Cornell Bowers C·IS
College of Computing and Information Science

Deep Learning

Week 7: The Variational Auto-Encoder (VAE)

Thanks to:

Varsha Kishore
Justin Lovelace
Zachary Ross
Madhav Aggarwal
Oliver Richardson

Cornell Bowers CIS

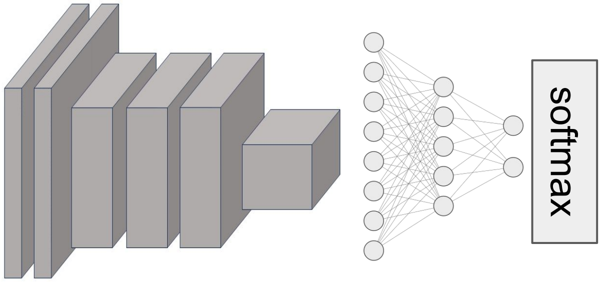
Discriminative Models

typically supervised

Goal: model $p(Y|X)$

from samples of p(X,Y)

(* so that we can predict most likely labels)



Generative Models

unsupervised

Goal: model $p(X)$

from samples of p(X)

(* so that we can generate artificial/new data)

Examples:

- GANs + variants
- Normalizing Flow Models
- Variational Autoencoders
  - Diffusion Models

Cornell Bowers C·IS

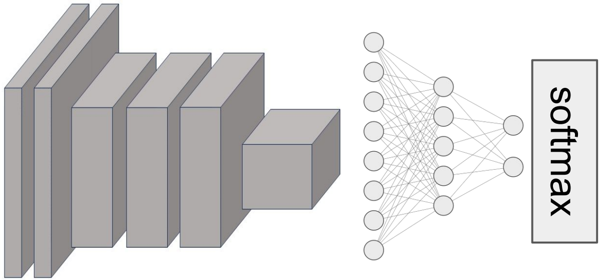
Discriminative Models

typically supervised

Goal: model $p(Y|X)$

from samples of p(X,Y)

(* so that we can predict most likely labels)



Generative Models

(Conditional generation)

Goal: model $p(X|Y)$

from samples of p(X,Y)

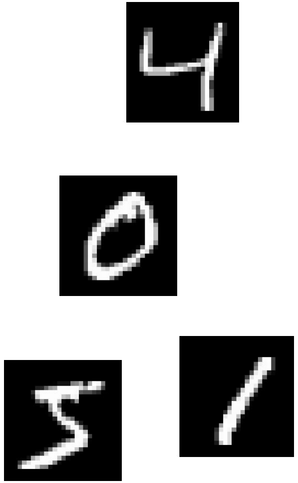
(* so that we can generate artificial/new data)

Examples:

- GANs + variants
- Normalizing Flow Models
- Variational Autoencoders
    - Diffusion Models

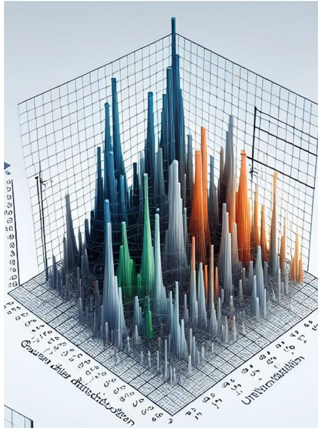
Big Picture

Data sampled from true (but elusive) P(X)



Learn approximate data distribution

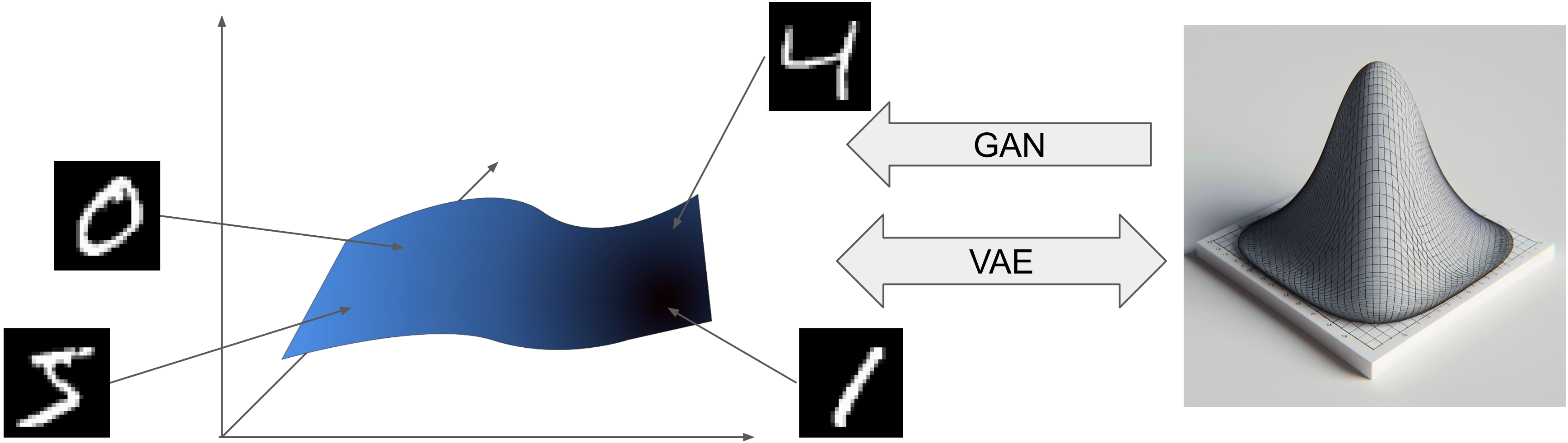
Q(X)≈P(X)



New (fake) data drawn from Q(x)

Data Manifold

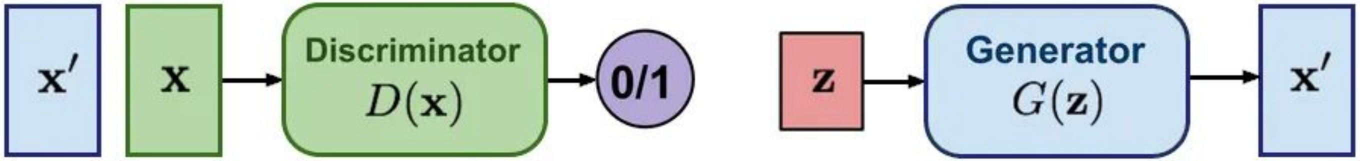
- Data distribution **P(X)** defines a manifold of valid images
- Problem: data manifold takes up **tiny** volume of ambient space
- Naive random samples (e.g. within $[0,1]^d$) are always **off manifold**
- Solution: Sample from a Gaussian, then learn mapping to and from manifold



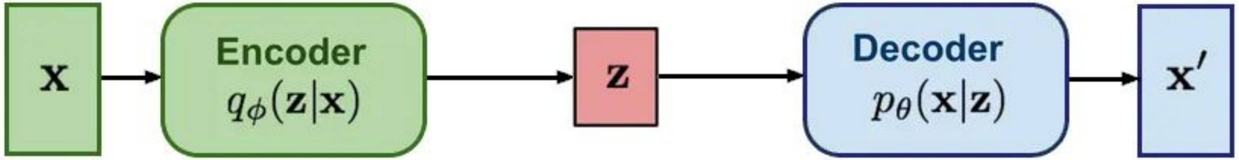
We want to sample from the elusive Image Manifold

We **can** sample from a Gaussian Distribution

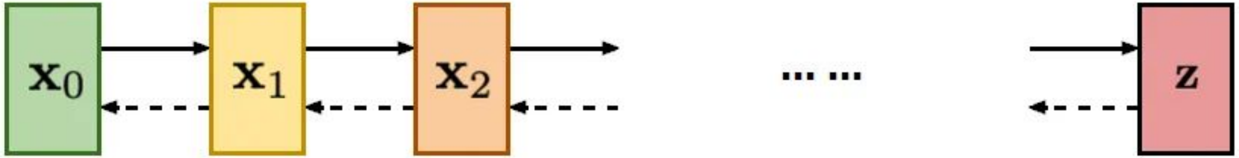
GAN: Adversarial training



VAE: maximize variational lower bound



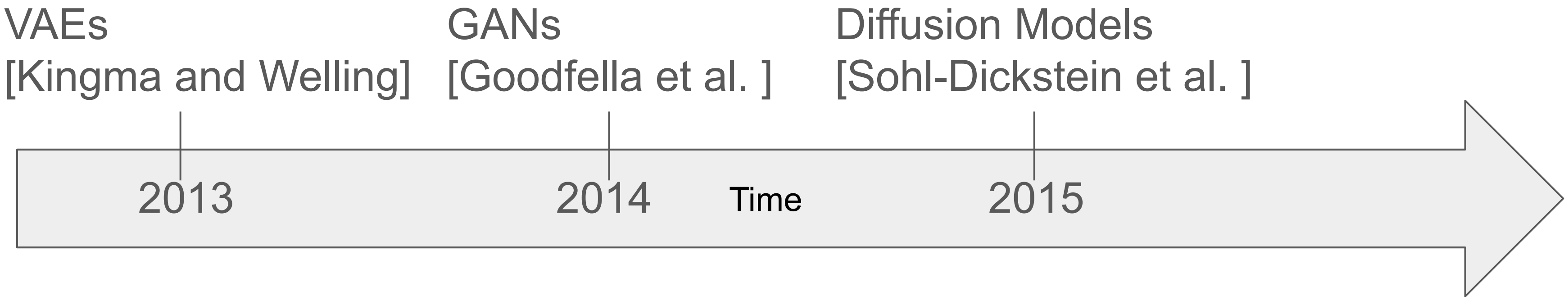
Diffusion models: Gradually add Gaussian noise and then reverse



https://miro.medium.com/v2/resize:fit:1400/format:webp/1*_5GpdejeOvt61ew4aPtT_g.png

Timeline

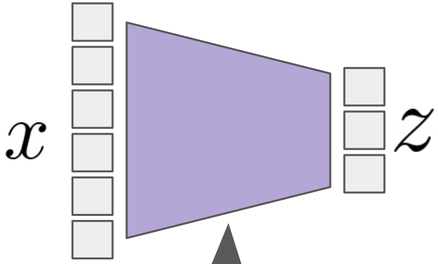
- VAEs preceded GANs.
- In fact, GANs were motivated to fix some of the problems of VAEs.
- VAEs are important to understand **Diffusion Models**



Cornell Bowers C·IS

Dimensionality Reduction

Data is typically from a **low dimensional** distribution embedded in a much **higher dimensional** ambient space.



Want to map images $x \in \mathbb{R}^D$
to low-dimensional $z \in \mathbb{R}^d$

Often for the purposes of

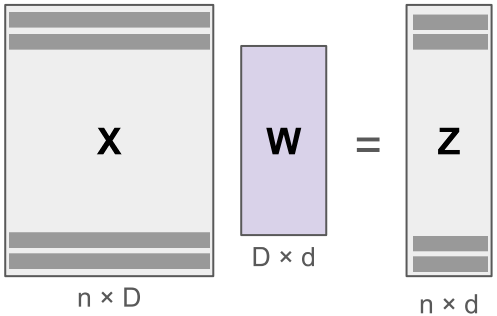
- visualization
- extracting important features
  (for downstream tasks)
- representing meaningful relationships between samples
- In this lecture sampling!

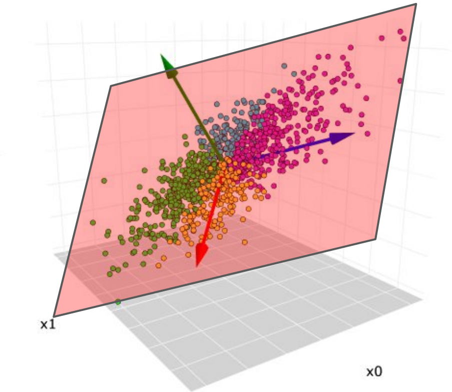
Question: what properties should this mapping have?

Cornell Bowers C·IS

Principal Component Analysis (PCA)

- **assumption: data manifold is a subspace**
- $\mathbf{z} = \mathbf{W}(\mathbf{x} - \mu)$ (a linear transformation)
- $\mathbf{x} \approx \mathbf{W}^\top \mathbf{z} + \mu$ (reconstruction)
- capture as much variance as possible



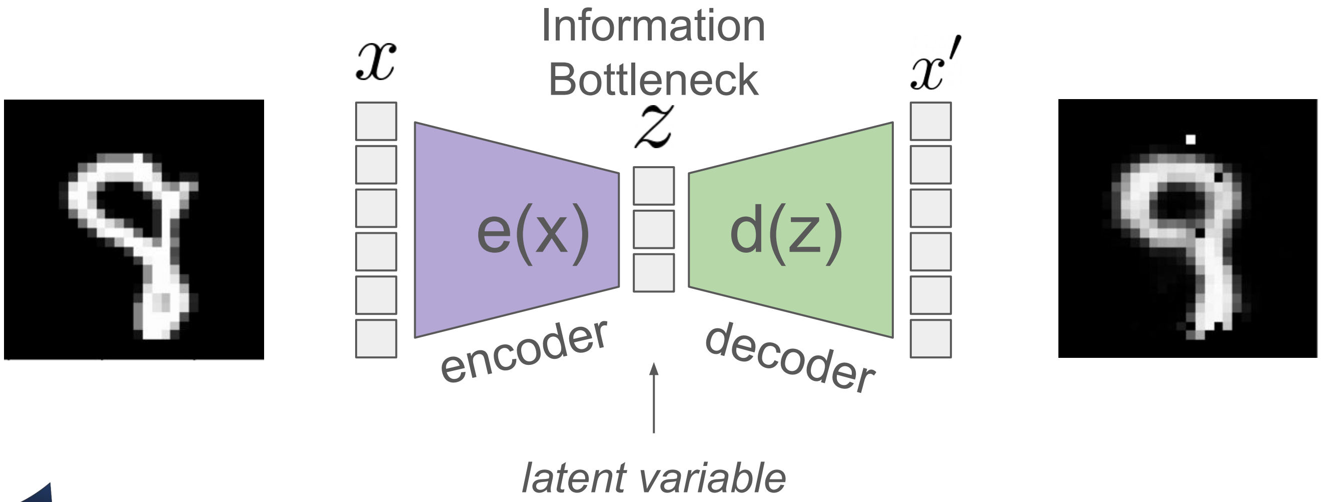


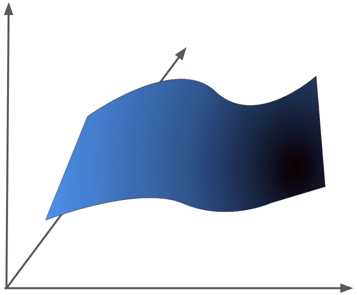
Can be computed directly with linear algebra: take leading eigenvectors of (centered) scatter matrix XX'!

https://towardsdatascience.com/principal-component-analysis-pca-explained-visually-with-zero-math-1cbf392b9e7d/

Autoencoders [Kramer, 1991]

Non-linear dimensionality reduction.



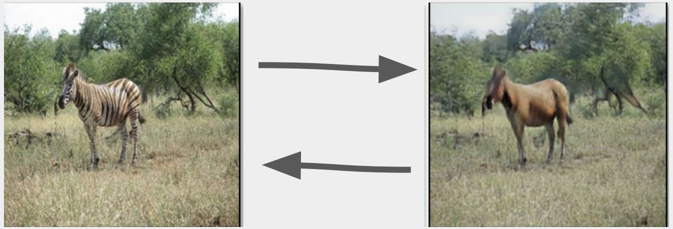


Question: What loss function should we use to learn e() and d()?
What happens if e(x) and d(z) are both linear functions?

Autoencoders [Kramer, 1991]

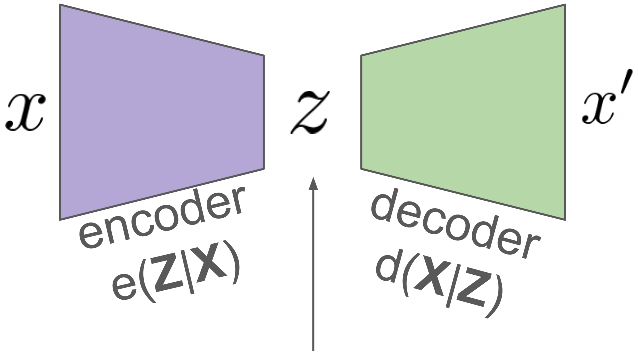
Typical loss: Squared loss, or absolute loss

Zebra Horse

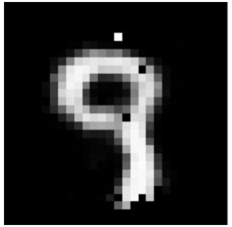


Remember cycle consistency?

Information Bottleneck



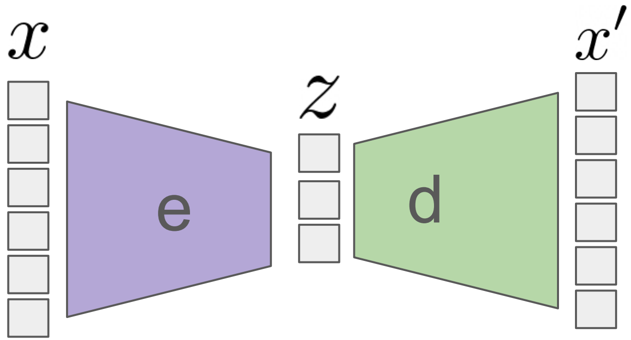
latent space



$$\min_{\phi,\theta} \sum_{\mathbf{x}\in\mathcal{D}} (\mathbf{x} - \mathbf{x}')^2$$

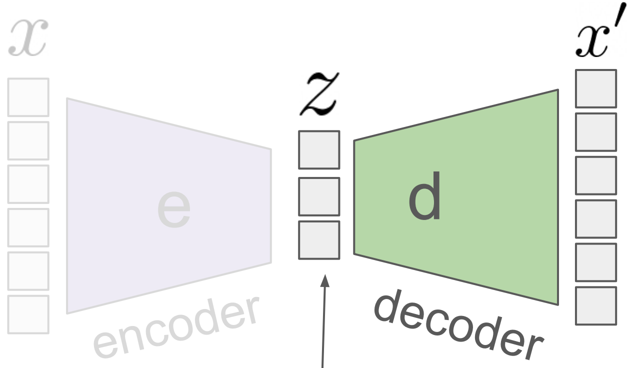
$$\mathbf{x}' = d_\theta(e_\phi(\mathbf{x}))$$

Idea: Sampling from a trained Autoencoder



- GANs train the decoder with a discriminator
- VAEs ensure quality with
  - Reconstruction loss
  - KL regularization (in a few slides)

Idea: Sampling from a trained Autoencoder



Feed in noise, sampled from some distribution P(z)

- GANs train the decoder with a discriminator
- VAEs ensure quality with
  - Reconstruction loss
  - KL regularization (in a few slides)

<u>Crucial insight:</u>
We can amend latent space so that it is easy to sample from it.

Autoencoder trained on MNIST: latent space

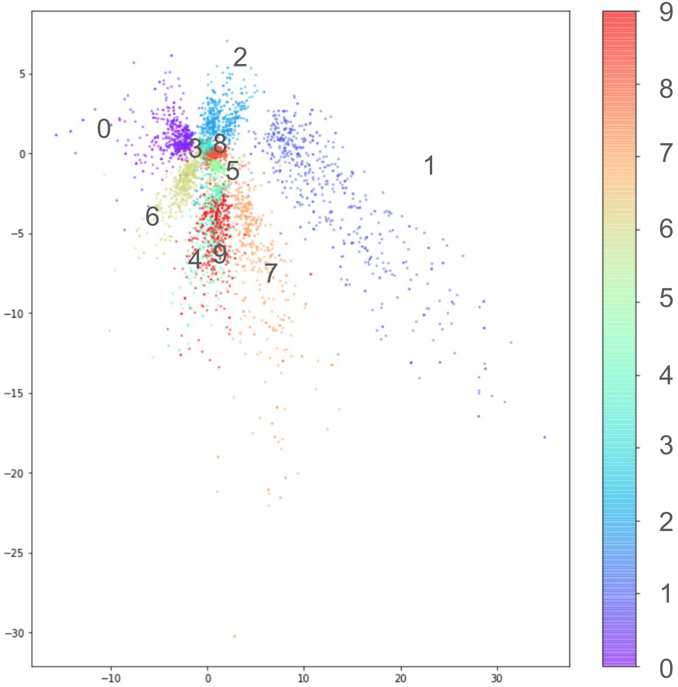
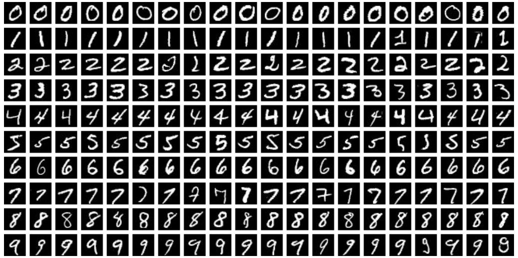


Figure 3-8. Plot of the latent space, colored by digit



Naive representation (without any special effort), not favorable:

- lots of empty space
- no symmetries between digit representations
- **Not easy to sample in latent space**

Naive sampling in latent space does not work

reconstructed sample

$$x' = d(e(x))$$

new image?

$$x' = d(\text{ noise })$$

Some Fundamentals

of probability and information

Building Blocks:

- Conditional and marginal probabilities
- Surprisal / Negative Log Likelihood
- Relative Entropy / KL Divergence

Conditional and Marginal Probabilities

$$p(X, Y) = p(Y|X)p(X)$$

$$p(X) = \int p(X, y)\, \mathrm{d}y$$

Equation (2): Quick 60s Stats Puzzle

Prove that for any random variable X,Z.

$$p(\mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{z}|\mathbf{x})}$$

Hint: Later this rule will come in handy. Remember it as “Equation (2)”.

Equivalences

surprisal

Negative log-likelihood

$$\min_\theta \log\left(\frac{1}{p_\theta(x)}\right) = \min_\theta -\log\left(p_\theta(x)\right)$$

$$= \max_\theta \log\left(p_\theta(x)\right)$$

log-likelihood

KL Divergence (a.k.a. relative entropy)

$$\boldsymbol{D}(\,p\,\|\,q\,) := \mathop{\mathbb{E}}_{x\sim p}\left[\log\frac{p(x)}{q(x)}\right]$$

reality
(e.g., dataset)

model

$$= \underbrace{\mathop{\mathbb{E}}_{x\sim p}\left[\log\frac{1}{q(x)}\right.}_{\text{Cross Entropy!}} - \boxed{\left.\log\frac{1}{p(x)}\right]}$$

Cross Entropy!

(constant; does not depend on model q)

- non-negative $\boldsymbol{D}(\,p\,\|\,q\,) \geq 0$
- zero means same $\boldsymbol{D}(\,p\,\|\,q\,) = 0 \iff p = q$
- not symmetric
- has many other, uniquely nice properties …

KL Divergence

Justin's Coin

Varsha's Coin

× 2

Question:
Is it just as easy to mistake the output of Justin's coin for that of Varsha's coin, as vice versa?

[link to visualization]

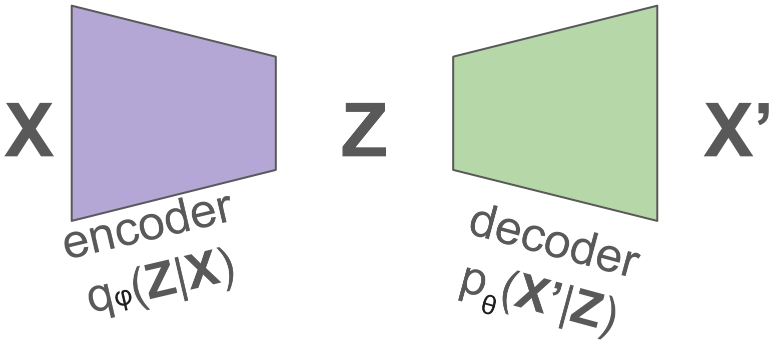
Building Blocks:

✅ Conditional and marginal probabilities
✅ Surprisal / Negative Log Likelihood
✅ Relative Entropy / KL Divergence

VAEs,
Step 1: Make AutoEncoder probabilistic

Reconstruction Loss, using surprisal

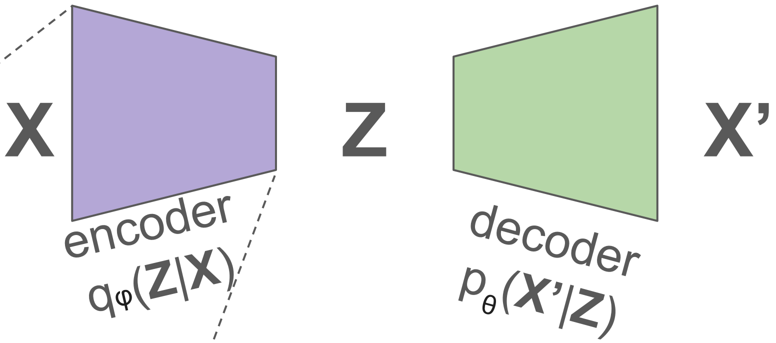
Back to our AutoEncoder, but this time we make everything **probabilistic**!

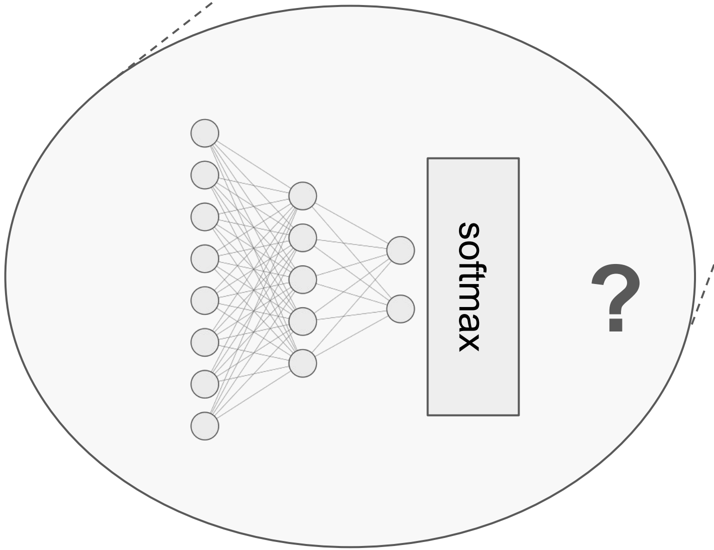


$$\max_{\phi,\theta} \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\left(p_\theta(\mathbf{x}|\mathbf{z})\right)\right]$$

How likely would it be to encode x, decode the result, and recover x?

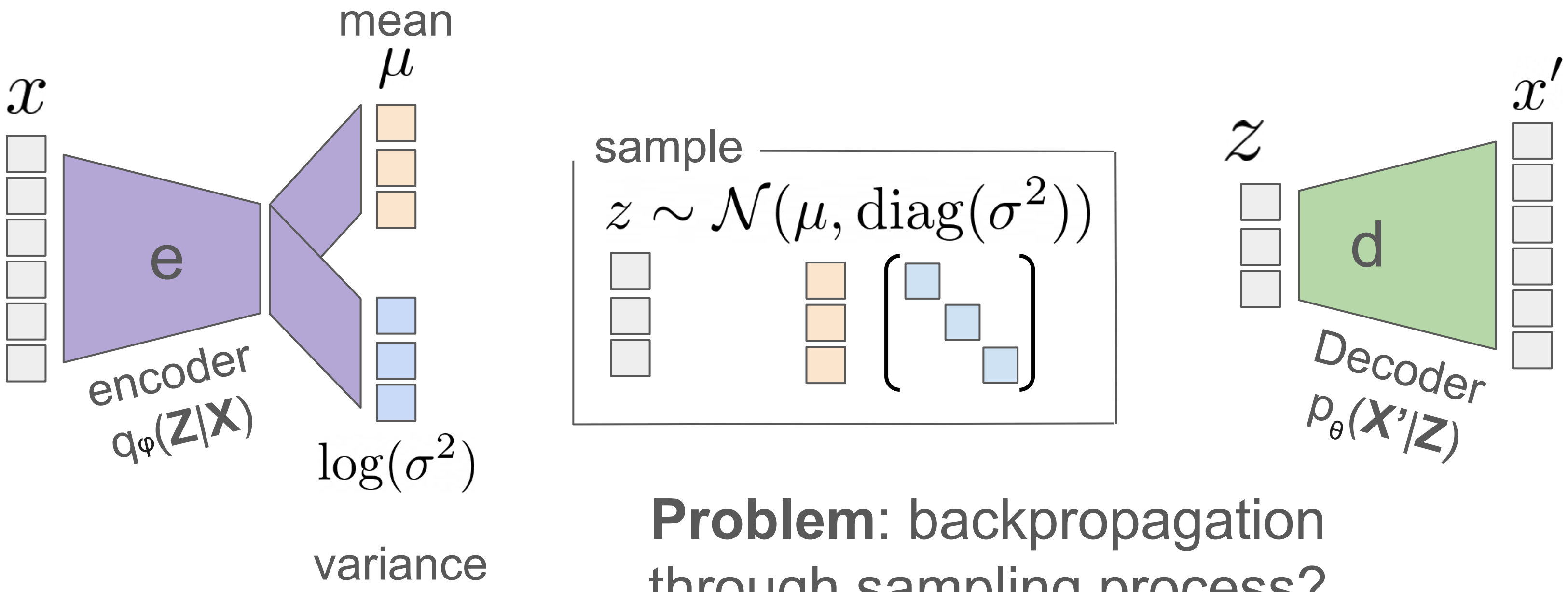
So far we have used the softmax to get probabilities…





Softmax gives us a **multinomial** distribution.
But our latent space is not discrete i.e. {1, ... , c},
but **continuous**, $\mathbb{R}^d$!
Gaussian would be better …

Probabilistic **Encoder** (Gaussian)

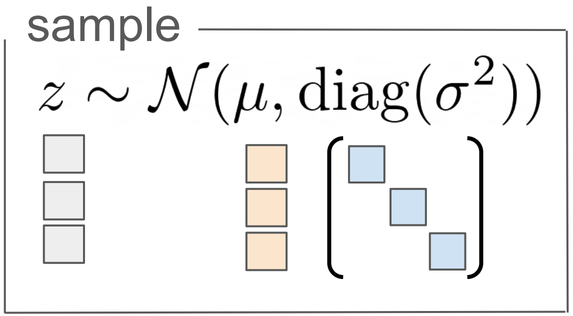


Problem: backpropagation through sampling process?

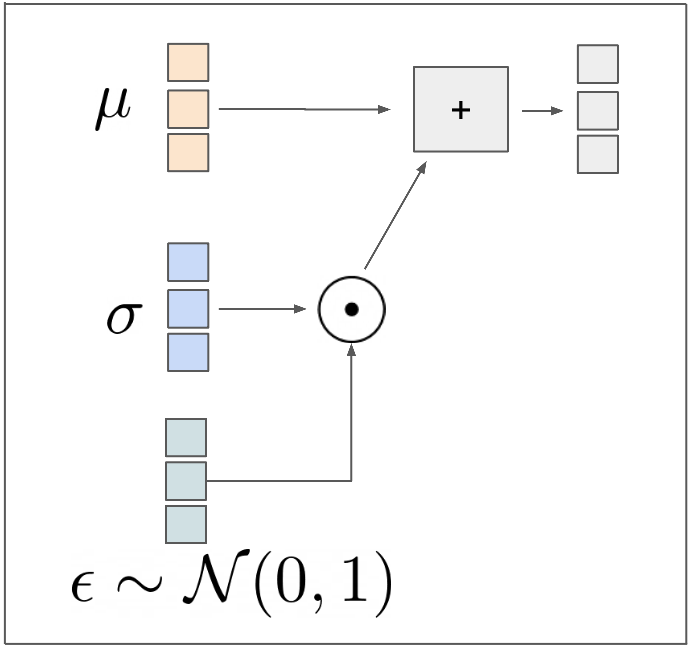
$$\max_{\phi,\theta} \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log(p_\theta(\mathbf{x}|\mathbf{z}))]$$

The Reparameterization Trick

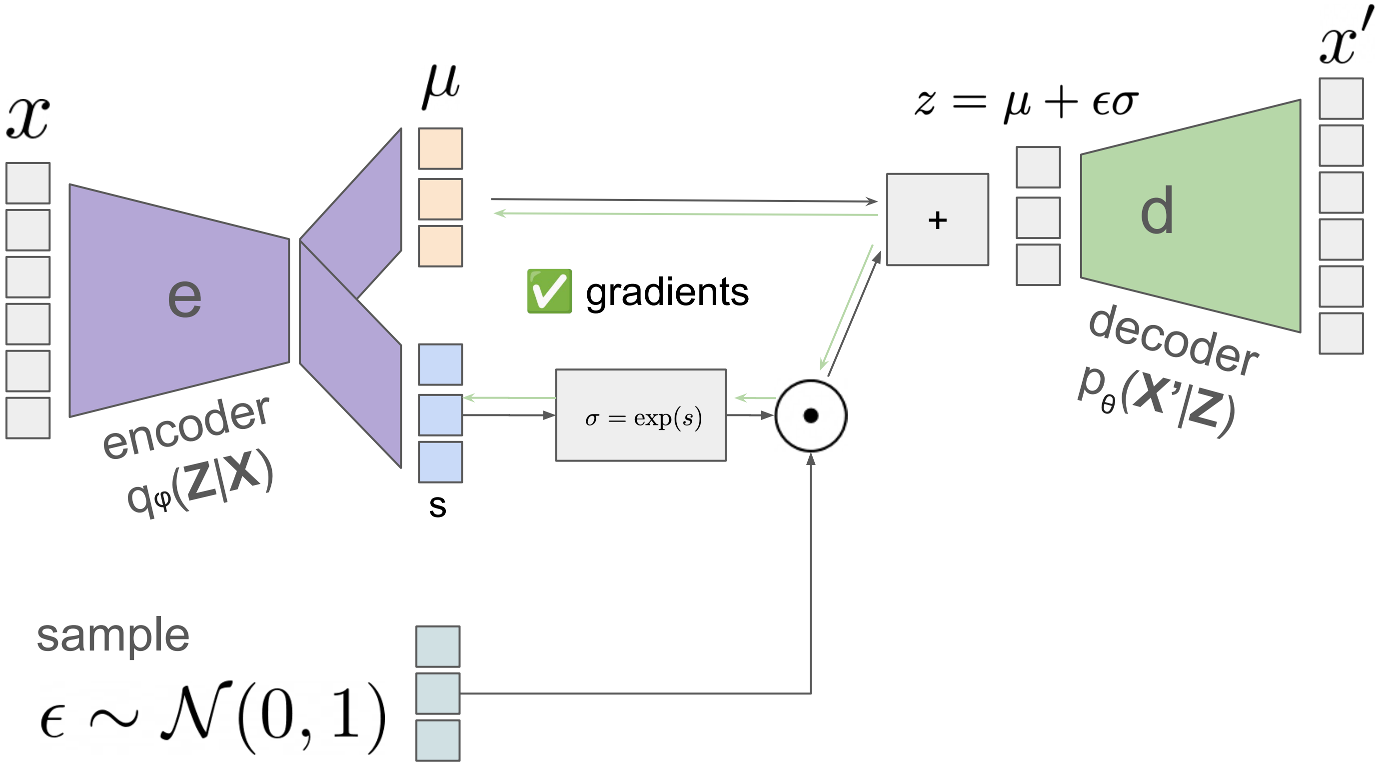
$$\mathcal{N}(\mu, \mathrm{diag}(\sigma^2)) = \mu + \sigma \odot \mathcal{N}(0, I)$$



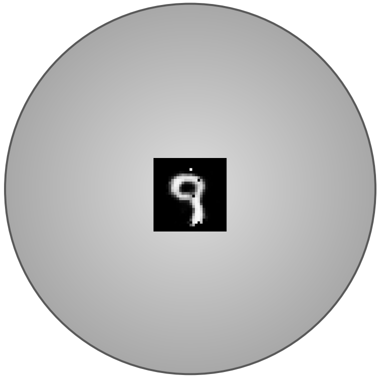
$=$

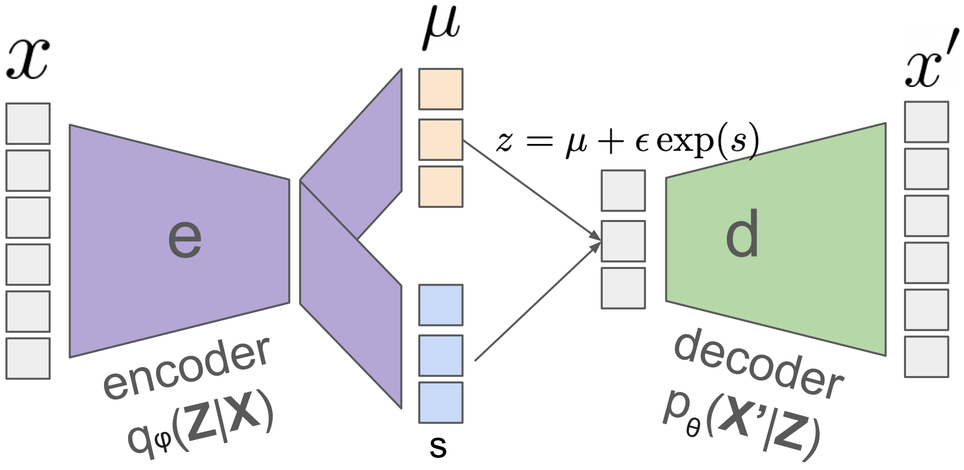


The Reparameterization Trick



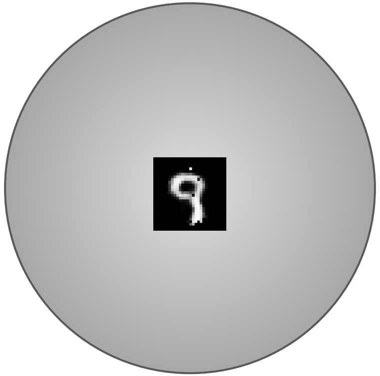
Probabilistic **decoder** (Gaussian)

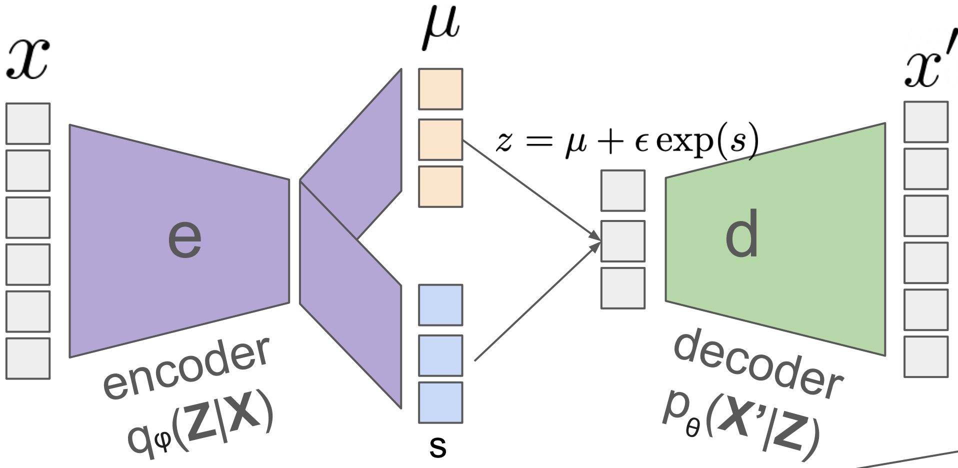




$$\mathbf{x}' \sim \mathcal{N}(d(\mathbf{z}), \tau^2)$$

Probabilistic **decoder** (Gaussian)





$$\mathbf{x}' \sim \mathcal{N}(d(\mathbf{z}), \tau^2)$$

$$P_\theta(\mathbf{x}'|\mathbf{z}) \sim \exp\left(-\frac{(\mathbf{x}' - d_\theta(\mathbf{z}))^2}{\tau^2}\right)$$

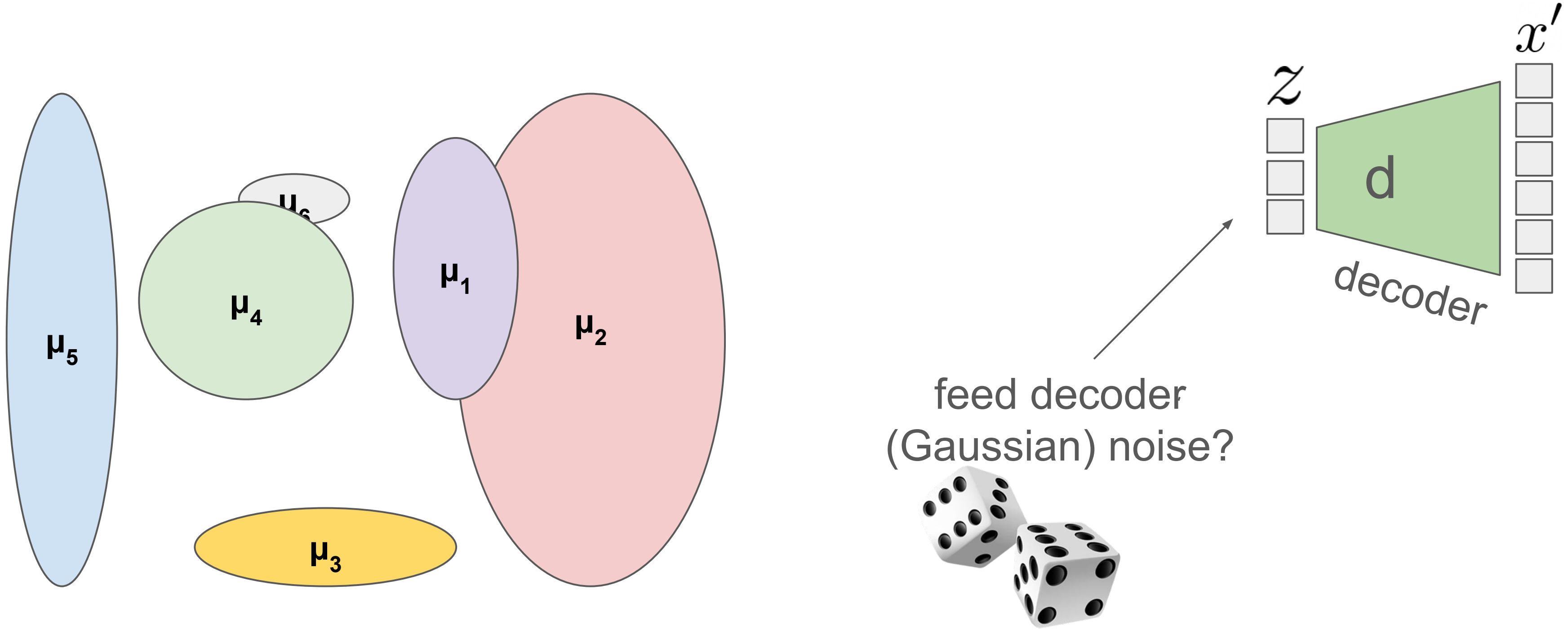
$$\max_{\phi,\theta} \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log\left(p_\theta(\mathbf{x}|\mathbf{z})\right)] = \min_{\phi,\theta} \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[(\mathbf{x} - d_\theta(\mathbf{z}))^2\right]$$

Plugging the output distribution into the reconstruction loss, results in the squared loss.

Cornell Bowers C·IS

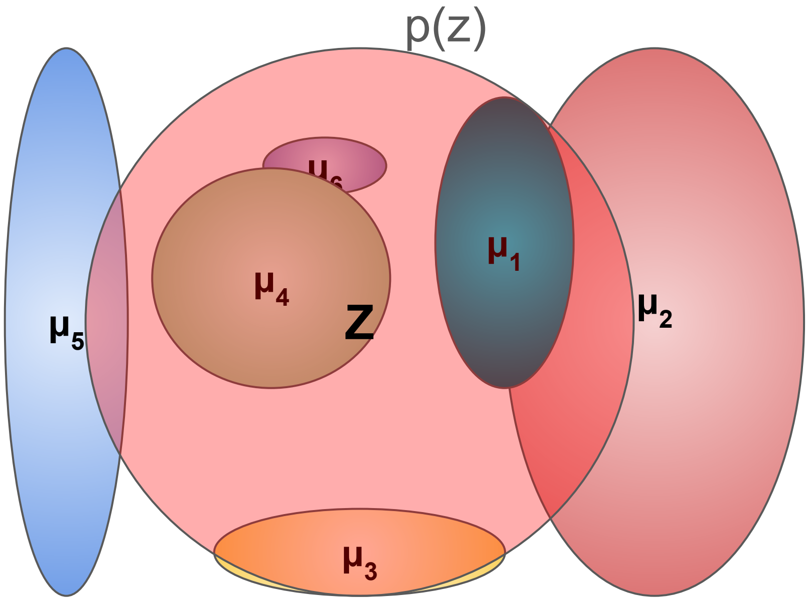
Step 2: How do we sample in latent space?

How can we sample, if each sample has its own latent distribution?



Step 2: How do we sample in latent space?

Solution: Regularize all distributions to be close to the standard normal ***N(0;I)***.

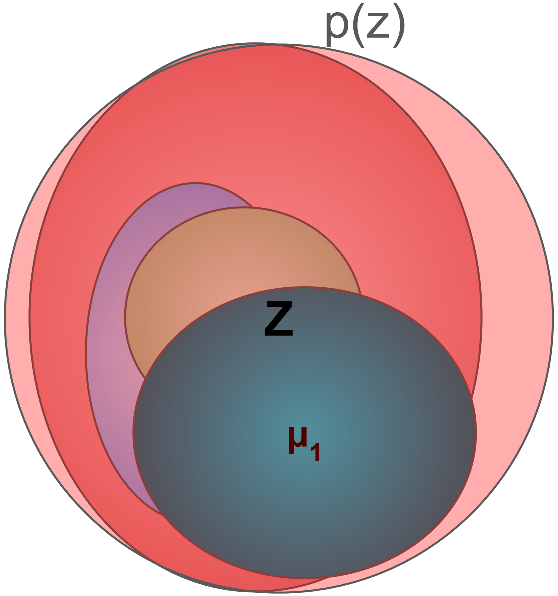


maximize

$$\underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z}))}_{\text{prior matching term}}$$

Step 2: How do we sample in latent space?

Solution: Regularize all distributions to be close to the standard normal ***N(0;I)***.

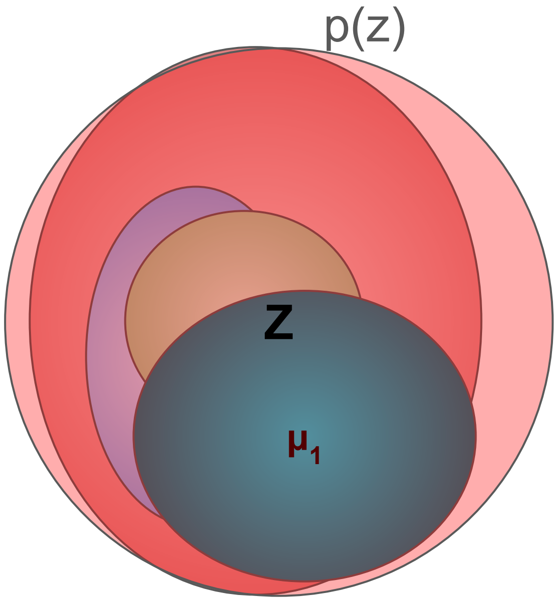


maximize

$$\underbrace{\mathbb{E}_{q_{\phi}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q_{\phi}(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z}))}_{\text{prior matching term}}$$

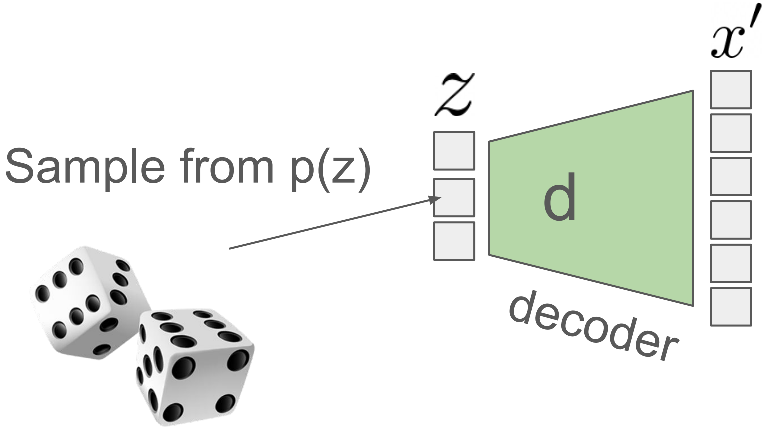
Step 2: How do we sample in latent space?

Solution: Regularize all distributions to be close to the standard normal ***N(0;I)***.



maximize

$$\underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z}))}_{\text{prior matching term}}$$



Evidence Lower Bound (ELBO)

$$
\begin{aligned}
\log p(\boldsymbol{x}) &= \log p(\boldsymbol{x}) \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) dz && \text{(Multiply by } 1 = \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z}\text{)} \\
&= \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})(\log p(\boldsymbol{x})) dz && \text{(Bring evidence into integral)} \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p(\boldsymbol{x})\right] && \text{(Definition of Expectation)} \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{x})}\right] && \text{(Apply Equation 2)} \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z}|\boldsymbol{x}) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right] && \text{(Multiply by } 1 = \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\text{)} \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z}|\boldsymbol{x})}\right] && \text{(Split the Expectation)} \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right] + D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z}|\boldsymbol{x})) && \text{(Definition of KL Divergence)} \\
&\geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right] && \text{(KL Divergence always} \geq 0\text{)}
\end{aligned}
$$

[Calvin Luo https://arxiv.org/abs/2208.11970]

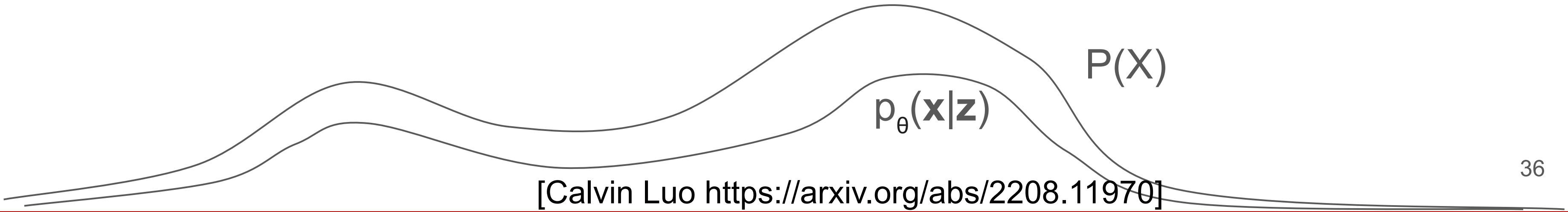
Cornell Bowers C·IS

Evidence Lower Bound (ELBO)

$$\begin{aligned}
\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right] &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right] && \text{(Chain Rule of Probability)} \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right] && \text{(Split the Expectation)} \\
&= \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z}))}_{\text{prior matching term}} && \text{(Definition of KL Divergence)}
\end{aligned}$$

(We are **maximizing** this lower bound.)

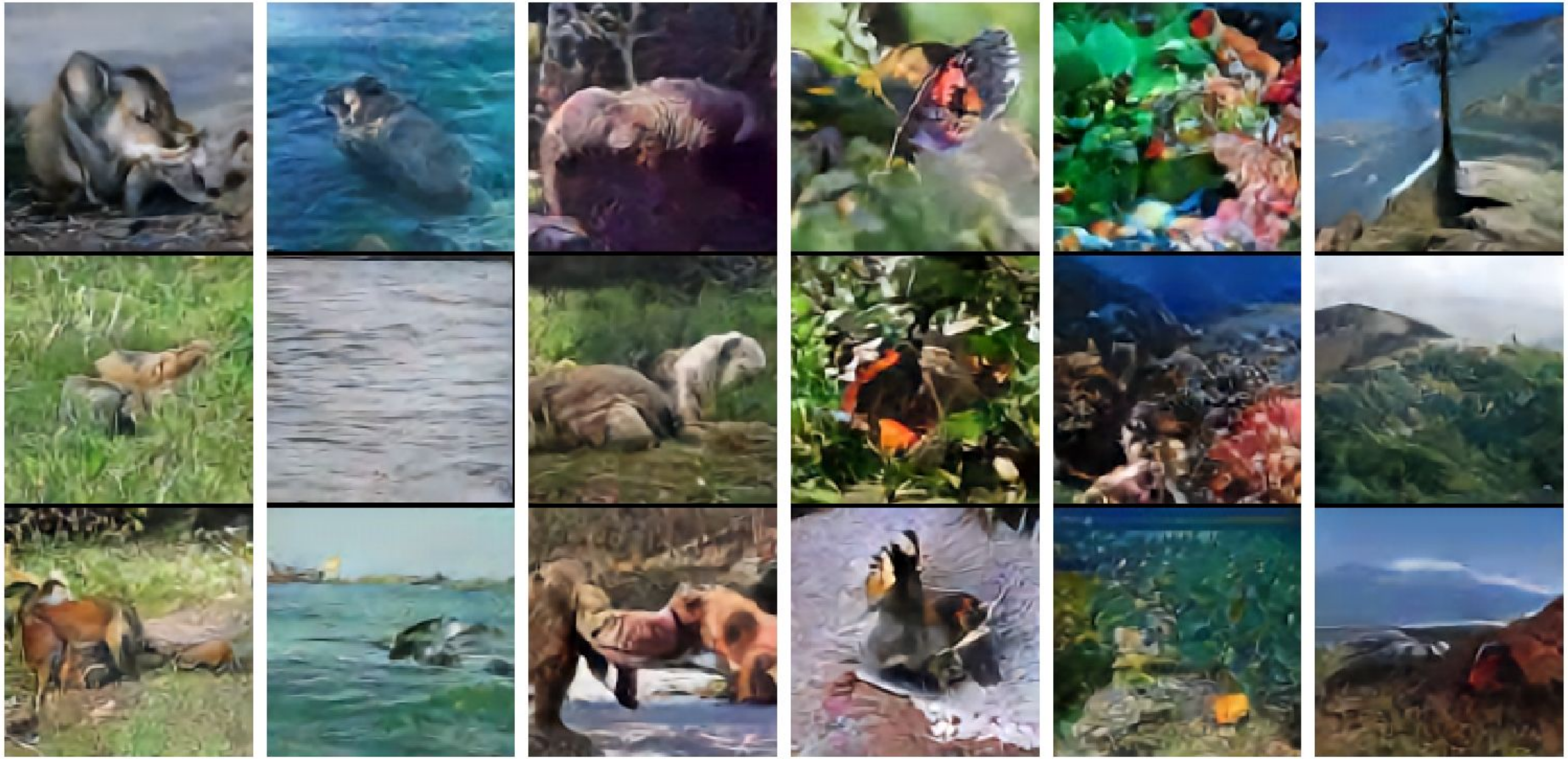
If we maximize p_θ(**x**|**z**) and minimize the D_{KL} we get close to P(**x**).



[Calvin Luo https://arxiv.org/abs/2208.11970]

Cornell Bowers C·IS

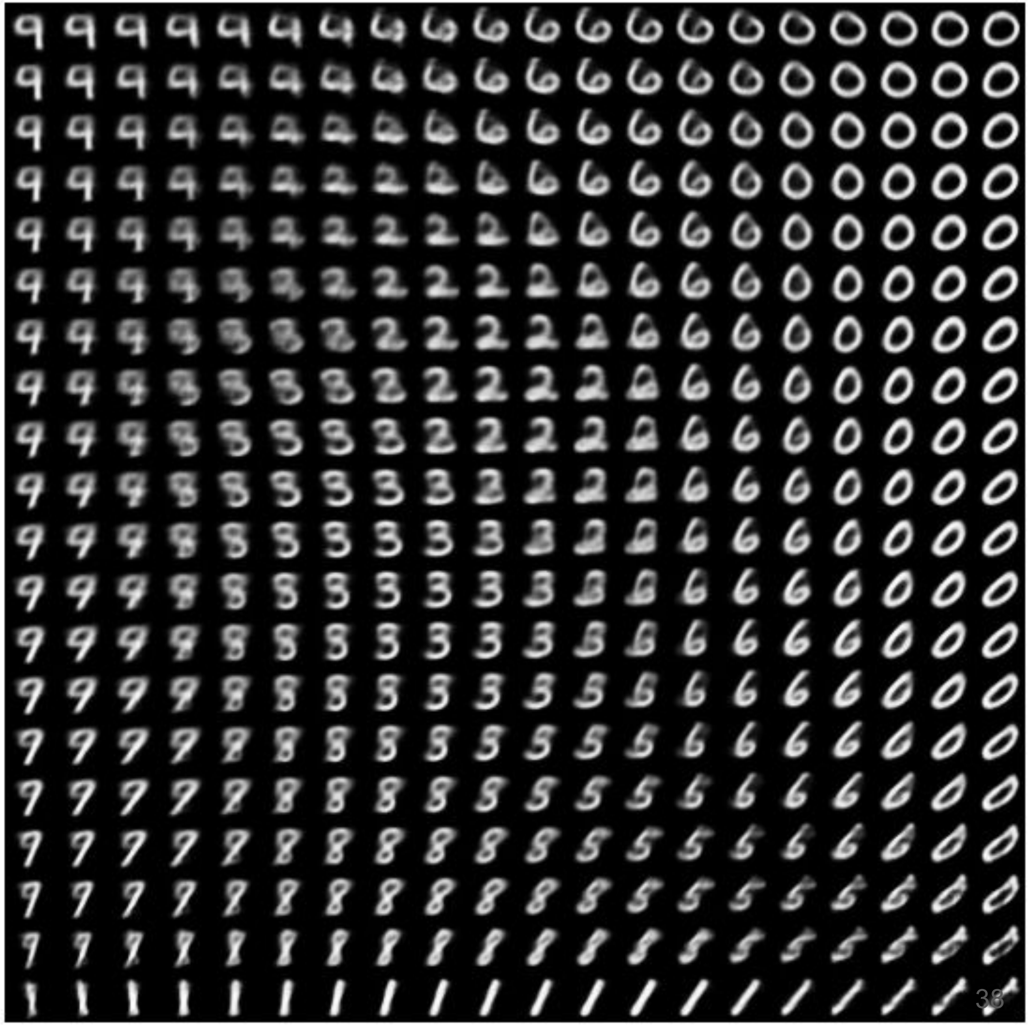
Examples of VAE generated images



Cornell Bowers C·IS

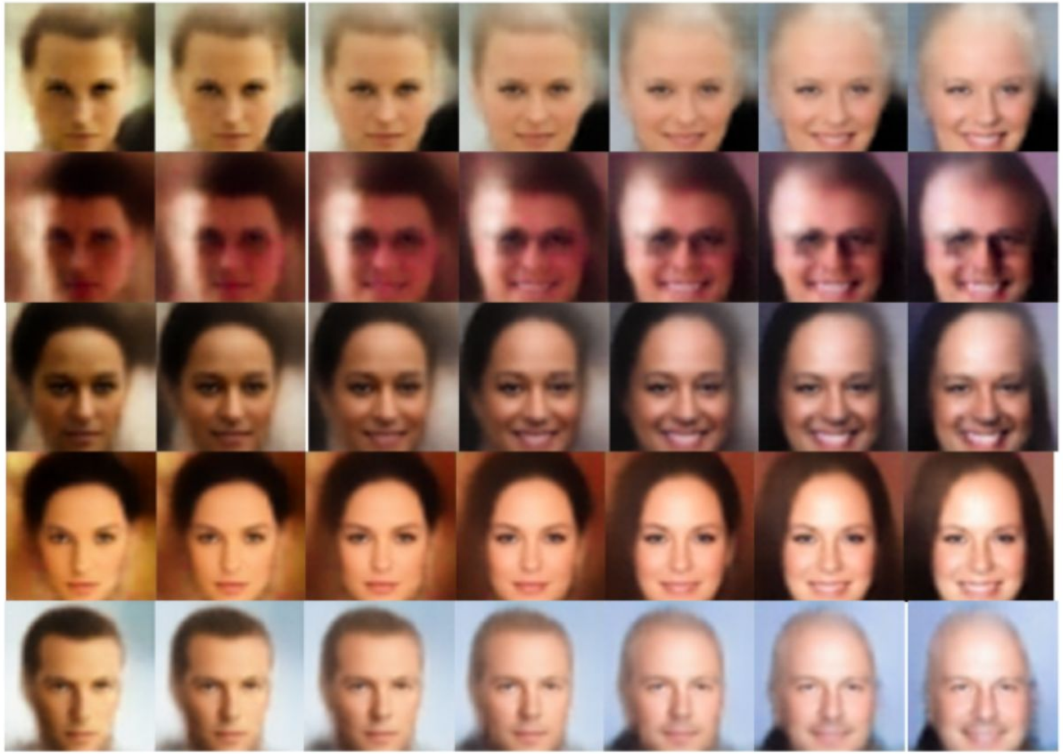
a much nicer space…

can smoothly interpolate digits in a meaningful, digit-y kind of way



a much nicer space

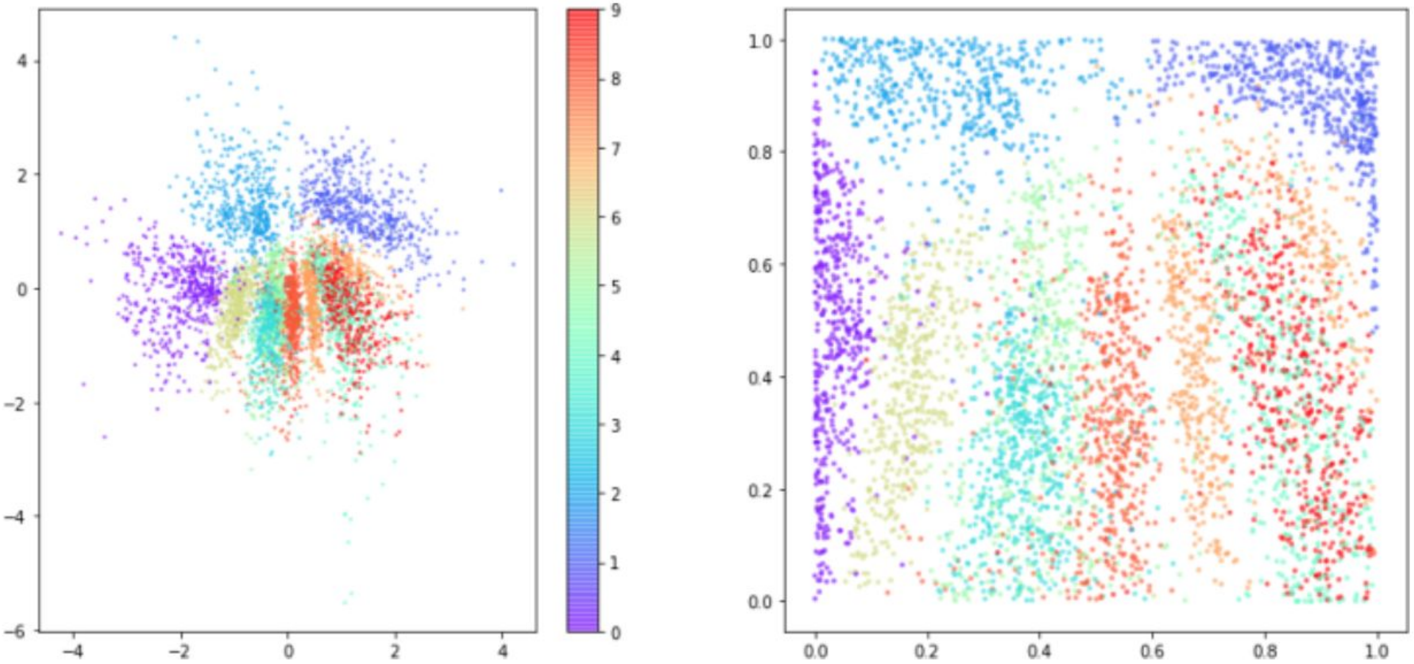
dimensions in latent space correspond to meaningful concepts, like sentiment and orientation



Cornell Bowers C·IS

Back to MNIST: Visualizing latent space again

VAE Latent space, note the distribution is centered, and each digit has an equal portion



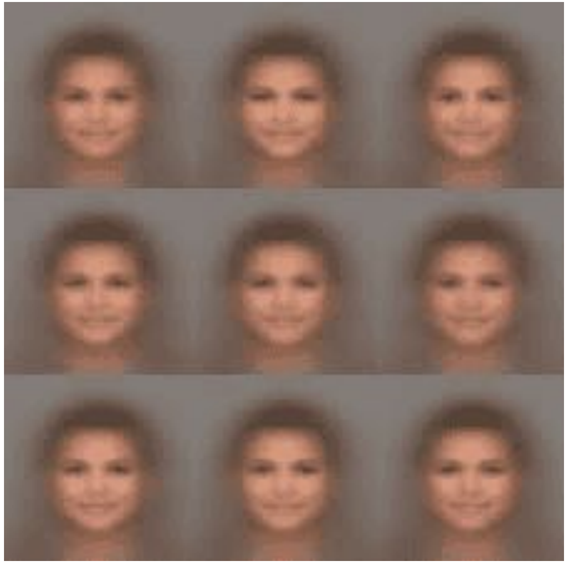
Cornell Bowers C·IS

The Biggest Drawback of VAEs

- Out of the box, generated images can be blurry.

Question: Why? How do GANs fix this problem?

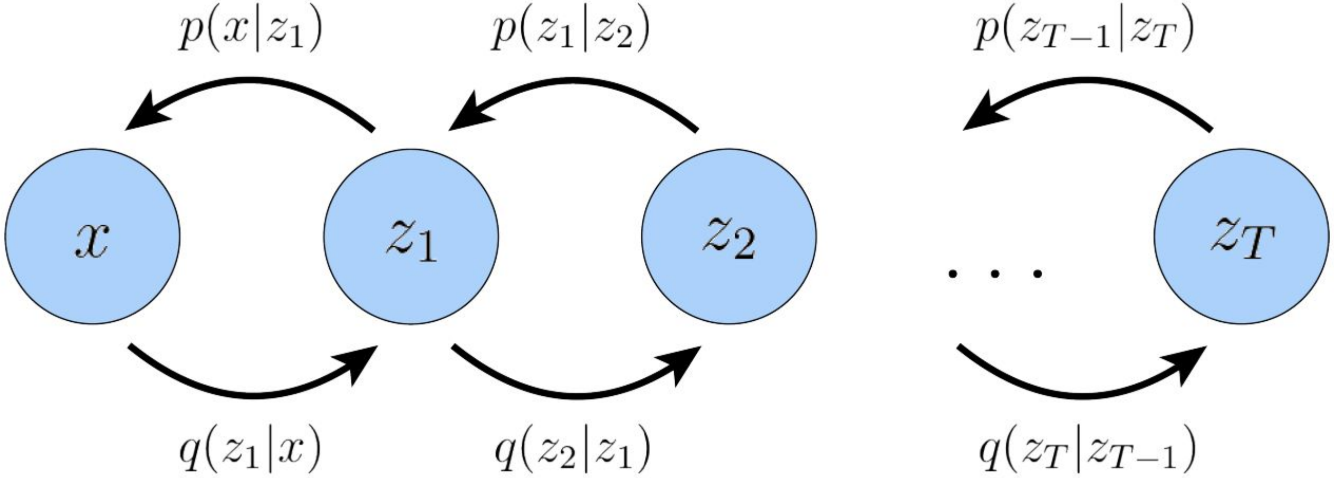
VAE v. GAN



https://borisburkov.net/2022-12-31-1/

Hierarchical VAEs

The generative process is modeled as a Markov chain, where each latent z_t is generated only from the previous latent z_{t+1}



Summary

- Generative Image models learn a mapping from the **Standard Normal Gaussian** to the **Image Manifold**
  - GANs learn this through a **discriminator**.
  - VAEs learn it through **variational autoencoders**
- AutoEncoders learn to **compress** and **reconstruct** data
- VAEs make these AutoEncoders **probabilistic**
  - Minimize the **reconstruction loss**
  - Latent space is sampled from Gaussian distributions
  - Sampling is made differentiable with the **Reparameterization Trick**
  - Deviations from the Prior (Standard Normal Gaussian) is penalized by **KL divergence**
- The ELBO is a **lower bound** of P(X)
  - Maximizing the ELBO, and minimizing the KL divergence makes P(x|z) close to P(x)