AUTOENCODERS

Based on cs109b slides

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Unsupervised/Predictive Learning (cake)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

- **(Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)**

Original LeCun cake analogy slide presented at NIPS 2016, the highlighted area has now been updated.

Y. LeCun

How Much Information is the Machine Given during Learning?

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

© 2019 IEEE International Solid-State Circuits Conference

LeCun updated his cake recipe last week at the 2019 International Solid-State Circuits Conference (ISSCC) in San Francisco, replacing "unsupervised learning" with "self-supervised learning," a variant of unsupervised learning where the data provides the supervision.

Outline

- Quick review of representation learning
- Brief history of encoding/decoding
- Autoencoder Variations:
    - Convolutional Autoencoders
    - Denoising Autoencoders
    - Sparse Autoencoders
- Variational Autoencoders

Quick Review. Neural Networks as function approximation.

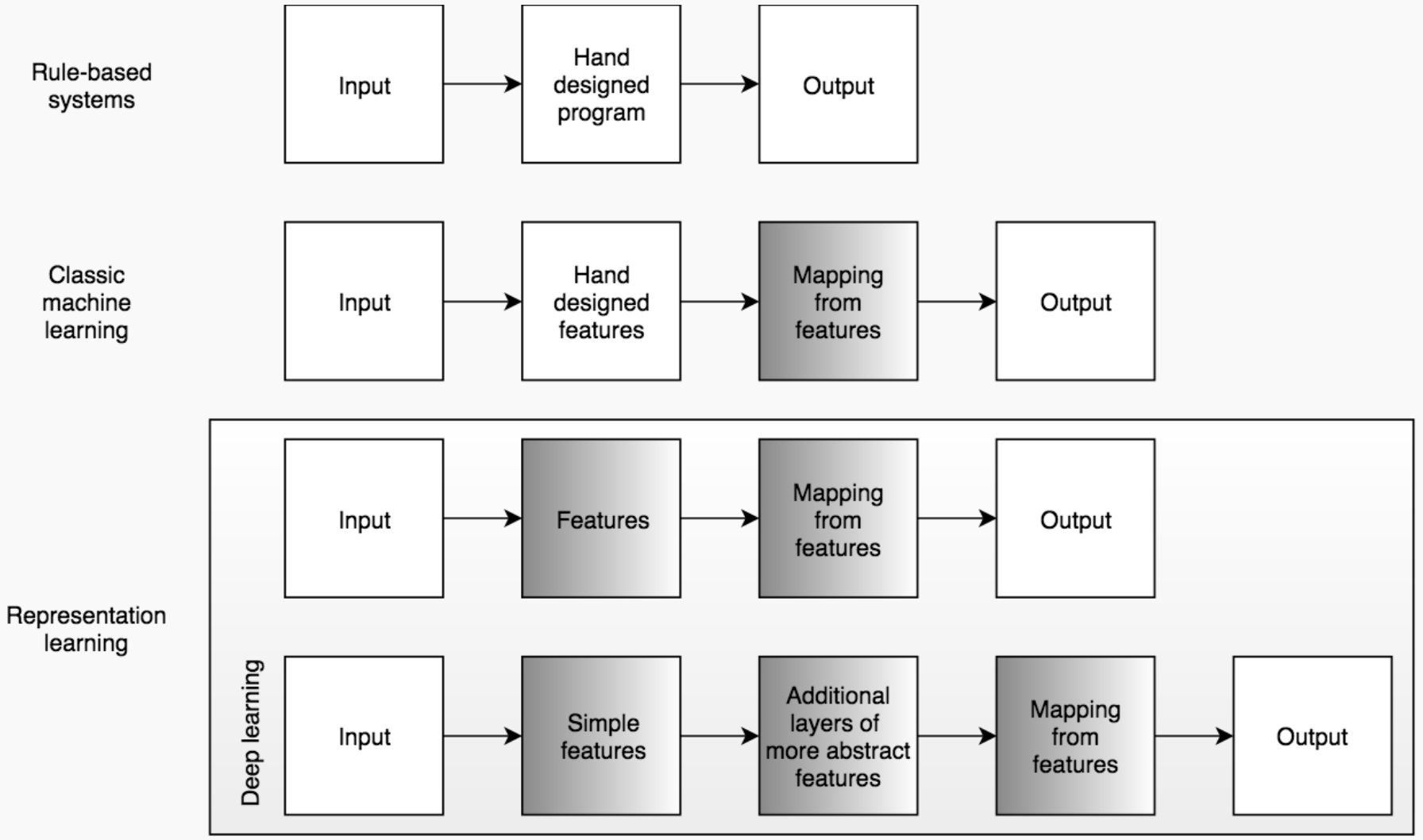
Given an input x and an output y there exists a mapping from input space to output space as follows:

Our goal is to find an estimate of $f(x)$ which we will call .

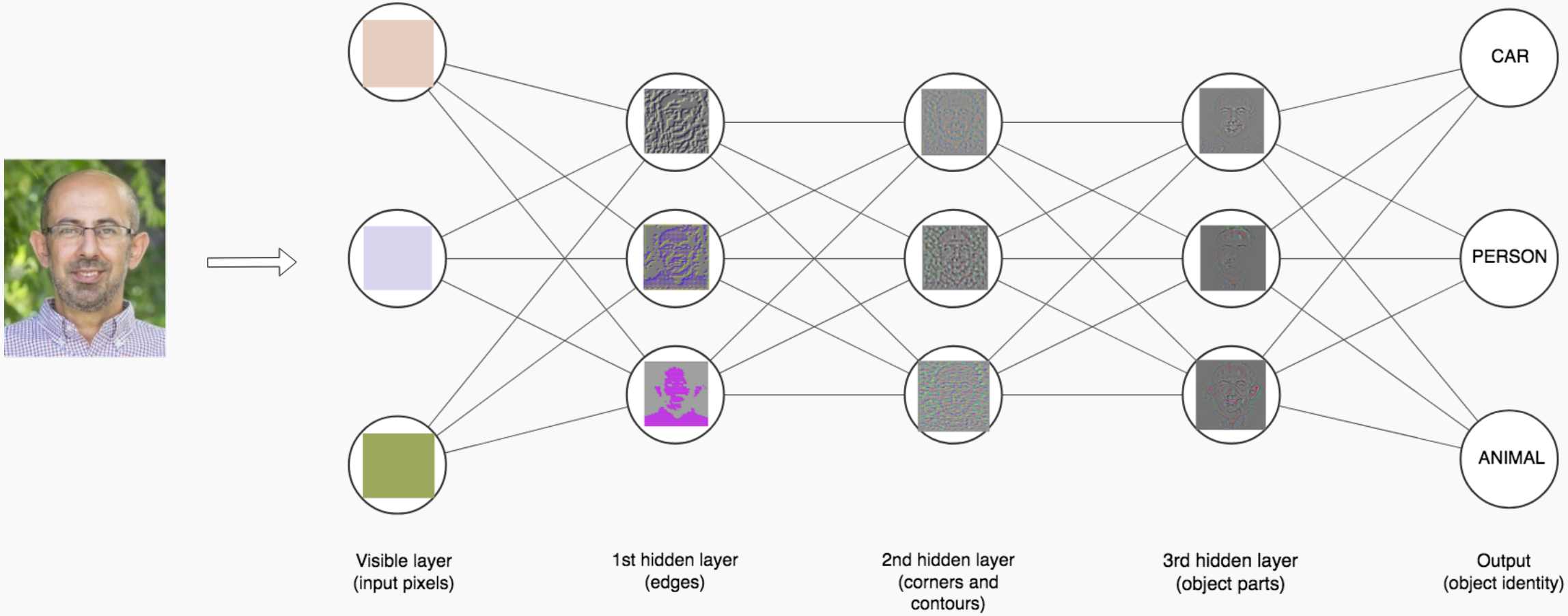
Statistical learning or modeling is the process of finding .

Neural networks are one of many possible methods we can use to obtain the estimate .

Representational Learning (cont)

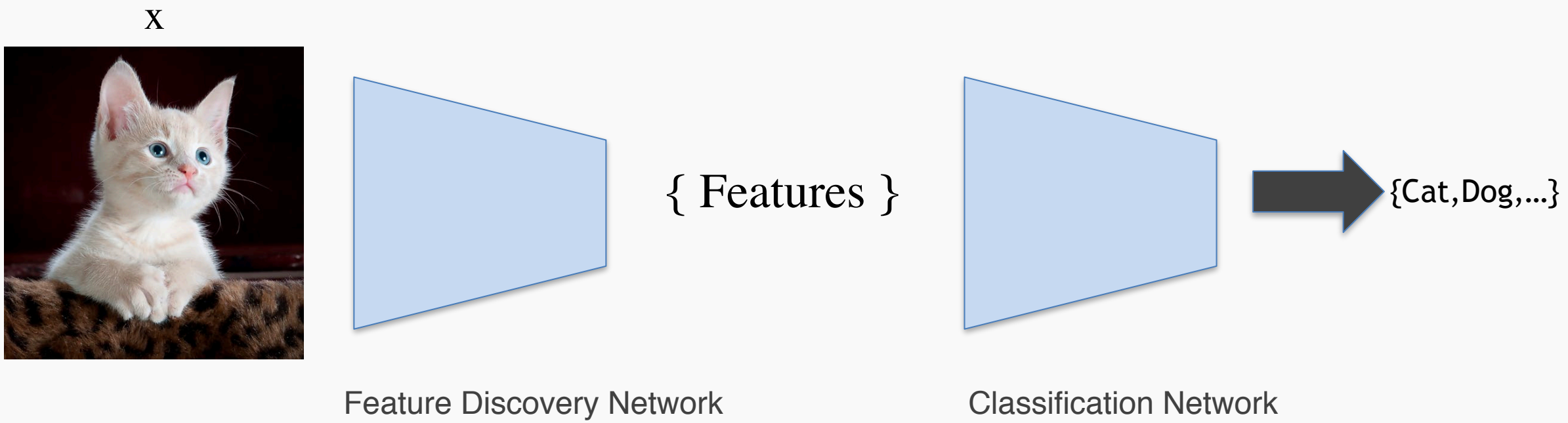


Representational Learning (cont)



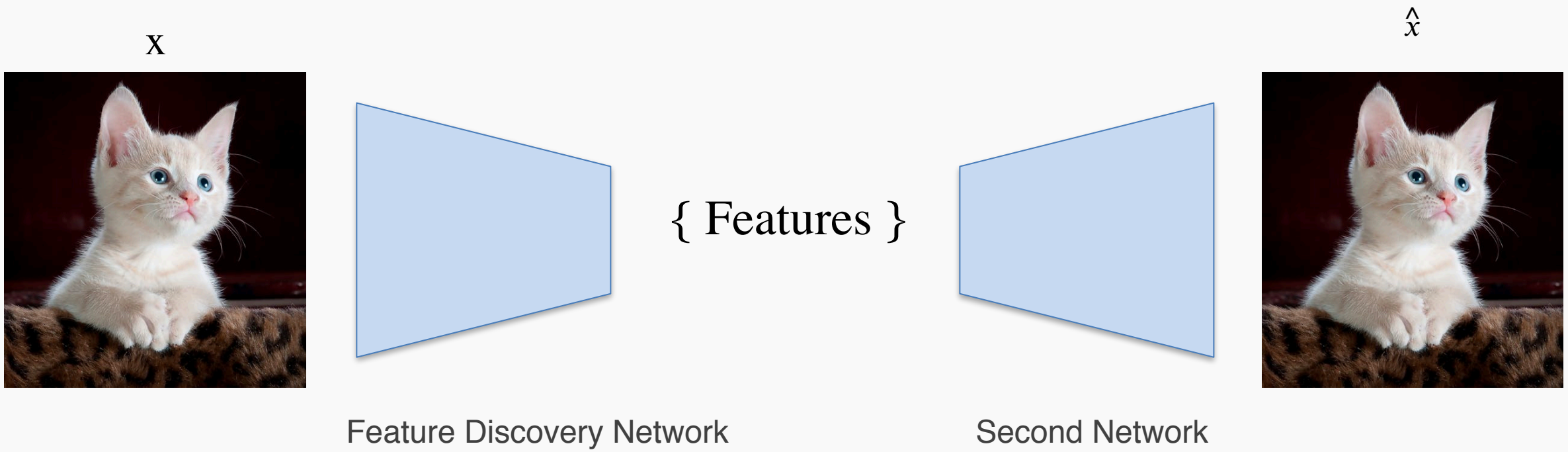
Representational Learning: Supervised Learning

We train the two networks by minimizing the loss function (cross entropy loss)



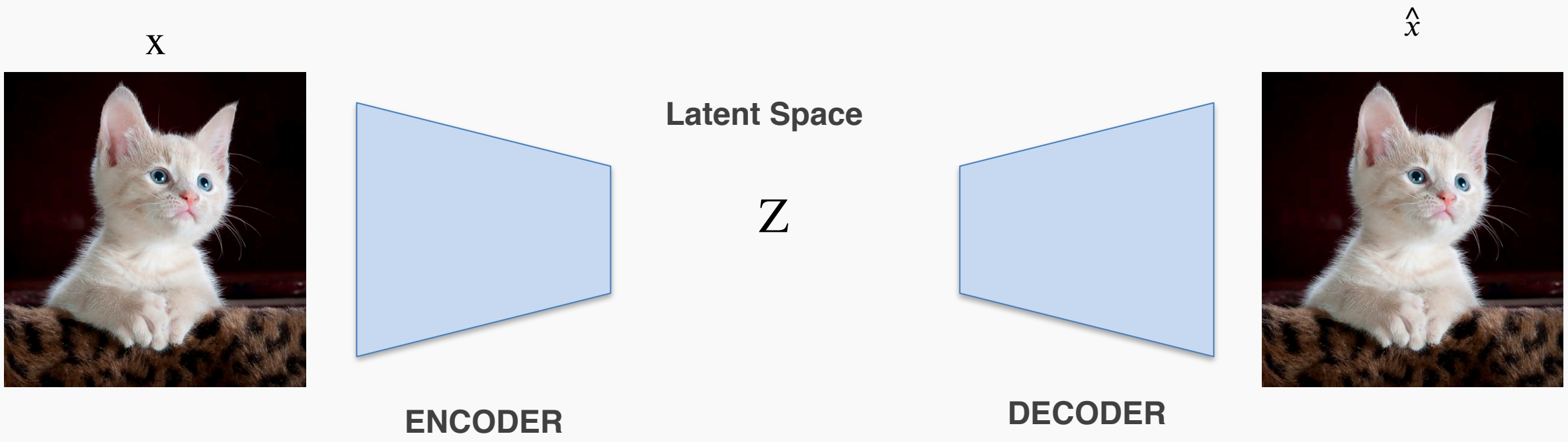
Representational Learning: Self-supervised Learning

We train the two networks by minimizing the reconstruction loss function:



Representational Learning: Self-supervised Learning

AUTOENCODER



This is an autoencoder. It gets that name because it automatically finds the best way to encode the input so that the decoded version is as close as possible to the input.

What are autoencoders?

- A particular kind of learning architecture
- A mechanism of compressing inputs into a form that can later be decompressed
- Similar to the way MP3 compresses audio and JPG compresses images
- Autoencoders are more general than either MP3 or JPG
- They are usually used to …
    - reduce data dimensionality or find a better suited representation for another task
    - blend inputs from one input to another.
    - denoise, infill, etc.

Lossless and Lossy Encoding

The greater the difference between the original version and the version post-decompression the greater the **loss**.

For example imagine you are in Boston and you want to write a birthday text to a special friend.

HANNAH HAPPY BIRTHDAY I LOVE YOU DAN

In the freezing (-20C) Boston winter you do not want to have your hands out in the open, so you shorten the message as much as possible using text-speak:

H HBD ILY D

Your 36 characters message is compressed to 11 characters

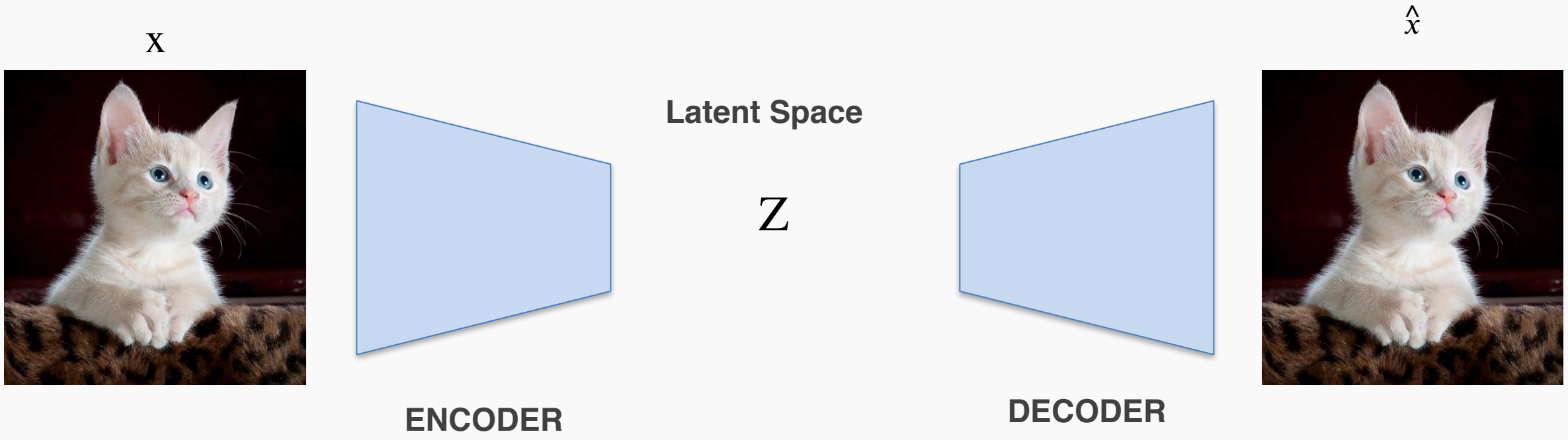
Lossless and Lossy Encoding (cont)

HPD is unambiguous given that it is her birthday today, but **D** could mean **Dan or David or Donny ...** You can imagine the potential drama**.** (Is this an example lossy or lossless compression?)

A way to test if a transformation is **lossy** or **lossless** is to consider if it can be inverted, or run backwards, to provide us with the original data.

Autoencoder

We train the two networks by minimizing the reconstruction loss function:



This is an autoencoder. It gets that name because it automatically finds the best way to encode the input so that the decoded version is as close as possible to the input.

The simplest autoencoder

Encode with a simple fully connected network (FCN)

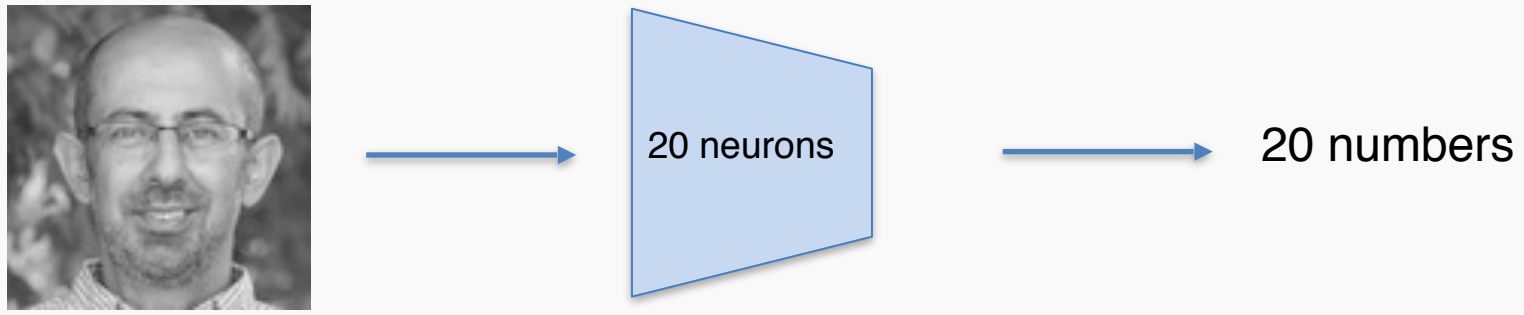
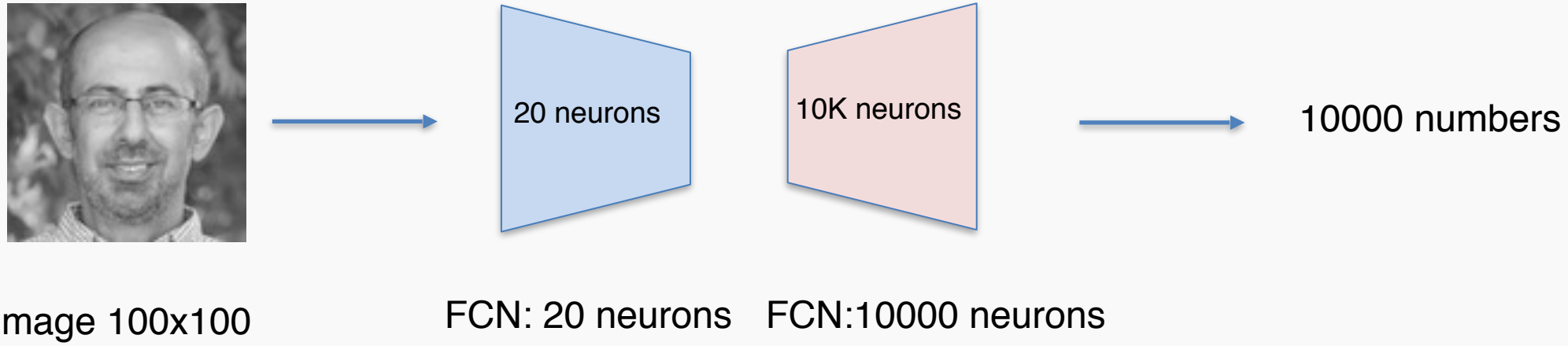


Image 100x100

FCN with 20 neurons

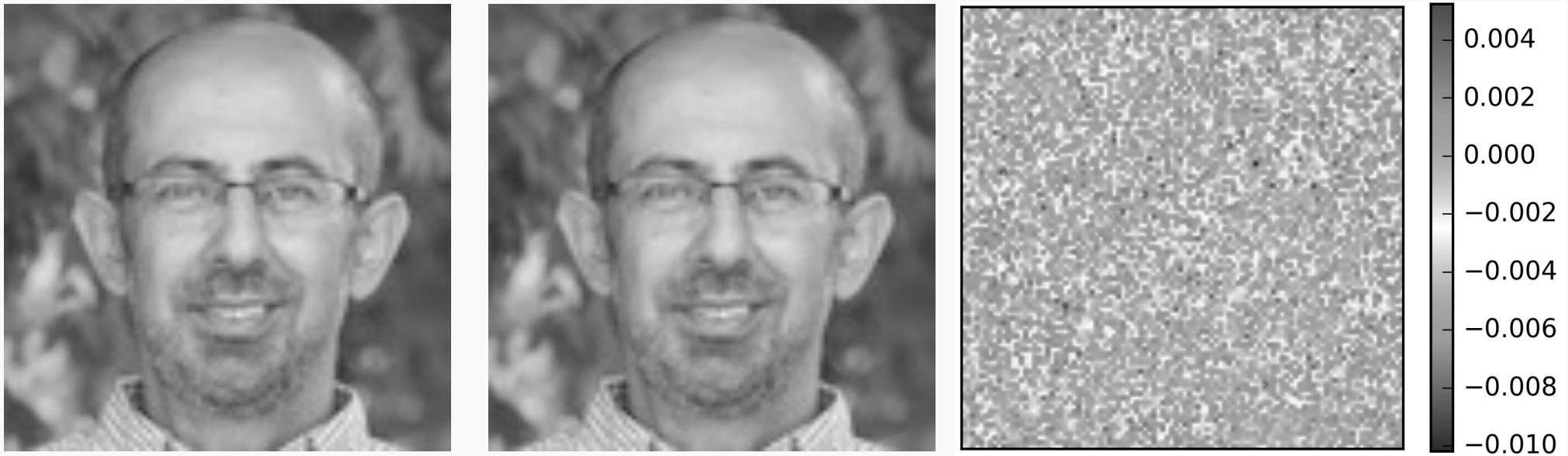
The simplest autoencoder

Encode and decode



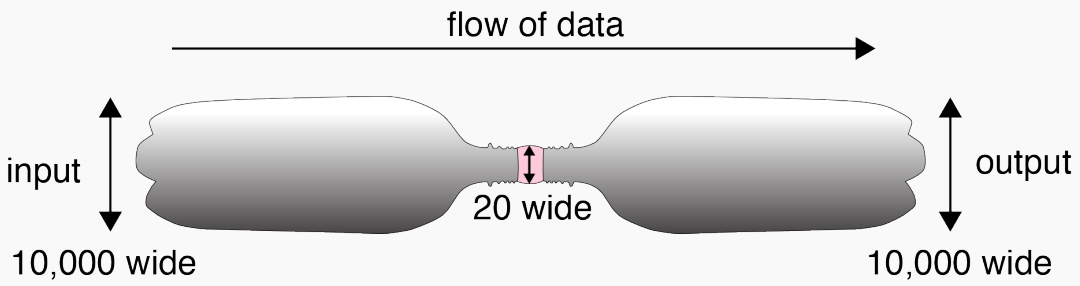
Autoencoders in action

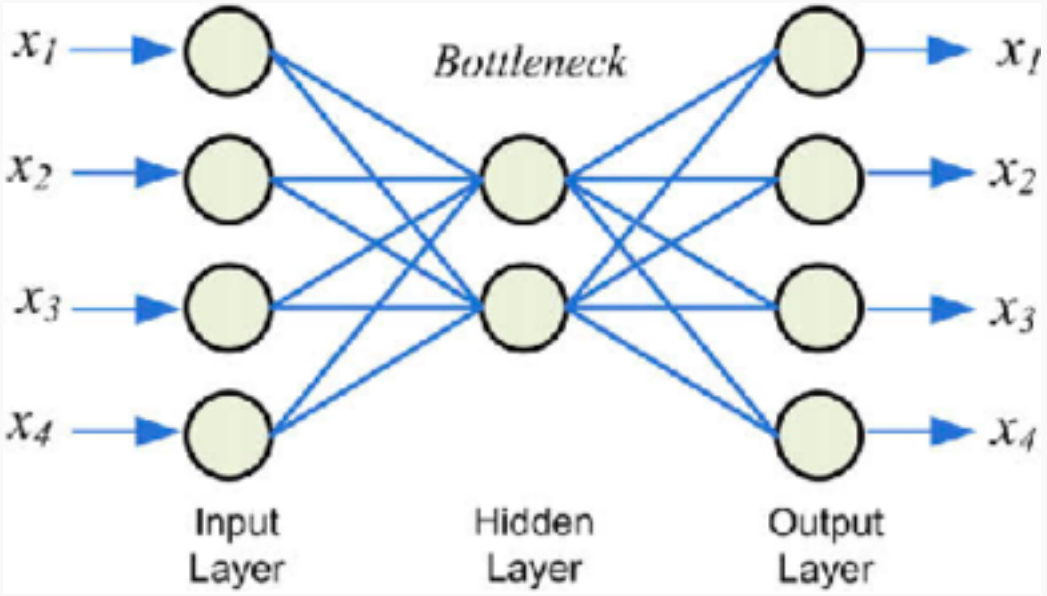
Comparing the input and output pixel by pixel.



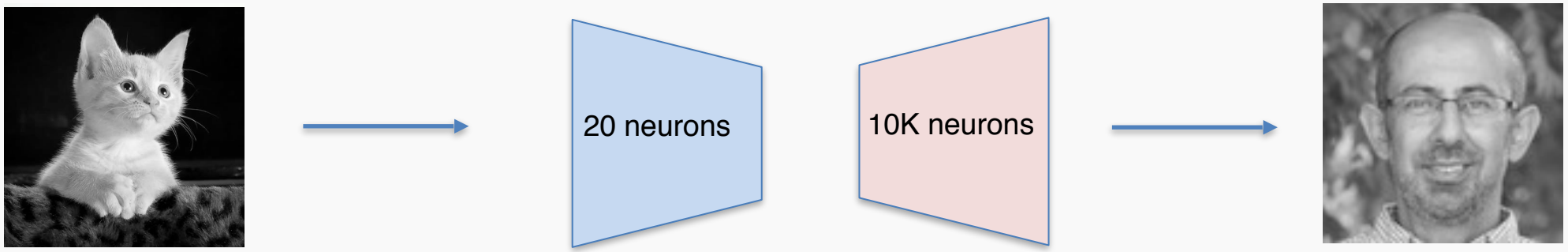
Bottleneck

- We start with 10,000 elements
- We have 20 in the middle
- And 10,000 elements again at the end

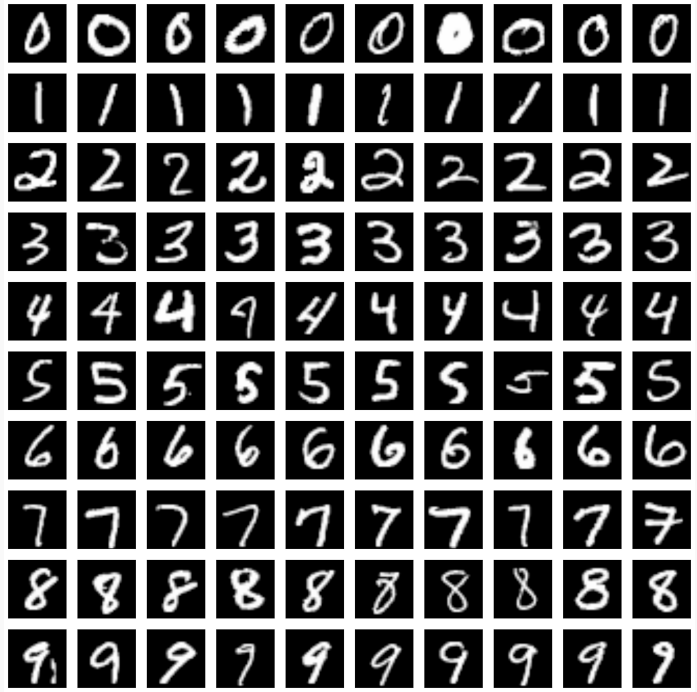


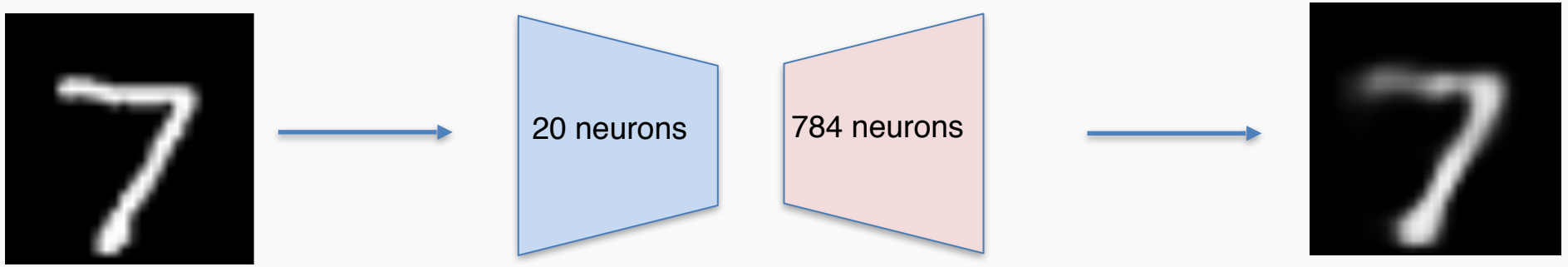


General Encoder Decoder Architectures



A Better autoencoder





20 latent variables

original

reconstructed

10 latent variables

original

reconstructed

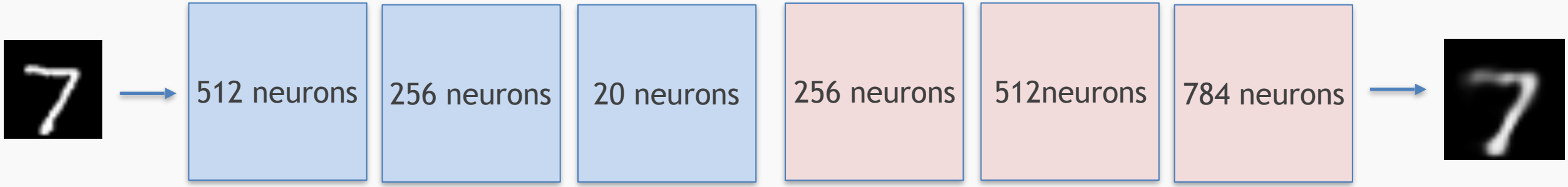
2 latent variables

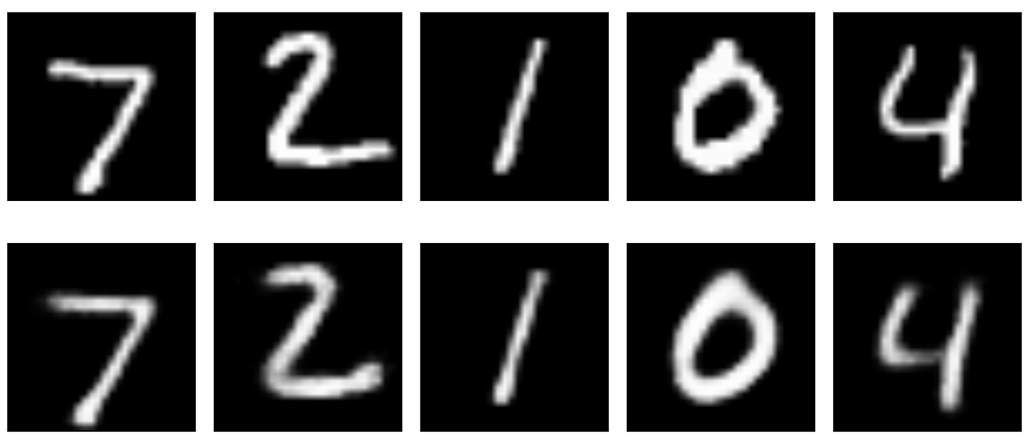
original

reconstructed

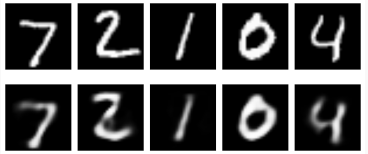
Image taken from A. Glassner, Deep Learning, Vol. 2: From Basics to Practice

Deeper

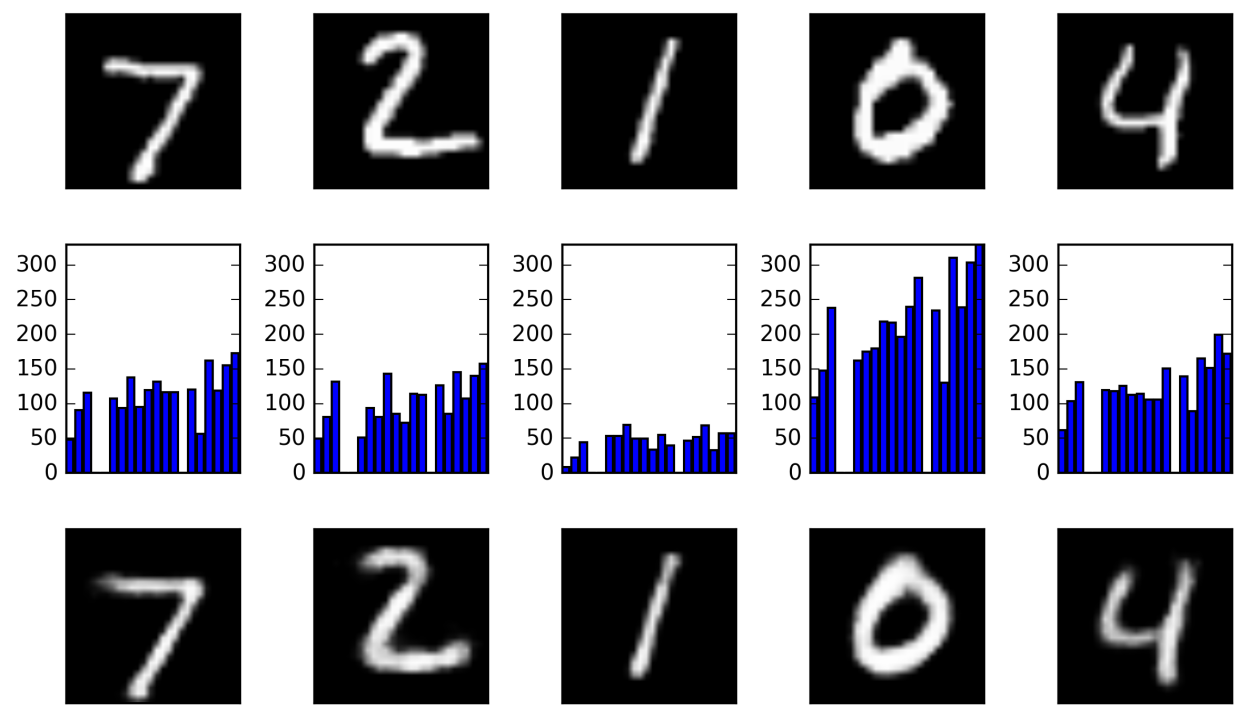




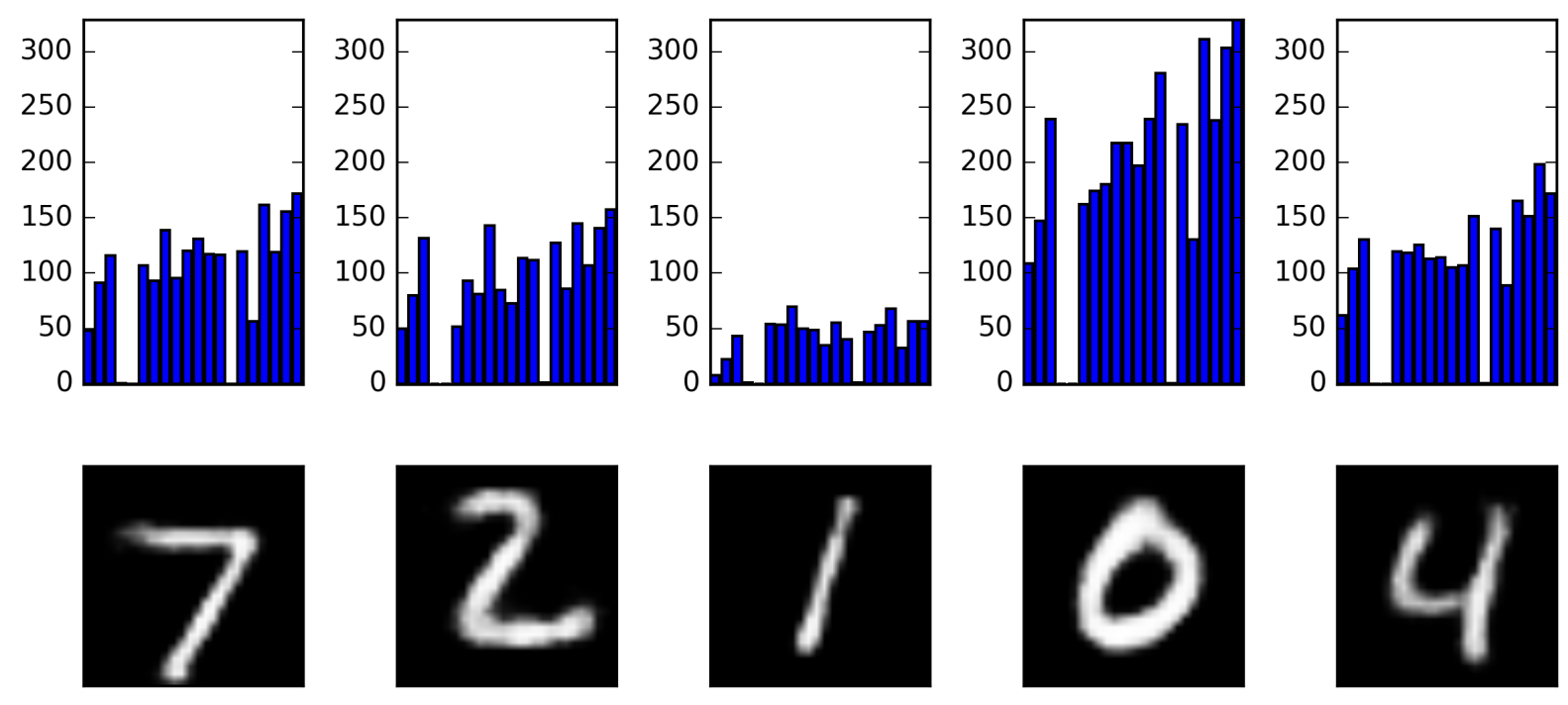
Shallow 20 latent variables



Exploring autoencoders

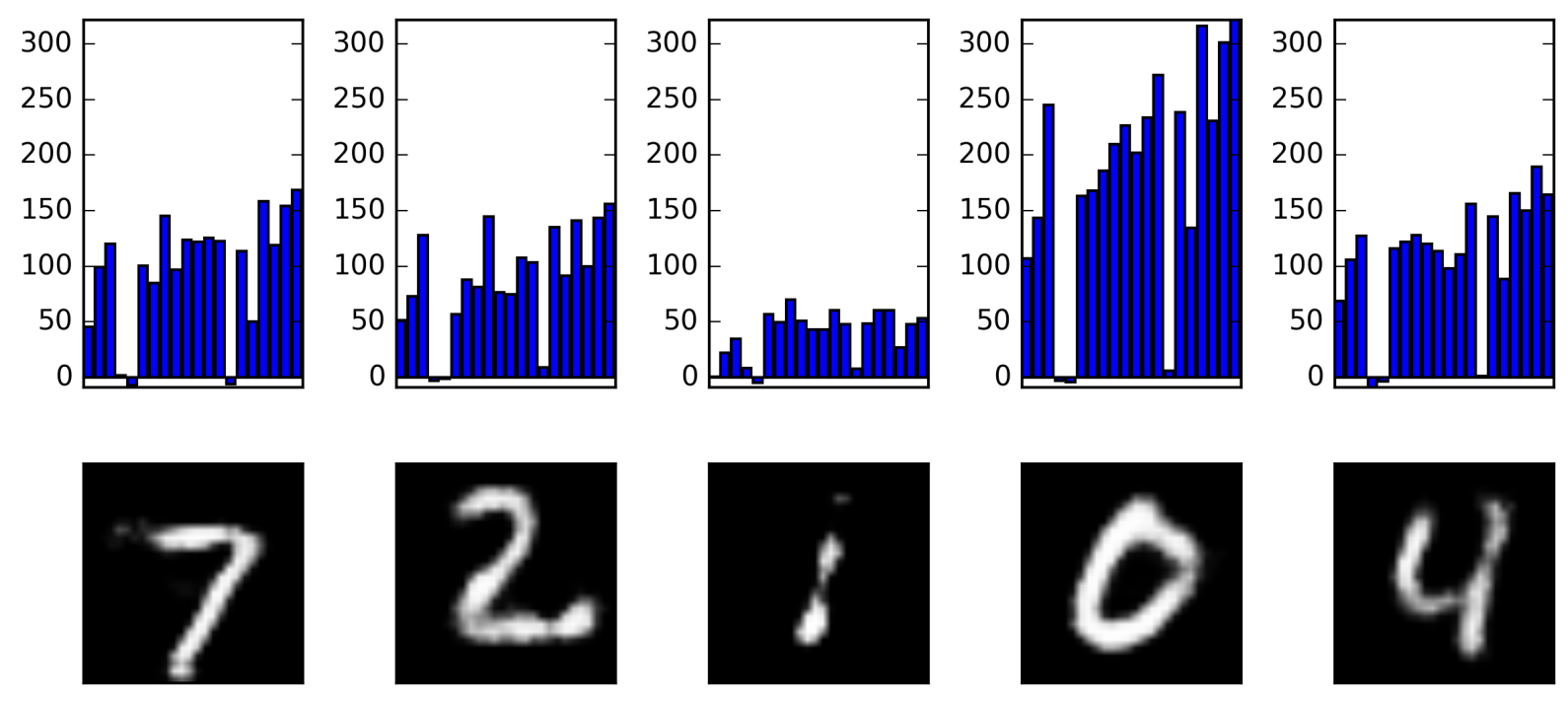


Exploring autoencoders (cont)



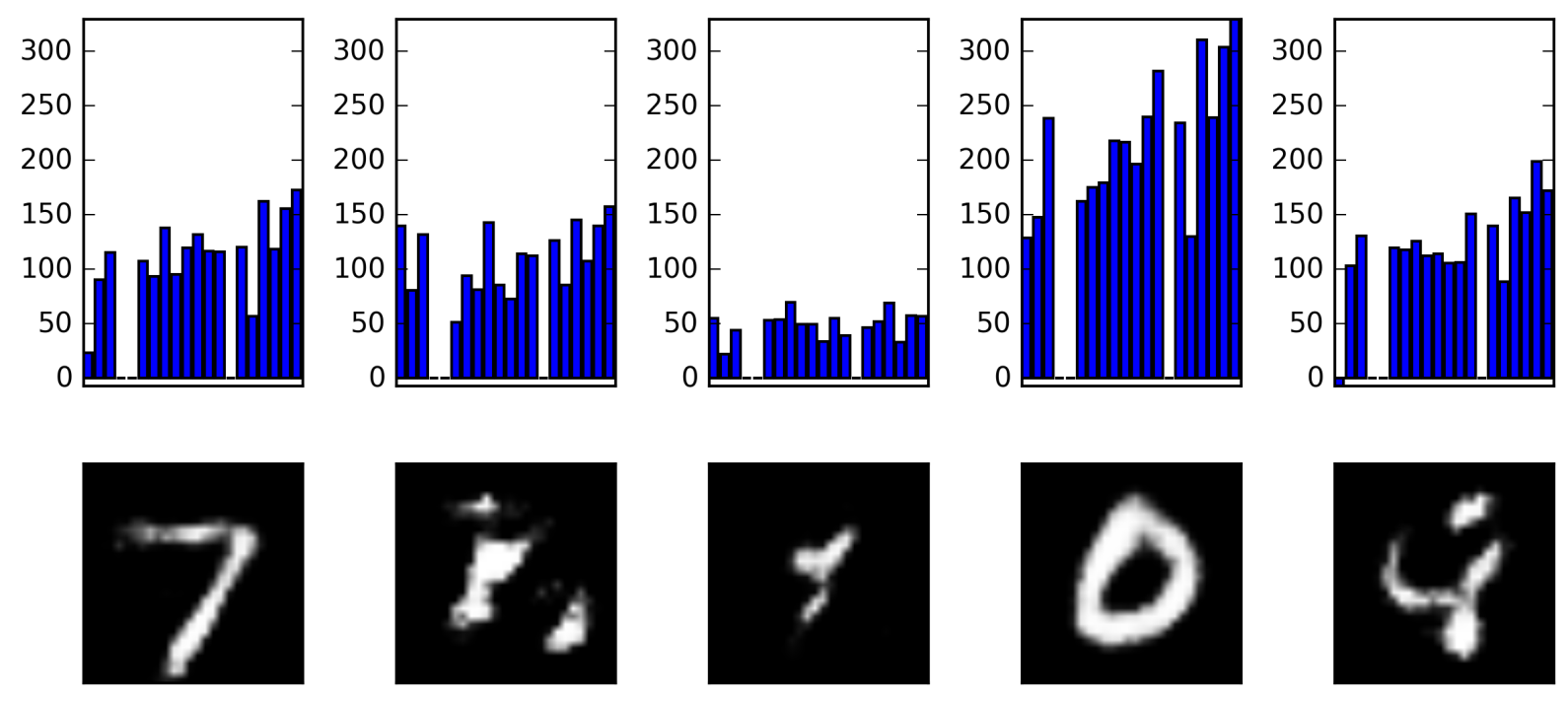
Change +/- 1 the latent variables

Exploring autoencoders (cont)



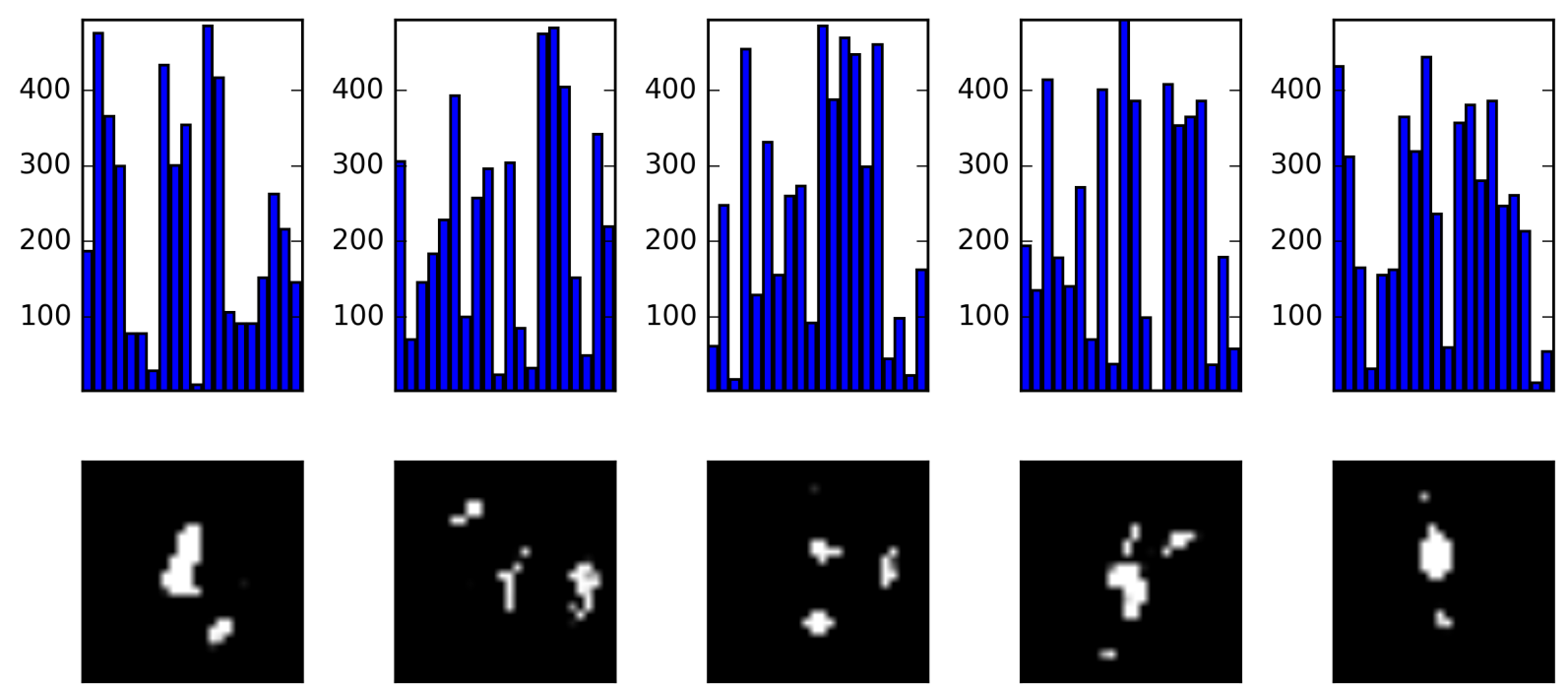
Change +/- 10 the latent variables

Exploring autoencoders (cont)



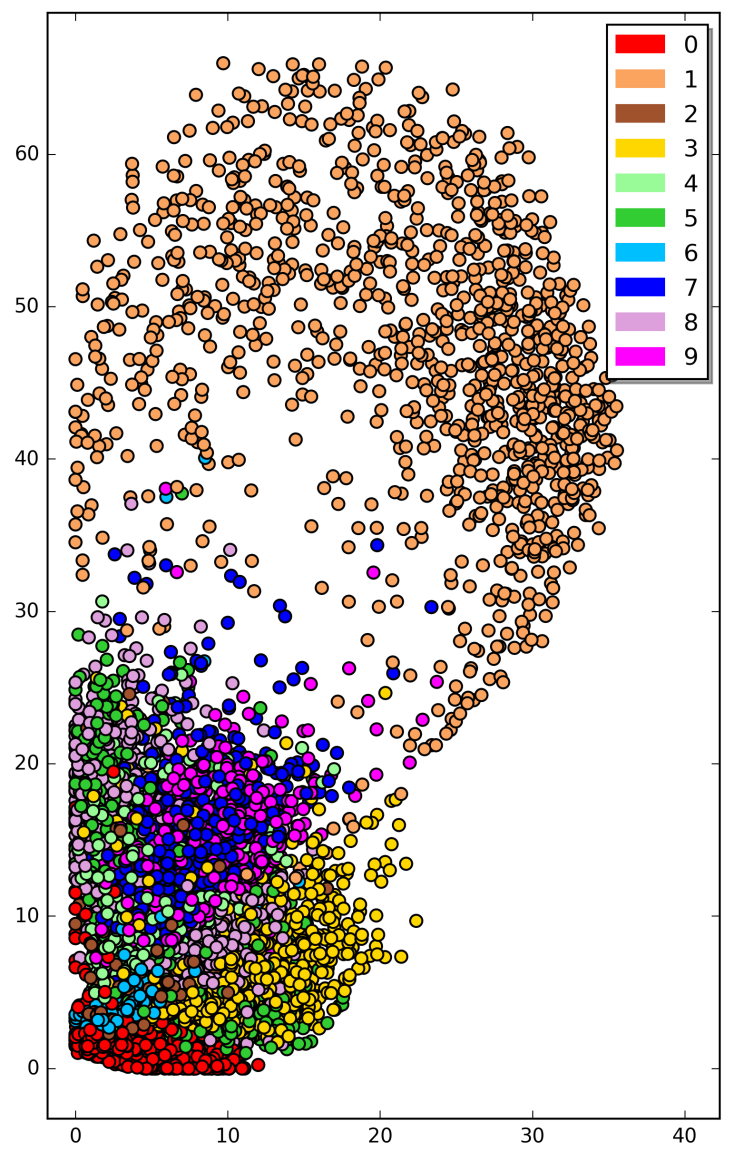
Change +/- 100 the first latent variables

Exploring autoencoders (cont)



Just noise

Parameter space of autoencoder



Blending

We blend inputs to create new data that is similar to the input data, but not exactly the same.

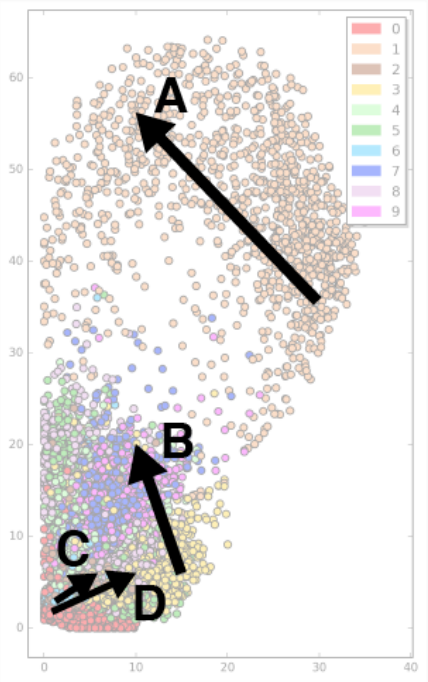
One example of blending is **content blending** where the content of two pieces of data is directly blended. An example is if we overlay images of a cow and zebra.

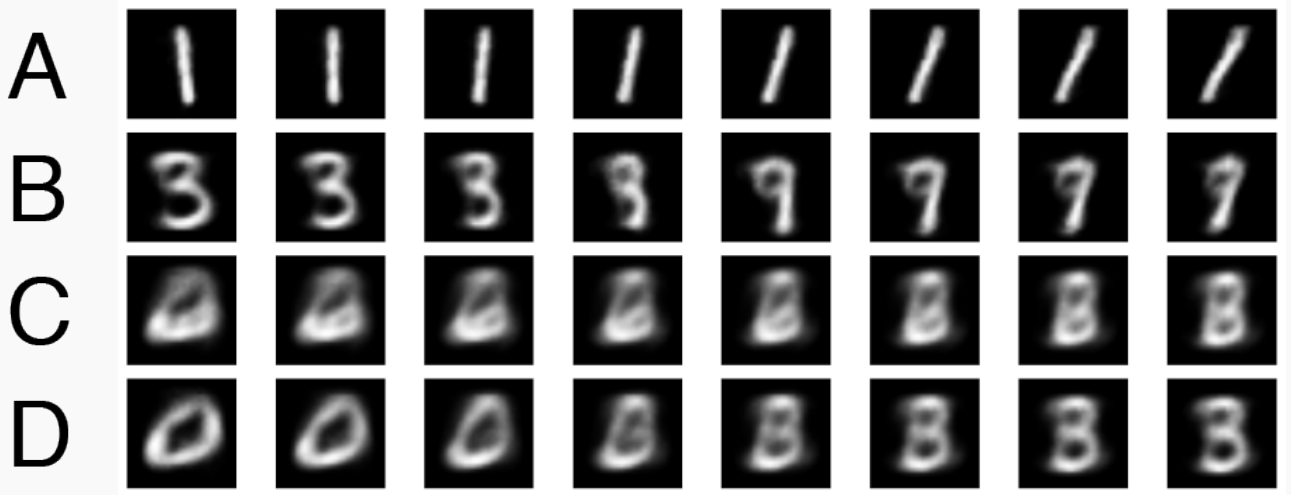


Image taken from A. Glassner, Deep Learning, Vol. 2: From Basics to Practice

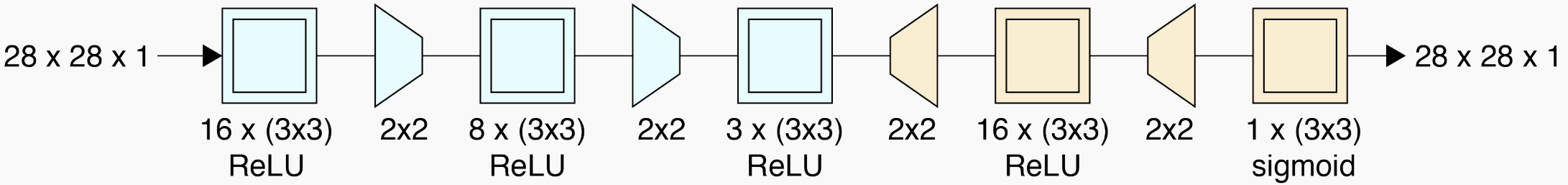
Blending Latent Variables

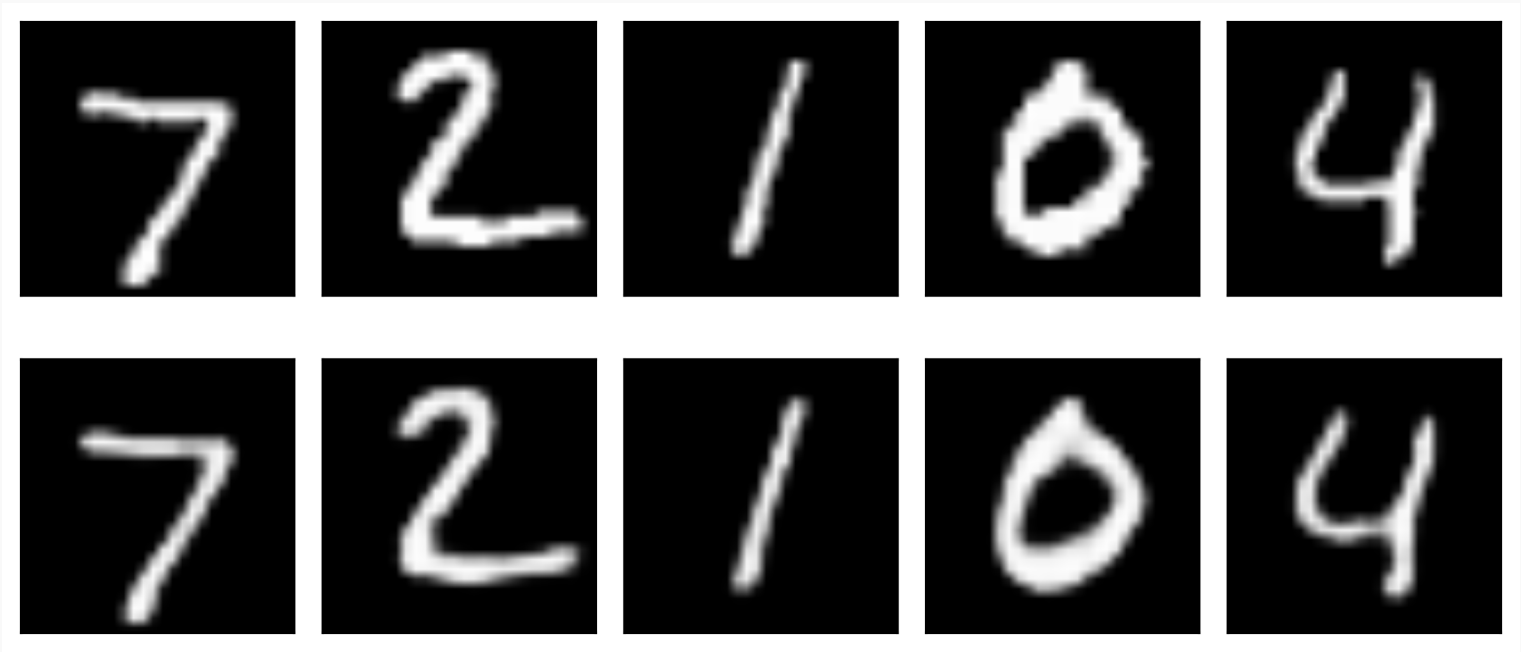
Back to the example of MNIST





Convolutional Autoencoders

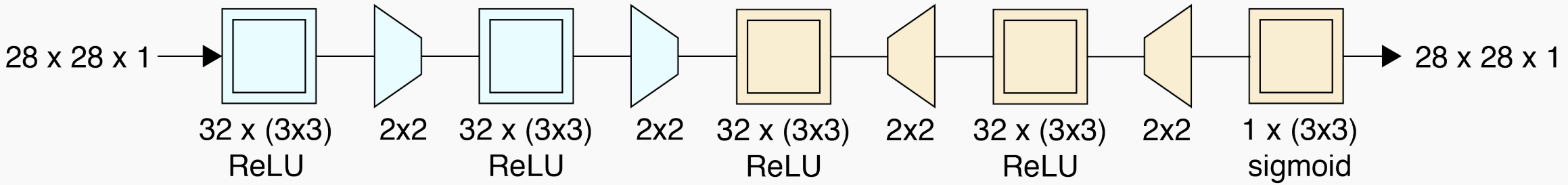




Denoising

A popular use of autoencoders is to remove noise from samples.

Denoising (cont)



Note that we start with a clean image, add noise and train our networks to return a clean decoded image.

Regularized Autoencoders (cont)

This trade-off requires the model to maintain only the variations in the data required to reconstruct the input without holding on to redundancies within the input.

Question: How to achieve this?

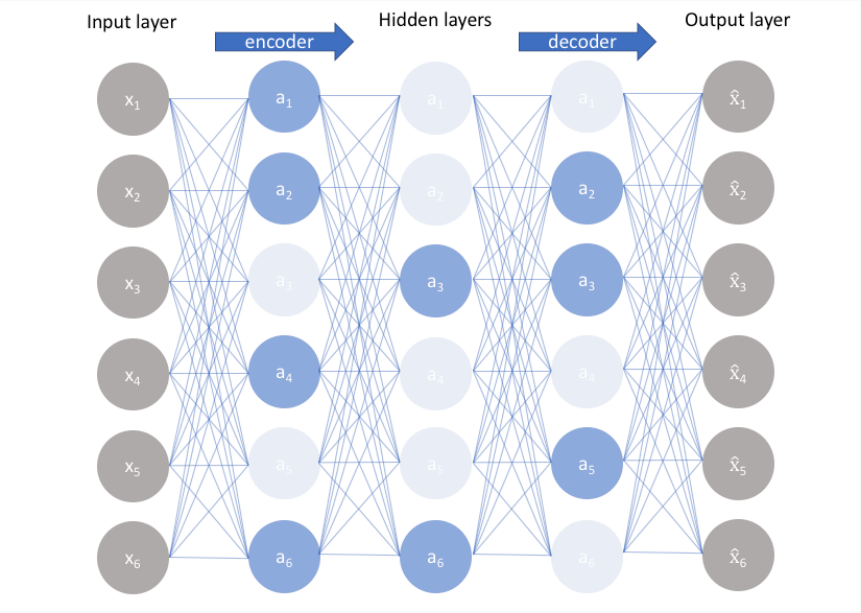
> For most cases, this involves constructing a loss function where one term encourages our model to be sensitive to the inputs (ie. reconstruction loss and a second term discourages memorization/overfitting (ie. an added regularizer).

Regularization on output of encoder, not on network parameters

Sparse Autoencoders

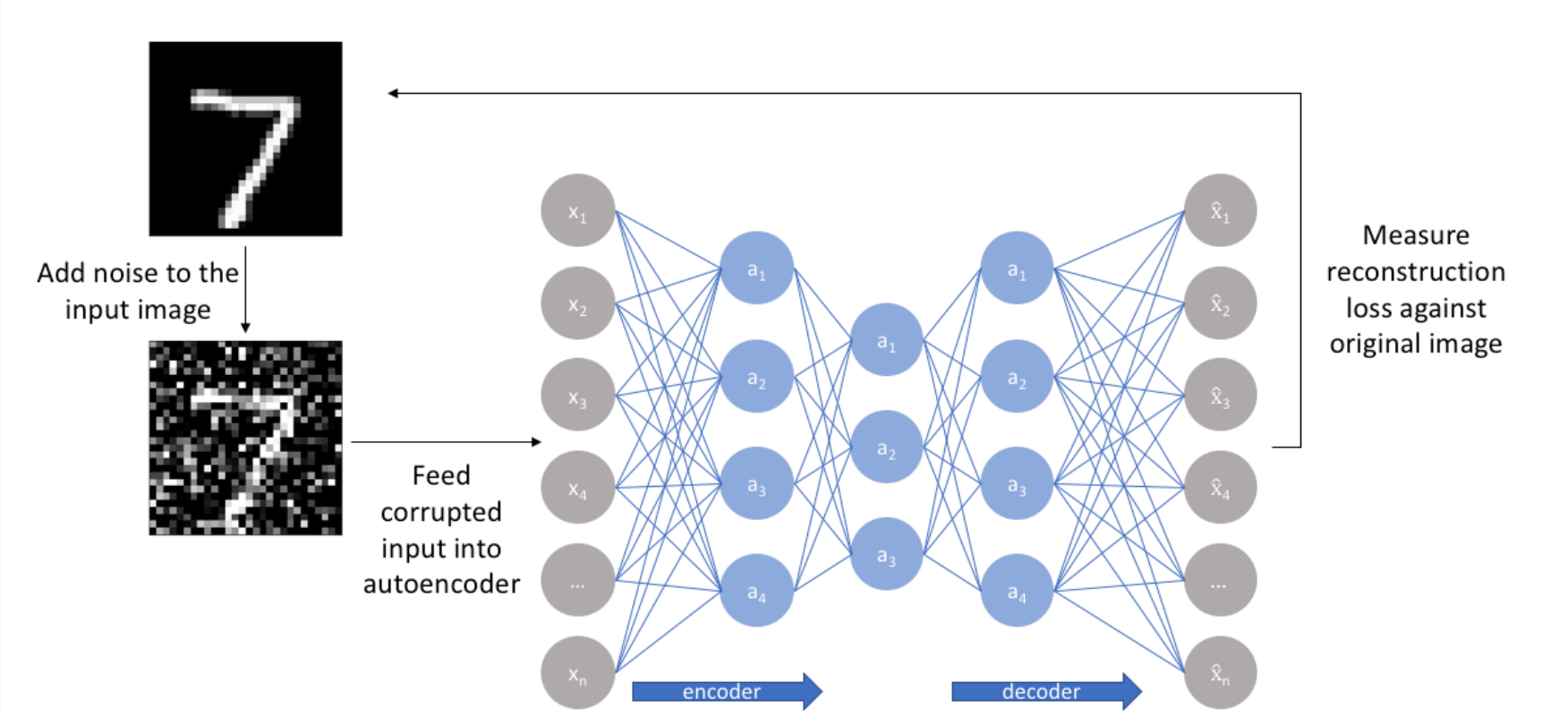
- We **allow our network to sensitize individual hidden layer nodes toward specific attributes of the input data**.
- A sparse autoencoder is selectively activate regions of the network depending on the input data.
- Limiting the network's capacity to memorize the input data without limiting the networks capability to extract features from the data.

$$\mathscr{L}\left(x,\ \hat{x}\right) + \lambda \sum_{i} |\, z_i \,|$$



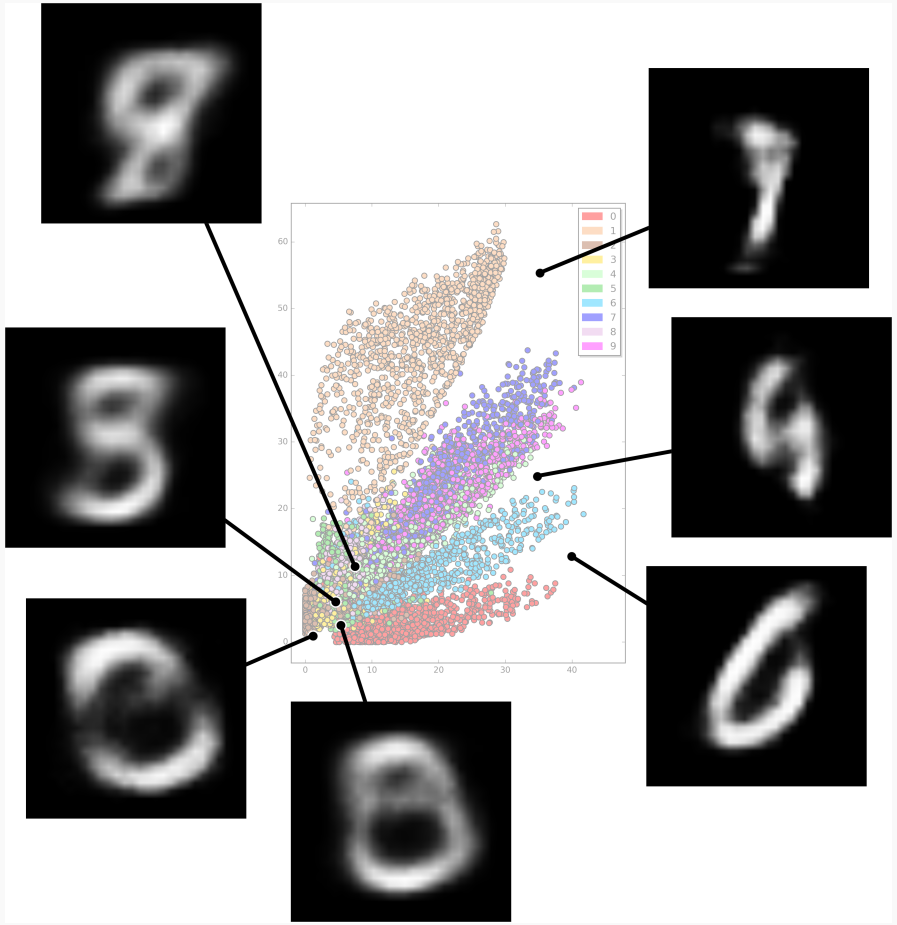
Denoising Autoencoders

Trained with corrupted data points, but to reconstruct original data

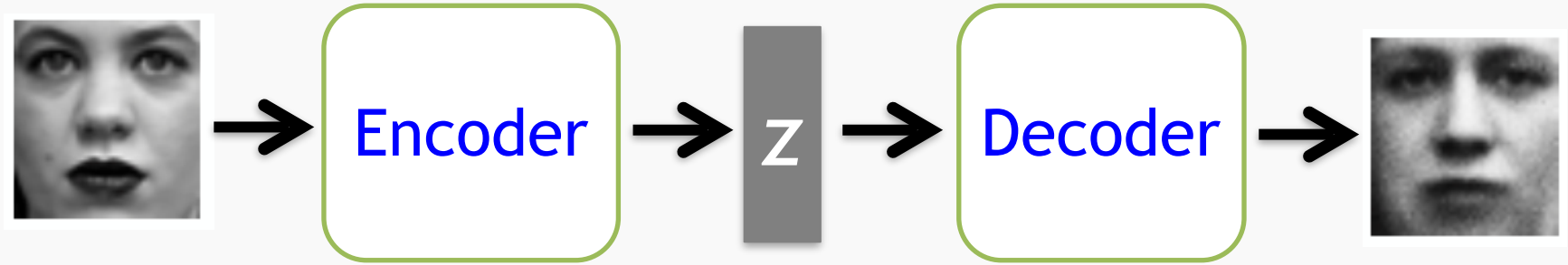


Problems with Autoencoders

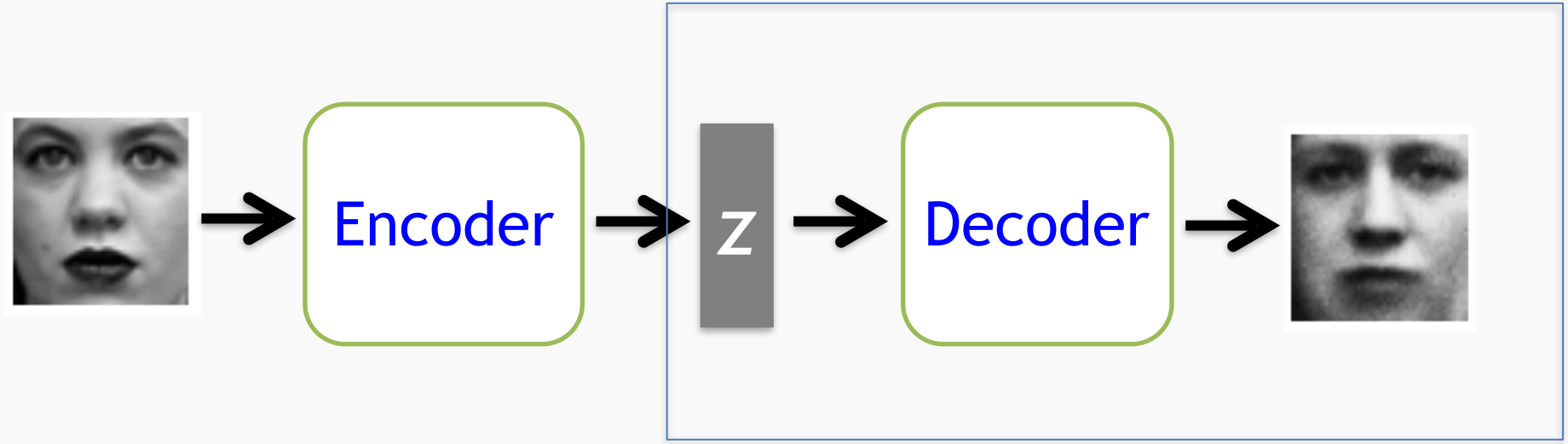
- Gaps in the latent space
- Separability in the latent space
- Discrete latent space



Variational Autoencoders

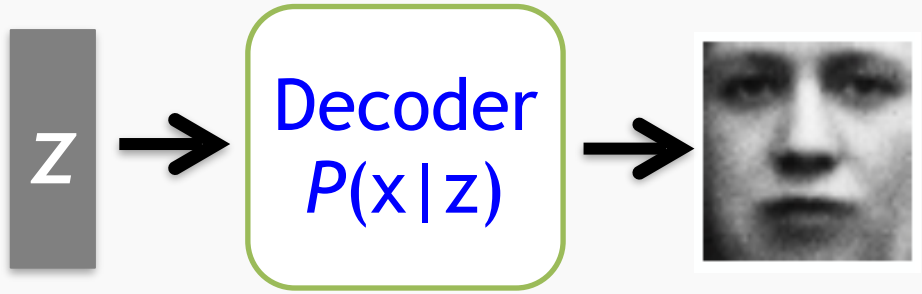


Variational Autoencoders

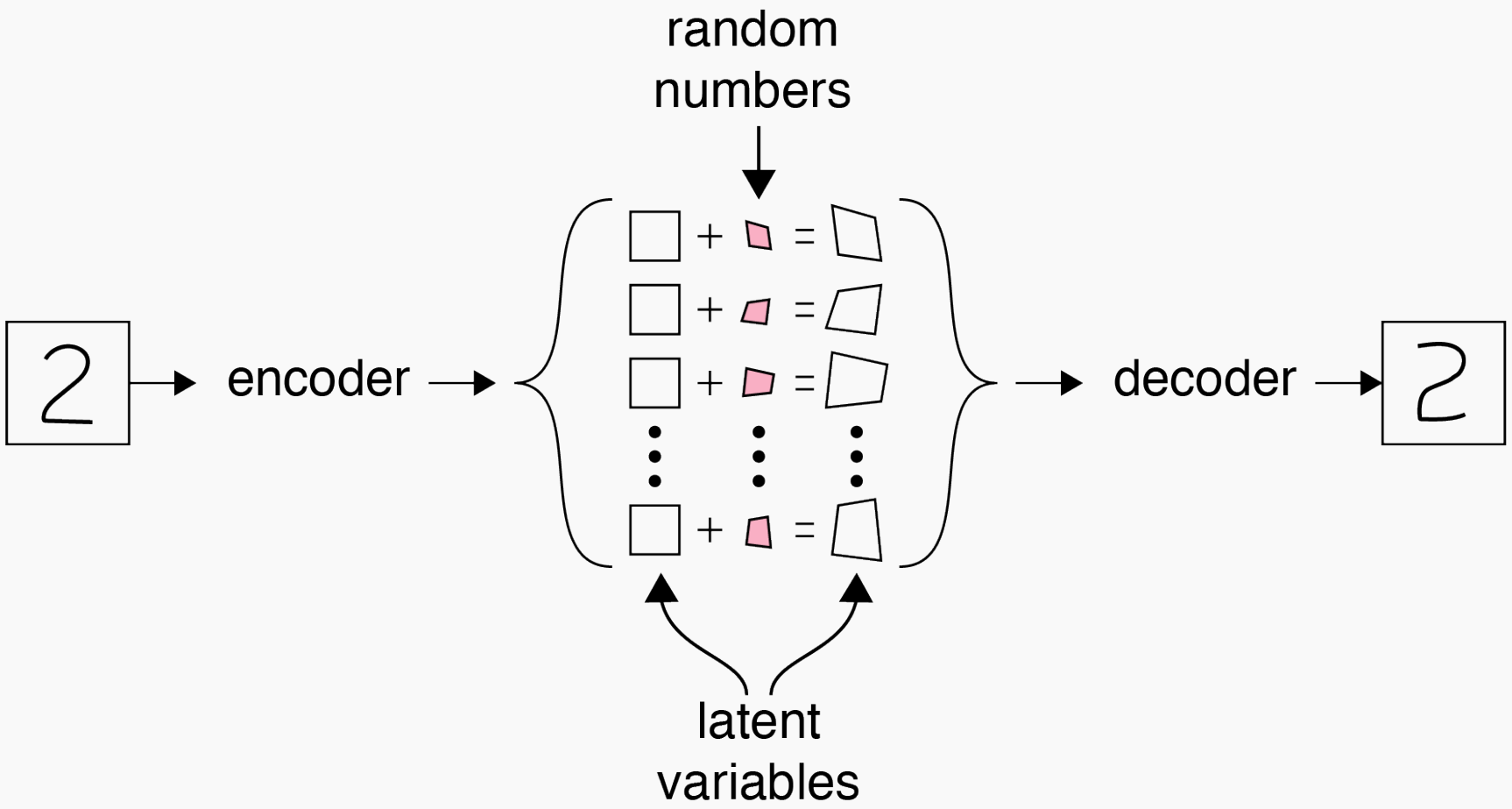


Variational Autoencoders

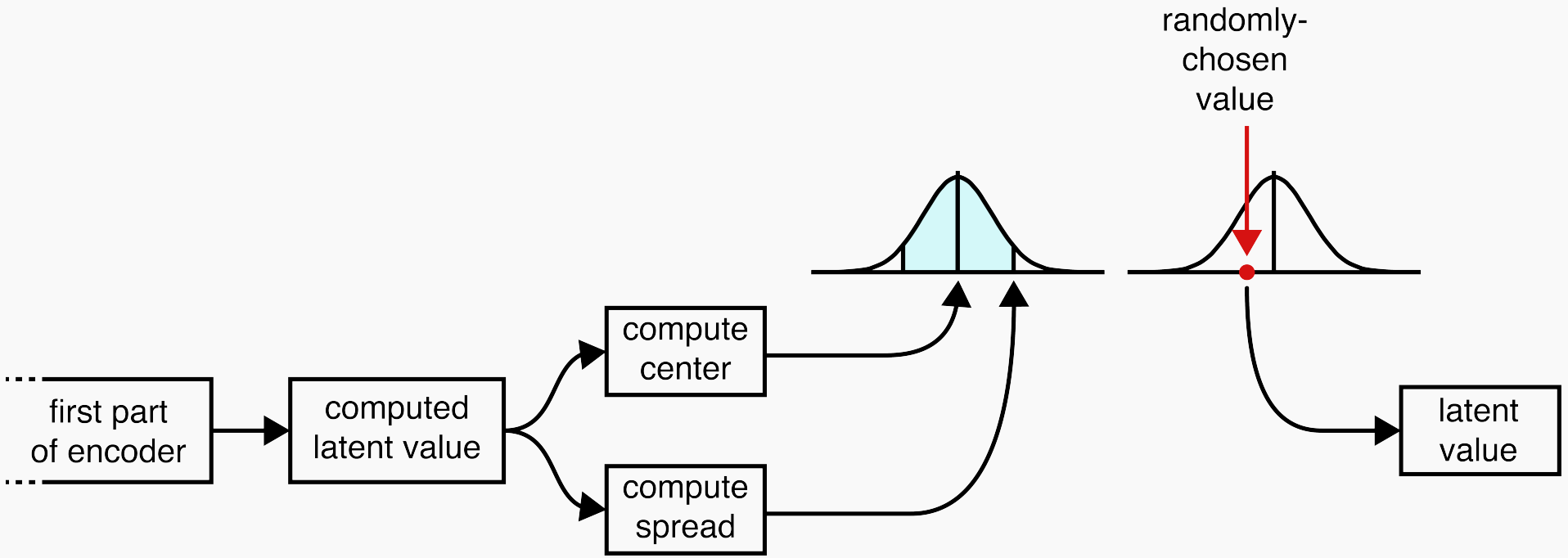
Sample from $P(z)$
Standard Gaussian



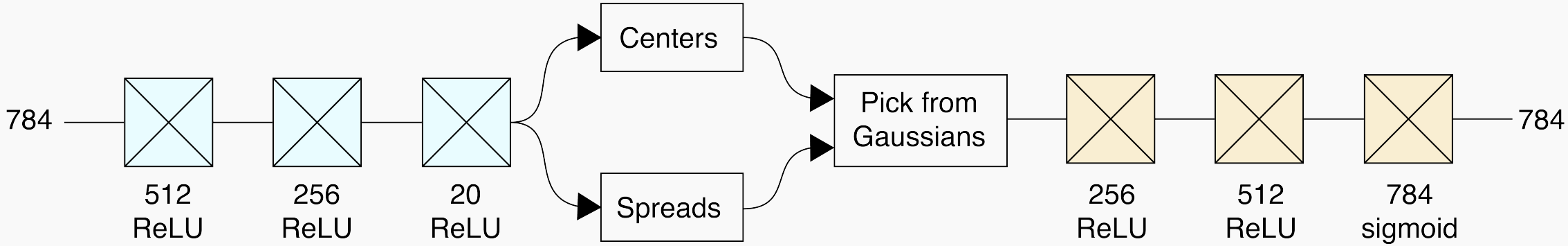
Variational Autoencoders



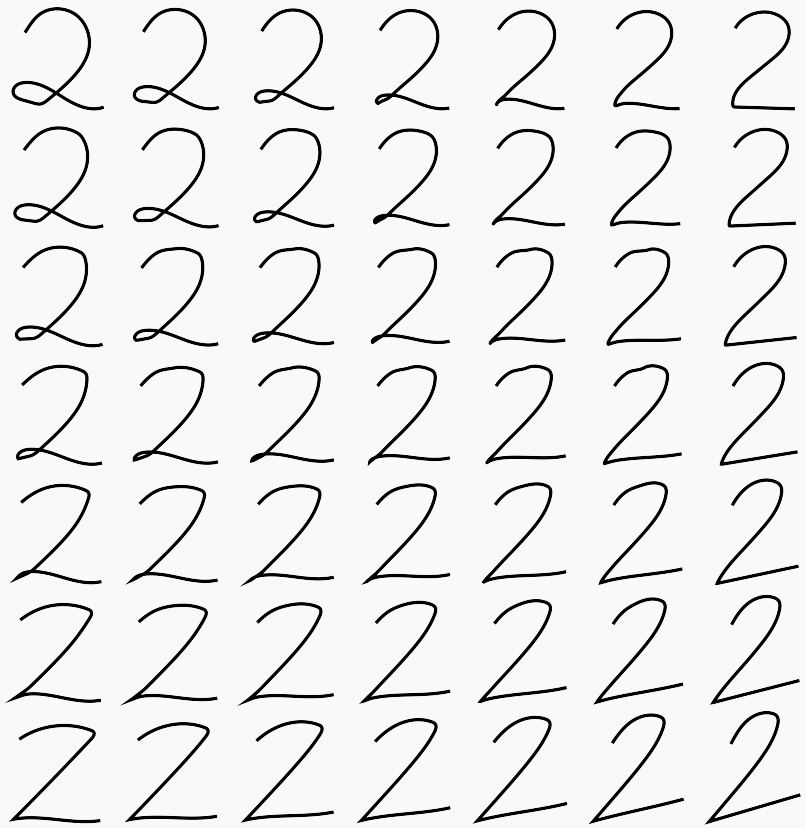
Variational Autoencoders



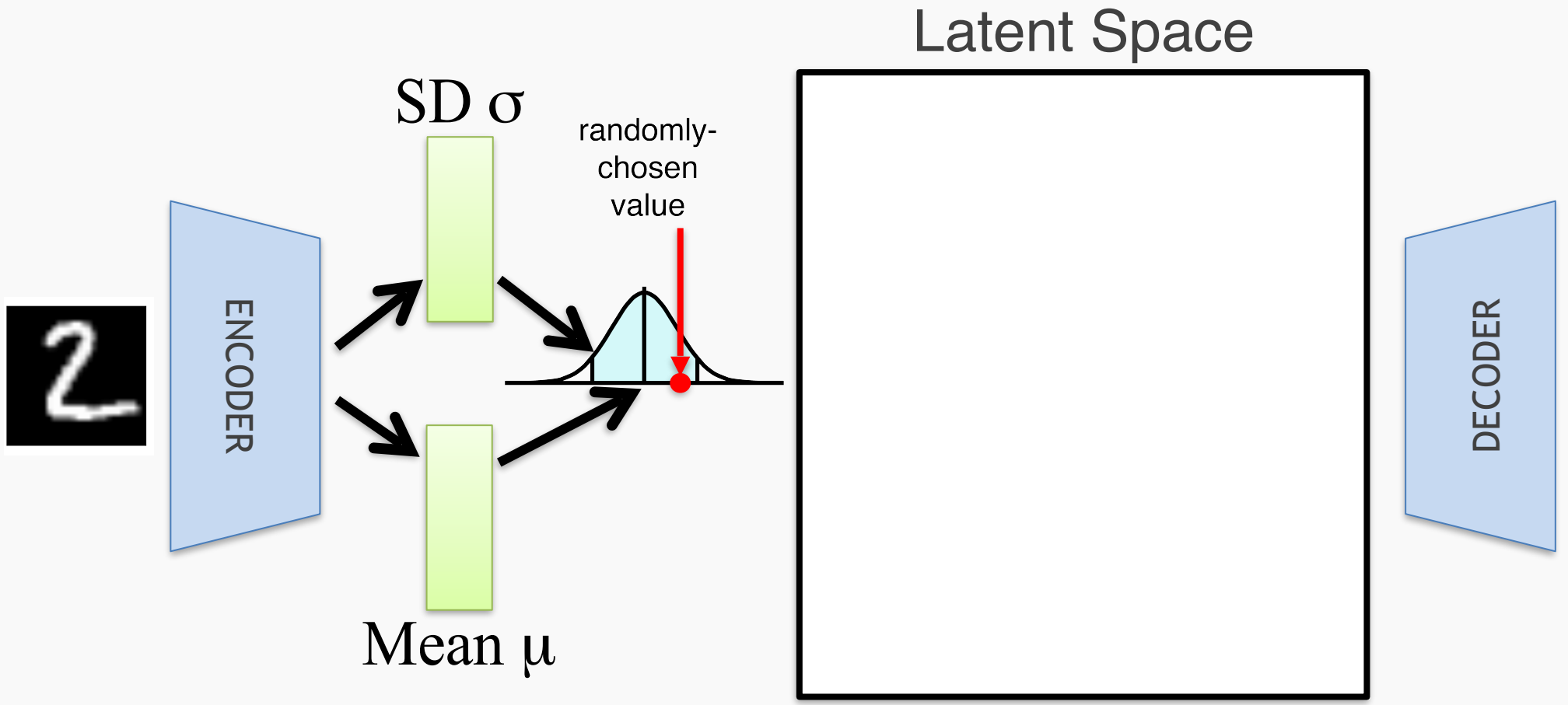
Variational Autoencoders



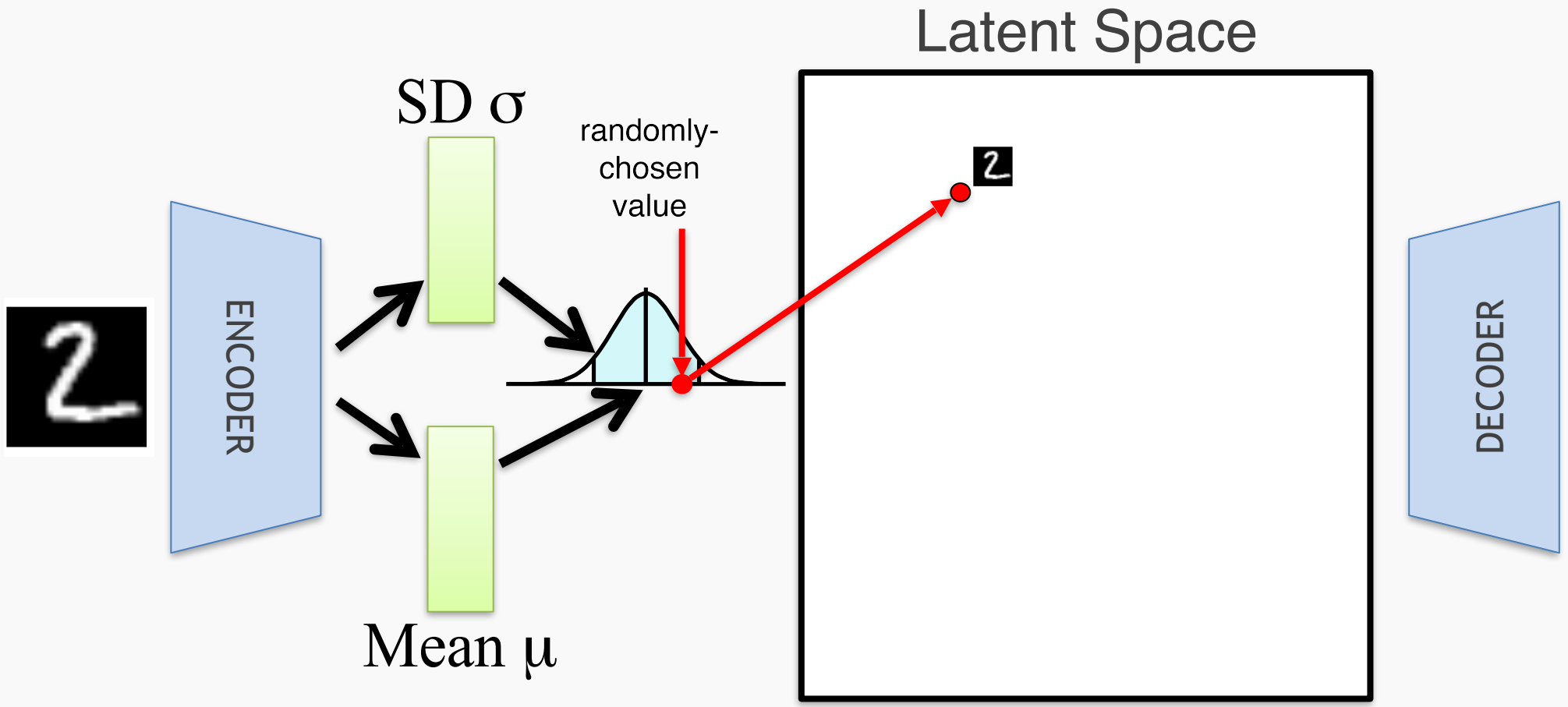
Variational Autoencoders



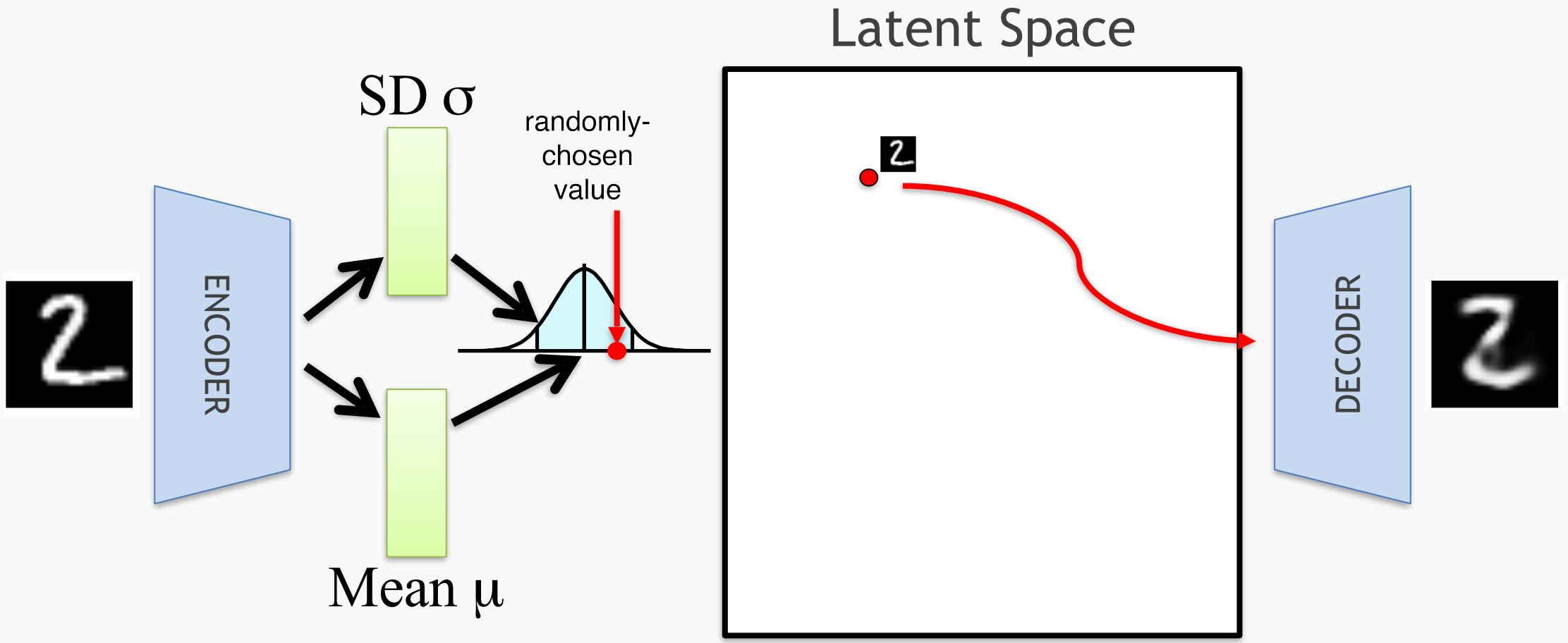
Blending Latent Variables



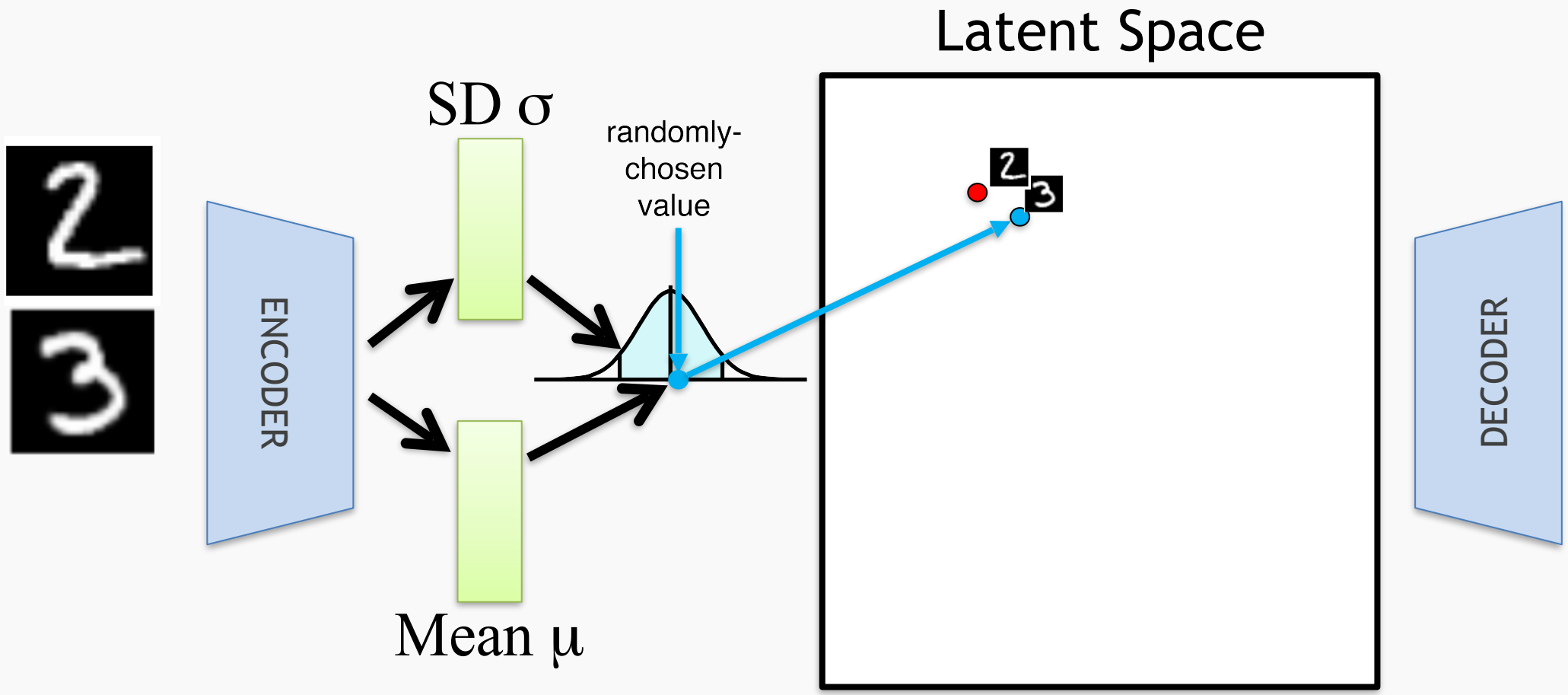
Blending Latent Variables



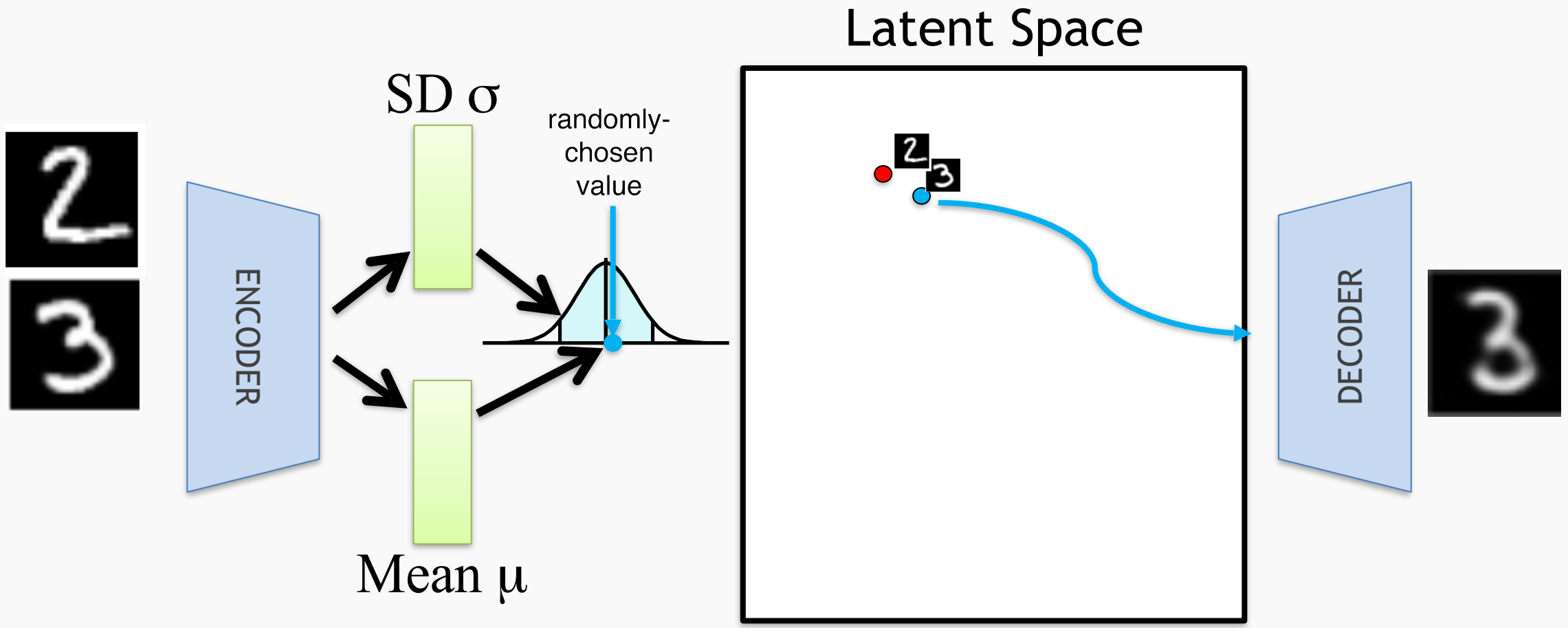
Blending Latent Variables



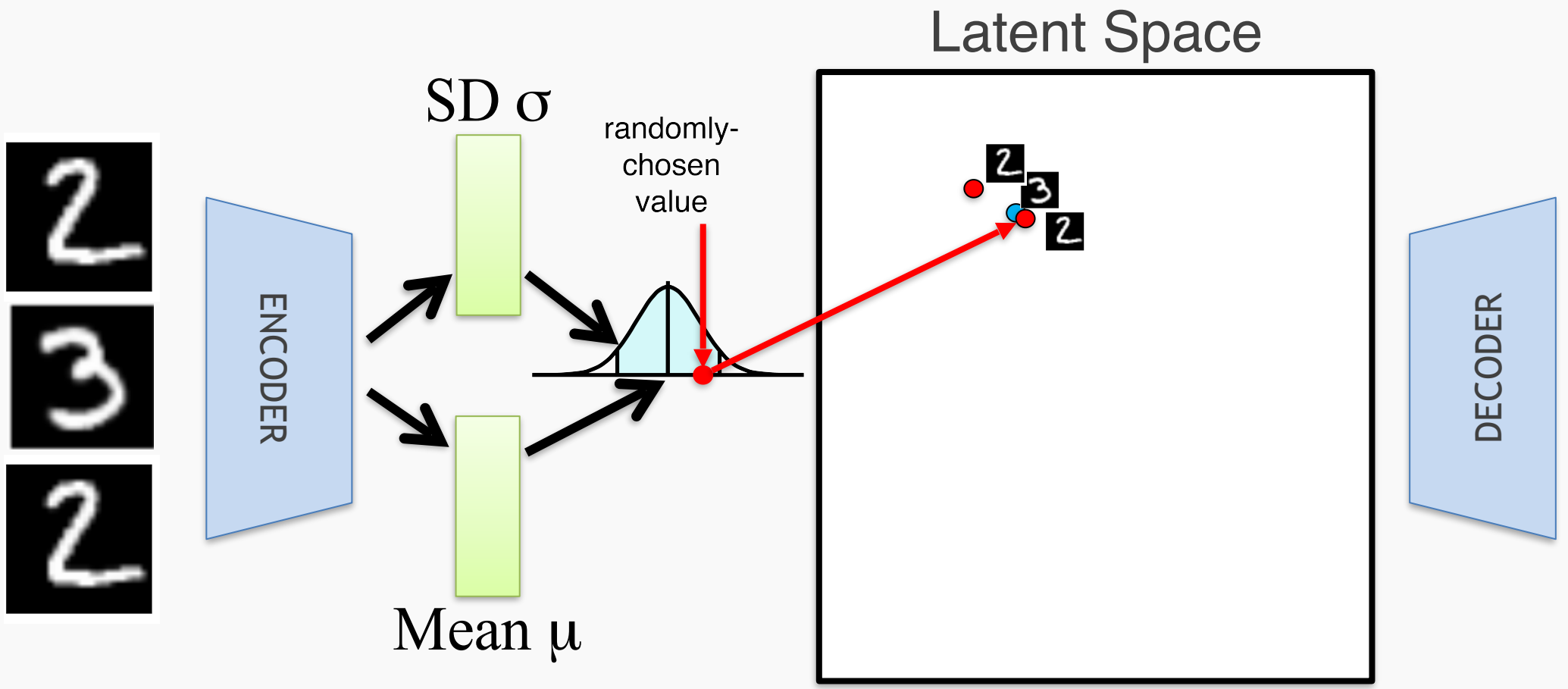
Blending Latent Variables



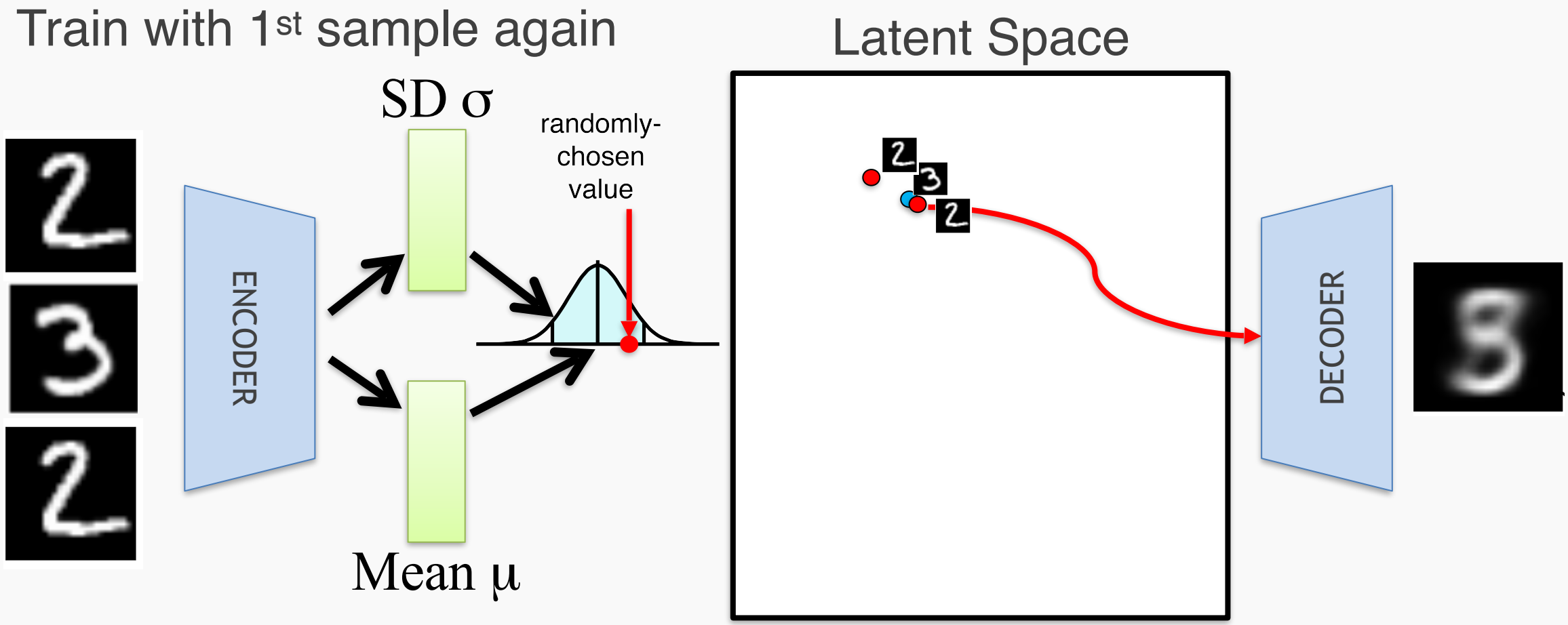
Blending Latent Variables



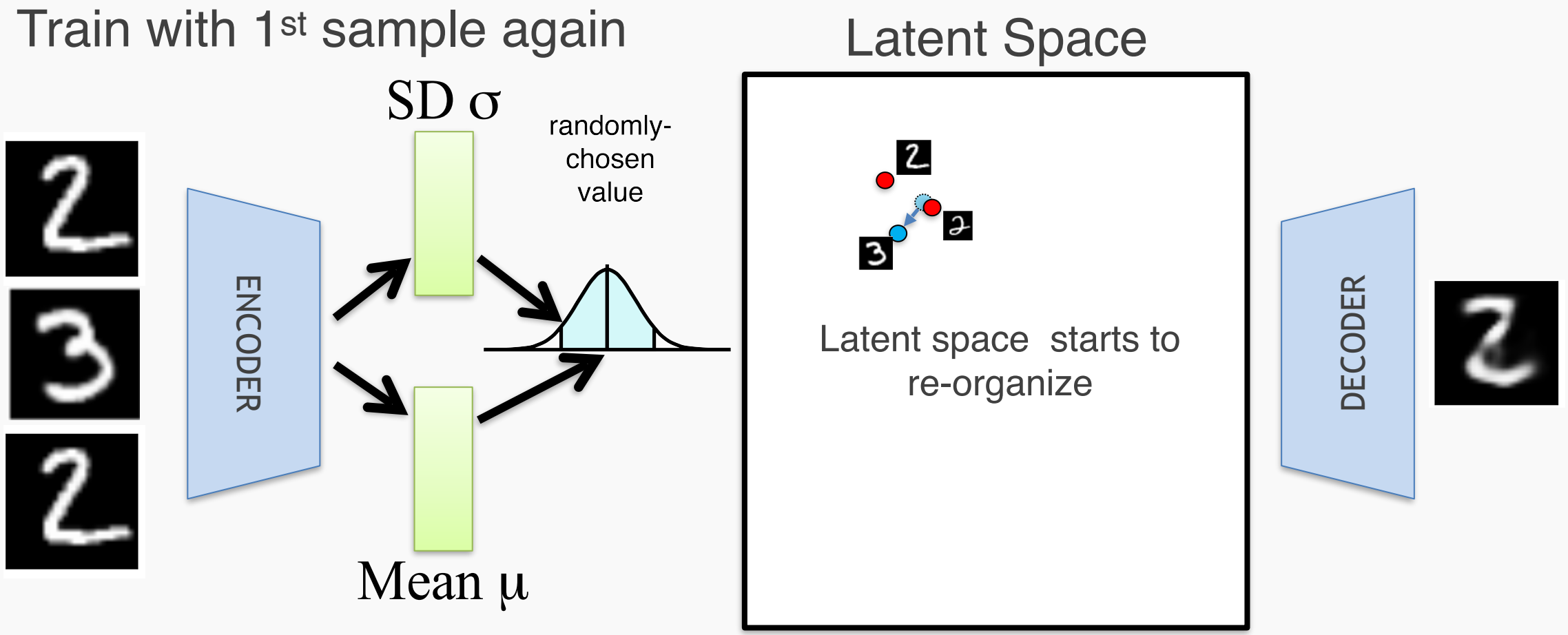
Blending Latent Variables



Blending Latent Variables

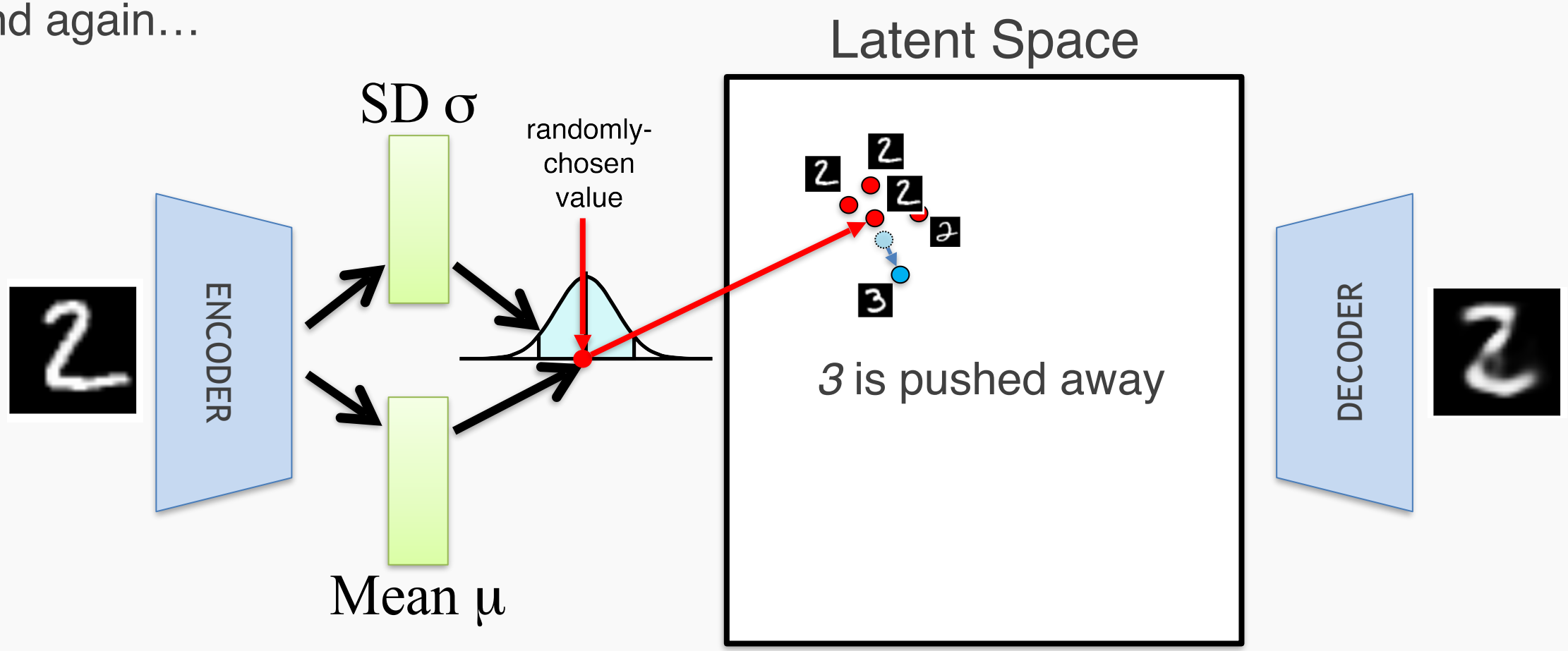


Blending Latent Variables



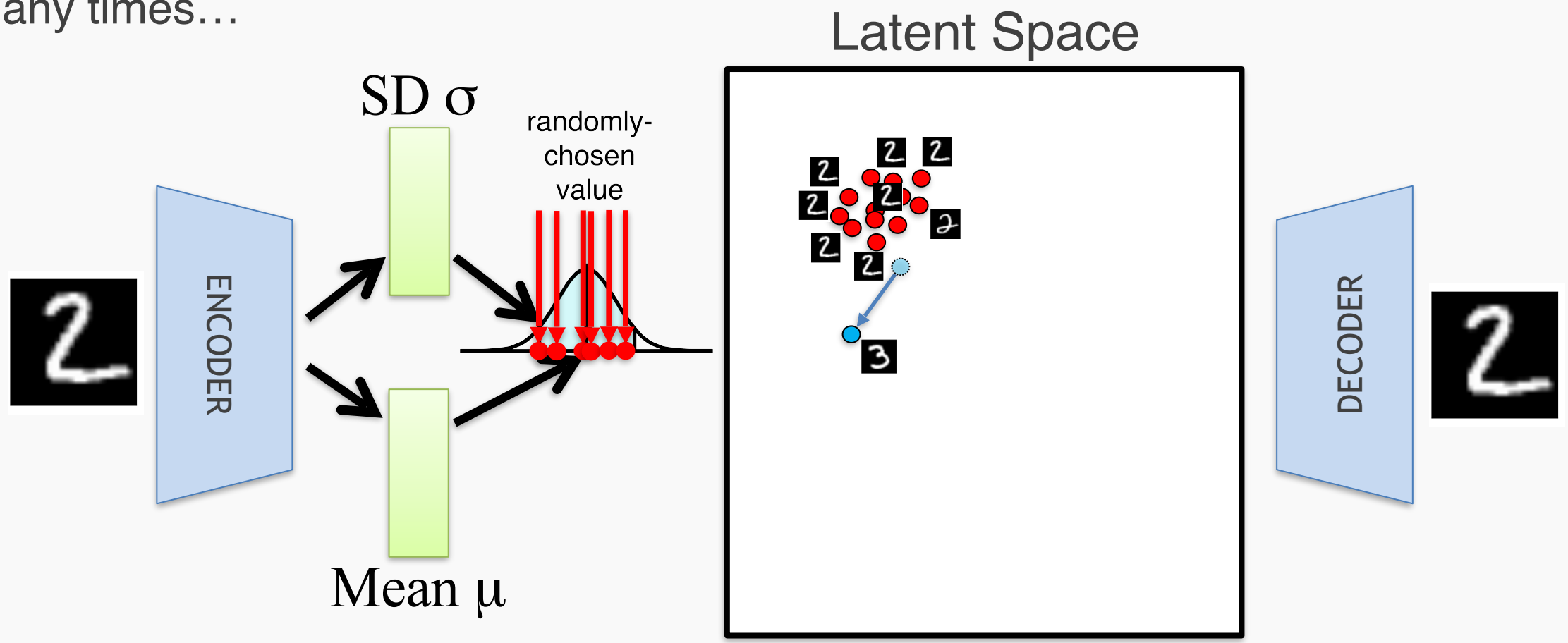
Blending Latent Variables

And again…



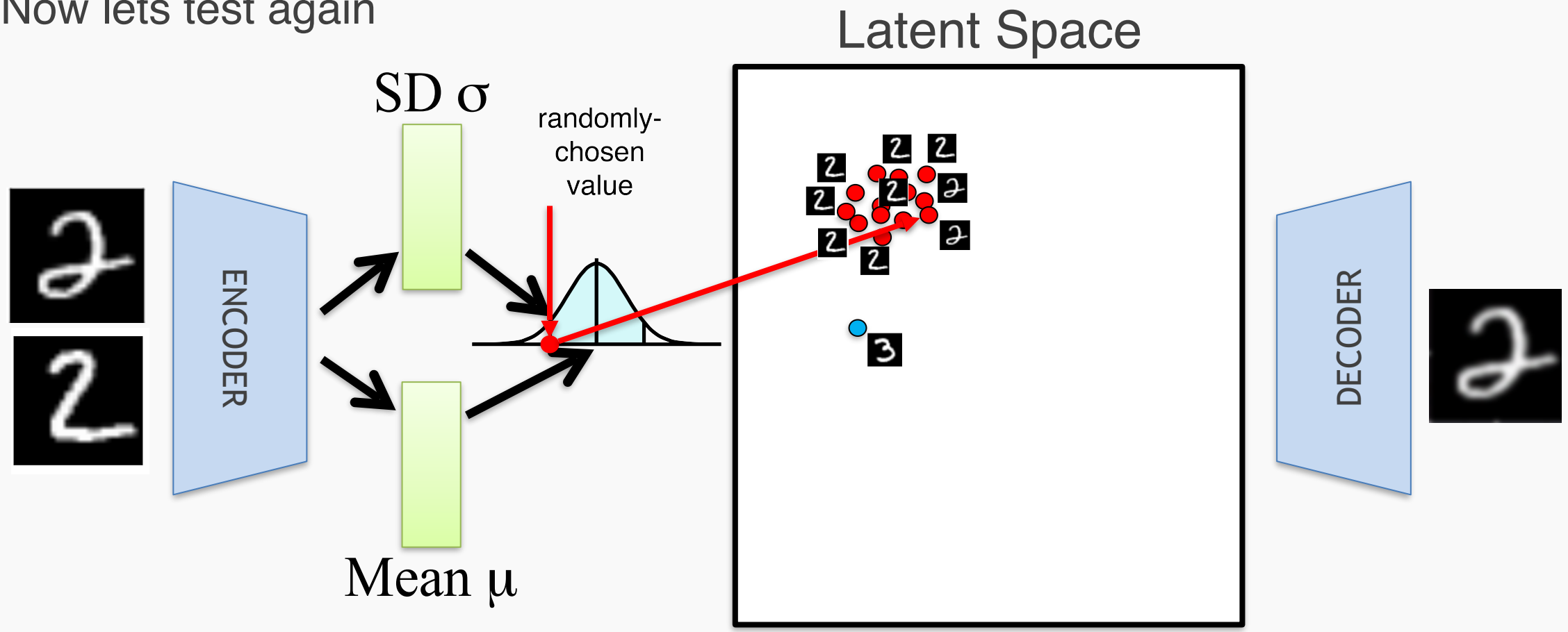
Blending Latent Variables

Many times…

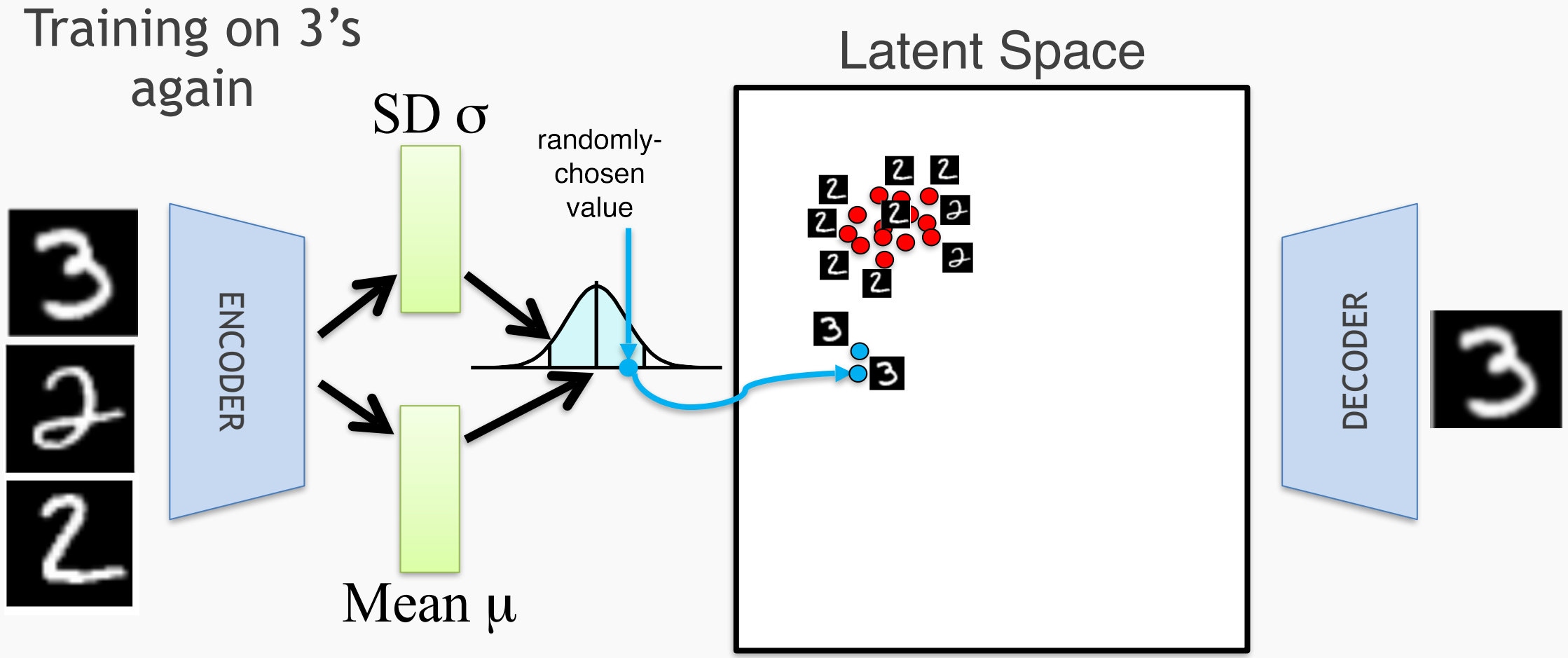


Blending Latent Variables

Now lets test again

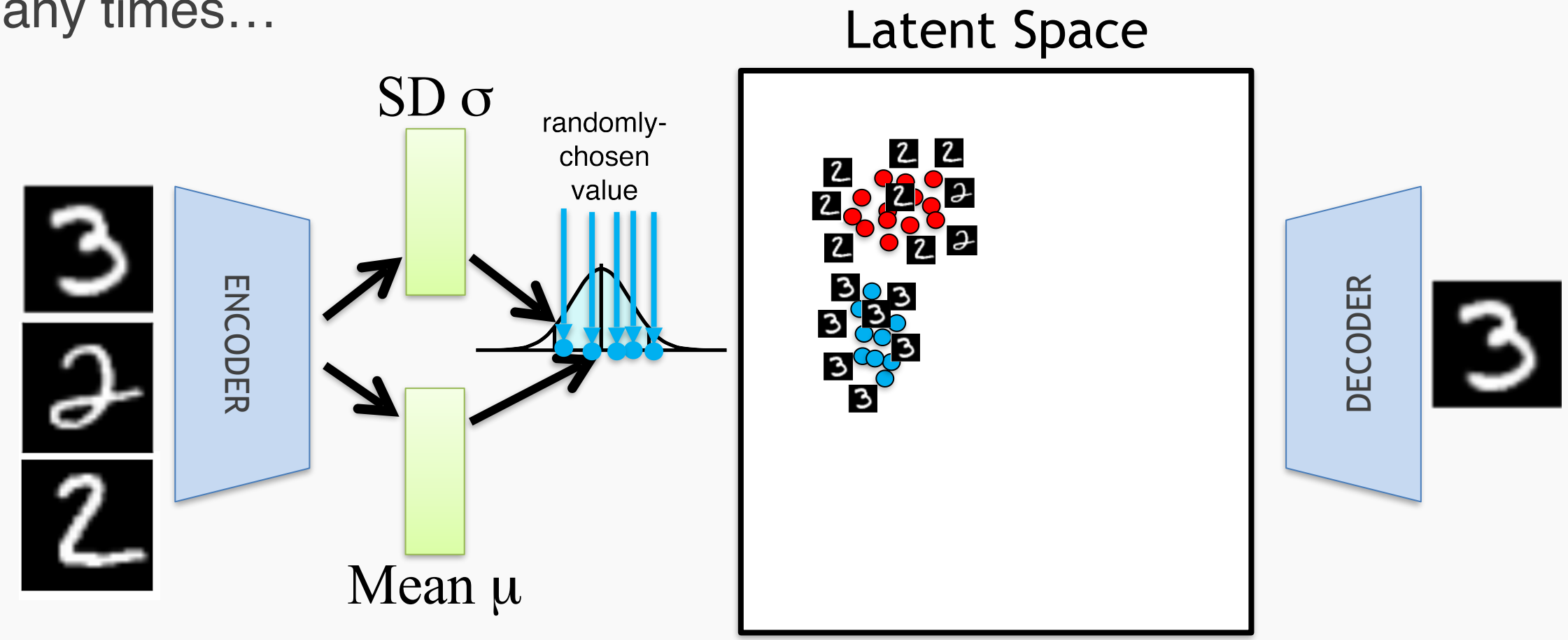


Blending Latent Variables



Blending Latent Variables

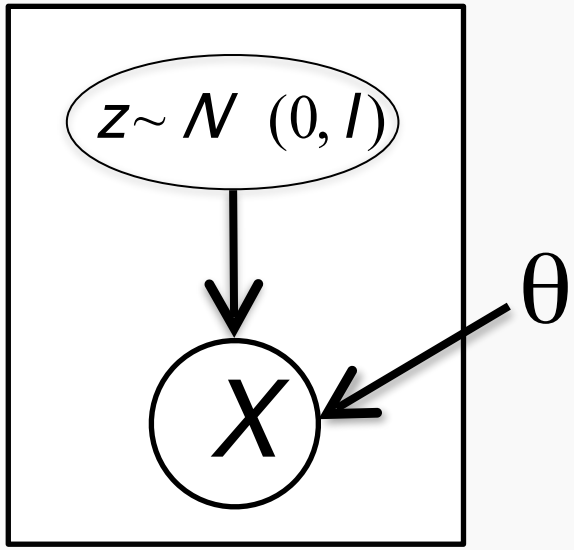
Many times…



VAE Likelihood

Neural network

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



Difficult to approximate in high dim through sampling

For most z values $p(x|z)$ close to 0

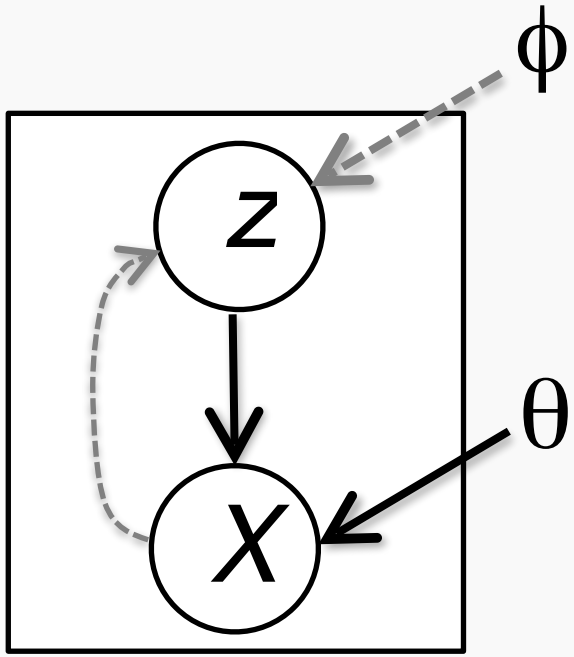
VAE Likelihood

Another neural net

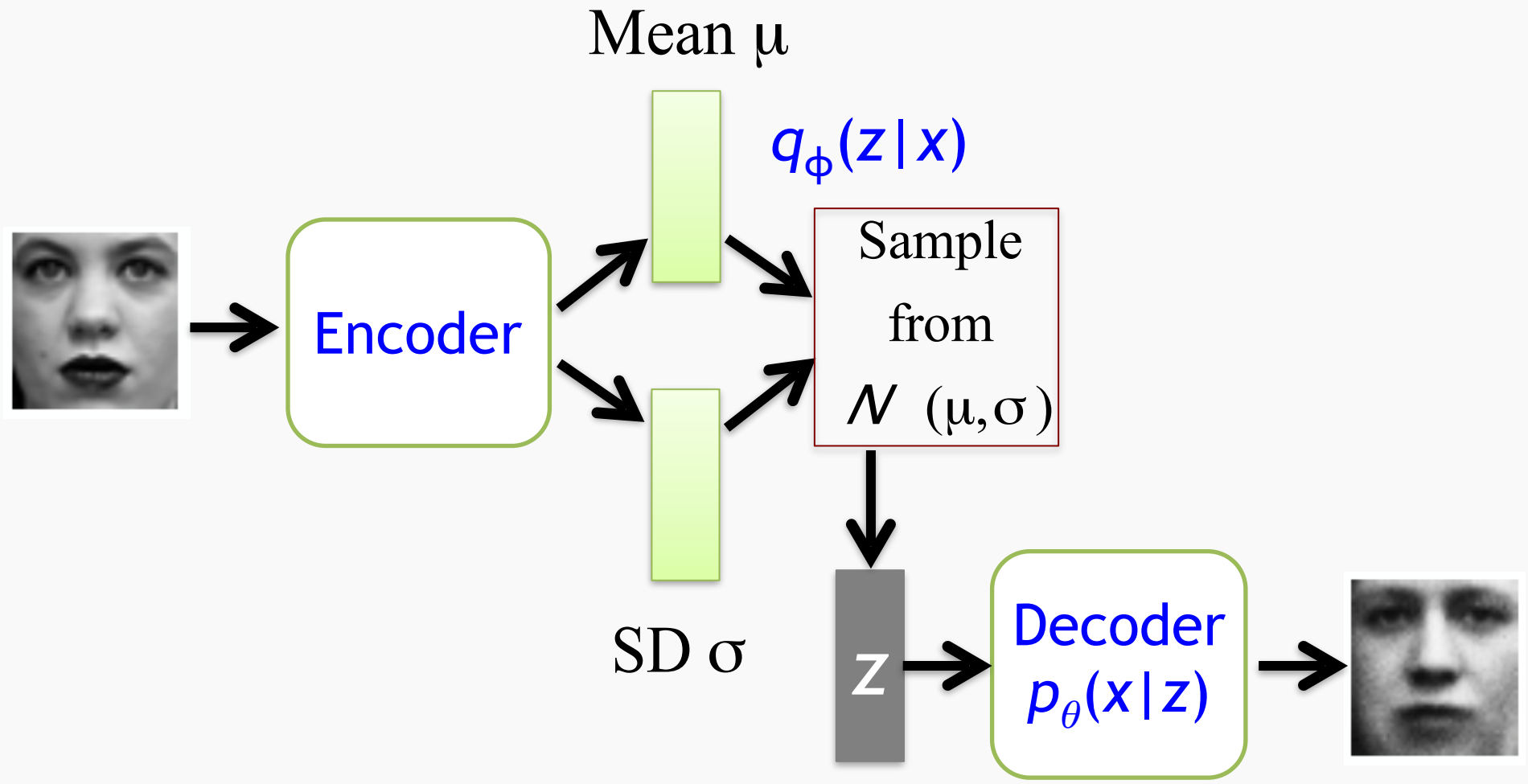
$$p_\theta(x) = \int_z p_\theta(x|z) q_\phi(z|x) dz$$

Proposal distribution:

Likely to produce values of x for which $p(x|z)$ is non-zero



VAE Architecture



VAE Loss

Reconstruction Loss

$$-\mathbf{E}_{z\sim q_\phi(z|x)} \log\left(p_\theta\left(x|z\right)\right)$$

VAE Loss

Reconstruction Loss

Proposal distribution should resemble a Gaussian

$$-\mathbf{E}_{z\sim q_\phi(z|x)}\log\left(p_\theta\left(x|z\right)\right) + KL\left(q_\phi\left(z|x\right)\|p_\theta(z)\right)$$

VAE Loss

Reconstruction Loss

Proposal distribution should resemble a Gaussian

$$-\mathbf{E}_{z\sim q_\phi(z|x)}\log\left(p_\theta\left(x|z\right)\right) + KL\left(q_\phi\left(z|x\right)\|p_\theta(z)\right)$$

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

Variational upper bound on loss we care about!

Training VAE

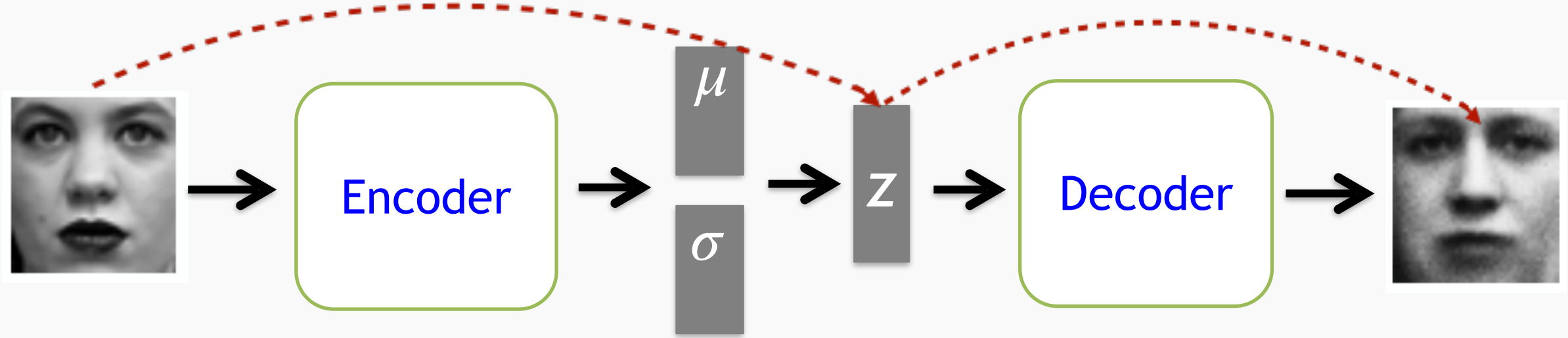
Apply stochastic gradient descent

Sampling step not differentiable

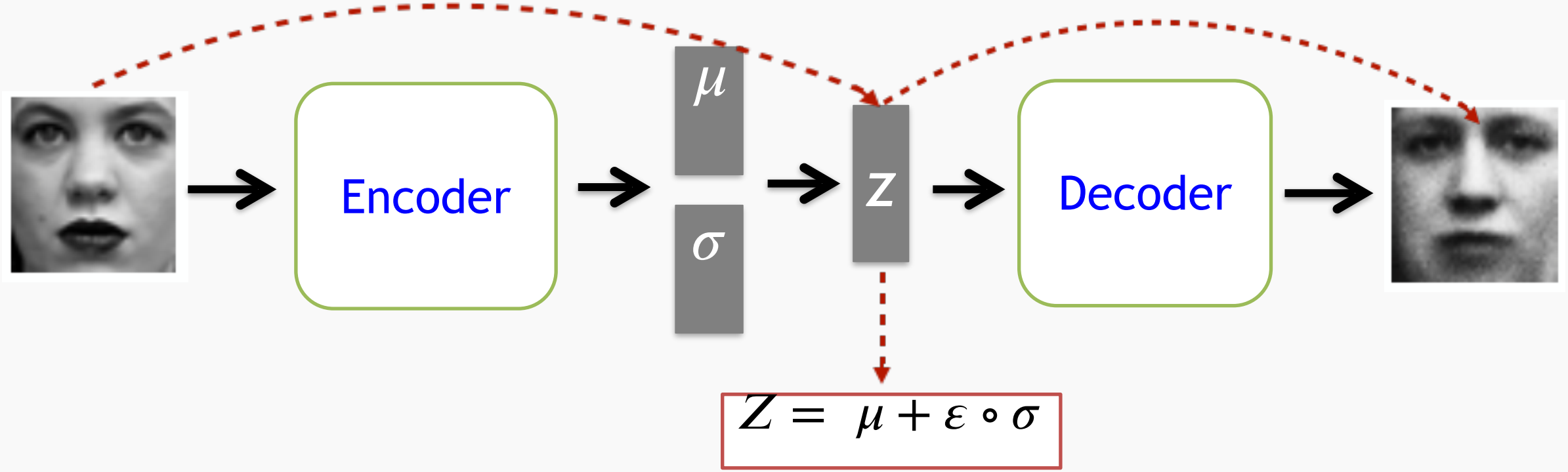
Use a re-parameterization trick

- Move sampling to input layer, so that the sampling step is independent of the model

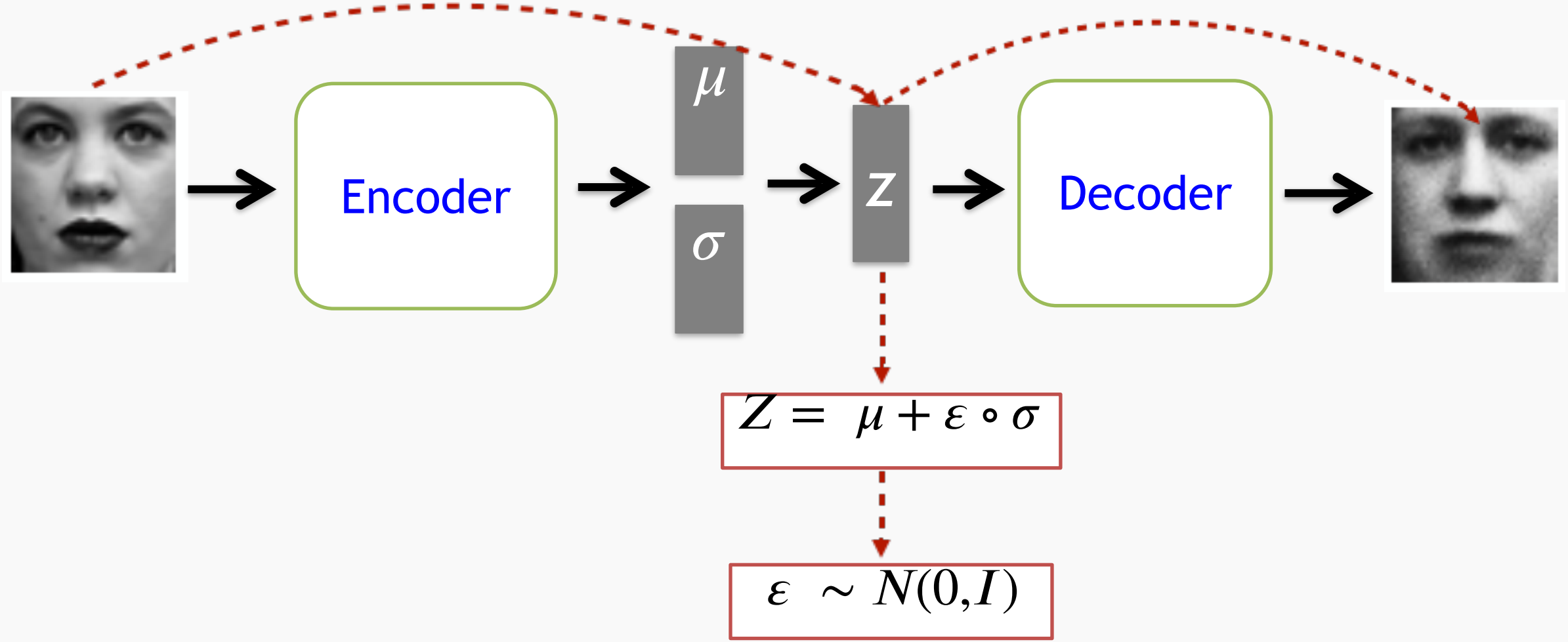
Reparametrization Trick



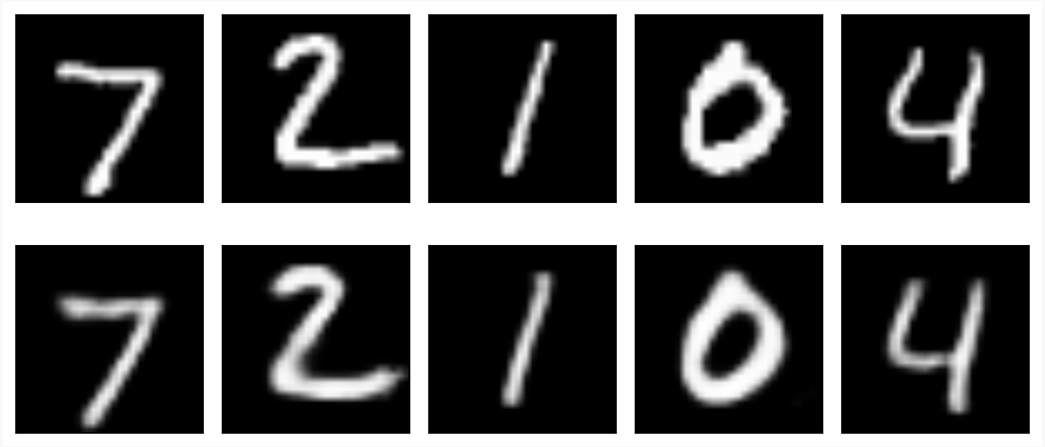
Reparametrization Trick

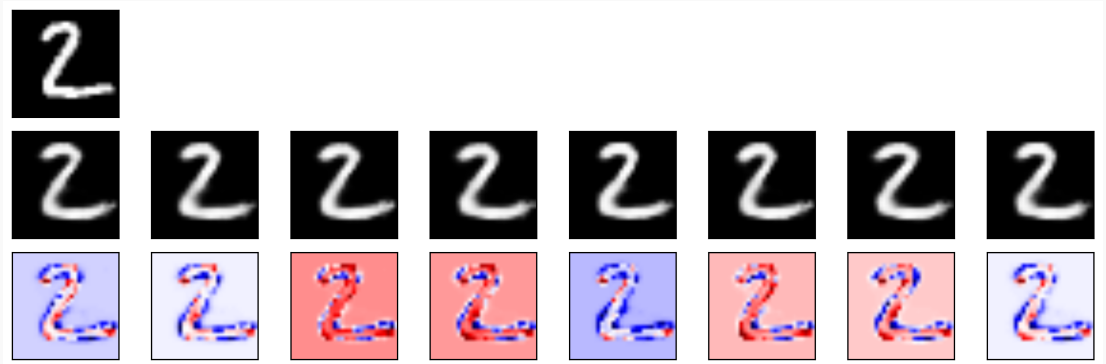


Reparametrization Trick

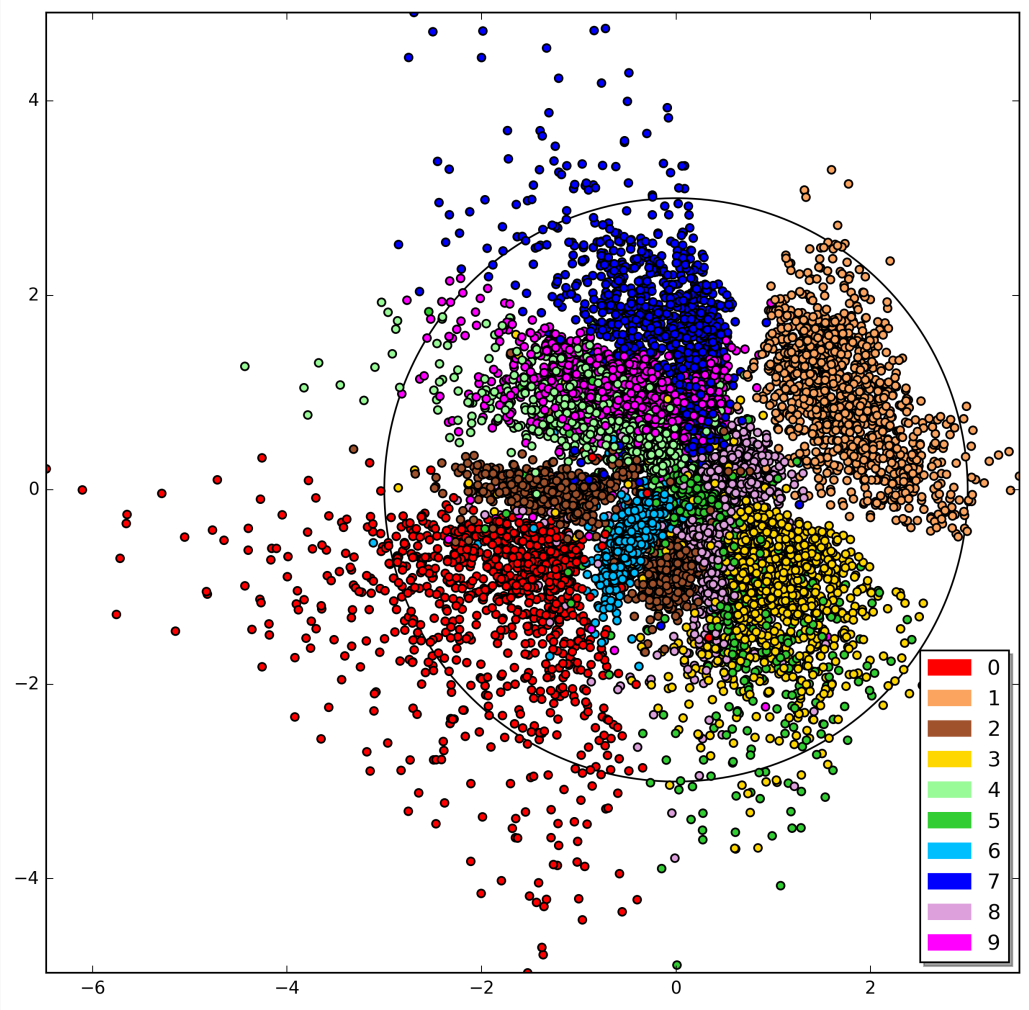


Training VAE





Parameter space VAE



Parameter space VAE