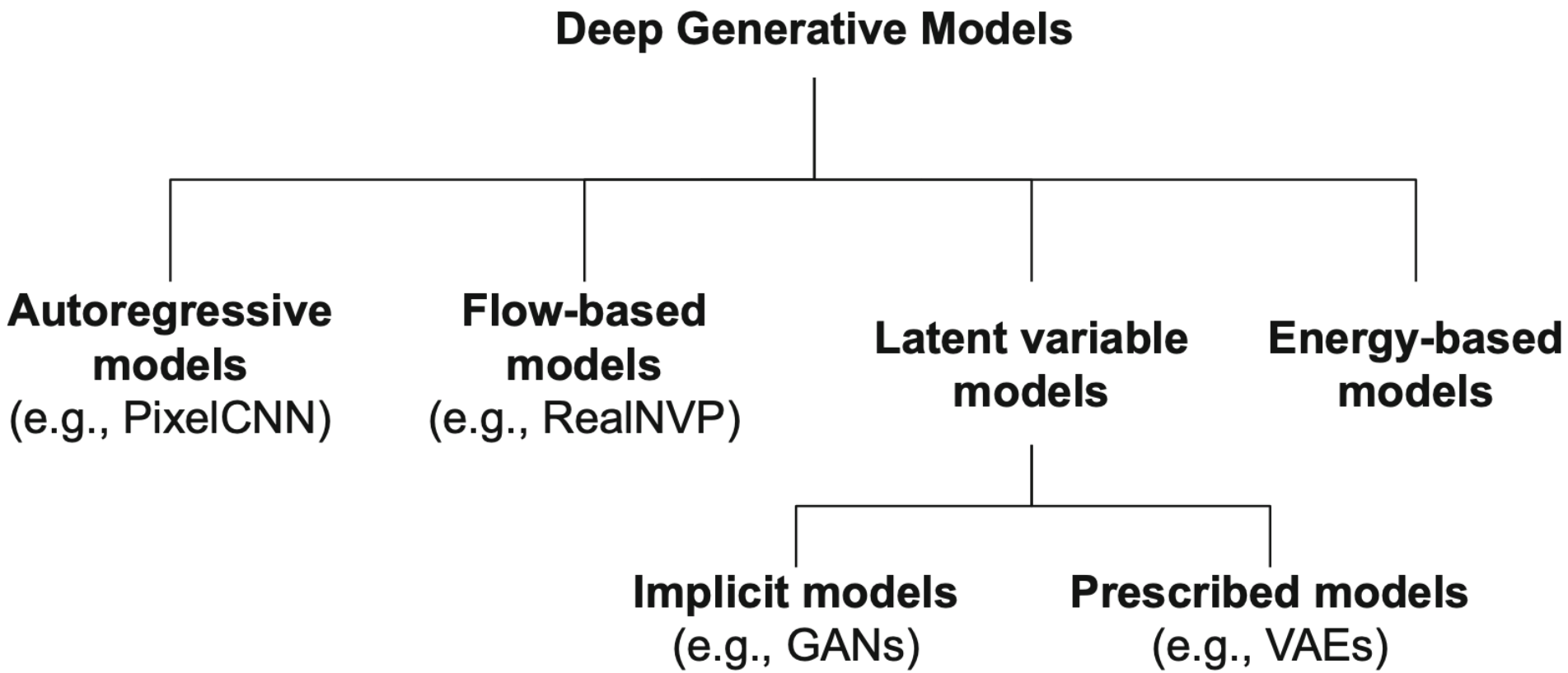
# Deep Generative Models: Variational Auto Encoders

Fall Semester 2024

René Vidal
Director of the Center for Innovation in Data Engineering and Science (IDEAS)
Rachleff University Professor, University of Pennsylvania
Amazon Scholar & Chief Scientist at NORCE

Penn
University of Pennsylvania

# Taxonomy of Generative Models



# The story up till now

- **Step 1**: We set out with our original goal of learning a model $p_\theta$ that gives maximum likelihood to our datapoints $x_i$

- **Step 2**: We introduced latent variables $z$ such that $z \sim p(z)$ and $x \mid z \sim p(x \mid z)$, which gave us the marginalization $p(x) = \int p(x \mid z)\, p(z)\, dz$
  - Step 2a: When we supposed $p(z)$ was Gaussian and $p(x \mid z) = N(Wz + b, \sigma^2 I)$, we could solve for $(W, b, \sigma^2)$ in closed form! This gave us PPCA

- **Step 3**: We set up variational inference because sadly, not everything in life is Gaussian and linear. This gave us a new objective

$$\max_\theta \log p_\theta(x_i) = \max_\theta \max_{q(\cdot|x_i),\forall i} \sum_{i=1}^{N} \int q(z|x_i) \log \frac{p_\theta(x_i, z)}{q\ (z|x_i)} dz$$

  - Step 3a: If $q(z|x)$ is easy to evaluate, we can alternate between optimizing w.r.t. $\theta$ with $q(z|x)$ fixed and vice versa, leading to the Expectation Maximization algorithm

# The story continues with Variational Autoencoders

- Before introducing VAEs formally, let us decompose ELBO further

$$\max_{\theta} \log p_\theta(x_i) = \max_{\theta} \max_{q} \sum_{i=1}^{N} \int q(z|x_i) \log \frac{p_\theta(x_i, z)}{q(z|x_i)} dz$$

$$\geq \max_{\theta,\psi} \sum_{i} E_{z\sim q_\psi(z|x_i)} \log \frac{p_\theta(x_i, z)}{q_\psi(z|x_i)}$$

Evidence Lower Bound (ELBO)

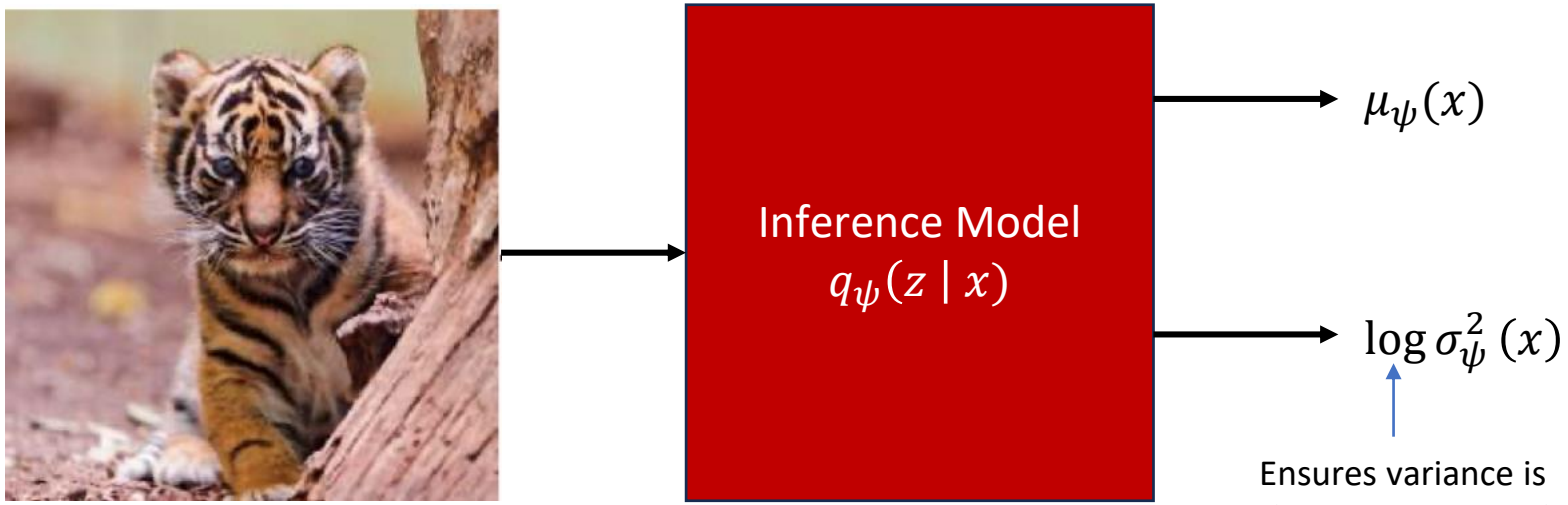
$$= \max_{\theta,\psi} \sum_{i} E_{z\sim q_\psi(z|x_i)} \log p_\theta(x_i|z) \frac{p(z)}{q_\psi(z|x_i)}$$

$$= \max_{\theta,\psi} \sum_{i} E_{z\sim q_\psi(z|x_i)} \log p_\theta(x_i|z) + E_{z\sim q_\psi(z|x_i)} \log \frac{p(z)}{q_\psi(z|x_i)}$$

$$-D_{KL}(q_\psi(z|x)||p(z))$$

# Variational AutoEncoders (VAEs): Setup

- We have three models we need to define for VAE model
  1. $q_\psi(z \mid x_i)$ (Inference model): We will define as $q_\psi(z \mid x_i) = N(z; \mu_\psi(x_i), \sigma^2_\psi(x_i)I)$ i.e. a normal distribution with learned mean and covariance



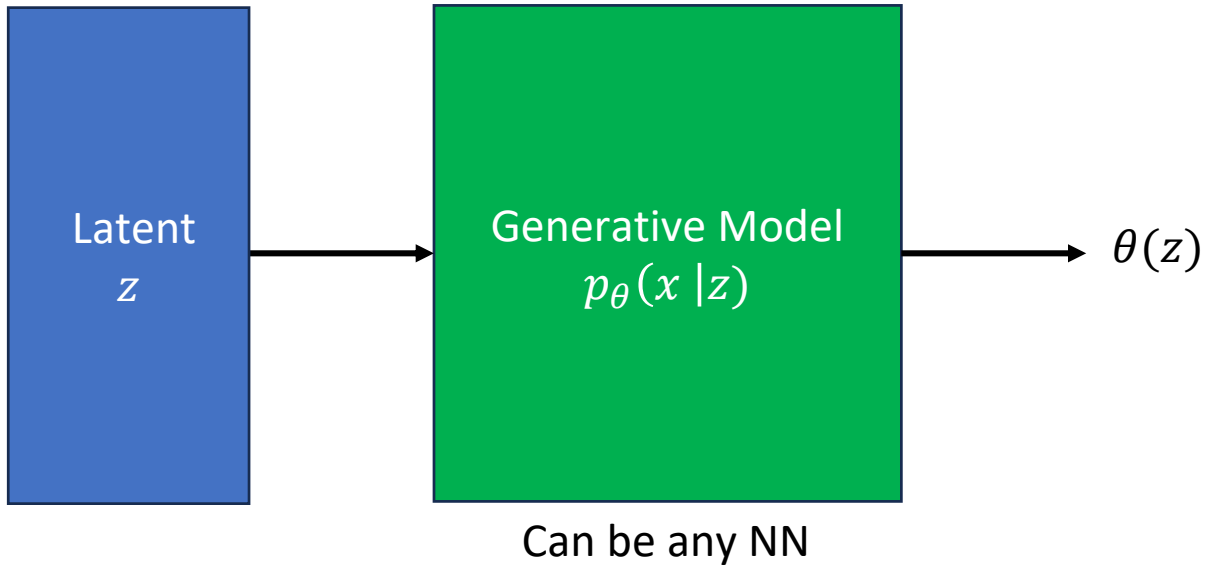

  2. $p(z)$ (Prior): We will define prior for latent variables as $p(z) = N(0, I)$

# Variational AutoEncoders (VAEs): Setup

- We have three models we need to define for VAE model

  3. $\mathrm{p}_\theta(\mathrm{x} \mid \mathrm{z})$ (Generative model): We will define as $p_\theta(x \mid z) = N(z; \theta(z), \eta^2 I)$ i.e. a normal distribution with learned mean and a constant user-defined variance

     - Note this can be defined in many different ways, yielding different models (such as a categorical distribution over 255 values of each pixel)



# Variational Autoencoders: Training

Datapoint $x_i$

Inference Model
$q_\psi(z \mid x_i)$

$\mu$

$\log \sigma^2$

Sample
$z \sim N(z \mid \mu_\theta(x_i), \sigma_\theta^2(x_i) I)$

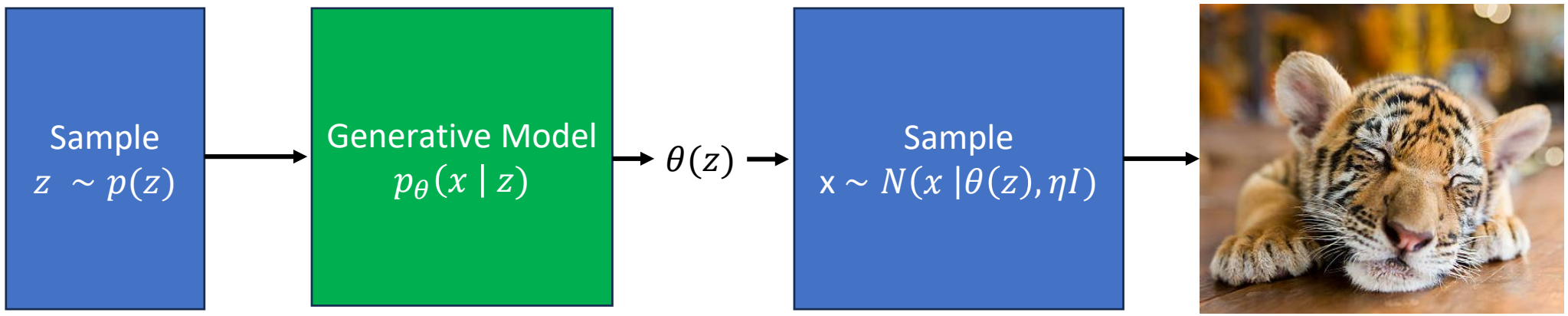
Generative Model
$p_\theta(x \mid z)$

$\theta(z)$

ELBO Objective
$\mathrm{E}_{z \sim q_\psi(z \mid x)}[\log p_\theta(x \mid z) - KL(q_\psi(z \mid x) || p(z)]$

# Variational Autoencoders after Training

- Suppose we have learned VAE using the ELBO loss (details to follow).
- Then, as a generative model, we just sample $z \sim p(z)$ and use the fixed generative model $p_\theta(x \mid z)$



# Computing the ELBO Loss

$$L_{\theta,\psi}(x) := E_{z\sim q_\psi(z\,|x)}\,\log p_\theta(x\,|z) + E_{z\sim q_\psi(z\,|\,x)}\log\frac{p(z)}{q_\psi(z\,|x)}$$

- Term 1 (Reconstruction Error): Because $p_\theta(x\,|z) = N(x\,|\theta(z), \eta I)$, we have $\log p_\theta(x\,|z) = -\frac{1}{2\eta}\left|\left|x - G_\theta(z)\right|\right|_2^2 + constant$
  - We approximate the expectation over $z \sim q_\psi(z\,|x)$ by an average over $q_\psi(z\,|x_i)$ for a dataset of $x_i$'s

- Term 2 (Regularization to Prior): Because $E_{z\sim q_\psi(z|x)}\log\frac{p(z)}{q_\psi(z|x)} = -D_{KL}(q_\psi(z|x)||p(z))$ and $q_\psi(z\,|x) = N(z\,|\mu_\psi(x), \sigma_\psi^2(x)I), p(z) = N(0, I)$ , the second term is a KL divergence between two Gaussians
  - Thankfully, this has a closed form solution for two $d$-dimensional Gaussians

$$KL(N(\mu_1, \sigma_1^2 I)||N(\mu_2, \sigma_2^2 I)) = \log\left(\frac{\sigma_2}{\sigma_1}\right) - \frac{d}{2} + \frac{d\sigma_1^2 + \left|\left|\mu_1 - \mu_2\right|\right|_2^2}{2\sigma_2^2}$$

  - For us, this becomes

$$KL\left(q_\psi(z\,|x)||p(z)\right) = -\log\left(\sigma_\psi(x)\right) + \frac{d\sigma_\psi^2(x) + \left|\left|\mu_\psi(x)\right|\right|_2^2}{2} + constant$$

# Maximizing ELBO: How to optimize?

- So now we want to solve the following optimization problem:

$$\max_{\theta,\psi} \sum_i L_{\theta,\psi}(x_i) = \sum_i E_{z\sim q_\psi(z\,|x_i)} \log \frac{p_\theta(x_i, z)}{q_\psi(z\,|x_i)}$$

- Simple idea: Just alternate gradient ascent wrt $\theta, \psi$ on objective function

$$\theta_{k+1} = \theta_k + \eta \frac{\delta L}{\delta \theta}(\theta_k, \psi_k)$$

$$\psi_{k+1} = \psi_k + \eta \frac{\delta L}{\delta \psi}(\theta_k, \psi_k)$$

# Stochastic Optimization of ELBO wrt $\theta$

- *Issue*: Computing gradient of ELBO wrt $\theta$ is intractable because there is an expectation in the gradient!
- *Solution*: Compute an unbiased estimator

$$\nabla_\theta L_{\theta,\psi}(x) = \nabla_\theta E_{z\sim q_\psi(z\,|x)}\left[\log p_\theta(x\,, z) - \log q_\psi(z\,|x)\right]$$

$$= E_{z\sim q_\psi(z\,|x)} \nabla_\theta \log p_\theta(x, z)$$

Just take sample averages to compute unbiased estimator

# Stochastic Optimization of ELBO wrt $\psi$

- Here, we cannot just switch gradient and expectation because both are wrt to $\psi$

$$\nabla_\psi L_{\theta,\psi}(x) = \nabla_\psi E_{z\sim q_\psi(z|x)}[\log p_\theta(x\,, z) - \log q_\psi(z\,|x)]$$

$$\neq E_{z\sim q_\psi(z|x)} \nabla_\psi [\log p_\theta(x, z) - \log q_\psi(z\,|x)]$$

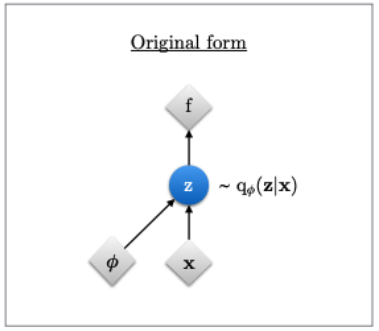
- To compute gradient, we will use that $q_\psi(z|x) = N(z; \mu_\psi(x), \sigma_\psi(x)I)$
  - We can rewrite samples $z \sim q_\psi(z\,|x)$ as $z = \mu_\psi(x) + \sigma_\psi(x)\epsilon$ for $\epsilon \sim N(0, I)$
  - This is a change of variables as we rewrite $z = g(\epsilon, \psi, x) = \mu_\psi(x) + \sigma_\psi(x)\epsilon$
  - We can then write

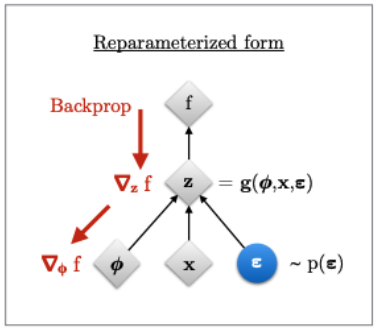
**Reparameterization trick!**

$$\nabla_\psi L_{\theta,\psi}(x) = \nabla_\psi E_{z\sim q_\psi(z\,|x)} [\log p_\theta(x, z) - \log q_\psi(z\,|x)]$$

$$= \nabla_\psi E_{\epsilon\sim N(0,I)} [\log p_\theta(x, z) - \log q_\psi(z\,|x)]$$

Just take sample averages to compute unbiased estimator



# Full Algorithm for Stochastic Optimization of ELBO

**Algorithm 1:** Stochastic optimization of the ELBO. Since noise originates from both the minibatch sampling and sampling of $p(\boldsymbol{\epsilon})$, this is a doubly stochastic optimization procedure. We also refer to this procedure as the *Auto-Encoding Variational Bayes* (AEVB) algorithm.

**Data:**

$\mathcal{D}$: Dataset

$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$: Inference model

$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$: Generative model

**Result:**

$\boldsymbol{\theta}, \boldsymbol{\phi}$: Learned parameters

$(\boldsymbol{\theta}, \boldsymbol{\phi}) \leftarrow$ Initialize parameters

**while** *SGD not converged* **do**

  $\mathcal{M} \sim \mathcal{D}$ (Random minibatch of data)

  $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$ (Random noise for every datapoint in $\mathcal{M}$)

  Compute $\tilde{\mathcal{L}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathcal{M}, \boldsymbol{\epsilon})$ and its gradients $\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \tilde{\mathcal{L}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathcal{M}, \boldsymbol{\epsilon})$

  Update $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ using SGD optimizer

**end**

# Putting it all together

- Variational Autoencoder
  - We modelled inference and generative model as deep networks
  - We interpreted ELBO as an expected reconstruction error plus a KL-regularization to prior
  - Then, we rewrote the sampling in the latent space using the reparameterization trick
  - Finally, we derived stochastic gradient estimates to optimize the ELBO and learn a VAE

# VAE's in Action

$$q_\psi(z\,|x) = N(z\,|\mu_\psi(x), \sigma^2_\psi(x))$$

$$p(z) = N(z|0, I)$$

$$p_\theta(x\,|z) = Categorical\ (x\,|\theta(z))$$

Note this is different from the model considered up till now!

- The *encoder network*:

$$\mathbf{x} \in \mathcal{X}^D \rightarrow \text{Linear}(D, 256) \rightarrow \text{LeakyReLU} \rightarrow$$

$$\text{Linear}(256, 2 \cdot M) \rightarrow \text{split} \rightarrow \mu \in \mathbb{R}^M,\ \log \sigma^2 \in \mathbb{R}^M.$$

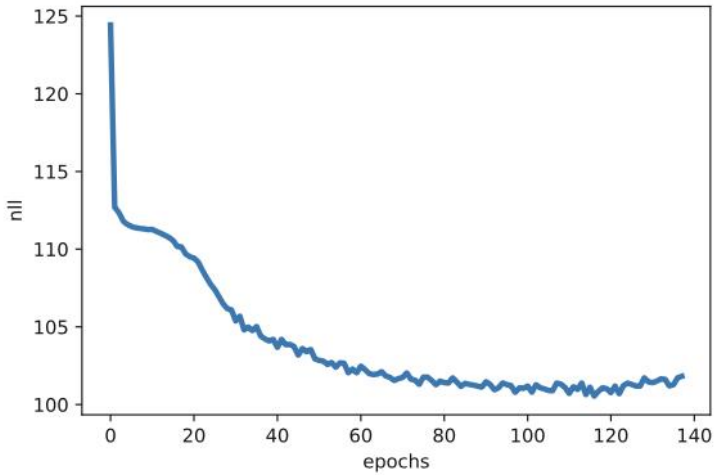
- The *decoder network*:

$$\mathbf{z} \in \mathbb{R}^M \rightarrow \text{Linear}(M, 256) \rightarrow \text{LeakyReLU} \rightarrow$$

$$\text{Linear}(256, D \cdot L) \rightarrow \text{reshape} \rightarrow \text{softmax} \rightarrow \theta \in [0, 1]^{D \times L}.$$

# VAE's for Generation of MNIST Digits

A




B C

**Fig. 4.4** An example of outcomes after the training: (**a**) Randomly selected real images. (**b**) Unconditional generations from the VAE. (**c**) The validation curve during training

# Typical Issues with VAEs

- ELBO: $\ln p(\mathbf{x}) = \underbrace{\mathbb{E}_{\mathbf{z}\sim q_\psi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x} \mid \mathbf{z})] - KL\left[q_\psi(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z})\right]}_{ELBO} + \underbrace{KL\left[q_\psi(\mathbf{z} \mid \mathbf{x}) \parallel p_\theta(\mathbf{z} \mid \mathbf{x})\right]}_{\geq 0}$
- Posterior collapse
  - If the decoder is so powerful that it treats $z$ as noise, then $\forall_{\mathbf{x}} q_\phi(\mathbf{z} \mid \mathbf{x}) = p(\mathbf{z})$
- Mismatch between prior $p(\boldsymbol{z})$ and aggregated posterior $q_\phi(\mathbf{z}) = \frac{1}{N}\sum_i q_\psi(\mathbf{z} \mid \mathbf{x}_i)$
  - Prior assigns high probability but aggregated posterior assigns low probability, or other way around.
  - Sampling from such regions provides unrealistic latent values and the decoder produces images of very low quality.
- Out-of-distribution samples

# Aggregated posterior

**Fig. 4.5** An example of the aggregated posterior. Individual points are encoded as Gaussians in the 2D latent space (magenta), and the mixture of variational posteriors (the aggregated posterior) is presented by contours

$$q_\psi(\mathbf{z}) = \frac{1}{N}\sum_i q_\psi(\mathbf{z} \mid \mathbf{x}_i)$$

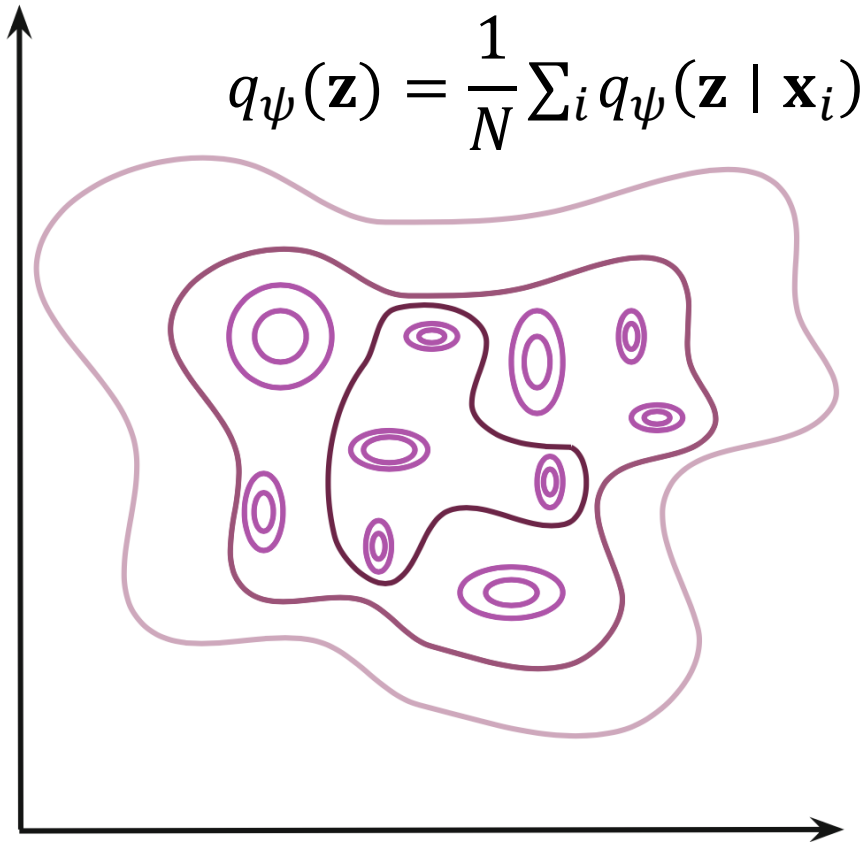
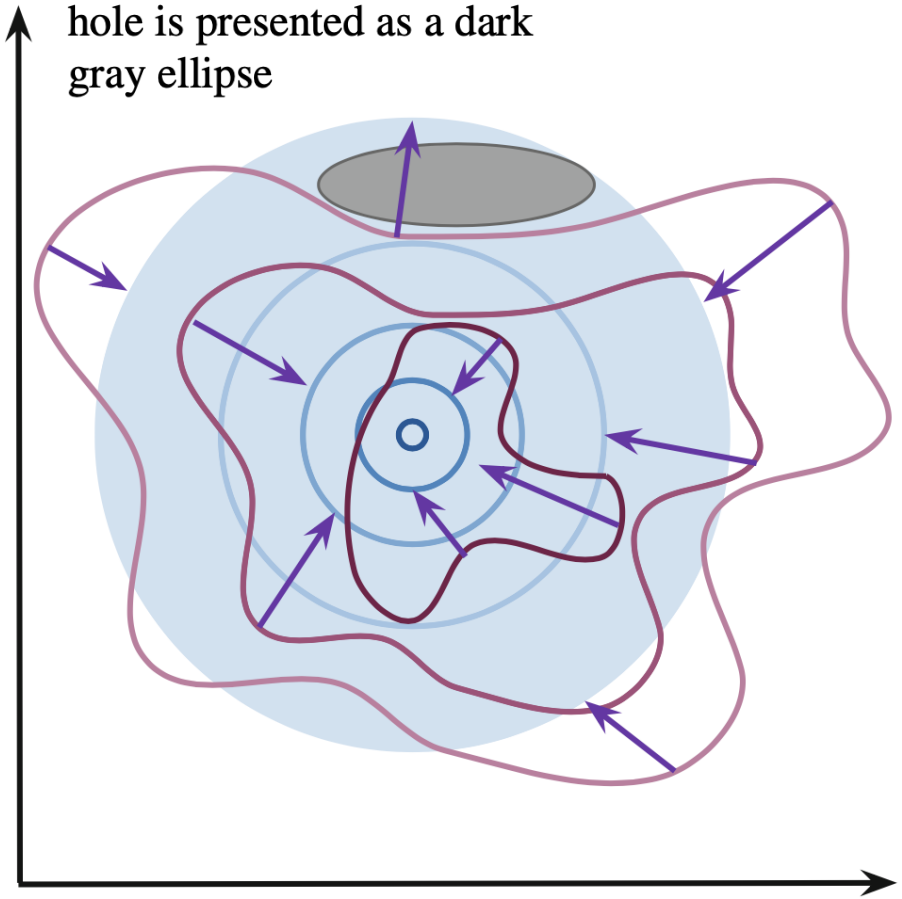
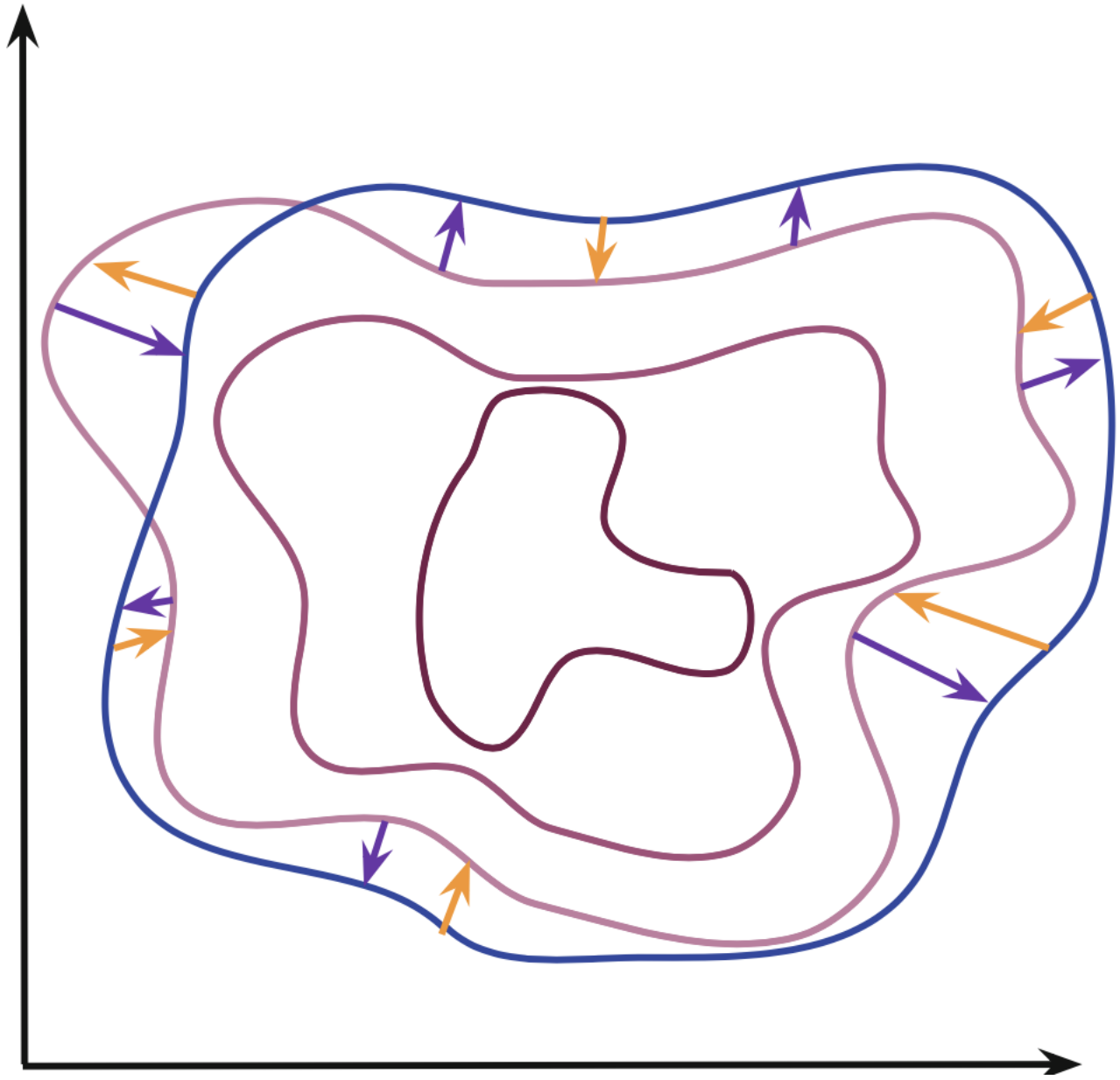


**Fig. 4.6** An example of the effect of the cross-entropy optimization with a non-learnable prior. The aggregated posterior (purple contours) tries to match the non-learnable prior (in blue). The purple arrows indicate the change of the aggregated posterior. An example of a hole is presented as a dark gray ellipse



# Learnable prior

**Fig. 4.7** An example of the effect of the cross-entropy optimization with a learnable prior. The aggregated posterior (purple contours) tries to match the learnable prior (blue contours). Notice that the aggregated posterior is modified to fit the prior (purple arrows), but also the prior is updated to cover the aggregated posterior (orange arrows)



# Improving VAEs

- The ELBO consists of two parts: first, the reconstruction error

$$RE \triangleq \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\mathbb{E}_{q_\psi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x} \mid \mathbf{z})]\right]$$

- Then the regularization term between the encoder and the prior

$$\begin{aligned}\Omega \quad &\triangleq \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\mathbb{E}_{q_\psi(\mathbf{z}|\mathbf{x})}\left[\ln p(\mathbf{z}) - \ln q_\psi(\mathbf{z} \mid \mathbf{x})\right]\right] \\ &= -KL\left[q_\psi(\mathbf{z}) \parallel p(\mathbf{z})\right] + \mathbb{H}\left[q_\psi(\mathbf{z} \mid \mathbf{x})\right]\end{aligned}$$

- For a Gaussian, the entropy is maximized when sigma -> infinity