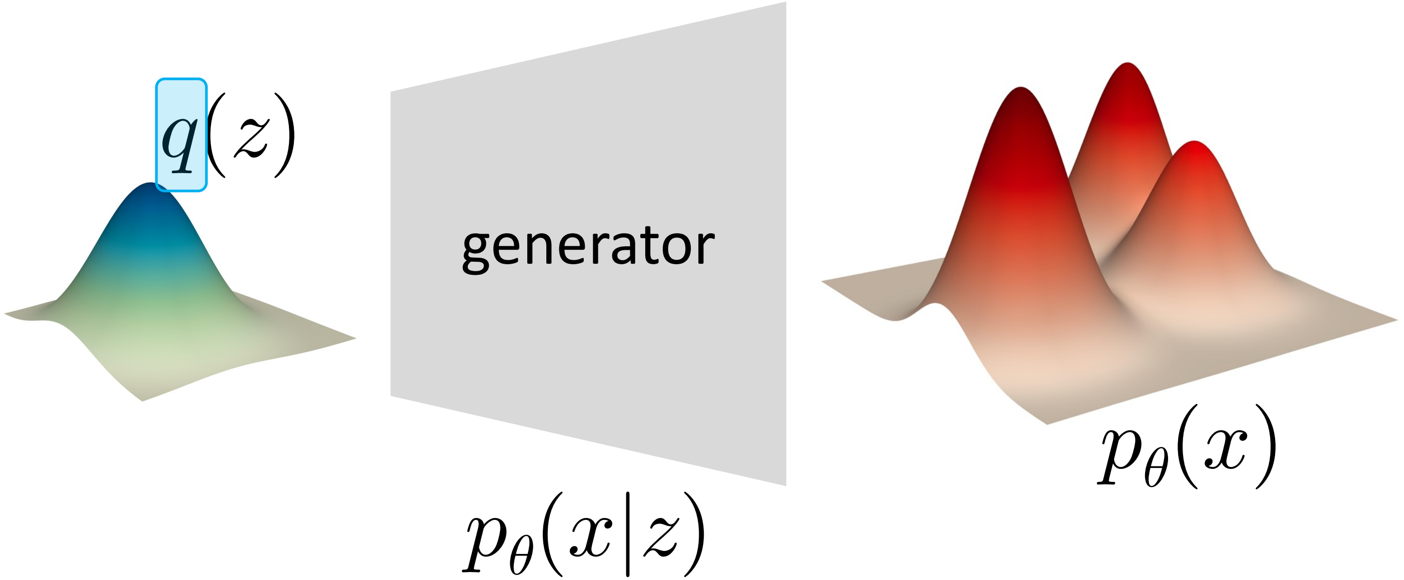
EM as A Max-Max Procedure

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

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

E-step: optimize for q

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



Details can be found in: Hastie et al. “The Elements of Statistical Learning”.

EM as A Max-Max Procedure

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

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

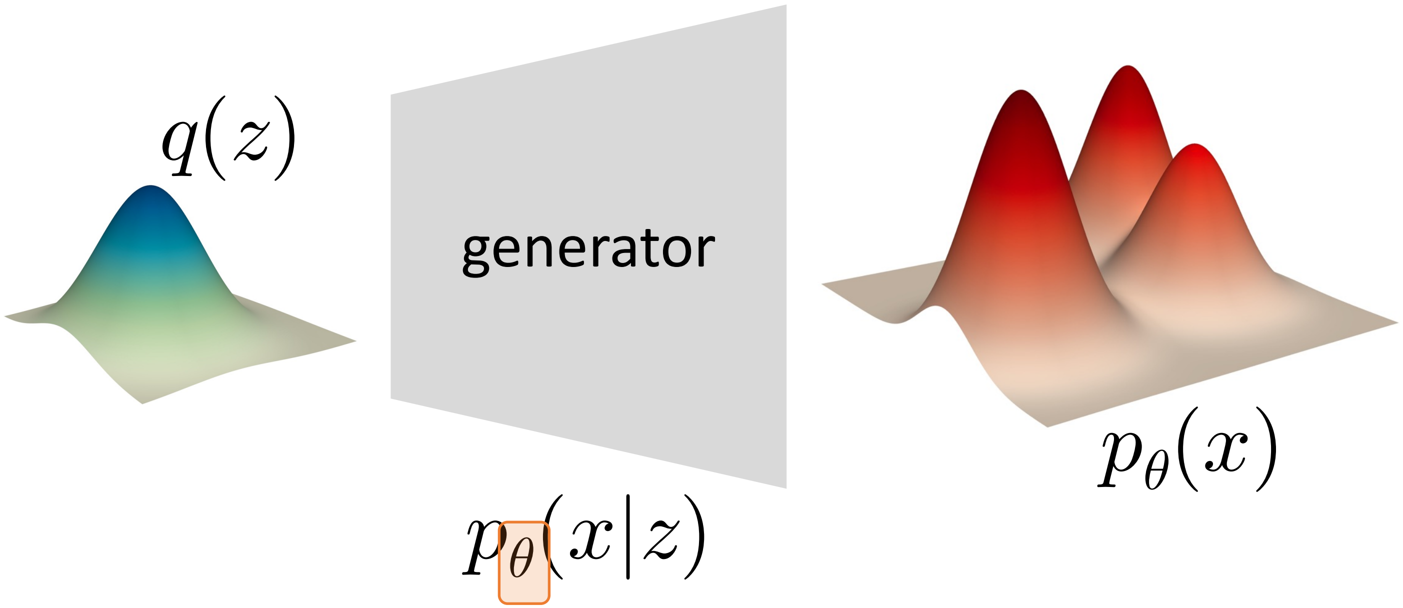
E-step: optimize for q

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

M-step: optimize for θ

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

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



Details can be found in: Hastie et al. “The Elements of Statistical Learning”.

EM as A Max-Max Procedure

$\mathrm{ELBO} = \mathbb{E}_{x \sim p_{data}(x)} \Big[\mathbb{E}_{z \sim q(z|x)} \Big[\log p_\theta$

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

q: often in analytical forms

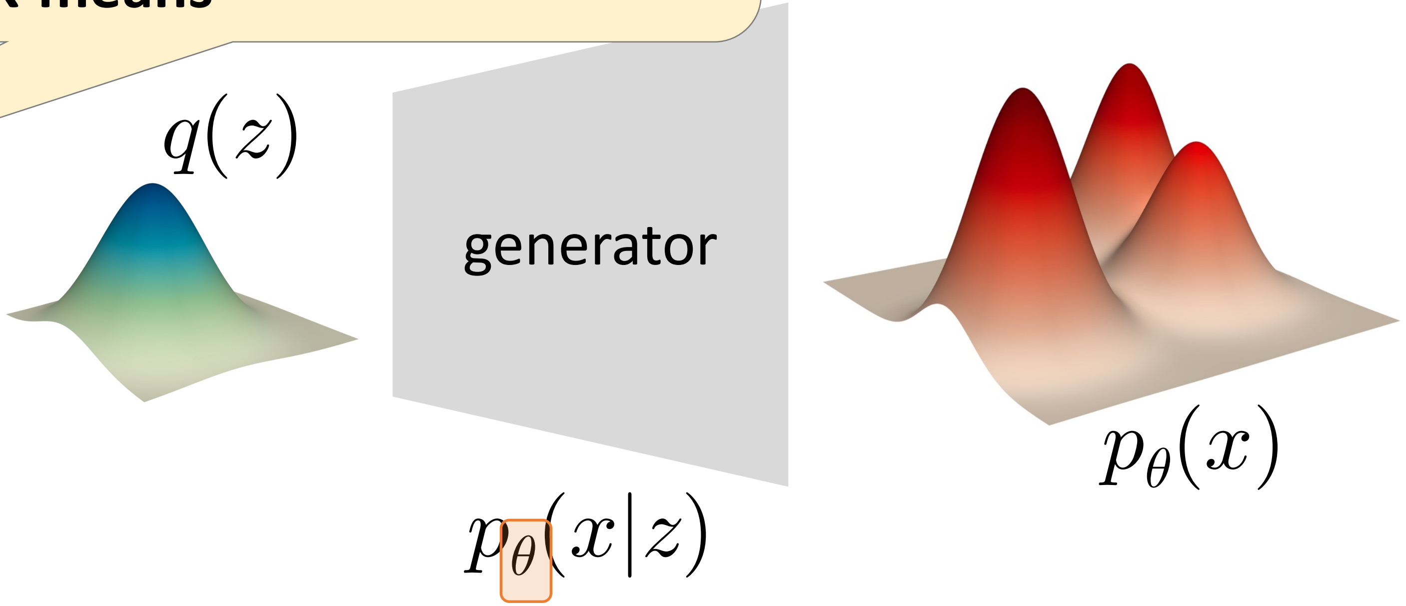
- Gaussian Mixtures
- **K-means**

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: $Q(\theta|\theta^{(t)}) = \mathbb{E}_{p_{data}(x)} \mathbb{E}_{p_{\theta^{(t)}}(z|x)}[\log p_\theta(x, z)]$

Details can be found in: Hastie et al. "The Elements of Statistical Learning".

A running example of EM: K-means

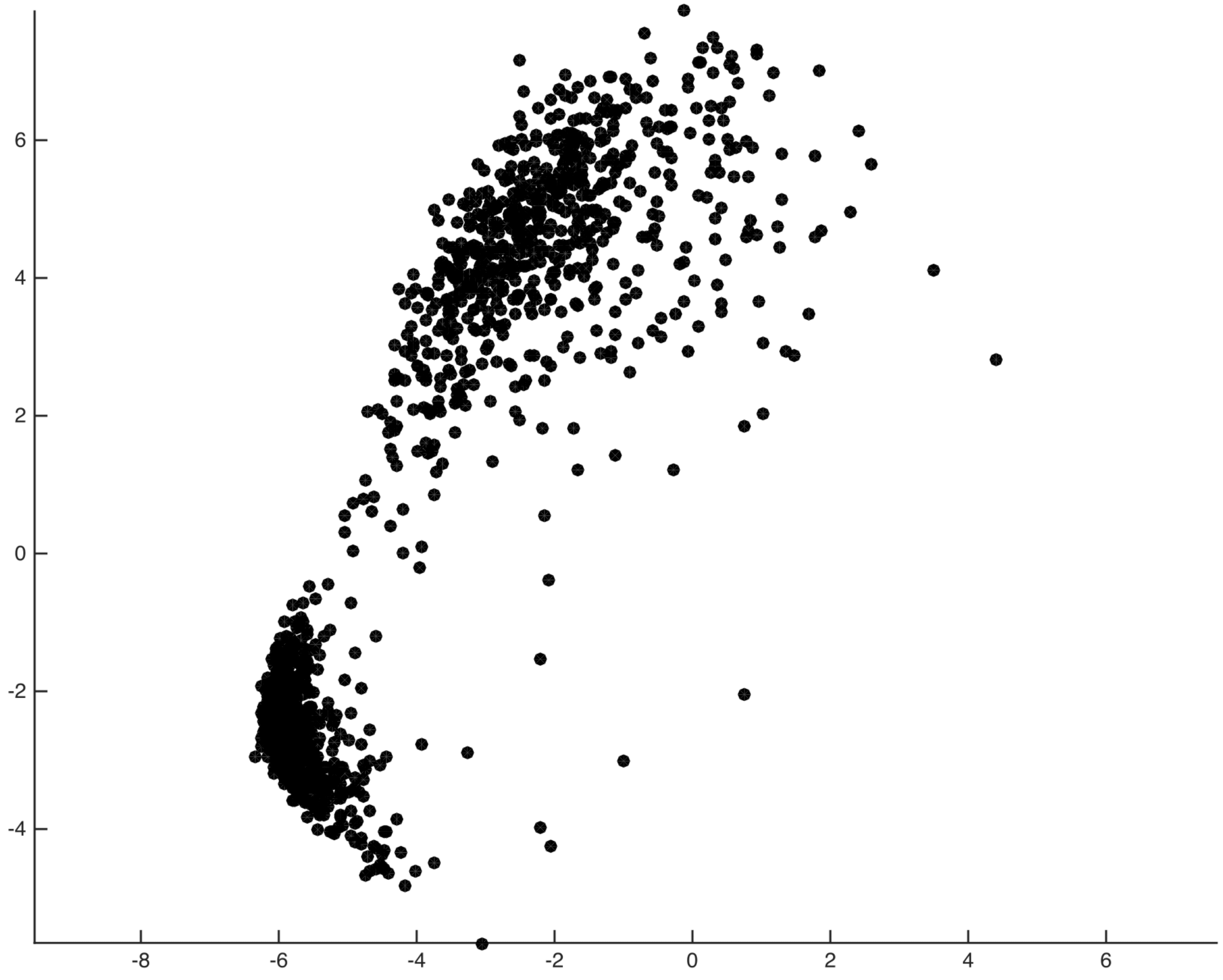


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

A running example of EM: K-means

- cluster centers: θ

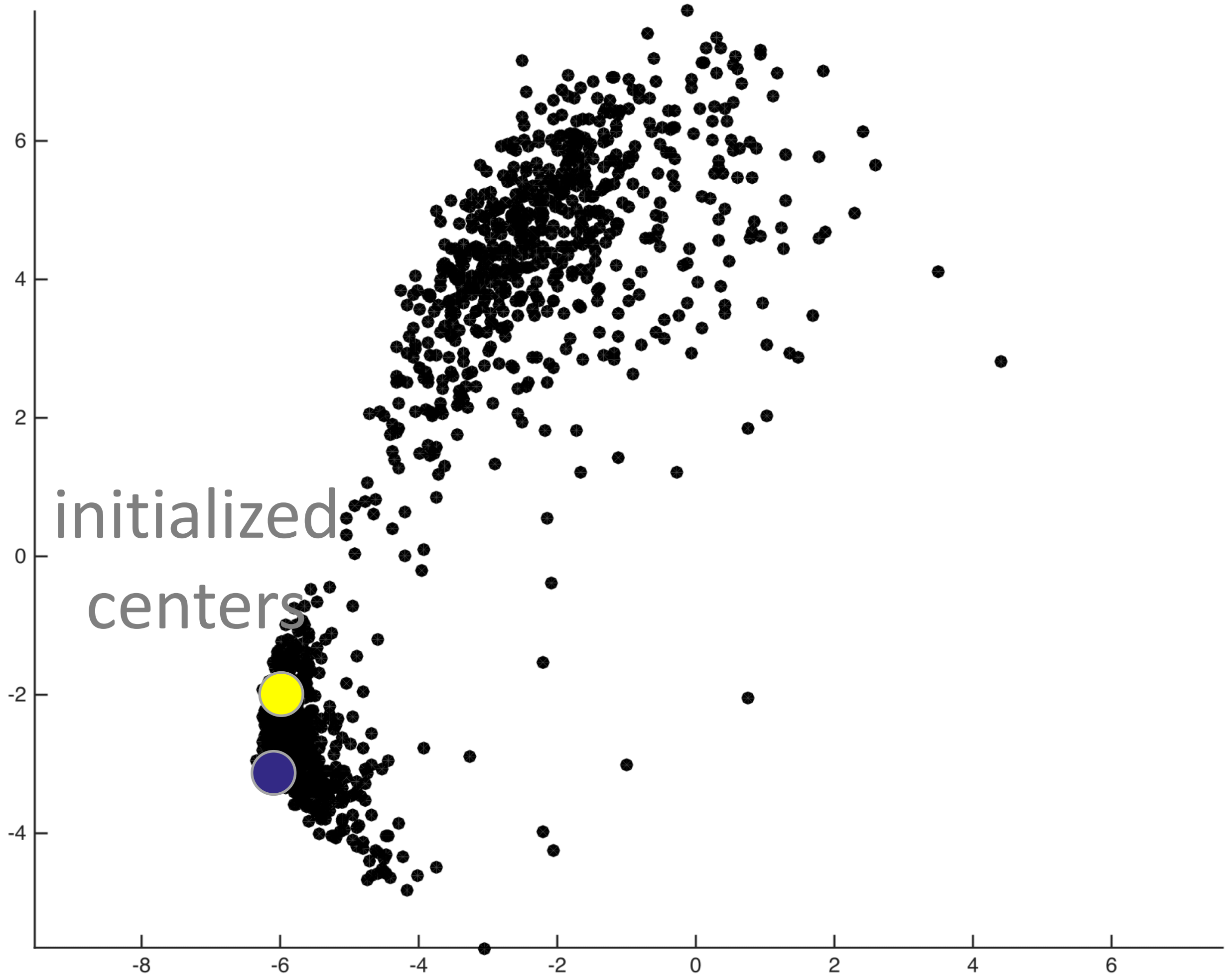


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

A running example of EM: K-means

- cluster centers: θ
- assignment:

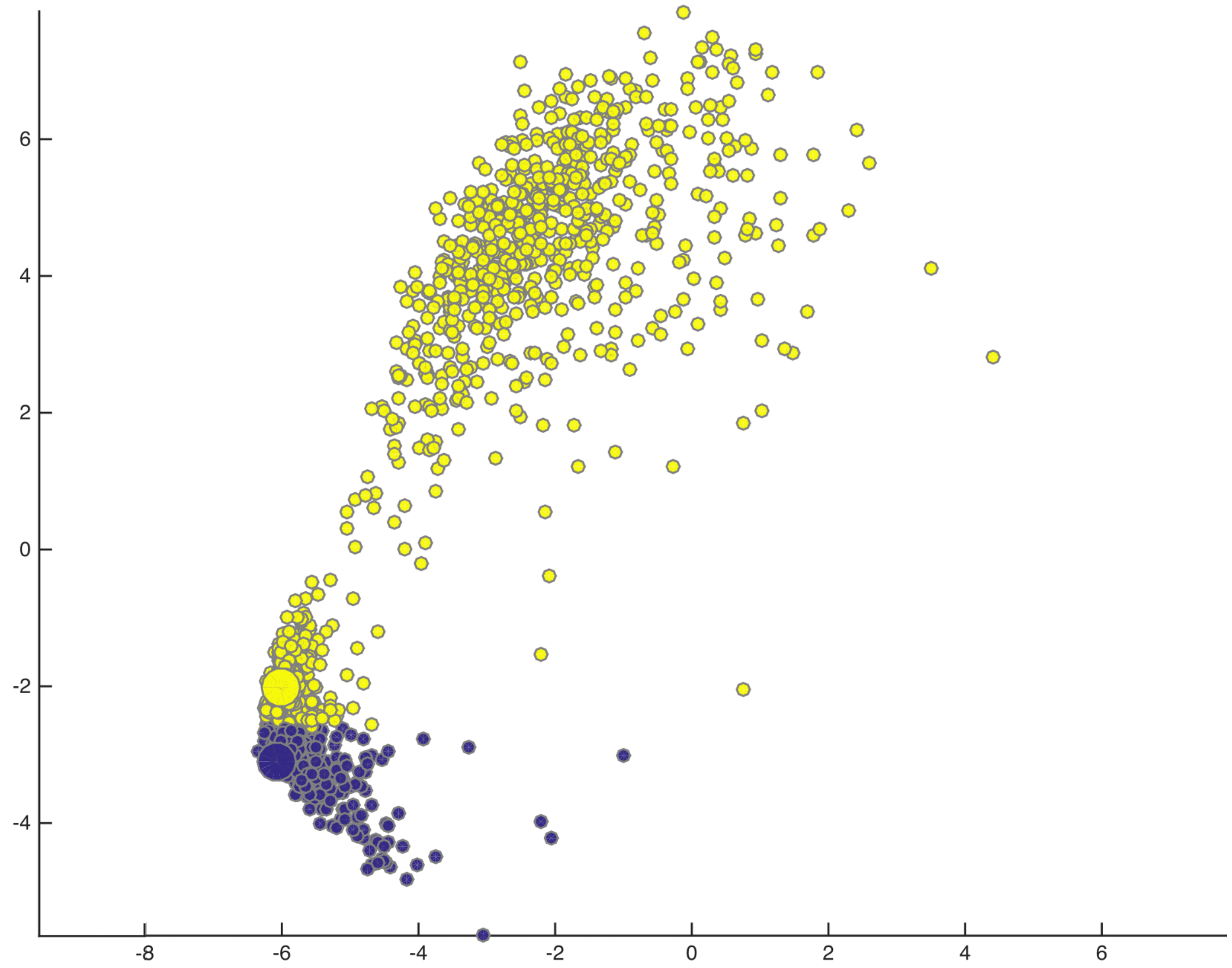


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

A running example of EM: K-means

- cluster centers: θ
- assignment: E-step

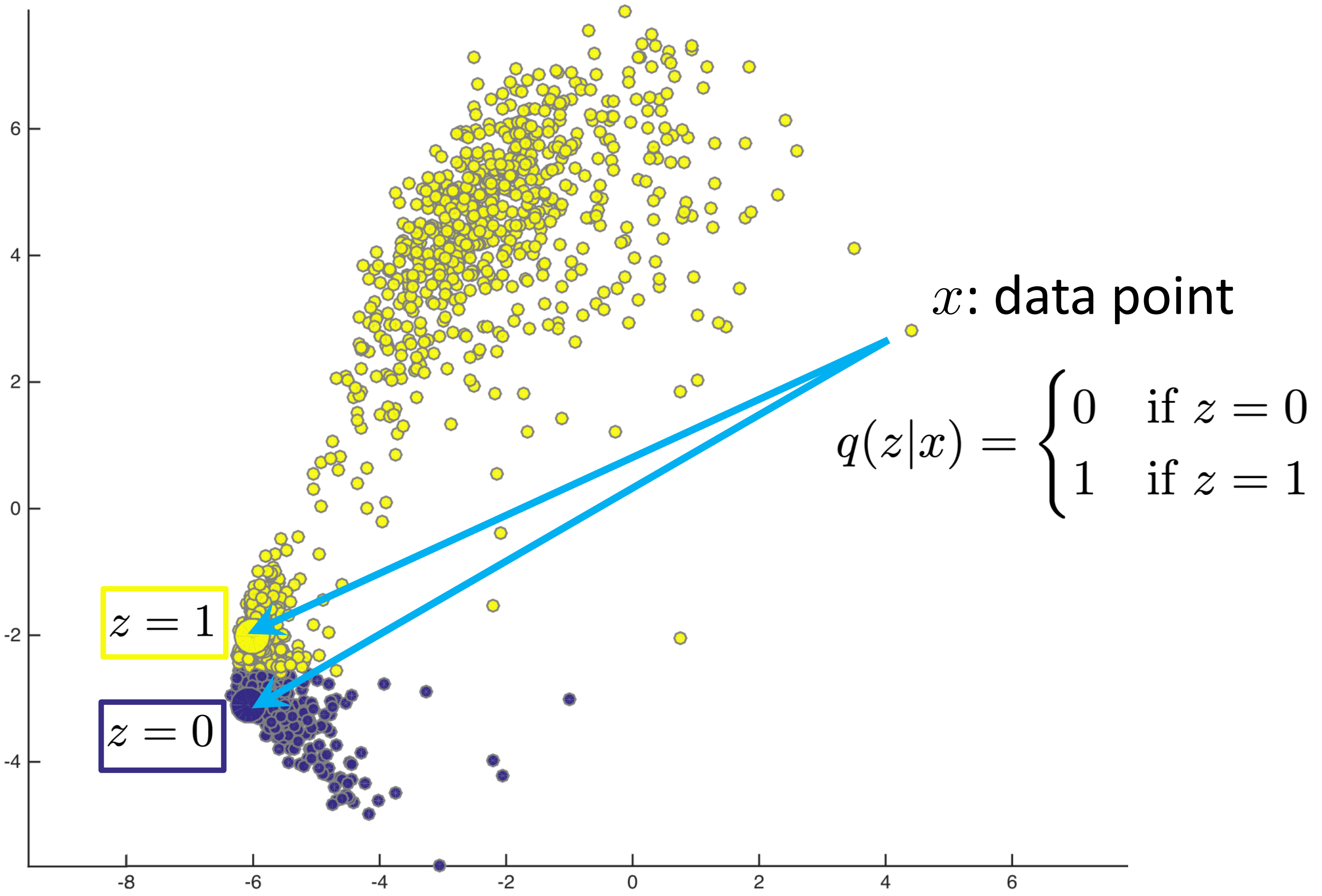
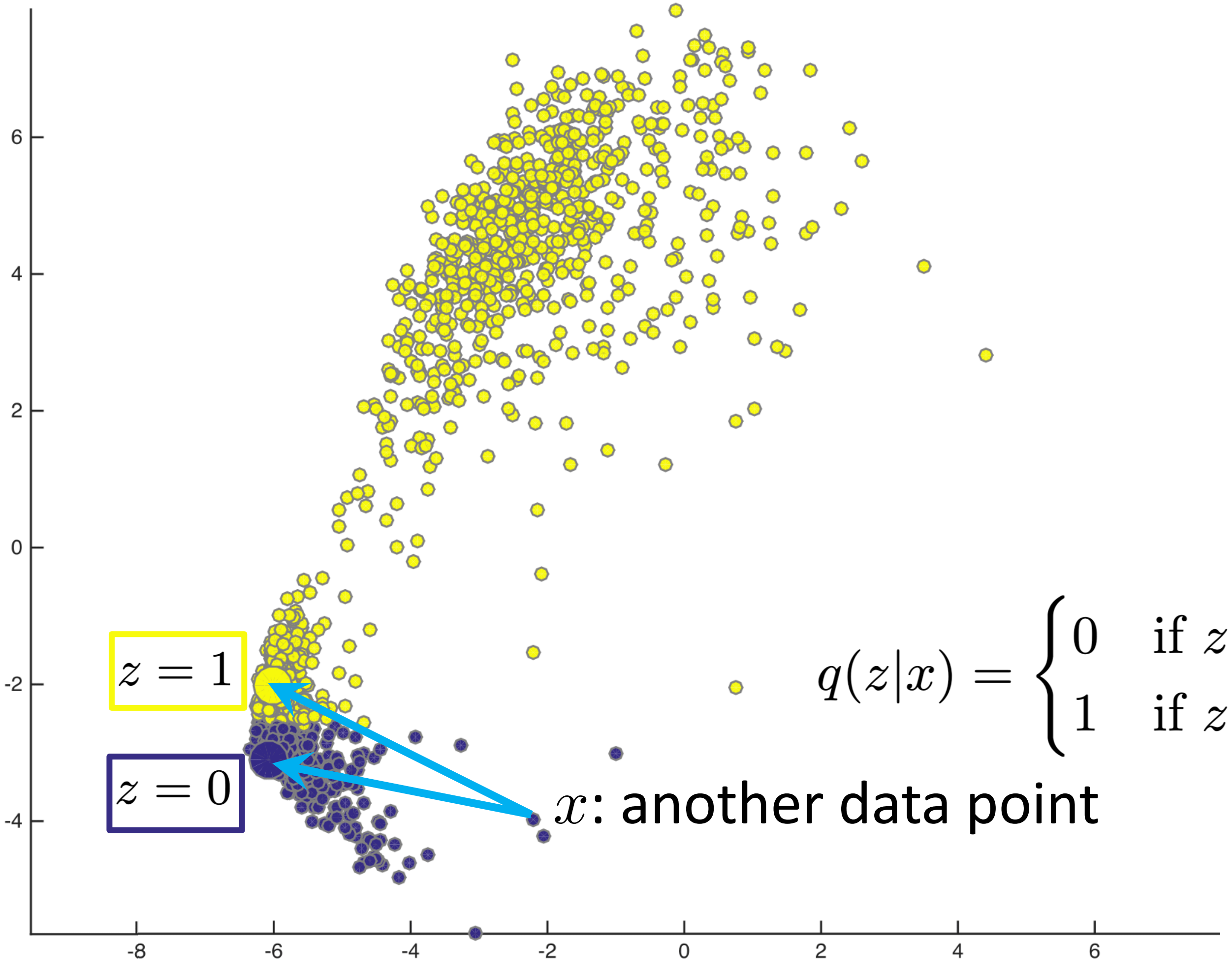


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

A running example of EM: K-means

- cluster centers: θ
- assignment: E-step

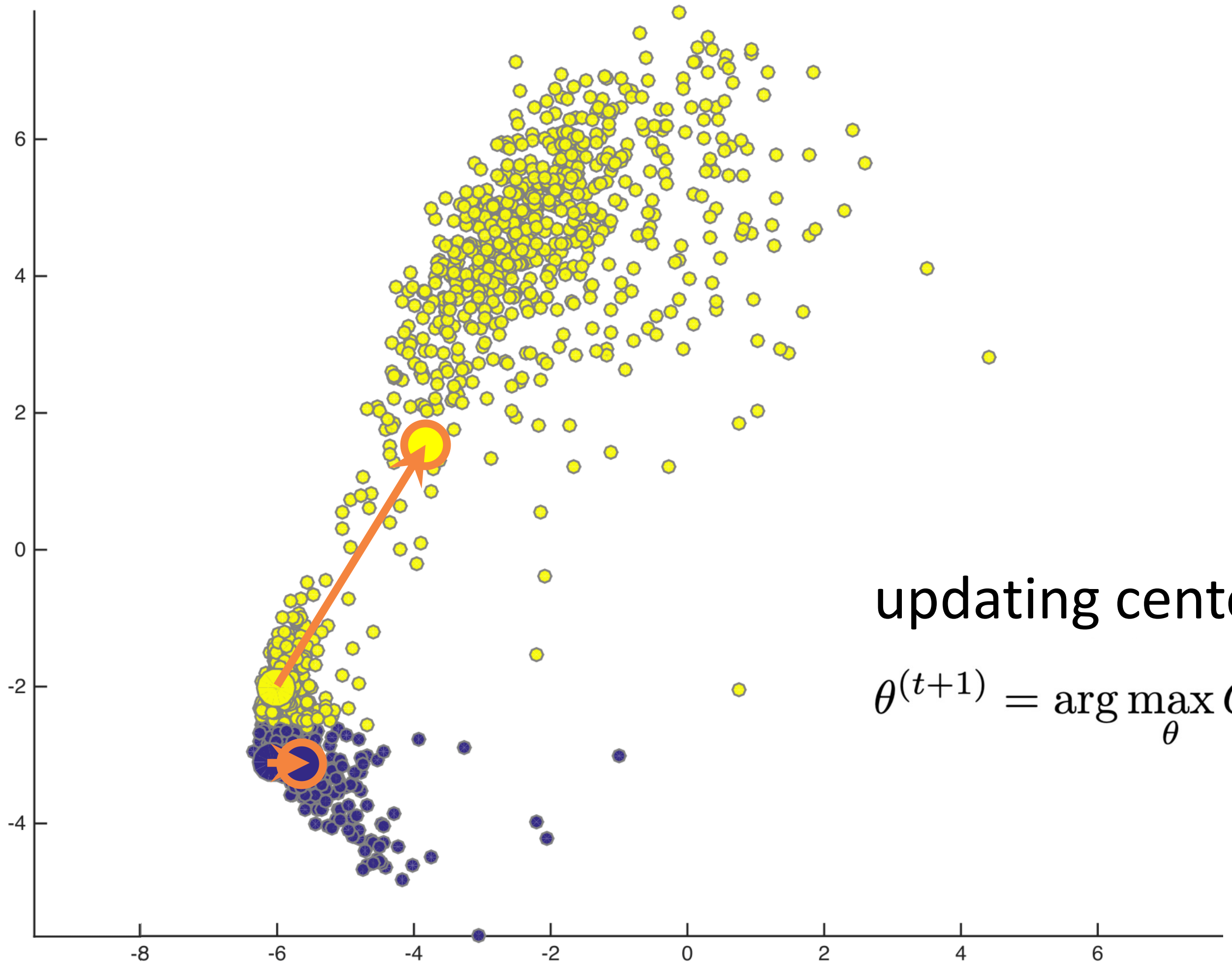


$$q(z|x) = \begin{cases} 0 & \text{if } z = 1 \\ 1 & \text{if } z = 0 \end{cases}$$

Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

A running example of EM: K-means

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



updating centers by:

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

Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

A running example of EM: K-means

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

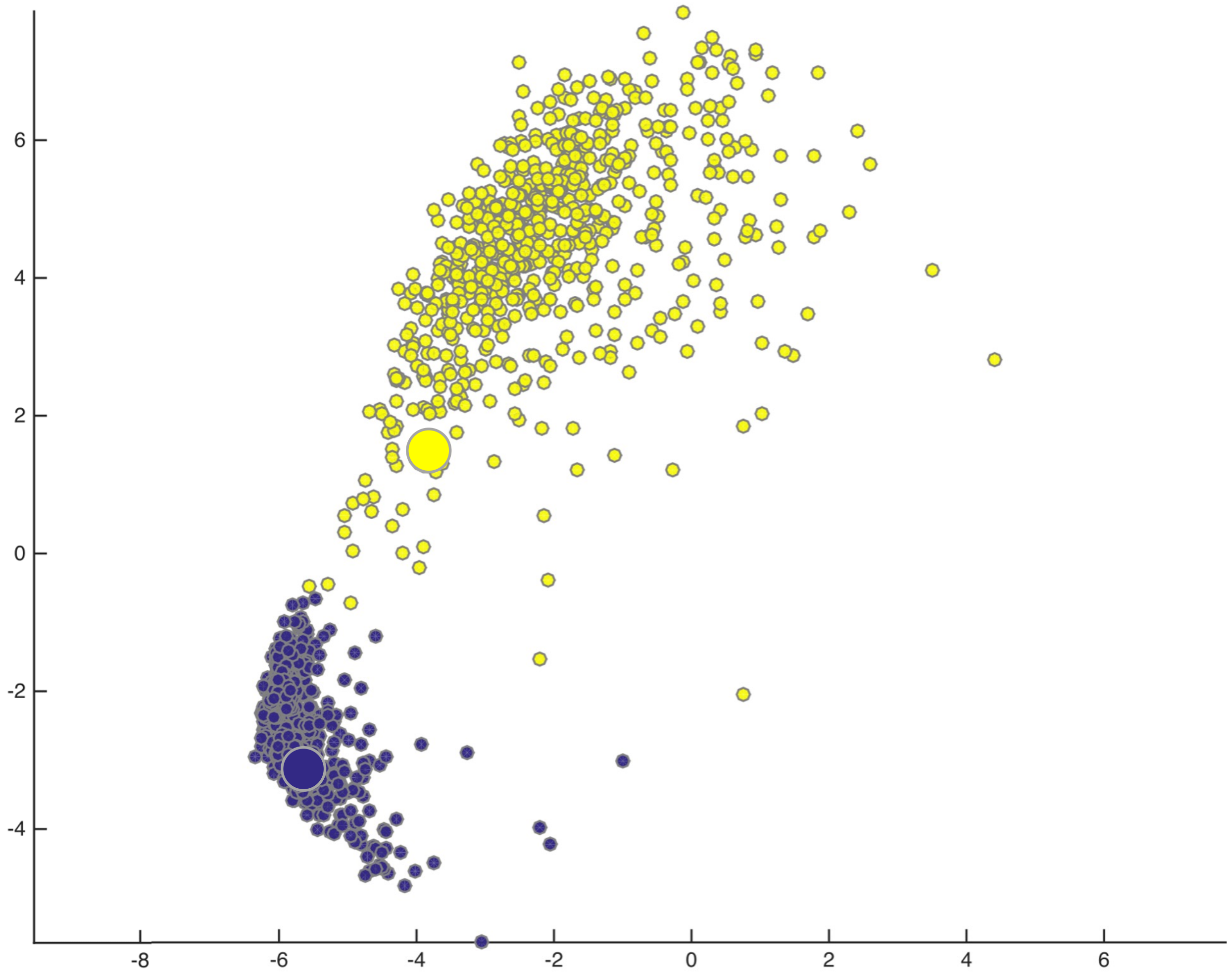
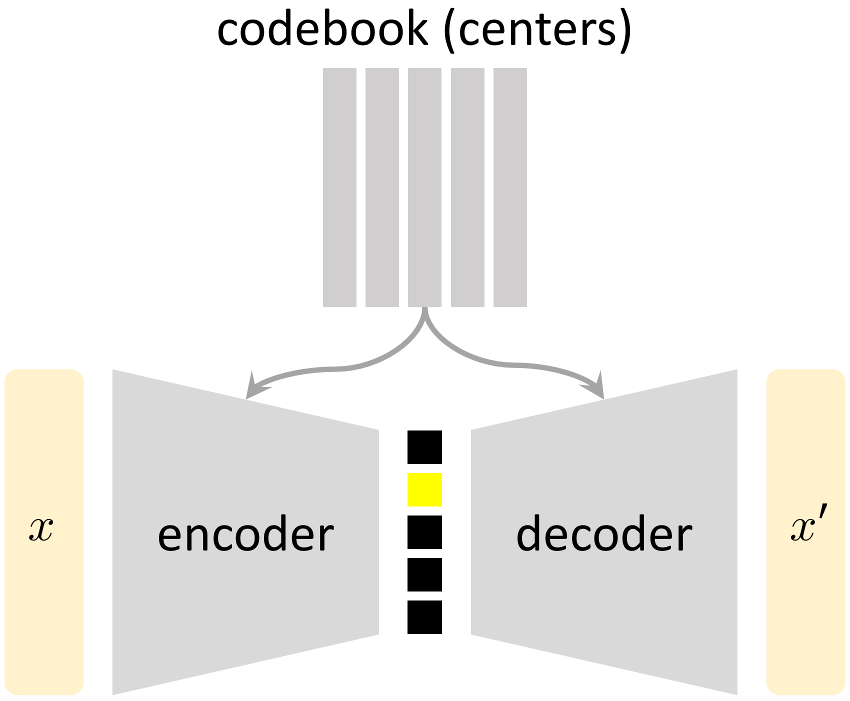
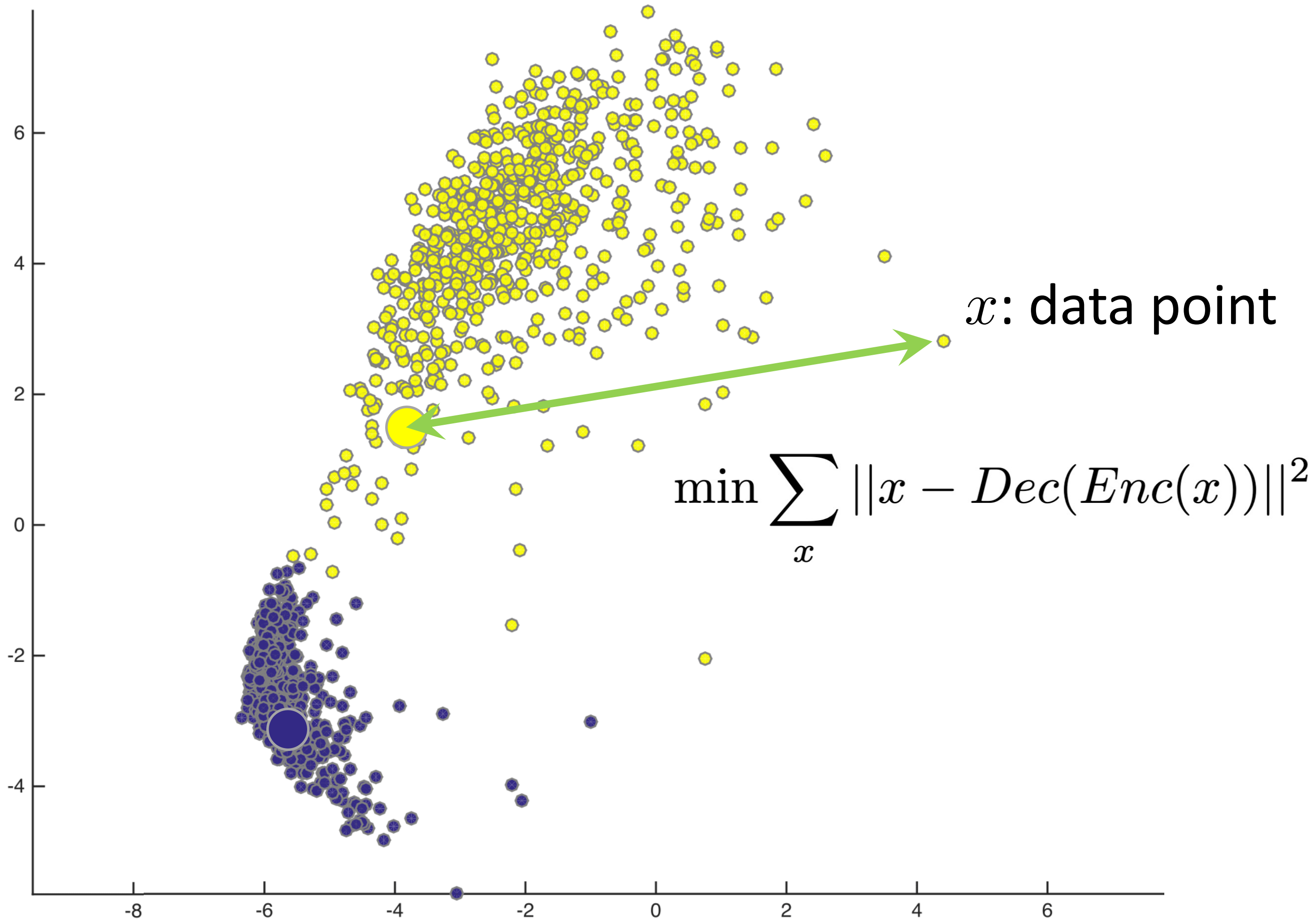


Figure adapted from: Marion Neumann, https://www.cse.wustl.edu/~m.neumann/sp2016/cse517/lecturenotes/lecturenote20.html

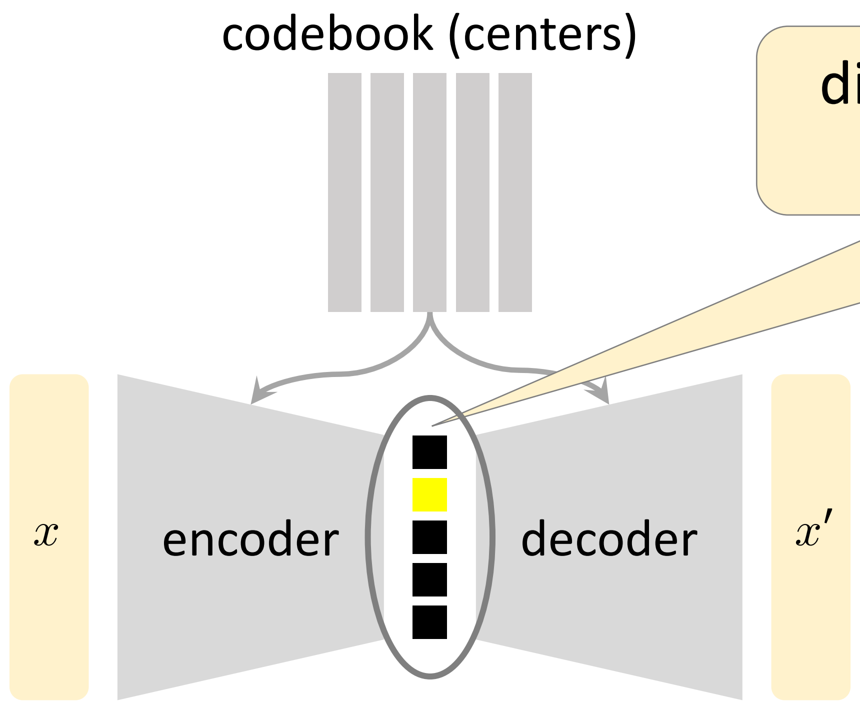
K-means as **Autoencoder**

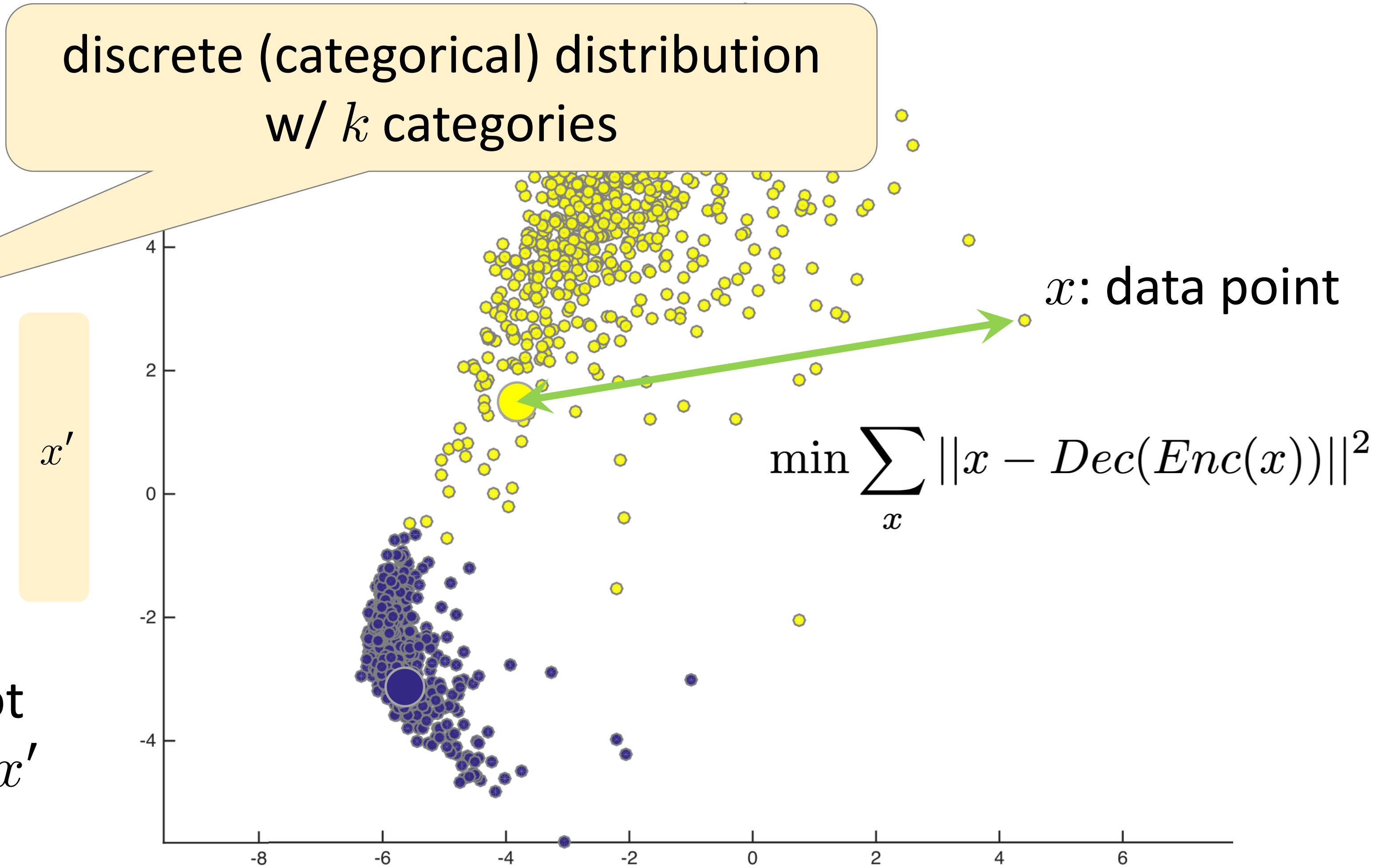


- encode: map x to one-hot
- decode: map one-hot to x'
- x' is a center



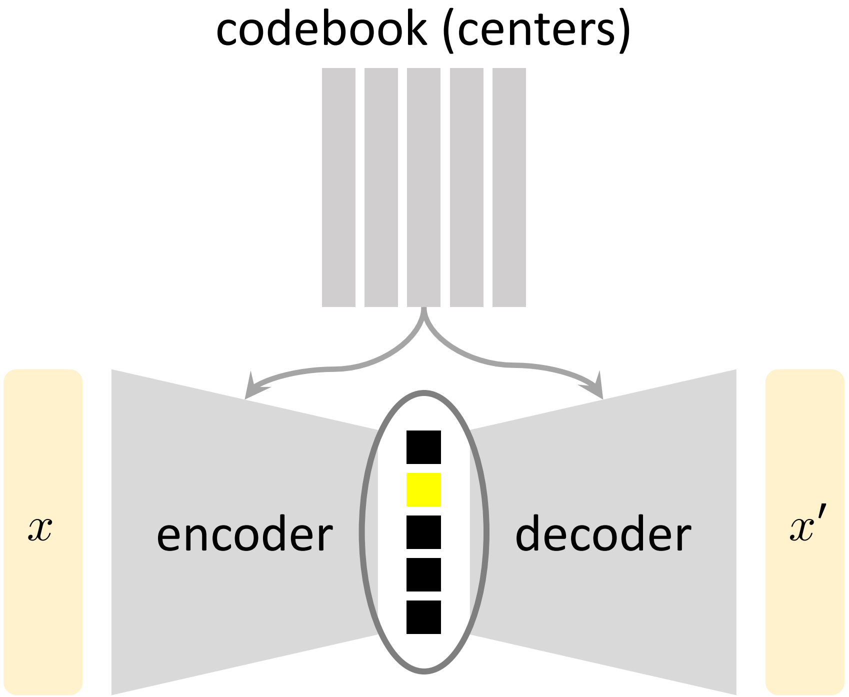
K-means as **Autoencoder**





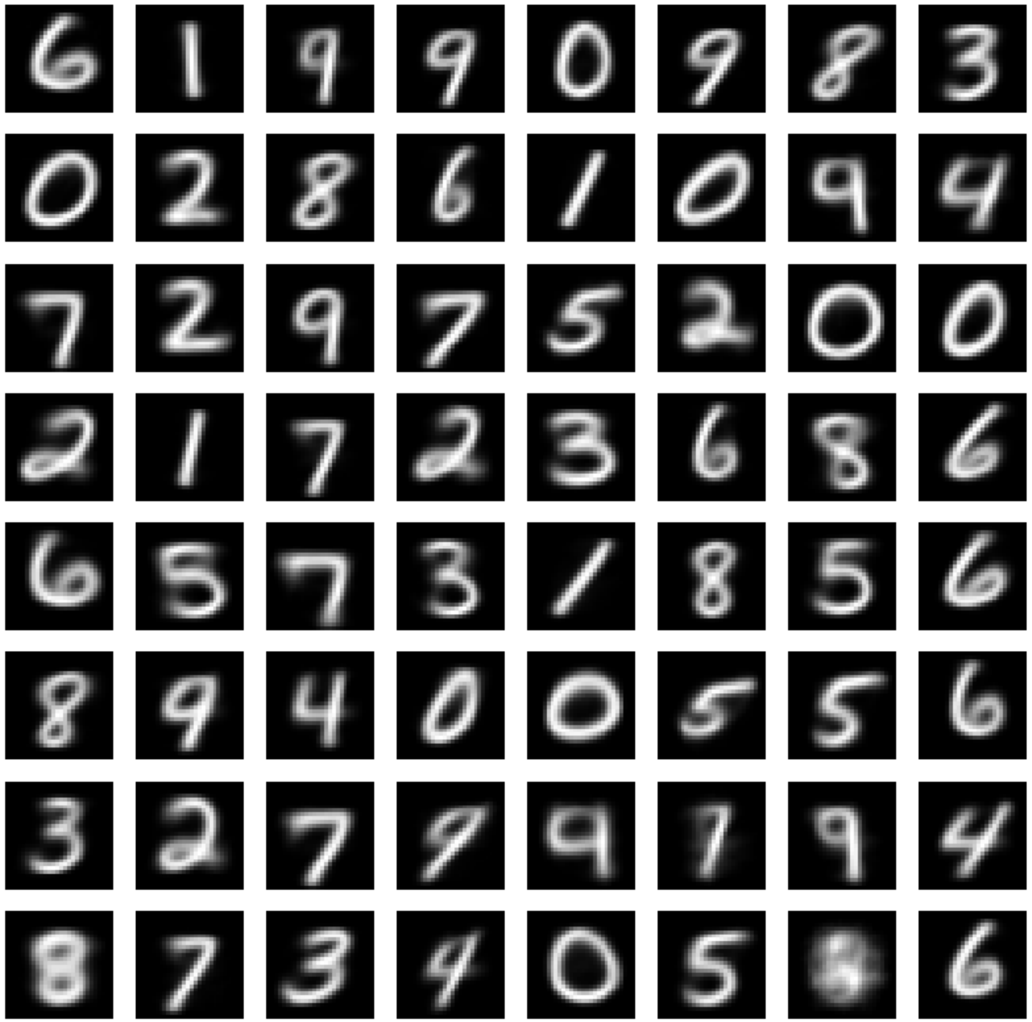
- encode: map x to one-hot
- decode: map one-hot to x'
- x' is a center

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

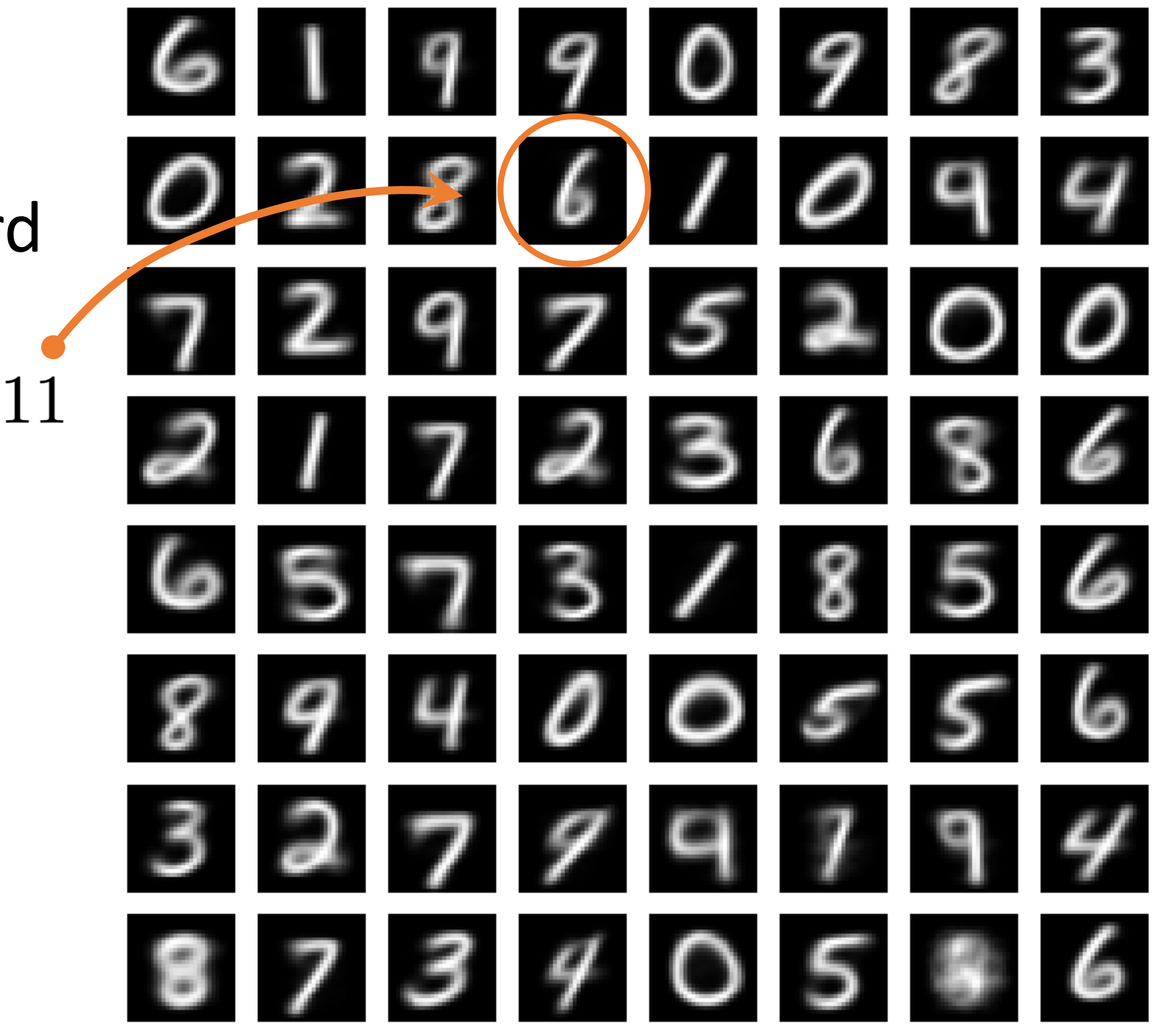


K-means as **Generative Models**

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

codebook on MNIST, k = 64

$z = 11$



- this is a valid generative model
- but not a "good" one
- but a good thought model

thus far, ...

- **VAE**: maximize ELBO
  - parameterize q by network
  - optimize by Stochastic Gradient Descent

- **EM**: maximize ELBO
  - parameterize q analytically
  - optimize by Coordinate Descent

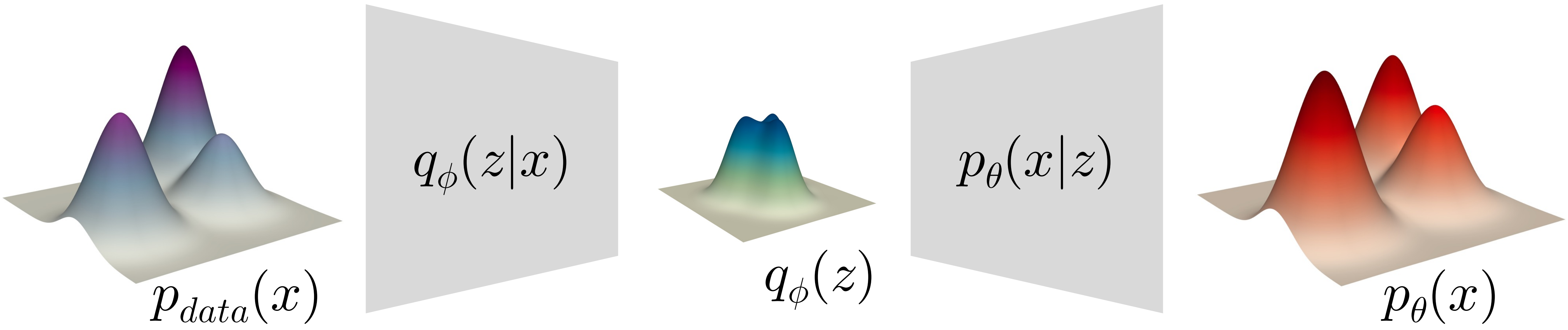
- **K-means**:
  - special case of EM; special case of AE
  - discrete distribution

- next: **VQ-VAE**

Vector Quantized VAE (VQ-VAE)

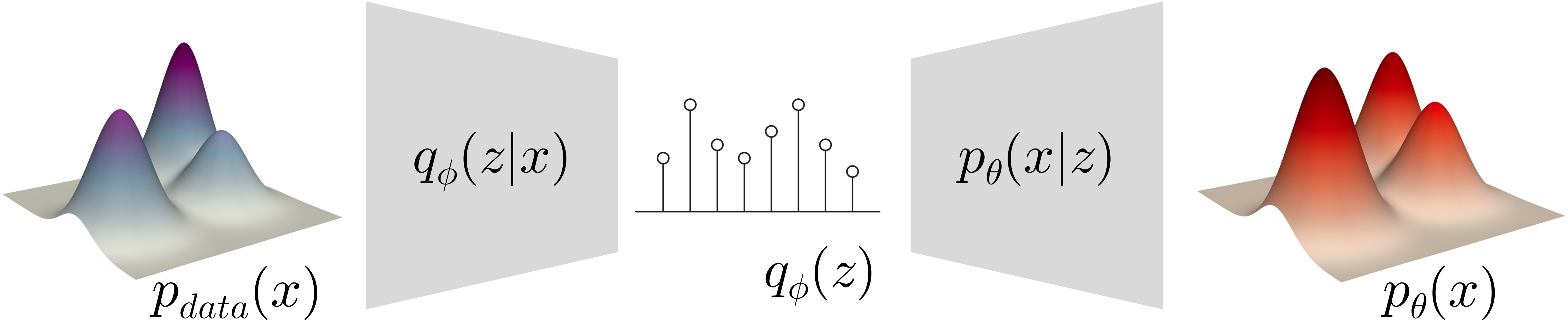
Recap

- Original VAE: latent variables are continuous



Discrete Latent Variables

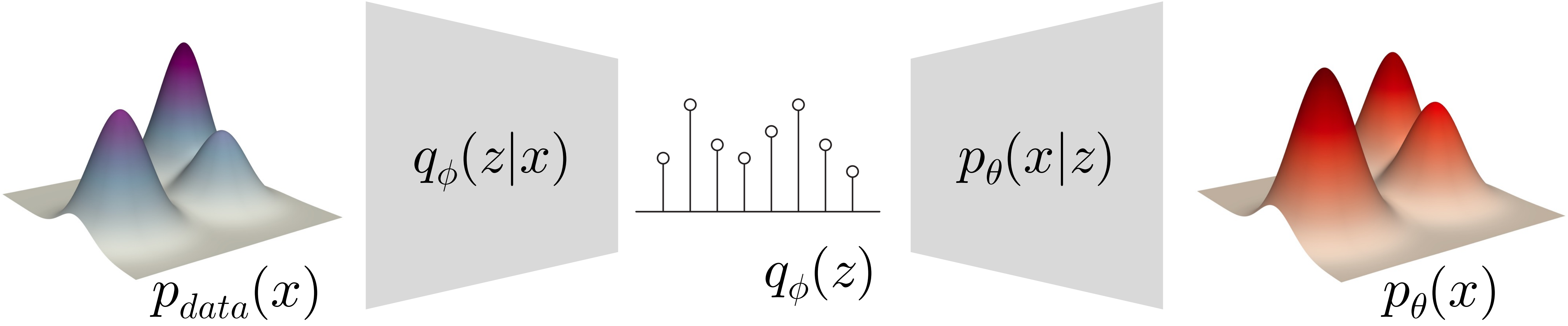
- model **multimodal** distributions
- **categorical**: no particular relation between numbers (SSN, zip code, ...)
- **symbolic**: language, speech, planning, ...



Discrete Latent Variables + VAE

Maximize ELBO

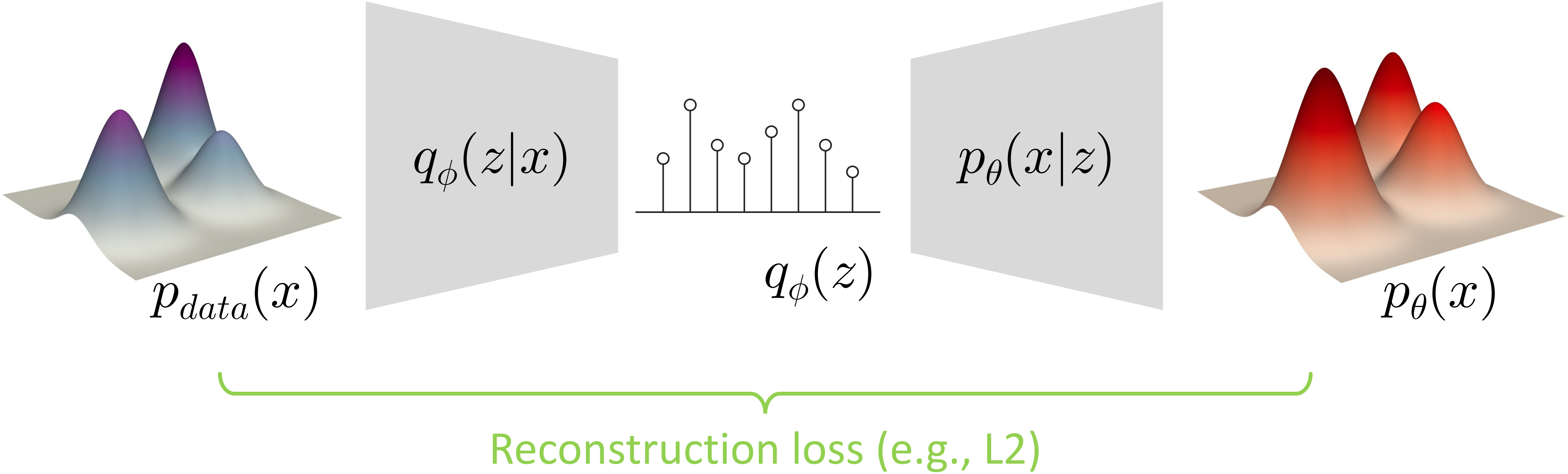
- Reconstruction loss: about x
- Regularization loss: about z (discrete)



Discrete Latent Variables + VAE

Reconstruction loss: about x

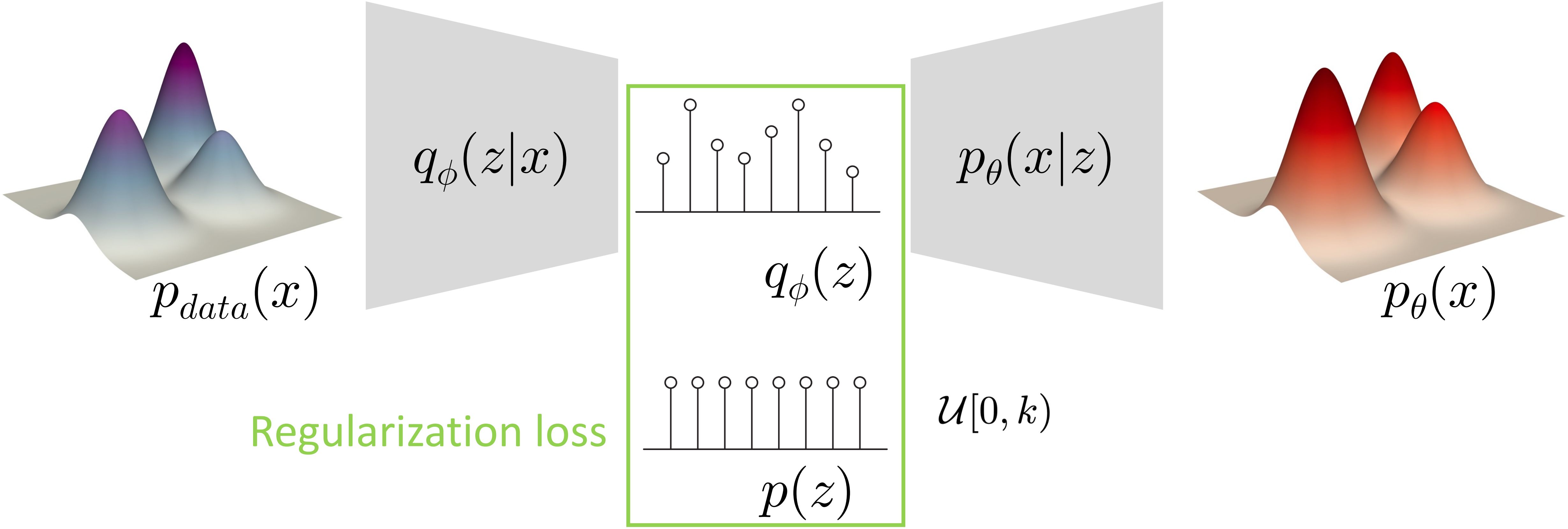
- same as VAE: $-\mathbb{E}_{z\sim q_\phi(z|x)}\left[\log p_\theta(x|z)\right]$



Discrete Latent Variables + VAE

Regularization loss: about z

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



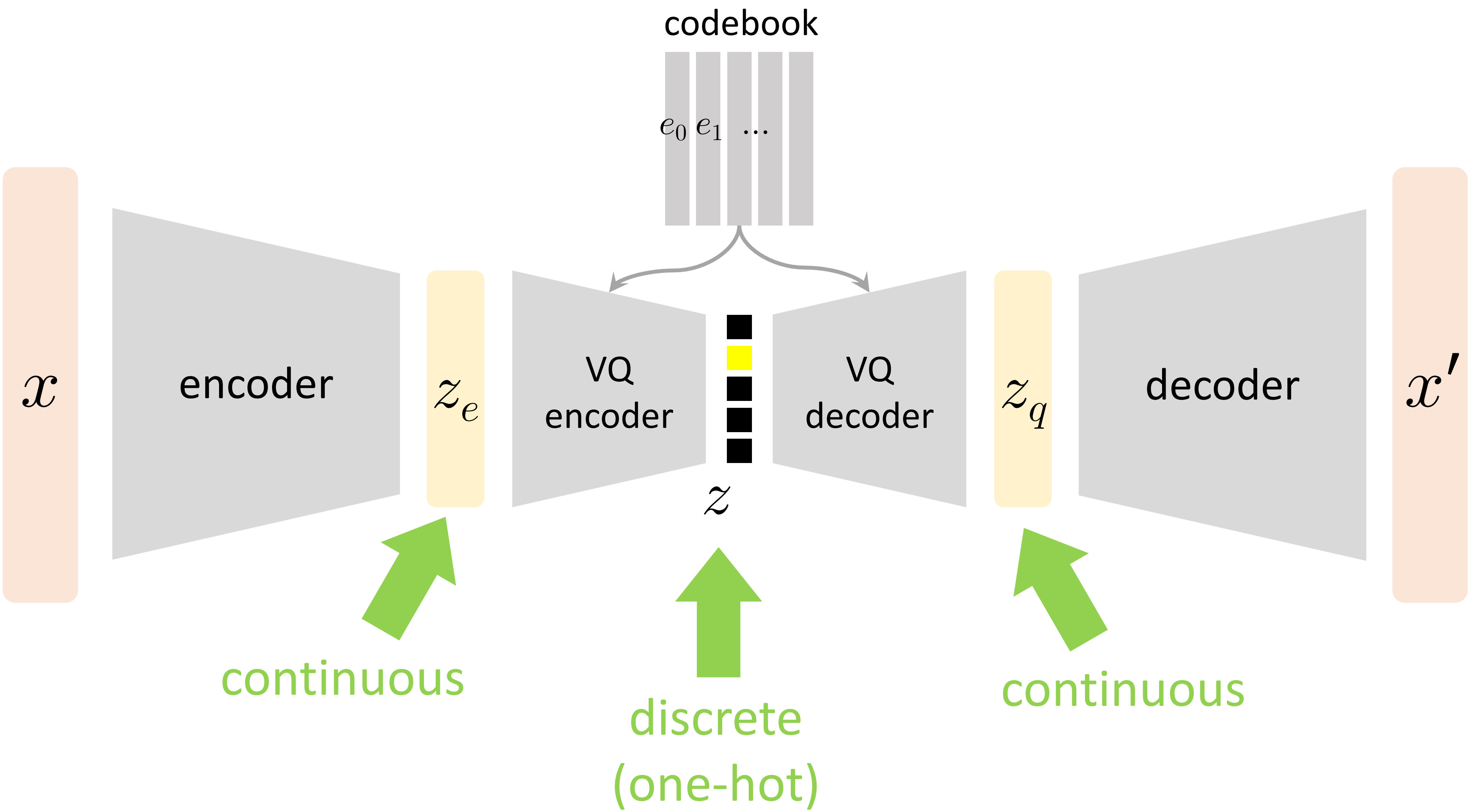
Discrete Latent Variables + VAE

Solution: K-means

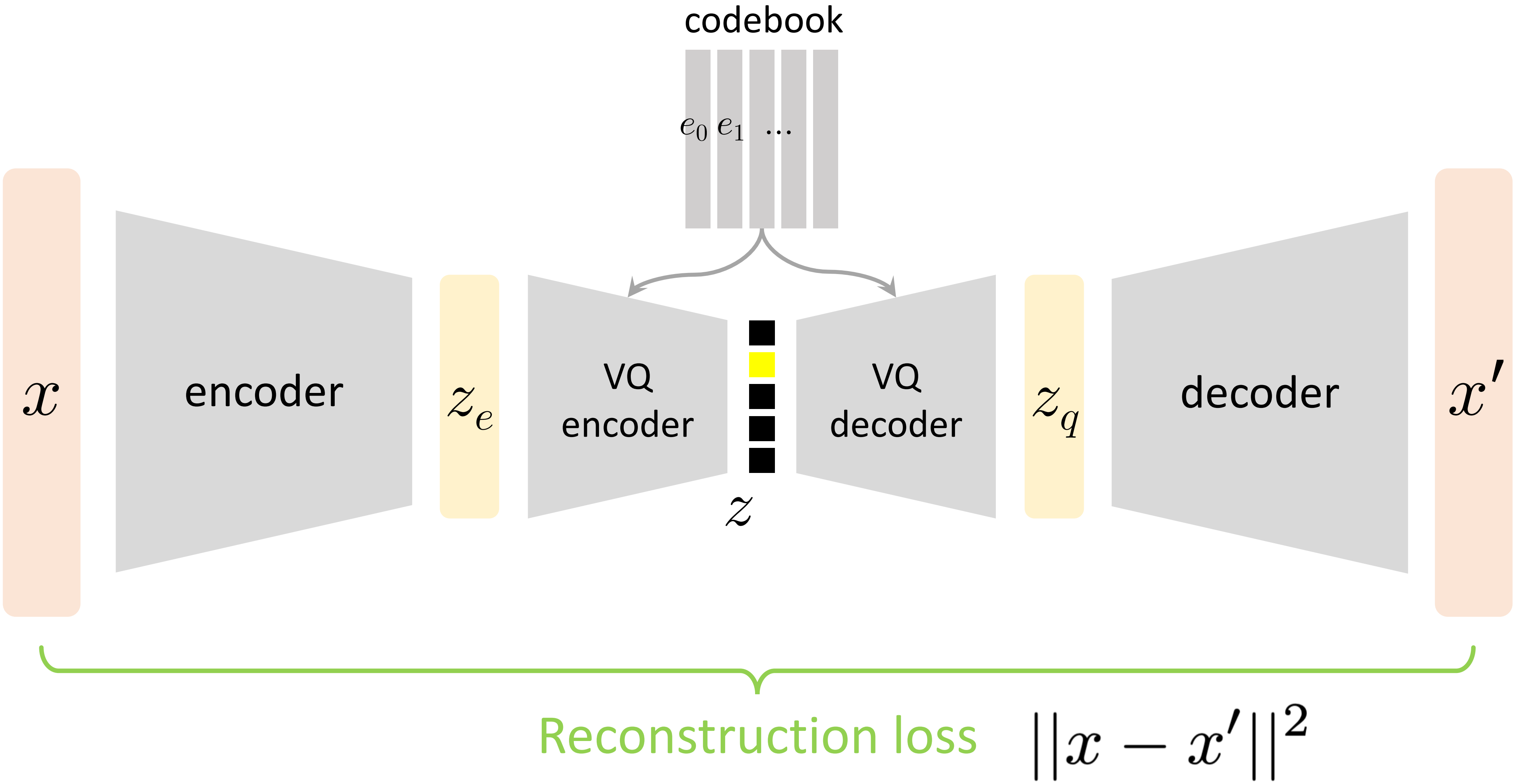
- K-means is autoencoding
- K-means has an objective function (reconstruction loss)
- K-means implicitly encourages codebook uniformity

This leads us to VQ-VAE ...

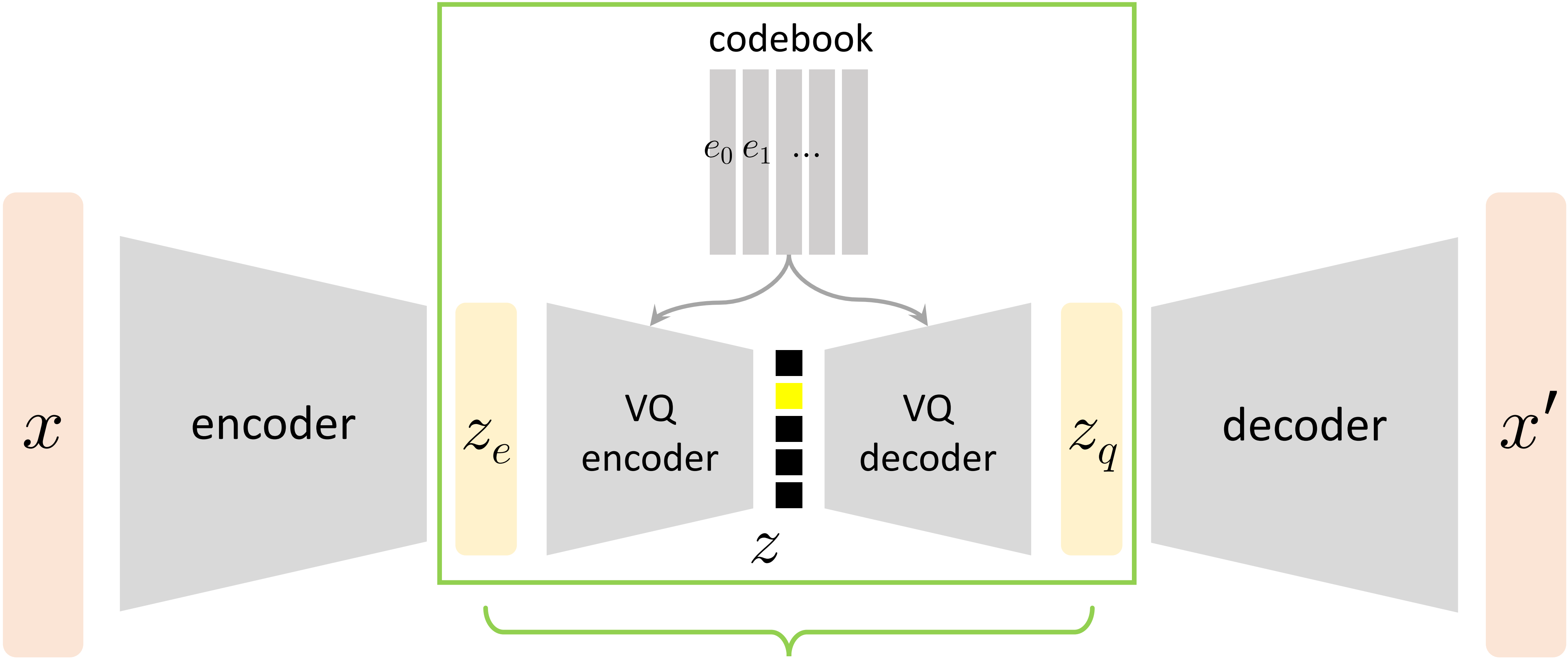
Vector Quantized VAE



Vector Quantized VAE



Vector Quantized VAE

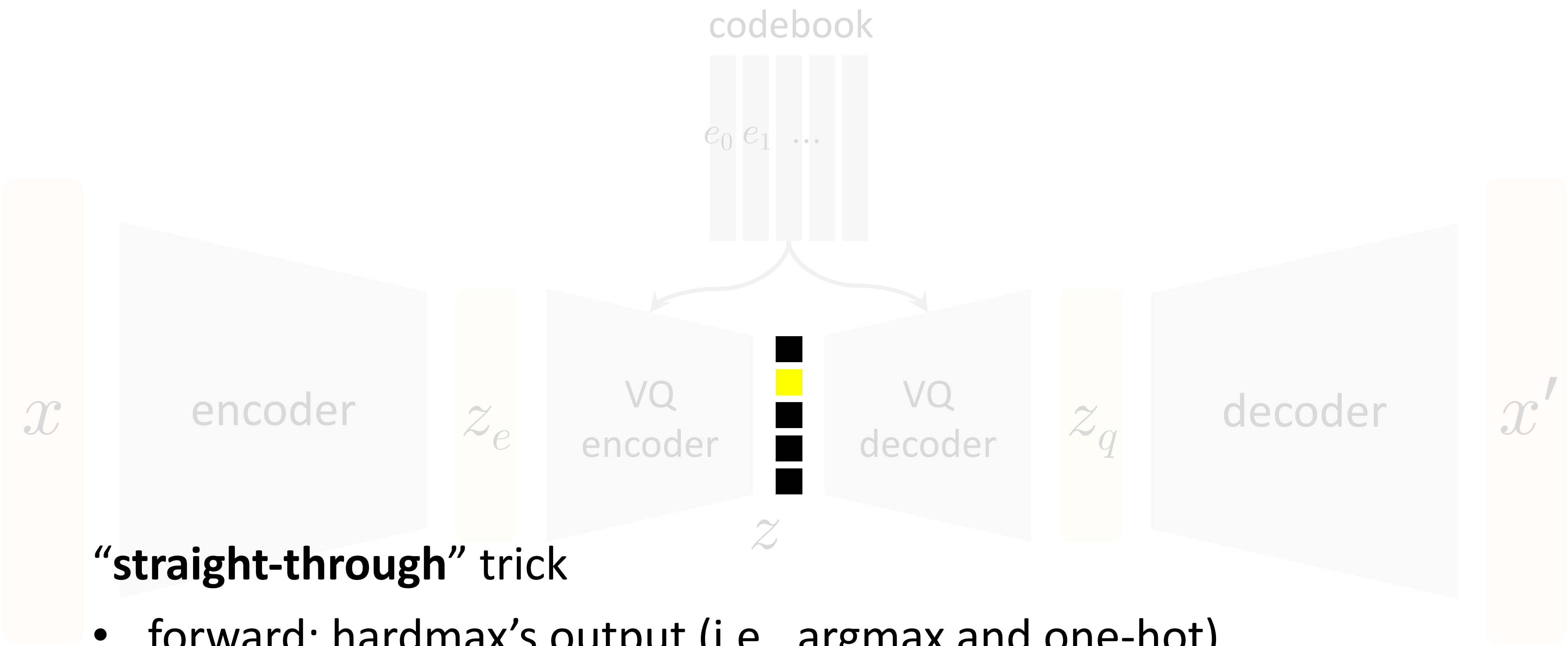


Regularization loss

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

*The VQ-VAE paper uses $\|\mathrm{sg}[z_e(x)] - e\|_2^2 + \beta\|z_e(x) - \mathrm{sg}[e]\|_2^2$ 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

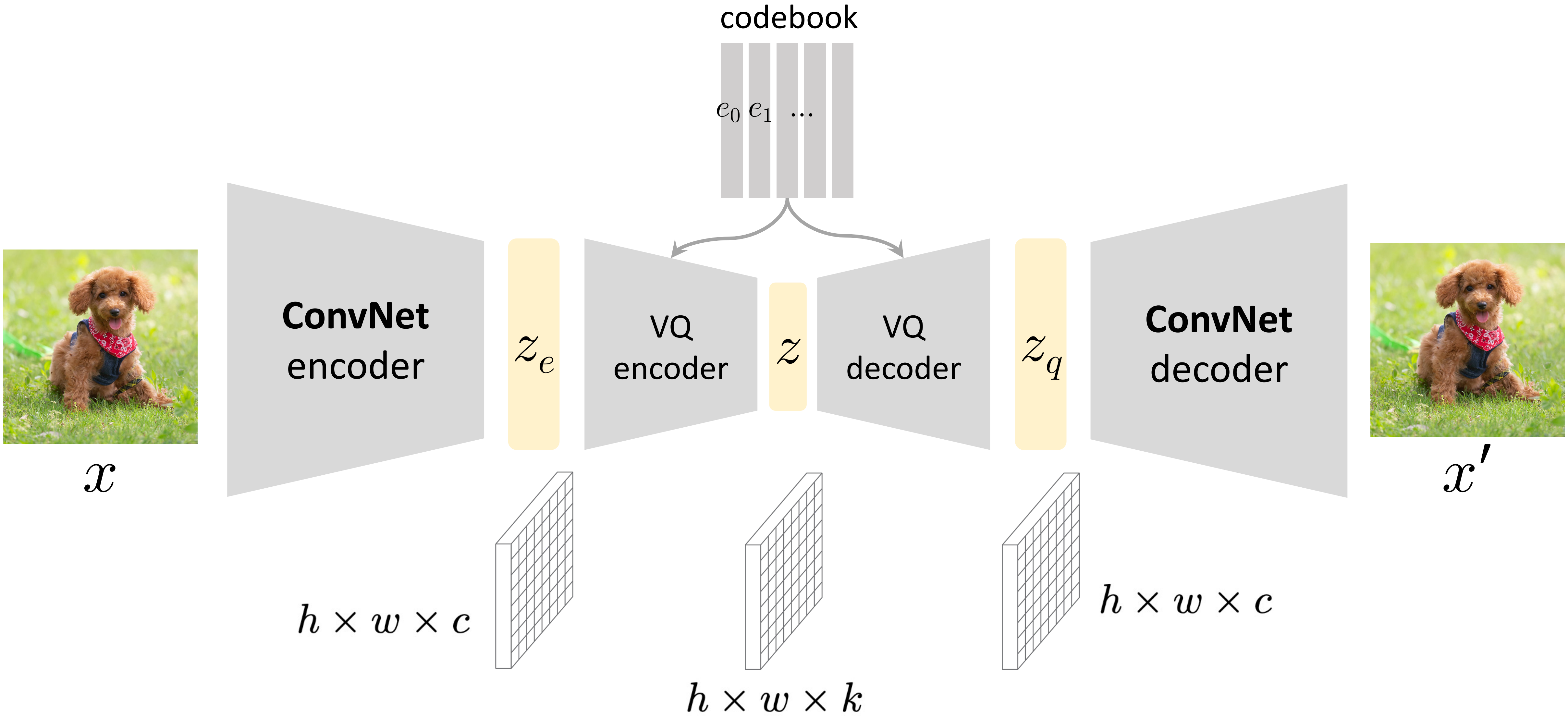
A single one-hot latent is not useful

- 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