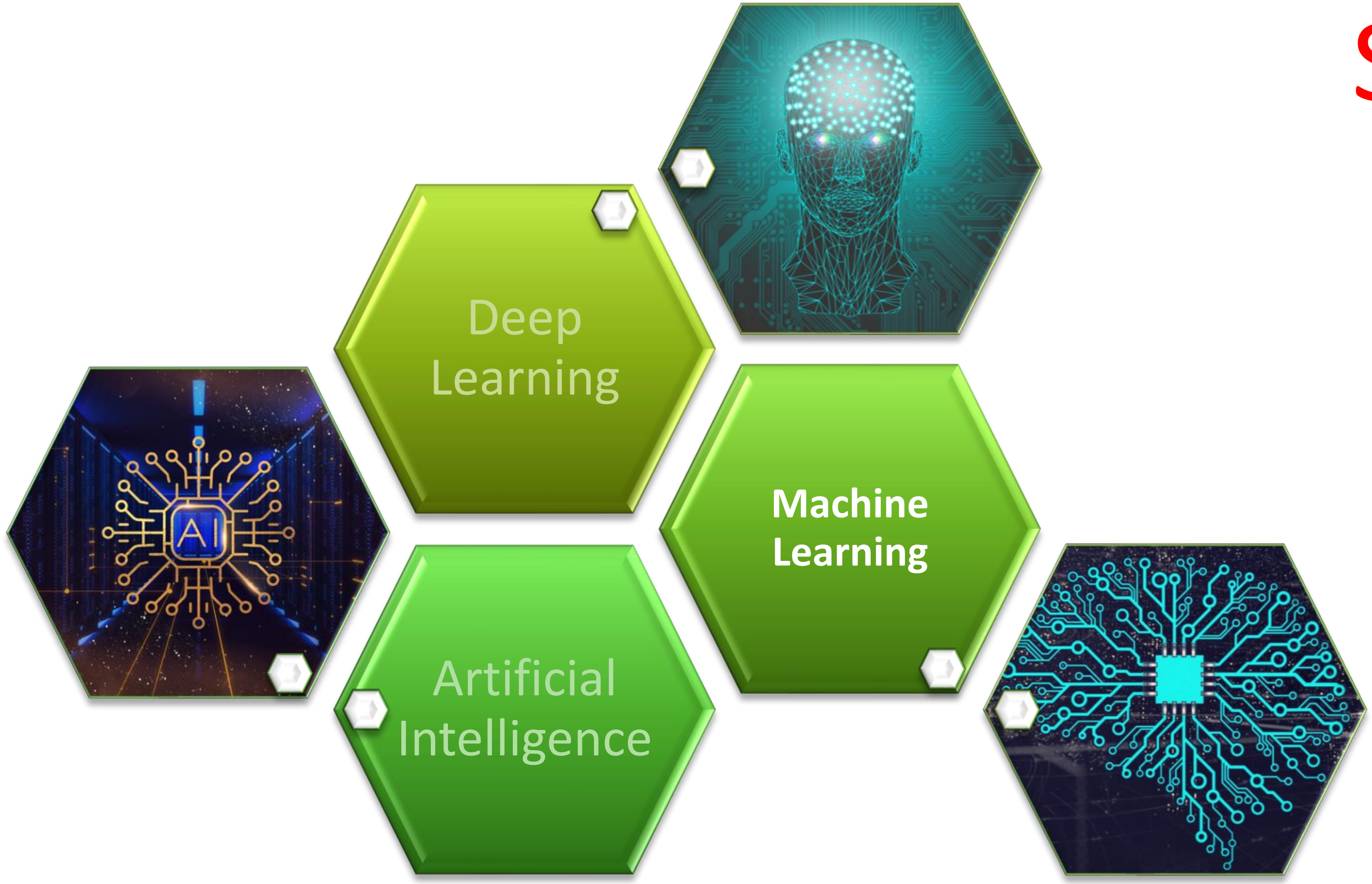
Artificial Intelligence Summer Internship/USRF

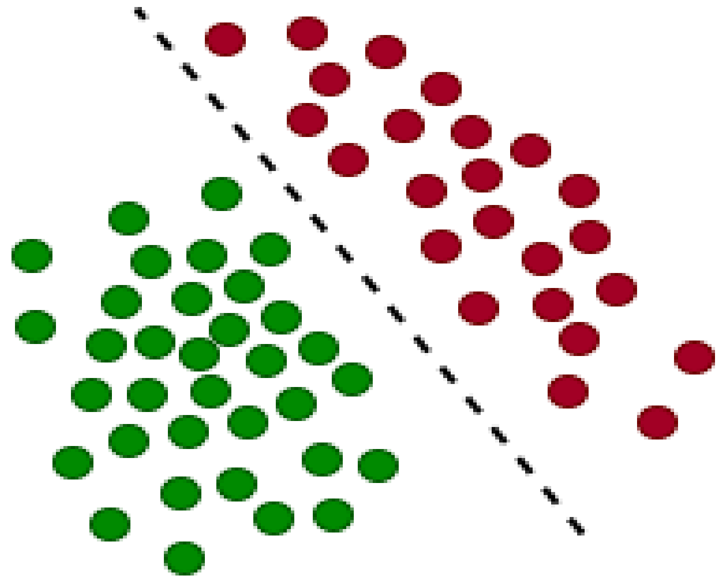


- **Variational Autoencoders**

Generative Models vs. Discriminative Models

Discriminative models:

- Learn the characteristics that distinguish different classes in a dataset.

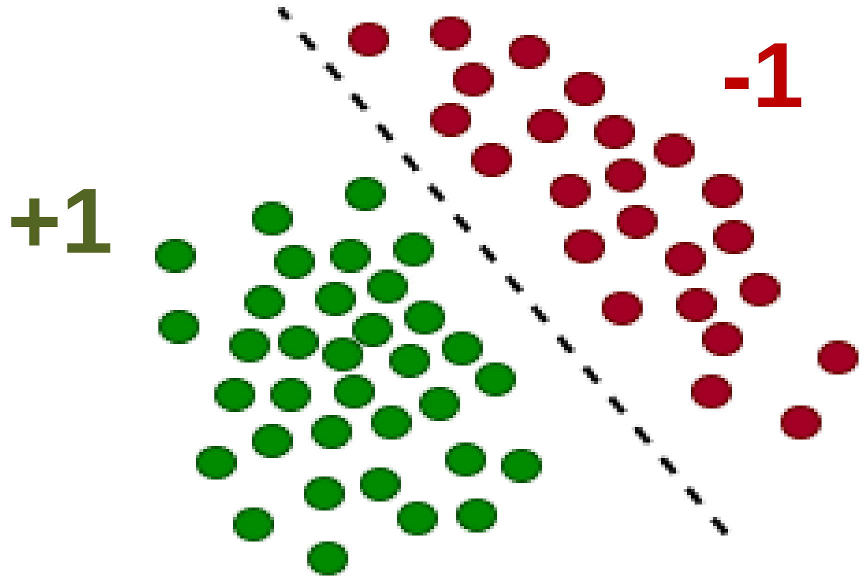


Discriminative

Generative Models vs. Discriminative Models

Discriminative models:

- Learn the characteristics that distinguish different classes in a dataset.



Discriminative

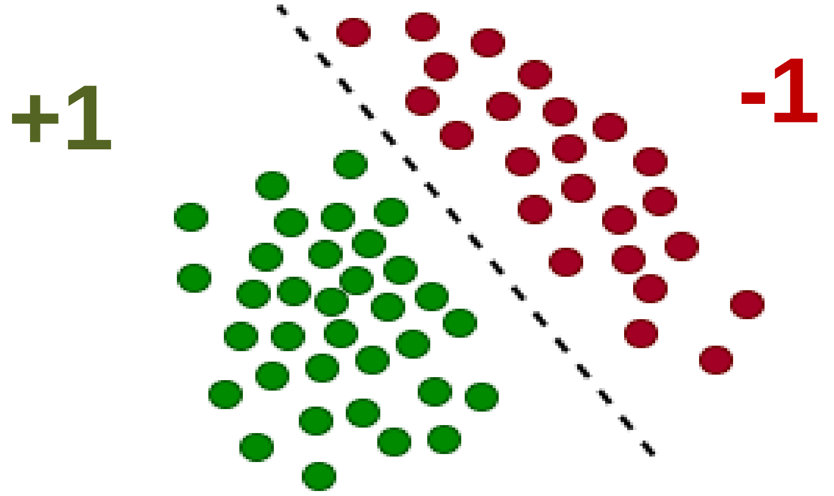
Generative Models vs. Discriminative Models

Discriminative models:

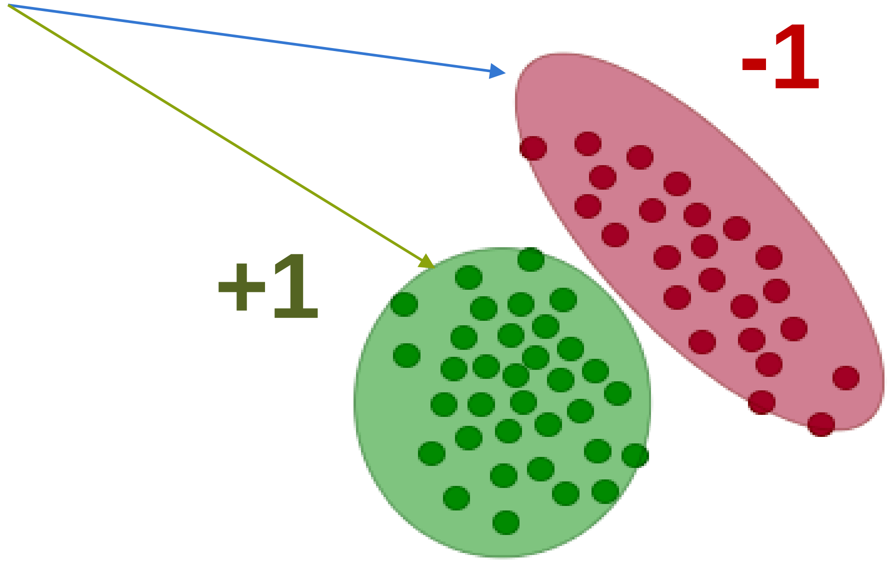
- Learn the characteristics that distinguish different classes in a dataset.

Generative models:

- Learn the fundamental distribution of each class inside a dataset.



Discriminative



Generative

Why Generative Models?

Image Super Resolution

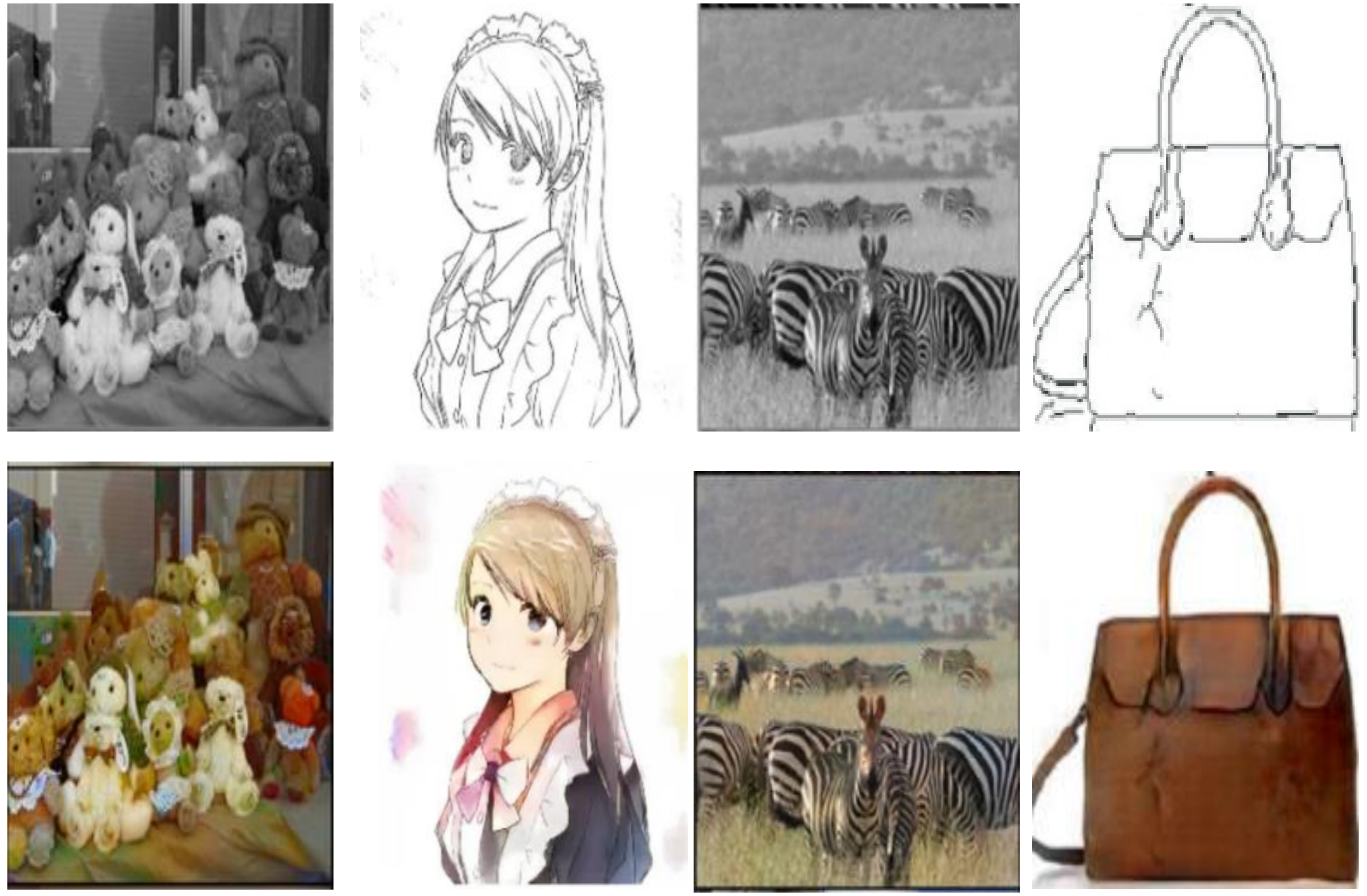
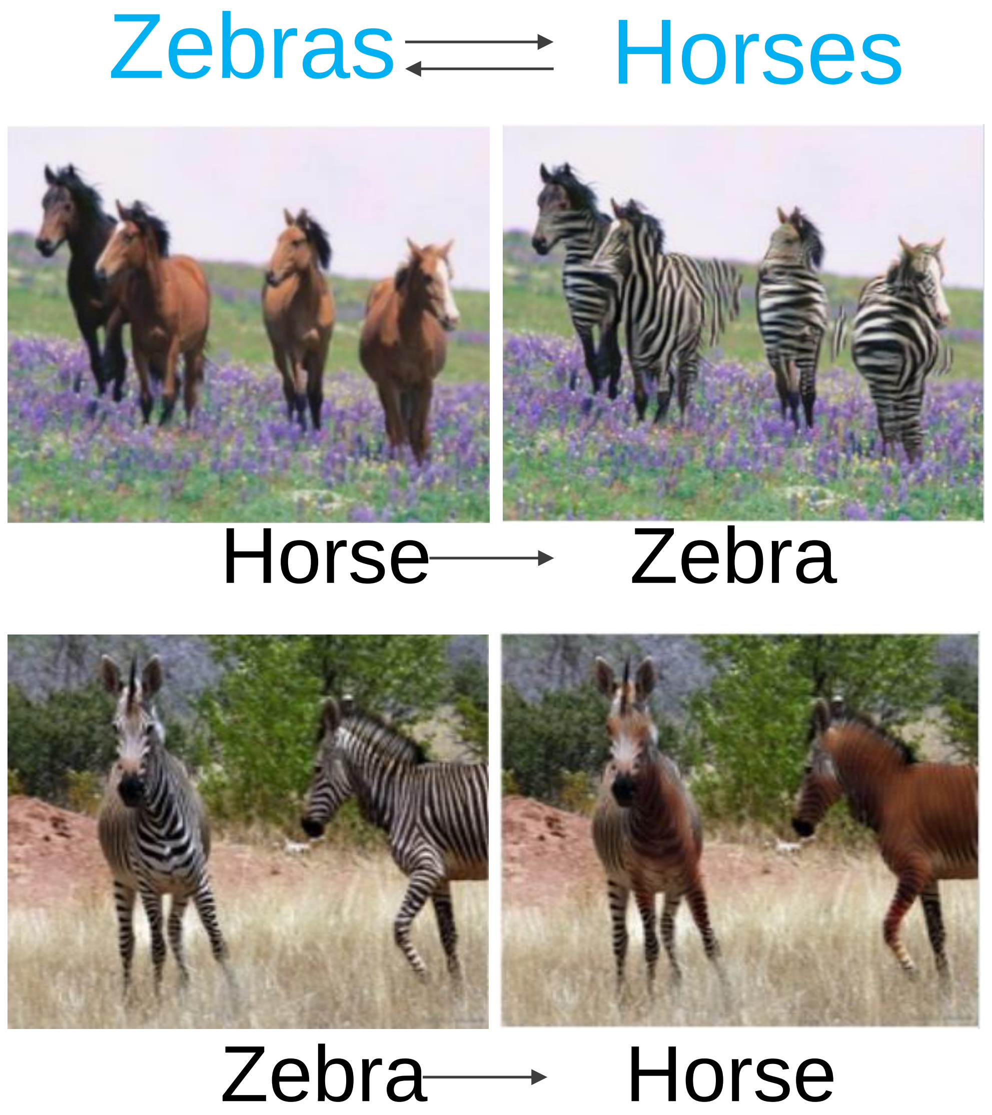
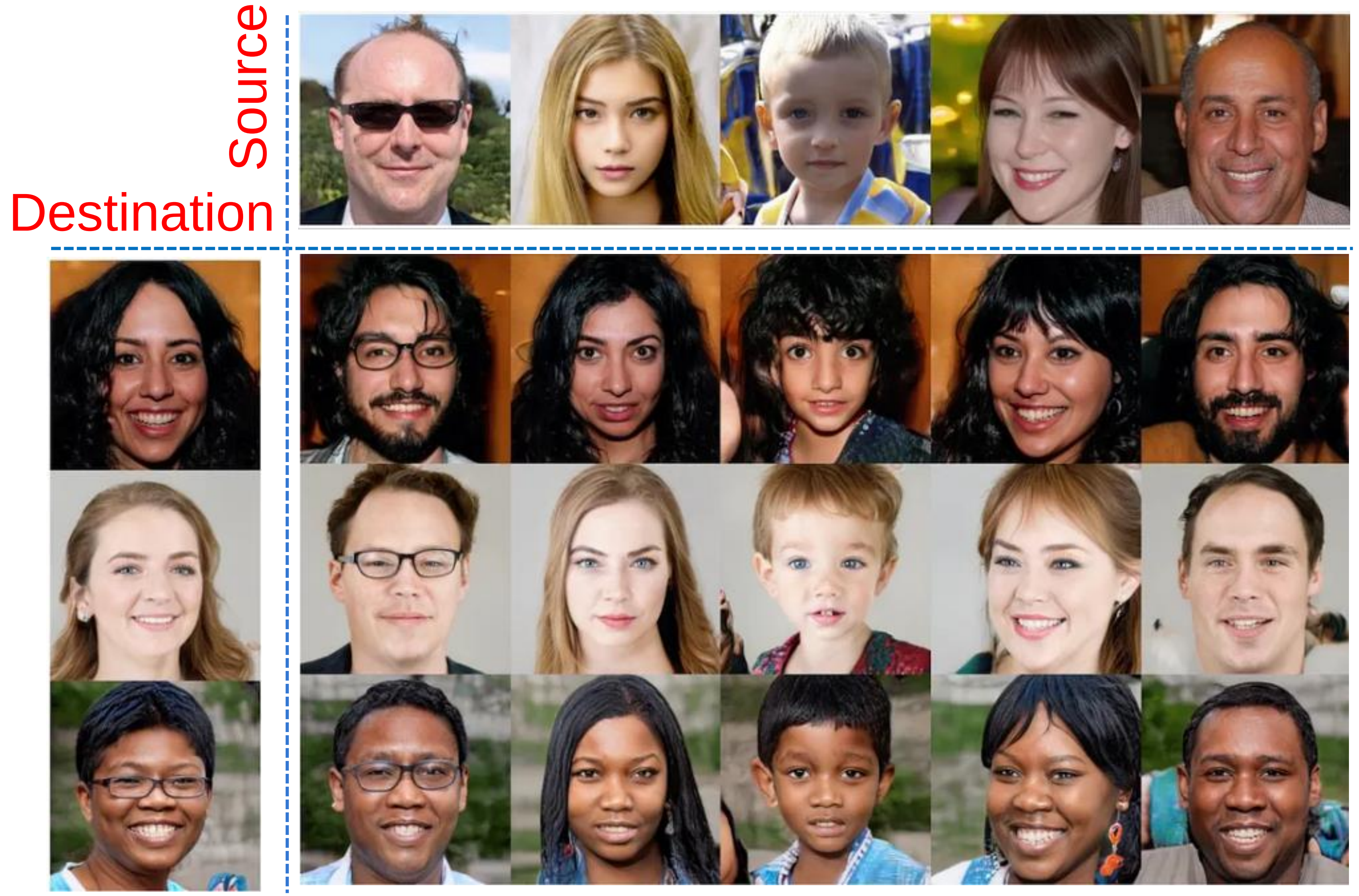


Image Colorization

Ref. https://arxiv.org/abs/1611.07004

Why Generative Models?

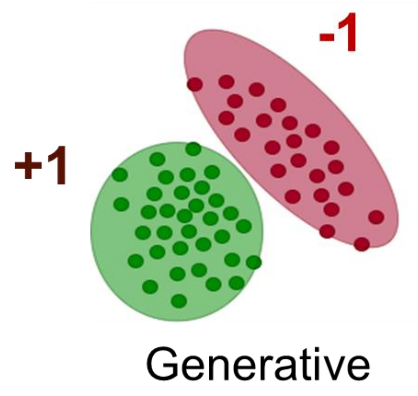




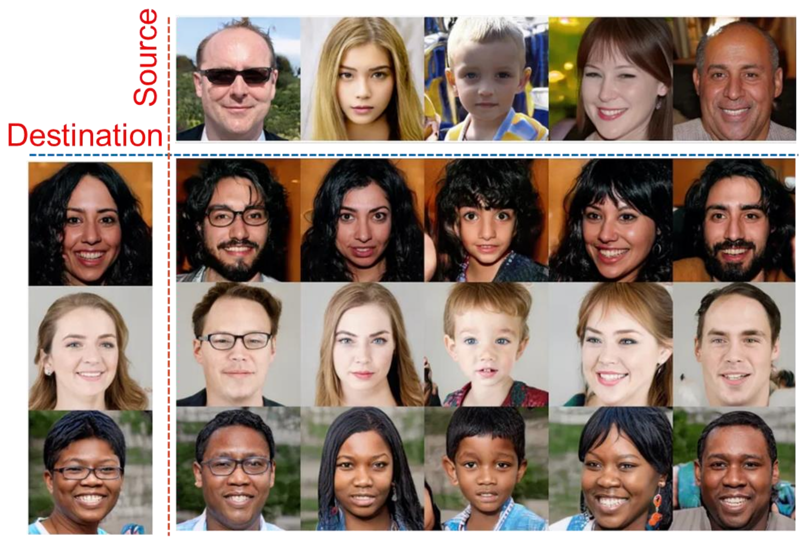
Cross-domain Image Translation

Generating Realistic Face Images

Why Generative Models?

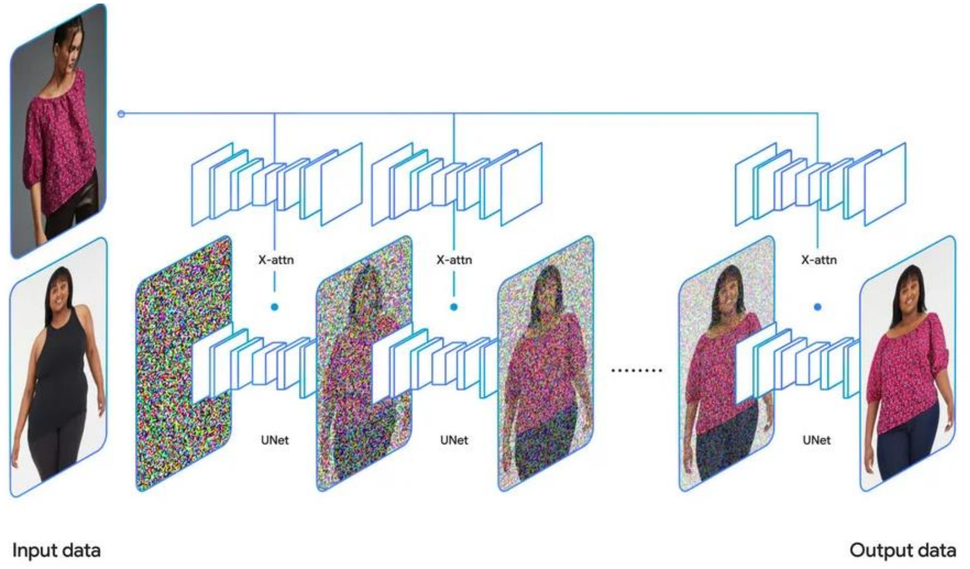


- Learn significant generalizable latent characteristics.



- Augment small datasets.

- Facilitate mixed-reality applications like Virtual Try-on.



- Many more..

Autoencoder generative?

Before we start

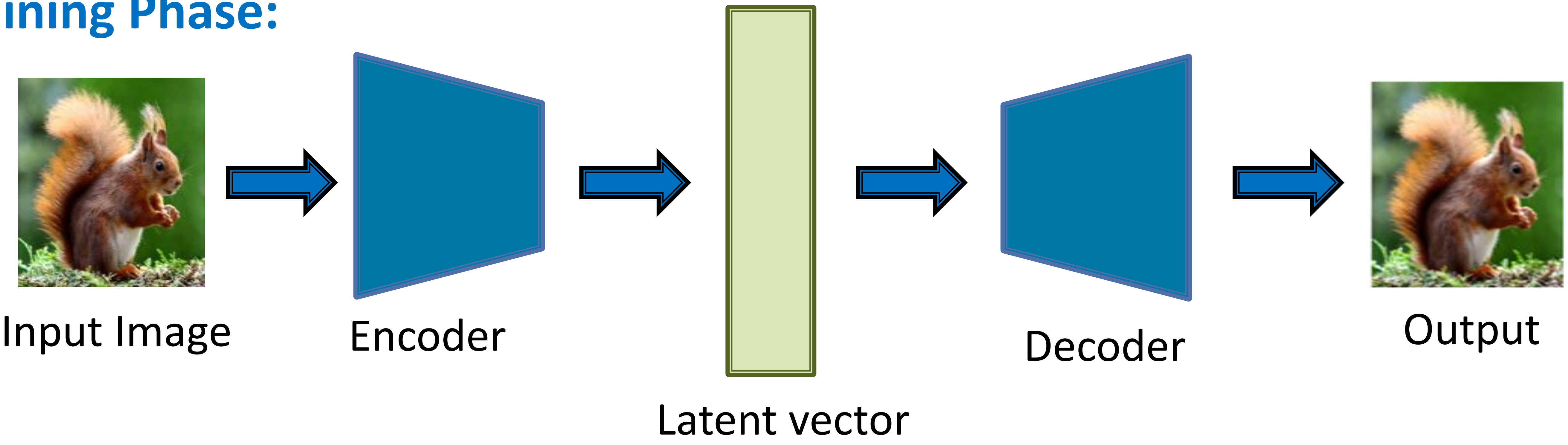
- **Question?**
  - Are the previous Autoencoders generative model?

Properties of Autoencoders

Data-specific:

Autoencoders are only able to compress data like what they have been trained on.

Training Phase:

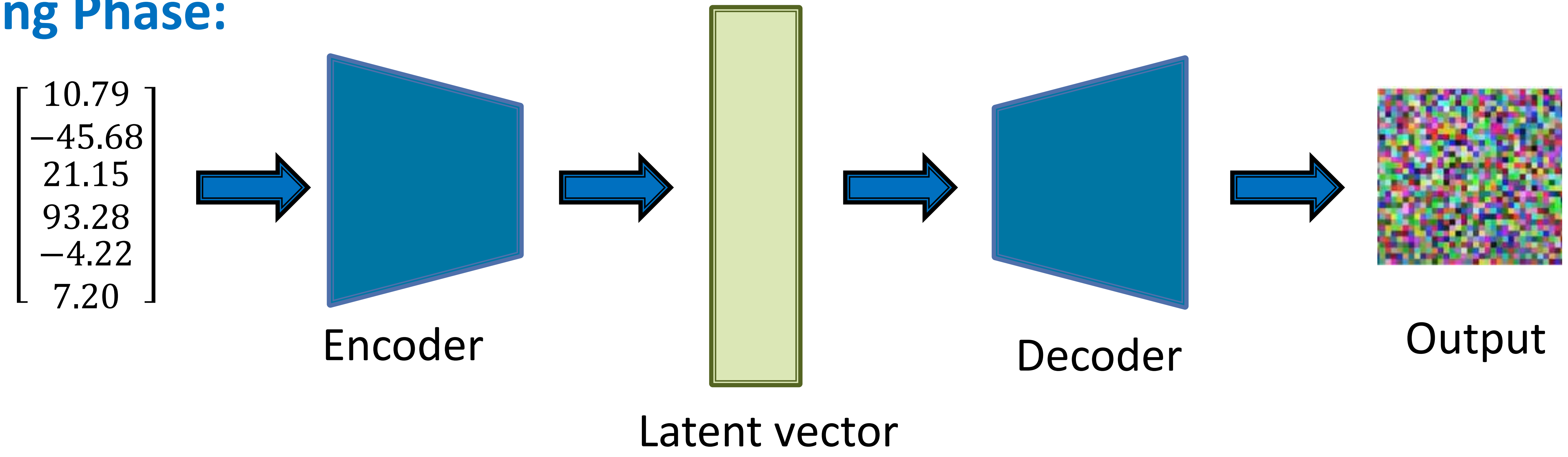


Properties of Autoencoders

Data-specific:

Autoencoders are only able to compress data similar to what they have been trained on.

Testing Phase:



Autoencoder generative? NO

Before we start

- **Question?**
  - Are the previous Autoencoders generative model?

  (NO, The compressed latent codes of autoencoders are not prior distributions, autoencoder cannot learn to represent the data distribution). Autoencoders learn the feature representation.

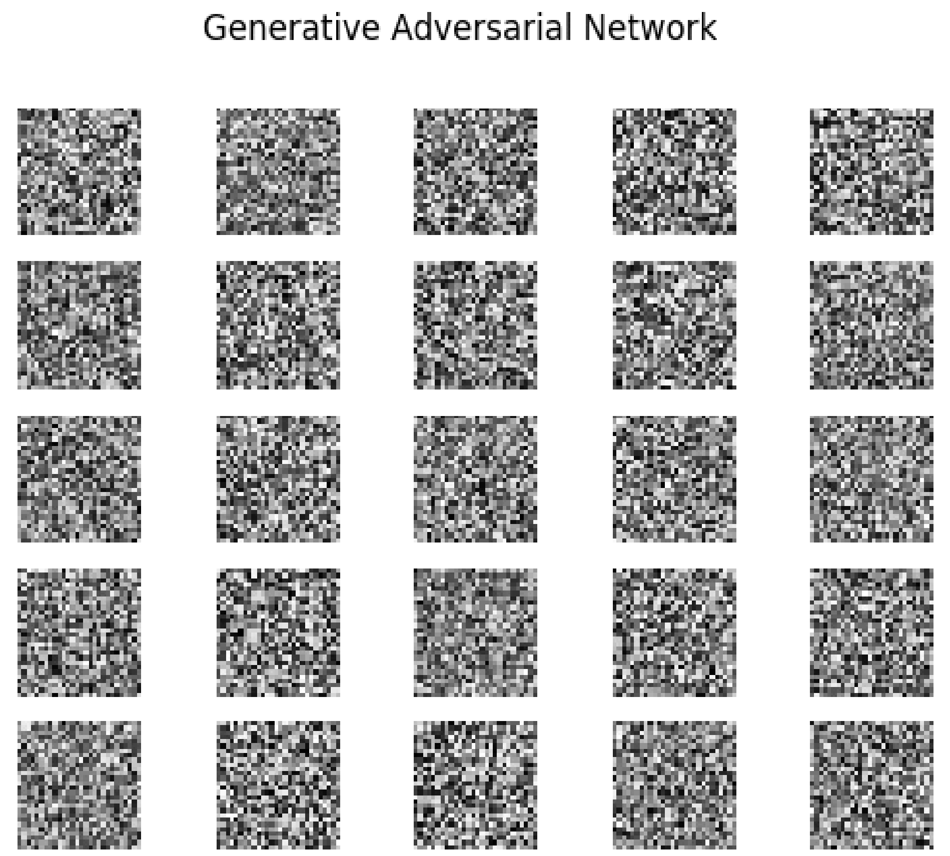
- Generative models are typically meant for modelling data distribution, P(X).

Generative Models

Goal: Given a training data, generate data samples similar to the one in the training set

- Assuming training data $X = \{x_1, x_2, \ldots, x_n\}$ comes from an underlying distribution $P_{data}(X)$, and a generative model samples data from a distribution $P_{model}(X)$.

- Aim is to **minimize** some notion of **distance** between $P_{data}(X)$ and $P_{model}(X)$.

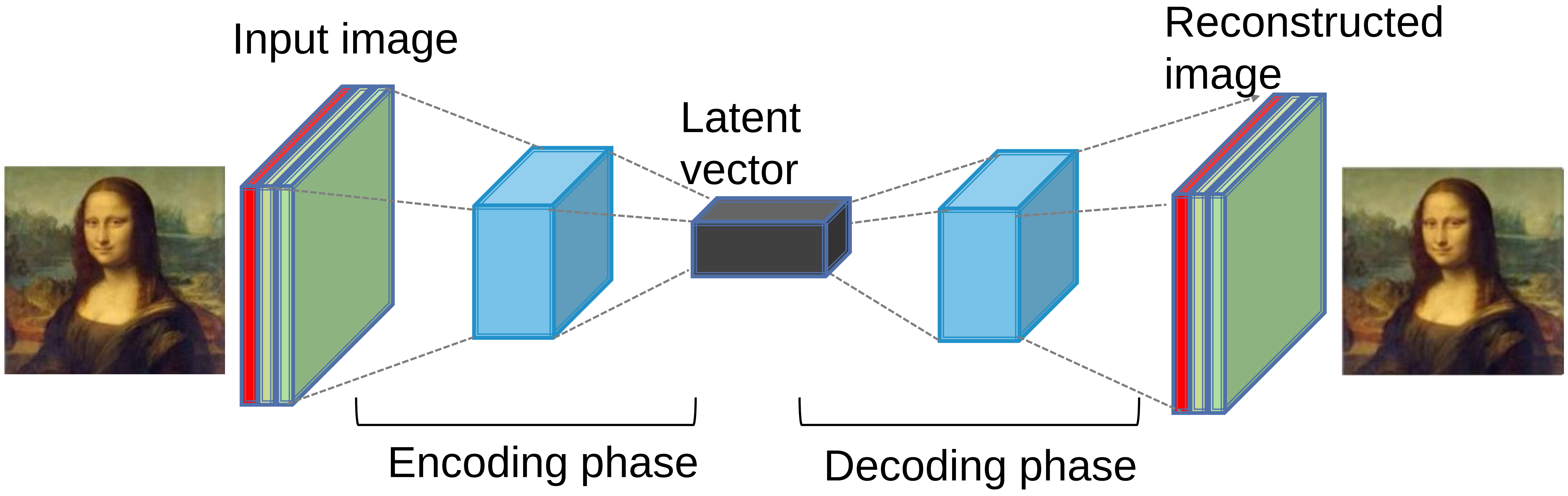


Generation of hand written digits over several iterations

Ref. https://github.com/eriklindernoren/Keras-GAN

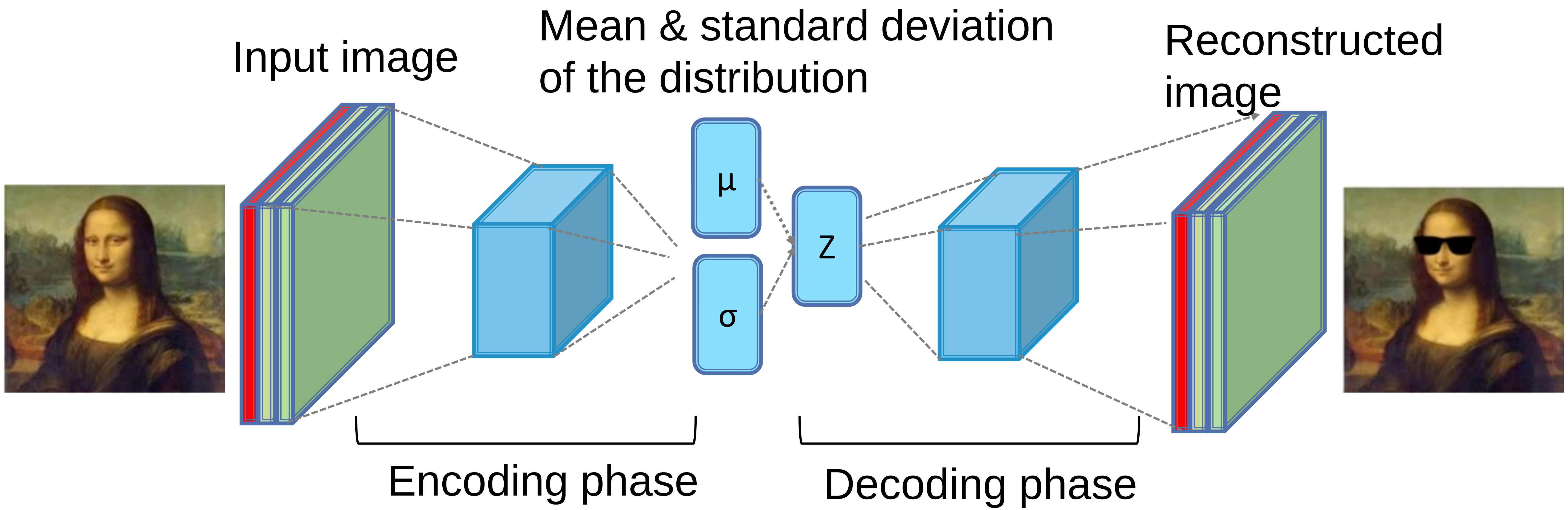
Variational Autoencoders (VAE)

Autoencoder



Autoencoders learn the feature representation
They are Data-specific

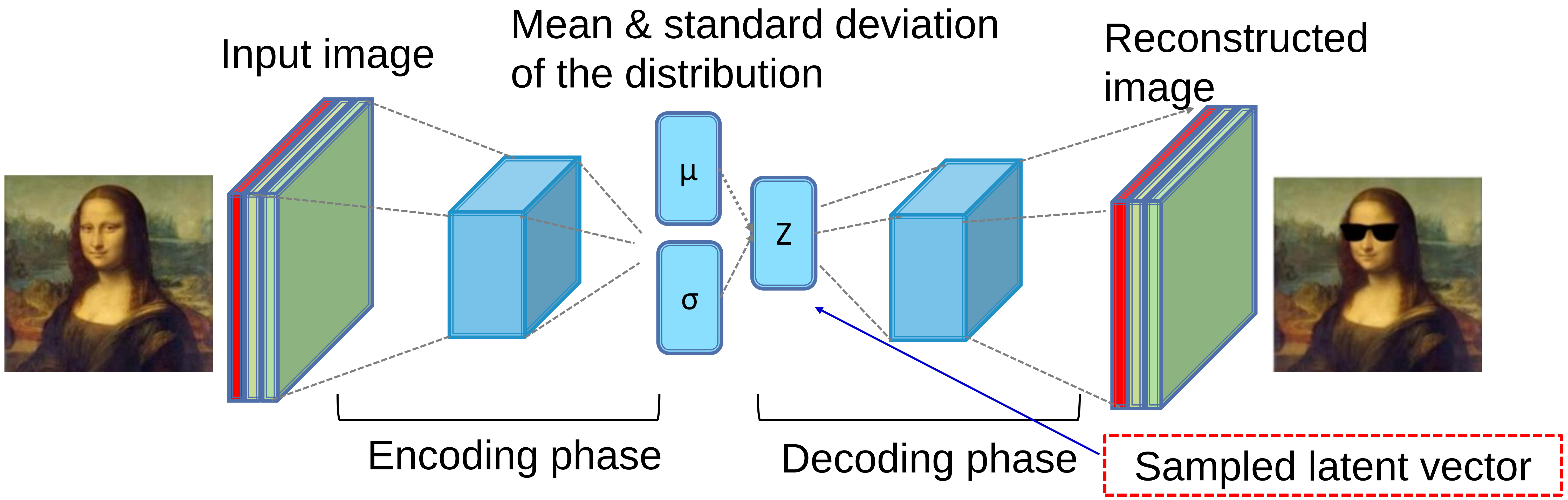
Variational Autoencoder



- Instead of learning the latent variables z directly for each input, VAE learns a mean and a variance associated with that latent variable.
- Mean and variance parameterize the probability distribution for that latent variable

https://www.youtube.com/watch?v=BUNl0To1IVw **(start from 17:19)**
https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Variational Autoencoder



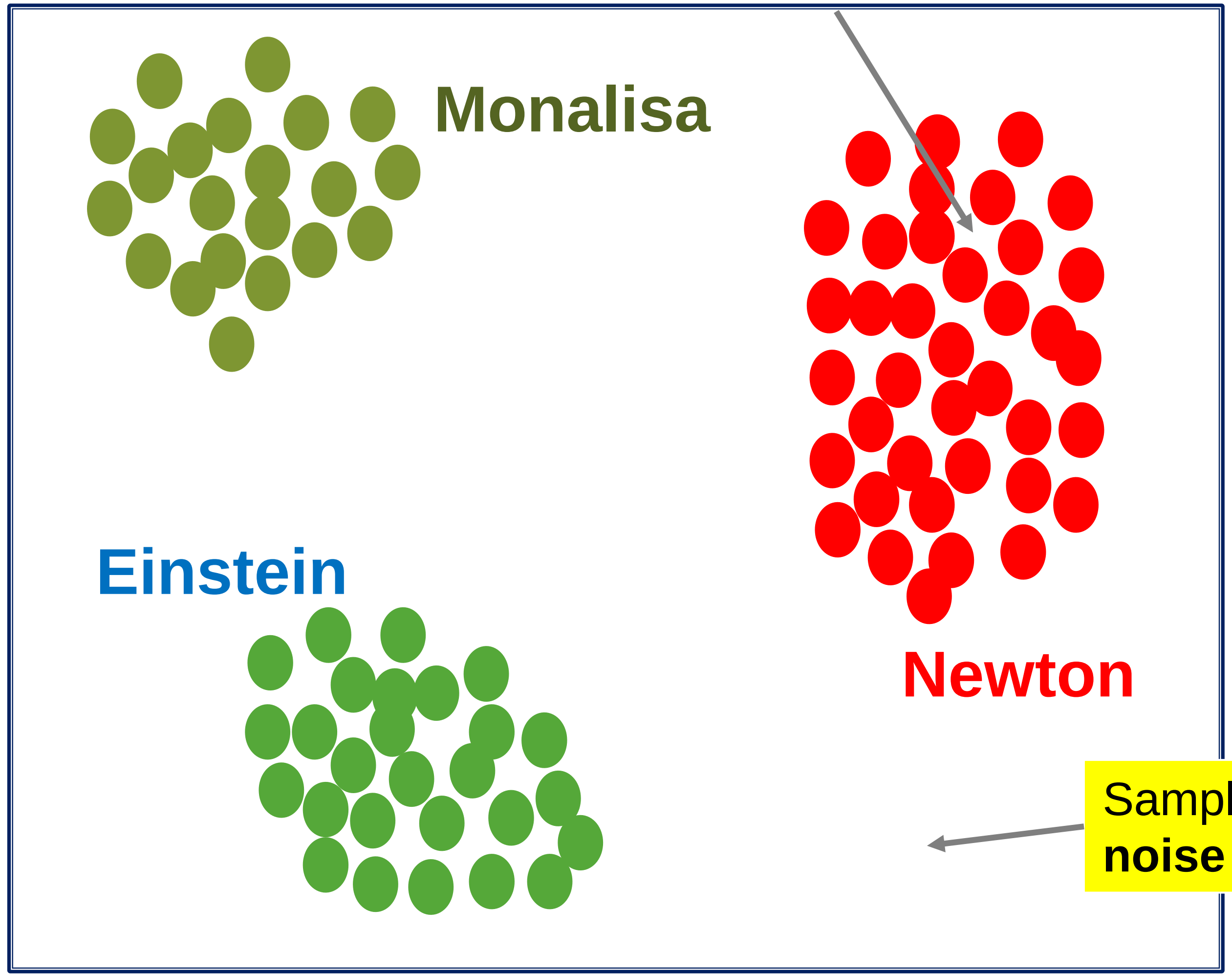
- Input to decoder is sampled from the probability distribution of the latent variable
- generate new data that is similar to but not direct reconstructions of the input data

https://www.youtube.com/watch?v=BUNl0To1IVw **(start from 17:19)**
https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

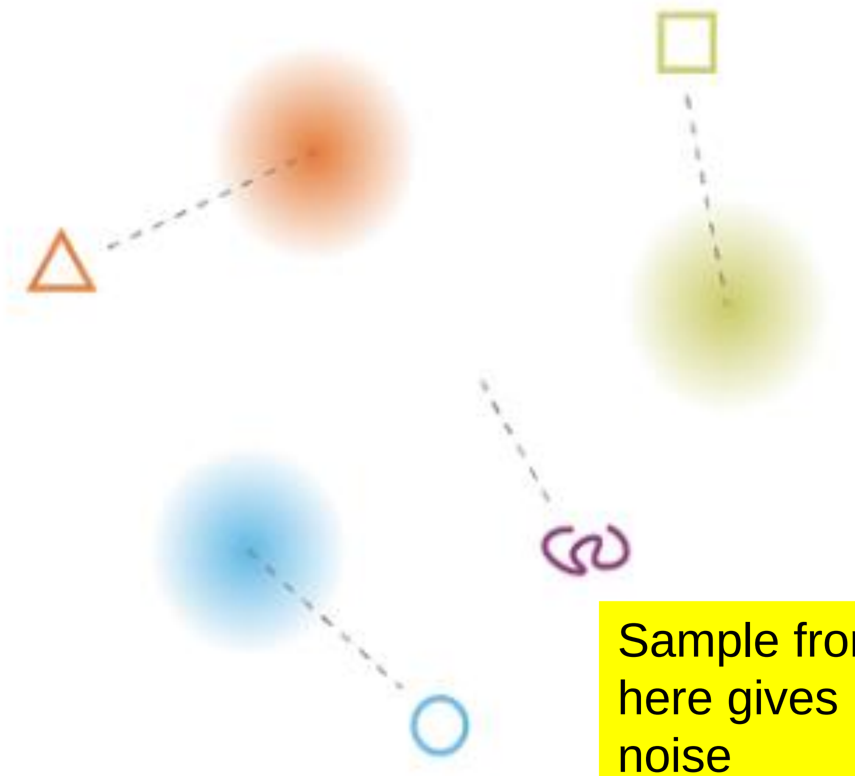
Variational Autoencoder

Latent Distribution for Autoencoder

Sampling randomly form here gives Newton



Sample from here gives **noise**



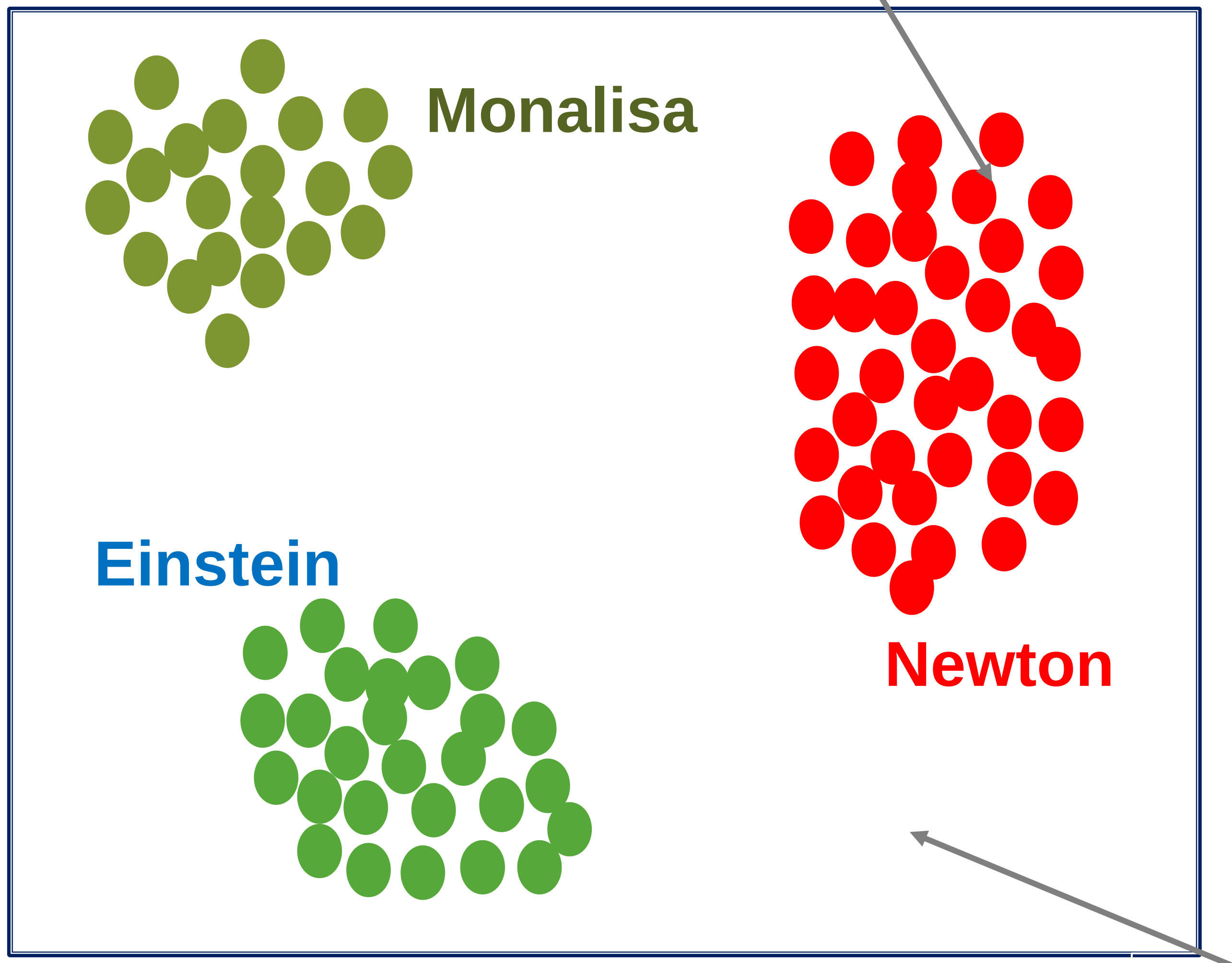
Sample from here gives noise

If we just randomly sample, then all the reconstructed value may mean nothing.

Variational Autoencoder

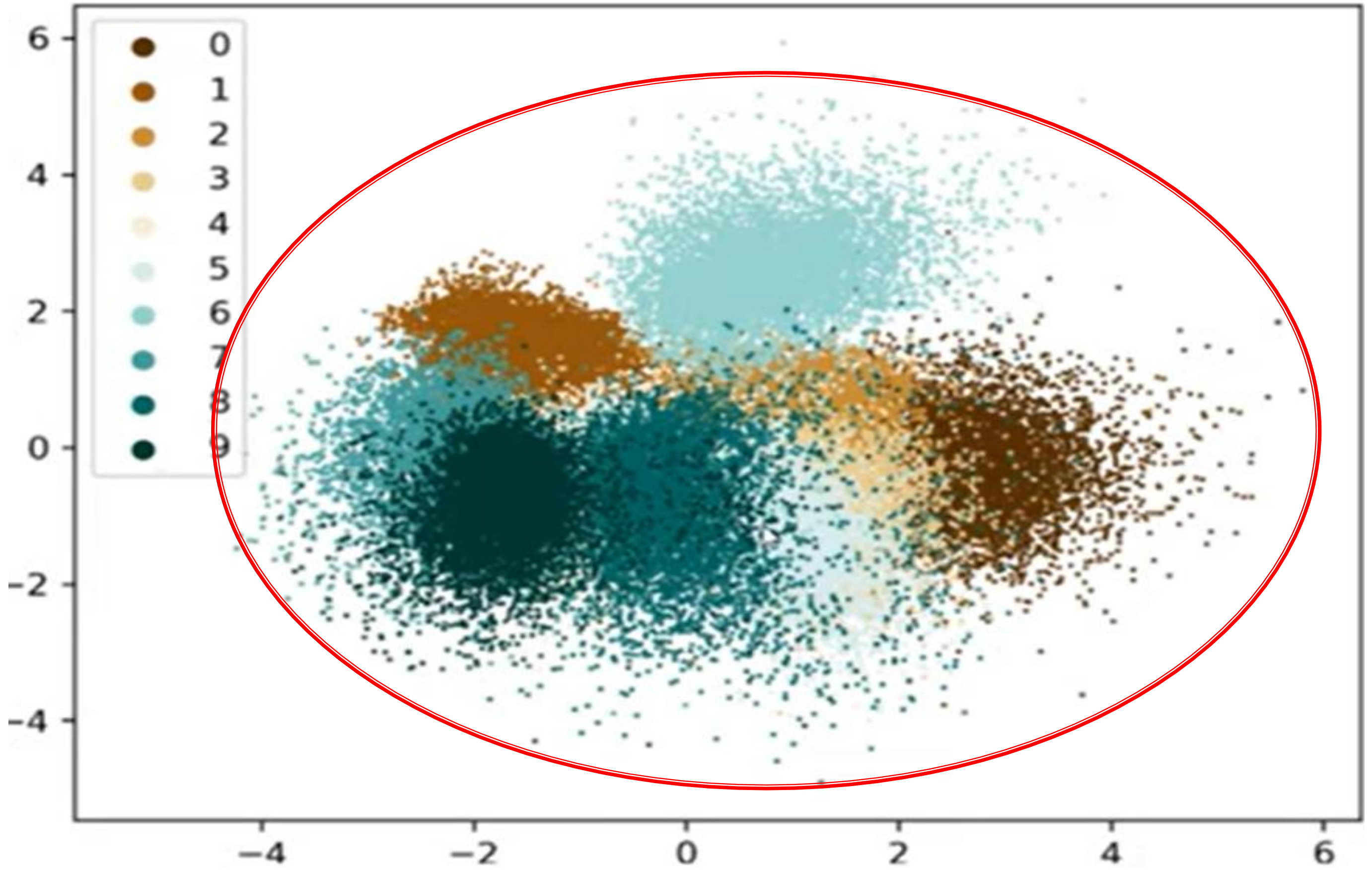
Latent Distribution for Autoencoder

Sampling randomly form here gives Newton



Sample from here gives noise

Constrain latent vector values to a specific continuous region



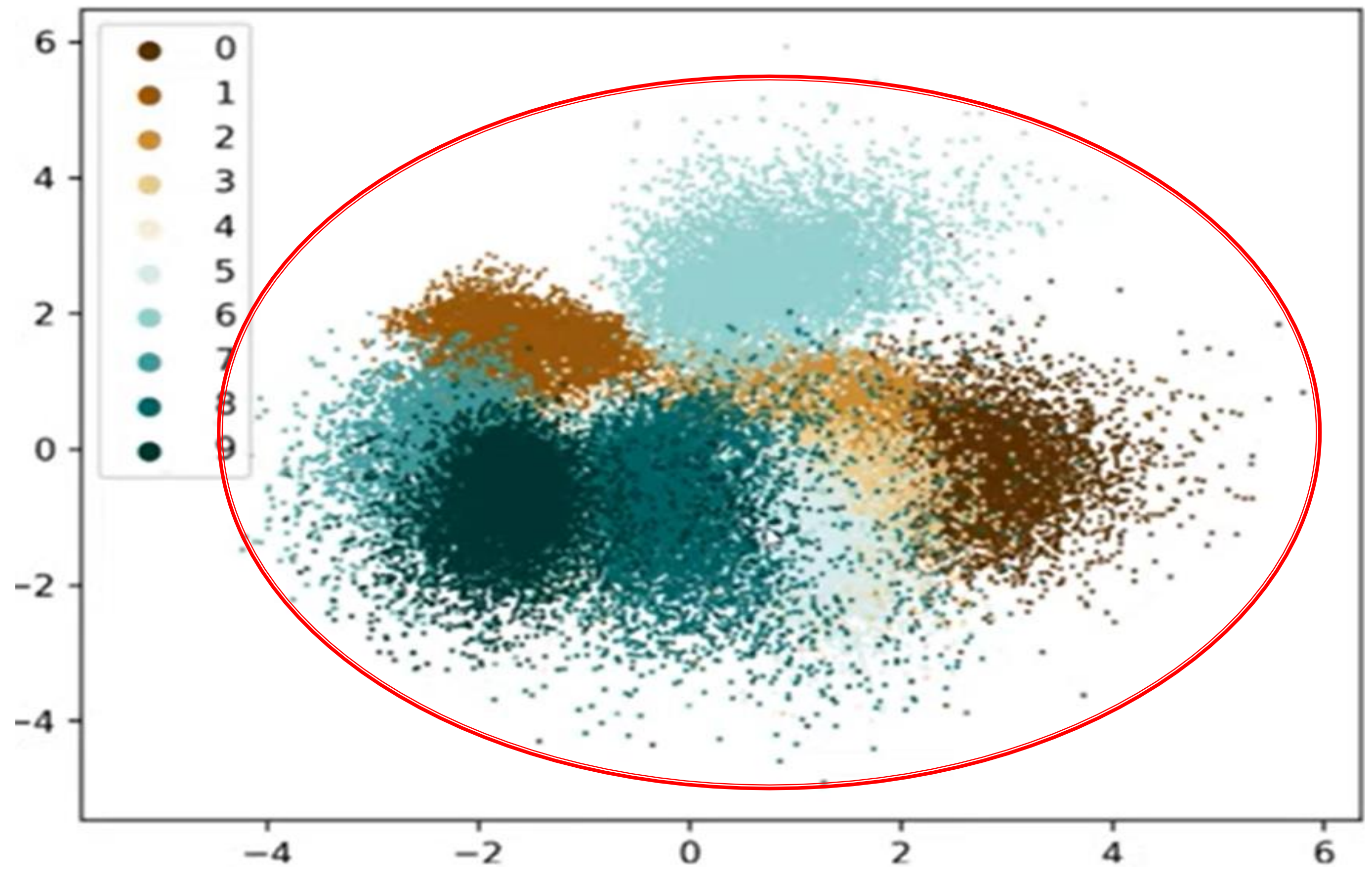
Variational Autoencoder

- Instead of mapping into a fixed latent vector, we map it to a distribution.

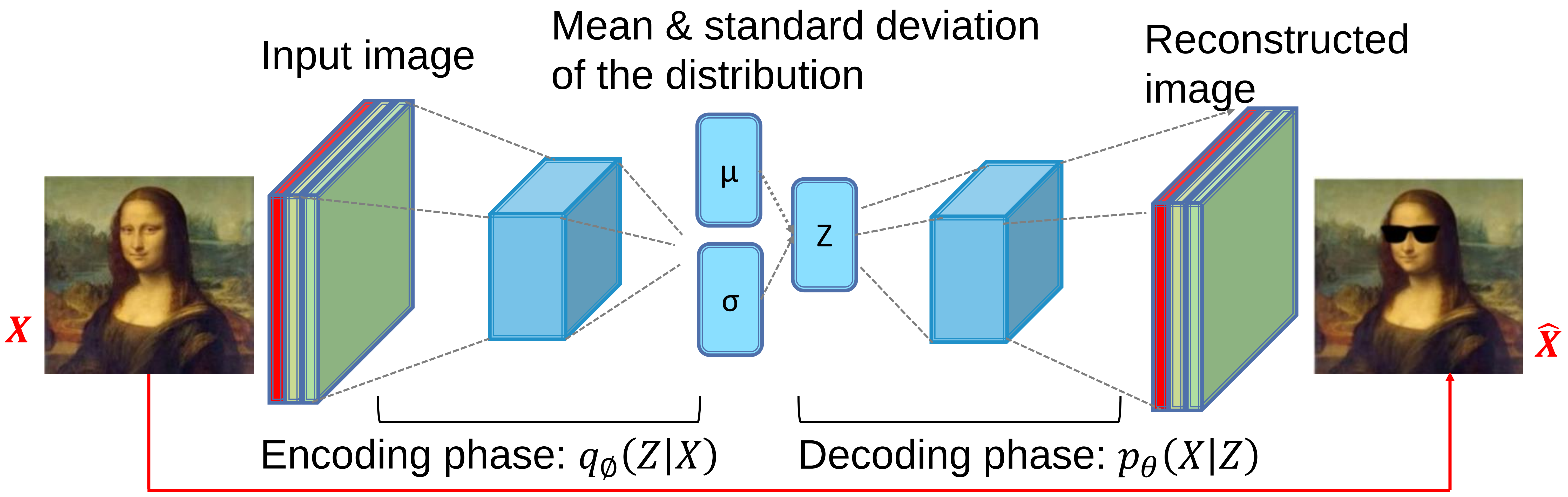
- **We constrain these latent variable to be *normally distributed***

- Instead of passing the encoder output to the decoder, use **mean** and **standard deviation** describing the distribution.

Constrain latent vector values to a specific continuous region



Reconstruction Loss



$\|X - \hat{X}\|^2$ $\quad$ $D\left(q_{\emptyset}(Z|X) \| \, p(Z)\right)$

Loss function = Reconstruction loss + KL divergence loss

Quantify the difference between the input and the reconstructed output

Quantify the distance between learned distribution and standard norm. distr. using Kullback-Leibler (KL) divergence

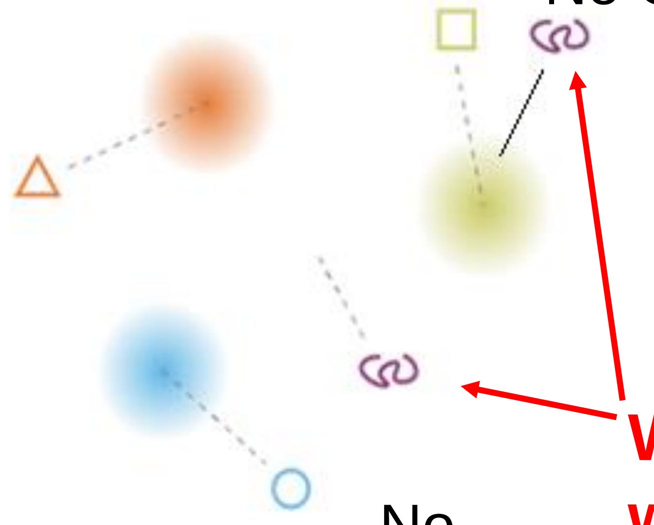
$\emptyset$: weights of Encoder

θ: weights of Decoder

Intuition on Regularization and Normal Prior

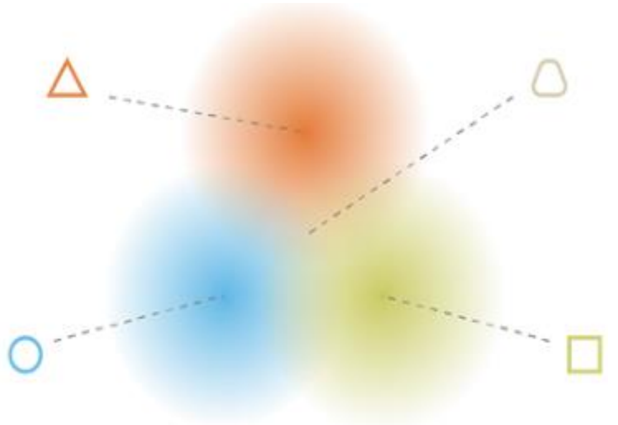
Which characteristics do we hope the regularization will produce?

- **Continuity:** Two close points in the latent space should not give two completely different contents once decoded.
- **Completeness:** For a chosen distribution, a point sampled from the latent space should give "meaningful" content once decoded



What can happen without regularization.

What we want to obtain with regularization



Centre means and regularize variance

Variational Autoencoder

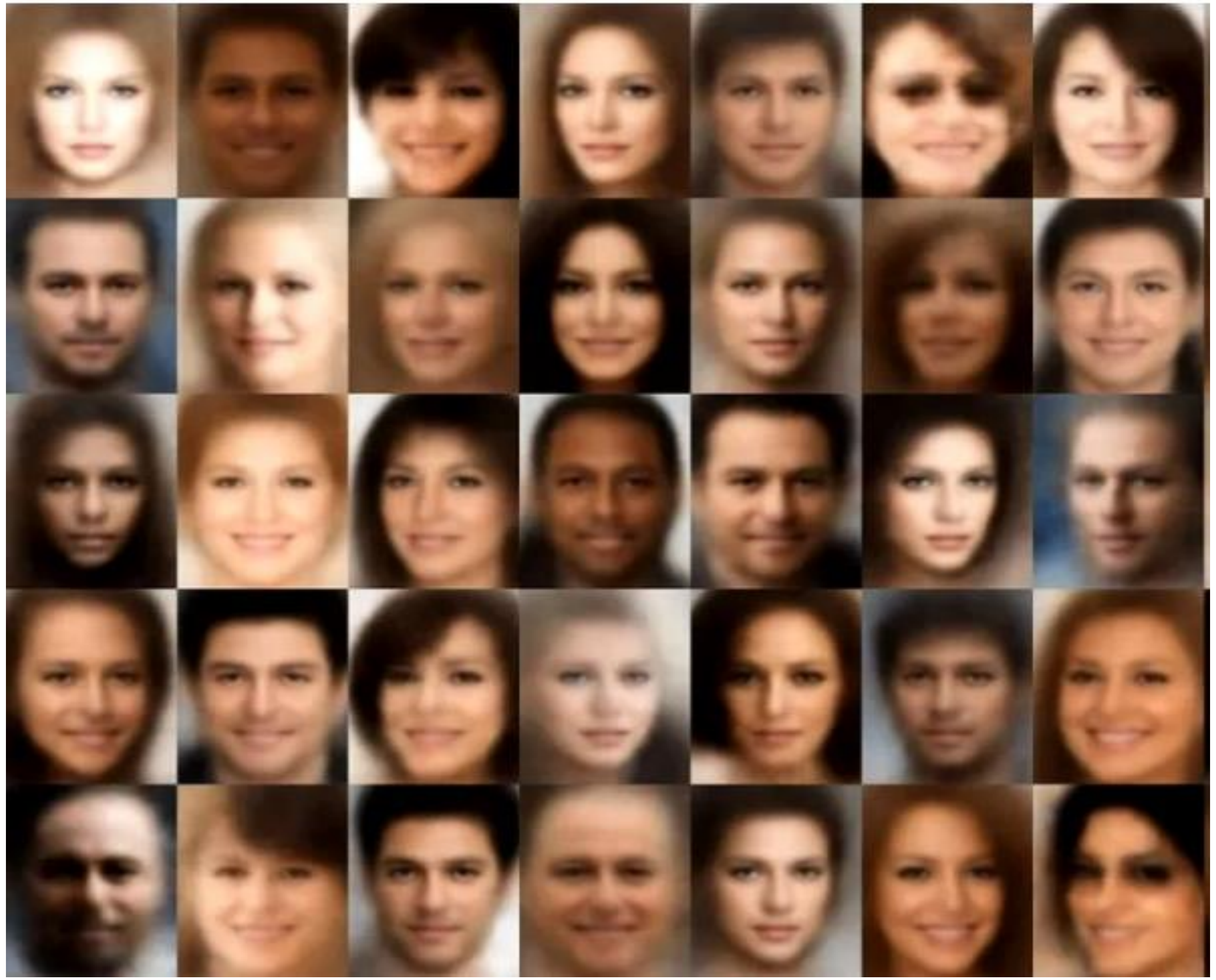
Advantages:

✔ Given an X, easy to find Z.

✔ Interpretable probability P(X).

Disadvantages:

✖ Usually outputs blurry images.



Face images generated with a Variational Autoencoder

Ref. https://github.com/wojciechmo/vae

What we learned today…

- Discriminative vs. Generative models
- Variational Autoencoders
  - Loss function = Reconstruction loss + KL divergence loss
  - Continuity and Completeness
  - Limitations of VAE

References

1. Machine Learning: A Probabilistic Perspective" by Kevin Murphy, published by MIT Press, 2012.
2. "Pattern Recognition and Machine Learning" by Christopher M. Bishop, published by Springer, 2006.
3. "Machine Learning Yearning" by Andrew Ng, published by Goodfellow Publishers, 2018.
4. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron, published by O'Reilly Media, 2019.
5. “Deep Learning”, Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press, 2016. https://www.deeplearningbook.org/

Thank You !

Any Questions