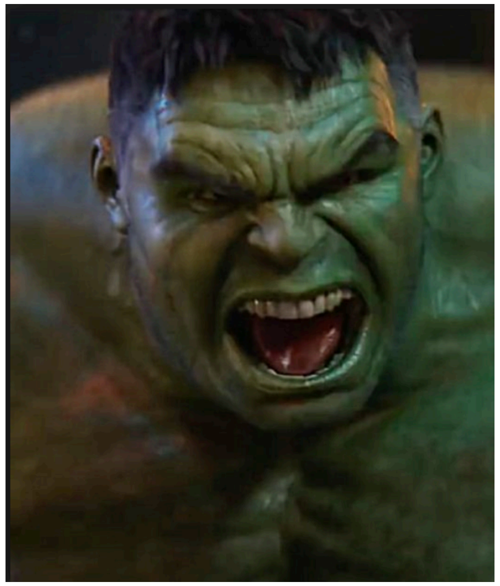
(Variational) Autoencoder

Jian Wang

04/24/2023

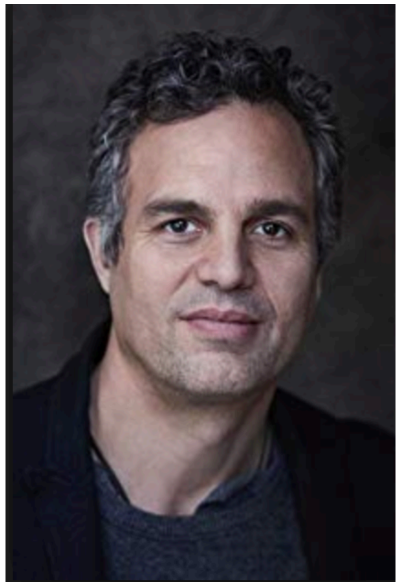
Recall: Data dimensionality reduction

High-dimensional



Complicated;
Hard to handle

Low-dimensional



Simplified clear;
Property preserved

Dimensionality reduction techs:
Autoencoder/PCA

Recall: PCA for dimensionality reduction

- One of the techniques of data dimensionality reduction;

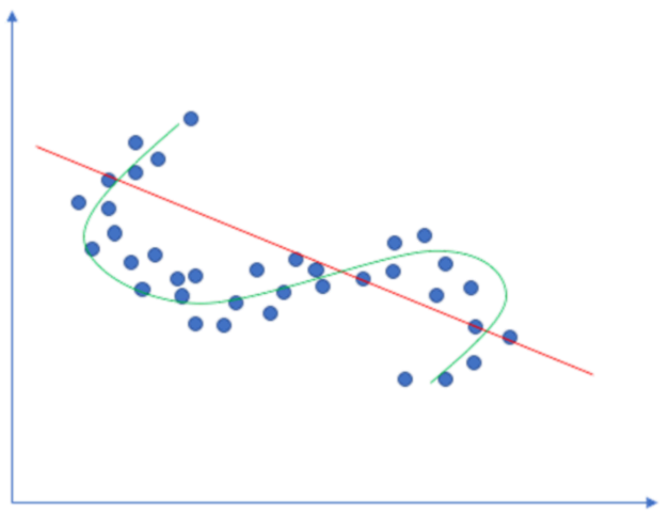
  1. Normalization;

  2. Construct covariance;

  3. Singular value decomposition (Eigenvector/ Eigenvalue);

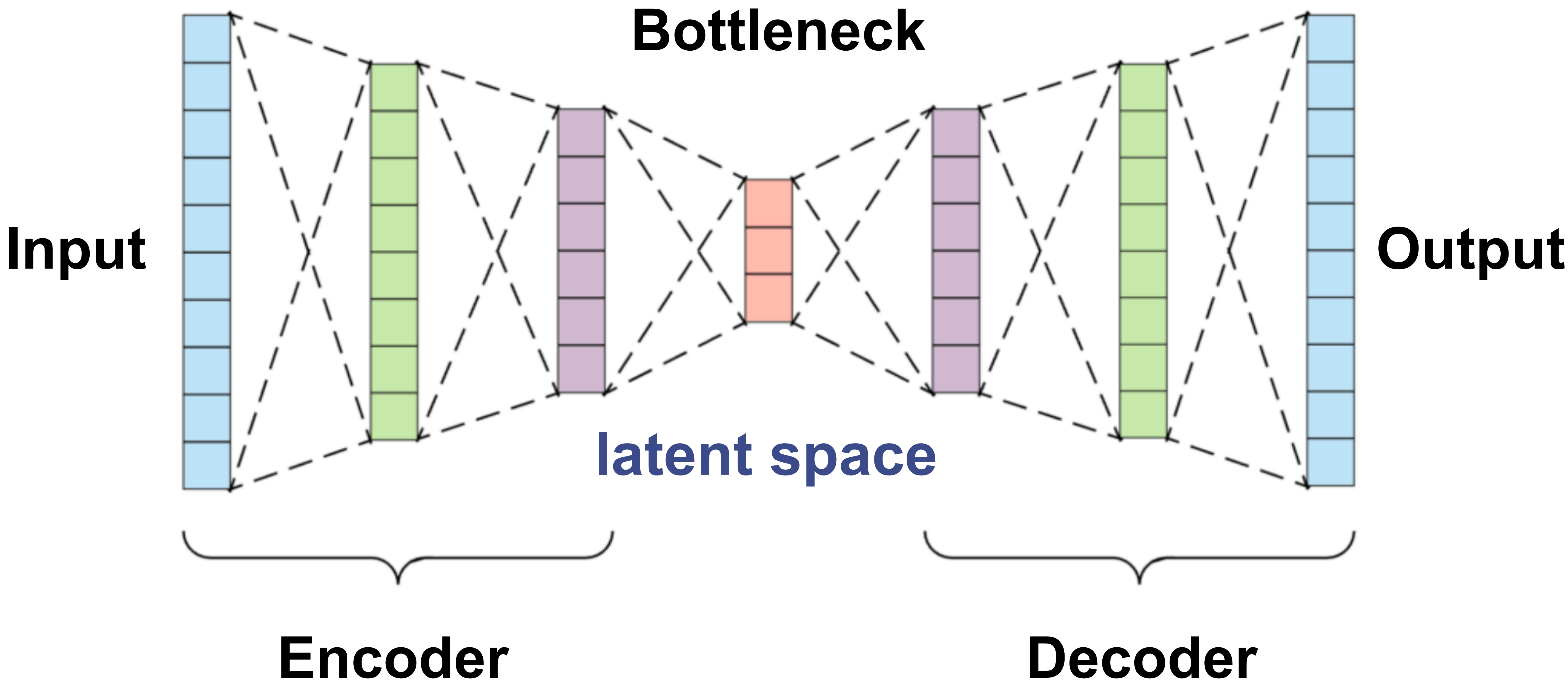
  4. Orthogonal basis selection;

  5. Data reconstruction.



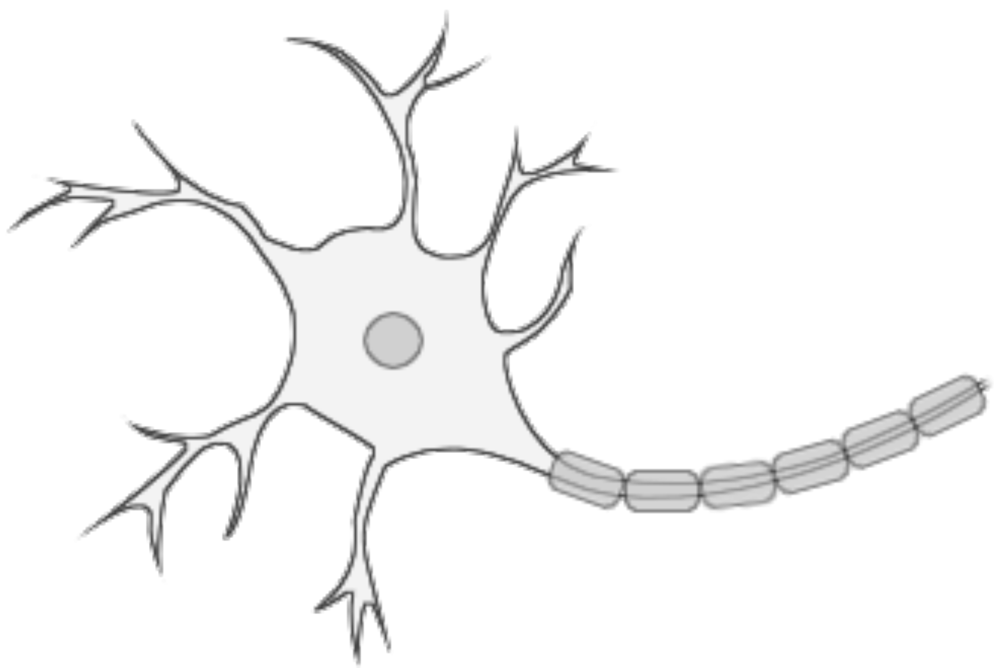
Architecture : autoencoder

- Core structure: **Encoder, Decoder** and **Bottleneck**

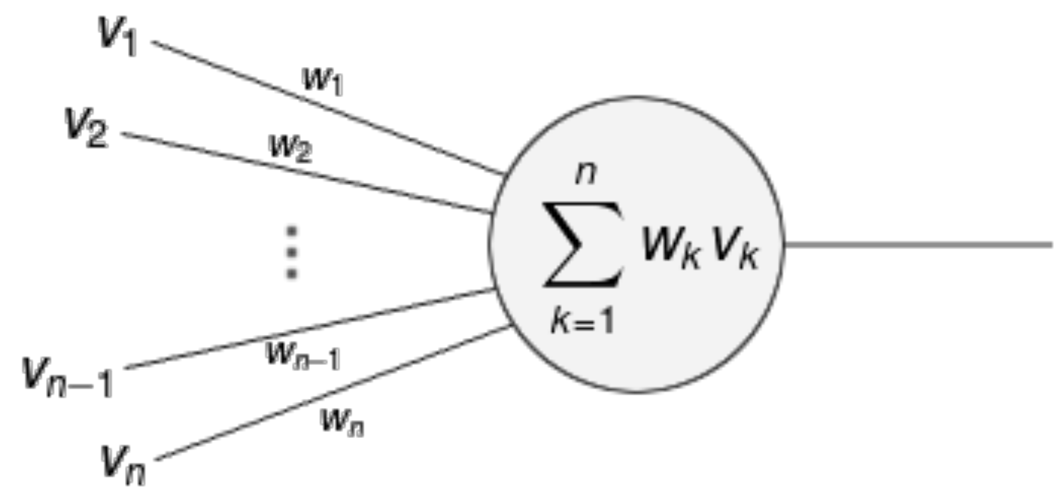


Recap: Neural Networks

- Relationship to neural networks: layers, neurons, and weights;



Brain neuron



Simulated neuron with input ***V*** and weights ***W***

- Reconstruction goal: minimize the reconstruction error.

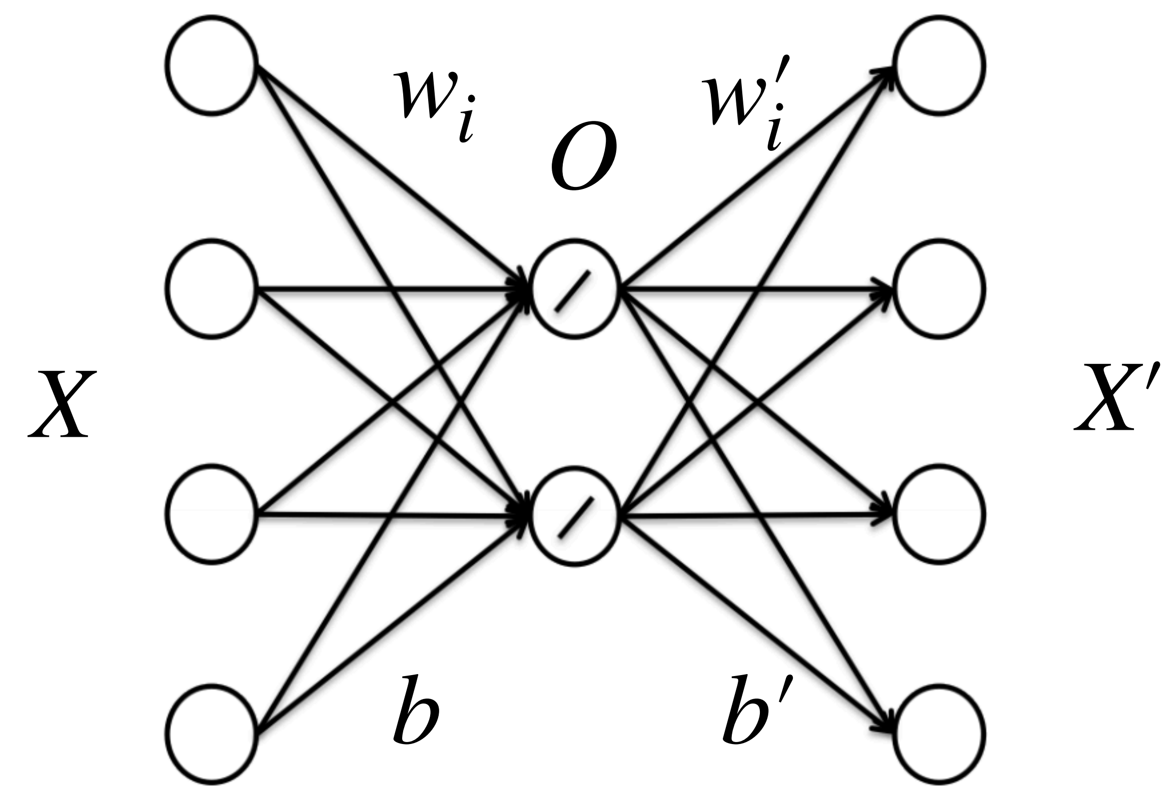
Formulation: Autoencoder

- Encoder and decoder

Original data： $[x_1, x_2, ... x_n]$ Output data： $[x_1, x_2, ... x_n]'$

Latent input/encoder output：

$$O = \sum_{i=1}^{n} w_i x + b$$



Latent output/decoder output：

$$X' = \sum_{i=1}^{n} w_i' O + b'$$

- Optimization

Loss function： $f = argmin(X - X') = ||X - (w'x + b)||^2$

Major parameters in autoencoder

- Layer size: more layers produce better accuracy?
- Size of bottleneck: number of nodes in the middle layer (smaller size results in more compression);
- Loss function: mean squared error and etc.
- Penalty term: L_1 and L_2.

Autoencoder Types

- **Vanilla autoencoder**

  The simplest form, two layers with one hidden layer;

- **Regularized autoencoder**

  Add regularization term to penalize loss function;

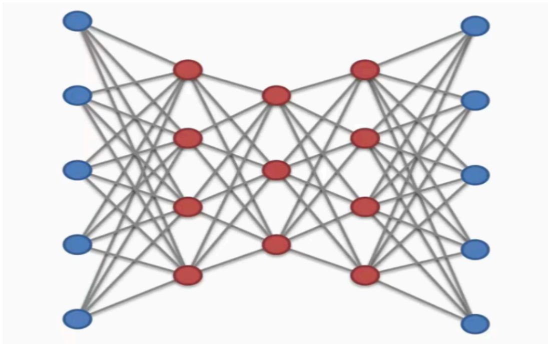
- **Multilayer autoencoder**

  Extended form based on vanilla, multiple hidden layers;

- **Convolutional autoencoder**

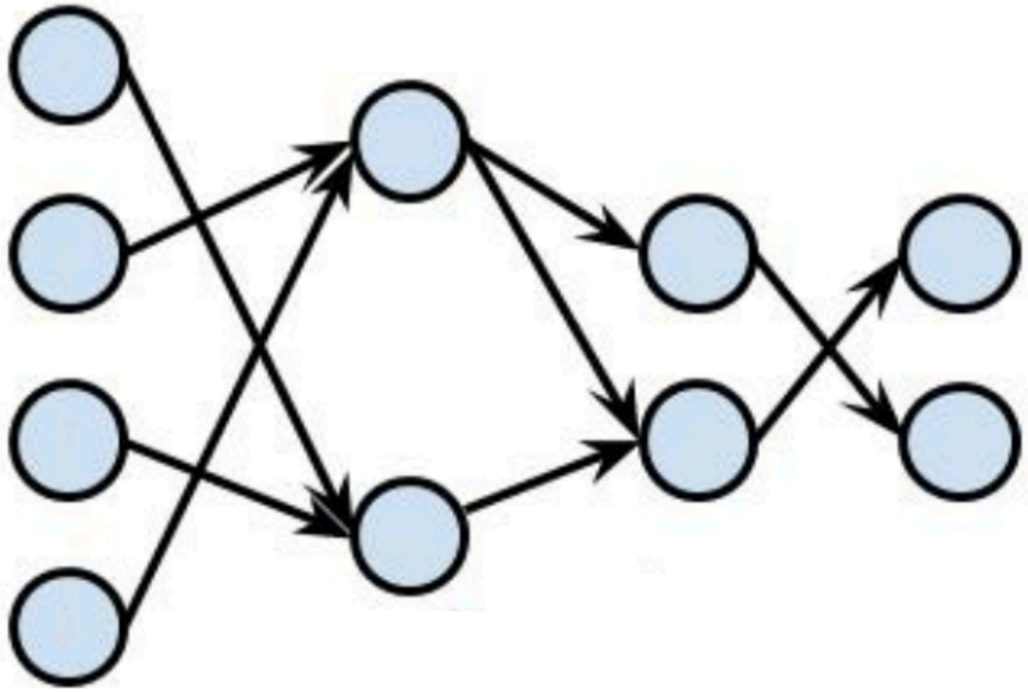
  Using pooling layers and convolution instead of fully connected layers;

Deep autoencoder →

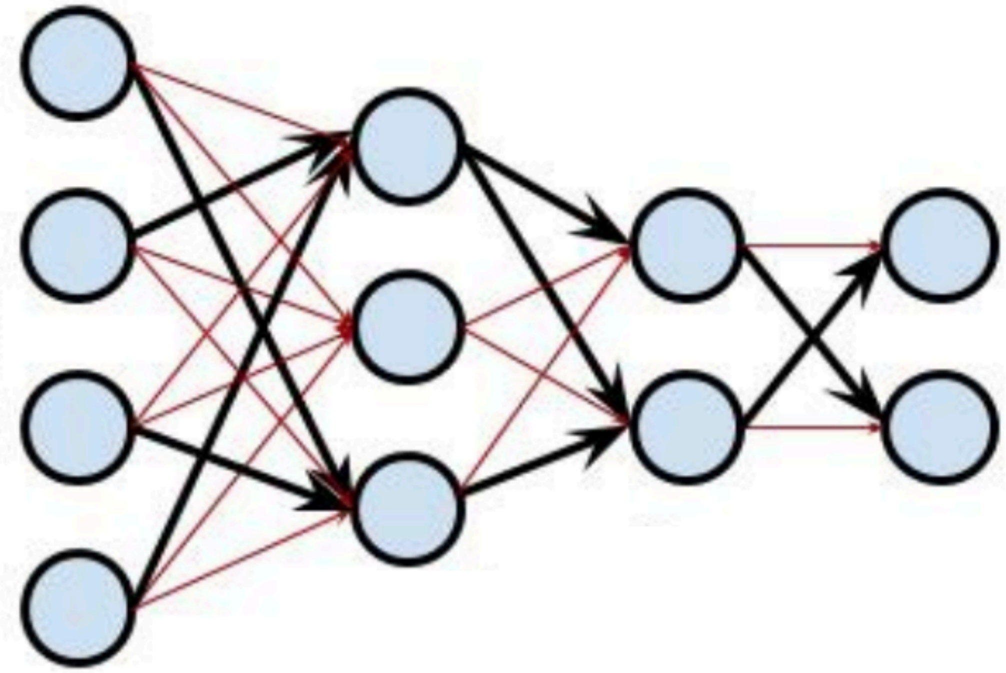


Types: Regularized (Sparse) Autoencoder

- Sparse representation:



Sparse autoencoder



Fully-connected autoencoder

- Optimization and loss function (Penalty term):

$$Loss = \mathrm{Dist}[\mathrm{X}, \mathrm{X}'] + \lambda \mathrm{Reg}(\mathrm{w})$$

Advantages: Autoencoder

- Good characterization: captures non-linear features/patterns;
- Low cost: doesn’t have to use dense layers;
- Fast convergence (pre-trained): weights initializations for other models;

Architecture : autoencoder

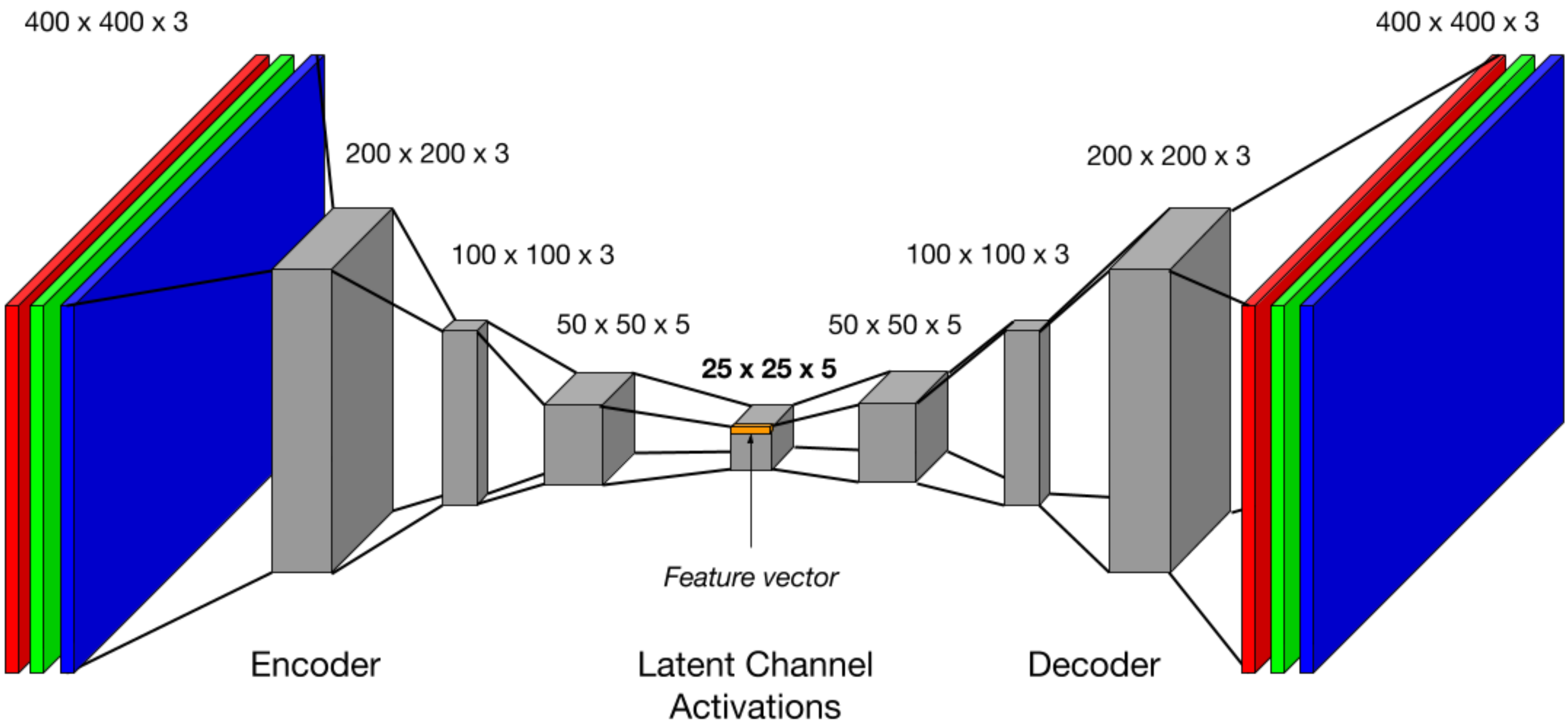
- **Encoder:** Compresses the input into a **latent space representation**.

- **Bottleneck:** This part of the network represents the **compressed** input which is fed to the decoder.

- **Decoder:** Decodes the encoded data back to the original dimension. The decoded image is a **lossy reconstruction** of the original image.

Types: Convolutional Autoencoder

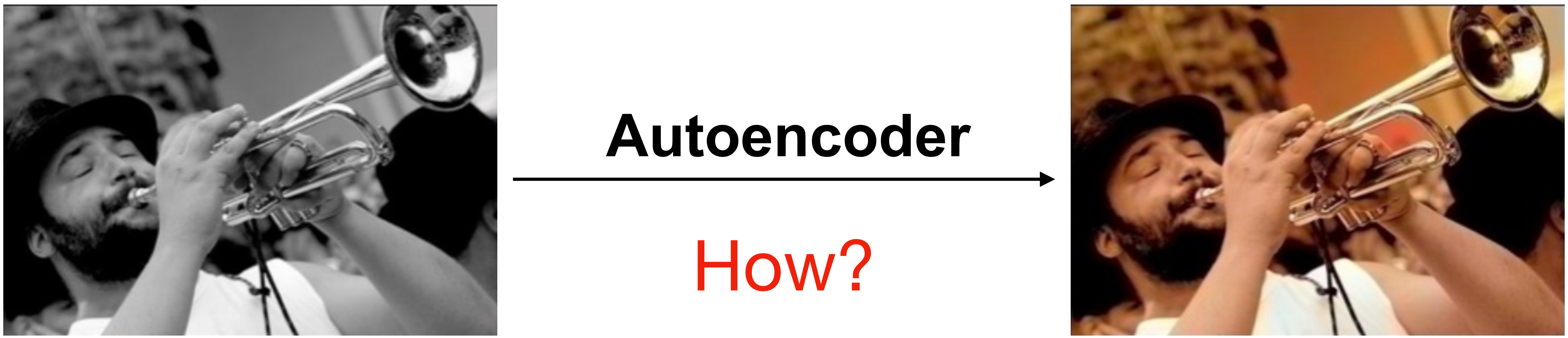
- Structure



- Same operations with convolutional neural network (CNN)

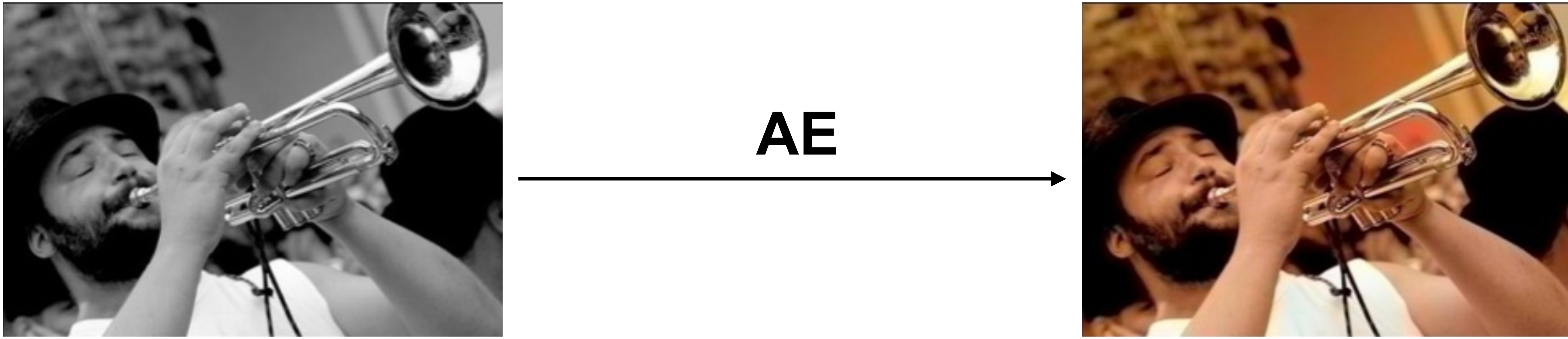
Applications of autoencoder: colorization

- Image coloring:

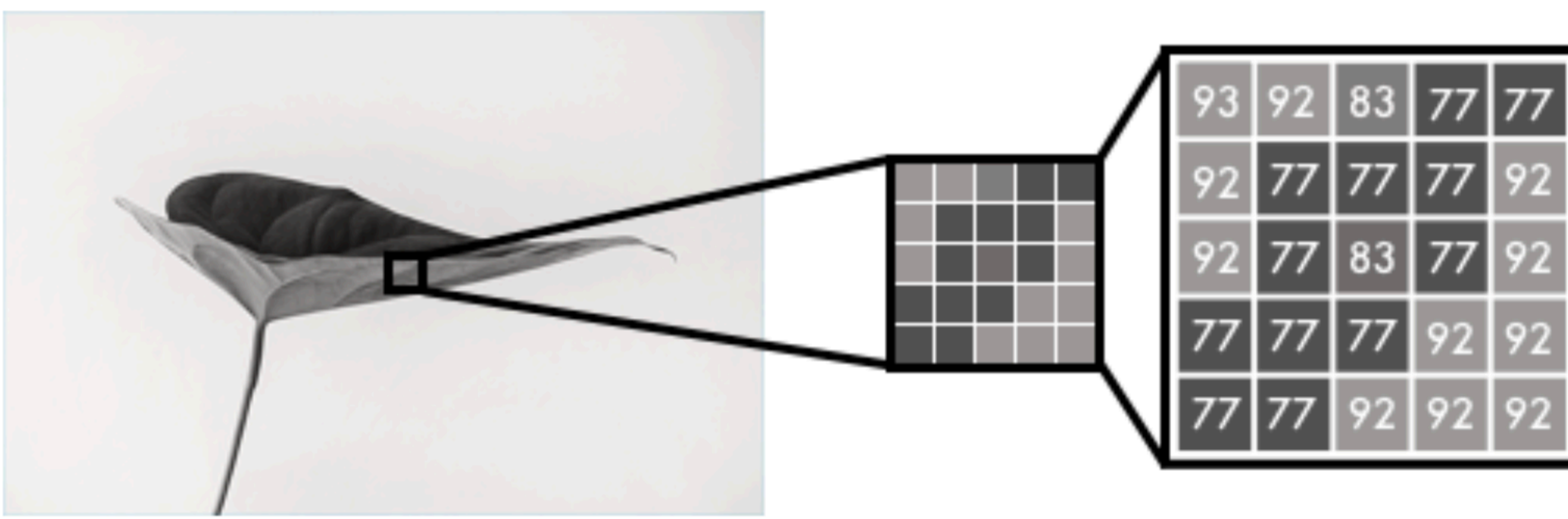


- Reconstruct an image from grey color space to RGB/HSV.

Example: Image coloring



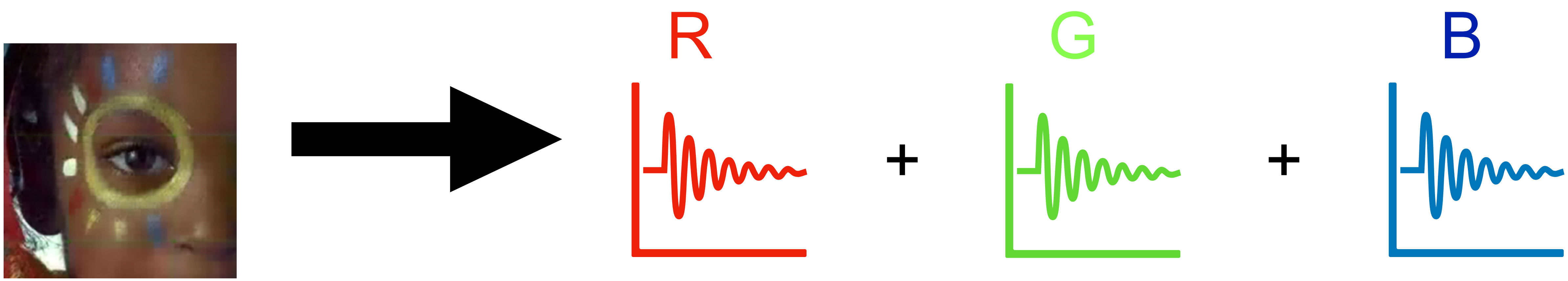
- Grey image color scale:



- Learn the intensity pattern!

Example: Image coloring

- Training:



Train autoencoders for three channels.

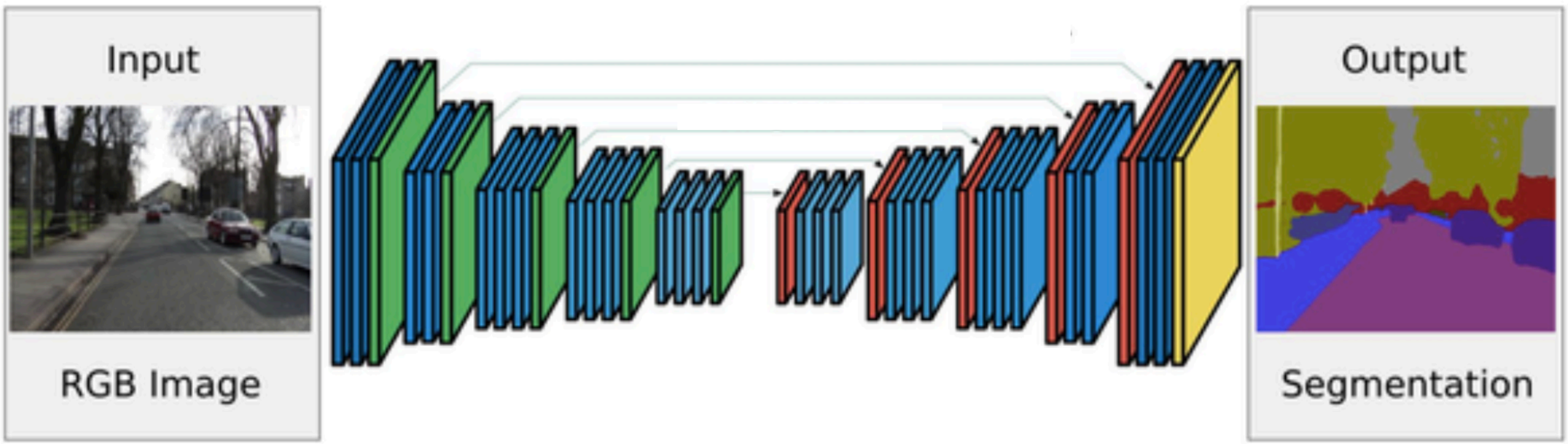
- Testing:

Well-trained autoencoder function

Predictions for each channel

Applications: Autoencoder

- Image segmentation:



- What do we need ?

    Ground-truth (segmentation labels)

Applications: Autoencoder

- Data reconstruction:

Training data

Fully-connected 16

Applications: Autoencoder

- Data reconstruction:

Training data

Fully-connected 32

Applications: Autoencoder

- Data reconstruction:

Training data

Fully-connected 128

Applications: Autoencoder

- Data reconstruction:

Conv16

Conv32

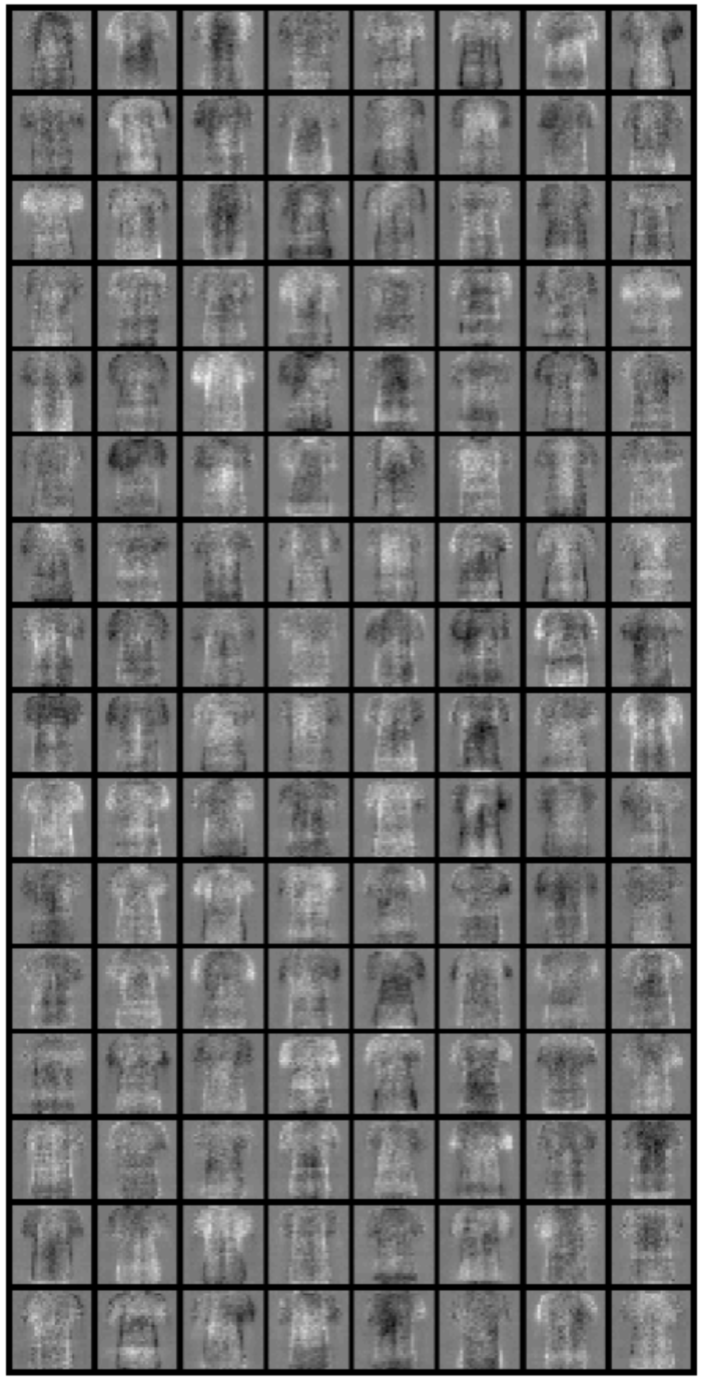
Conv128

Applications: Autoencoder

- Data feature extraction:

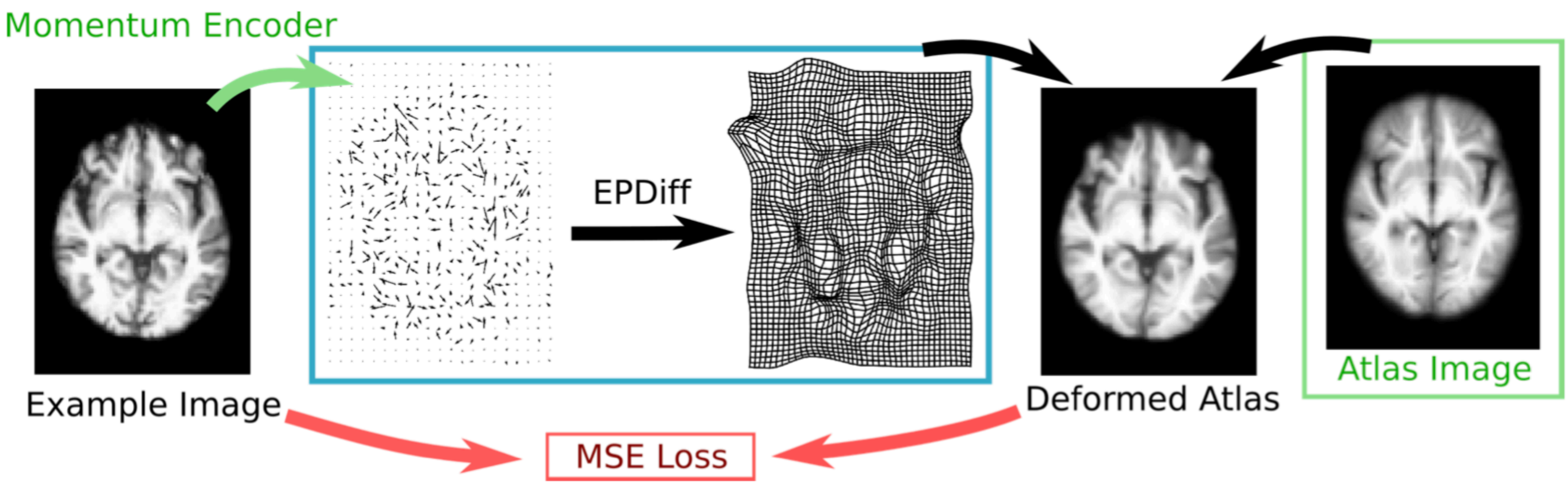
New image

Feature space



Applications: Autoencoder

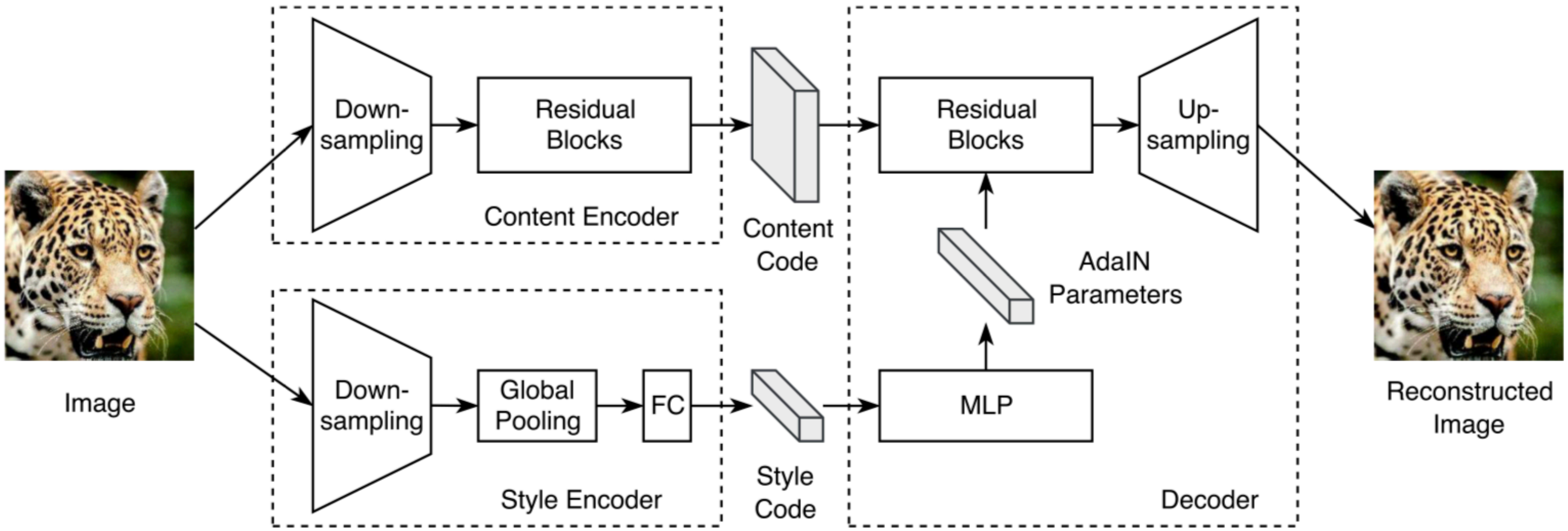
- Diffeomorphic autoencoder [1]:



[1]. Jacob Hinkle, 2018, Diffeomorphic Autoencoders

Applications: Autoencoder

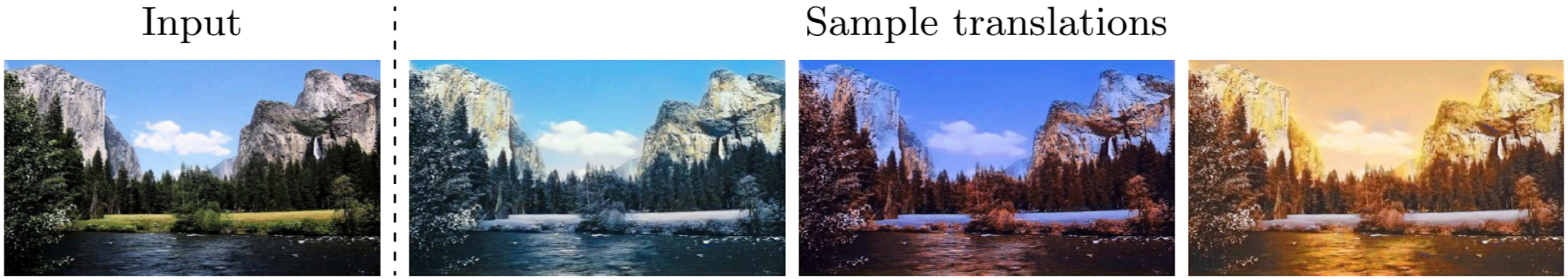
- Image translation[1]:



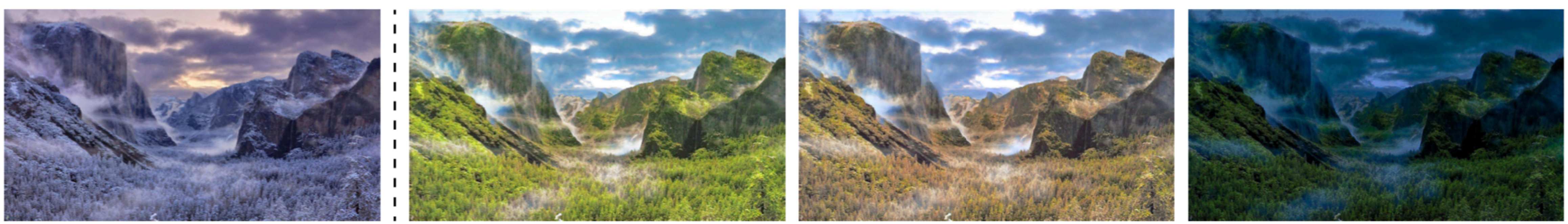
[1]. Huang, 2018, Image-to-Image Translation

Applications: Autoencoder

- Image translation[1]:



(a) Yosemite summer → winter



(b) Yosemite winter → summer

[1]. Huang, 2018, Image-to-Image Translation

Applications: Autoencoder

- Autoencoder-based joint learning framework: Geo-SIC

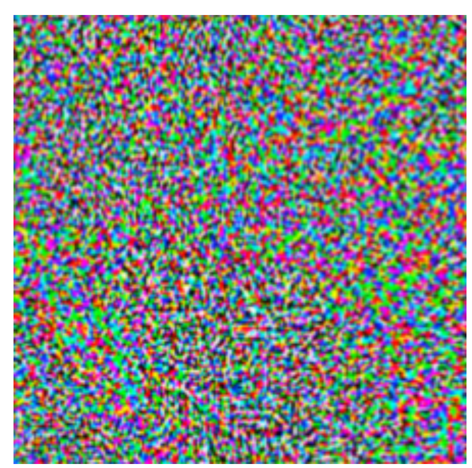
Indian elephant

Tabby cat

Texture-based shape[1]
(mis-classified as Indian elephant)

Fish

\+ 0.005 X



Universal adversarial noise

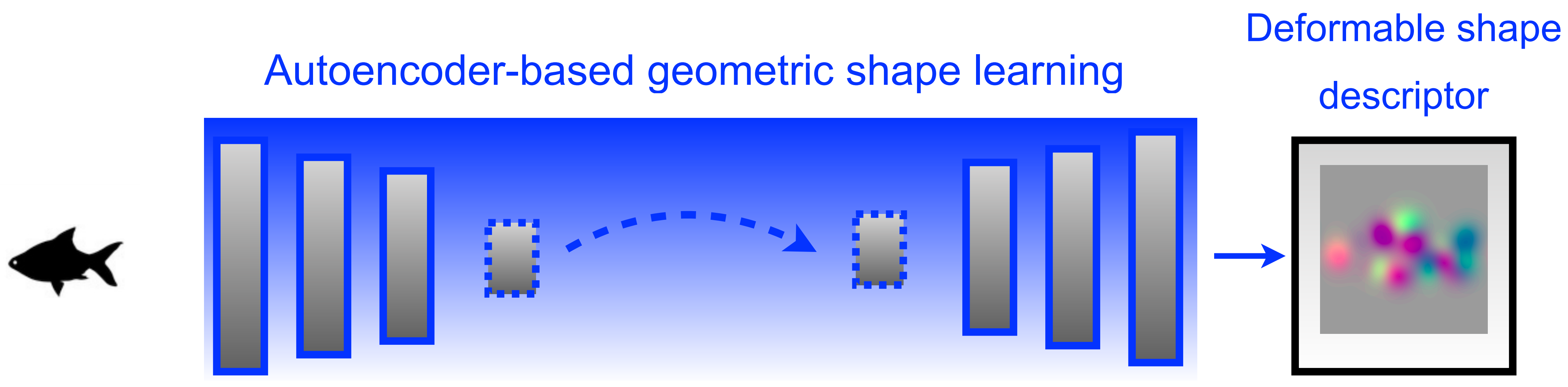
=

Predicted: Airplane

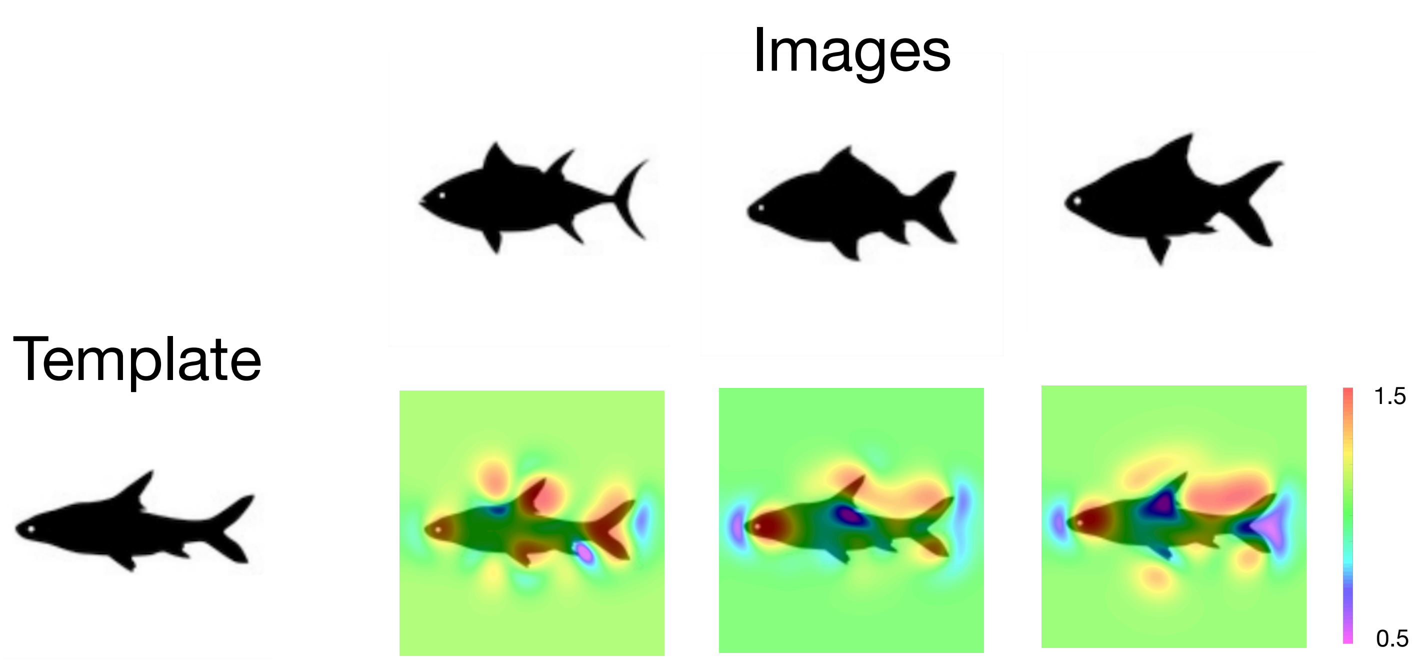
[1]. Geirhos et. al, 2019, ICLR

Applications: Autoencoder

- Autoencoder-based multi-task learning

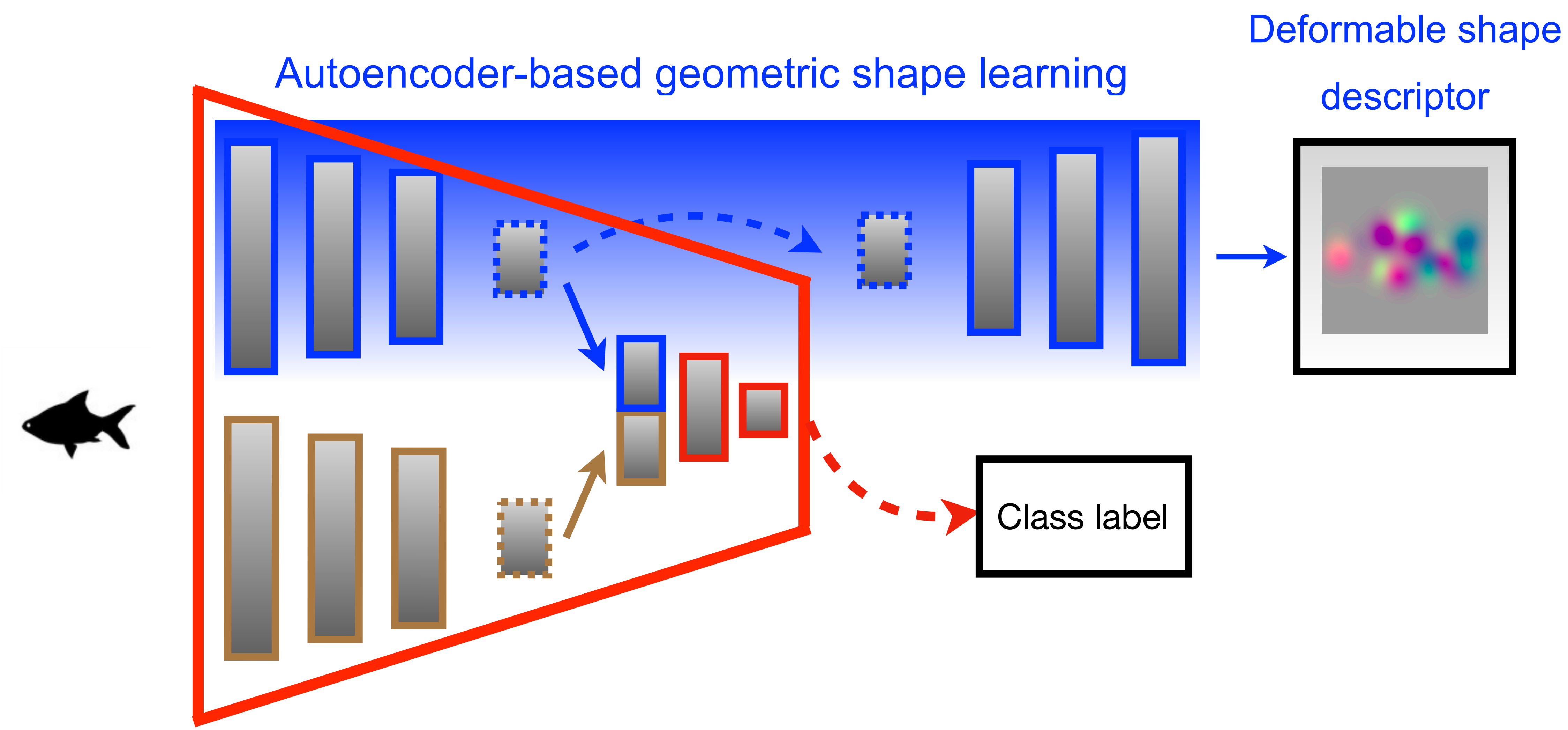


- Deformable geometric shape representations



Applications: Autoencoder

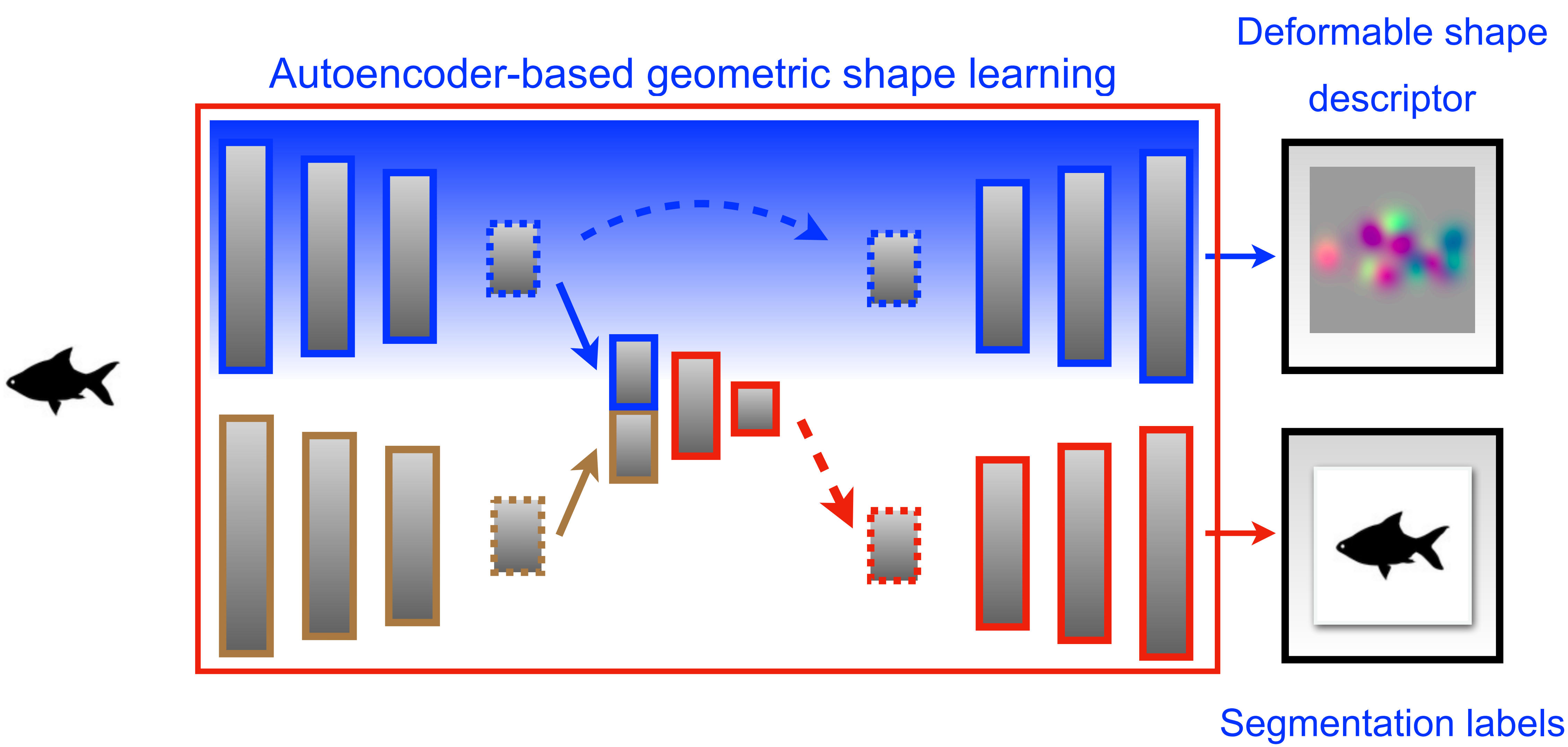
- Autoencoder-based joint learning framework



- Atlas building + image classification

Applications: Autoencoder

- Autoencoder-based joint learning framework



- Atlas building + image segmentation

Applications: Autoencoder

- Feature extraction
- Image denoising
- Image colorization
- Image segmentation
- Image registration
- Data Compression

Applications: Autoencoder

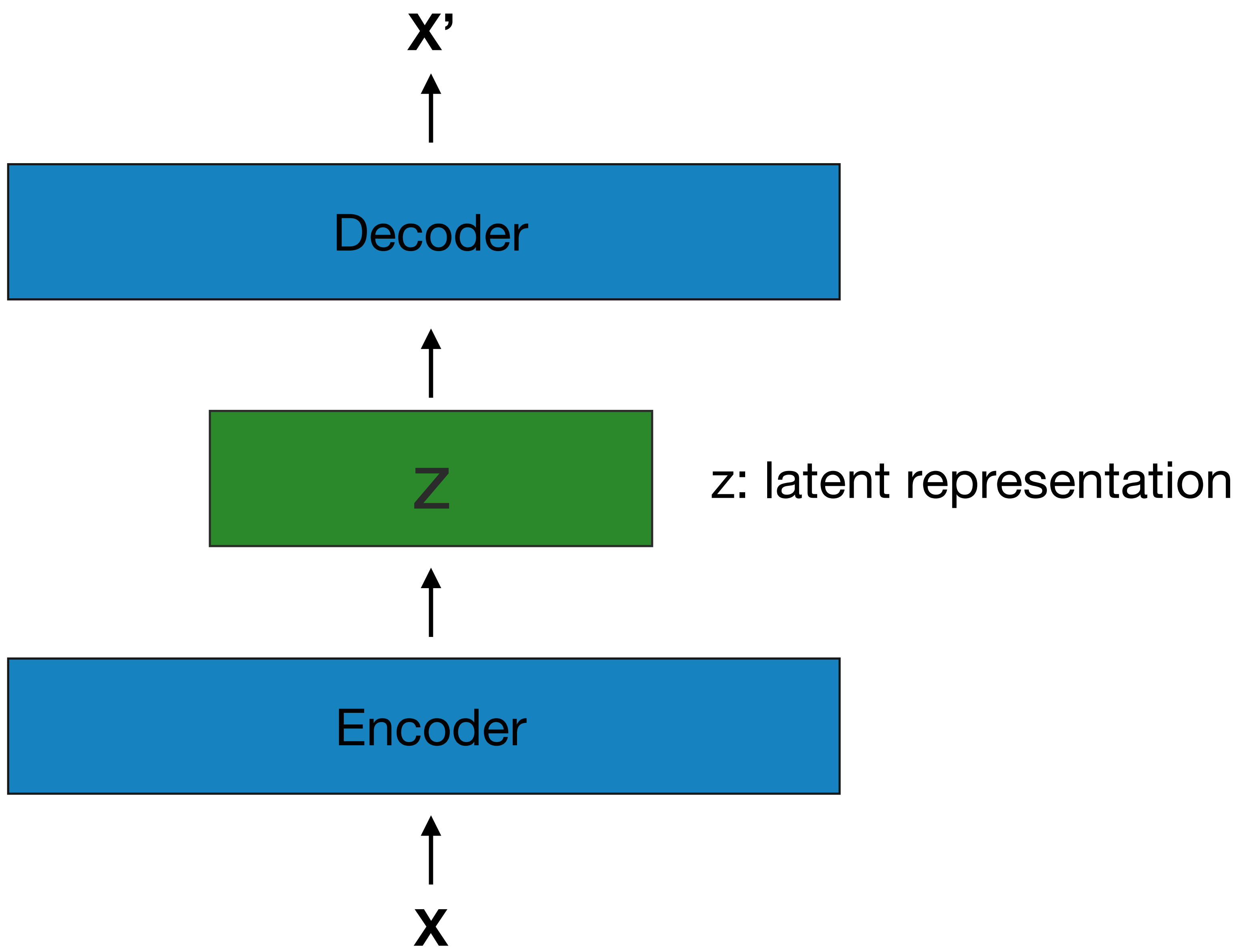
- Feature extraction

  …

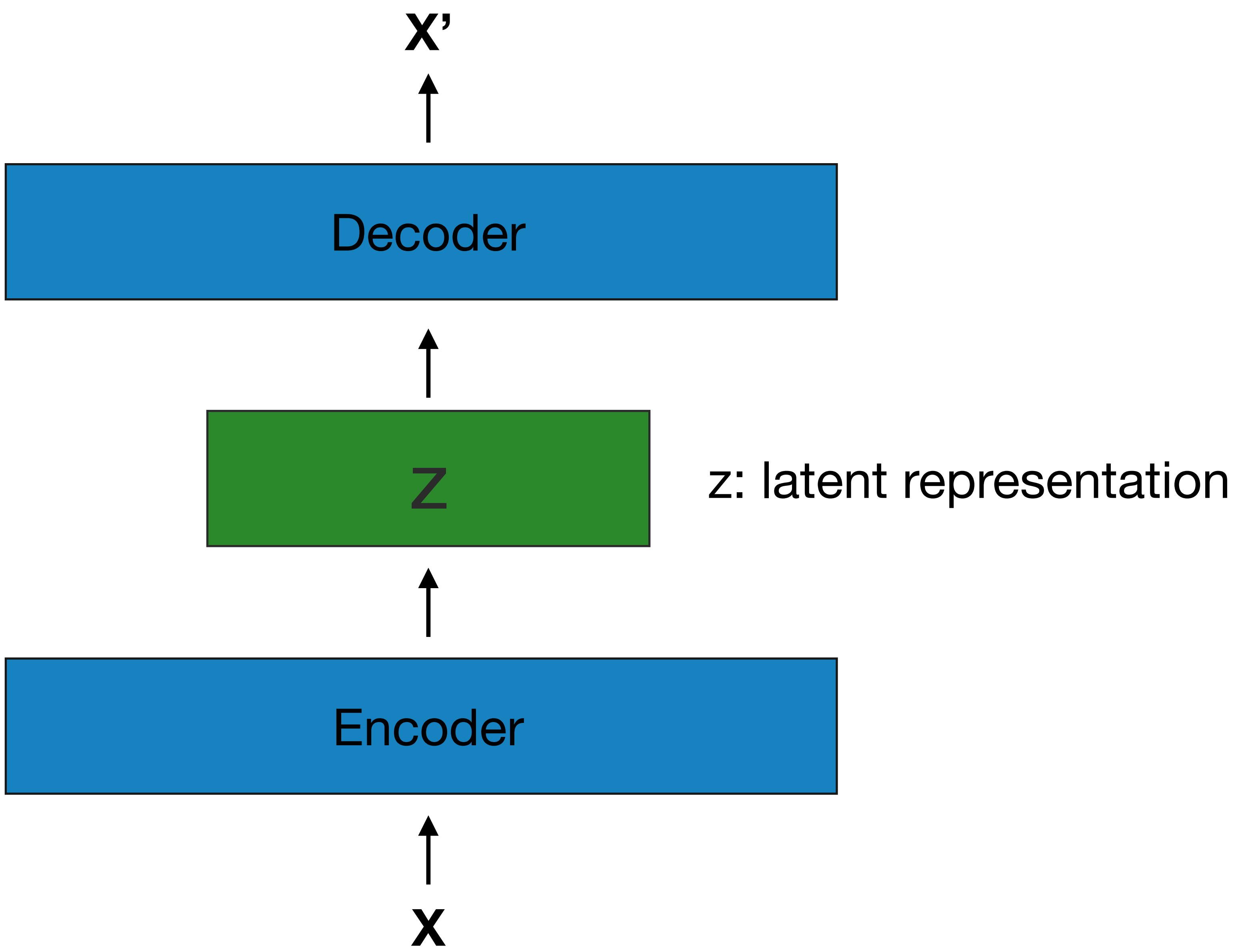
- Data Compression

Generating new data?

Autoencoder (AE)



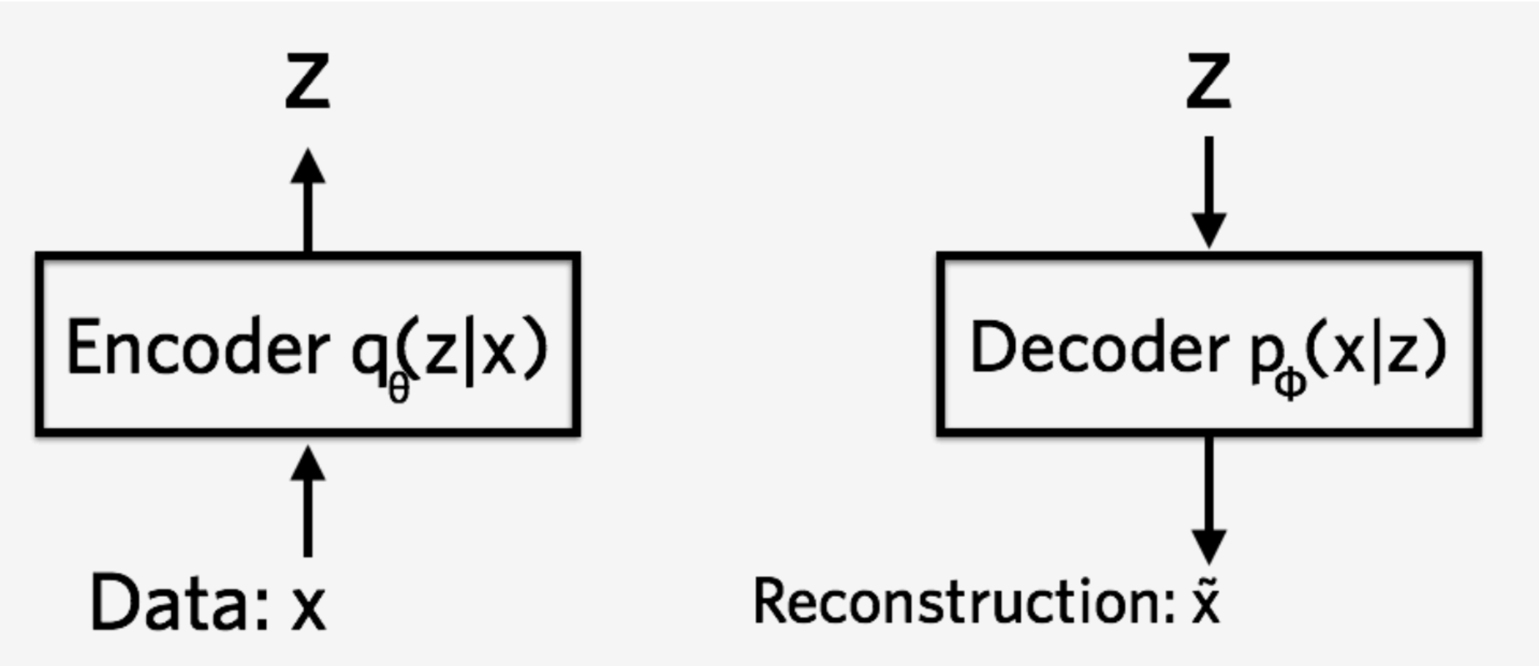
Autoencoder (AE)



- Advanced variants of autoencoders: Variational Autoencoder (VAE).

Variational Autoencoder (VAE) [1]

- Key idea: make both the encoder and the decoder probabilistic;
- I.e., the latent variables, z, are drawn from a probability distribution depending on the input, X, and the reconstruction is chosen probabilistically from z.



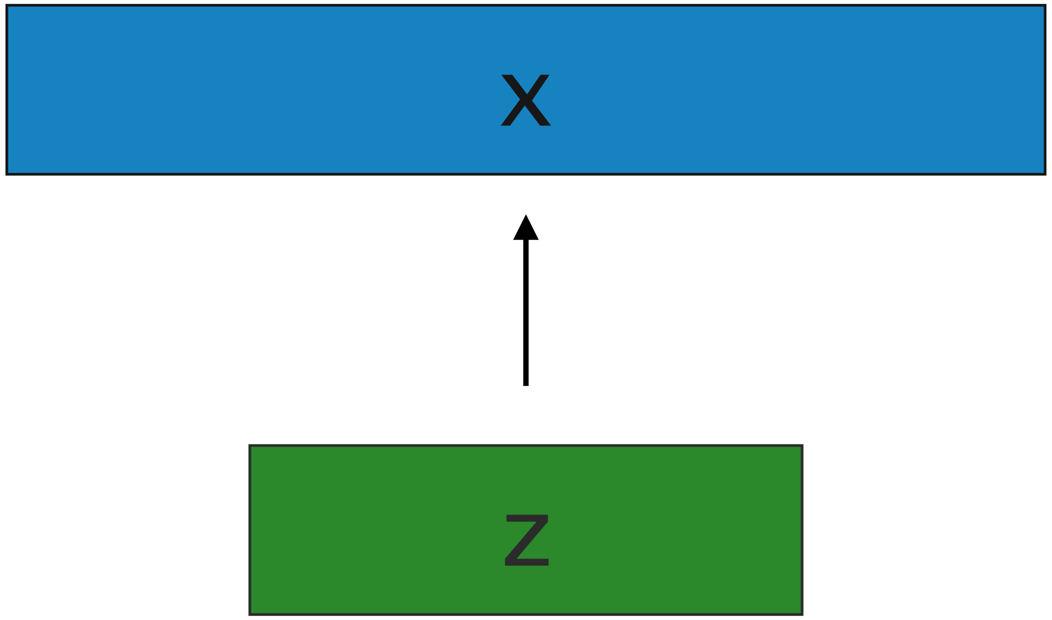
[1]. Kingma and Weilling: Auto-Encoding Variational Bayes.

Variational Autoencoder (VAE)

- Probabilistic spin on autoencoders: sample from the generative model then to generate data!

Sample from true conditional

$p_\theta(x \mid z)$



Sample from true prior

$p_\theta(z)$

Variational Autoencoder (VAE)

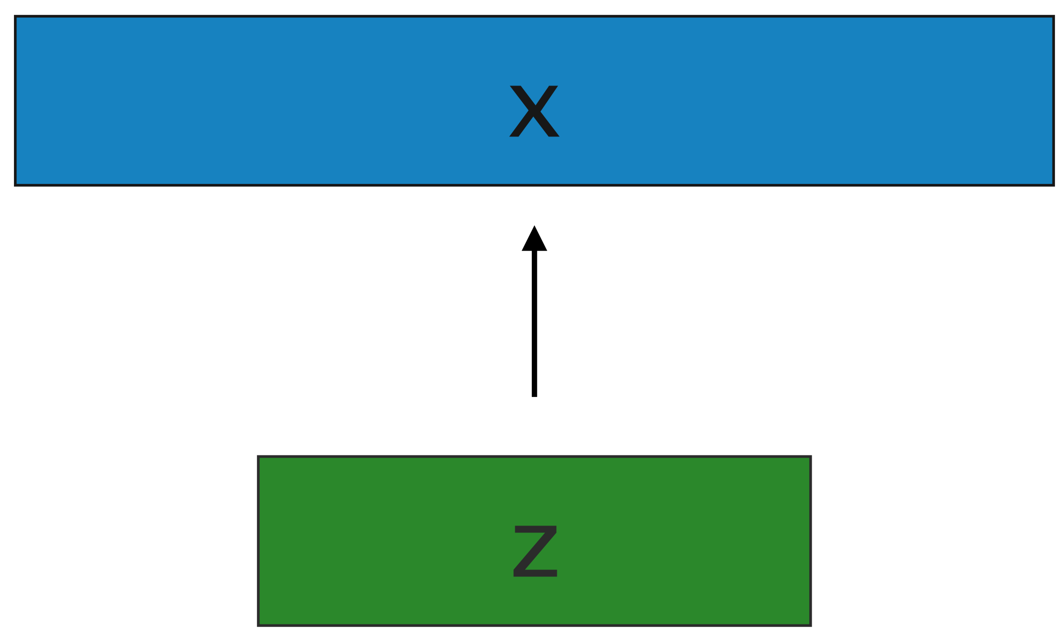
- Probabilistic spin on autoencoders: sample from the generative model then to generate data!

Sample from true conditional

$p_\theta(x \mid z)$

Sample from true prior

$p_\theta(z)$



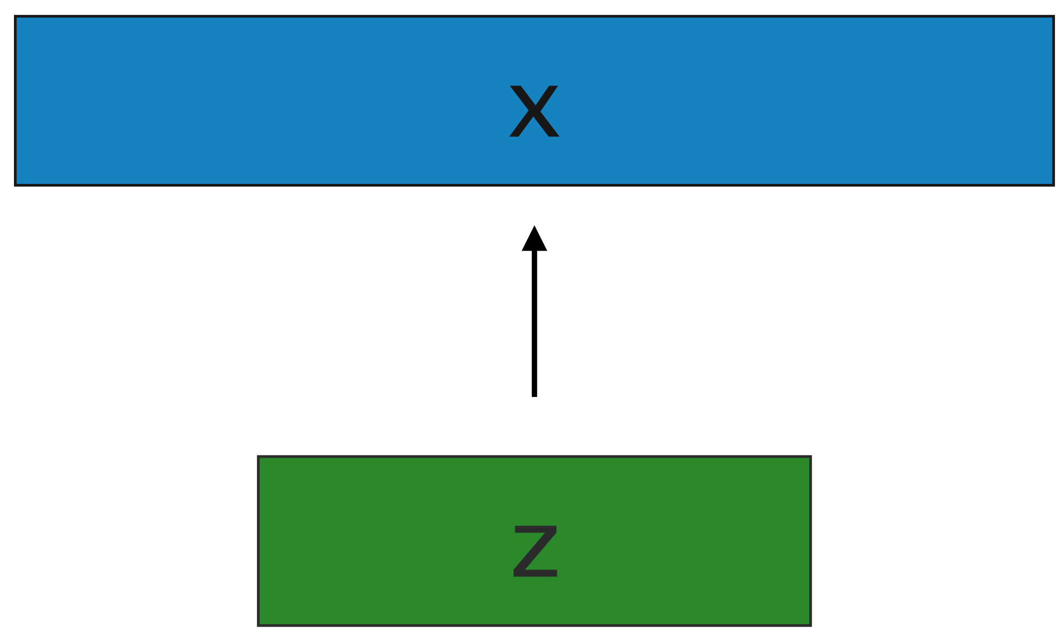
How should we represent this model?

Variational Autoencoder (VAE)

- Probabilistic spin on autoencoders: sample from the generative model then to generate data!

Sample from true conditional

$p_\theta(x \mid z)$



Sample from true prior

$p_\theta(z)$

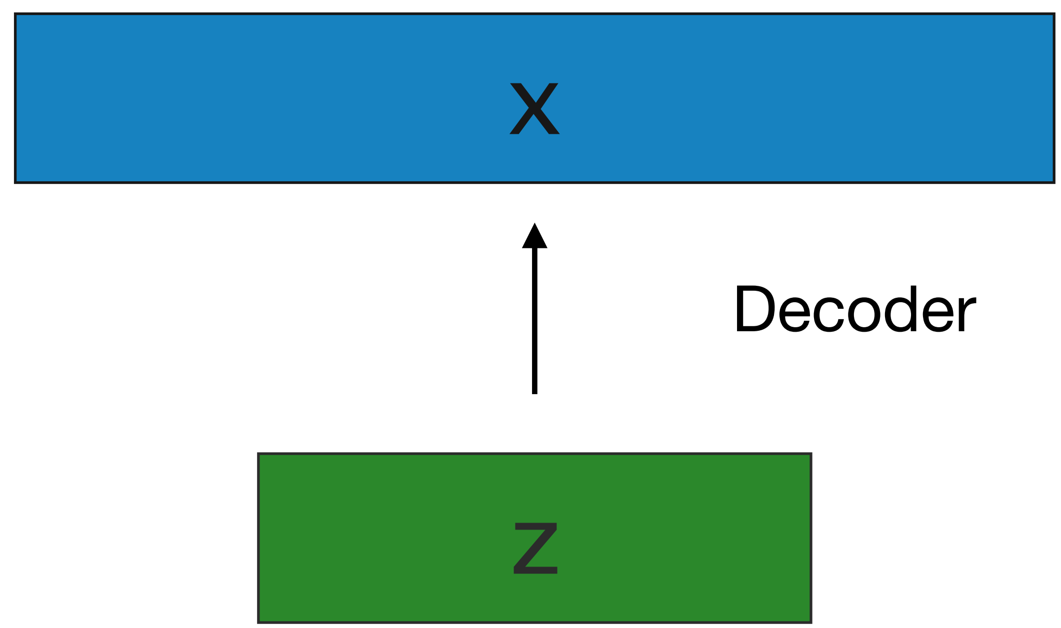
Prior p(z) to be simple, e.g. Gaussian.

Variational Autoencoder (VAE)

- Probabilistic spin on autoencoders: sample from the generative model then to generate data!

Sample from true conditional

$p_\theta(x \mid z)$



Sample from true prior

$p_\theta(z)$

θ: parameters of decoder.

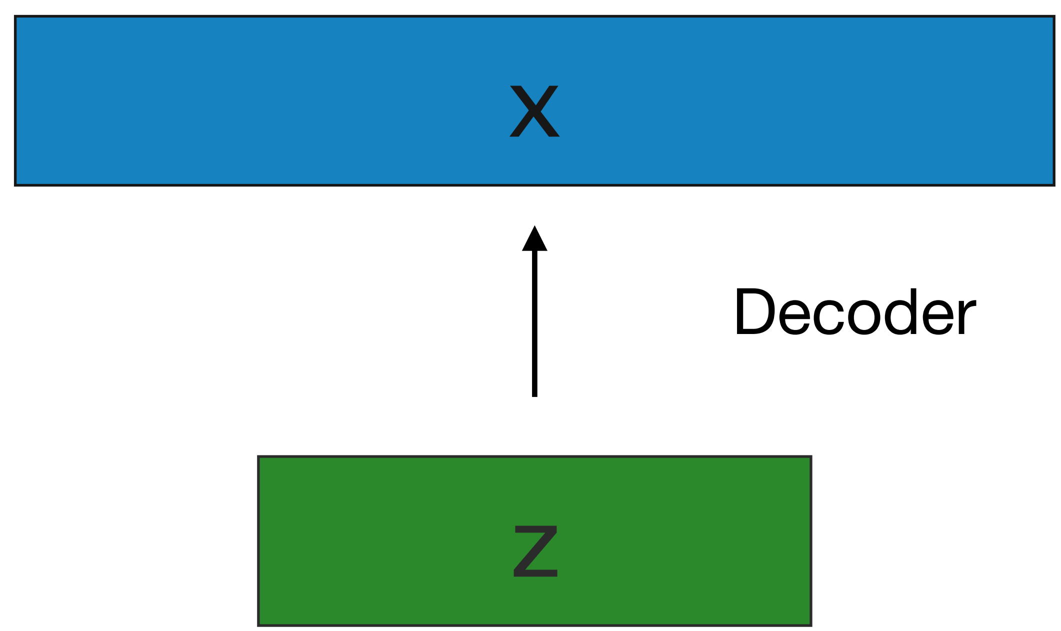
Conditional p(x|z) is for generating image
=> represent with neural network

Variational Autoencoder (VAE)

- Probabilistic spin on autoencoders: sample from the generative model then to generate data!

Sample from true conditional

$p_\theta(x \mid z)$



Sample from true prior

$p_\theta(z)$

θ: parameters of decoder.

How should we train this model?

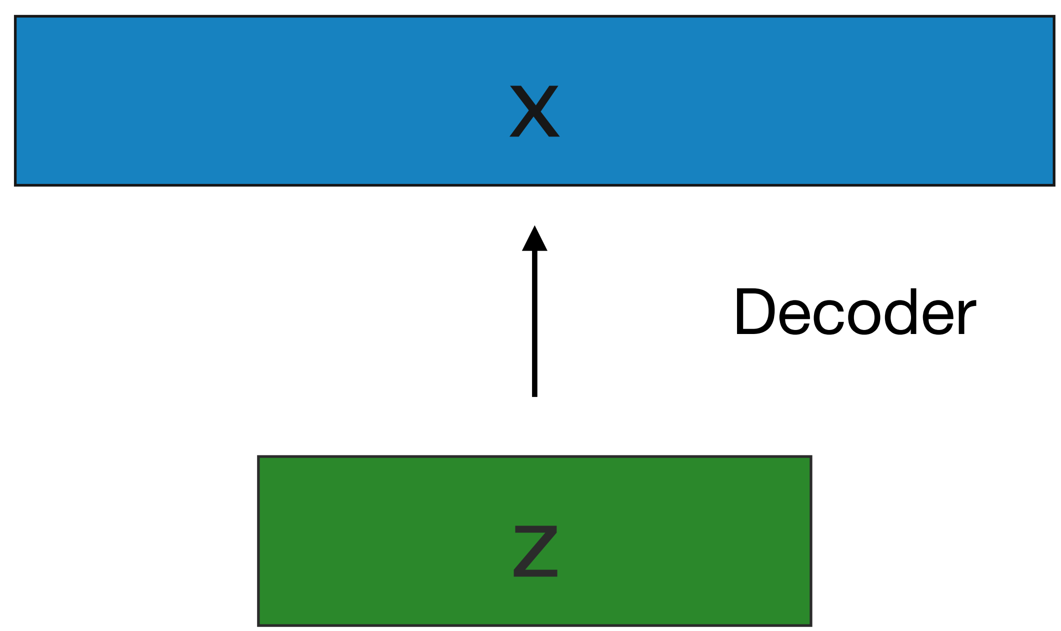
Learn model parameters to maximize likelihood of training data.

Variational Autoencoder (VAE)

- Probabilistic spin on autoencoders: sample from the generative model then to generate data!

Sample from true conditional

$p_\theta(x \mid z)$



Sample from true prior

$p_\theta(z)$

How should we train this model?

θ: parameters of decoder.

$$p_\theta(x) = \int p_\theta(z) p_\theta(x \mid z) dz$$

Variational Autoencoder (VAE)

- Data likelihood:

$$p_\theta(x) = \int p_\theta(z) p_\theta(x \mid z) dz$$

Variational Autoencoder (VAE)

- Data likelihood:

$$p_\theta(x) = \int p_\theta(z) p_\theta(x \mid z) dz$$

Intractable!

- Posterior:

$$p_\theta(z \mid x) = p_\theta(x \mid z) p_\theta(z) / p_\theta(x)$$

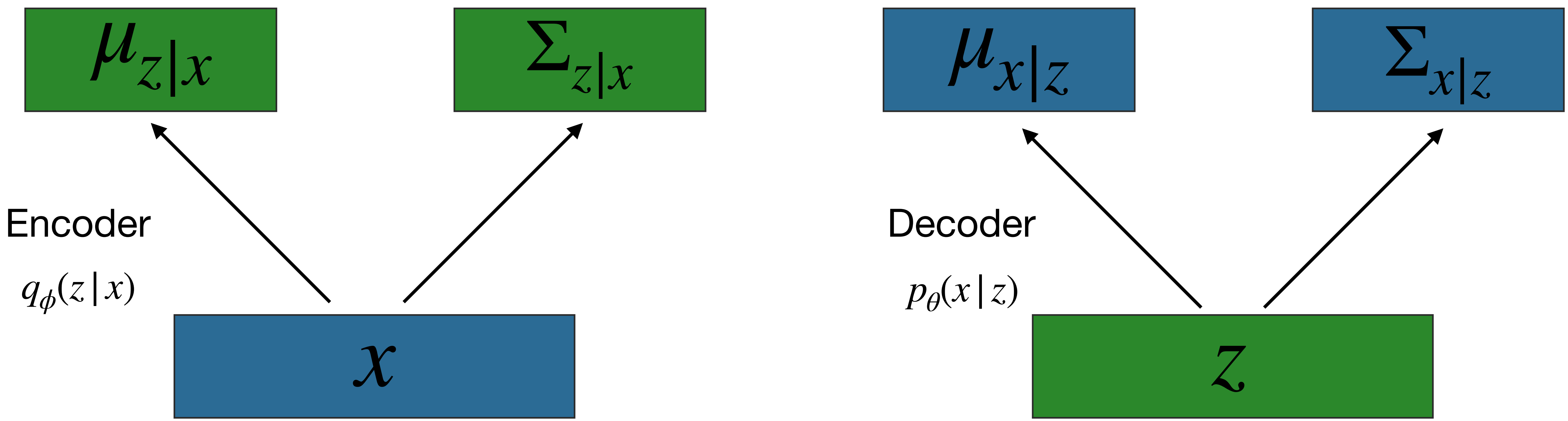
- Solution:

In addition to decoder network modeling $p_\theta(x \mid z)$, define additional encoder network $q_\phi(z \mid x)$ that approximates $p_\theta(z \mid x)$.

Variational Autoencoder (VAE)

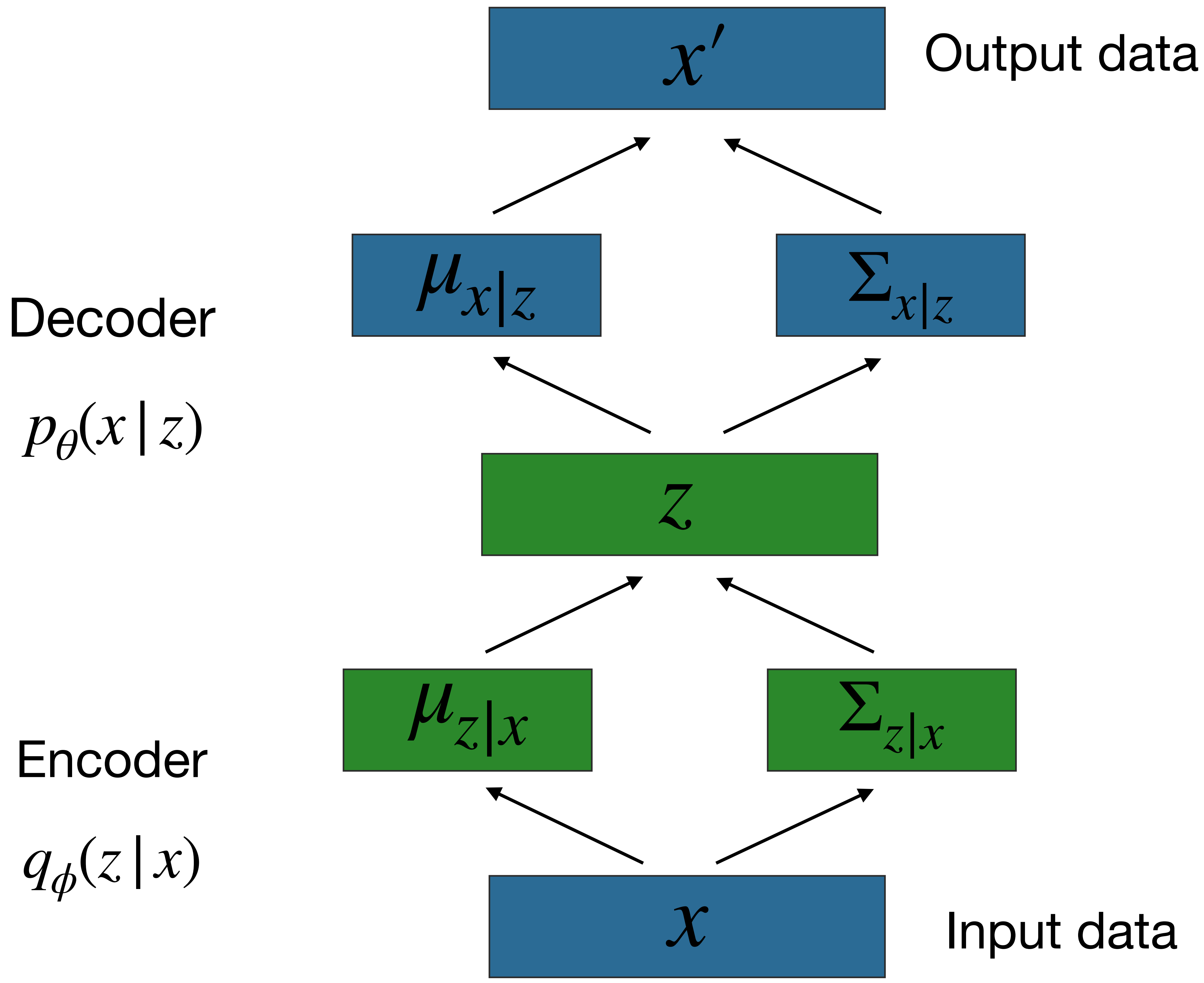
Sample z from a Gaussian
of latent representation

Sample x|z from a Gaussian
of output data



With this equipped neural network, now
let's work out the data likelihood!

Variational Autoencoder (VAE)

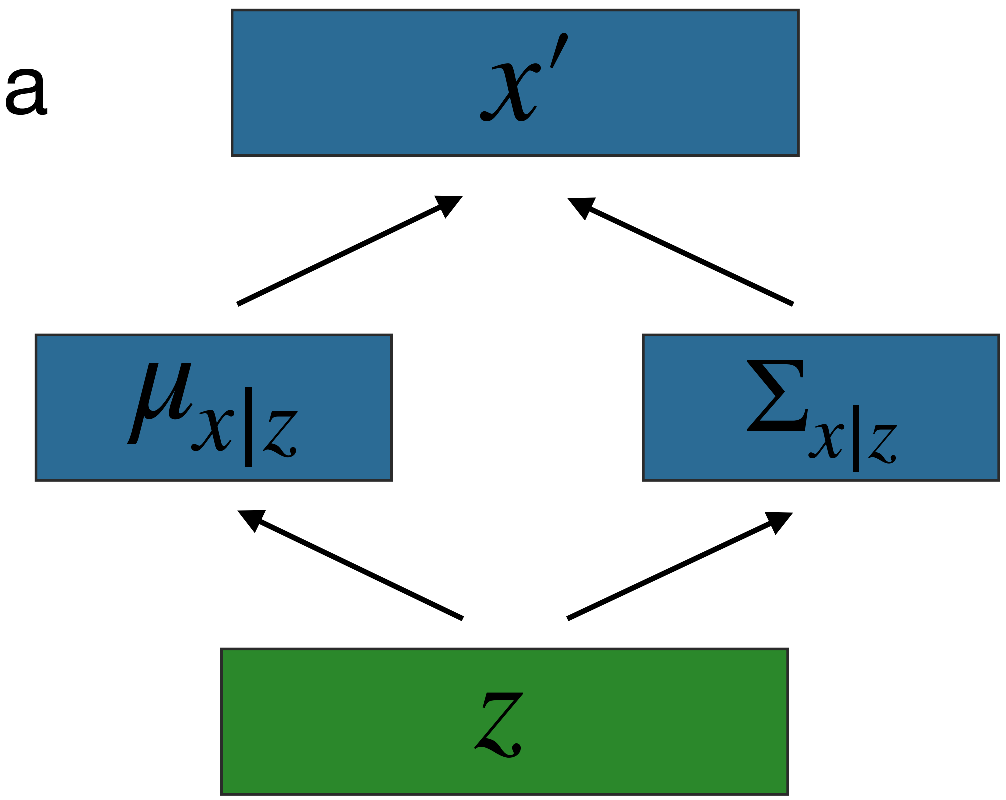


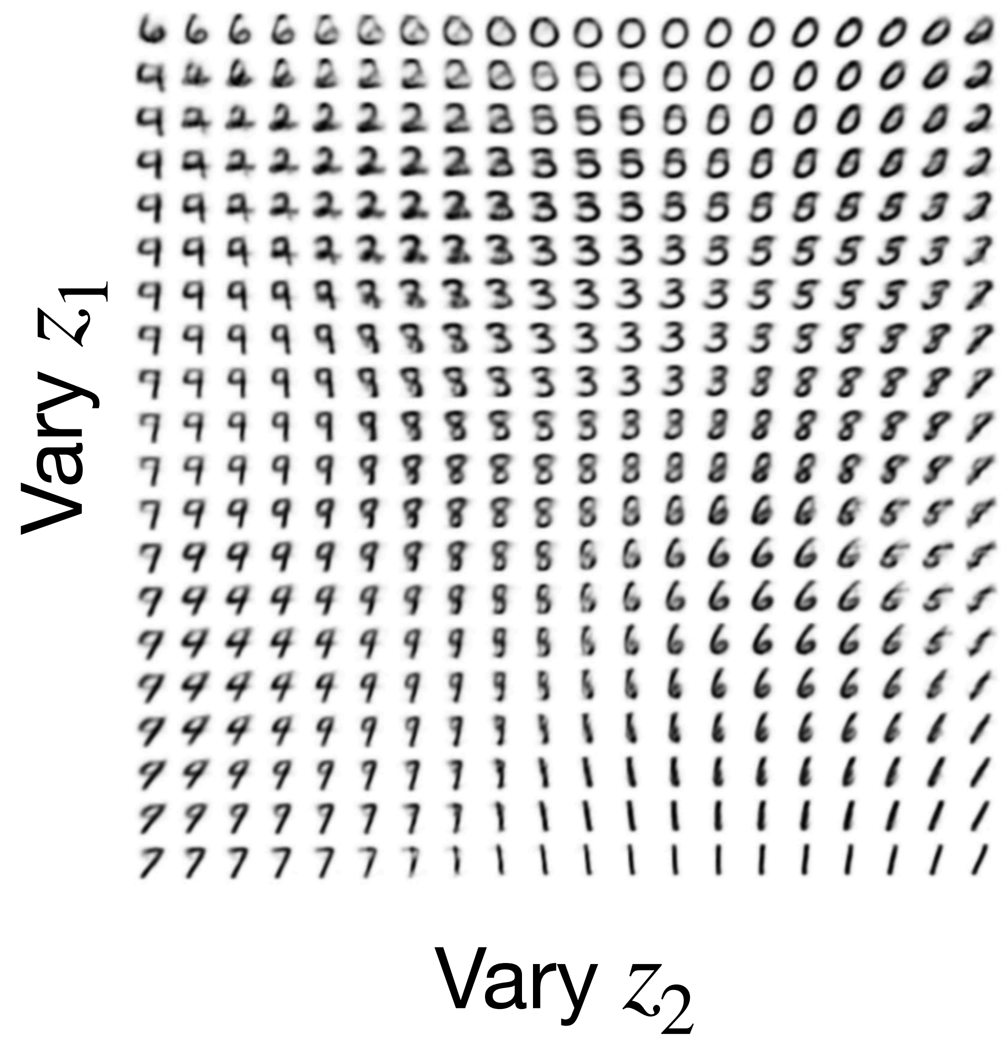
Variational Autoencoder (VAE)

Output data

Decoder

$p_\theta(x \mid z)$



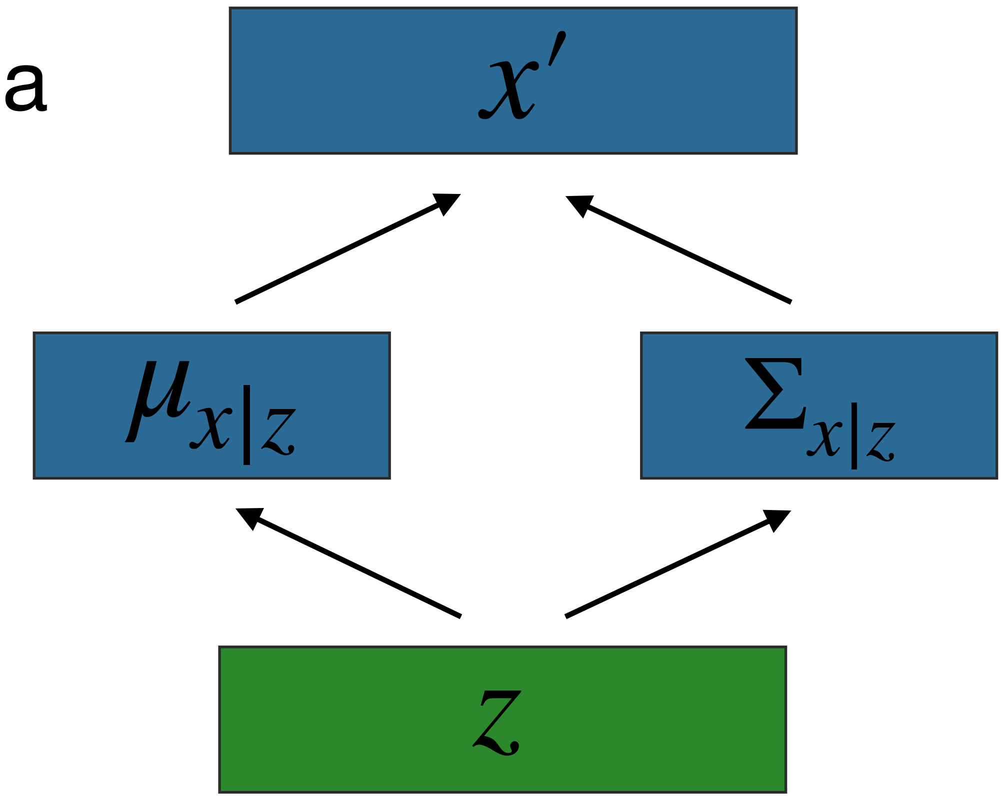


Variational Autoencoder (VAE)

Output data

Decoder

$p_\theta(x \mid z)$

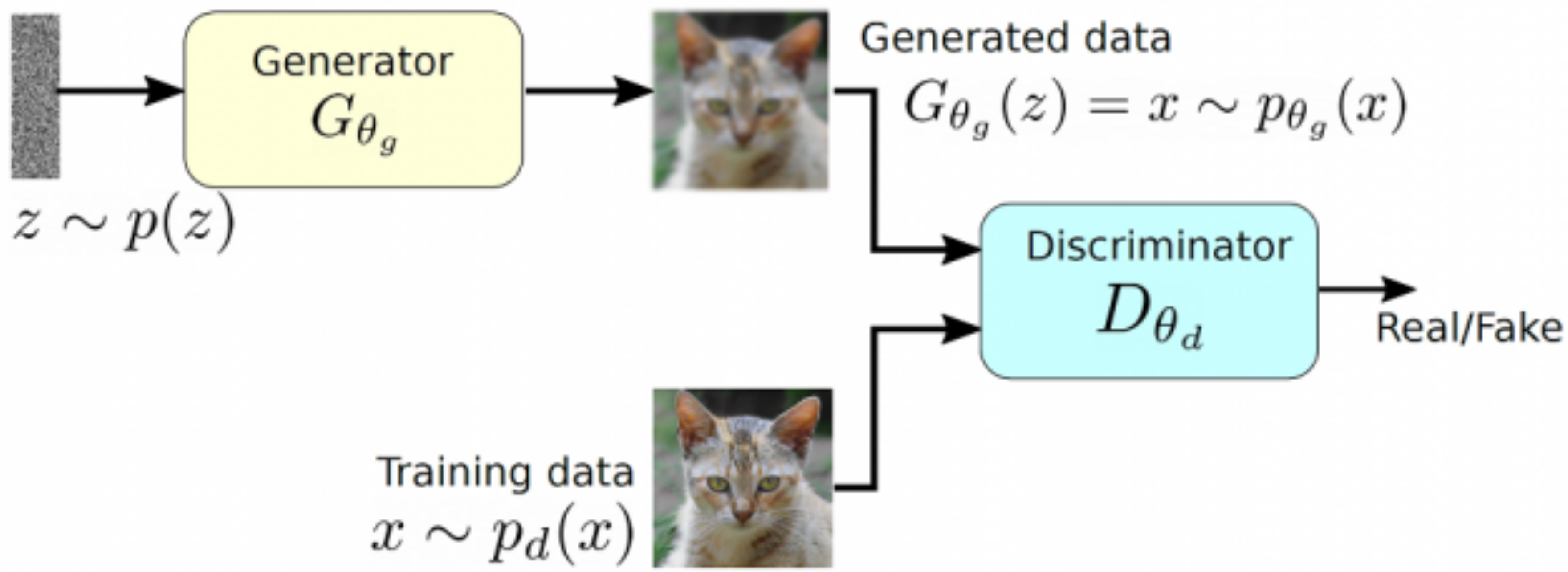


Vary z_1

Vary z_2

Advanced generative model

- Generative adversarial network (GAN)



- Discriminator takes that image and predicts whether the image belongs to a target distribution
- GANs generally produce better photo-realistic images