INTRODUCTION TO DEEP LEARNING

PART II – DEEP GENERATIVE MODELS

Auliya Fitri, Sai Vemuri, Sreerag Naveenachandran

Machine Learning Group
Institute of Data Science

DLR

Schedule

Date	Time	Activity
13.11.2025 Day 1	09:00 - 10:00	Introduction and basics
	10:00 - 10:30	Hands-on I
	10:30 - 10:45	Coffee break
	10:45 - 11:45	Advanced concept and Convolutional Neural Network
	11:45 - 12:15	Hands-on II
	12:15 - 12:30	Recap Day 1
14.11.2025 Day 2	09:00 - 10:00	Deep Generative Models
	10:00 - 10:30	Hands-on III
	10:30 - 10:45	Coffee break
	10:45 - 11:45	Transformers, LLM, and other interesting architectures
	11:45 - 12:15	Hands-on IV
	12:15 - 12:30	Code and knowledge sources + closing

We're here!

Curriculum

- Unsupervised learning
- Generative modeling
- Autoencoder (AE)
- Variational Autoencoder (VAE)
- Generative adversarial networks (GANs)
- Diffusion models

*some slides are adopted from MIT Deep learning course

Unsupervised learning

Unsupervised learning
How much information does the machine need to predict?

- The number of samples required to train a large learning machine (for any task) depends on the amount of information that we ask it to predict.
  - The more you ask of the machine, the larger it can be.
- “The brain has about 10^14 synapses and we only live for about 10^9 seconds. So we have a lot more parameters than data. This motivates the idea that we must do a lot of unsupervised learning since the perceptual input (including proprioception) is the only place we can get 10^5 dimensions of constraint per second.”
  - Geoffrey Hinton (in his 2014 AMA on Reddit)
  - (but he has been saying that since the late 1970s)

*Yann Lecunn's slides keynote NIPS 2016

Unsupervised learning
How much information does the machine need to predict?

- Predicting human-provided labels is not enough
- Predicting a value function is not enough
- Instead we have to learn representations of data like images, sentences, etc.

*Yann Lecunn's slides keynote NIPS 2016

Unsupervised learning
How much information does the machine need to predict?

- "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

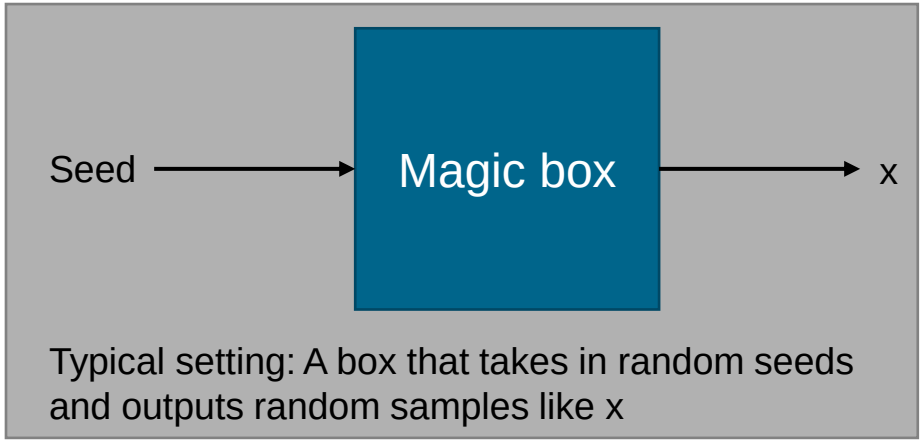
*Yann Lecunn's slides keynote NIPS 2016

Unsupervised learning

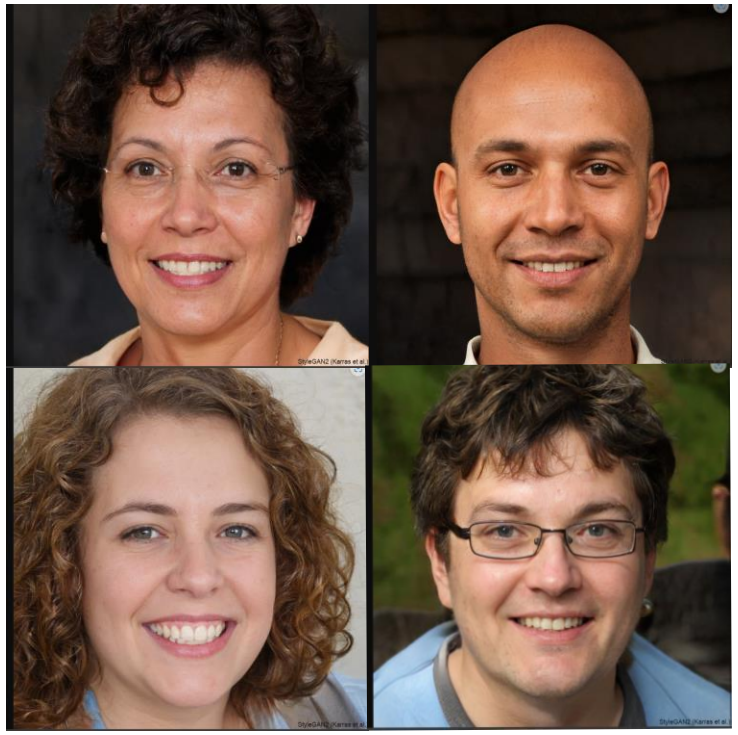
- Supervised learning
  - Data: (x,y)
  - x is data, y is label
  - Goal: Learn a function to map
    x → y
  - Examples: Classification, regression, object detection, semantic segmentation etc.

- Unsupervised learning
  - Data: x
  - x is data, no labels!
  - Goal: Learn the hidden or underlying structure of the data
  - Examples: Clustering, feature or dimensionality reduction, generative modeling etc.

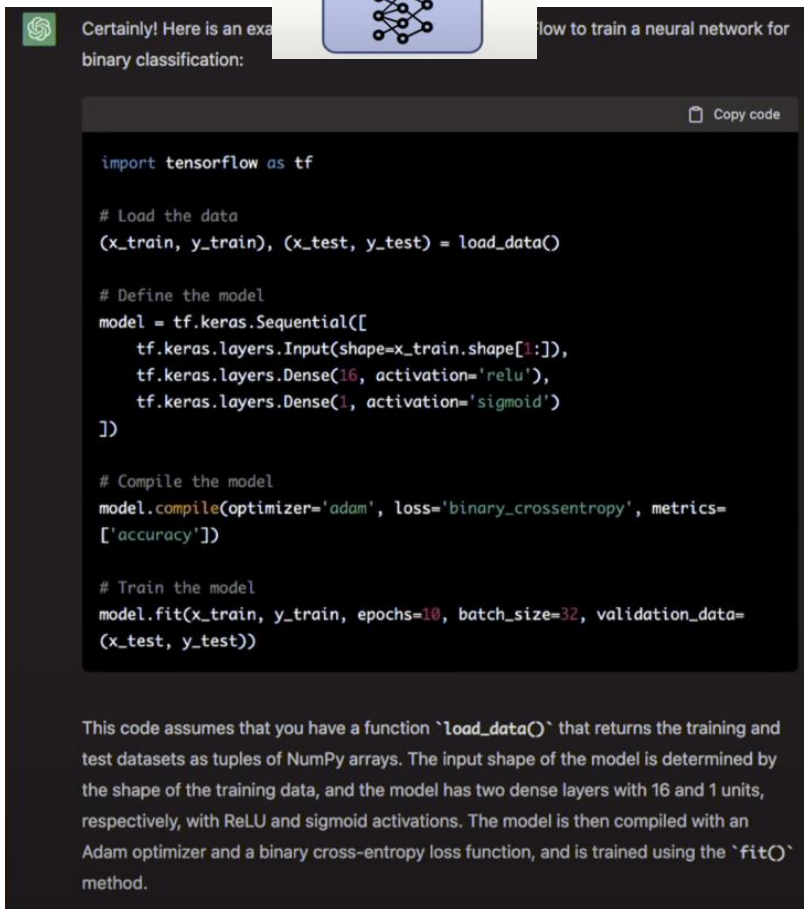
Generative modeling



Generative modeling
Text-to-image / Text-to-Text / Image generation

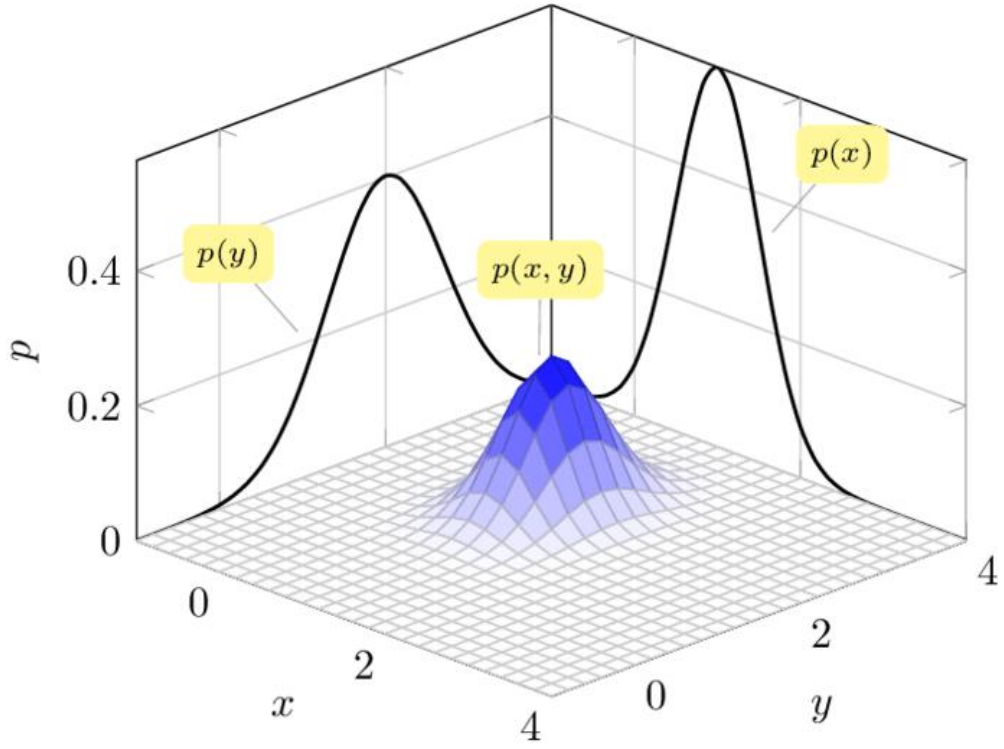


*Goodfellow et.al, This person doesn't exist



Generative modeling
Data distribution

- Goal: Take as input training samples from some distribution and learn a model that represents that distribution



Input samples

Training data ~ $P_{data}(x)$

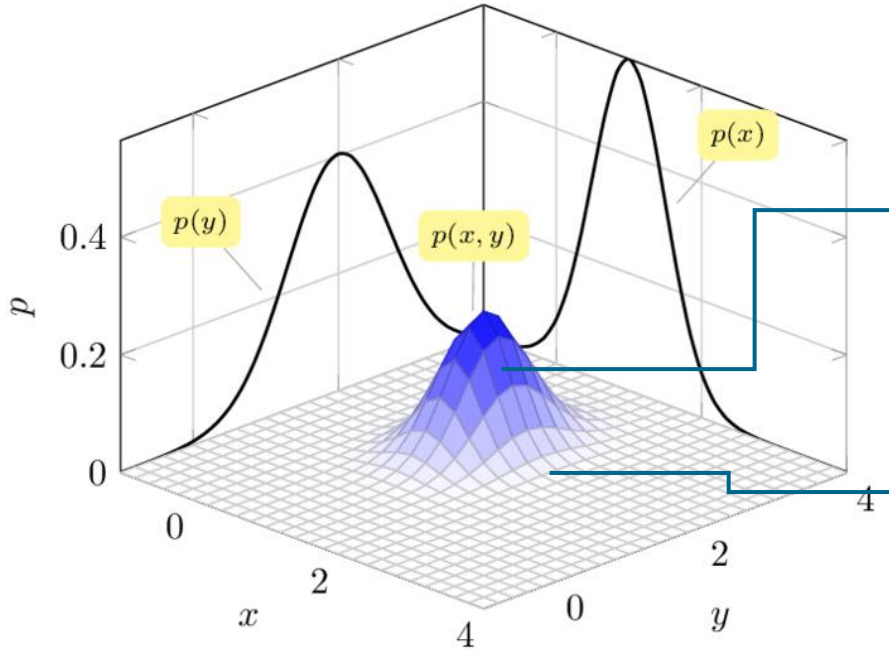
Generated samples

Generated ~ $P_{model}(x)$

How can we learn $P_{data}(x)$ similar to $P_{model}(x)$

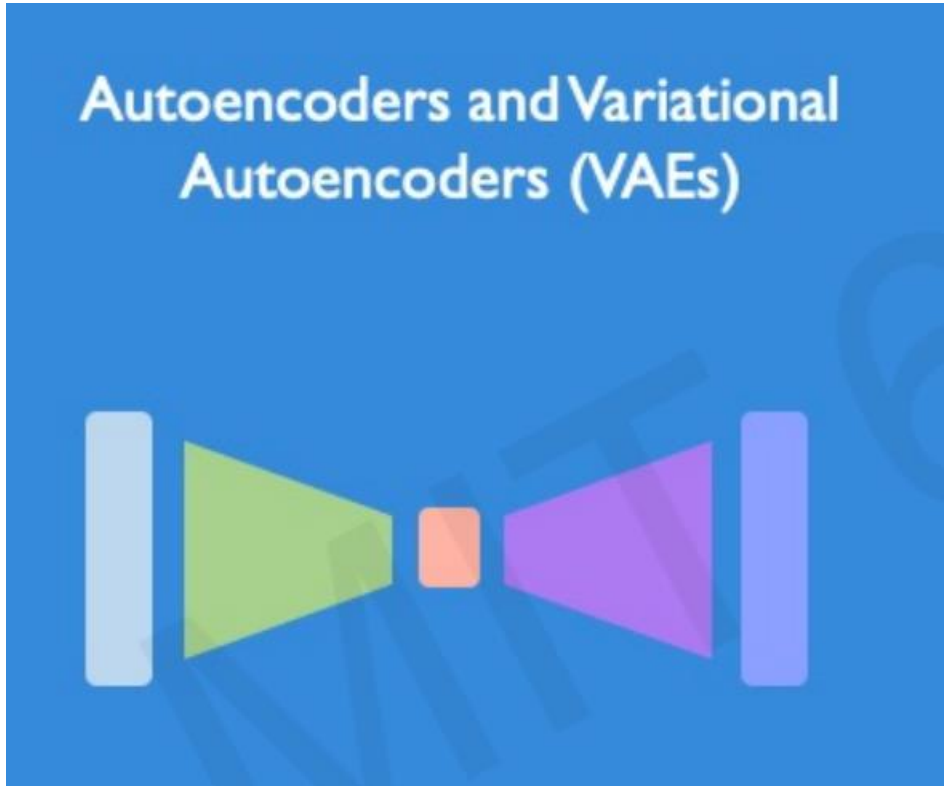
Generative modeling
Application: Outlier detection

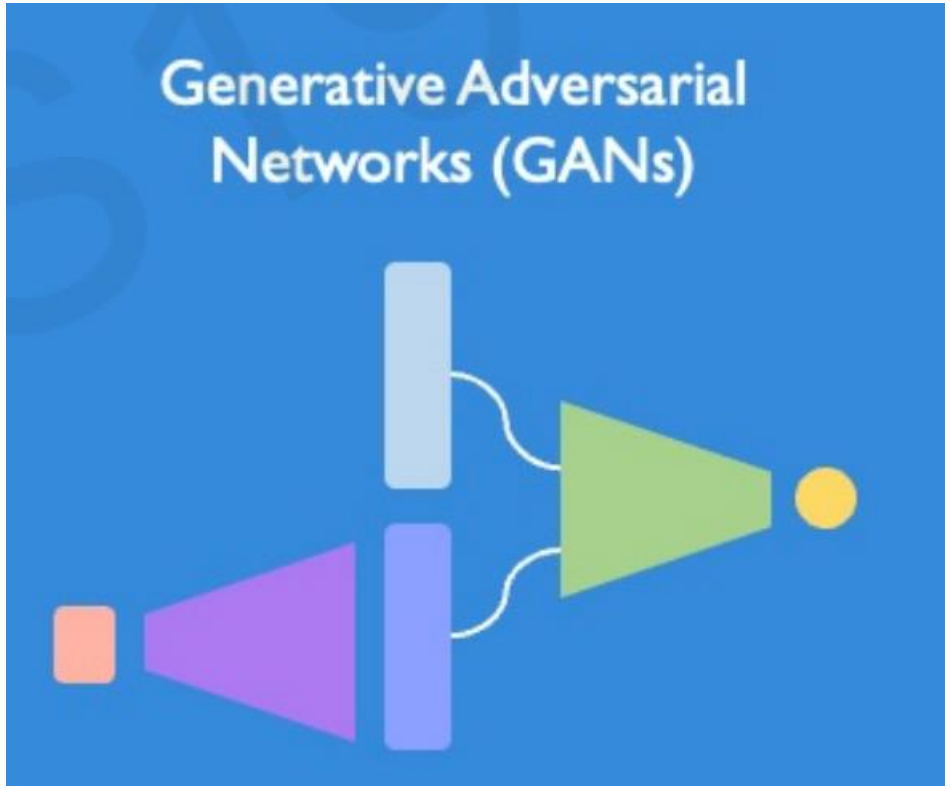
- Problem: How can we detect when we encounter something new or rate?
- Strategy: Leverage generative models, detect outliers in the distribution
- Use outliers during training to improve even more



95% Training data
(1) Sunny, (2) Highway, (3) Straight road

Generative modeling
Latent variables





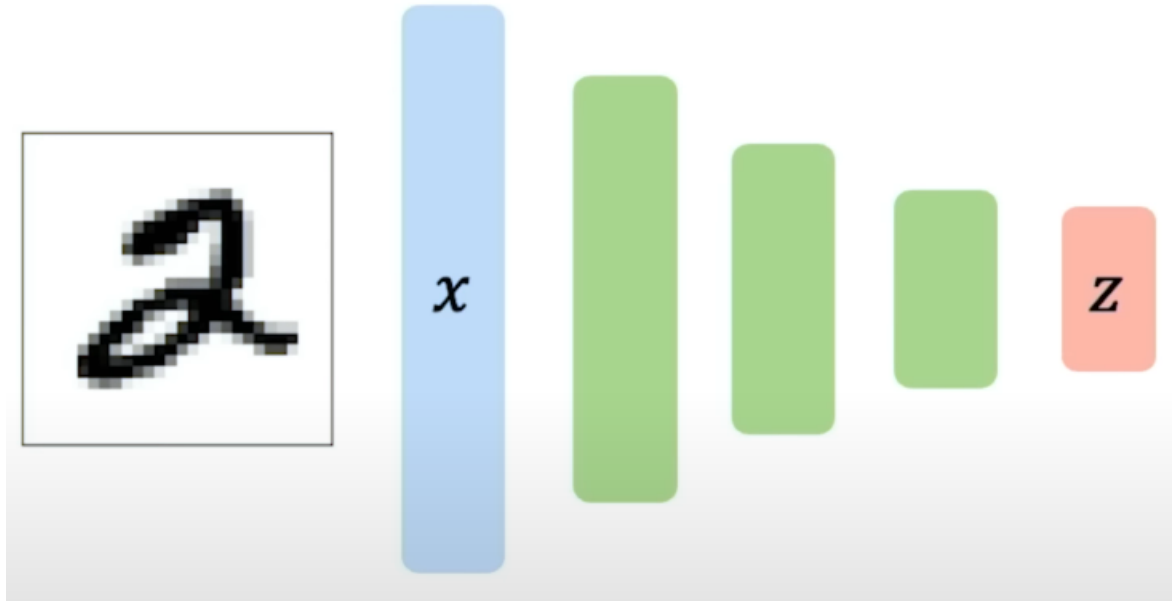
Generative modeling
Latent variables

Can we learn the true explanatory factors, e.g. Latent variables, from only observed data?

Autoencoder

Autoencoder
Latent variables

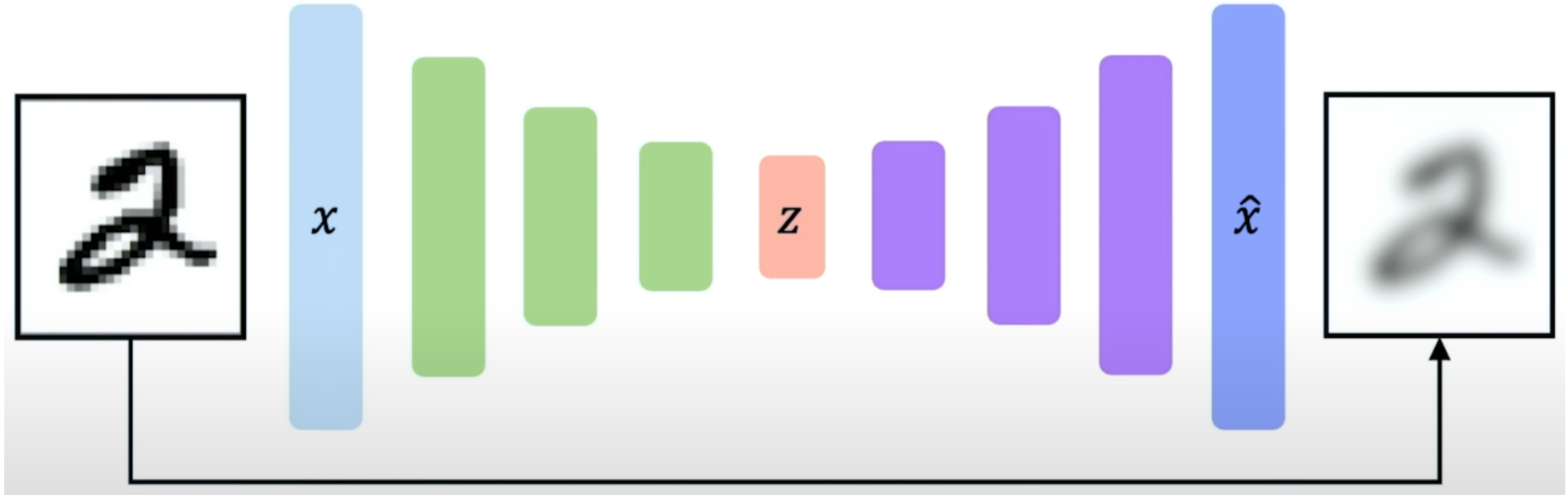
- Unsupervised approach for learning a lower dimensional representation from unlabeled training data



- “Encoder” learns mapping from the data, x, to a low dimensional latent space, z

Autoencoder
Data reconstruction

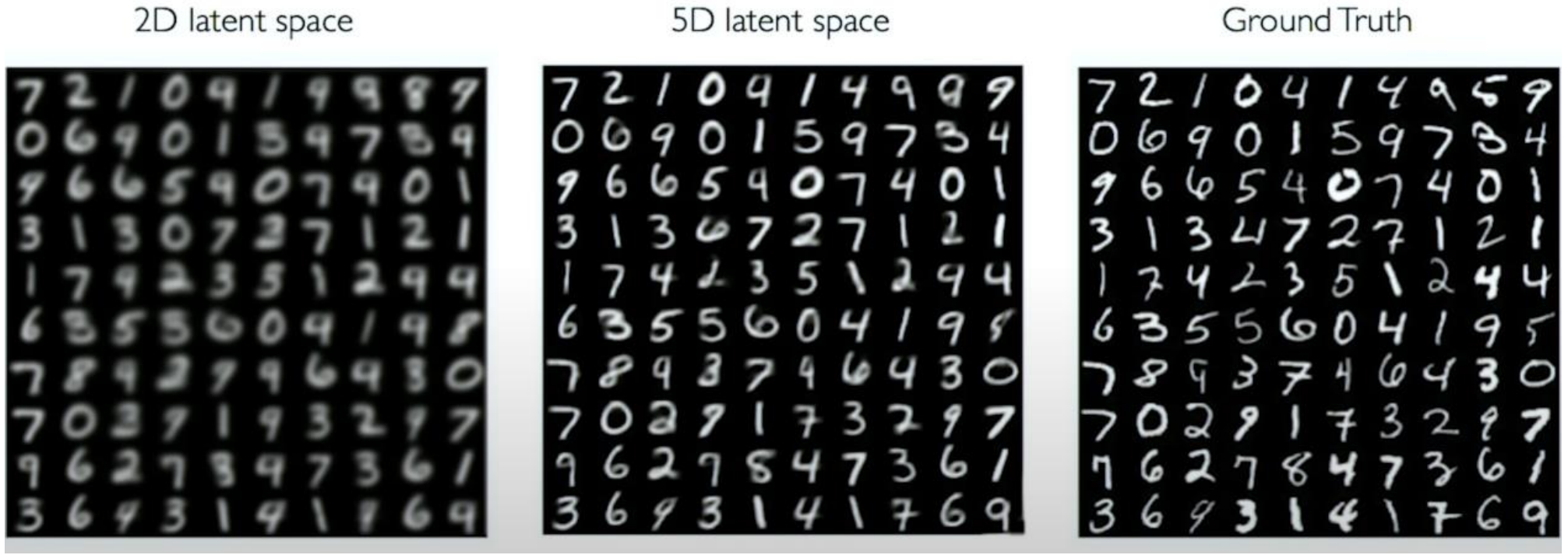
- How can we learn this latent space?
- Train the model to use these features to reconstruct the original data



$$\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|^2$$

Autoencoder Bottleneck

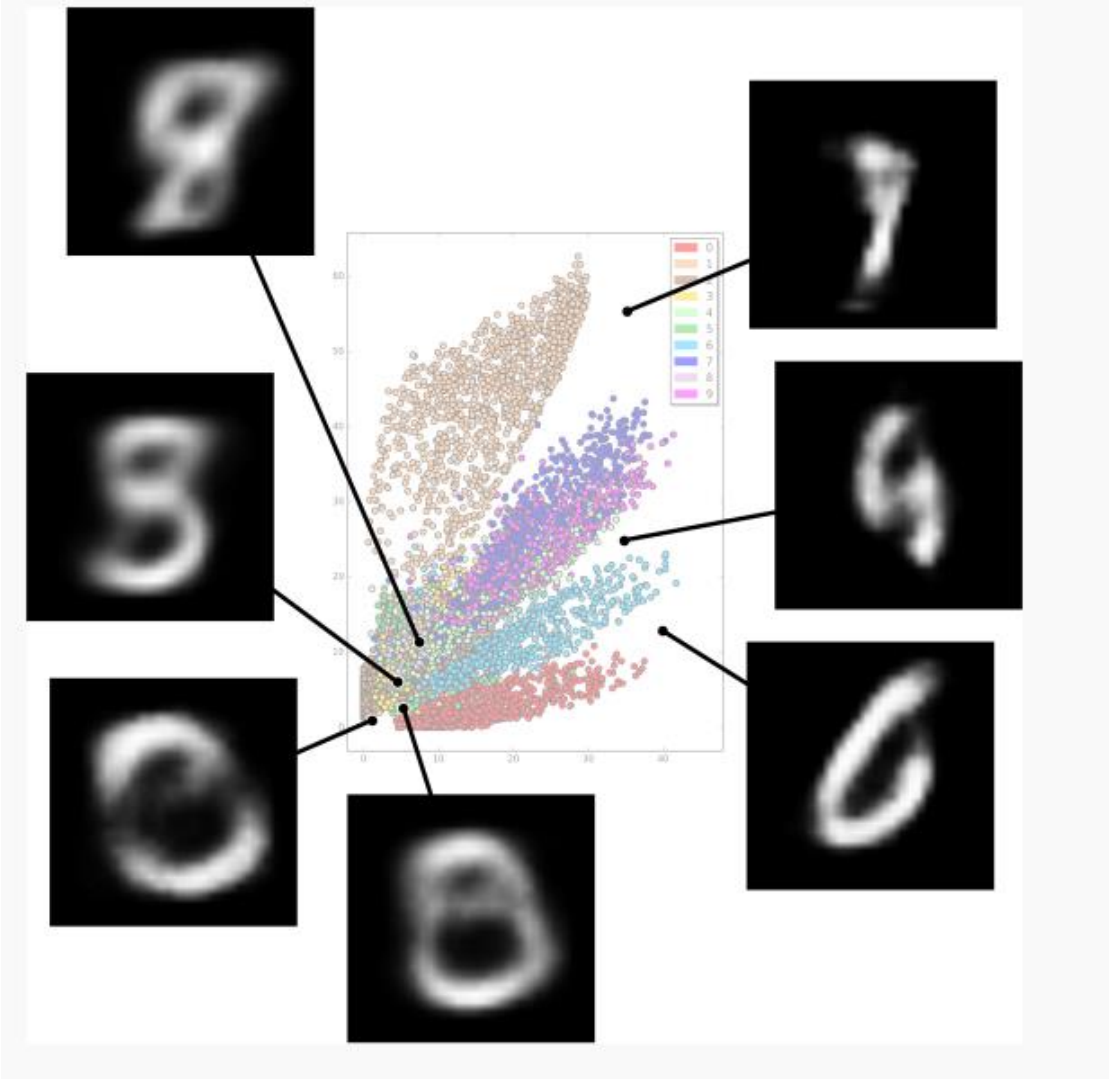
- Autoencoding is a form of compression! Smaller latent space will force a larger training bottleneck.



- Dimensionality of latent space → Reconstruction quality

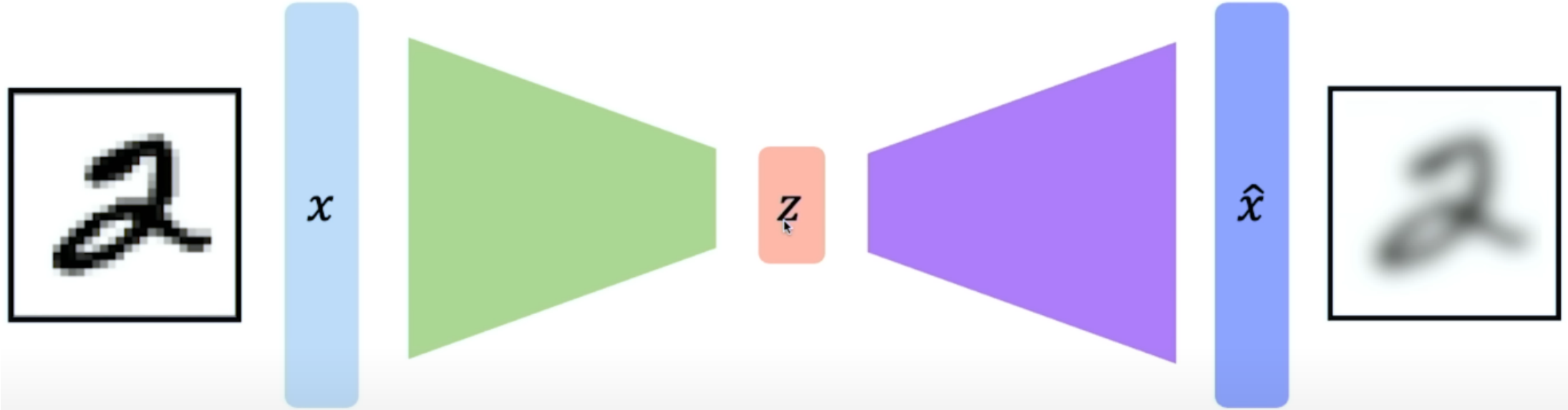
Autoencoder
Bottleneck

- Bottleneck hidden layer forces network to learn a compressed latent representation
- Reconstruction loss forces the latent representation to capture (or encode) as much “information” about the data as possible
- Autoencoding = Automatically encode data; : “Auto” = Self-encoding



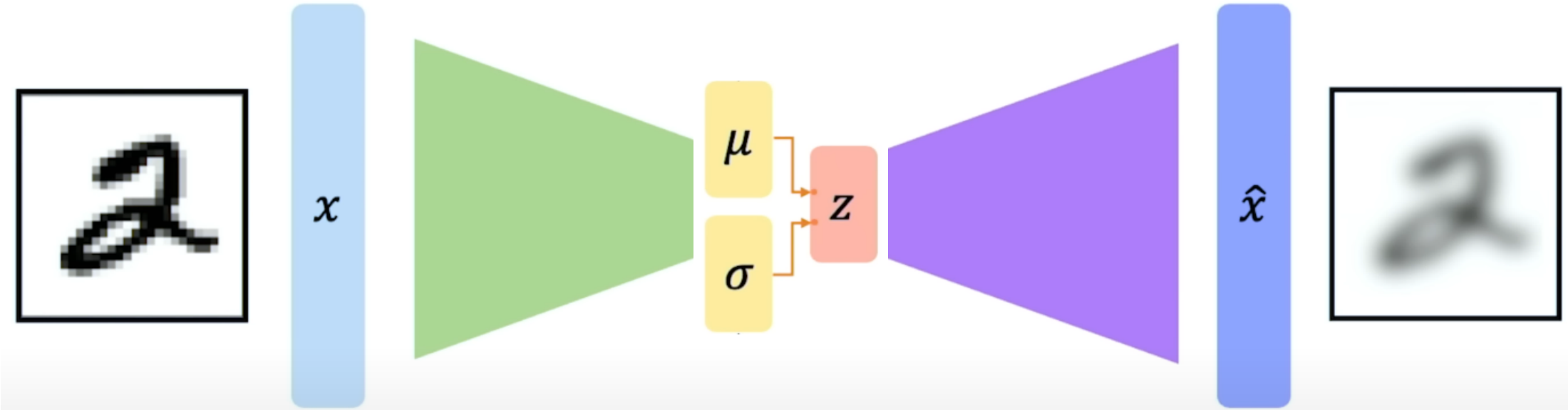
Variational Autoencoder (VAE)

Autoencoder



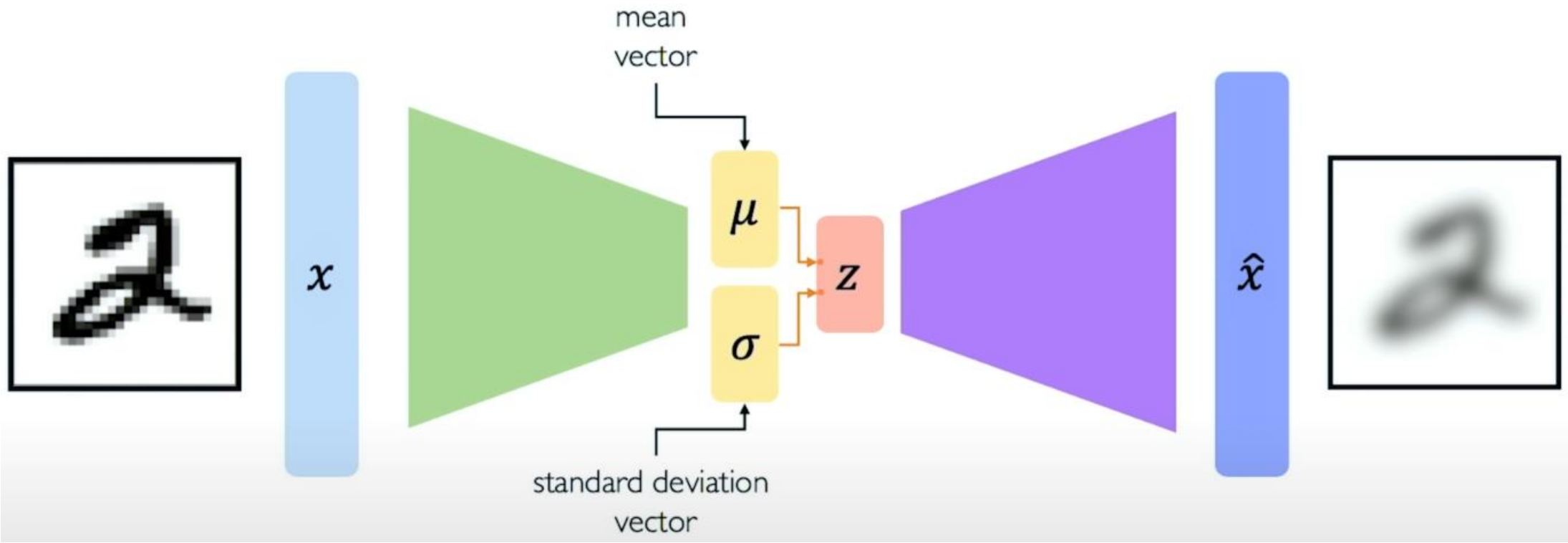
Autoencoder

Variational Autoencoder



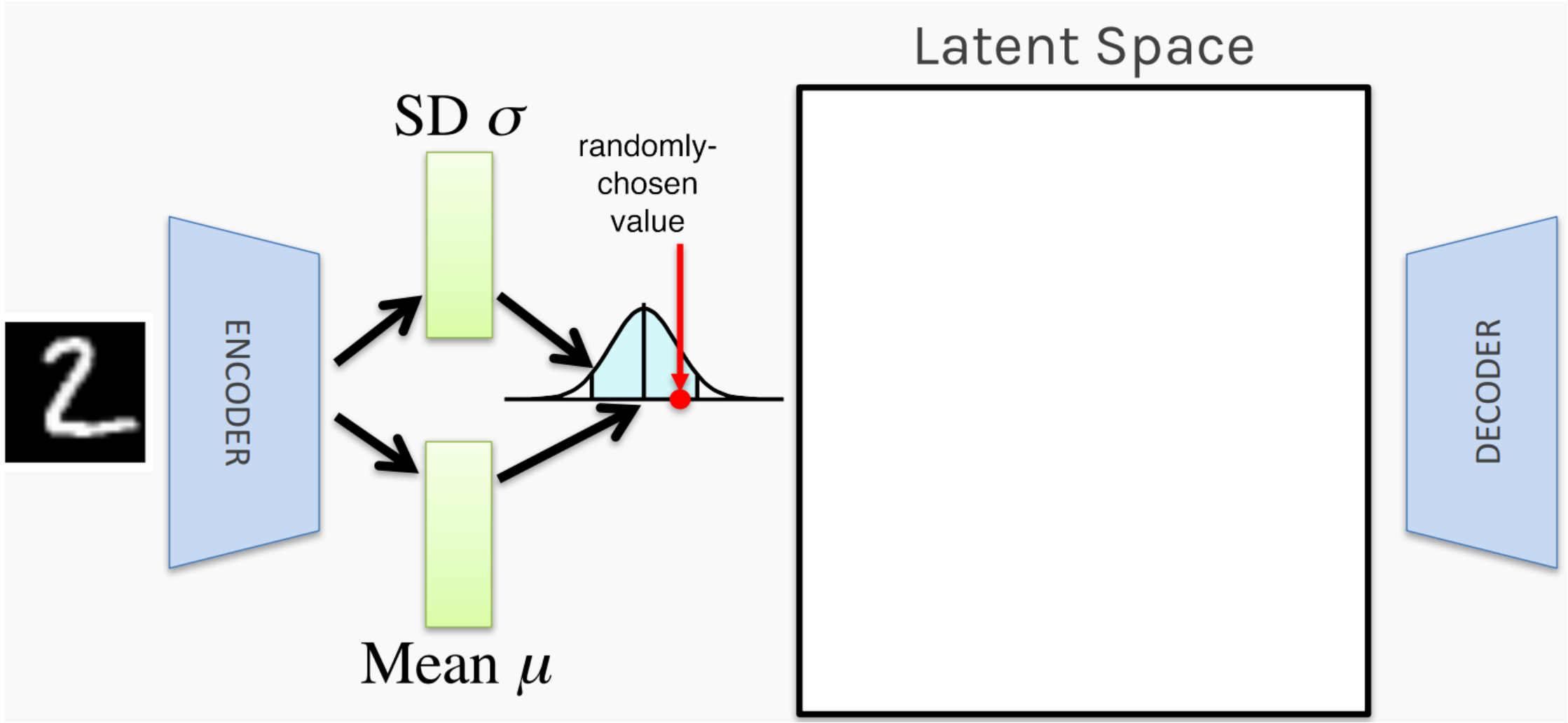
Variational Autoencoder

Variational Autoencoder

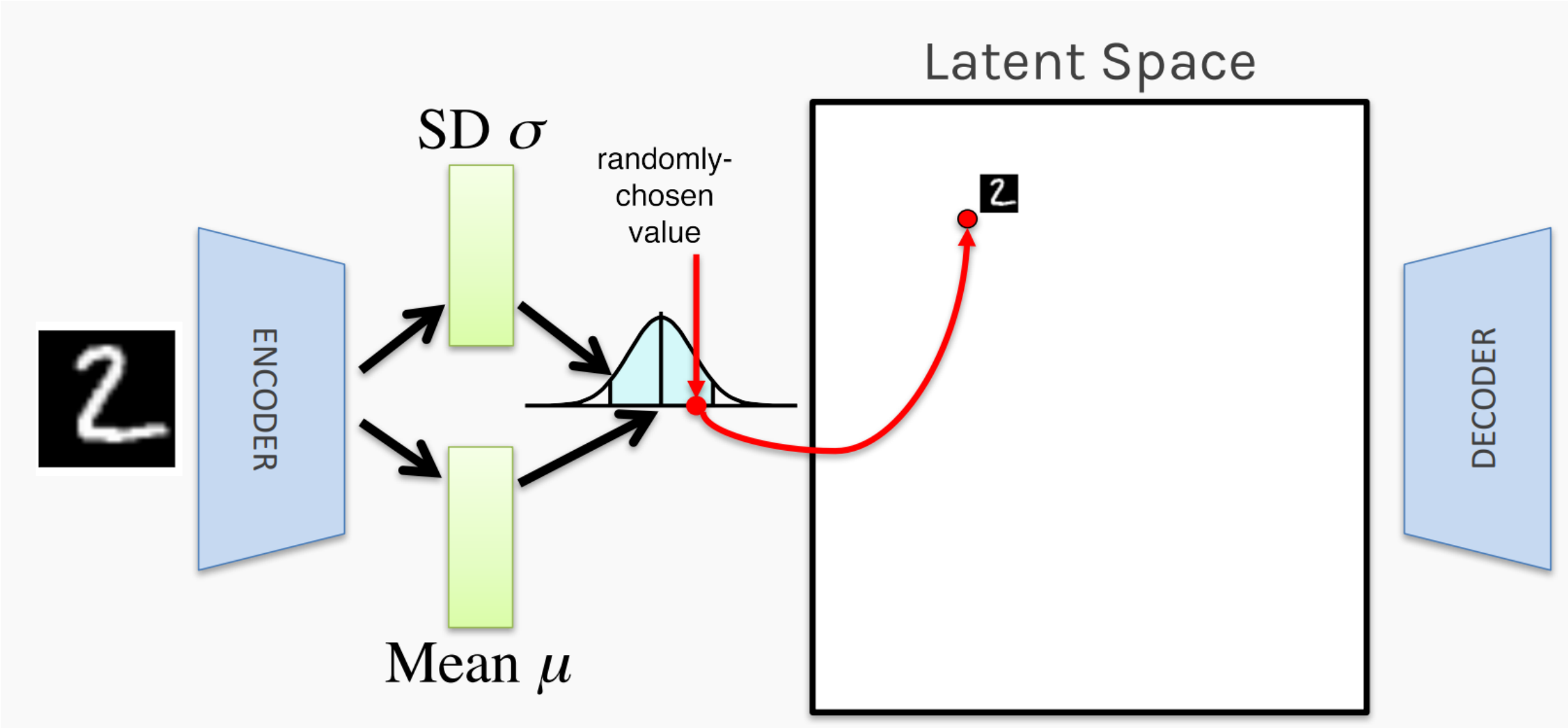


- Variational autoencoders are a probabilistic twist on autoencoders!
- Sample from the mean and standard deviation to compute latent space.

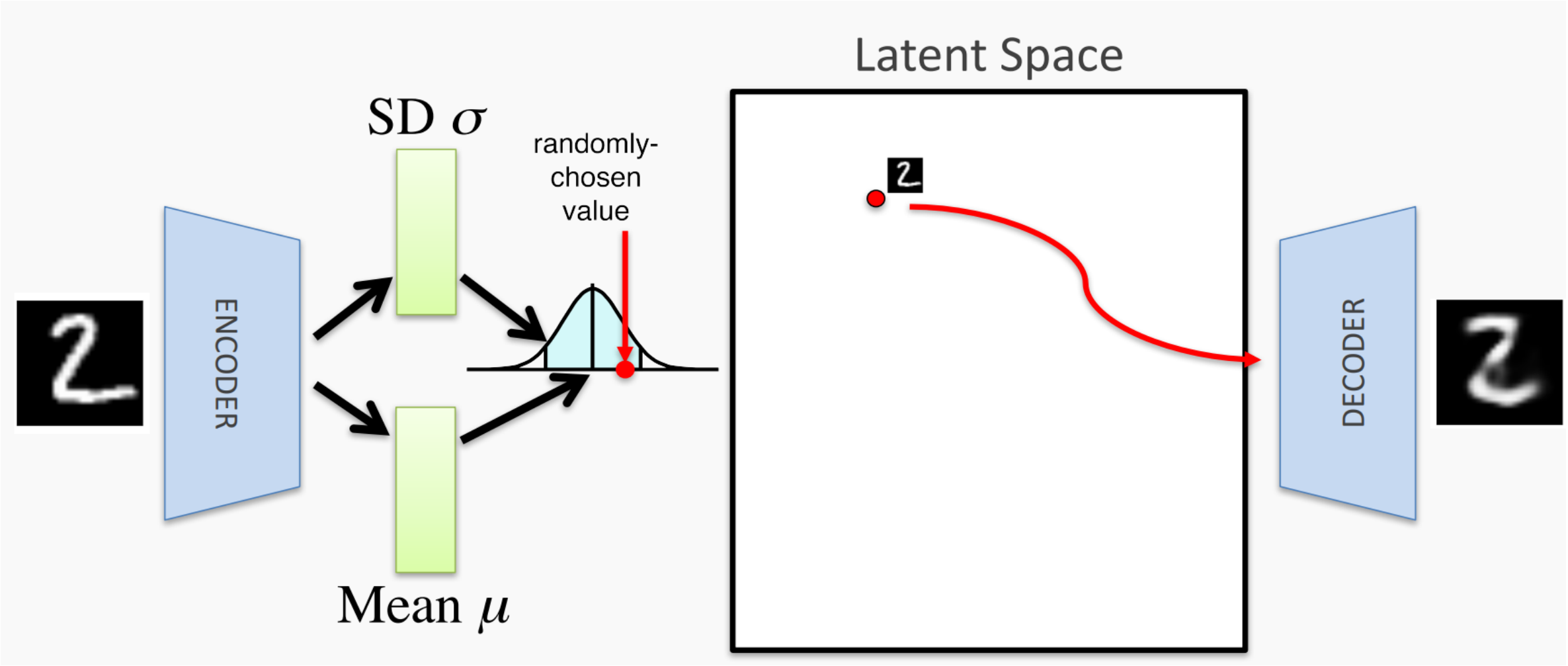
Variational Autoencoder



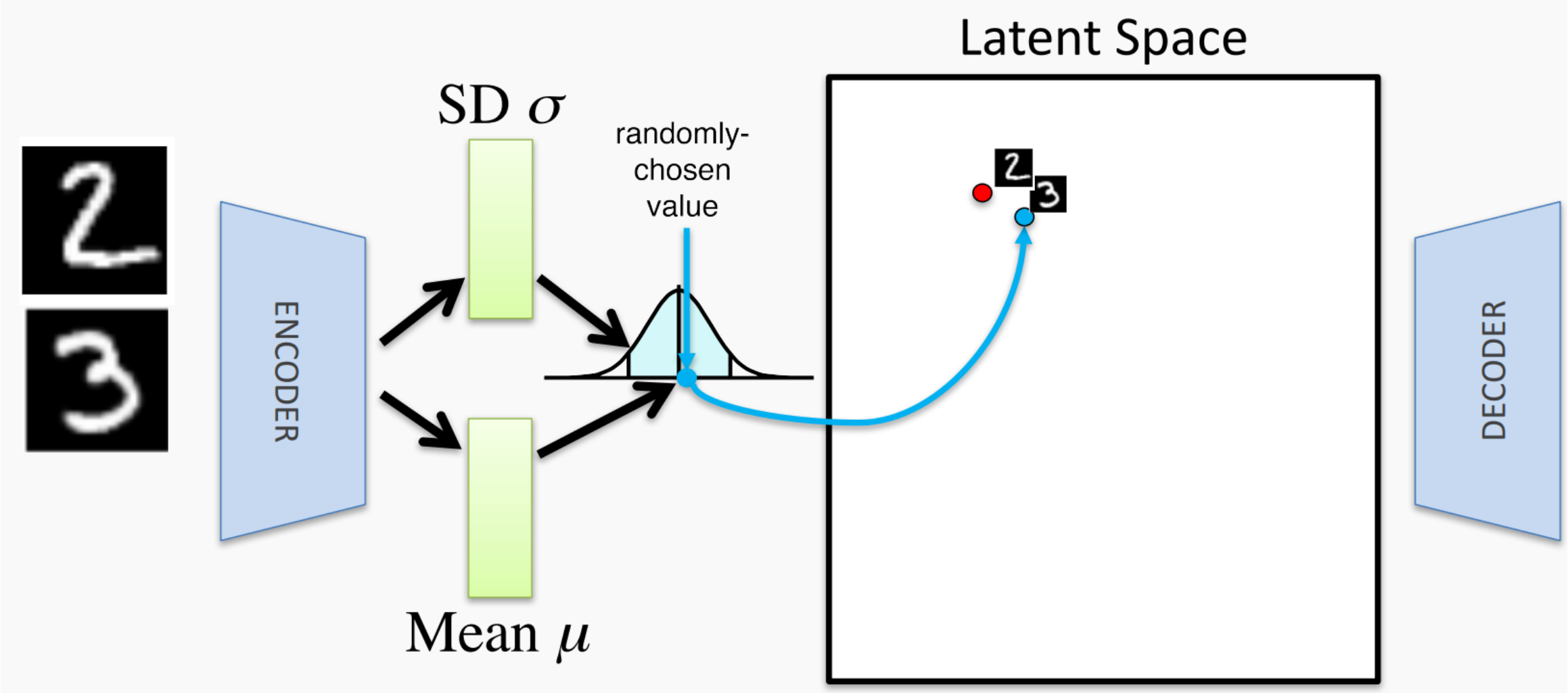
Variational Autoencoder



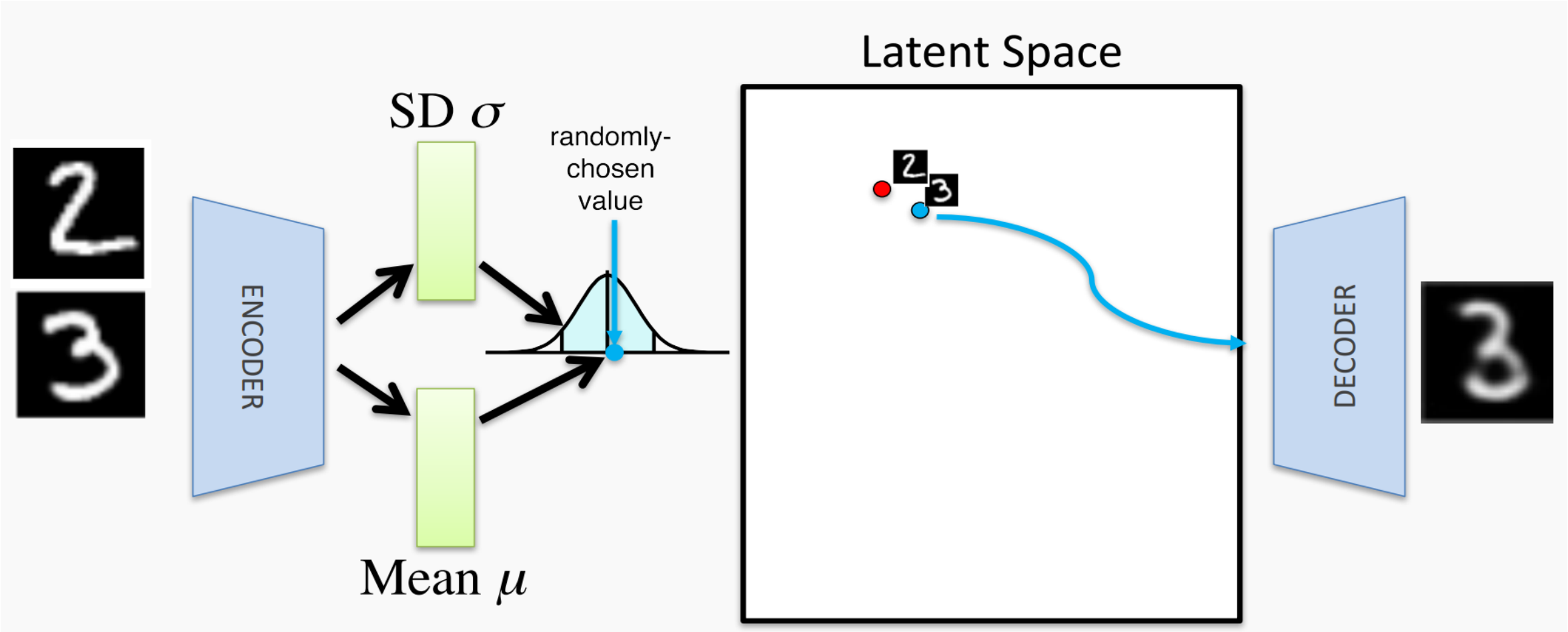
Variational Autoencoder



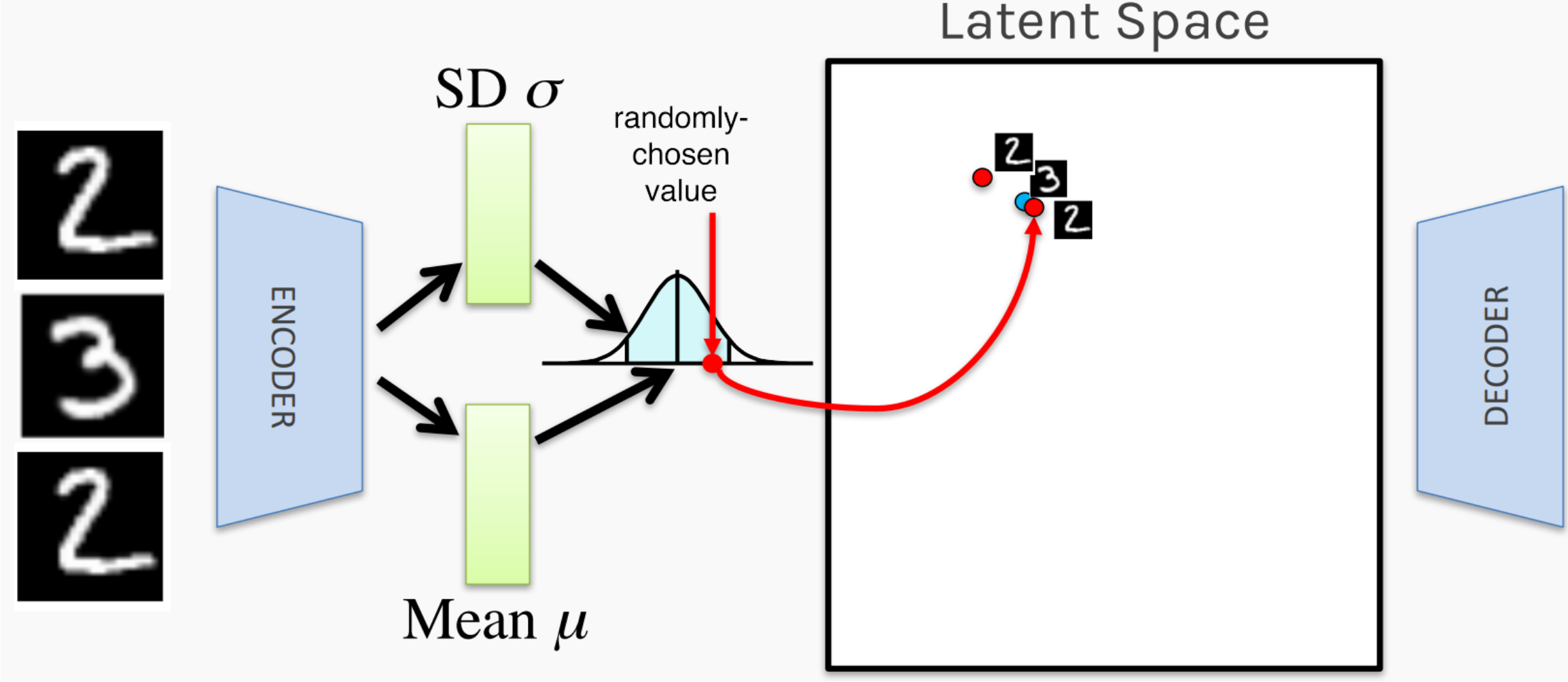
Variational Autoencoder



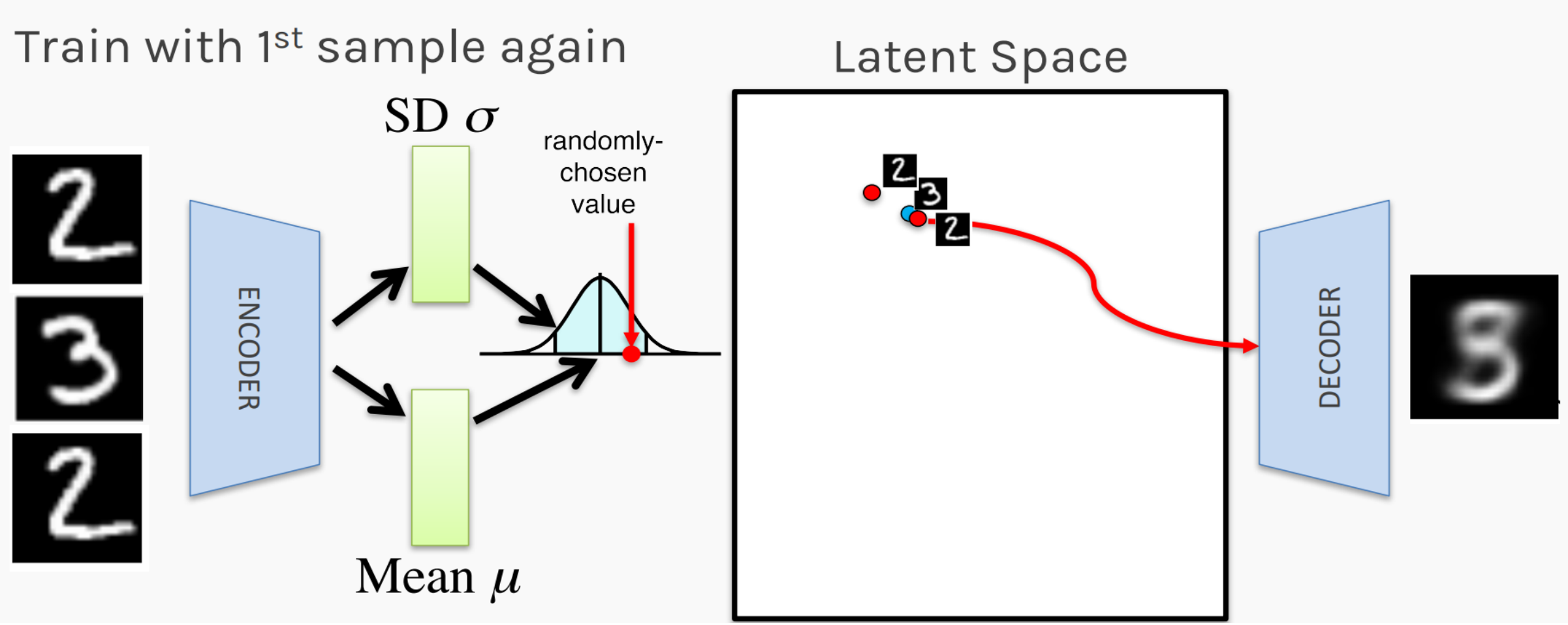
Variational Autoencoder



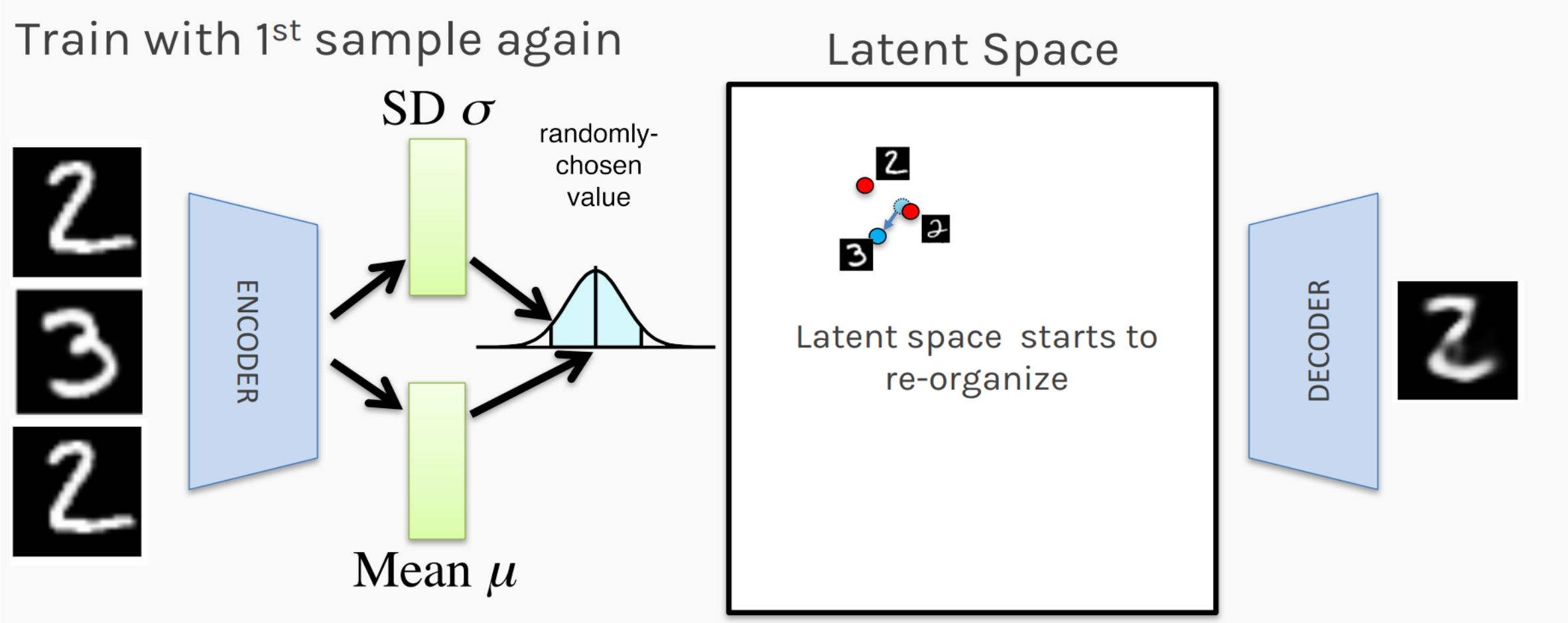
Variational Autoencoder



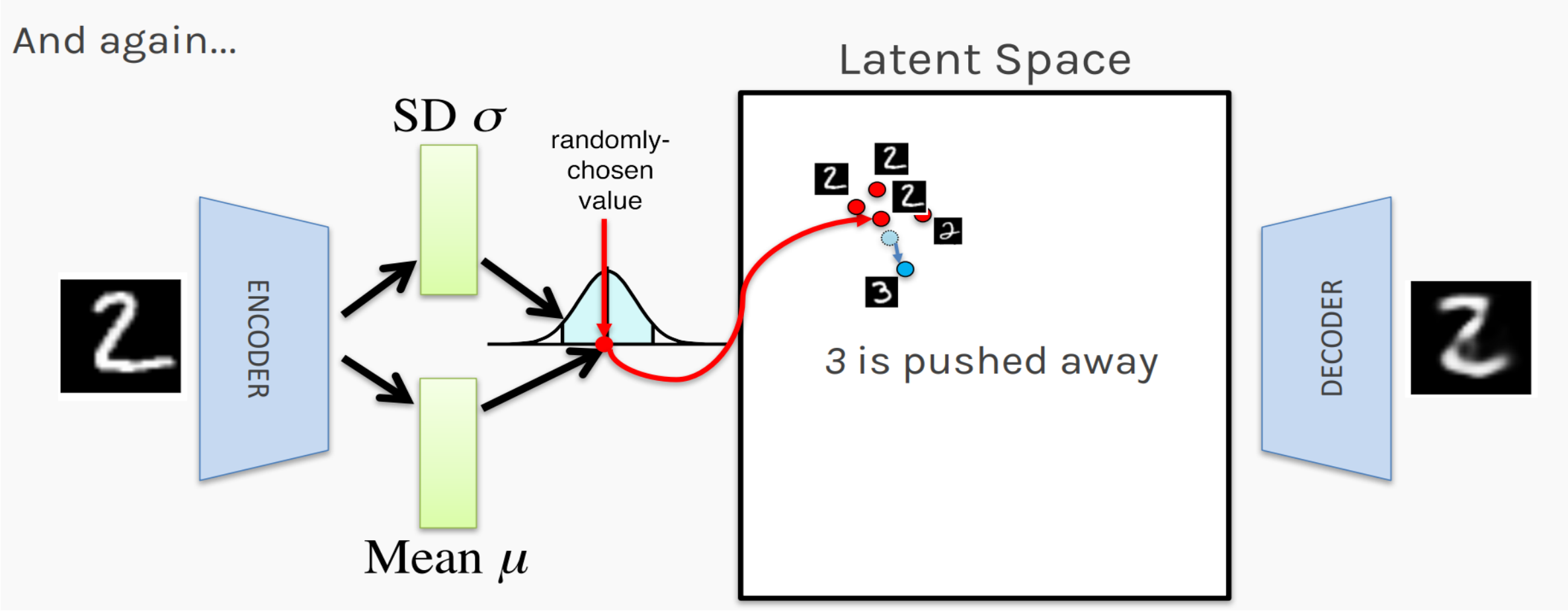
Variational Autoencoder



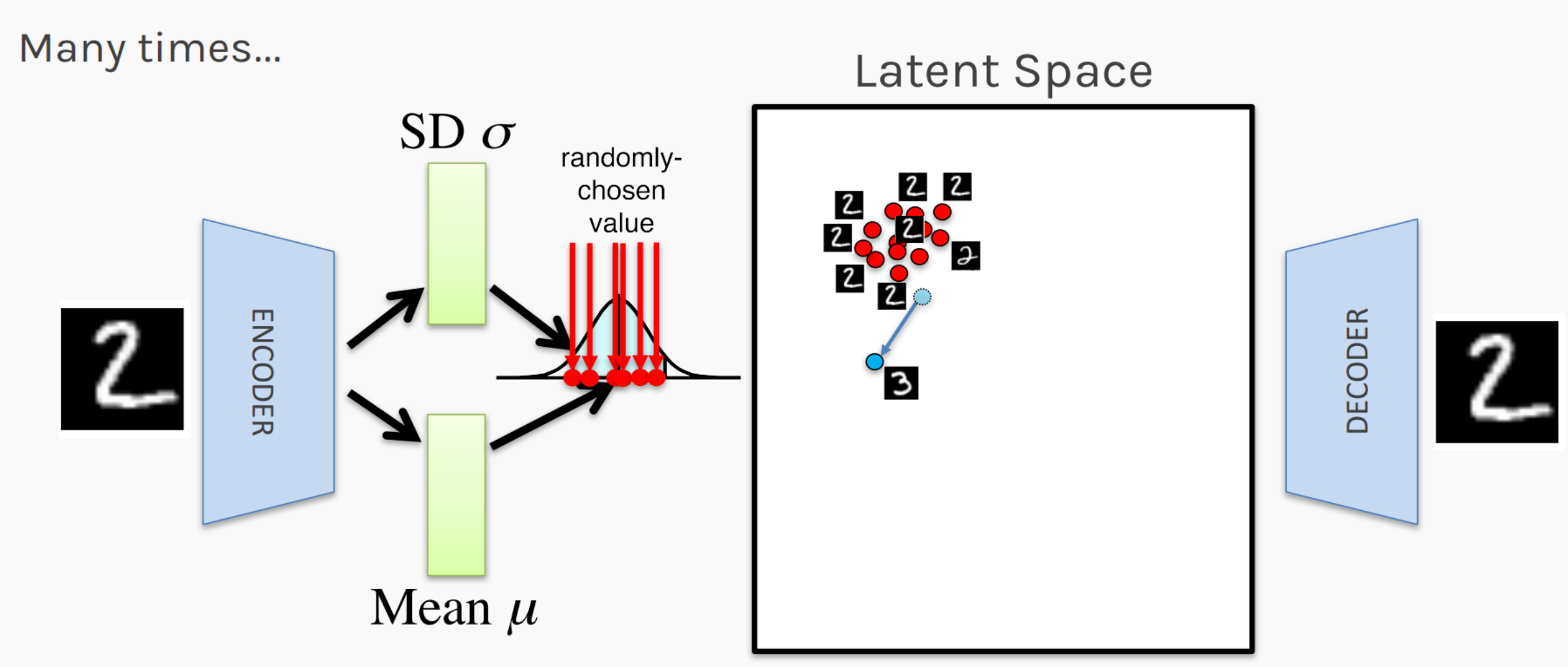
Variational Autoencoder



Variational Autoencoder

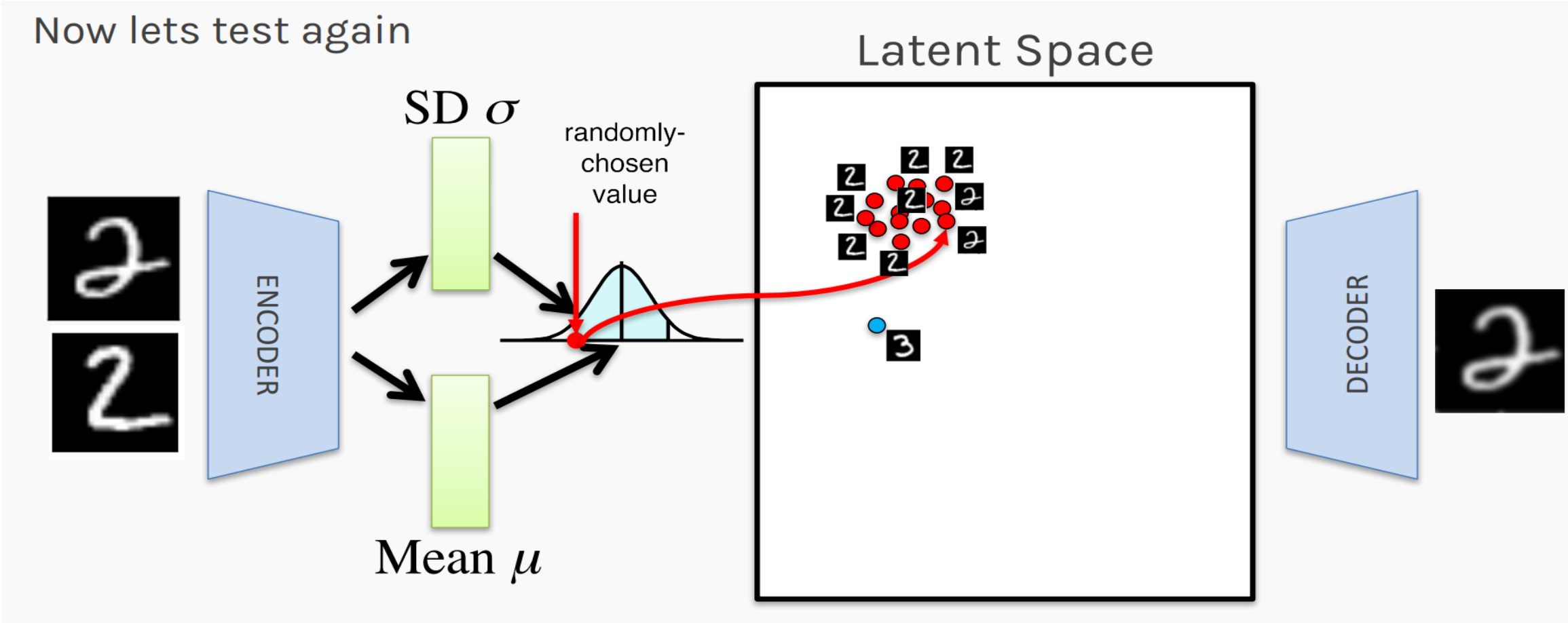


Variational Autoencoder

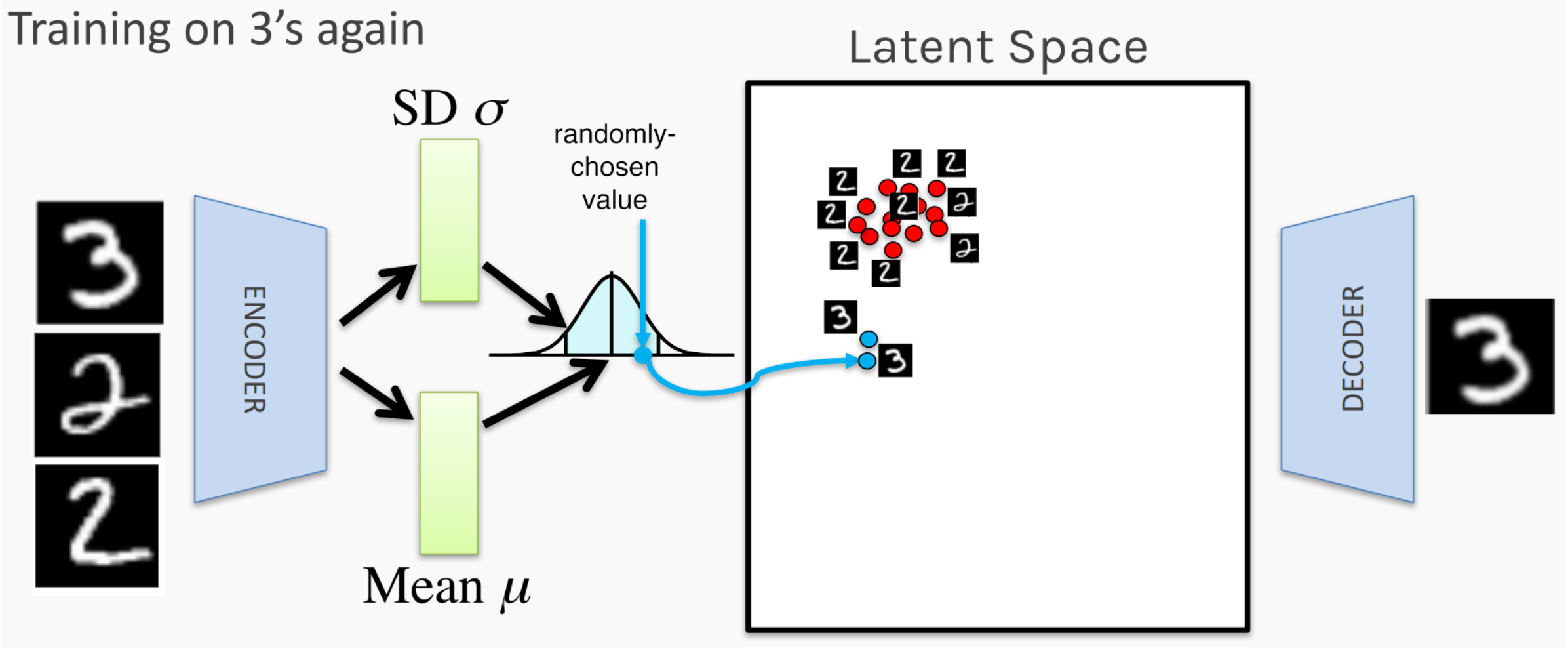


Variational Autoencoder

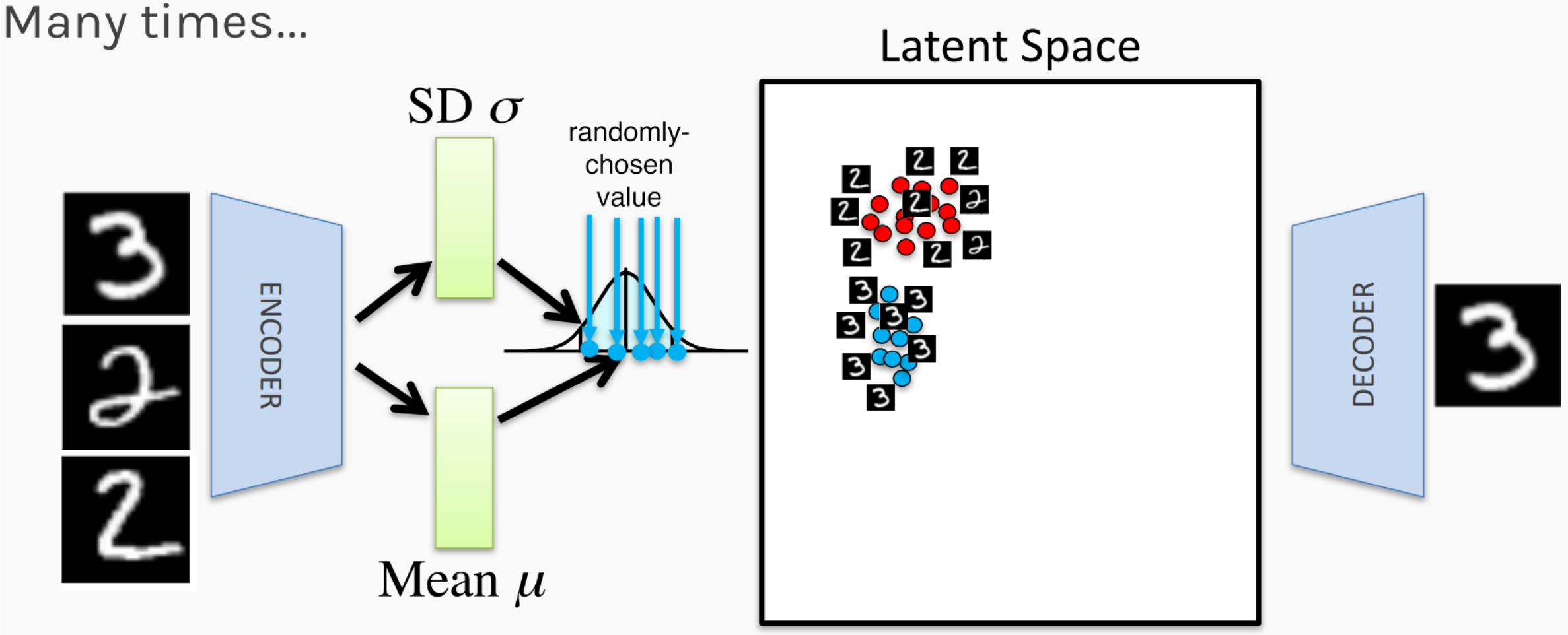
- Blending the latent variables



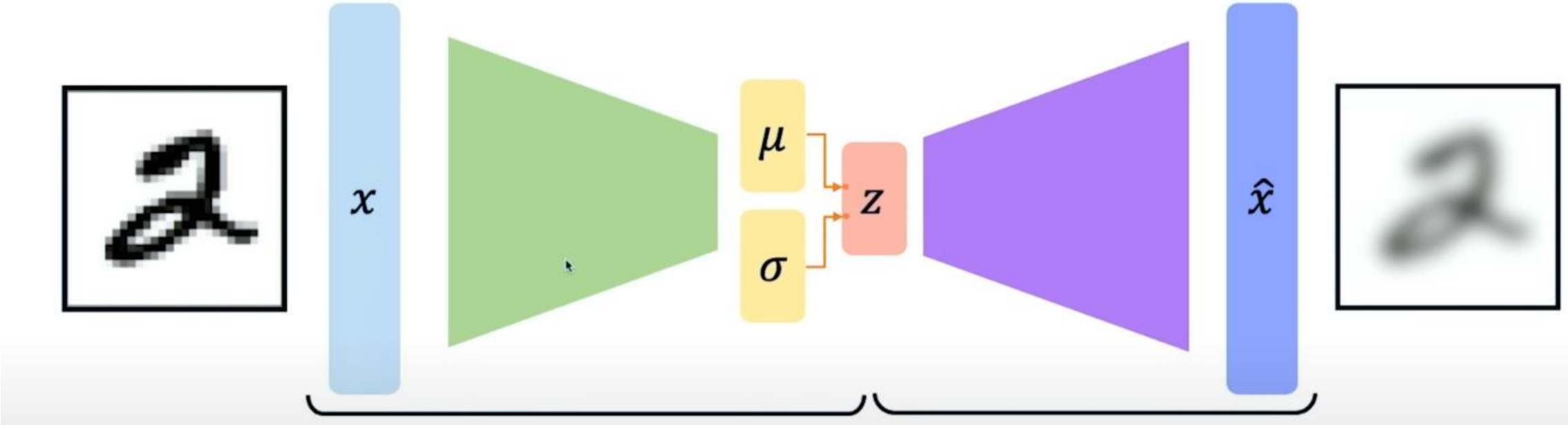
Variational Autoencoder



Variational Autoencoder

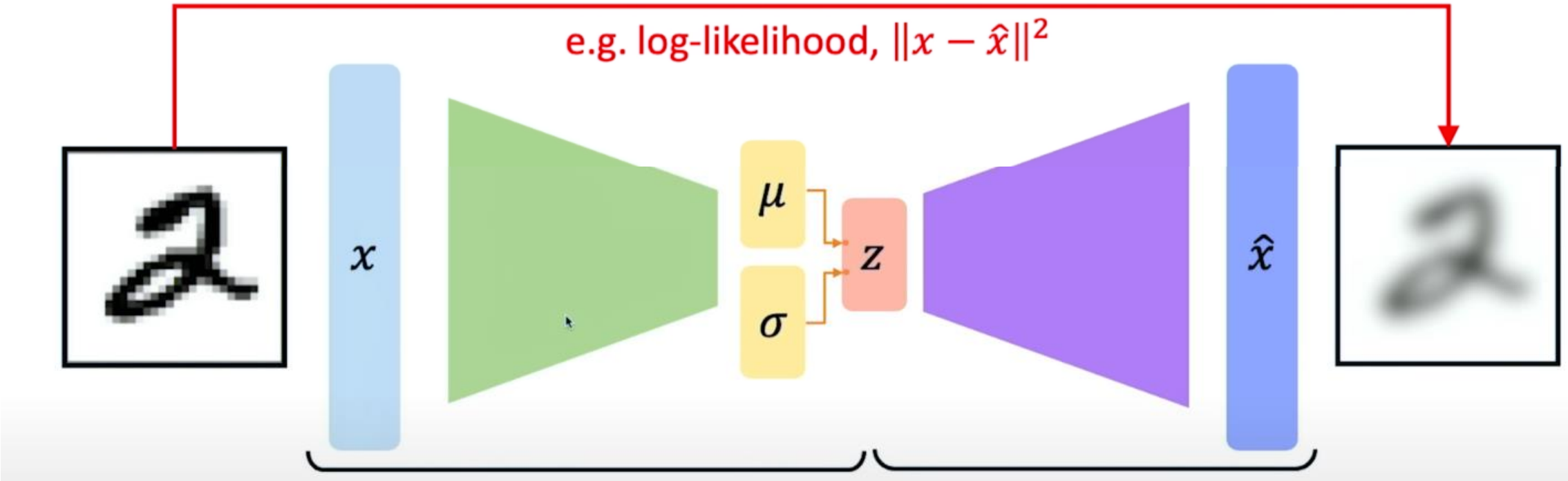


Variational Autoencoder Optimization



Encoder computes $q_\phi(z|x)$ Decoder computes $p_\theta(x|z))$

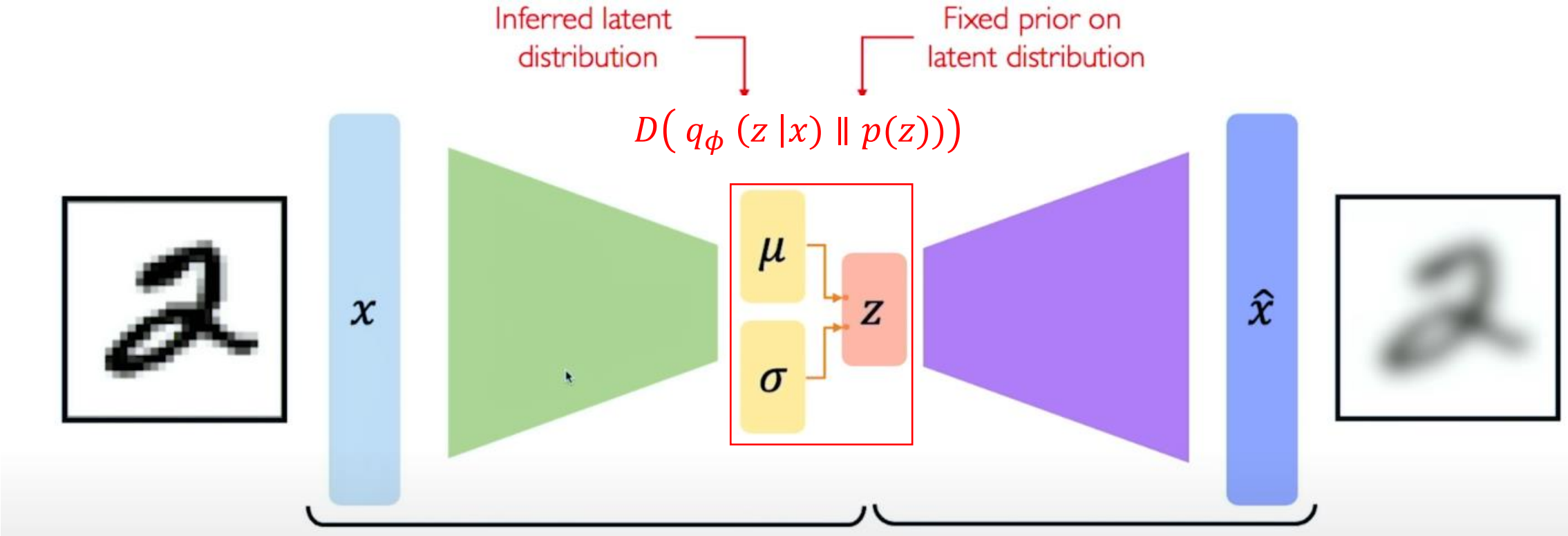
Variational Autoencoder Optimization



Encoder computes $q_{\phi}(z|x)$ Decoder computes $p_{\theta}(x|z))$

- $\mathcal{L}(\phi, \theta, \mathrm{x}) =$ Reconstruction loss $+$ Regularisation term

Variational Autoencoder Optimization



- $\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}, \mathrm{x}) = \text{Reconstruction loss} + \text{Regularisation term}$

Variational Autoencoder
Optimization – Prior on latent distribution

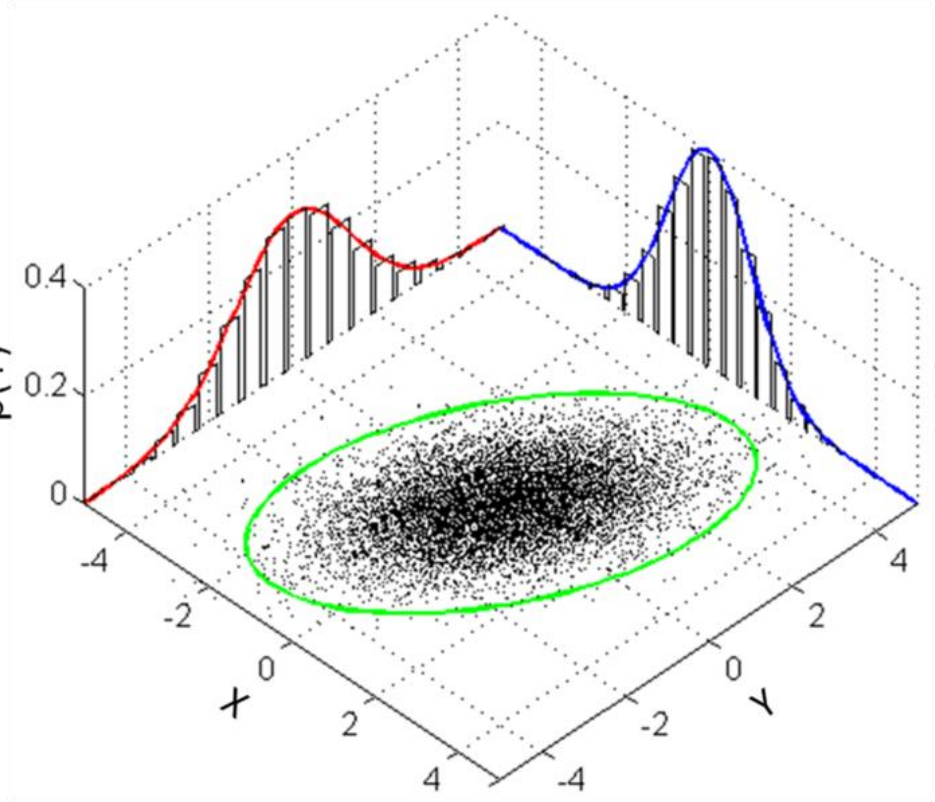
- Common choice of prior – Normal distribution

$$p(z) = \mathcal{N}(\mu = 0, \sigma^2 = 1)$$

  - Encourages encoding to distribute encodings evenly around the center of the latent space
  - Penalize the network when it tries to cheat by clustering points in specific region (i.e. By memorizing the data)

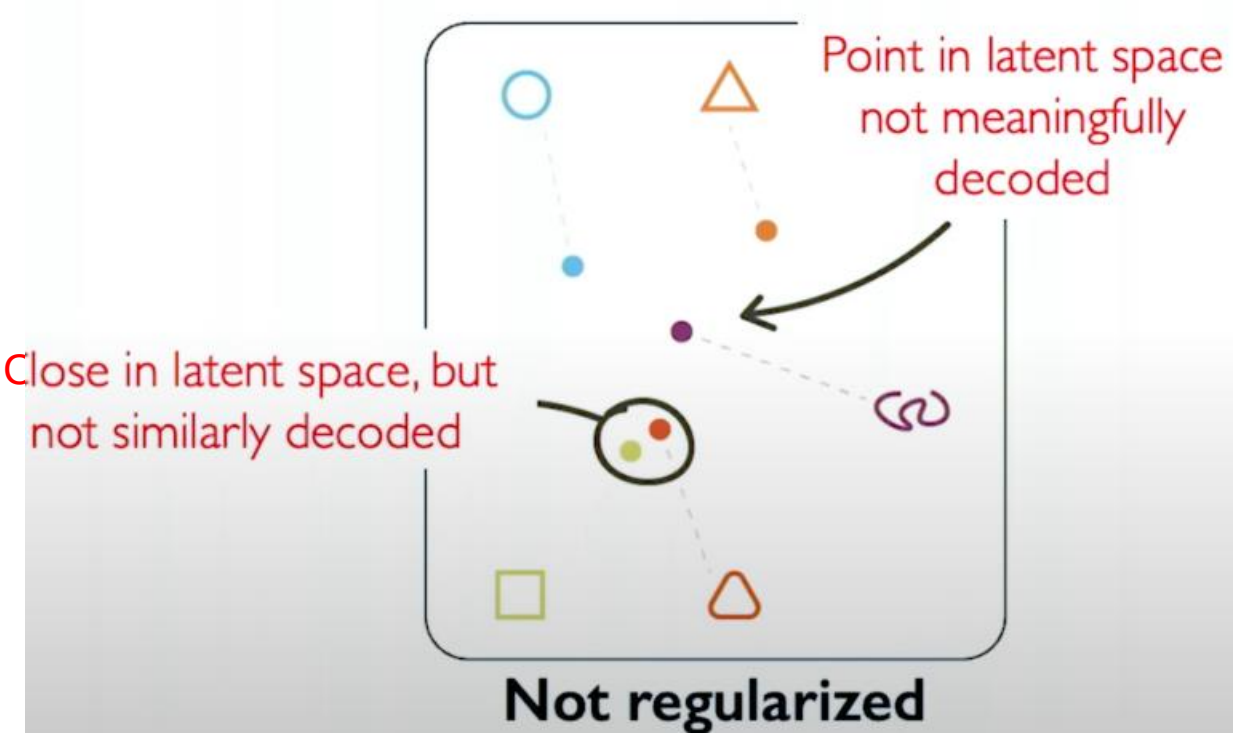
- KL Divergence

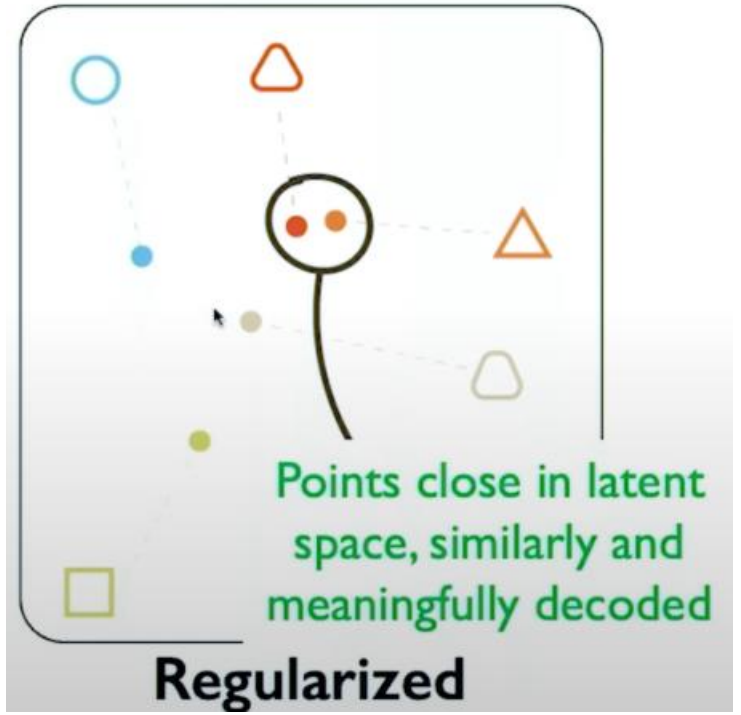
$$D\big(\, q_\phi\,(z\,|x)\,\|\,p(z)\big) = \,-\frac{1}{2}\sum_{j=0}^{k-1}(\sigma_j\, + \mu_j^2\, -\, 1\, -\, log\,\sigma_j)$$



Variational Autoencoder
Optimization – Intuition on regularization and normal prior

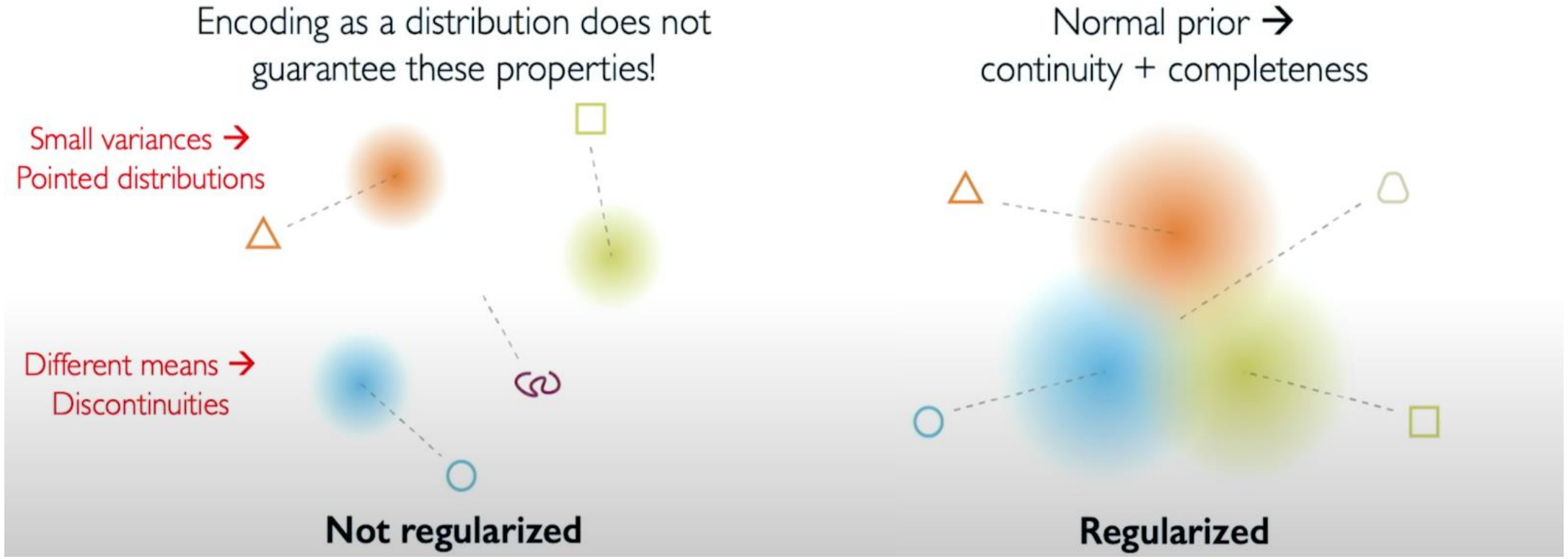
- Regularization
  - Continuity: Points that are close in latent space → Similar content after decoding
  - Completeness: Sampling from latent space → “Meaningful” content after decoding





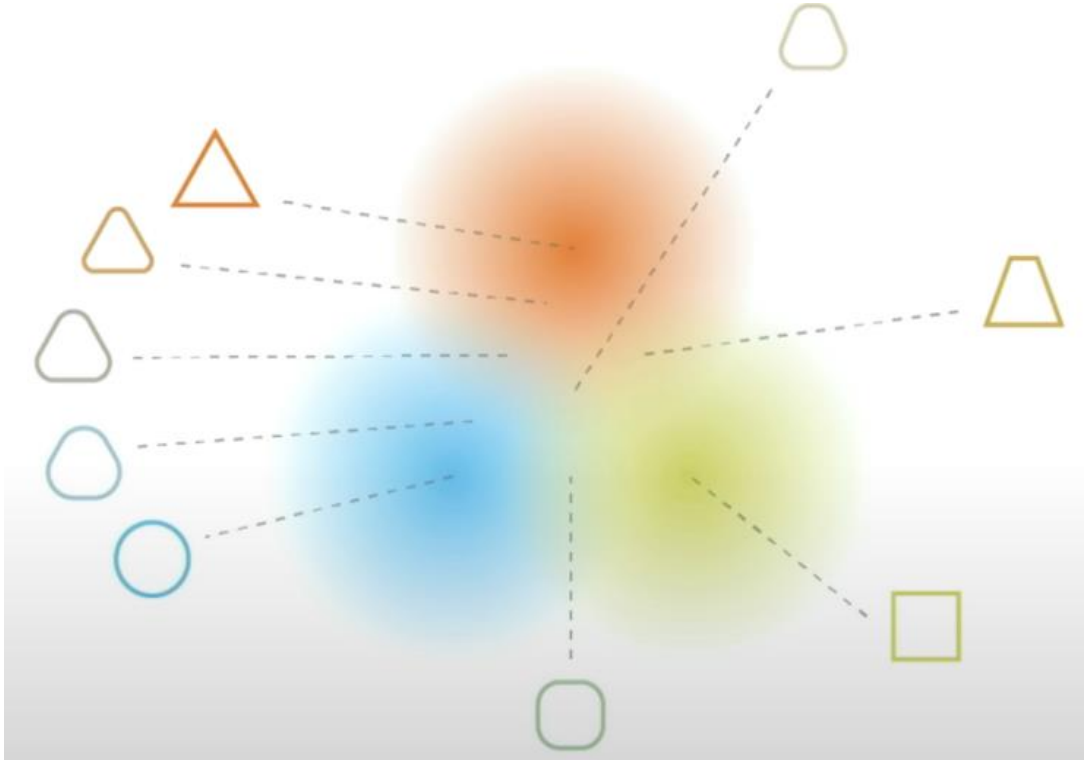
Variational Autoencoder
Optimization – Intuition on regularization and normal prior

- Regularization
  - Continuity: Points that are close in latent space → Similar content after decoding
  - Completeness: Sampling from latent space → “Meaningful” content after decoding



Variational Autoencoder
Optimization – Intuition on regularization and normal prior

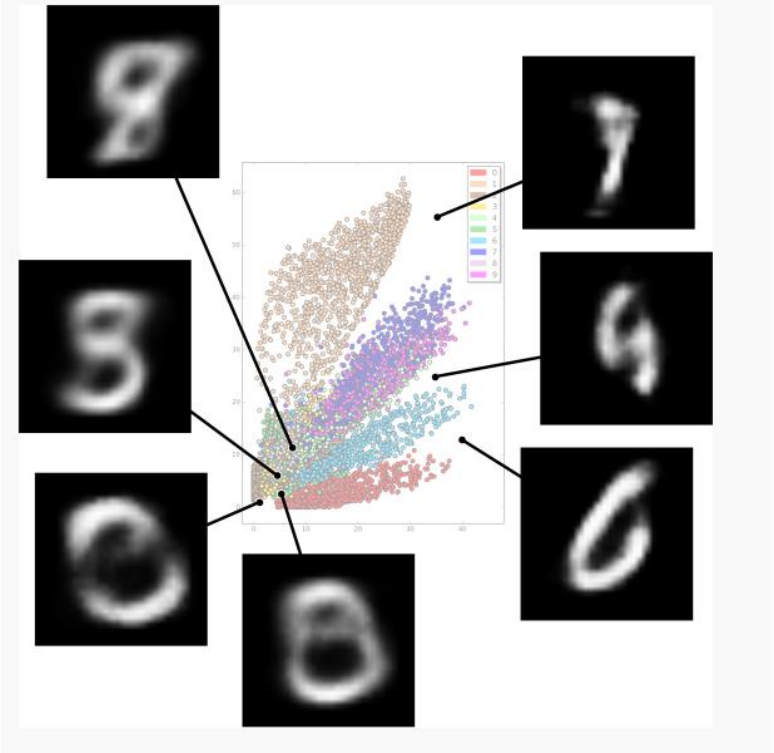
- Regularization
  - Continuity: Points that are close in latent space → Similar content after decoding
  - Completeness: Sampling from latent space → “Meaningful” content after decoding



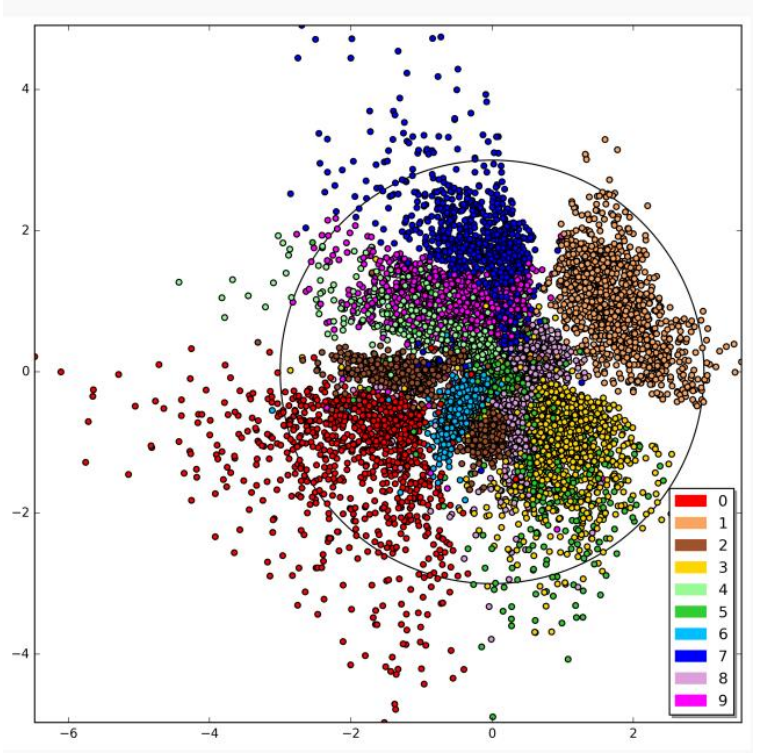
Regularization with normal prior helps enforce information gradient in the latent space!

Variational Autoencoder
Optimization – Intuition on regularization and normal prior

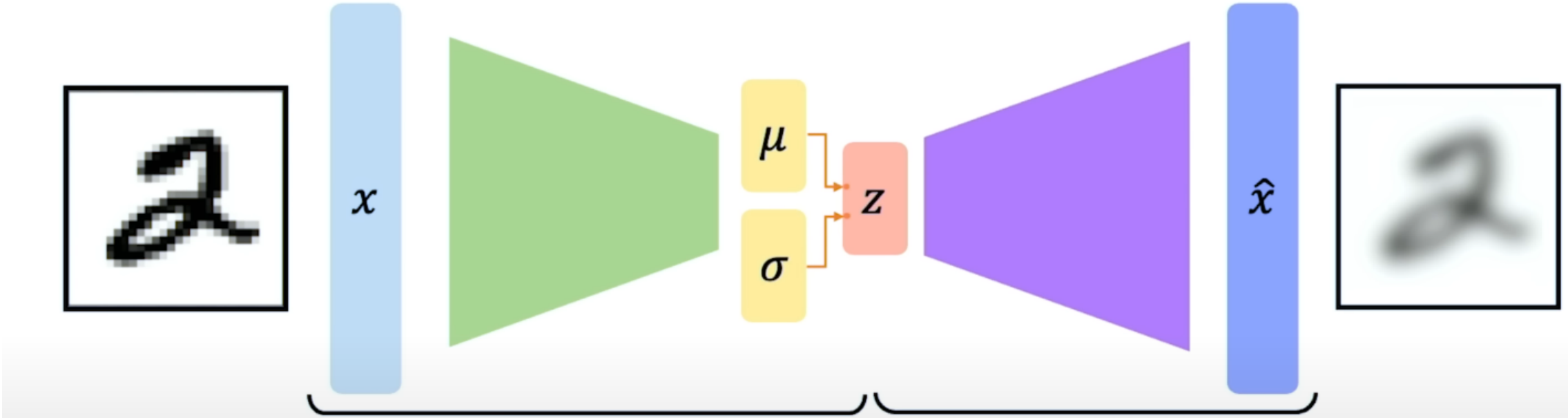
- Regularization
  - Continuity: Points that are close in latent space → Similar content after decoding
  - Completeness: Sampling from latent space → “Meaningful” content after decoding



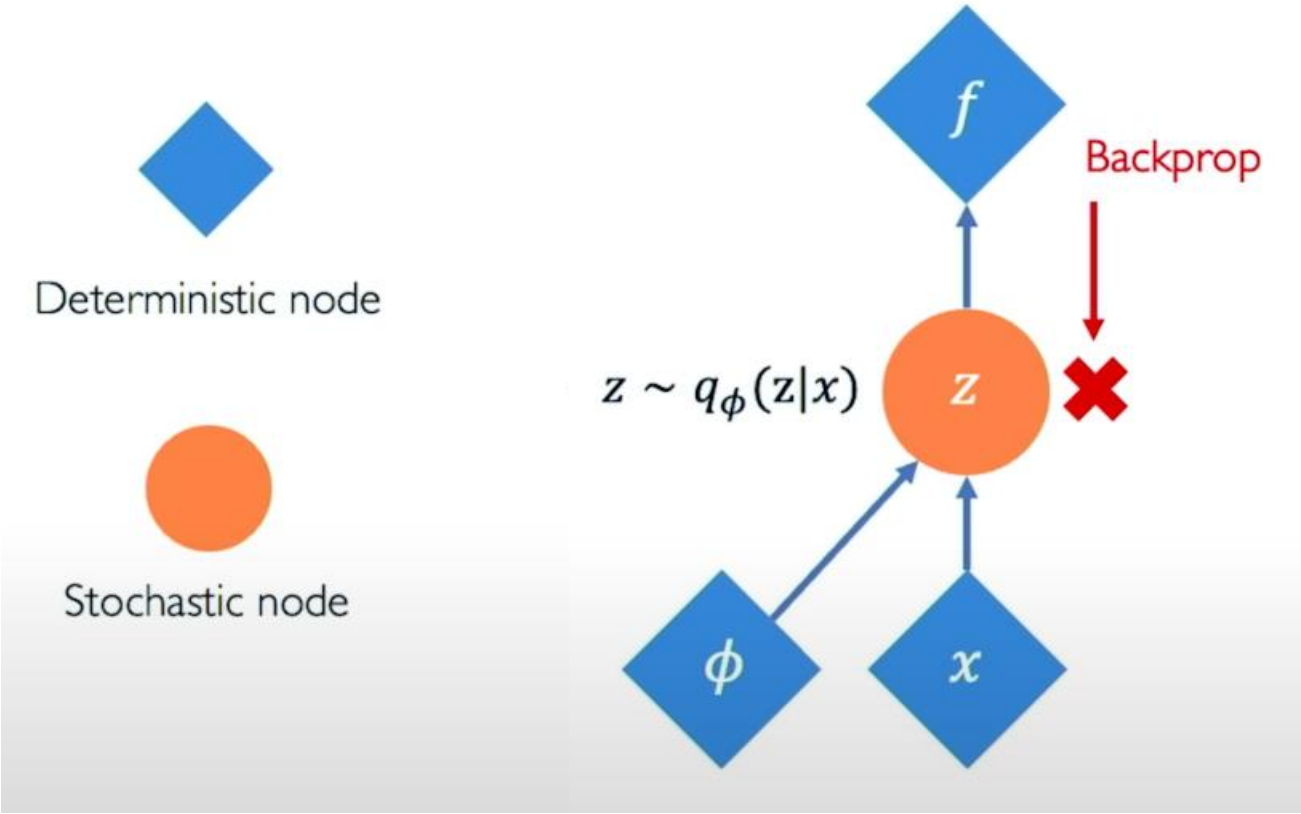
Regularization



Variational Autoencoder Optimization



Variational Autoencoder
Optimization – Reparametrizing the sampling layer



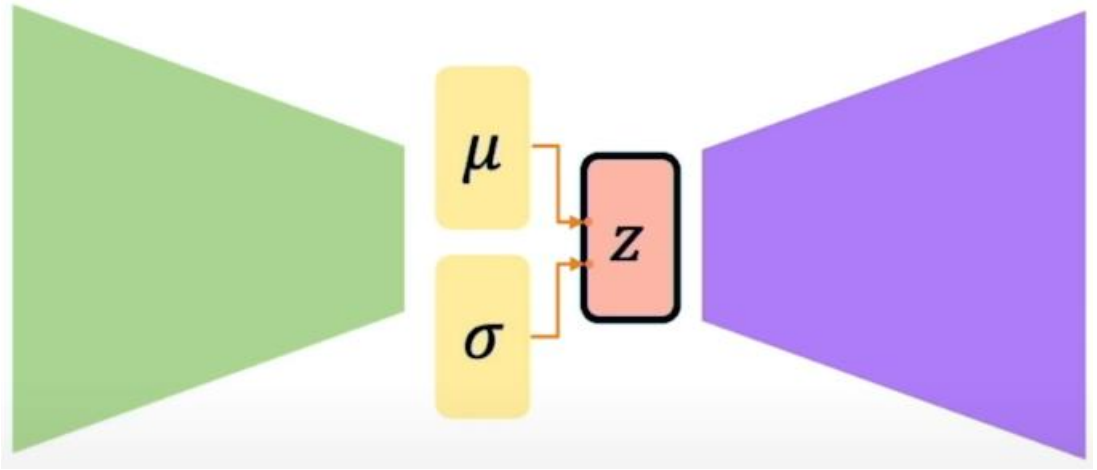
Variational Autoencoder
Optimization – Reparametrizing the sampling layer

- Sampled latent vector z

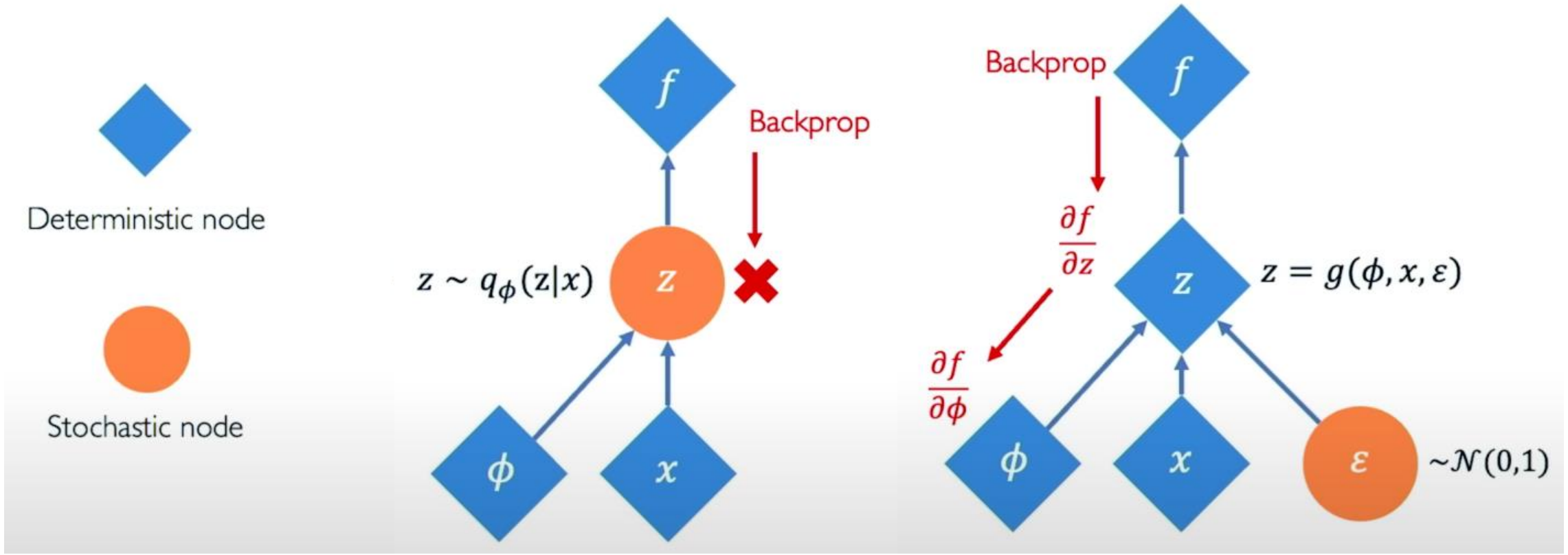
$$z \sim \mathcal{N}(\mu, \sigma^2)$$

- Consider z as the sum of
  - A fixed μ vector
  - And fixed σ vector, scaled by random constant drawn from the prior distribution

$$z = \mu + \sigma \odot \epsilon$$

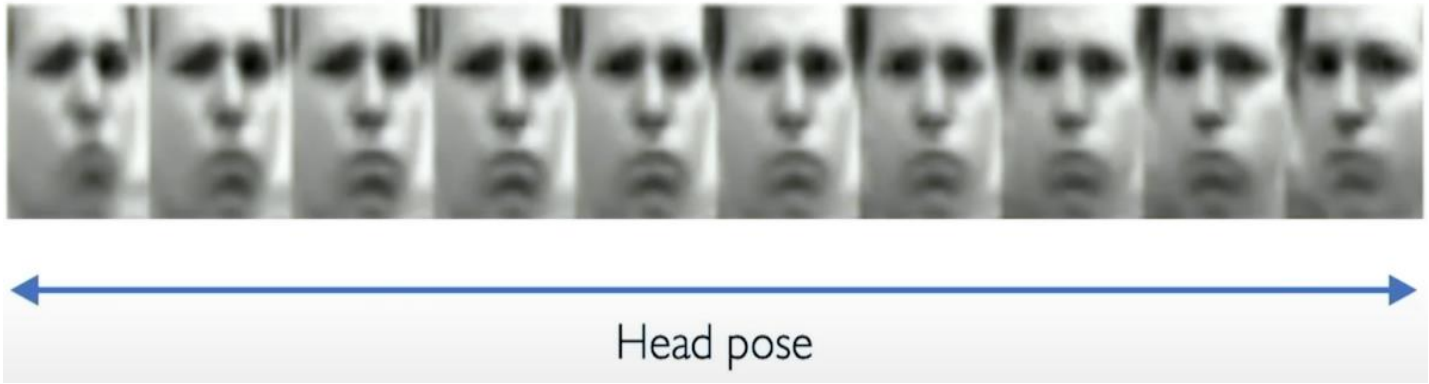


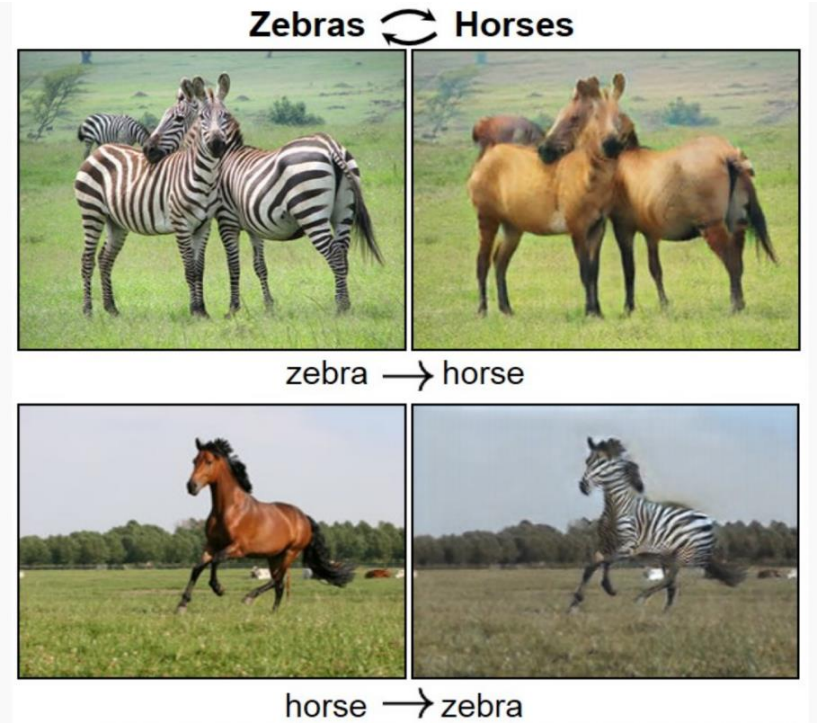
Variational Autoencoder
Optimization – Reparametrizing the sampling layer



Variational Autoencoder
Optimization – Latent perturbation

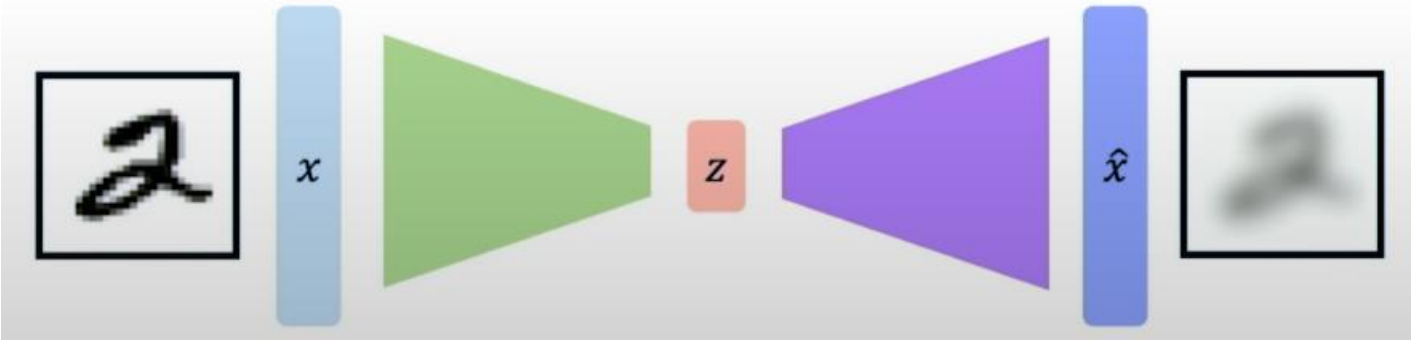
- Different dimensions of z encodes different interpretable latent features
- Slowly increase of decrease a single latent variable while keep all other variables fixed





Variational Autoencoder
Summary

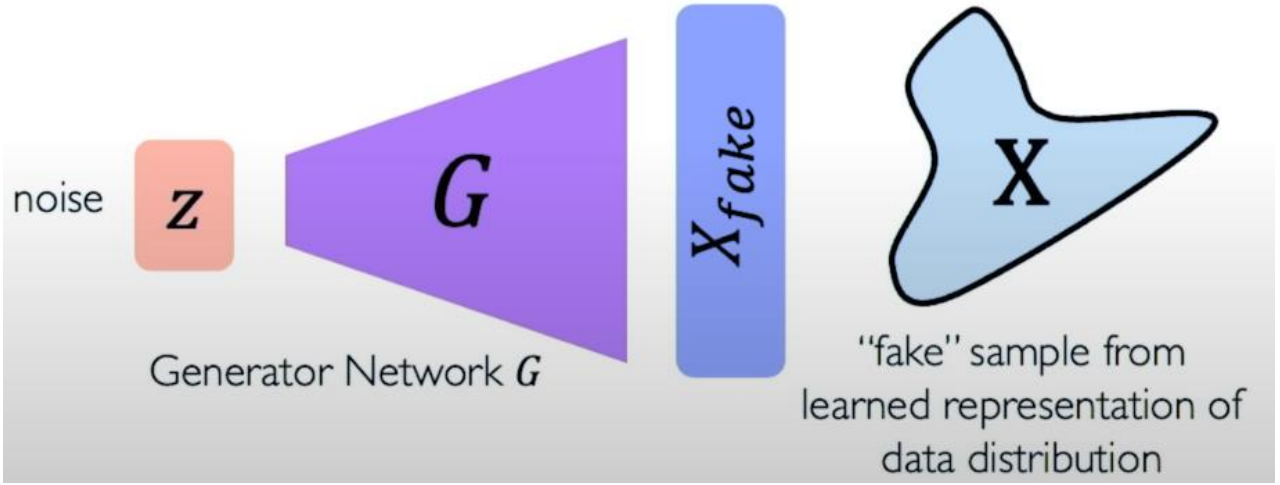
- Compress the representation of world to something we can use to learn
- Reconstruction allows for unsupervised learning (no labels!)
- Reparameterization trick to train end-to-end.
- Interpret hidden latent variables using perturbation
- Generating new samples.



Generative adversarial networks (GANs)

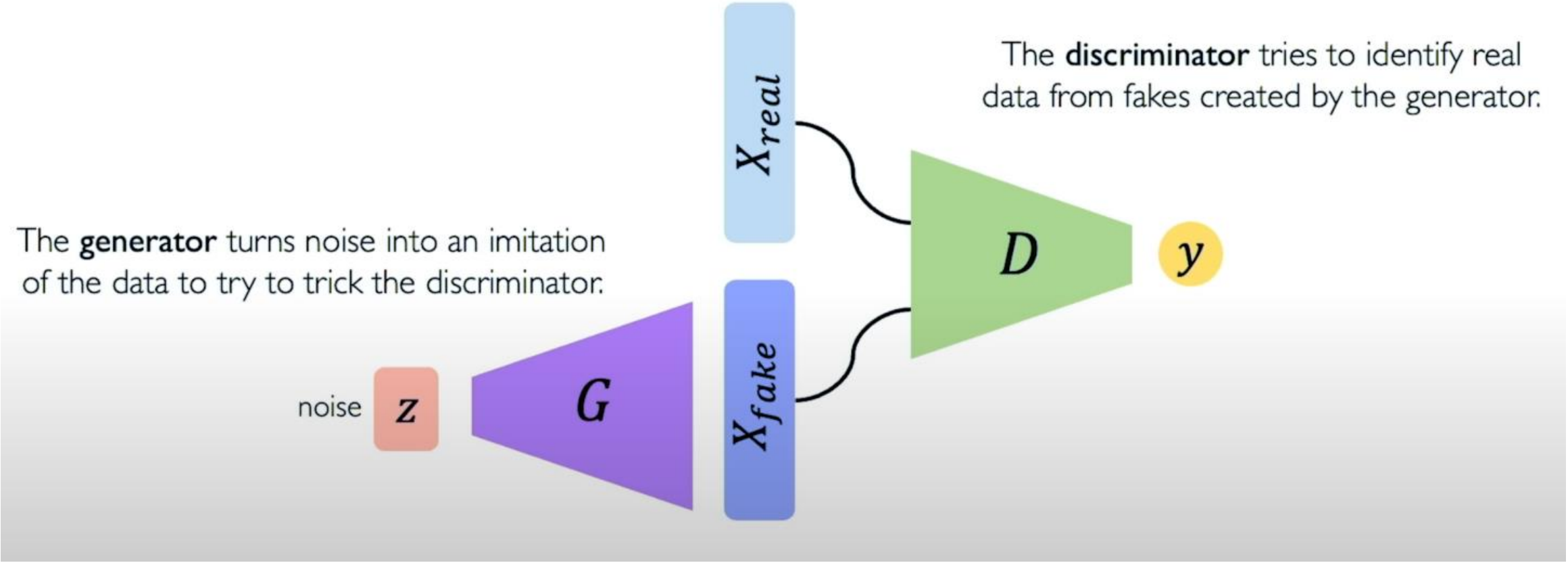
What if we just want to sample?

- Idea: Don’t explicitly model density, instead just sample to generate new instances
- Problem: want to sample from complex distribution – can’t do this directly!
- Solution: sample from something simple (e.g. noise), learn a transformation to the data distribution.



Generative adversarial networks (GANs)
Introduction

- Generative Adversarial Networks (GANs) are a way to make a generative model by having two neural network compete with each other.

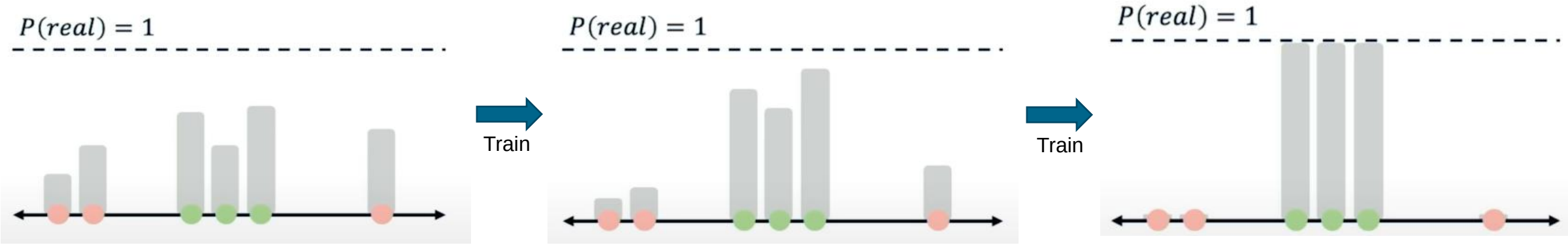


Generative adversarial networks (GANs)
Intuition: Iteration 1

- Generator starts from noise to try to create an imitation of the data

Generator

- Discriminator tries to predict what's real and what's fake

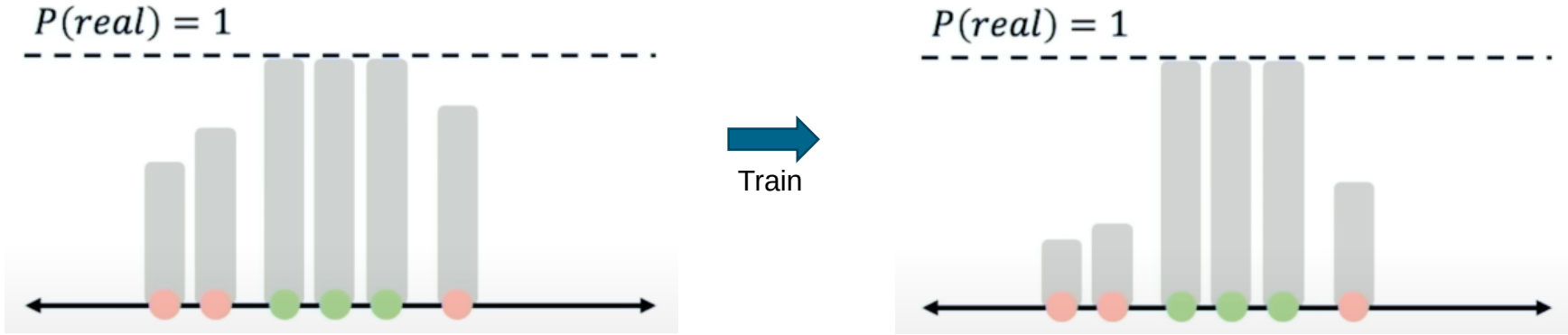


Discriminator

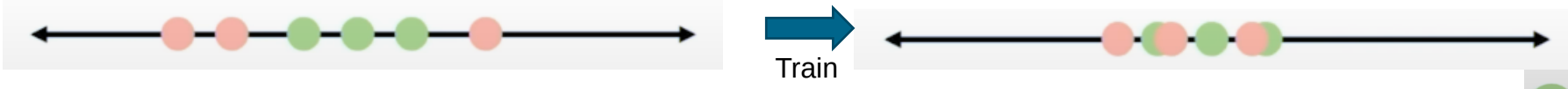
Generative adversarial networks (GANs)
Intuition: Iteration 2

- Generator tries to improve its imitation of the data

- Discriminator tries to predict what's real and what's fake

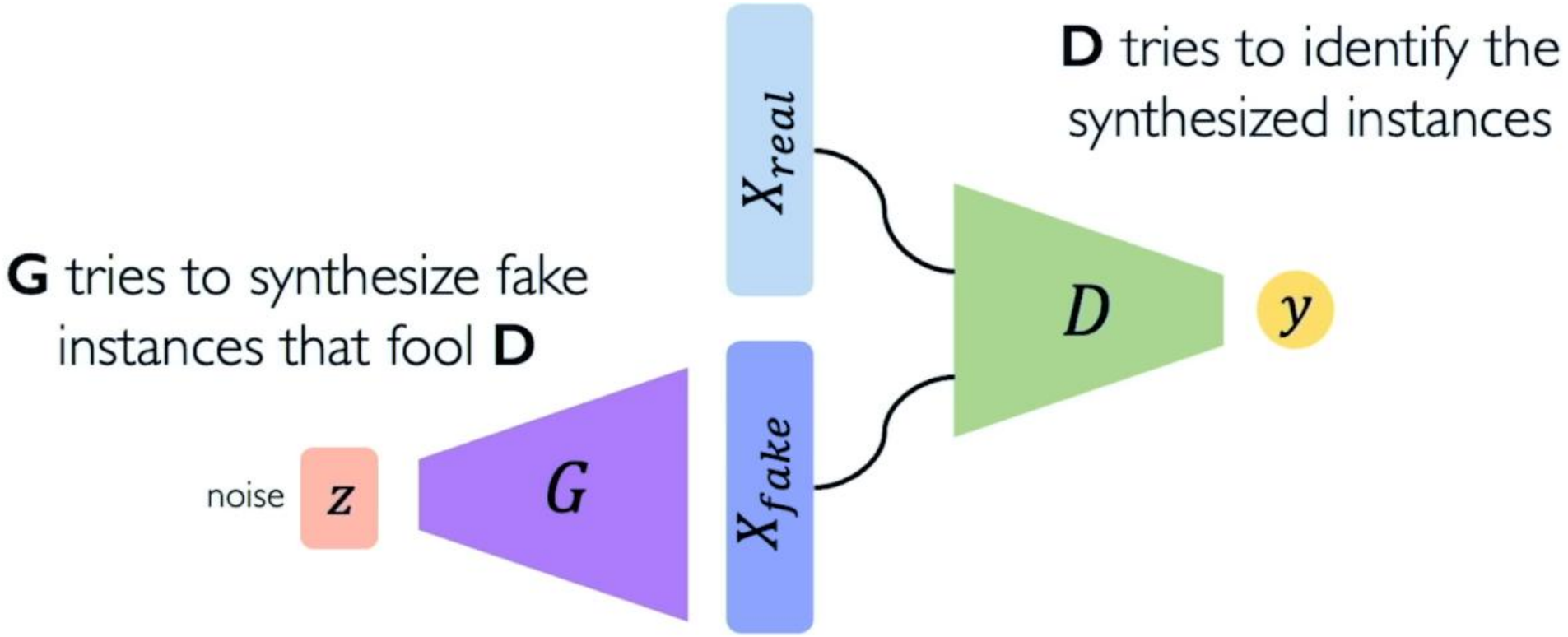


- Generator tries to improve its imitation of the data



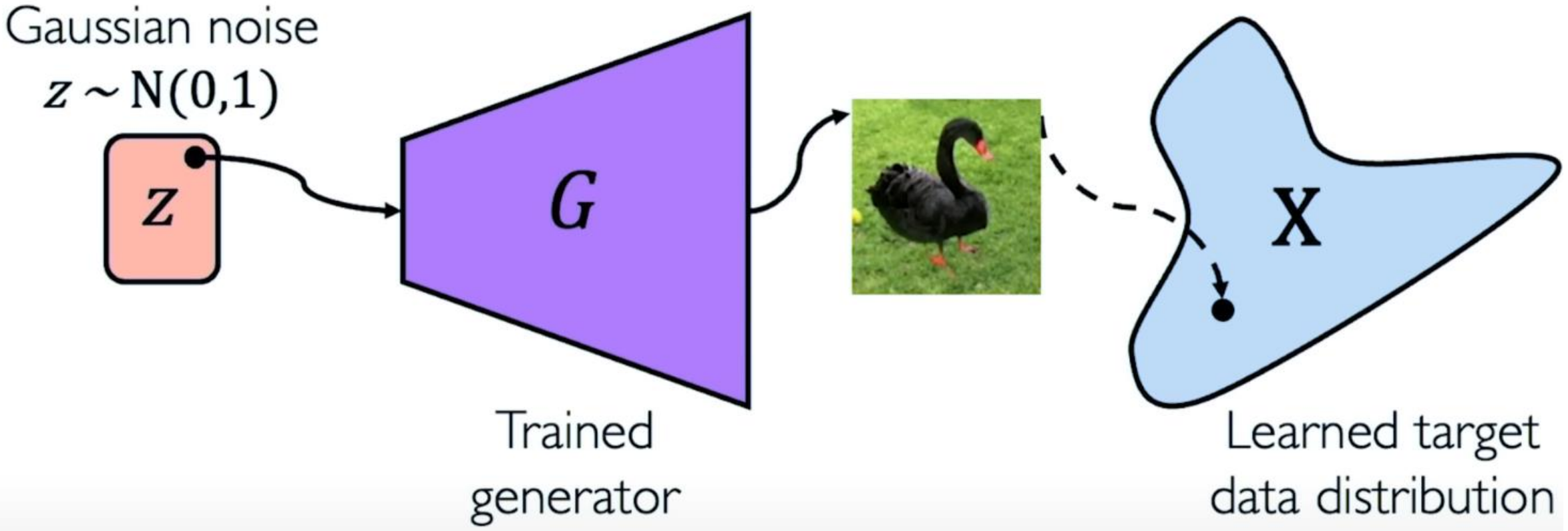
Generative adversarial networks (GANs)
Training

- Training: Adversarial objectives for D and G
- Global optimum: G reproduces the true distribution

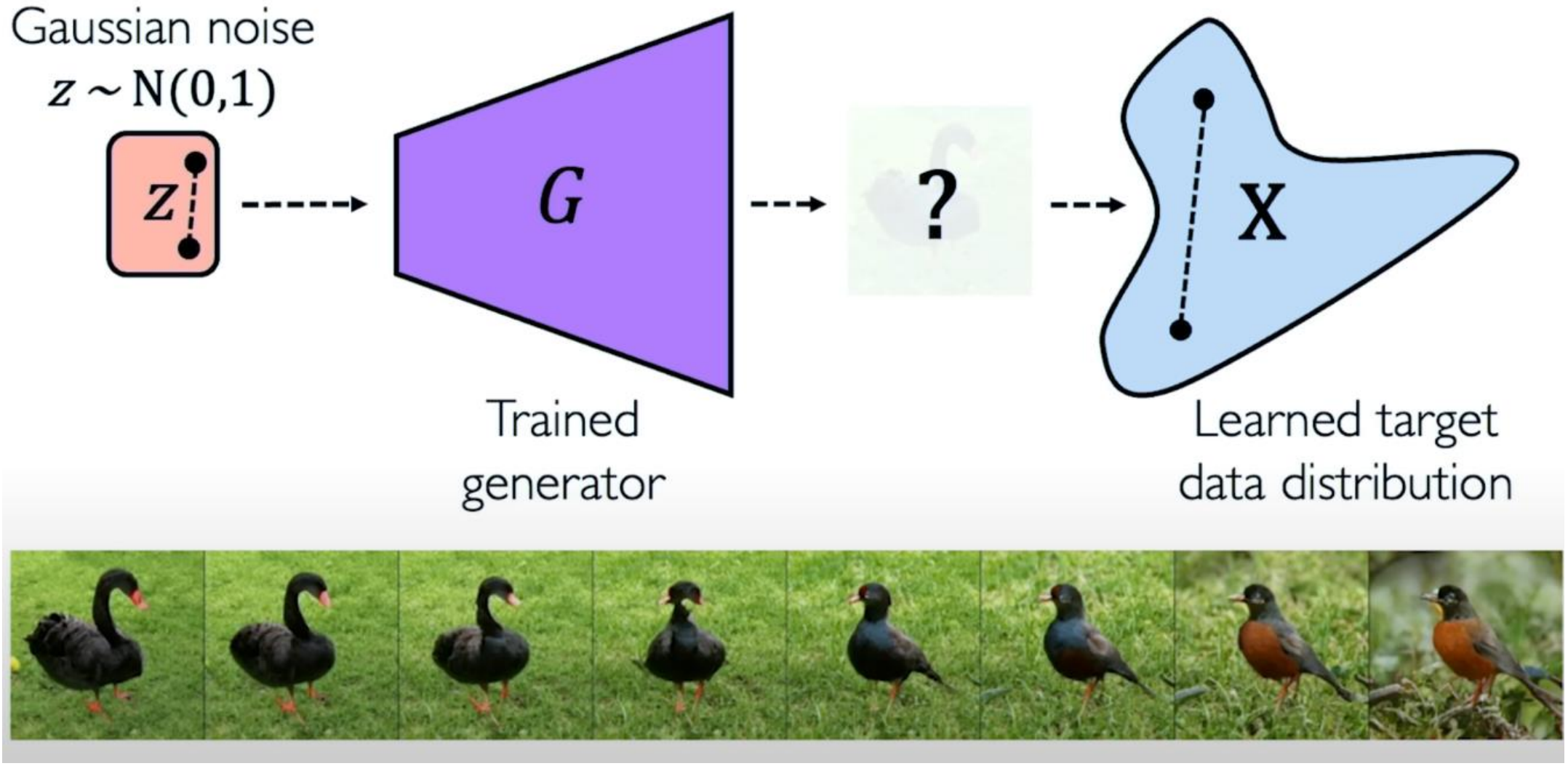


Generative adversarial networks (GANs)
Generating new data/ Sampling

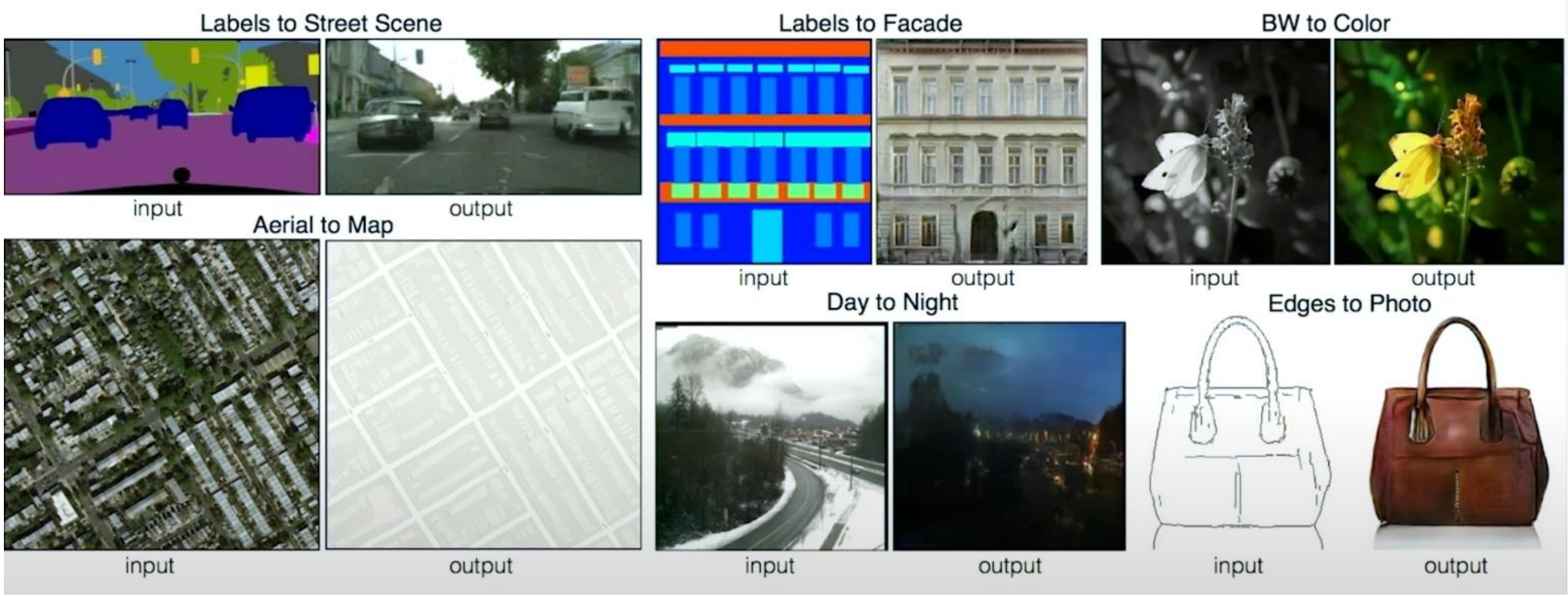
- After training, use generator network to create new data that's never been seen before



Generative adversarial networks (GANs)
Application – Distribution transformers



Generative adversarial networks (GANs)
Application – Paired translation



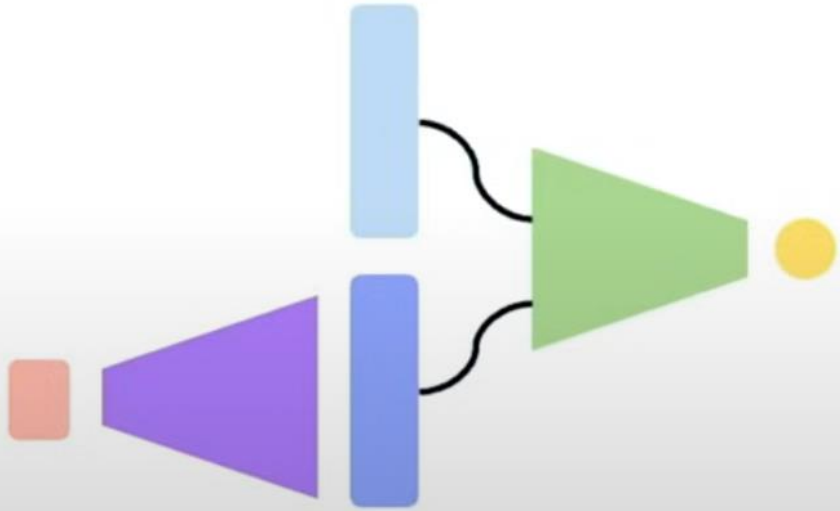
Generative modeling
Latent variables Models – GANs - Summary

Autoencoders and Variational Autoencoders (VAEs)

Learn lower-dimensional **latent space** and **sample** to generate input reconstructions

Generative Adversarial Networks (GANs)

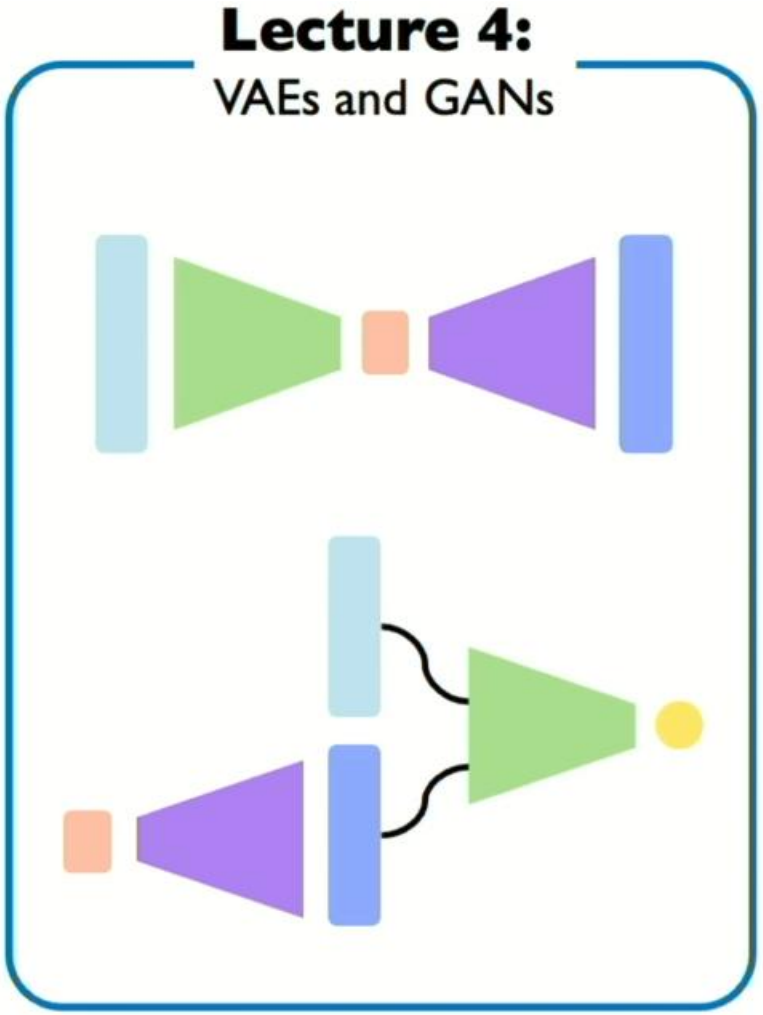
Competing **generator** and **discriminator** networks



Generative modeling: New frontiers

- Diffusion models
- Large language models

Generative modeling Landscape



Limitations

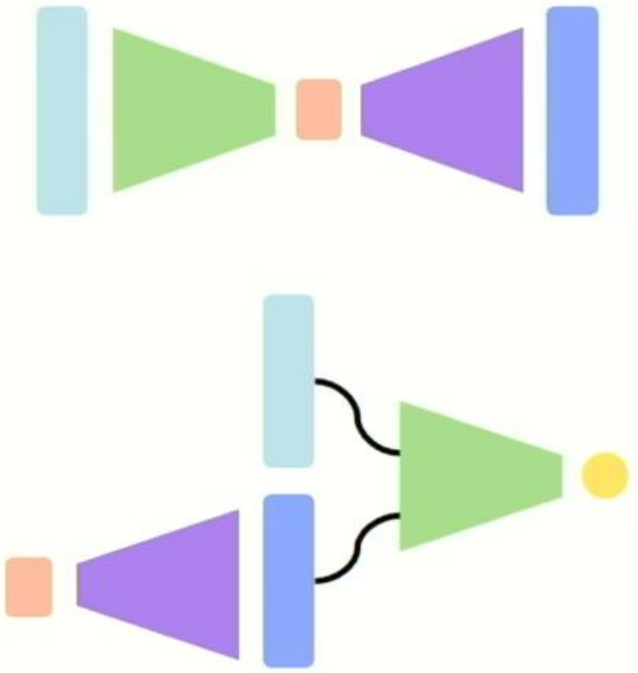
- Mode collapse
- Generating OOD
- Hard to train

Challenges

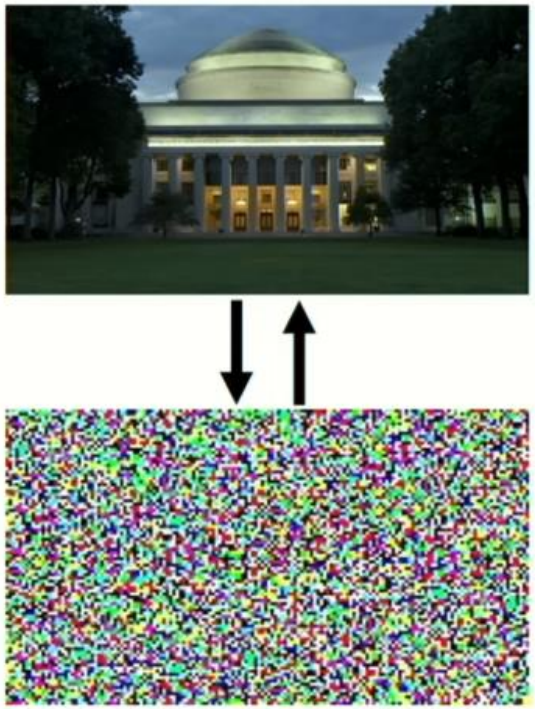
- Stability
- Efficiency
- Quality
- Novelty

Generative modeling Landscape

Lecture 4:
VAEs and GANs



Diffusion Models

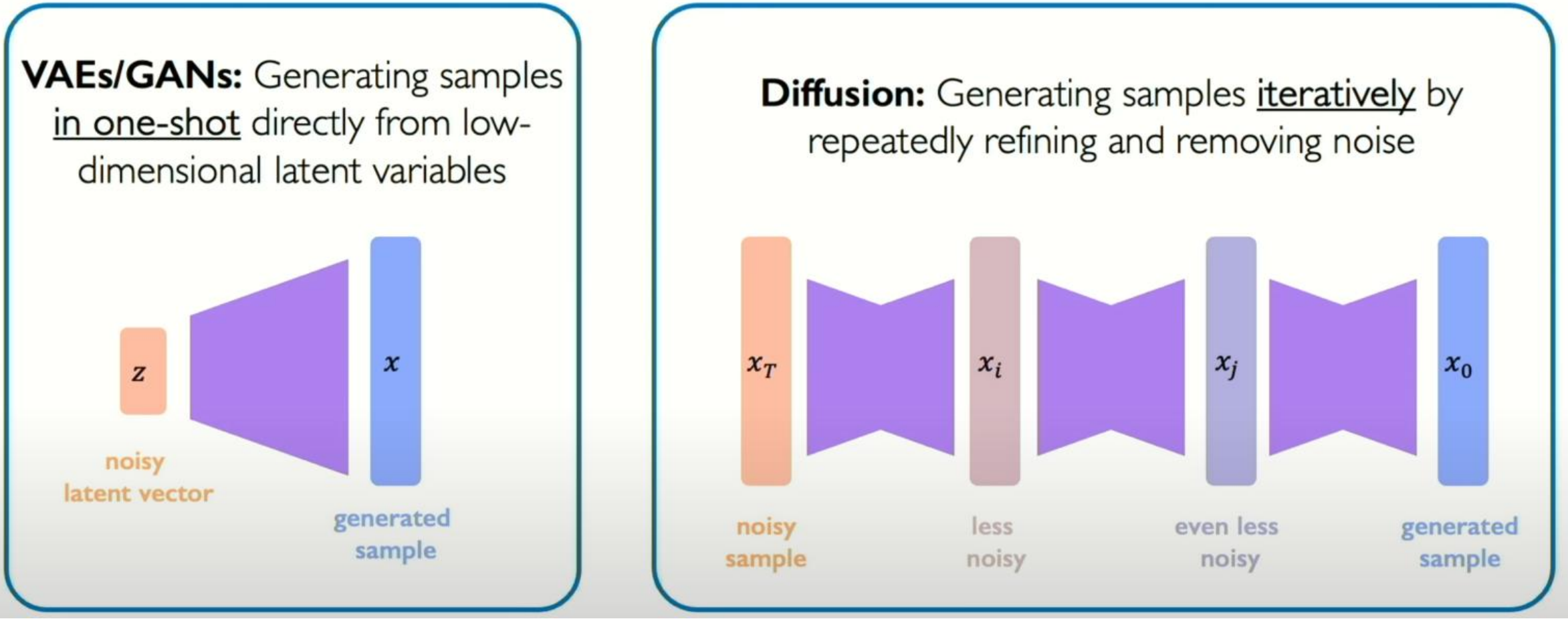


Text-to-Image

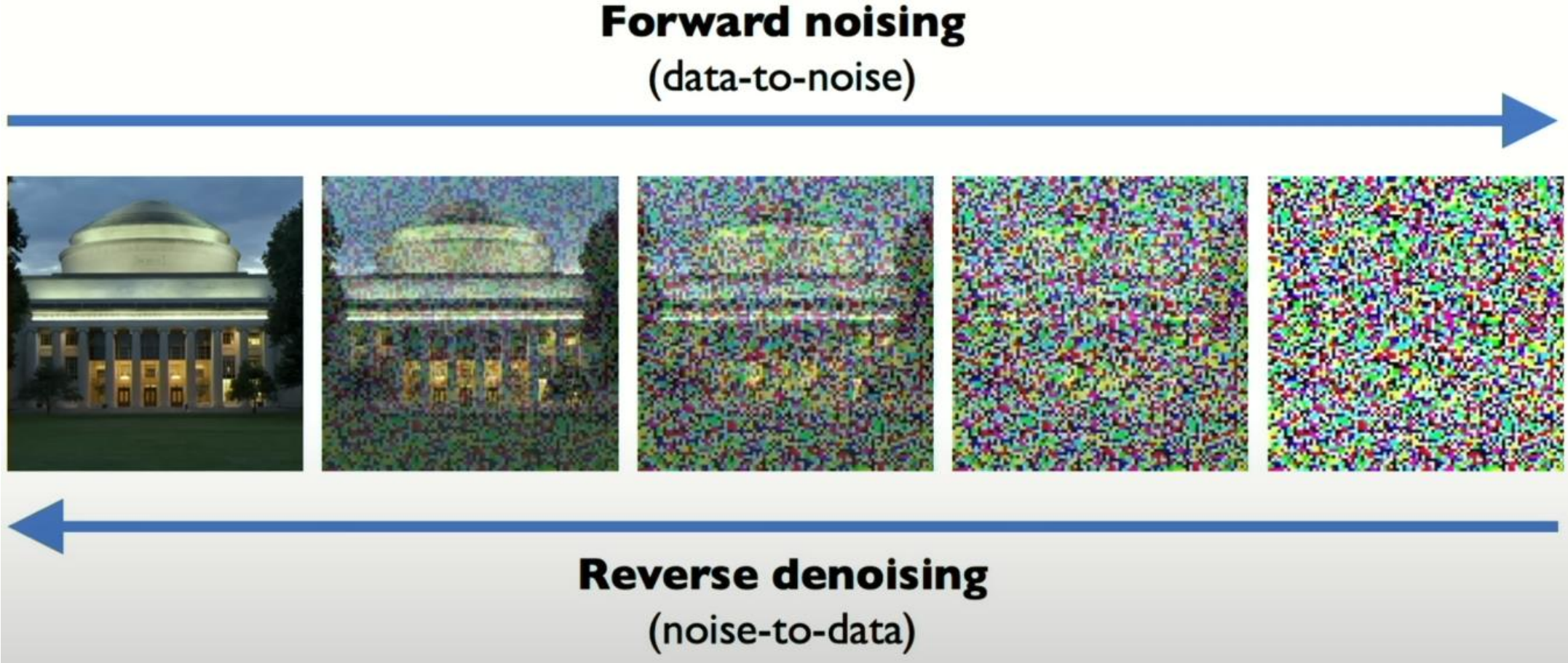
"Two cats doing research"

Generative modeling
Diffusion models

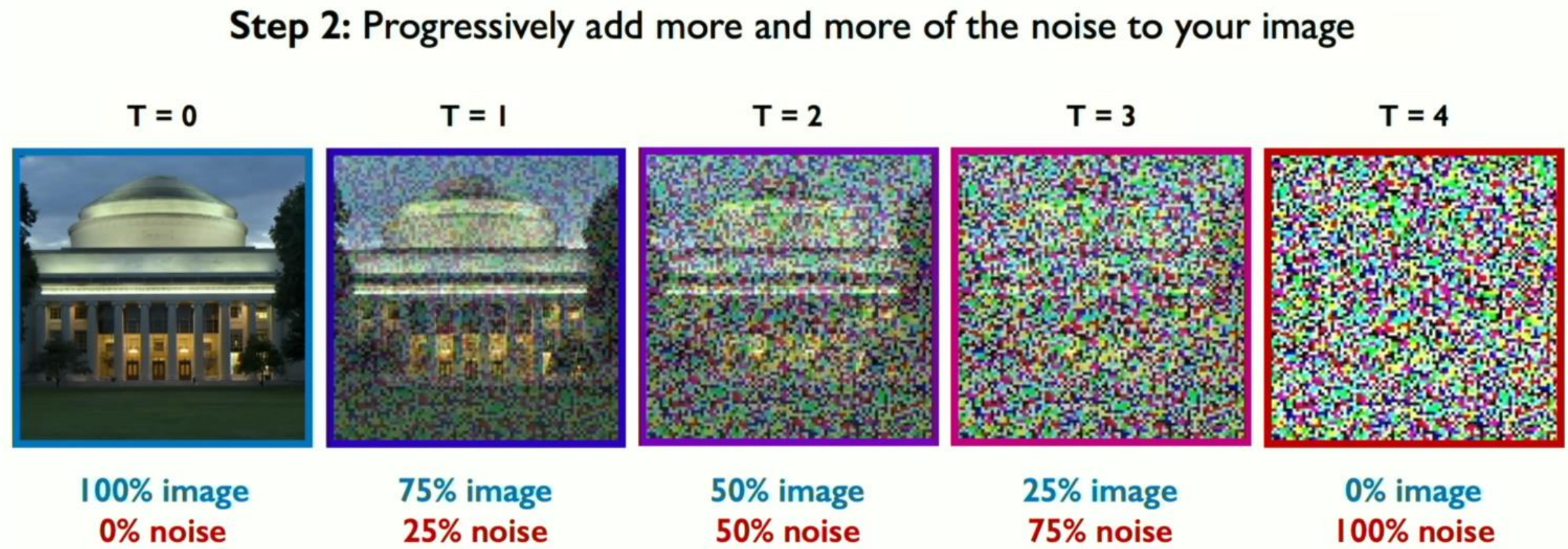
Generative modeling
Diffusion models



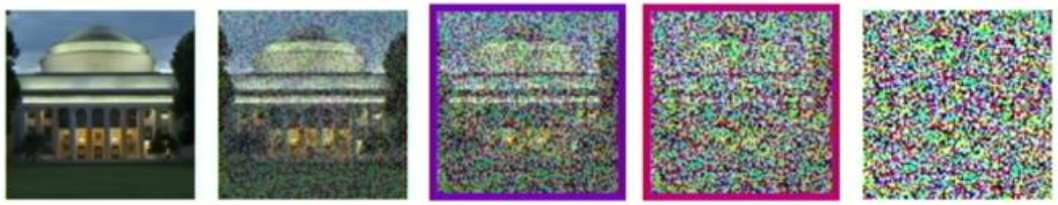
Diffusion models
Diffusion process



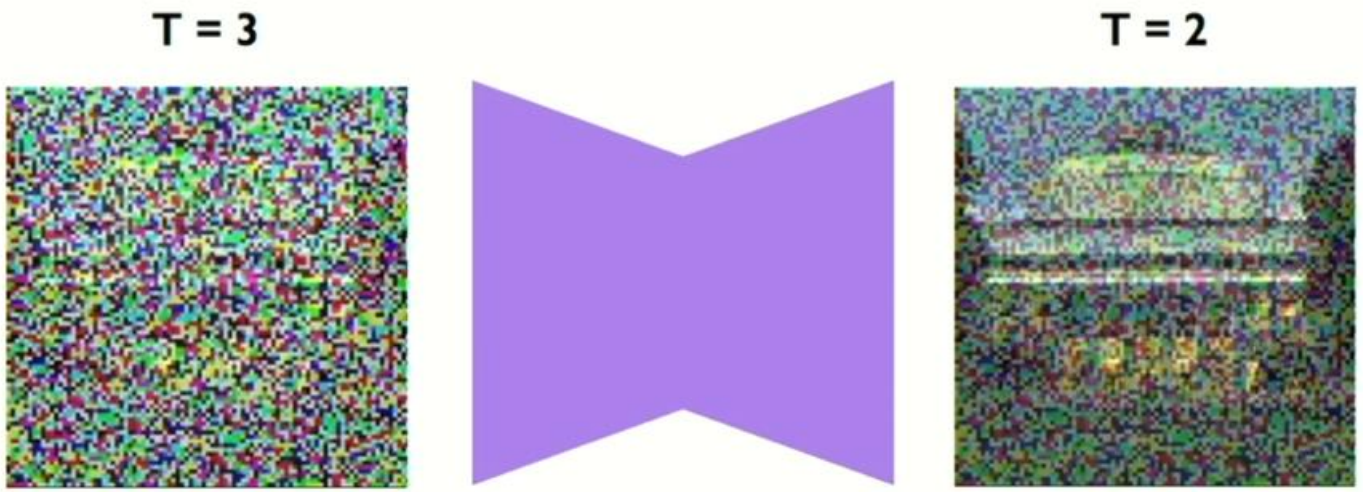
Diffusion models
Forward noising



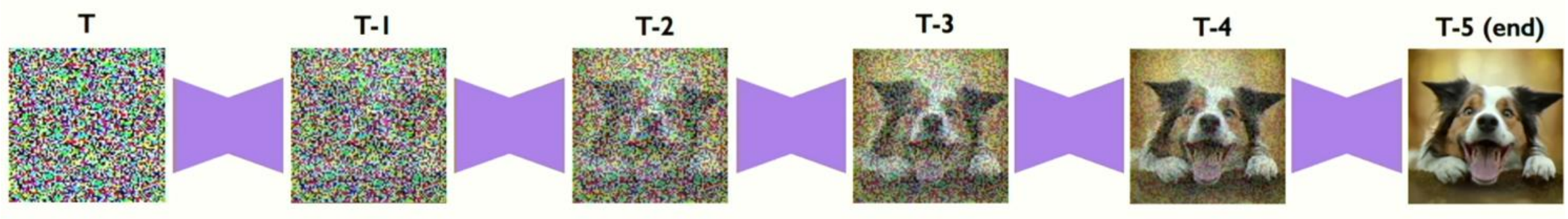
Diffusion models
Reverse denoising



Goal: Given image at **T**, can we **learn** to estimate image at **T-1**?



Diffusion models
Generating new samples



Diffusion models
Text to image generation

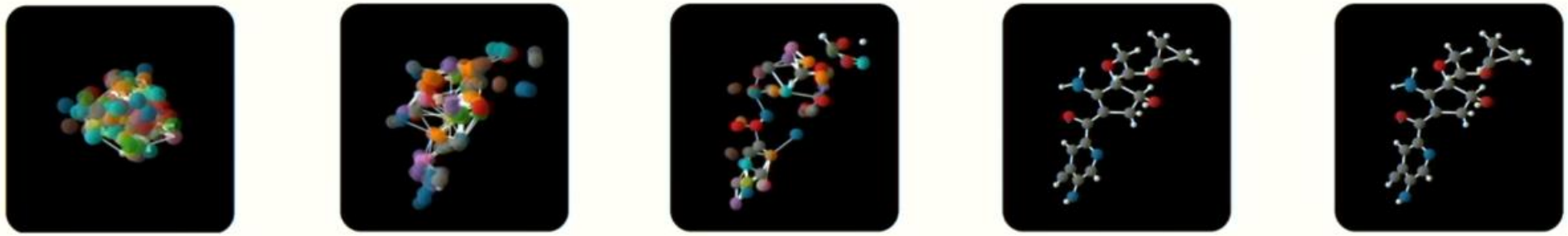
"a painting of a fox sitting in a field at sunrise in the style of Claude Monet"

"an ibis in the wild, painted in the style of John Audubon"

"close-up of a snow leopard in the snow hunting, rack focus, nature photography"

Diffusion models
Beyond images: Molecular Design

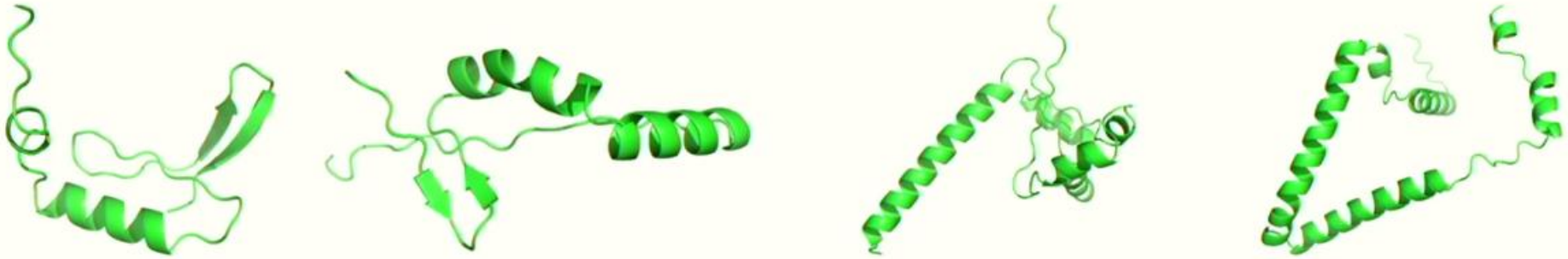
Chemistry: Generating Molecules in 3D



Noise ---------- Molecule

Hoogeboom+ *ICML* 2022, Jing+ *NeurIPS* 2022.

Biology: Generating Novel Proteins



Wu+ arXiv 2022, Anand+ *arXiv* 2022, Trippe+ *arXiv* 2022, Jumper+ *arXiv* 2022, and more …

Curriculum

Next:

- Hands-on III

Imprint

Topic:	**Introduction to Deep Learning** Part III – Deep Generative Models
Date:	2025-11-14
Author:	Auliya Fitri, Sai Vemuri, Sreerag Naveenachandran
Institute:	Data Science
Image sources:	All images “DLR (CC BY-NC-ND 3.0)” unless otherwise stated