

# Deep Generative Models

## Lecture 4

Roman Isachenko

Moscow Institute of Physics and Technology  
Yandex School of Data Analysis

2024, Autumn

# Recap of previous lecture

## Forward KL for flow model

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})) + \log |\det(\mathbf{J}_{\mathbf{f}})|$$

## Reverse KL for flow model

$$KL(p||\pi) = \mathbb{E}_{p(\mathbf{z})} [\log p(\mathbf{z}) - \log |\det(\mathbf{J}_{\mathbf{g}})| - \log \pi(\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{z}))]$$

## Flow KL duality

$$\arg \min_{\boldsymbol{\theta}} KL(\pi(\mathbf{x})||p(\mathbf{x}|\boldsymbol{\theta})) = \arg \min_{\boldsymbol{\theta}} KL(p(\mathbf{z}|\boldsymbol{\theta})||p(\mathbf{z}))$$

- ▶  $p(\mathbf{z})$  is a base distribution;  $\pi(\mathbf{x})$  is a data distribution;
- ▶  $\mathbf{z} \sim p(\mathbf{z})$ ,  $\mathbf{x} = \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{z})$ ,  $\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})$ ;
- ▶  $\mathbf{x} \sim \pi(\mathbf{x})$ ,  $\mathbf{z} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$ ,  $\mathbf{z} \sim p(\mathbf{z}|\boldsymbol{\theta})$ .

# Recap of previous lecture

## Posterior distribution (Bayes theorem)

$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\theta)p(\theta)}{\int p(\mathbf{x}|\theta)p(\theta)d\theta}$$

- ▶  $\mathbf{x}$  – observed variables;
- ▶  $\theta$  – unobserved variables (latent variables/parameters);
- ▶  $p(\mathbf{x}|\theta)$  – likelihood;
- ▶  $p(\mathbf{x}) = \int p(\mathbf{x}|\theta)p(\theta)d\theta$  – evidence;
- ▶  $p(\theta)$  – prior distribution;
- ▶  $p(\theta|\mathbf{x})$  – posterior distribution.

# Recap of previous lecture

## Latent variable models (LVM)

$$p(\mathbf{x}|\boldsymbol{\theta}) = \int p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})d\mathbf{z} = \int p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z})d\mathbf{z}.$$

## MLE problem for LVM

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg \max_{\boldsymbol{\theta}} \log p(\mathbf{X}|\boldsymbol{\theta}) = \arg \max_{\boldsymbol{\theta}} \sum_{i=1}^n \log p(\mathbf{x}_i|\boldsymbol{\theta}) = \\ &= \arg \max_{\boldsymbol{\theta}} \sum_{i=1}^n \log \int p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta})p(\mathbf{z}_i)d\mathbf{z}_i.\end{aligned}$$

## Naive Monte-Carlo estimation

$$p(\mathbf{x}|\boldsymbol{\theta}) = \int p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z})d\mathbf{z} = \mathbb{E}_{p(\mathbf{z})}p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) \approx \frac{1}{K} \sum_{k=1}^K p(\mathbf{x}|\mathbf{z}_k, \boldsymbol{\theta}),$$

where  $\mathbf{z}_k \sim p(\mathbf{z})$ .

# Recap of previous lecture

## ELBO derivation 1 (inequality)

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log \int p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) d\mathbf{z} \geq \mathbb{E}_q \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} = \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x})$$

## ELBO derivation 2 (equality)

$$\begin{aligned}\mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) &= \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} = \int q(\mathbf{z}) \log \frac{p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} = \\ &= \log p(\mathbf{x}|\boldsymbol{\theta}) - KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}))\end{aligned}$$

## Variational decomposition

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) + KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) \geq \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}).$$

# Recap of previous lecture

## Variational lower Bound (ELBO)

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) + KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) \geq \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}).$$

$$\mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) = \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} = \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z})||p(\mathbf{z}))$$

## Log-likelihood decomposition

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z})||p(\mathbf{z})) + KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

- Instead of maximizing incomplete likelihood, maximize ELBO

$$\max_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta}) \rightarrow \max_{q,\boldsymbol{\theta}} \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x})$$

- Maximization of ELBO by variational distribution  $q$  is equivalent to minimization of KL

$$\arg \max_q \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) \equiv \arg \min_q KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

# Outline

## 1. EM-algorithm

Amortized inference

ELBO gradients, reparametrization trick

## 2. Variational autoencoder (VAE)

## 3. Normalizing flows as VAE model

## 4. Discrete VAE latent representations

# Outline

## 1. EM-algorithm

Amortized inference

ELBO gradients, reparametrization trick

## 2. Variational autoencoder (VAE)

## 3. Normalizing flows as VAE model

## 4. Discrete VAE latent representations



# Outline

## 1. EM-algorithm

Amortized inference

ELBO gradients, reparametrization trick

## 2. Variational autoencoder (VAE)

## 3. Normalizing flows as VAE model

## 4. Discrete VAE latent representations

# Amortized variational inference

## E-step

$$q(\mathbf{z}) = \arg \max_q \mathcal{L}_{q, \theta^*}(\mathbf{x}) = \arg \min_q KL(q||p) = p(\mathbf{z}|\mathbf{x}, \theta^*).$$

$q(\mathbf{z})$  approximates true posterior distribution  $p(\mathbf{z}|\mathbf{x}, \theta^*)$ , that is why it is called **variational posterior**.

- ▶  $p(\mathbf{z}|\mathbf{x}, \theta^*)$  could be **intractable**;
- ▶  $q(\mathbf{z})$  is different for each object  $\mathbf{x}$ .

## Variational Bayes

Restrict a family of all possible distributions  $q(\mathbf{z})$  to a parametric class  $q(\mathbf{z}|\mathbf{x}, \phi)$  **conditioned on samples  $\mathbf{x}$  with parameters  $\phi$** .

- ▶ E-step

$$\phi_k = \phi_{k-1} + \eta \cdot \nabla_{\phi} \mathcal{L}_{\phi, \theta_{k-1}}(\mathbf{x})|_{\phi=\phi_{k-1}}$$

- ▶ M-step

$$\theta_k = \theta_{k-1} + \eta \cdot \nabla_{\theta} \mathcal{L}_{\phi_k, \theta}(\mathbf{x})|_{\theta=\theta_{k-1}}$$

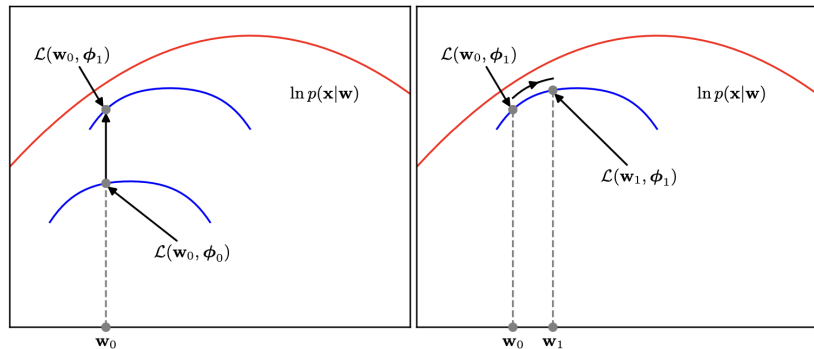
# Variational EM illustration

## ► E-step

$$\phi_k = \phi_{k-1} + \eta \nabla_{\phi} \mathcal{L}_{\phi, \theta_{k-1}}(\mathbf{x}) \Big|_{\phi=\phi_{k-1}}$$

## ► M-step

$$\theta_k = \theta_{k-1} + \eta \nabla_{\theta} \mathcal{L}_{\phi_k, \theta}(\mathbf{x}) \Big|_{\theta=\theta_{k-1}}$$



# Variational EM-algorithm

## ELBO

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x}) + KL(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) \geq \mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x}).$$

$$\mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}))$$

### ► E-step

$$\phi_k = \phi_{k-1} + \eta \cdot \nabla_{\phi} \mathcal{L}_{\phi,\boldsymbol{\theta}_{k-1}}(\mathbf{x})|_{\phi=\phi_{k-1}},$$

where  $\phi$  – parameters of the variational posterior distribution  $q(\mathbf{z}|\mathbf{x}, \phi)$ .

### ► M-step

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \eta \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\phi_k,\boldsymbol{\theta}}(\mathbf{x})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{k-1}},$$

where  $\boldsymbol{\theta}$  – parameters of the generative distribution  $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ .

Now all that is left is to obtain **unbiased** Monte Carlo estimates of the gradients:  $\nabla_{\phi} \mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x})$ ,  $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\phi,\boldsymbol{\theta}}(\mathbf{x})$ .

# Outline

## 1. EM-algorithm

Amortized inference

ELBO gradients, reparametrization trick

## 2. Variational autoencoder (VAE)

## 3. Normalizing flows as VAE model

## 4. Discrete VAE latent representations

## ELBO gradients, (M-step, $\nabla_{\theta} \mathcal{L}_{\phi, \theta}(\mathbf{x})$ )

$$\mathcal{L}_{q, \theta}(\mathbf{x}) = \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \theta) - KL(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z}))$$

M-step:  $\nabla_{\theta} \mathcal{L}_{\phi, \theta}(\mathbf{x})$

$$\begin{aligned} \nabla_{\theta} \mathcal{L}_{\phi, \theta}(\mathbf{x}) &= \nabla_{\theta} \int q(\mathbf{z}|\mathbf{x}, \phi) \log p(\mathbf{x}|\mathbf{z}, \theta) d\mathbf{z} = \\ &= \int q(\mathbf{z}|\mathbf{x}, \phi) \nabla_{\theta} \log p(\mathbf{x}|\mathbf{z}, \theta) d\mathbf{z} \approx \\ &\approx \nabla_{\theta} \log p(\mathbf{x}|\mathbf{z}^*, \theta), \quad \mathbf{z}^* \sim q(\mathbf{z}|\mathbf{x}, \phi). \end{aligned}$$

Naive Monte-Carlo estimation

$$p(\mathbf{x}|\theta) = \int p(\mathbf{x}|\mathbf{z}, \theta) p(\mathbf{z}) d\mathbf{z} \approx \frac{1}{K} \sum_{k=1}^K p(\mathbf{x}|\mathbf{z}_k, \theta), \quad \mathbf{z}_k \sim p(\mathbf{z}).$$

The variational posterior  $q(\mathbf{z}|\mathbf{x}, \phi)$  assigns typically more probability mass in a smaller region than the prior  $p(\mathbf{z})$ .

## ELBO gradients, (E-step, $\nabla_{\phi} \mathcal{L}_{\phi, \theta}(\mathbf{x})$ )

E-step:  $\nabla_{\phi} \mathcal{L}_{\phi, \theta}(\mathbf{x})$

Difference from M-step: density function  $q(\mathbf{z}|\mathbf{x}, \phi)$  depends on the parameters  $\phi$ , it is impossible to use the Monte-Carlo estimation:

$$\begin{aligned}\nabla_{\phi} \mathcal{L}_{\phi, \theta}(\mathbf{x}) &= \nabla_{\phi} \int q(\mathbf{z}|\mathbf{x}, \phi) \log p(\mathbf{x}|\mathbf{z}, \theta) d\mathbf{z} - \nabla_{\phi} KL(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z})) \\ &\neq \int q(\mathbf{z}|\mathbf{x}, \phi) \nabla_{\phi} \log p(\mathbf{x}|\mathbf{z}, \theta) d\mathbf{z} - \nabla_{\phi} KL(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z}))\end{aligned}$$

## Reparametrization trick (LOTUS trick)

Suppose that  $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}, \phi)$  is a random variable that is induced by the random variable  $\epsilon \sim p(\epsilon)$  using the deterministic transform  $\mathbf{z} = \mathbf{g}_{\phi}(\mathbf{x}, \epsilon)$ . Then

$$\mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}, \phi)} \mathbf{f}(\mathbf{z}) = \mathbb{E}_{\epsilon \sim p(\epsilon)} \mathbf{f}(\mathbf{g}_{\phi}(\mathbf{x}, \epsilon))$$

Note that LHS takes the expectation by the parametric distribution  $q(\mathbf{z}|\mathbf{x}, \phi)$  and the RHS uses non-parametric distribution  $p(\epsilon)$ .

# ELBO gradients, (E-step, $\nabla_{\phi} \mathcal{L}_{\phi, \theta}(\mathbf{x})$ )

## Reparametrization trick (LOTUS trick)

$$\begin{aligned}\nabla_{\phi} \int q(\mathbf{z}|\mathbf{x}, \phi) \mathbf{f}(\mathbf{z}) d\mathbf{z} &= \nabla_{\phi} \int p(\epsilon) \mathbf{f}(\mathbf{g}_{\phi}(\mathbf{x}, \epsilon)) d\epsilon \\ &= \int p(\epsilon) \nabla_{\phi} \mathbf{f}(\mathbf{g}_{\phi}(\mathbf{x}, \epsilon)) d\epsilon \approx \nabla_{\phi} \mathbf{f}(\mathbf{g}_{\phi}(\mathbf{x}, \epsilon^*)),\end{aligned}$$

where  $\epsilon^* \sim p(\epsilon)$ .

## Variational assumption

$$p(\epsilon) = \mathcal{N}(0, \mathbf{I}); \quad \mathbf{z} = \mathbf{g}_{\phi}(\mathbf{x}, \epsilon) = \sigma_{\phi}(\mathbf{x}) \odot \epsilon + \mu_{\phi}(\mathbf{x});$$

$$q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mu_{\phi}(\mathbf{x}), \sigma_{\phi}^2(\mathbf{x})).$$

Here  $\mu_{\phi}(\cdot)$ ,  $\sigma_{\phi}(\cdot)$  are parameterized functions (outputs of neural network).

We will say that  $q(\mathbf{z}|\mathbf{x}, \phi) = \text{NN}_e(\mathbf{x}, \phi)$  is the **encoder**.



## ELBO gradient (E-step, $\nabla_{\phi} \mathcal{L}_{\phi, \theta}(\mathbf{x})$ )

$$\nabla_{\phi} \mathcal{L}_{\phi, \theta}(\mathbf{x}) = \nabla_{\phi} \int q(\mathbf{z}|\mathbf{x}, \phi) \log p(\mathbf{x}|\mathbf{z}, \theta) d\mathbf{z} - \nabla_{\phi} KL(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z}))$$

### Reconstruction term

$$\begin{aligned} \nabla_{\phi} \int q(\mathbf{z}|\mathbf{x}, \phi) \log p(\mathbf{x}|\mathbf{z}, \theta) d\mathbf{z} &= \int p(\epsilon) \nabla_{\phi} \log p(\mathbf{x}|\mathbf{g}_{\phi}(\mathbf{x}, \epsilon), \theta) d\epsilon \approx \\ &\approx \nabla_{\phi} \log p(\mathbf{x}|\sigma_{\phi}(\mathbf{x}) \odot \epsilon^* + \mu_{\phi}(\mathbf{x}), \theta), \quad \text{where } \epsilon^* \sim \mathcal{N}(0, \mathbf{I}) \end{aligned}$$

Let the generative distribution  $p(\mathbf{x}|\mathbf{z}, \theta)$  be the neural network.

We will say that  $p(\mathbf{x}|\mathbf{z}, \theta) = \text{NN}_d(\mathbf{z}, \theta)$  is the **decoder**.

### KL term

$p(\mathbf{z})$  is the prior distribution on the latent variables  $\mathbf{z}$ . Let assume  $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$ .

$$\nabla_{\phi} KL(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z})) = \nabla_{\phi} KL(\mathcal{N}(\mu_{\phi}(\mathbf{x}), \sigma_{\phi}^2(\mathbf{x})) || \mathcal{N}(0, \mathbf{I}))$$

This expression has analytical formula.

# Outline

## 1. EM-algorithm

Amortized inference

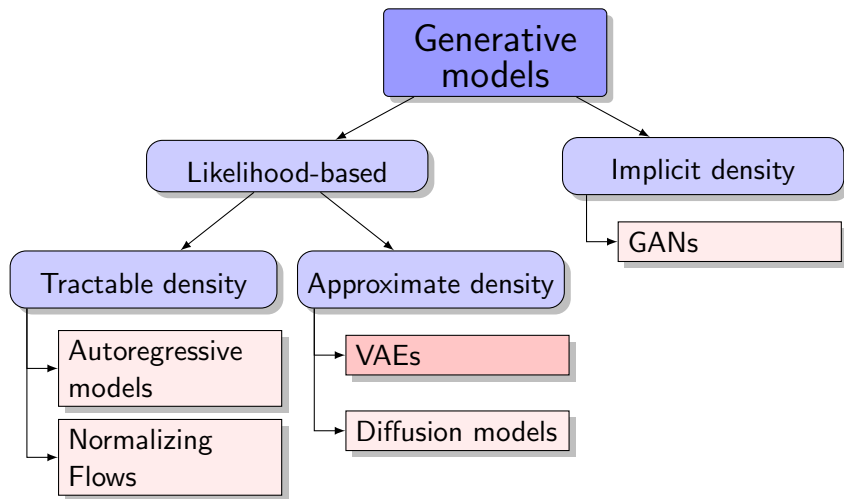
ELBO gradients, reparametrization trick

## 2. Variational autoencoder (VAE)

## 3. Normalizing flows as VAE model

## 4. Discrete VAE latent representations

# Generative models zoo



# Variational autoencoder (VAE)

## Training (EM-algorithm)

- ▶ pick random sample  $\mathbf{x}_i, i \sim \text{Uniform}\{1, n\}$  (or batch).
- ▶ compute the objective (using reparametrization trick):

$$\epsilon^* \sim r(\epsilon); \quad \mathbf{z}^* = \mathbf{g}_\phi(\mathbf{x}, \epsilon^*);$$

$$\mathcal{L}_{\phi, \theta}(\mathbf{x}) \approx \log p(\mathbf{x}|\mathbf{z}^*, \theta) - KL(q(\mathbf{z}^*|\mathbf{x}, \phi) || p(\mathbf{z}^*)).$$

- ▶ make gradient step using stochastic gradients w.r.t.  $\phi$  and  $\theta$  via autograd

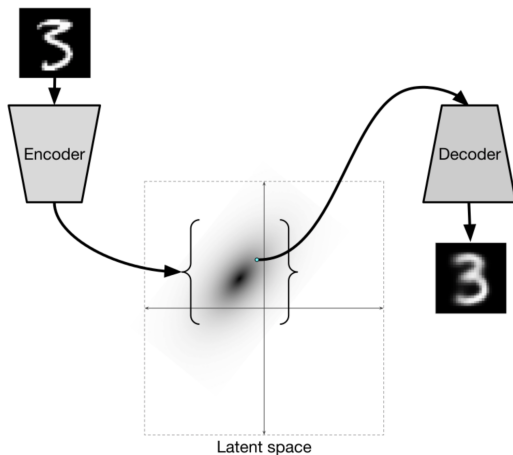
## Inference

- ▶ sample  $\mathbf{z}^*$  from the prior distribution  $p(\mathbf{z}) (\mathcal{N}(0, \mathbf{I}))$ ;
- ▶ sample from the decoder  $p(\mathbf{x}|\mathbf{z}^*, \theta)$ .

**Note:** you do not need the encoder  $q(\mathbf{z}|\mathbf{x}, \phi)$  during the generation.

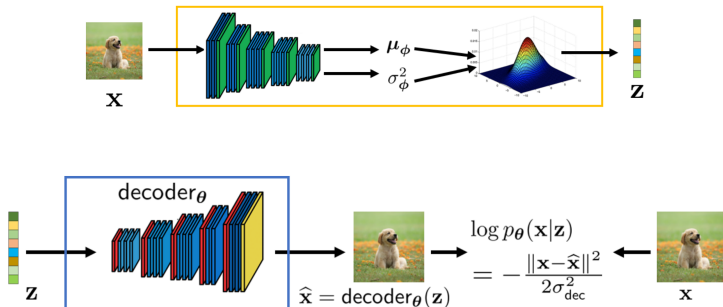
# Variational Autoencoder

$$\mathcal{L}_{\phi, \theta}(\mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \left[ \log p(\mathbf{x}|\mathbf{z}, \theta) - \log \frac{q(\mathbf{z}|\mathbf{x}, \phi)}{p(\mathbf{z})} \right] \rightarrow \max_{\phi, \theta}.$$



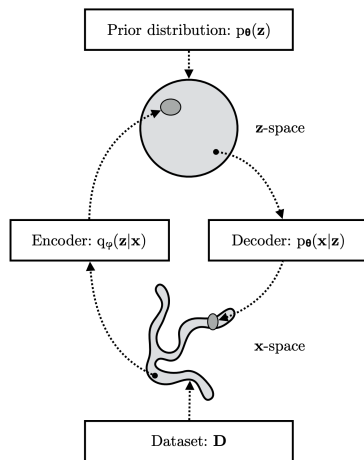
# Variational autoencoder (VAE)

- ▶ Encoder  $q(\mathbf{z}|\mathbf{x}, \phi) = \text{NN}_e(\mathbf{x}, \phi)$  outputs  $\mu_\phi(\mathbf{x})$  and  $\sigma_\phi(\mathbf{x})$ .
- ▶ Decoder  $p(\mathbf{x}|\mathbf{z}, \theta) = \text{NN}_d(\mathbf{z}, \theta)$  outputs parameters of the sample distribution.



# Variational autoencoder (VAE)

- ▶ VAE learns stochastic mapping between  $\mathbf{x}$ -space, from complicated distribution  $\pi(\mathbf{x})$ , and a latent  $\mathbf{z}$ -space, with simple distribution.
- ▶ The generative model learns a joint distribution  $p(\mathbf{x}, \mathbf{z}|\theta) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z}, \theta)$ , with a prior distribution  $p(\mathbf{z})$ , and a stochastic decoder  $p(\mathbf{x}|\mathbf{z}, \theta)$ .
- ▶ The stochastic encoder  $q(\mathbf{z}|\mathbf{x}, \phi)$  (inference model), approximates the true but intractable posterior  $p(\mathbf{z}|\mathbf{x}, \theta)$  of the generative model.



# Outline

## 1. EM-algorithm

Amortized inference

ELBO gradients, reparametrization trick

## 2. Variational autoencoder (VAE)

## 3. Normalizing flows as VAE model

## 4. Discrete VAE latent representations



# VAE vs Normalizing flows

|            | VAE                                                              | NF                                                                                                                                                           |
|------------|------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Objective  | ELBO $\mathcal{L}$                                               | Forward KL/MLE                                                                                                                                               |
| Encoder    | stochastic<br>$\mathbf{z} \sim q(\mathbf{z} \mathbf{x}, \phi)$   | deterministic<br>$\mathbf{z} = \mathbf{f}_{\theta}(\mathbf{x})$<br>$q(\mathbf{z} \mathbf{x}, \theta) = \delta(\mathbf{z} - \mathbf{f}_{\theta}(\mathbf{x}))$ |
| Decoder    | stochastic<br>$\mathbf{x} \sim p(\mathbf{x} \mathbf{z}, \theta)$ | deterministic<br>$\mathbf{x} = \mathbf{g}_{\theta}(\mathbf{z})$<br>$p(\mathbf{x} \mathbf{z}, \theta) = \delta(\mathbf{x} - \mathbf{g}_{\theta}(\mathbf{z}))$ |
| Parameters | $\phi, \theta$                                                   | $\theta \equiv \phi$                                                                                                                                         |

## Theorem

MLE for normalizing flow is equivalent to maximization of ELBO for VAE model with deterministic encoder and decoder:

$$p(\mathbf{x}|\mathbf{z}, \theta) = \delta(\mathbf{x} - \mathbf{f}_{\theta}^{-1}(\mathbf{z})) = \delta(\mathbf{x} - \mathbf{g}_{\theta}(\mathbf{z}));$$

$$q(\mathbf{z}|\mathbf{x}, \theta) = \delta(\mathbf{z} - \mathbf{f}_{\theta}(\mathbf{x})).$$

# Outline

## 1. EM-algorithm

Amortized inference

ELBO gradients, reparametrization trick

## 2. Variational autoencoder (VAE)

## 3. Normalizing flows as VAE model

## 4. Discrete VAE latent representations

# Discrete VAE latents

## Motivation

- ▶ Previous VAE models had **continuous** latent variables  $\mathbf{z}$ .
- ▶ **Discrete** representations  $\mathbf{z}$  are potentially a more natural fit for many of the modalities.
- ▶ Powerful autoregressive models (like PixelCNN) have been developed for modelling distributions over discrete variables.
- ▶ All cool transformer-like models work with discrete tokens.

## ELBO

$$\mathcal{L}_{\phi, \theta}(\mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \log p(\mathbf{x}|\mathbf{z}, \theta) - KL(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z})) \rightarrow \max_{\phi, \theta}.$$

- ▶ Reparametrization trick to get unbiased gradients.
- ▶ Normal assumptions for  $q(\mathbf{z}|\mathbf{x}, \phi)$  and  $p(\mathbf{z})$  to compute KL analytically.

# Discrete VAE latents

## Assumptions

- ▶ Let  $c \sim \text{Categorical}(\boldsymbol{\pi})$ , where

$$\boldsymbol{\pi} = (\pi_1, \dots, \pi_K), \quad \pi_k = P(c = k), \quad \sum_{k=1}^K \pi_k = 1.$$

- ▶ Let VAE model has discrete latent representation  $c$  with prior  $p(c) = \text{Uniform}\{1, \dots, K\}$ .

## ELBO

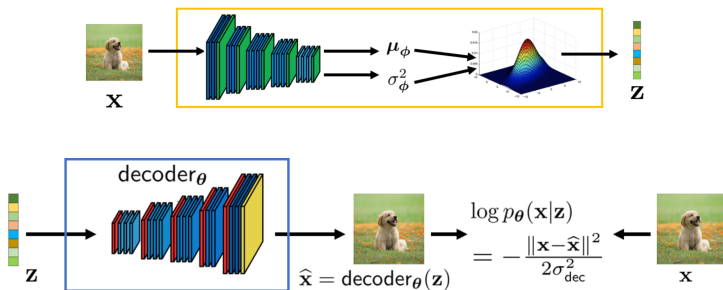
$$\mathcal{L}_{\phi, \theta}(\mathbf{x}) = \mathbb{E}_{q(c|\mathbf{x}, \phi)} \log p(\mathbf{x}|c, \theta) - \textcolor{brown}{KL(q(c|\mathbf{x}, \phi) || p(c))} \rightarrow \max_{\phi, \theta}.$$

$$\begin{aligned} \textcolor{brown}{KL(q(c|\mathbf{x}, \phi) || p(c))} &= \sum_{k=1}^K q(k|\mathbf{x}, \phi) \log \frac{q(k|\mathbf{x}, \phi)}{p(k)} = \\ &= \sum_{k=1}^K q(k|\mathbf{x}, \phi) \log q(k|\mathbf{x}, \phi) - \sum_{k=1}^K q(k|\mathbf{x}, \phi) \log p(k) = \\ &= -H(q(c|\mathbf{x}, \phi)) + \log K. \end{aligned}$$

# Discrete VAE latents

$$\mathcal{L}_{\phi, \theta}(\mathbf{x}) = \mathbb{E}_{q(c|\mathbf{x}, \phi)} \log p(\mathbf{x}|c, \theta) + H(q(c|\mathbf{x}, \phi)) - \log K \rightarrow \max_{\phi, \theta}.$$

- ▶ Our encoder should output discrete distribution  $q(c|\mathbf{x}, \phi)$ .
- ▶ We need the analogue of the reparametrization trick for the discrete distribution  $q(c|\mathbf{x}, \phi)$ .
- ▶ Our decoder  $p(\mathbf{x}|c, \theta)$  should input discrete random variable  $c$ .



# Summary

- ▶ Amortized variational inference allows to efficiently compute the stochastic gradients for ELBO using Monte-Carlo estimation.
- ▶ The reparametrization trick gets unbiased gradients w.r.t to the variational posterior distribution  $q(\mathbf{z}|\mathbf{x}, \phi)$ .
- ▶ The VAE model is an LVM with two neural network: stochastic encoder  $q(\mathbf{z}|\mathbf{x}, \phi)$  and stochastic decoder  $p(\mathbf{x}|\mathbf{z}, \theta)$ .
- ▶ NF models could be treated as VAE model with deterministic encoder and decoder.
- ▶ Discrete VAE representations is a natural form of latent variables.