

Generative Models I

Dr. Parlett-Pelleriti

Outline

- Generative Model Examples
- Types of Generative Models
- Simple Generative Models
- Autoencoders Review
- Variational Autoencoders

Generative Models Examples

Generating New Faces



Image from: <https://github.com/tdrussell/IllustrationGAN>



Image from: <https://arxiv.org/pdf/1710.10196v3.pdf>

Generating Faces

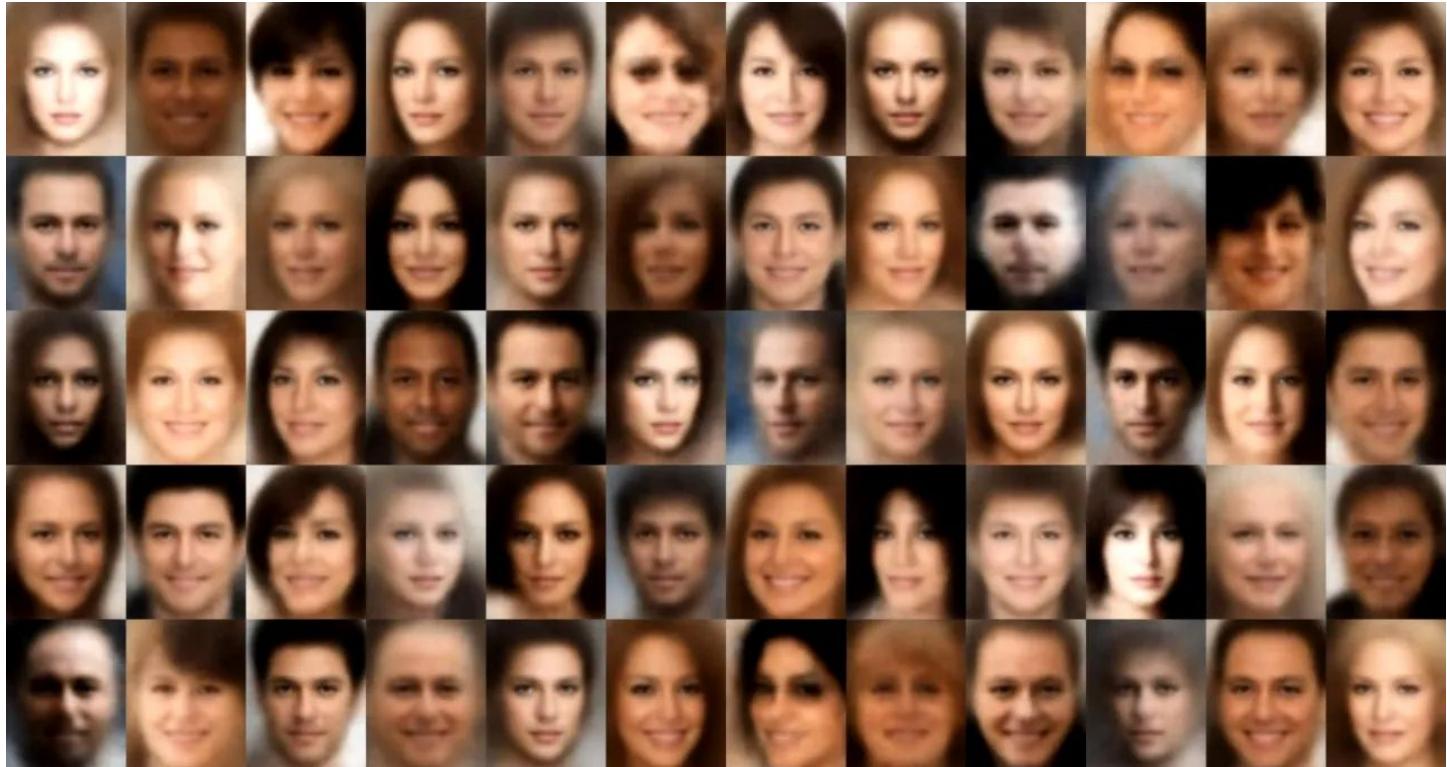


Image from: <https://github.com/wojciechmo/vae>

Generating Photos/Art

machine learning professor

Run



Neural Inpainting GANs

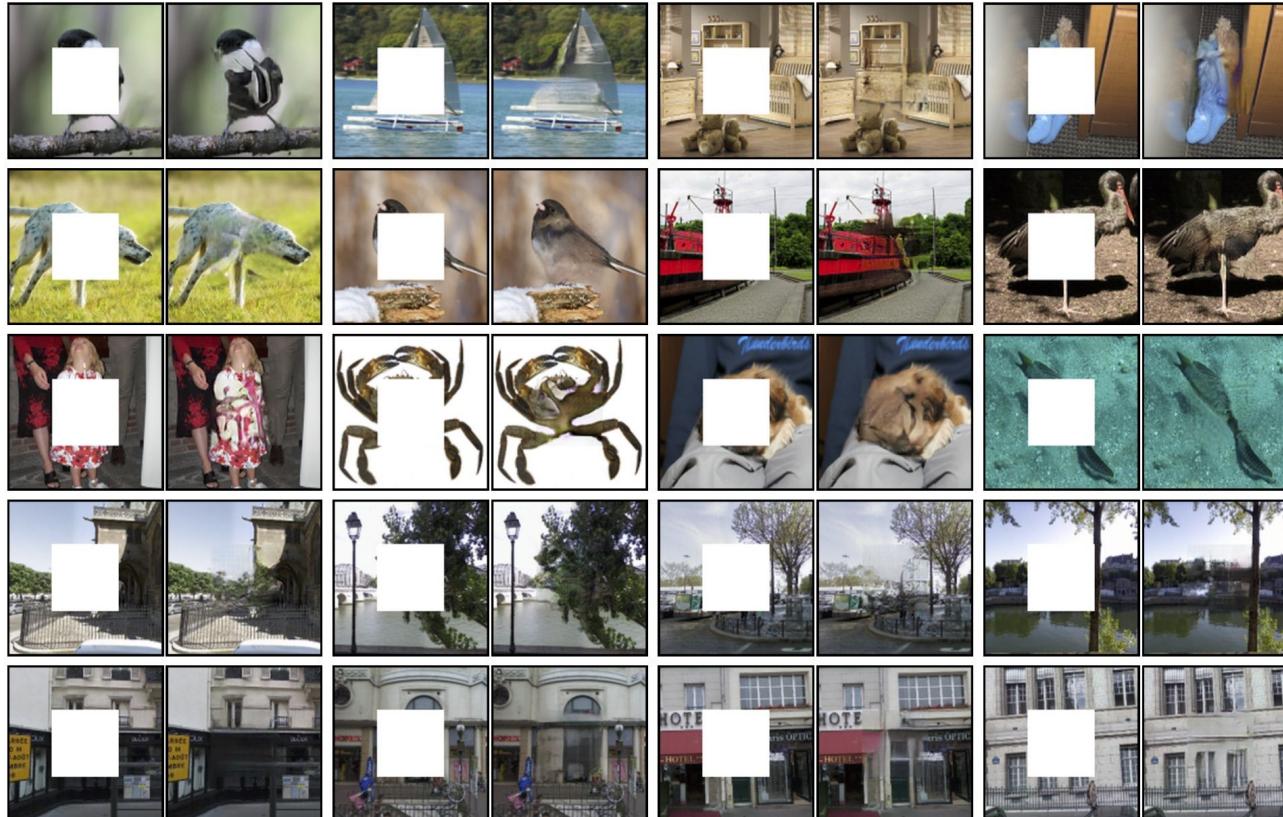


Image from: <https://github.com/pathak22/context-encoder>

Generating Text

C

What is Generative AI?



Generative AI refers to a subset of artificial intelligence techniques that involve creating new content or data, such as images, videos, audio, or text, based on patterns and insights learned from large datasets. These systems are designed to generate content that is similar in style, structure, and content to the training data but is not an exact copy of it.



Generative AI algorithms use deep learning models that learn patterns in the data and use those patterns to generate new content. For example, a generative AI model could be trained on a large dataset of images of dogs, and then generate new, realistic images of dogs that do not exist in the original dataset.

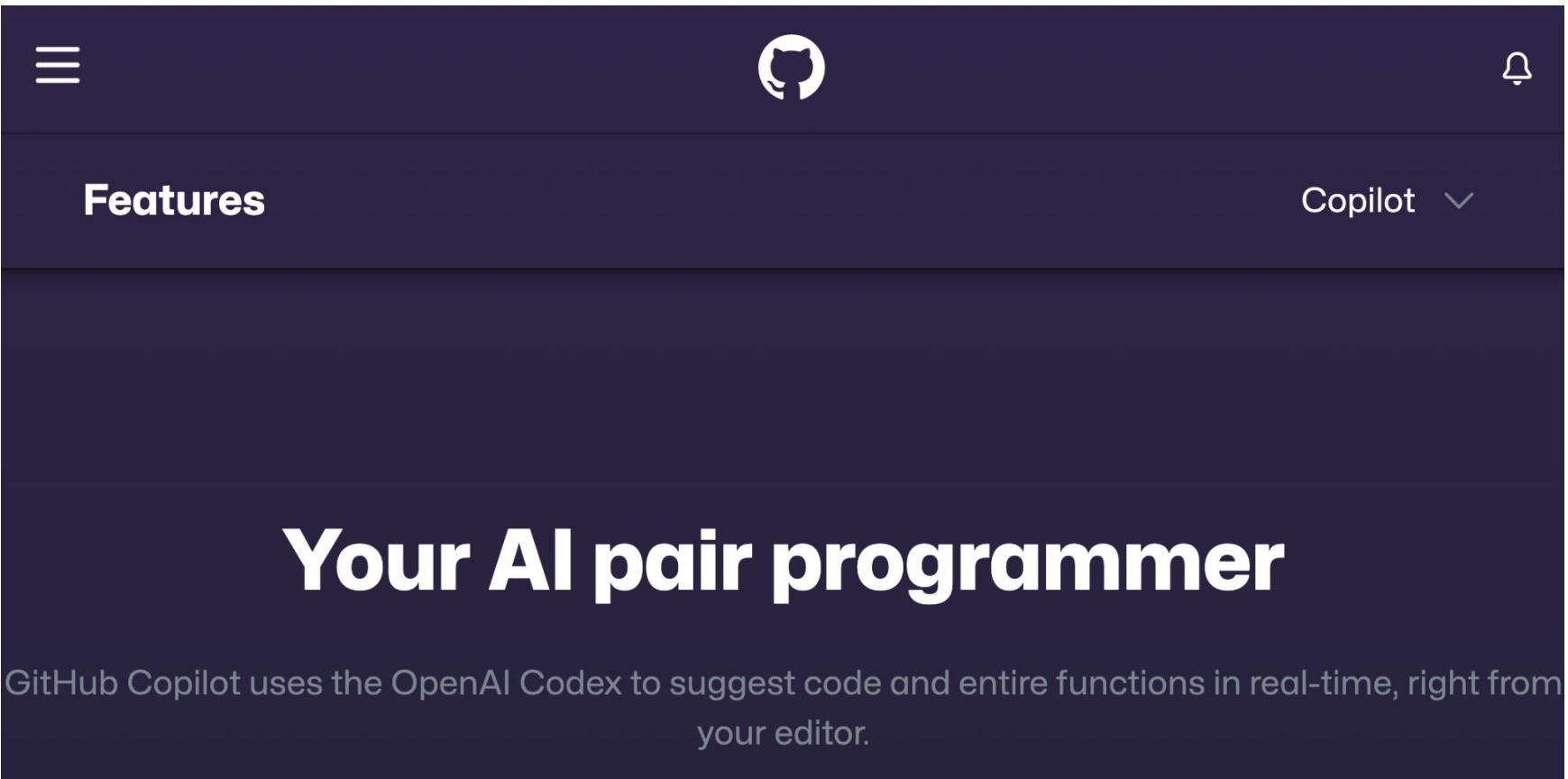
Regenerate response

Generating Text



Image from: <https://www.twitch.tv/watchmeforever> via
<https://techcrunch.com/2023/02/03/nothing-forever-ai-generated-seinfeld-twitch/>

Generating Code

A screenshot of the GitHub Copilot interface. At the top, there's a dark header bar with a menu icon (three horizontal lines), the GitHub logo, and a bell icon for notifications. Below the header, the word "Features" is displayed in white. To the right of "Features", there's a "Copilot" button with a dropdown arrow. The main content area features a large, bold, white text "Your AI pair programmer". Below this, a smaller, lighter gray text explains the functionality: "GitHub Copilot uses the OpenAI Codex to suggest code and entire functions in real-time, right from your editor."

≡

GitHub

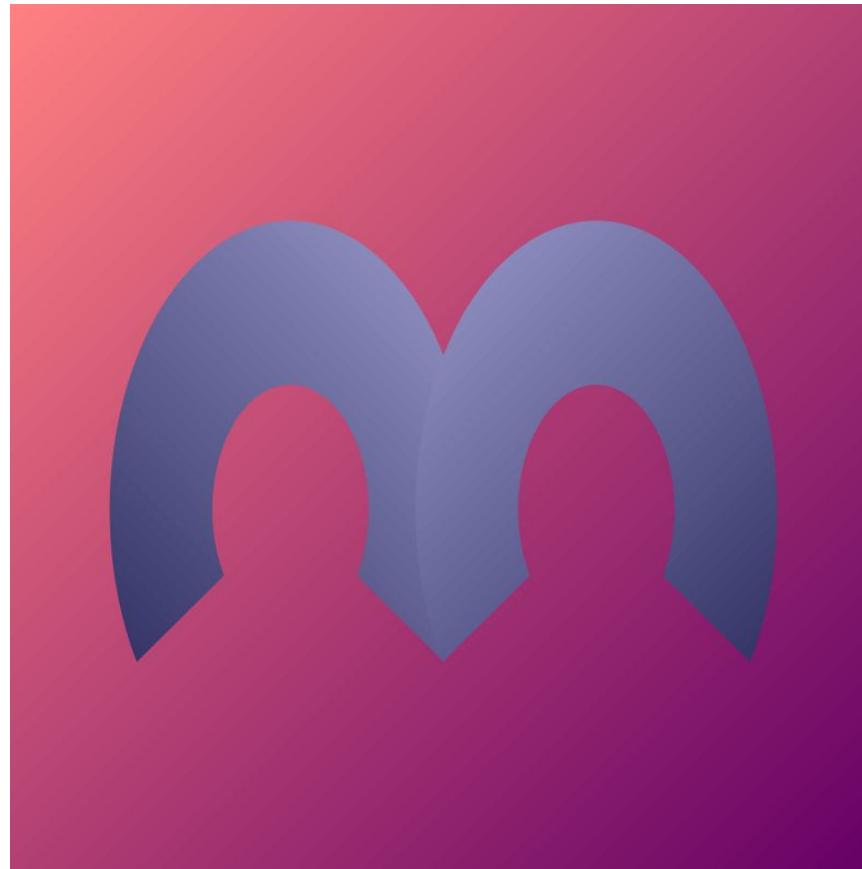
Features

Copilot ▾

Your AI pair programmer

GitHub Copilot uses the OpenAI Codex to suggest code and entire functions in real-time, right from your editor.

Generating Music



Generating Video Game Levels

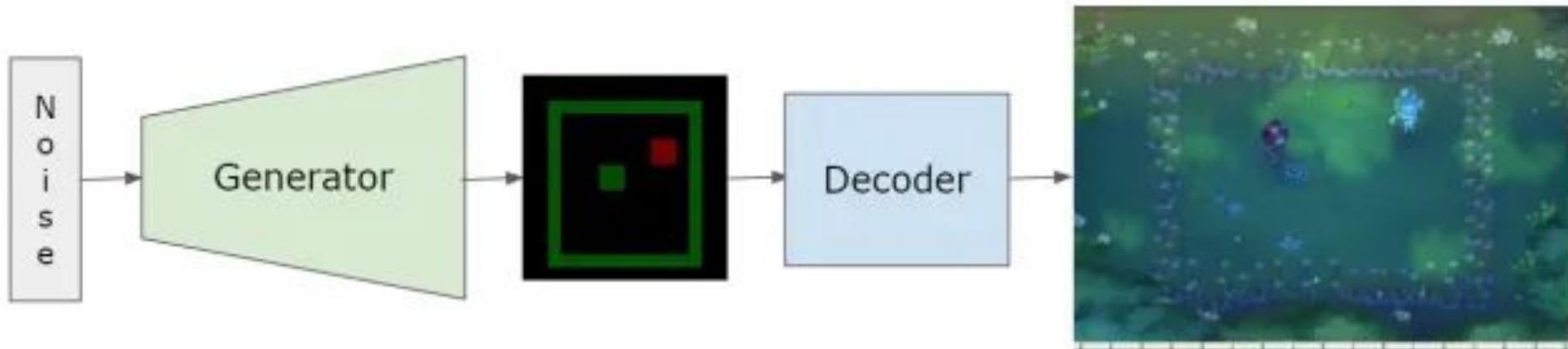


Image from: <https://www.gamedeveloper.com/programming/game-level-generation-using-neural-networks>

Style Transfer

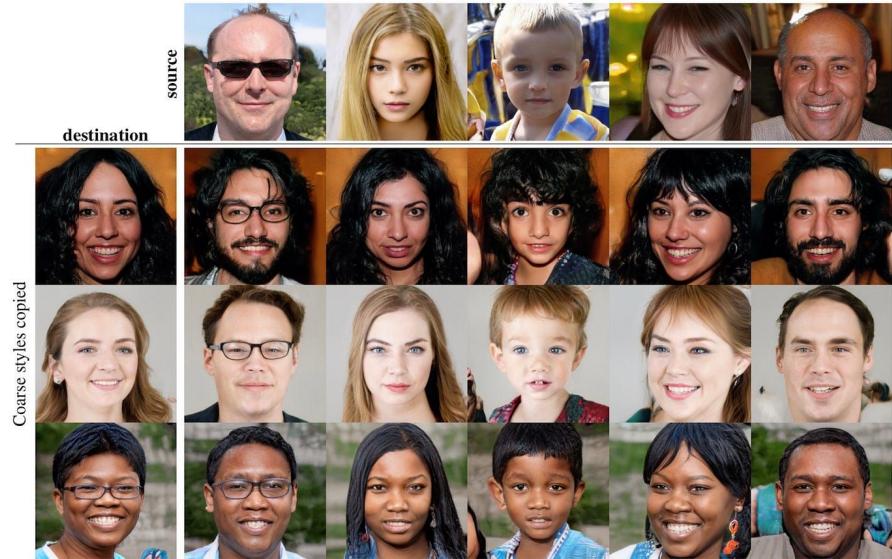


Image from: <https://arxiv.org/abs/1812.04948>



Image from:
https://www.reddit.com/r/Breath_of_the_Wild/comments/eio394/botw_x_starry_night_neural_style_transfer/

Face Aging GAN

0-18



19-29



30-39



40-49



50-59



60+

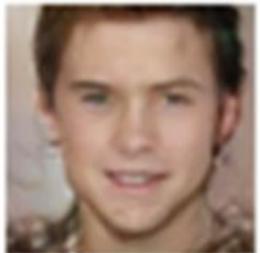


Image from: <https://arxiv.org/pdf/1702.01983.pdf>

Image Upscaling GANs

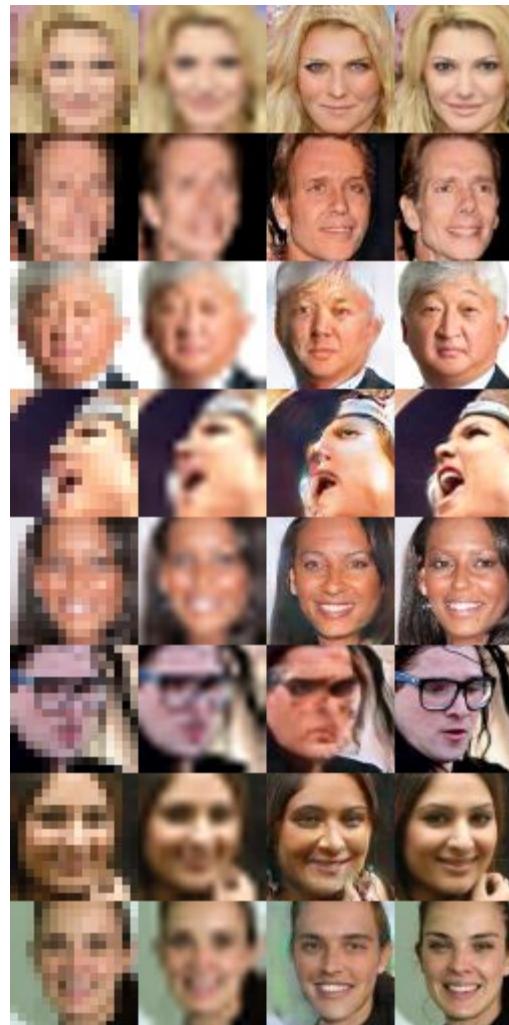


Image from: <https://github.com/david-gpu/srez>

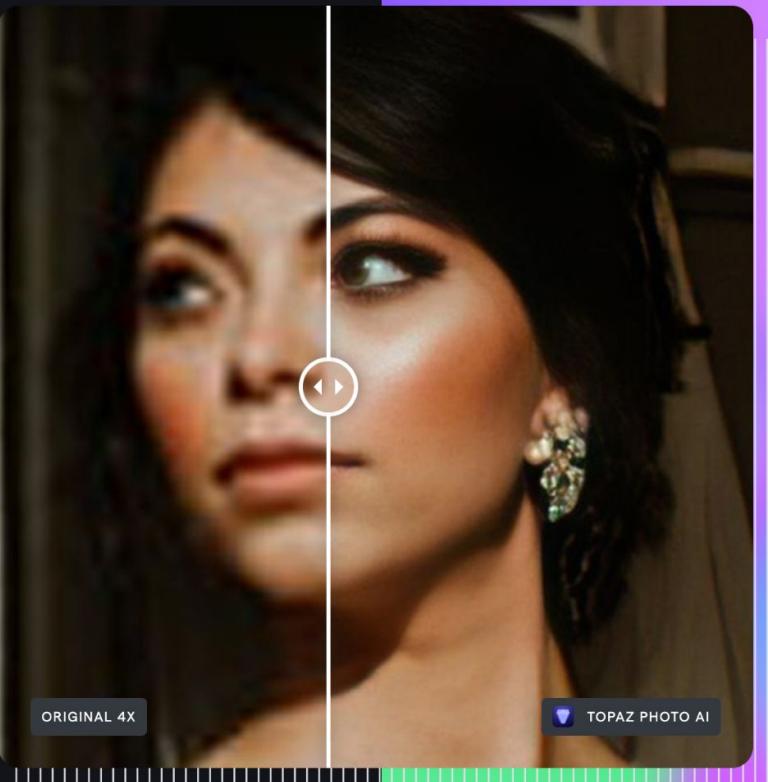
Image Quality Improvement

New: Topaz Photo AI >

Image quality powered by AI.

Magically improve your photo and video quality with cutting-edge image enhancement technology.

[Learn more](#)



The image shows a woman's face split vertically down the middle. The left side is labeled "ORIGINAL 4X" and the right side is labeled "TOPAZ PHOTO AI". A white vertical line separates the two sides. In the center of the line is a circular icon with a left-pointing arrow on the left and a right-pointing arrow on the right. The background of the main image is dark, and there is a purple border around the entire image frame.

ORIGINAL 4X

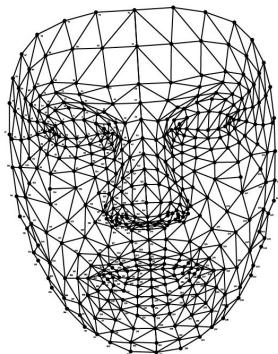
TOPAZ PHOTO AI

Deep Fakes



Image from:
<https://thedirect.com/article/star-wars-luke-skywalker-deepfakes-cgi>

TikTok Generative Filters



468 points

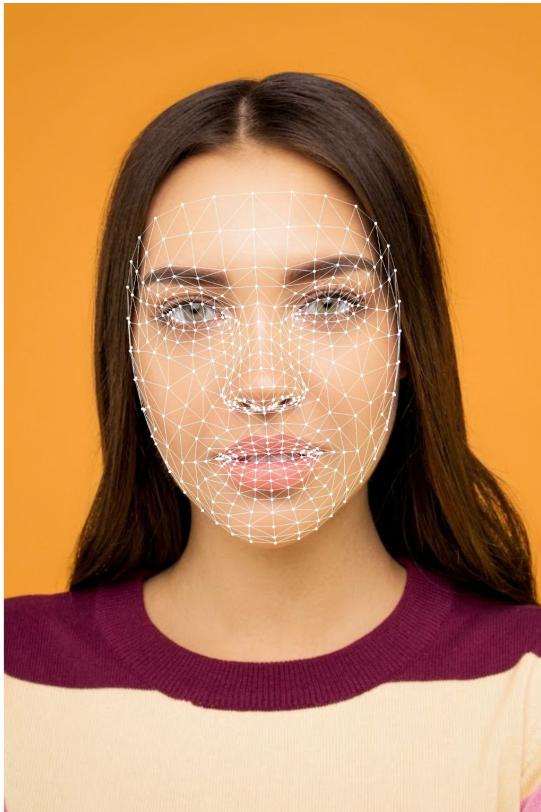


Image from: <https://developers.google.com/ml-kit/vision/face-mesh-detection>



Image from:
<https://www.theverge.com/2023/3/2/23621751/bold-glamour-tiktok-face-filter-beauty-ai-ar-body-dismorphia>

Scam Robo-Calls

Scammers are using AI to impersonate your loved ones. Here's what to watch out for

The next time you get a call from a family member or friend in need, you might want to make sure it's not a robot first.



Written by **Sabrina Ortiz**, Associate Editor on March 9, 2023



Kiito Chan/Getty Images

Image from: Kiito Chan/Getty Images via
<https://www.zdnet.com/article/scammers-are-using-ai-to-impersonate-your-loved-ones-heres-what-to-watch-for/>

NUTANIX
Multicloud Simplified

[Find out more](#)

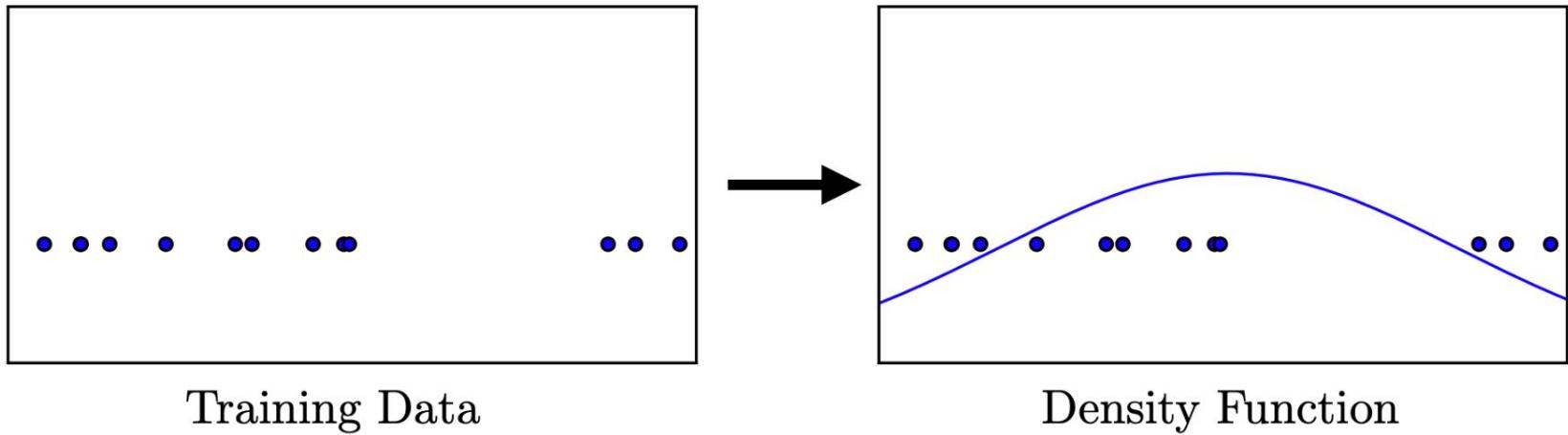
/ related

ChatGPT
Learned a modern language in a few days
Open

OpenAI will pay you to hunt for ChatGPT bugs

Types of Generative Models

Generative Modeling: Density Estimation



Generative Modeling: Sample Generation



Training Data
(CelebA)

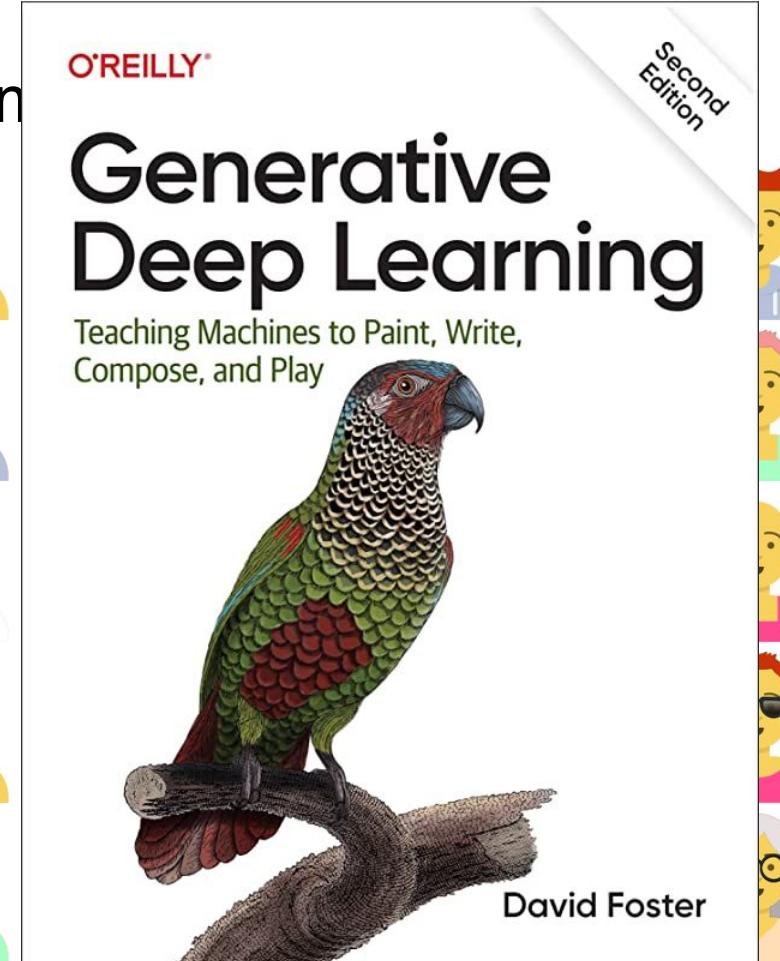


Sample Generator
(Karras et al, 2017)

Generative Models You Know

- Naive Bayes
- Gaussian Mixture Models

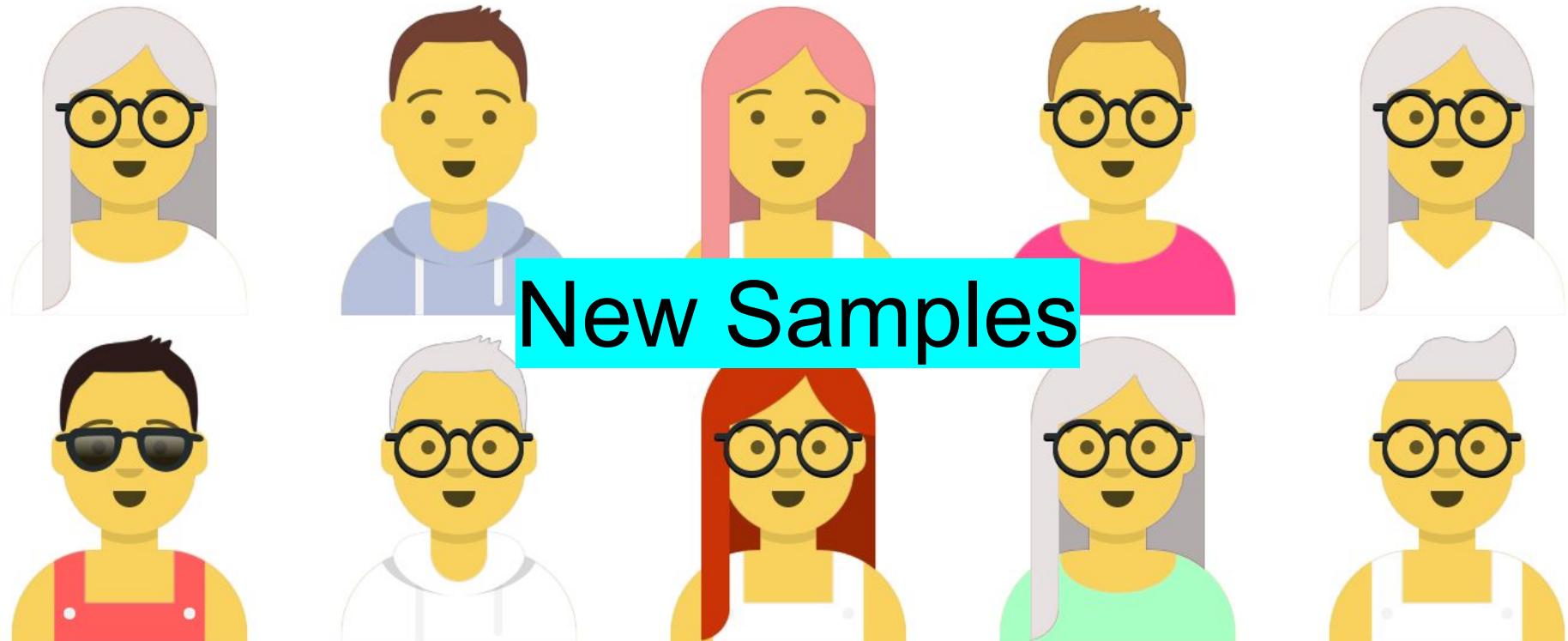
Naive Bayes Example



Naive Bayes Example



Naive Bayes

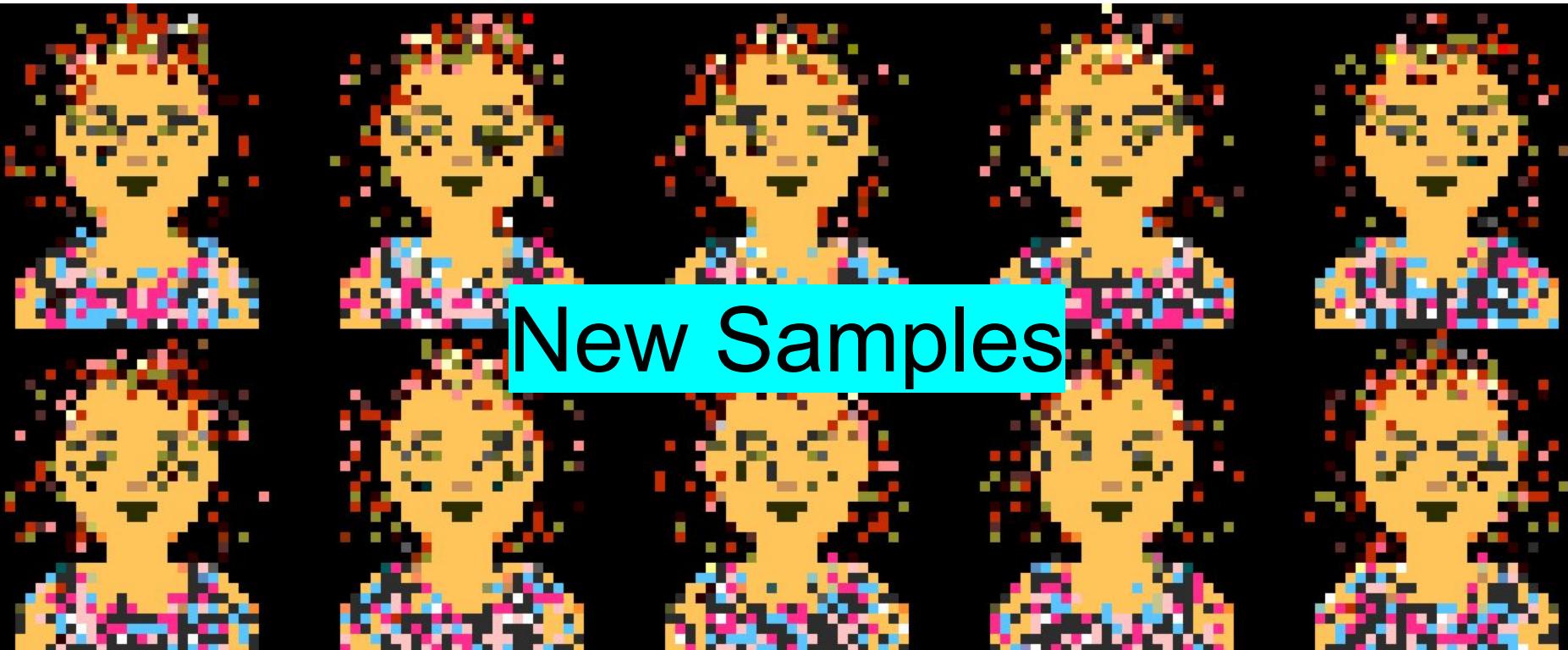


Naive Bayes (Unlabeled)



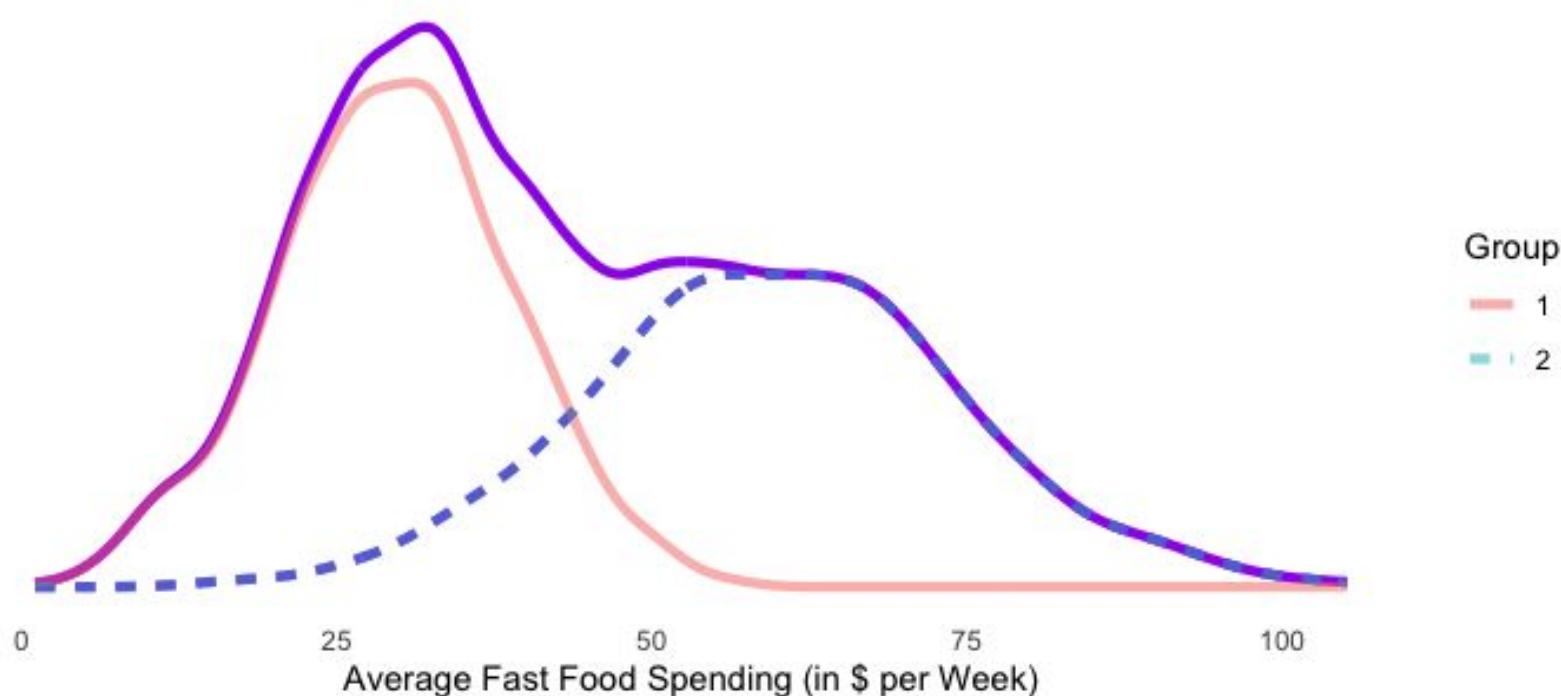
Image from: Generative Deep Learning: Teaching Machines to Paint, Write, Compose and Play by David Foster

Naive Bayes (Unlabeled)



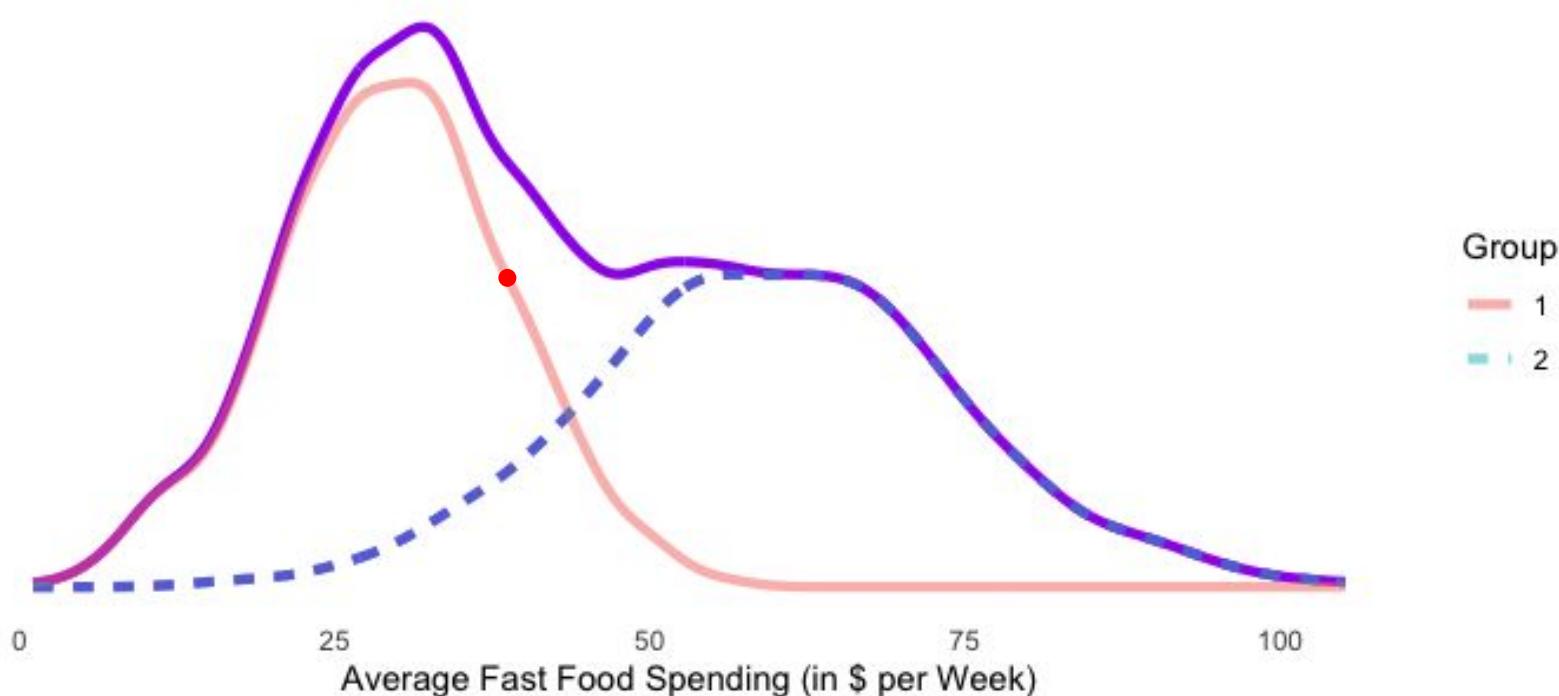
Gaussian Mixture Models

Average Fast Food Spending (Mixture Model)



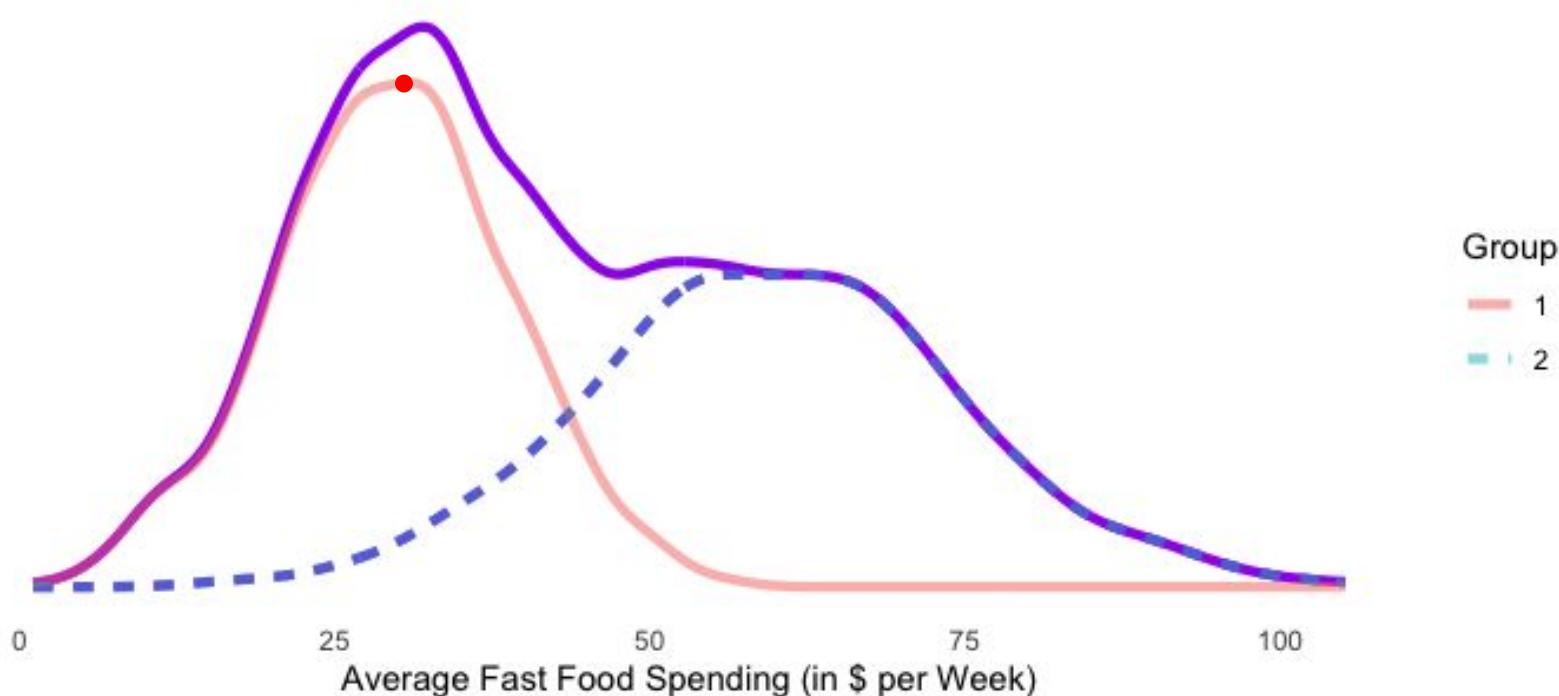
Gaussian Mixture Models

Average Fast Food Spending (Mixture Model)



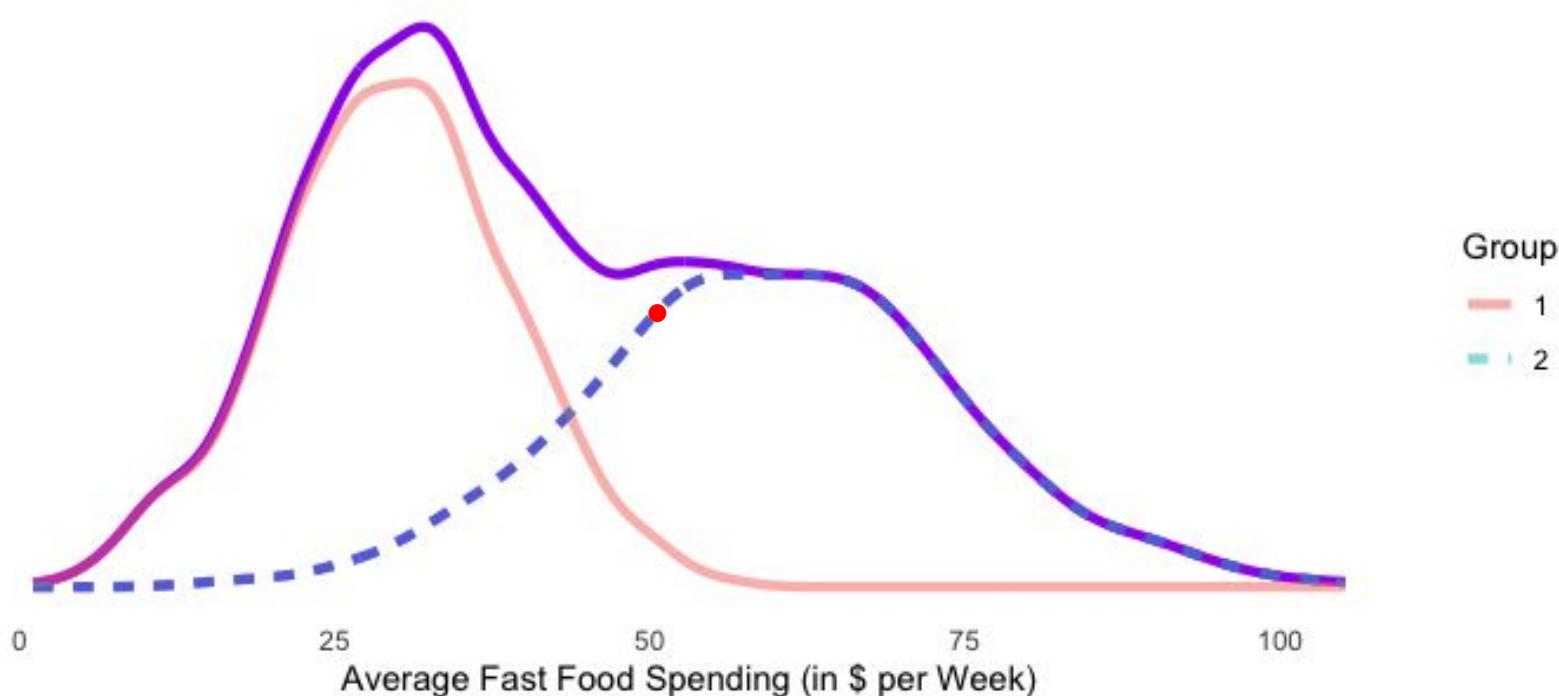
Gaussian Mixture Models

Average Fast Food Spending (Mixture Model)



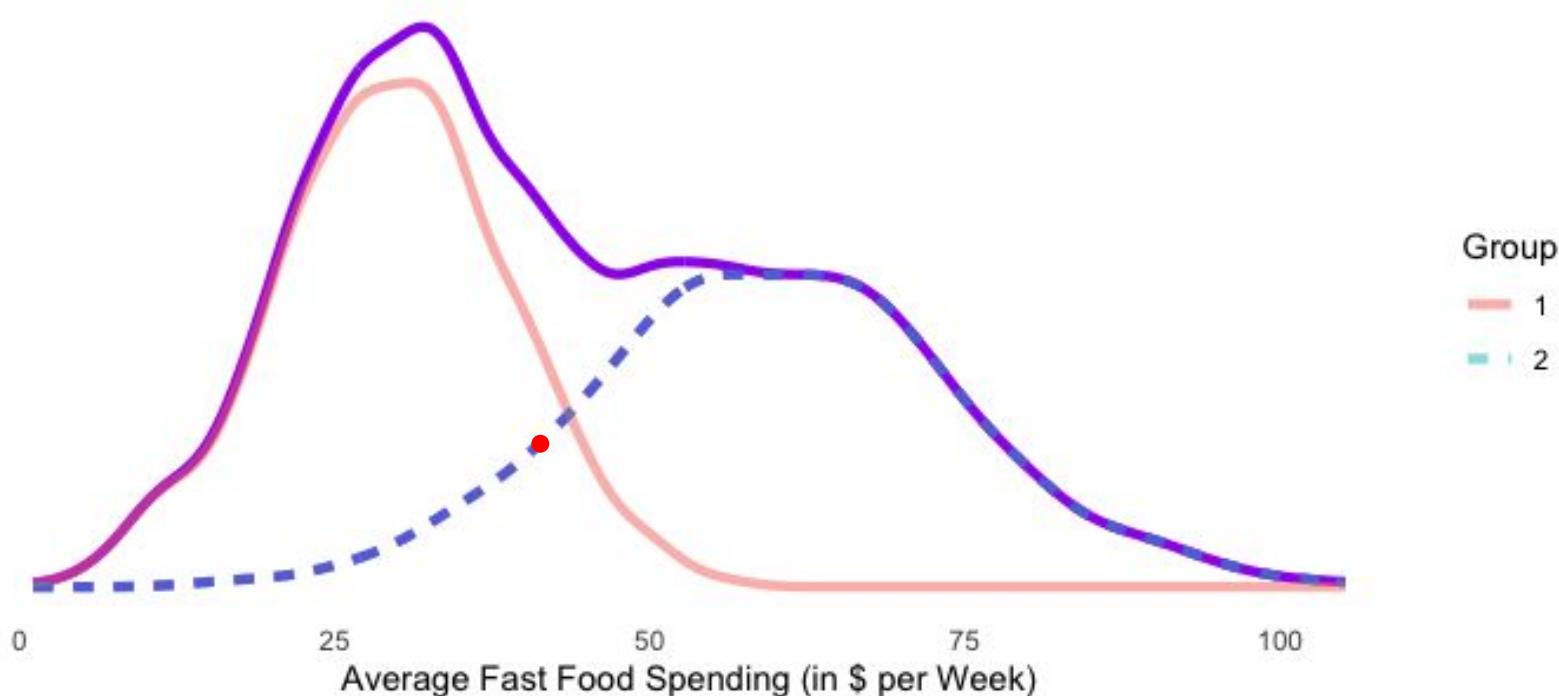
Gaussian Mixture Models

Average Fast Food Spending (Mixture Model)

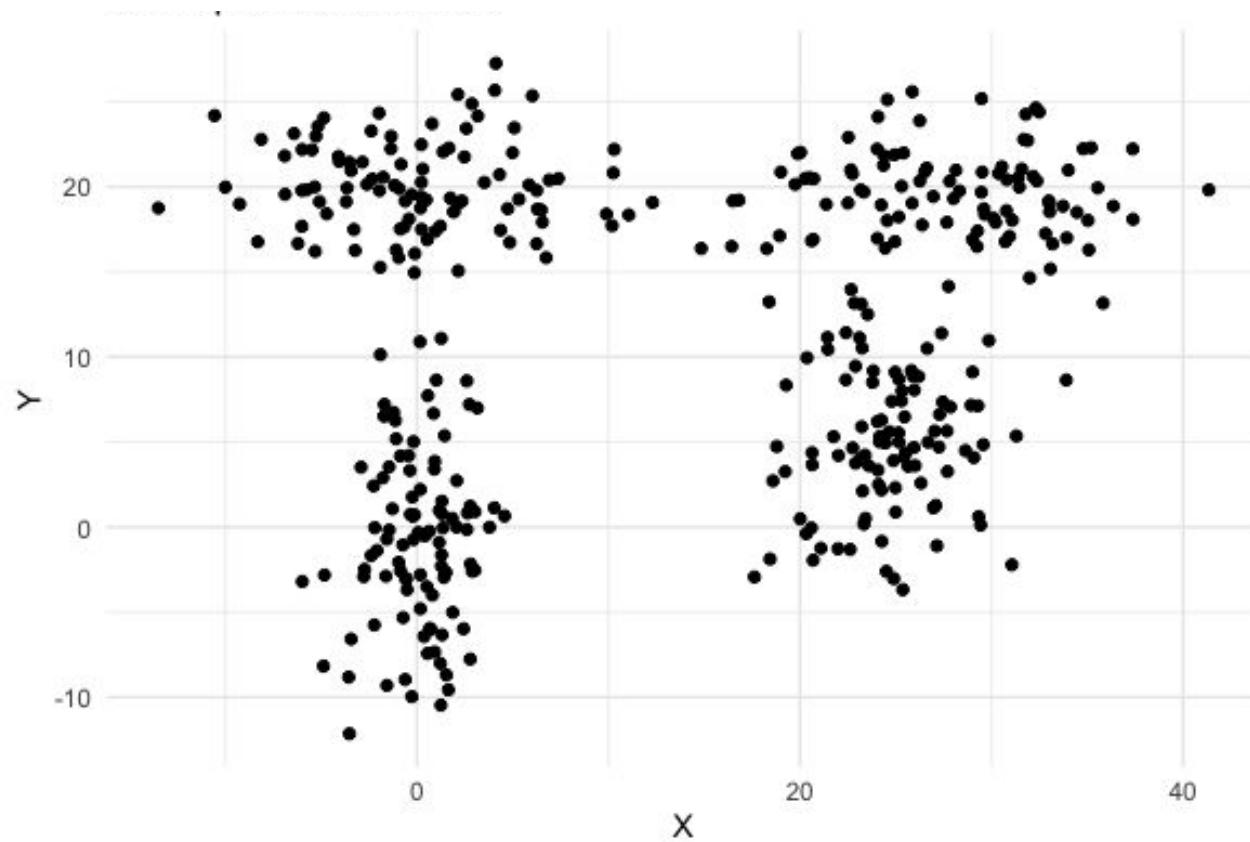


Gaussian Mixture Models

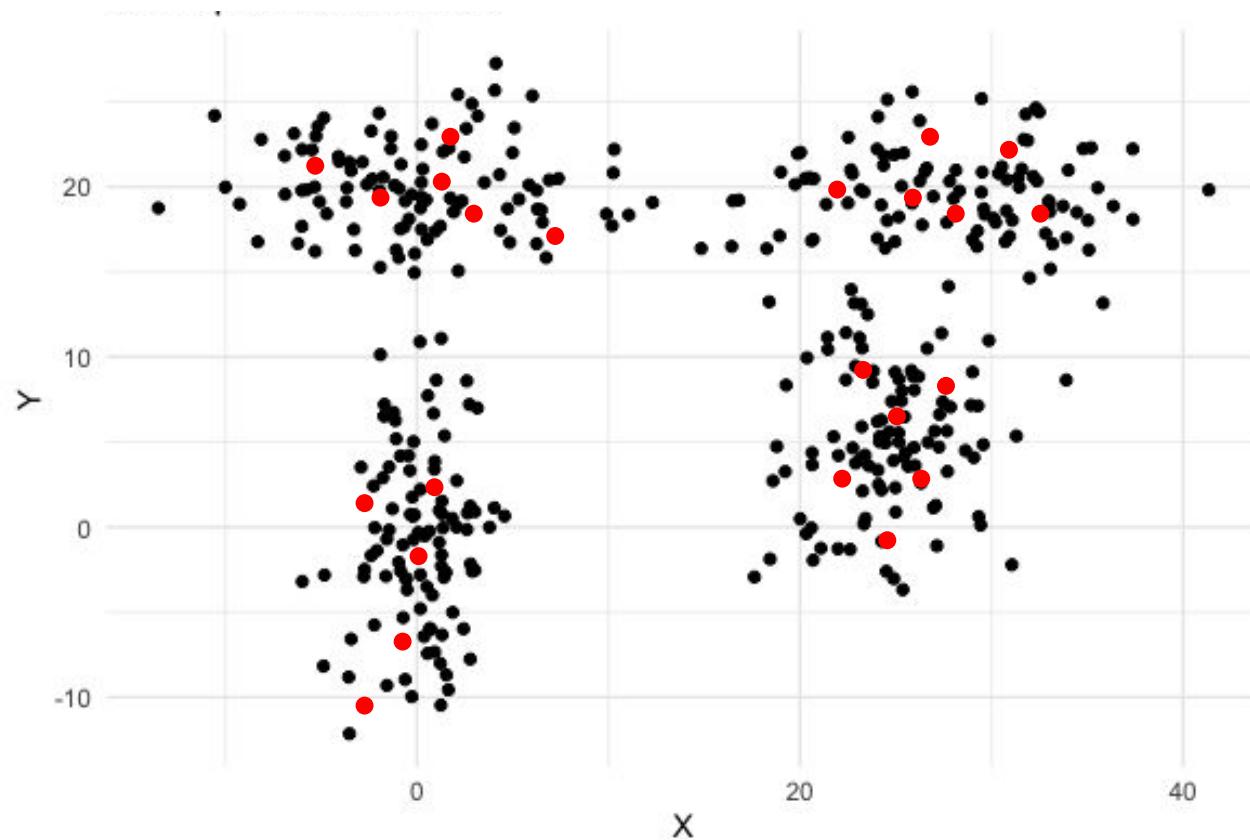
Average Fast Food Spending (Mixture Model)



Gaussian Mixture Models



Gaussian Mixture Models



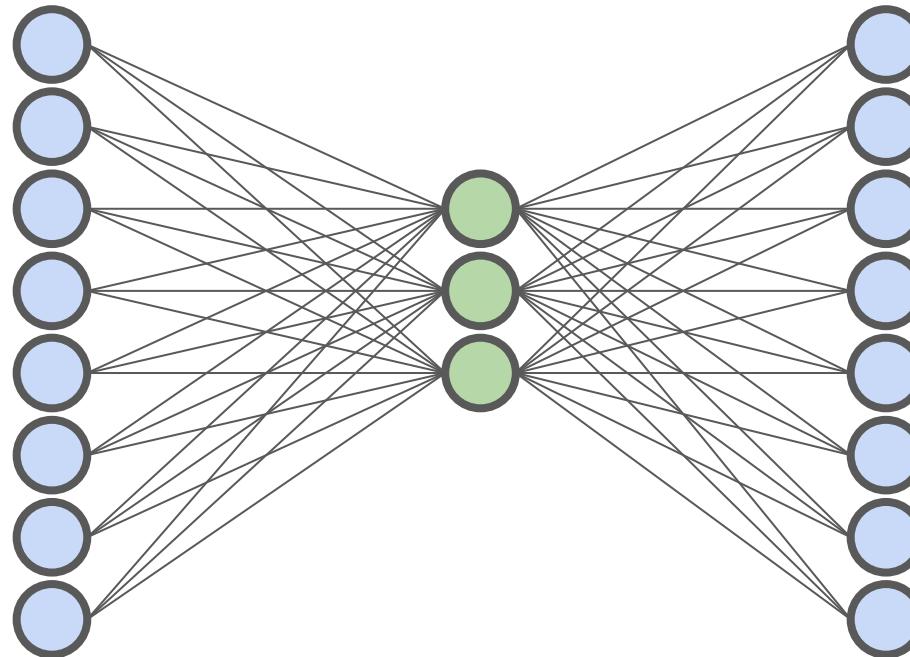
Density Estimation

Both **GMM** and **NB** do explicit density estimation, however they're limited

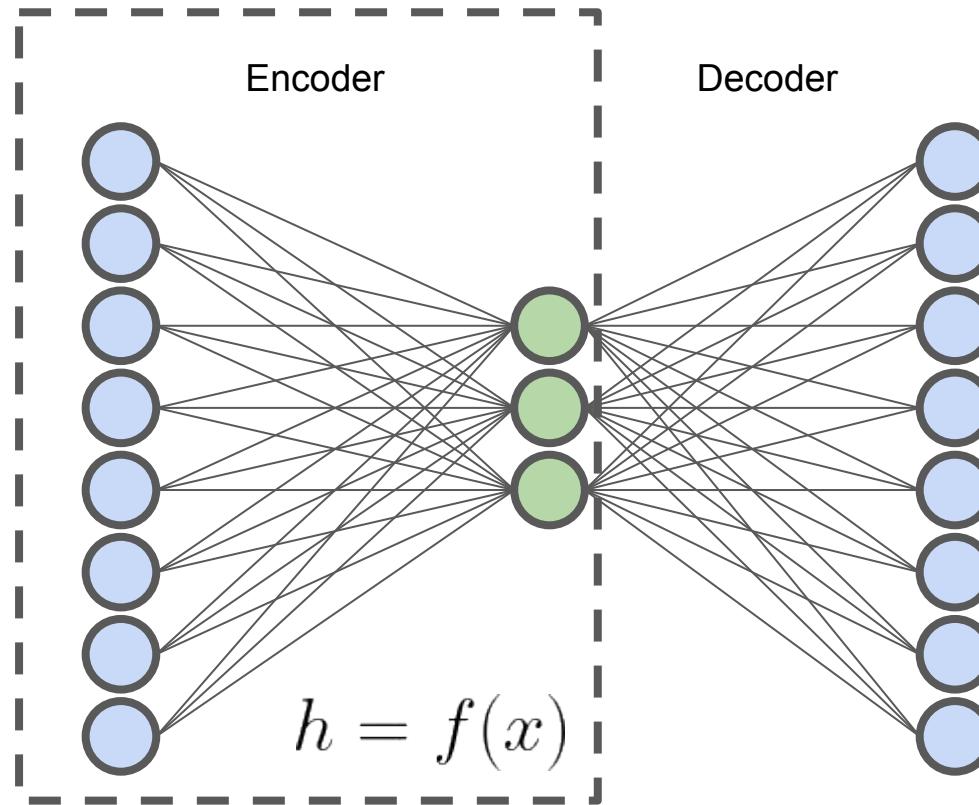
Enter: **Variational Autoencoders**

Autoencoders Review

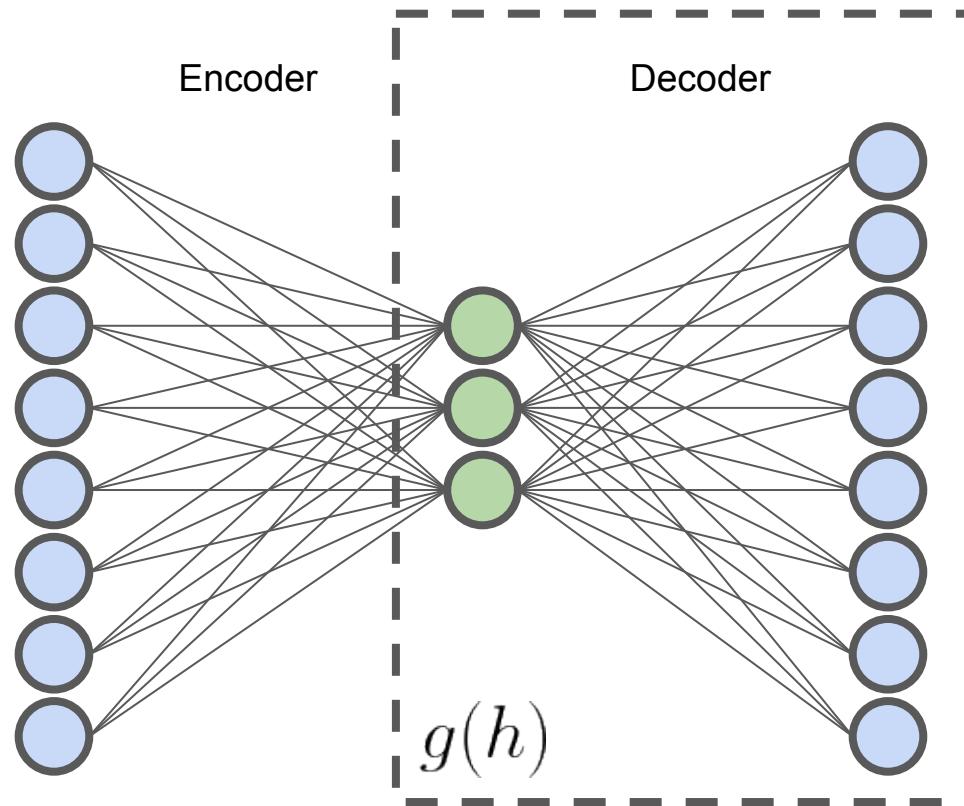
Autoencoder Architecture



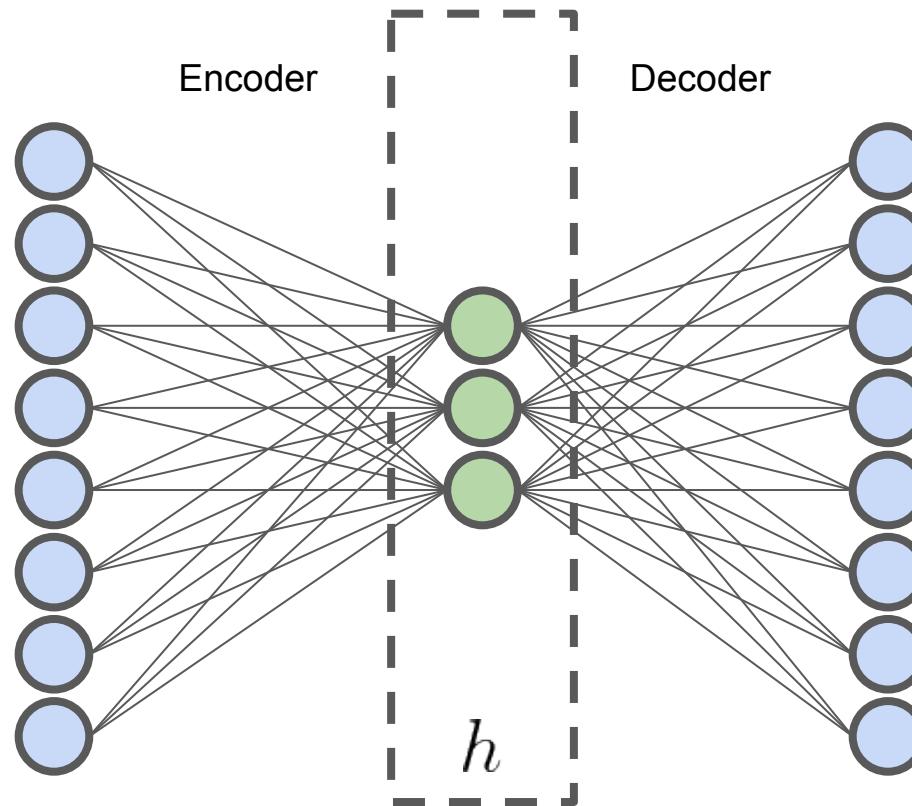
Autoencoder Architecture



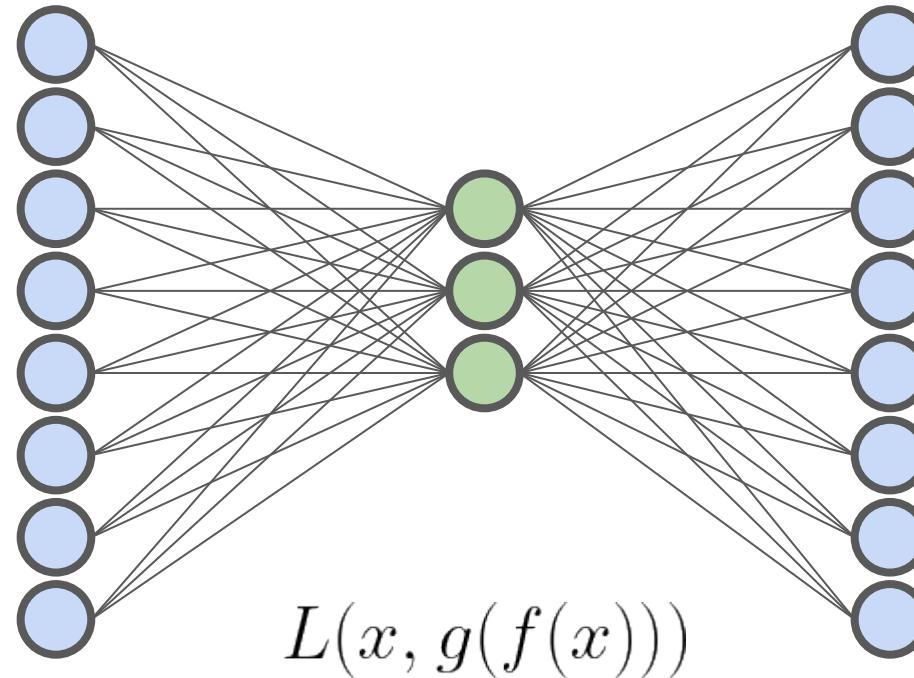
Autoencoder Architecture



Autoencoder Architecture



Autoencoder Architecture



Penalizing Derivatives

$$L(x, g(f(x))) + \boxed{\Omega(h, x)}$$

$$\boxed{\Omega(h, x)} = \lambda \sum_i \|\nabla_x h_i\|^2$$

Variational Autoencoders

Concept Vectors

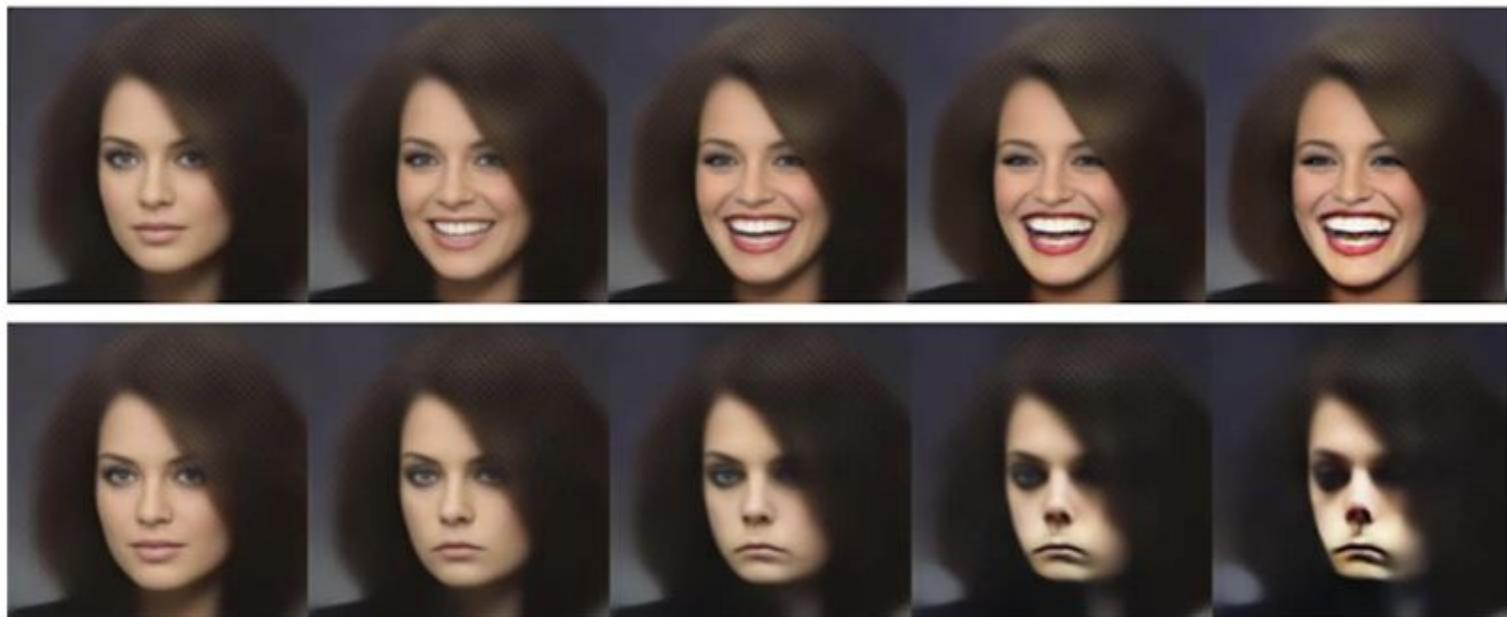
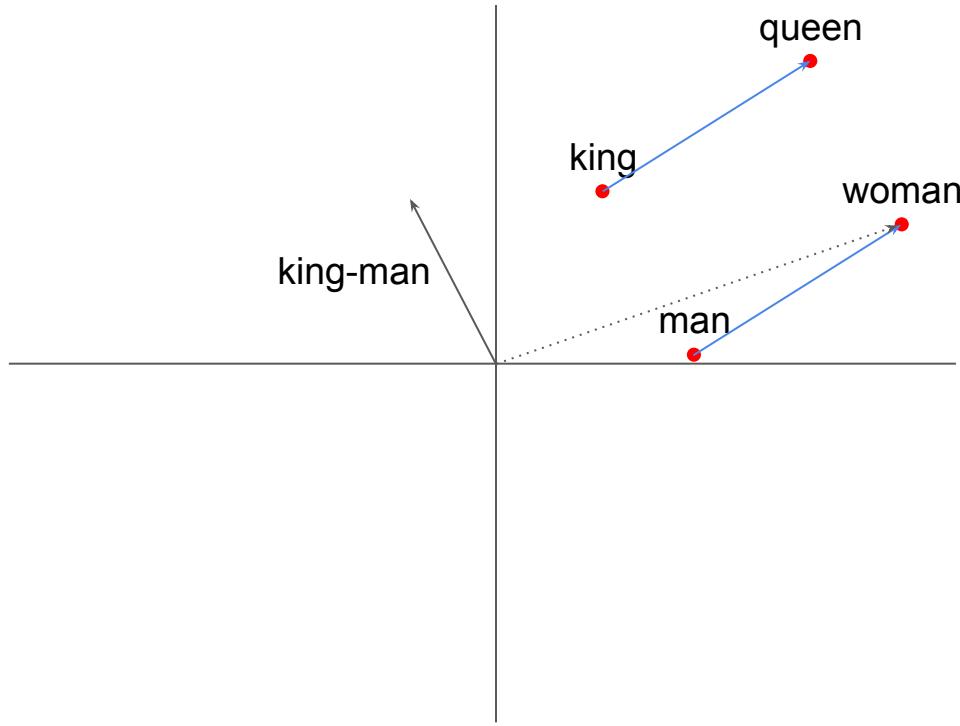
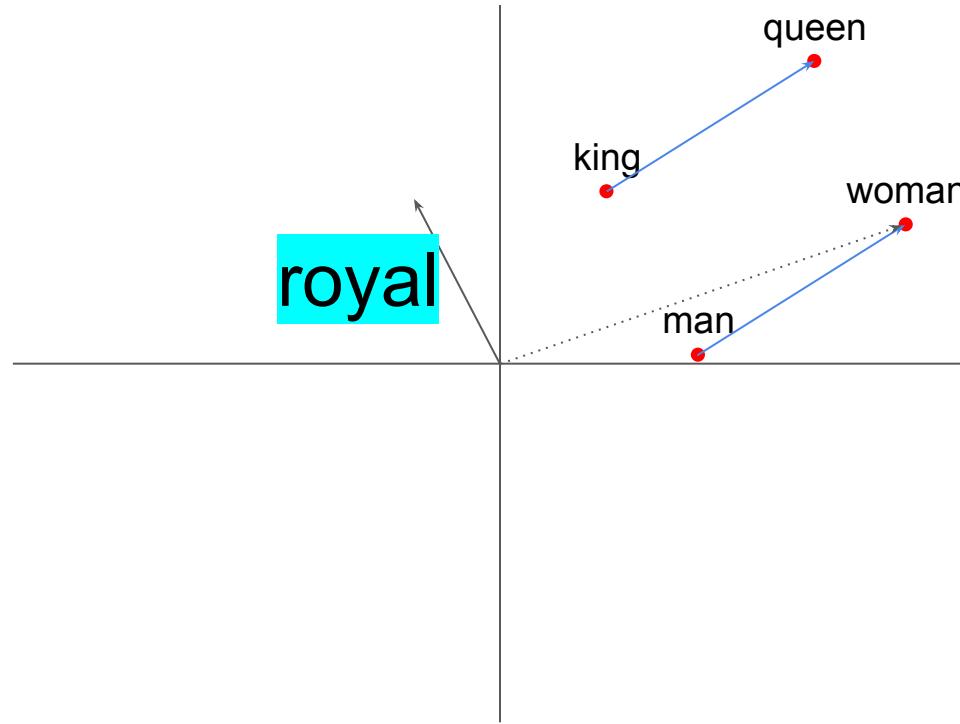


Figure 12.15 The smile vector

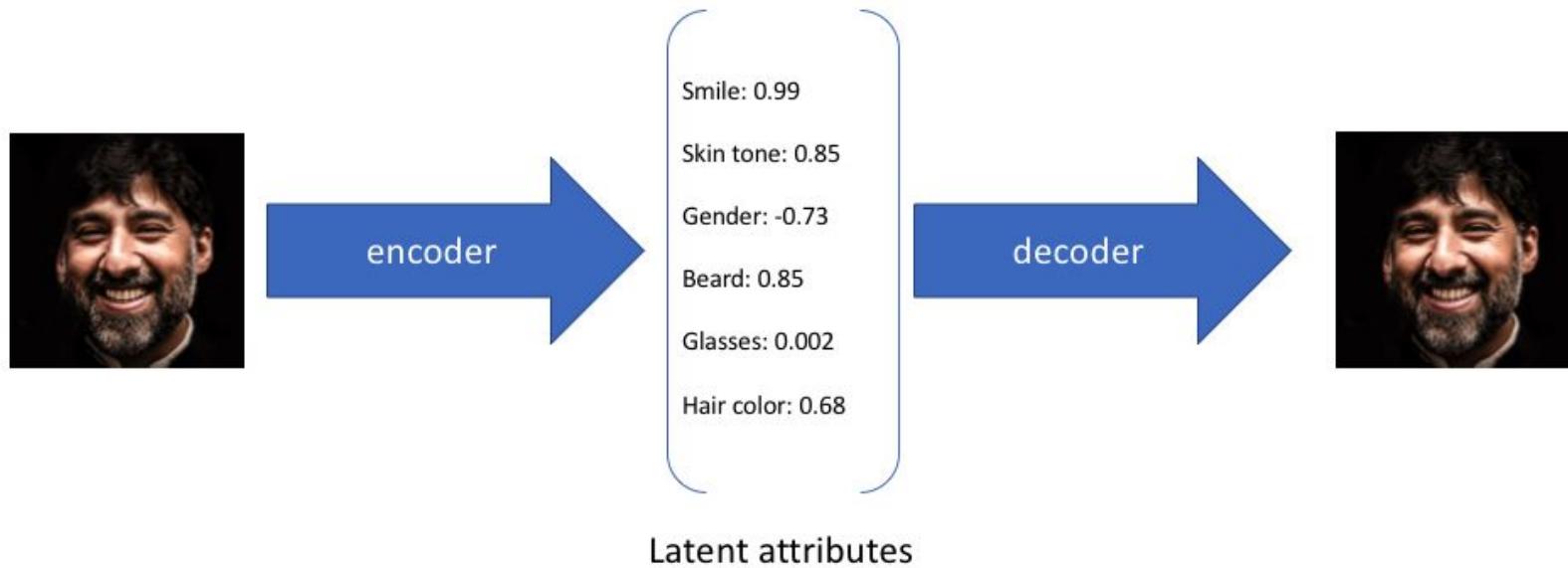
Word Concept Vectors



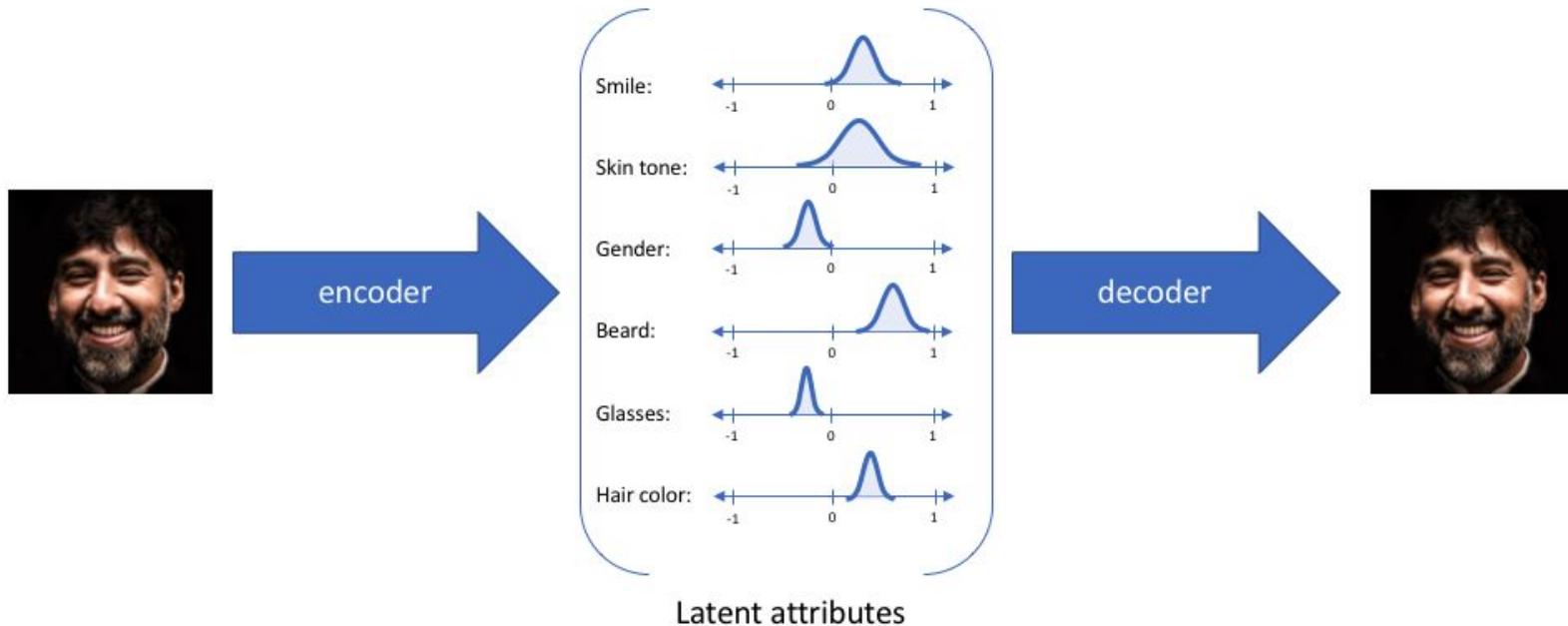
Word Concept Vectors



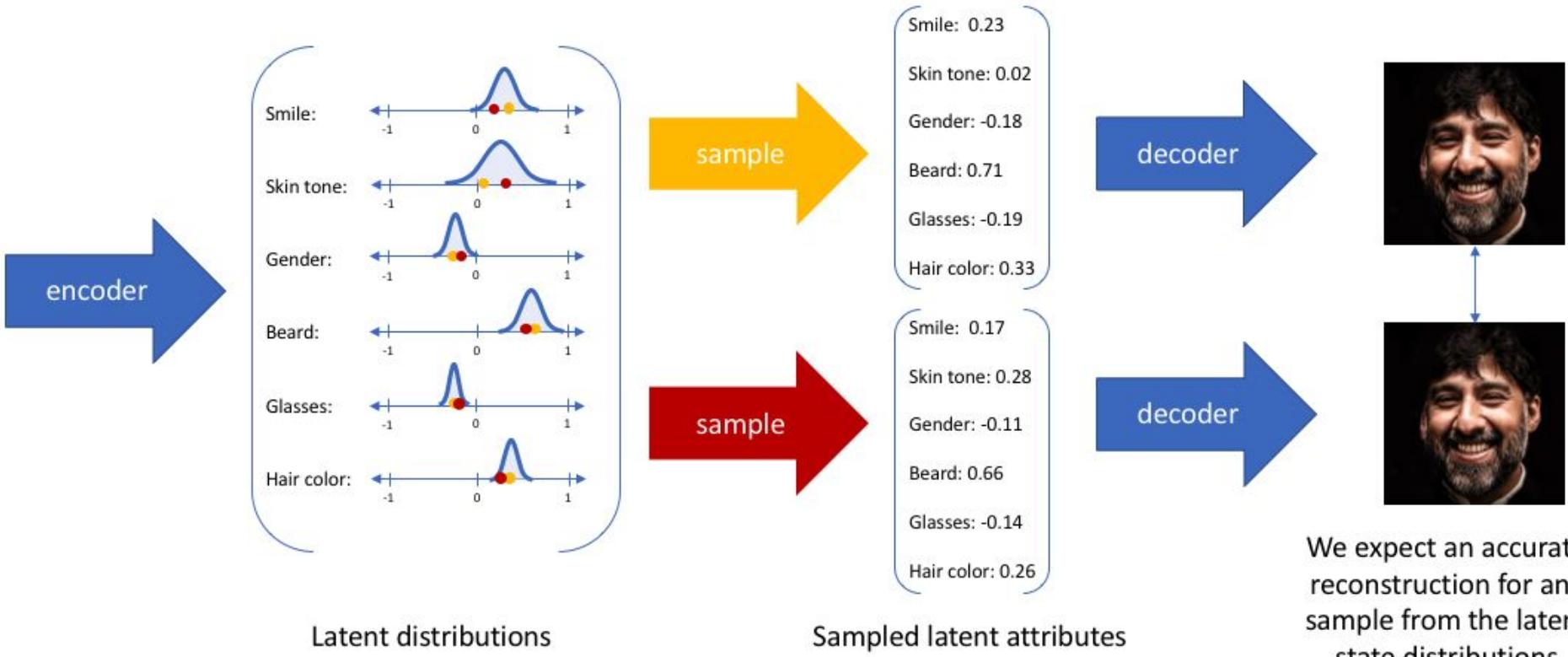
VAE Intuition



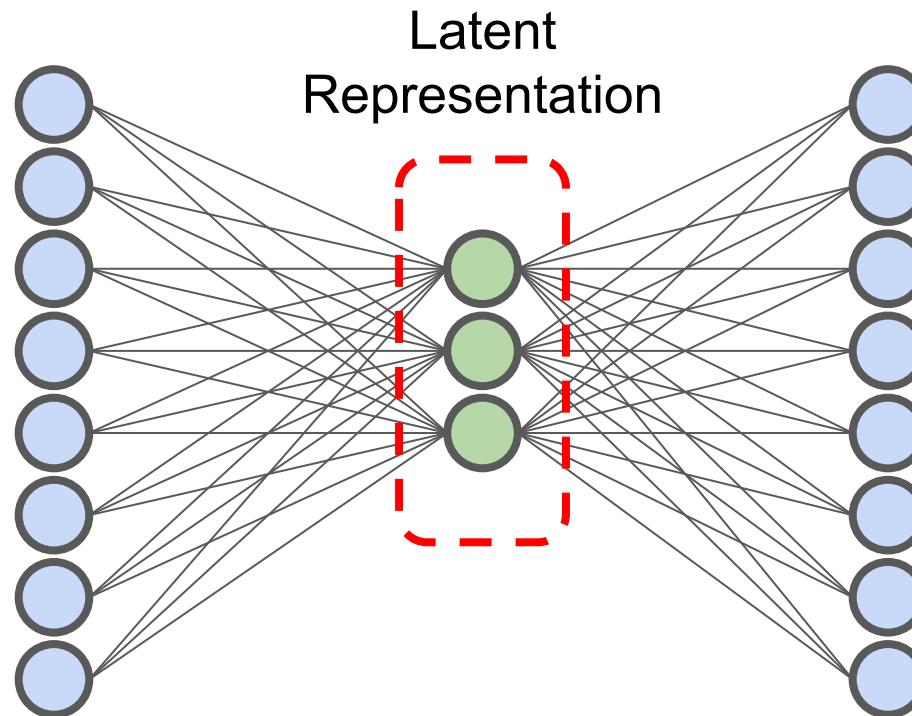
VAE Intuition



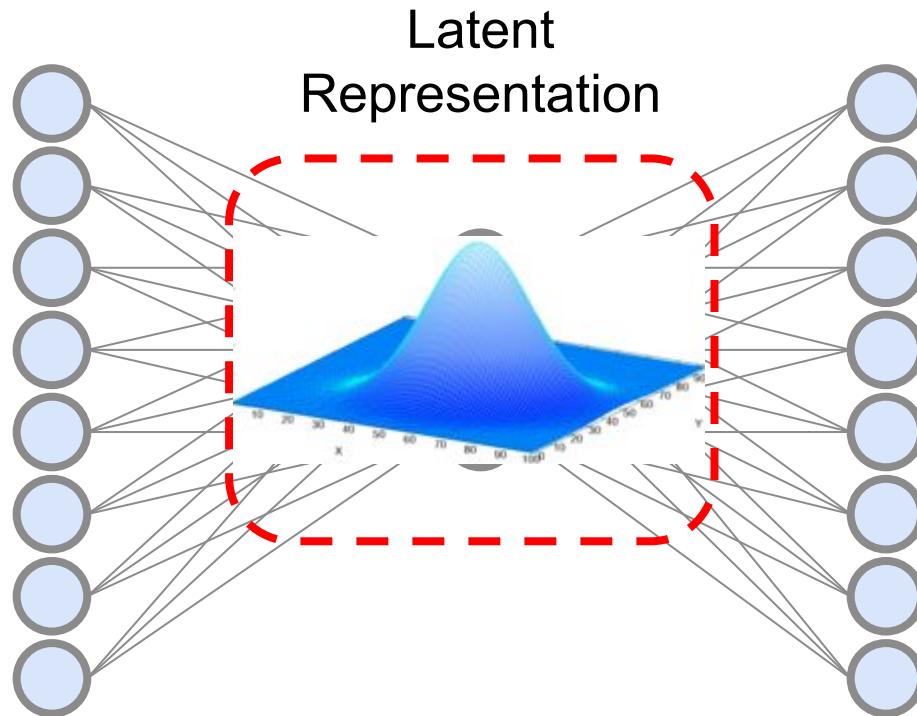
VAE Intuition



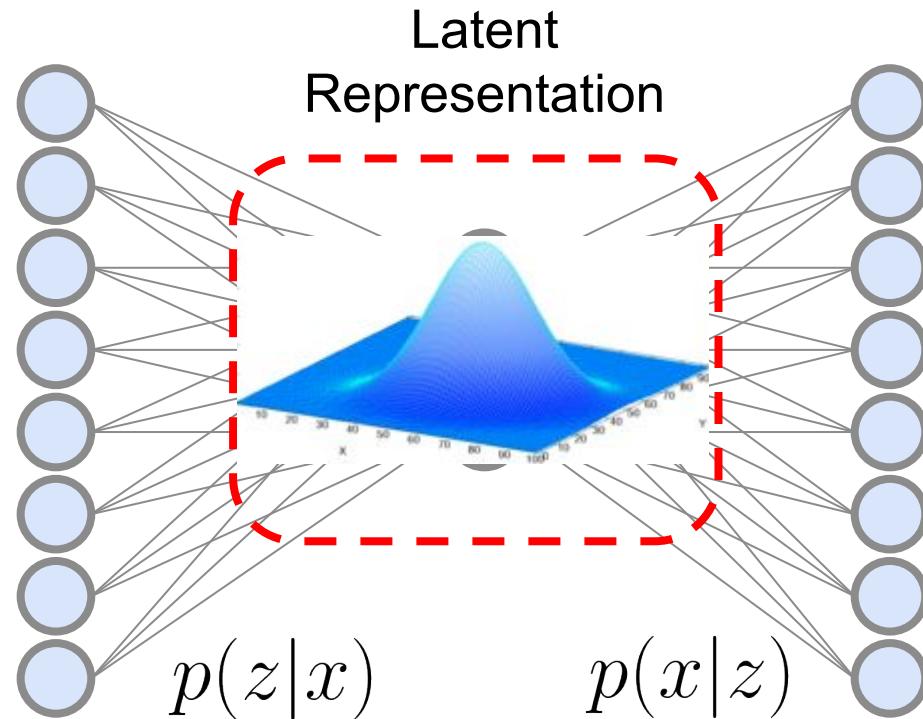
Autoencoder Architecture



Variational Autoencoder Architecture



Variational Autoencoder Architecture



Variational Autoencoder Architecture

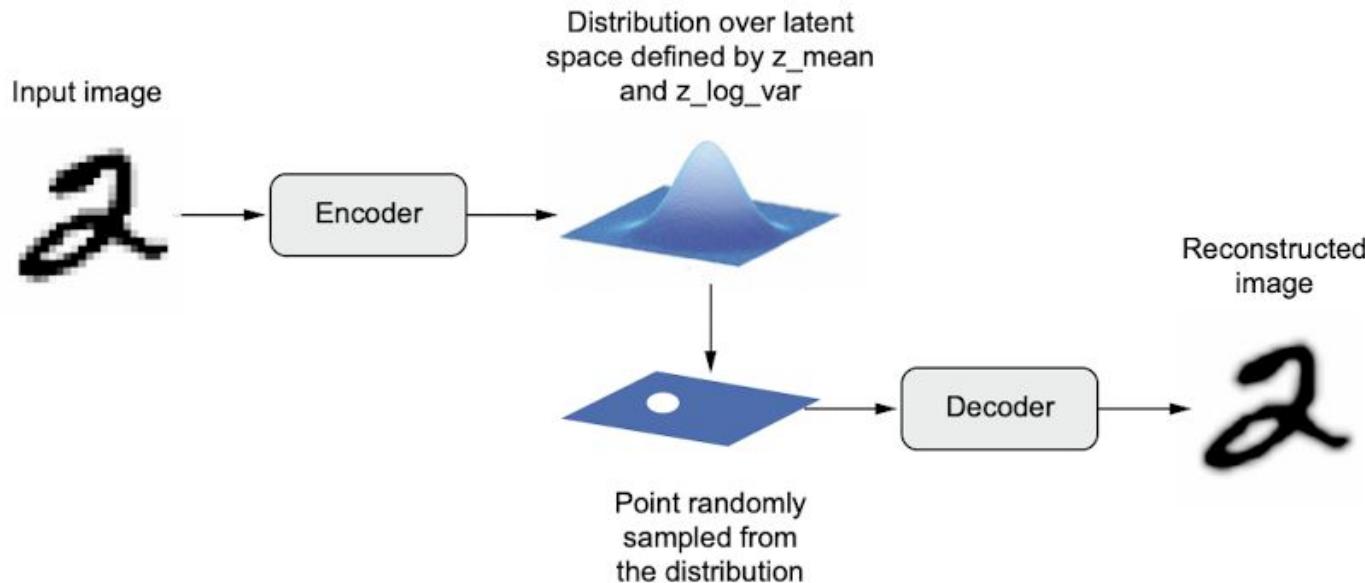
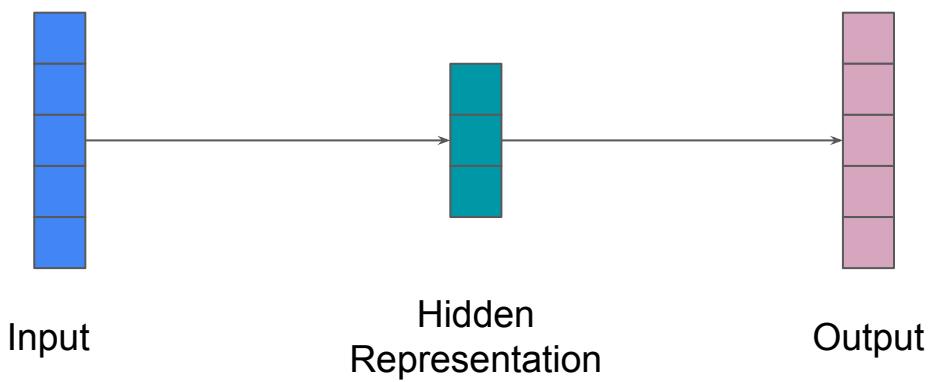
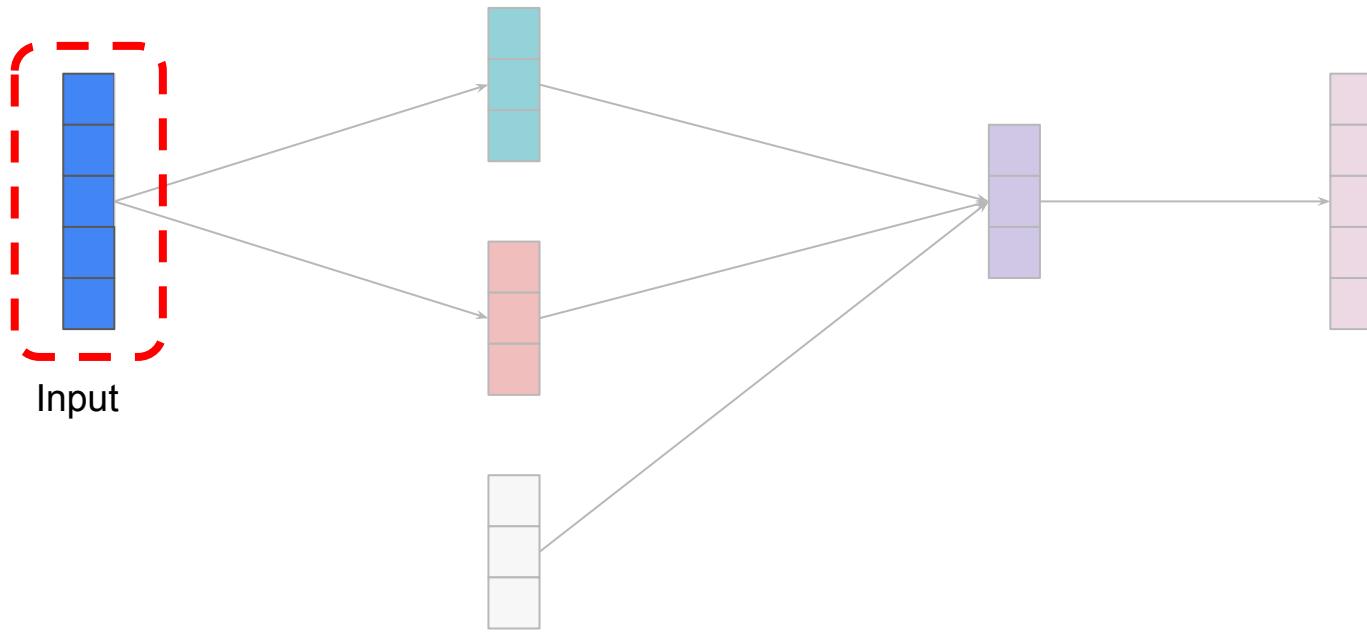


Figure 12.17 A VAE maps an image to two vectors, `z_mean` and `z_log_sigma`, which define a probability distribution over the latent space, used to sample a latent point to decode.

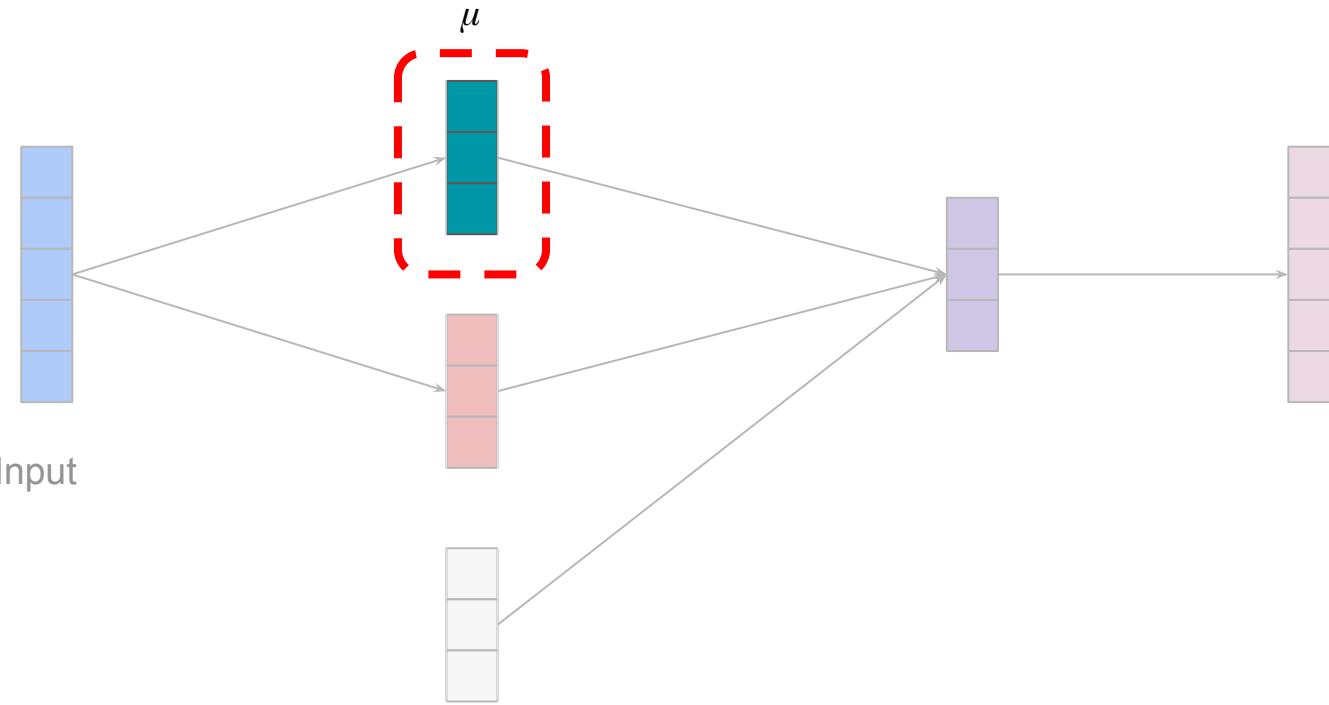
AE



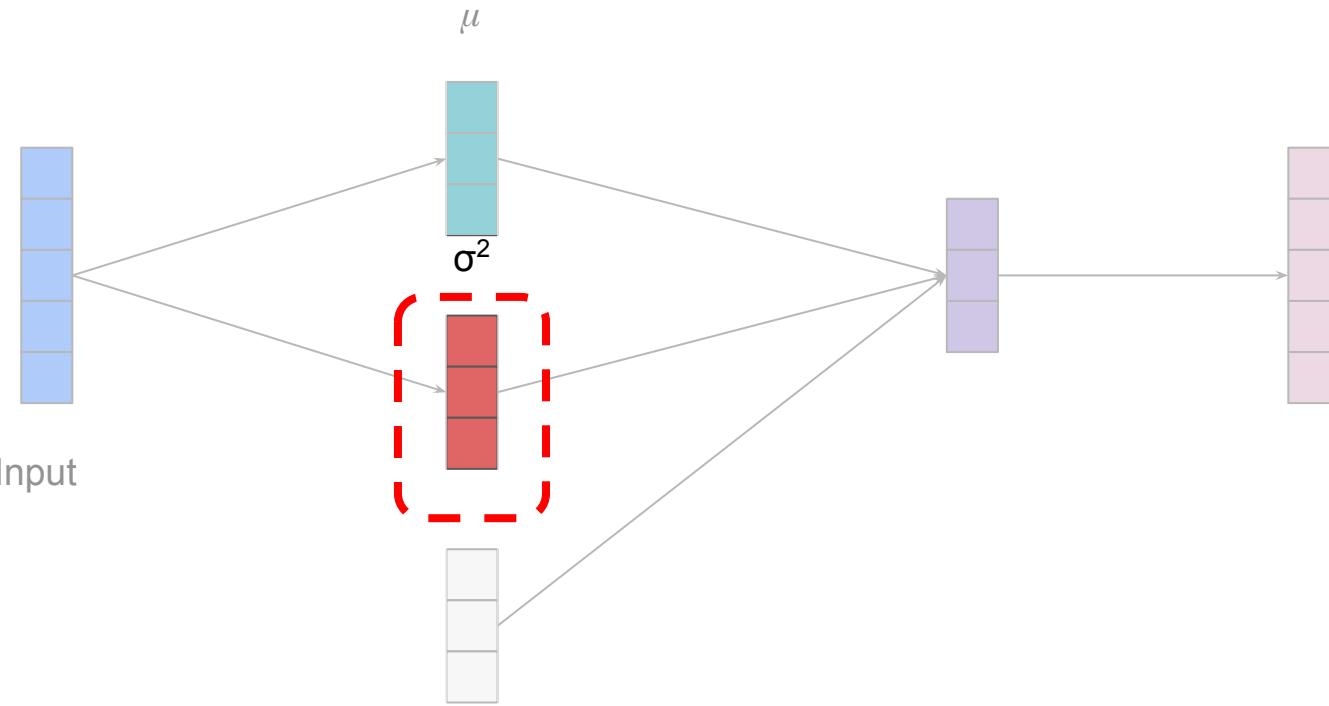
VAE



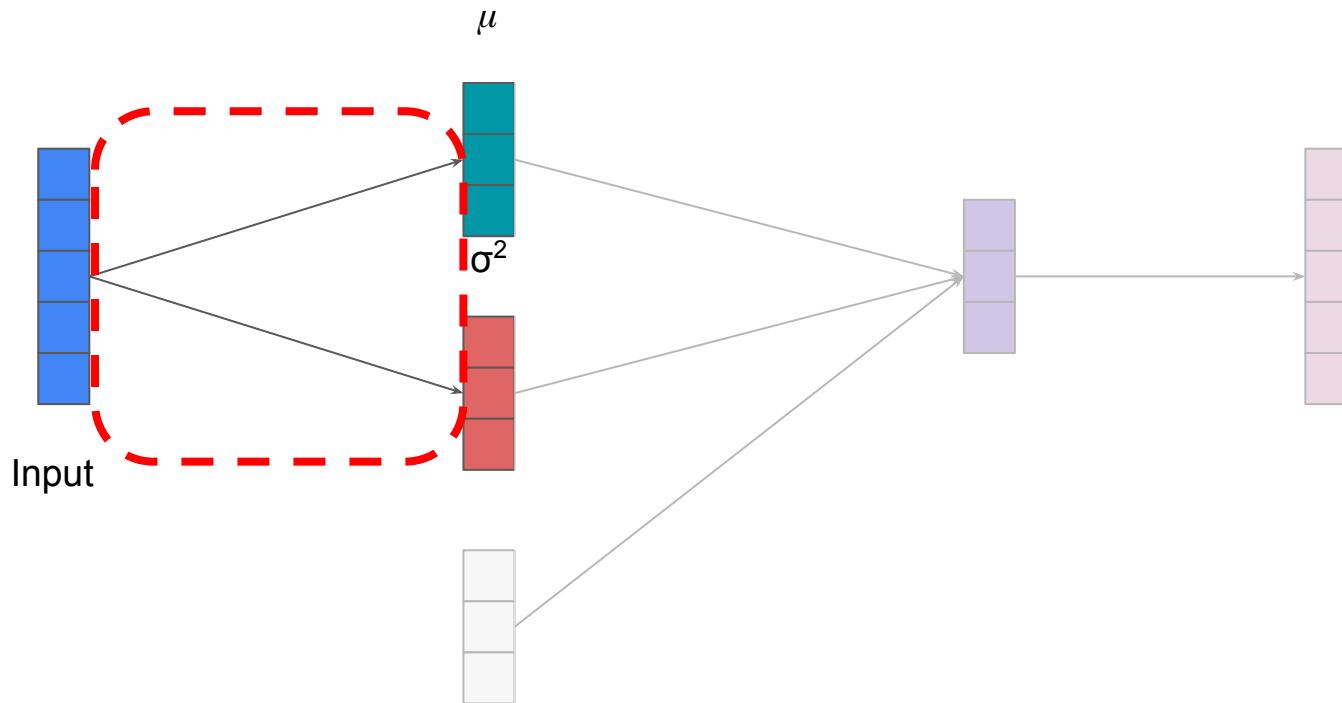
VAE



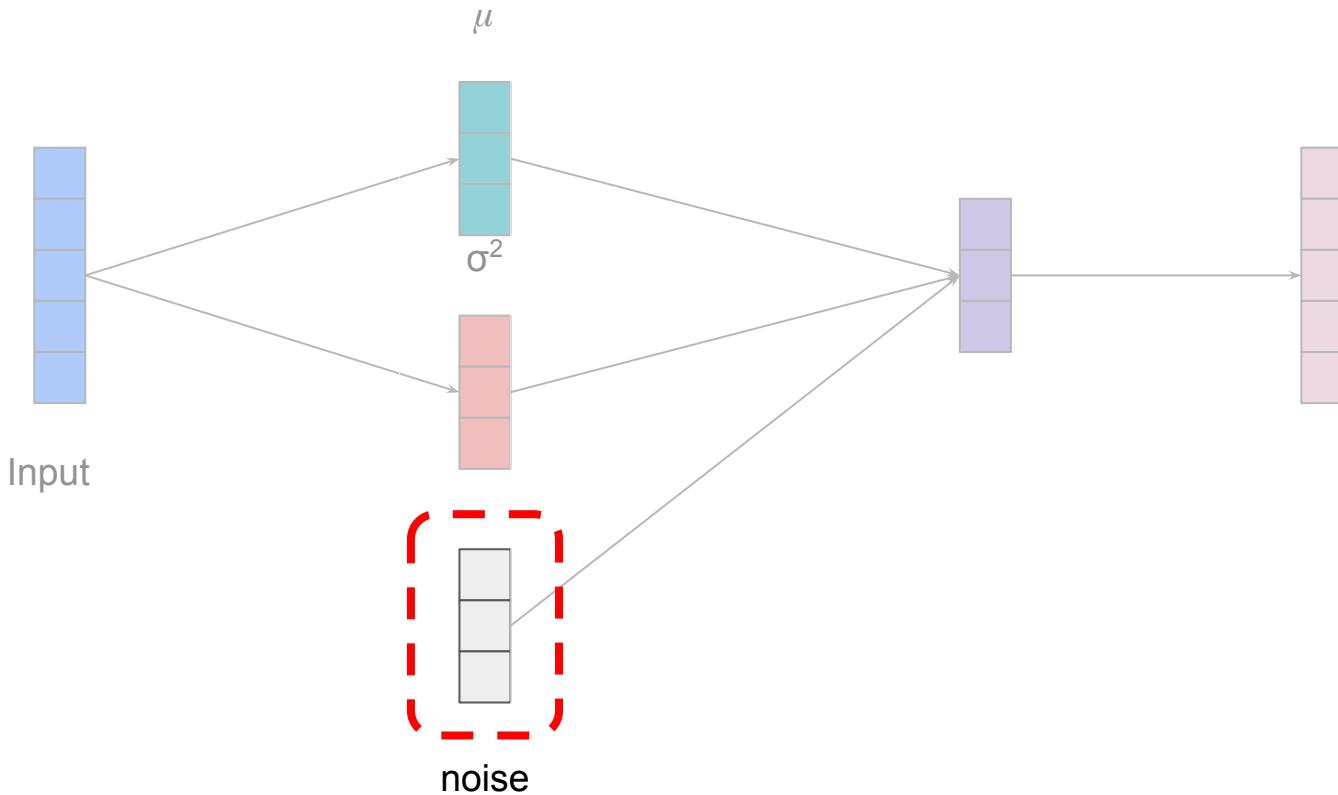
VAE



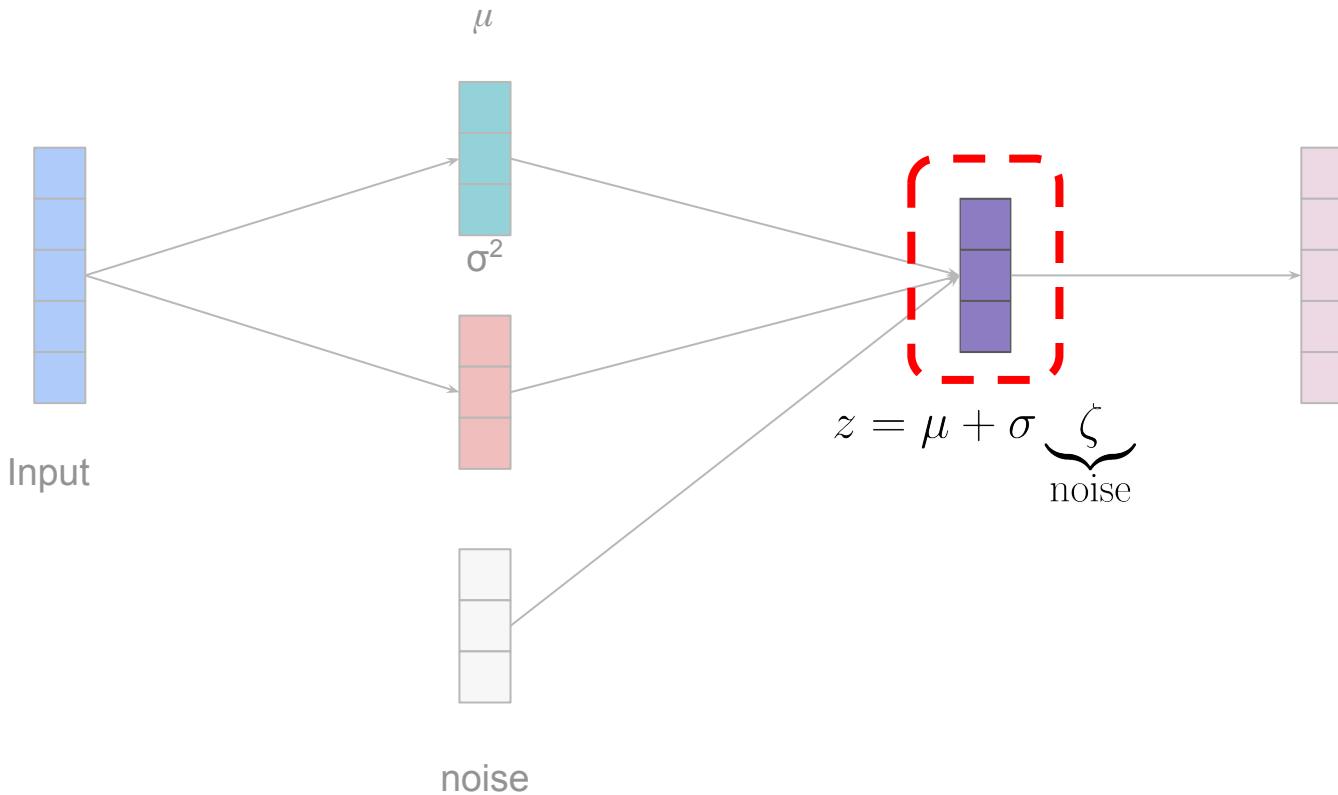
VAE



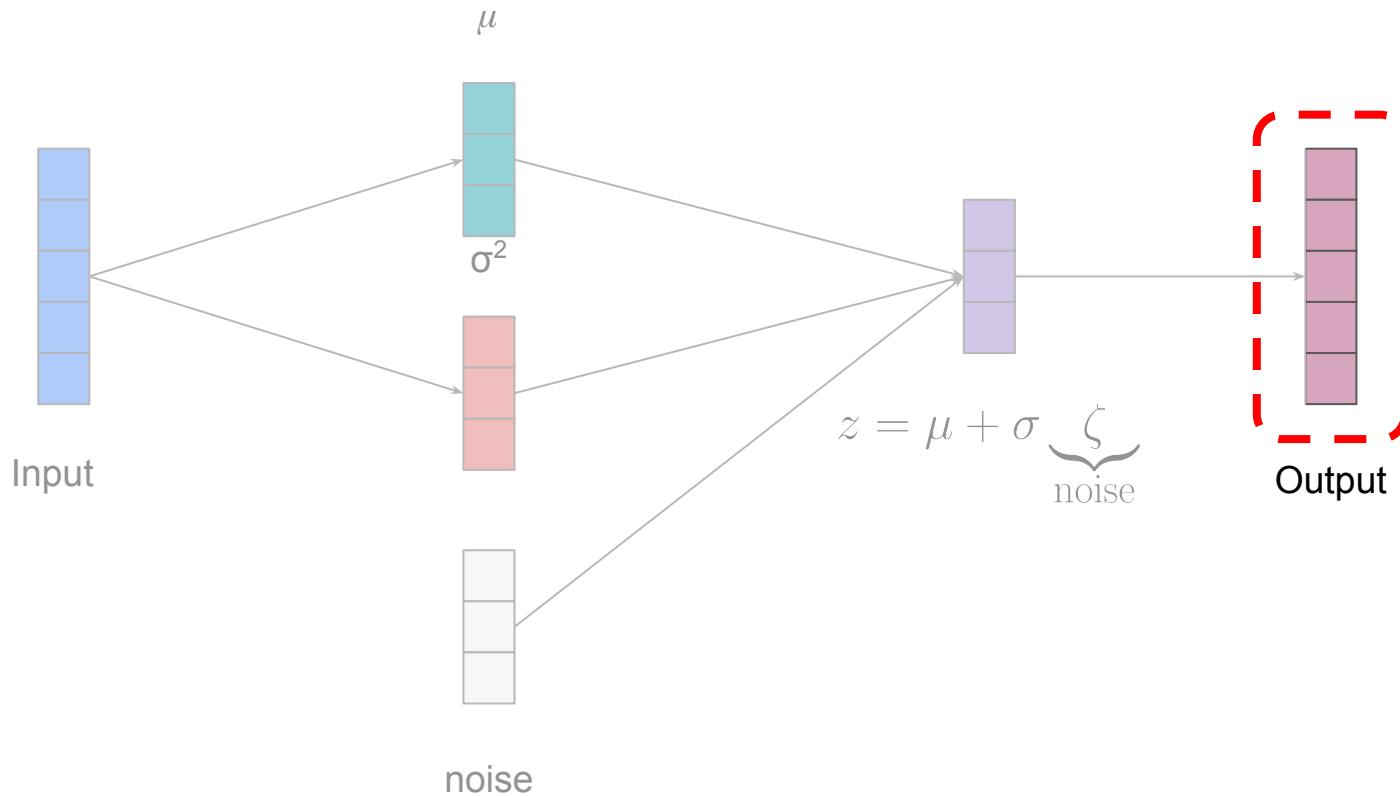
VAE



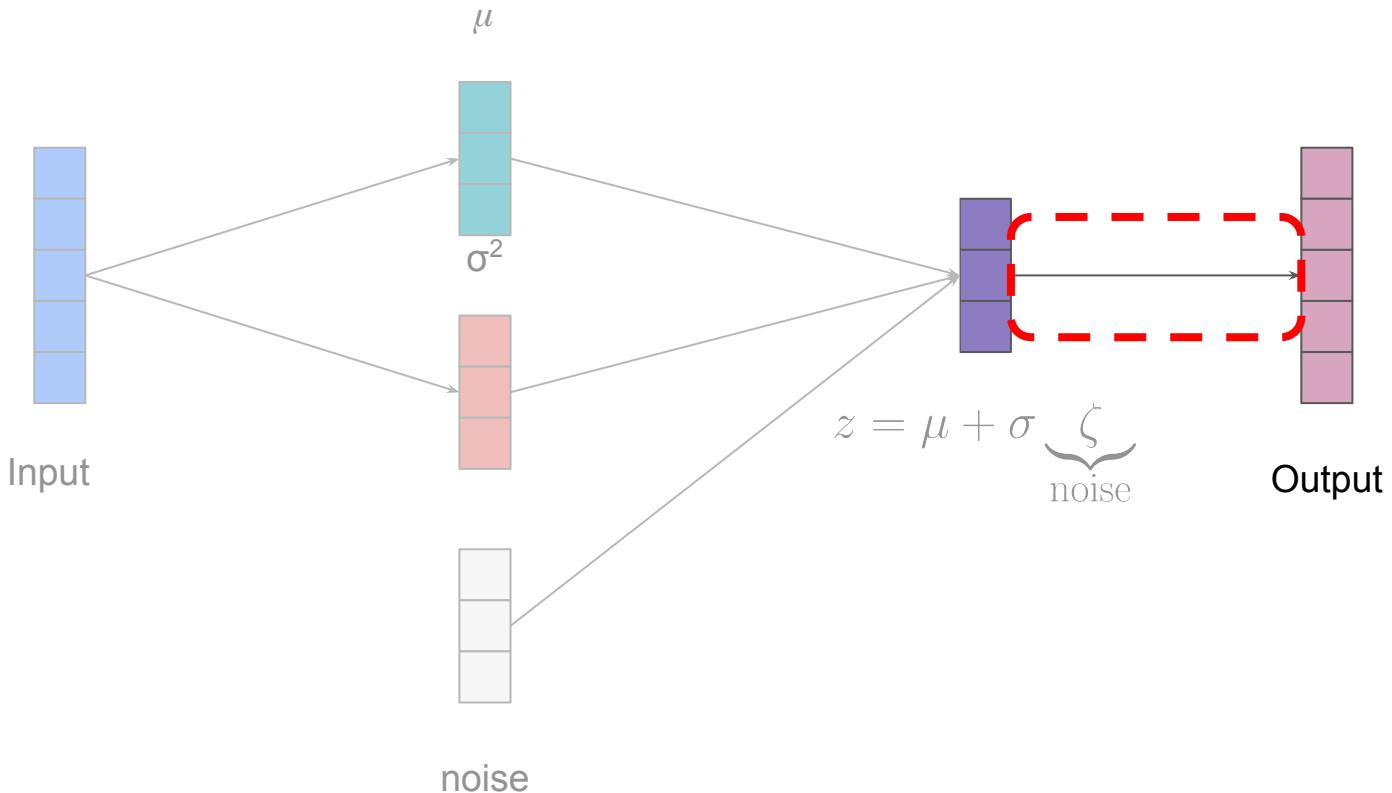
VAE



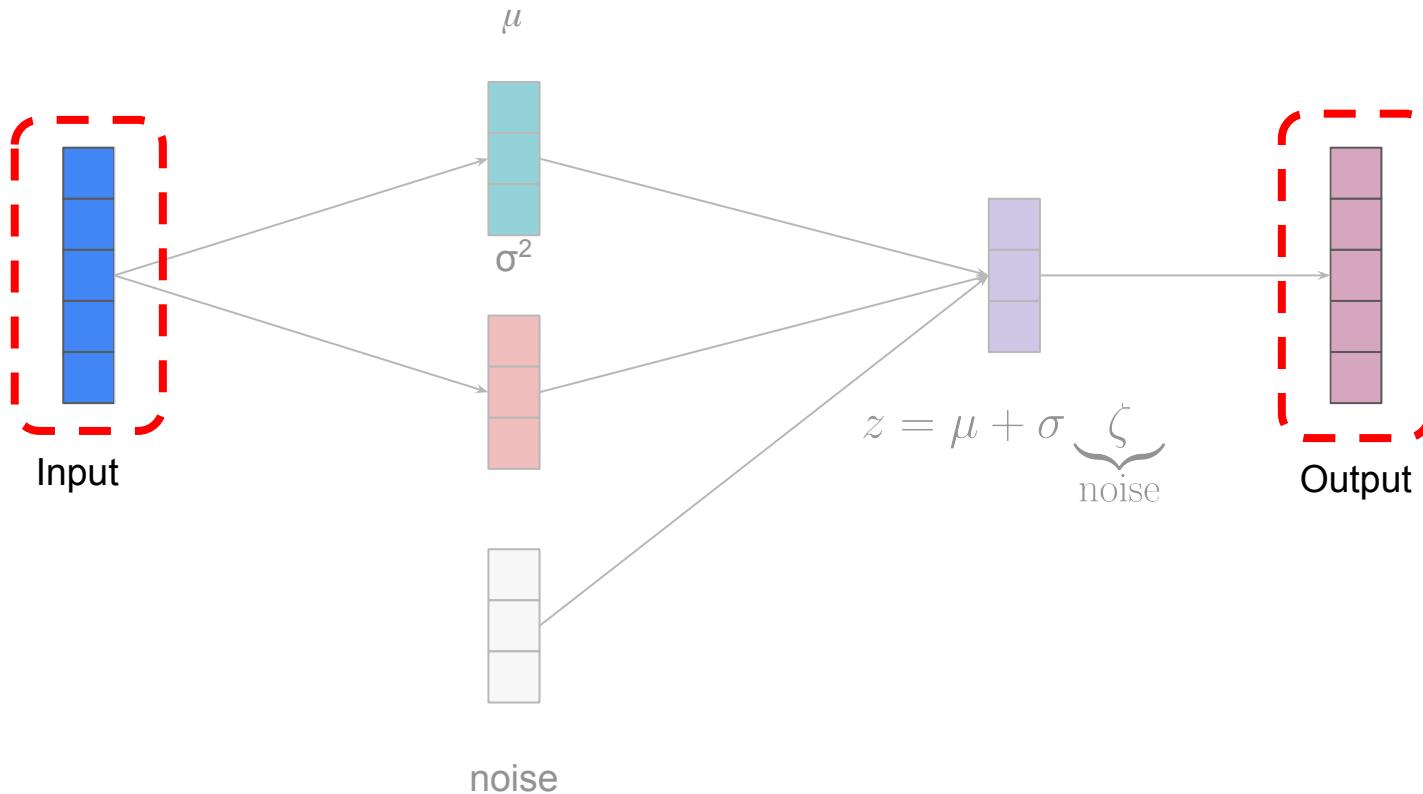
VAE



VAE

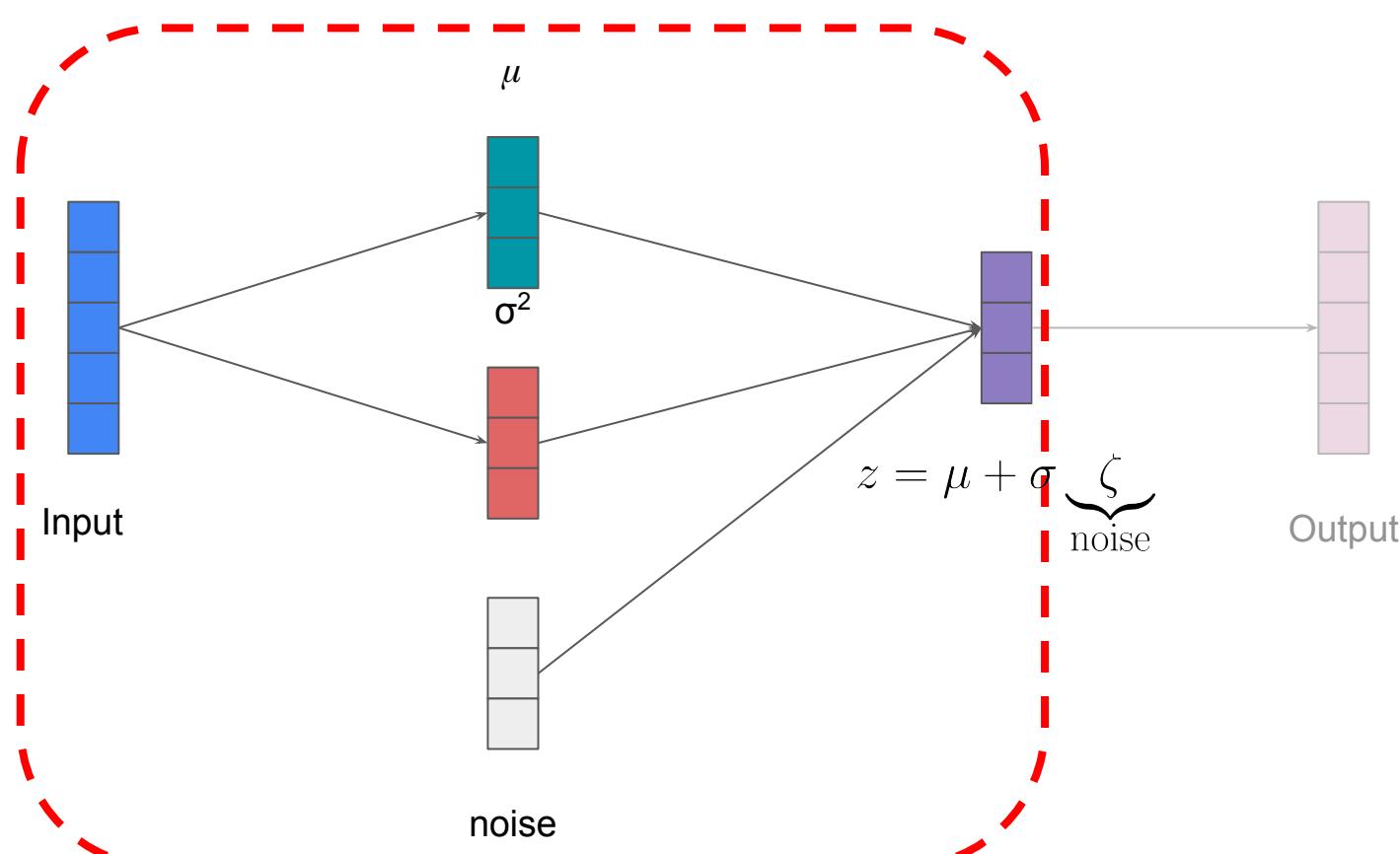


VAE

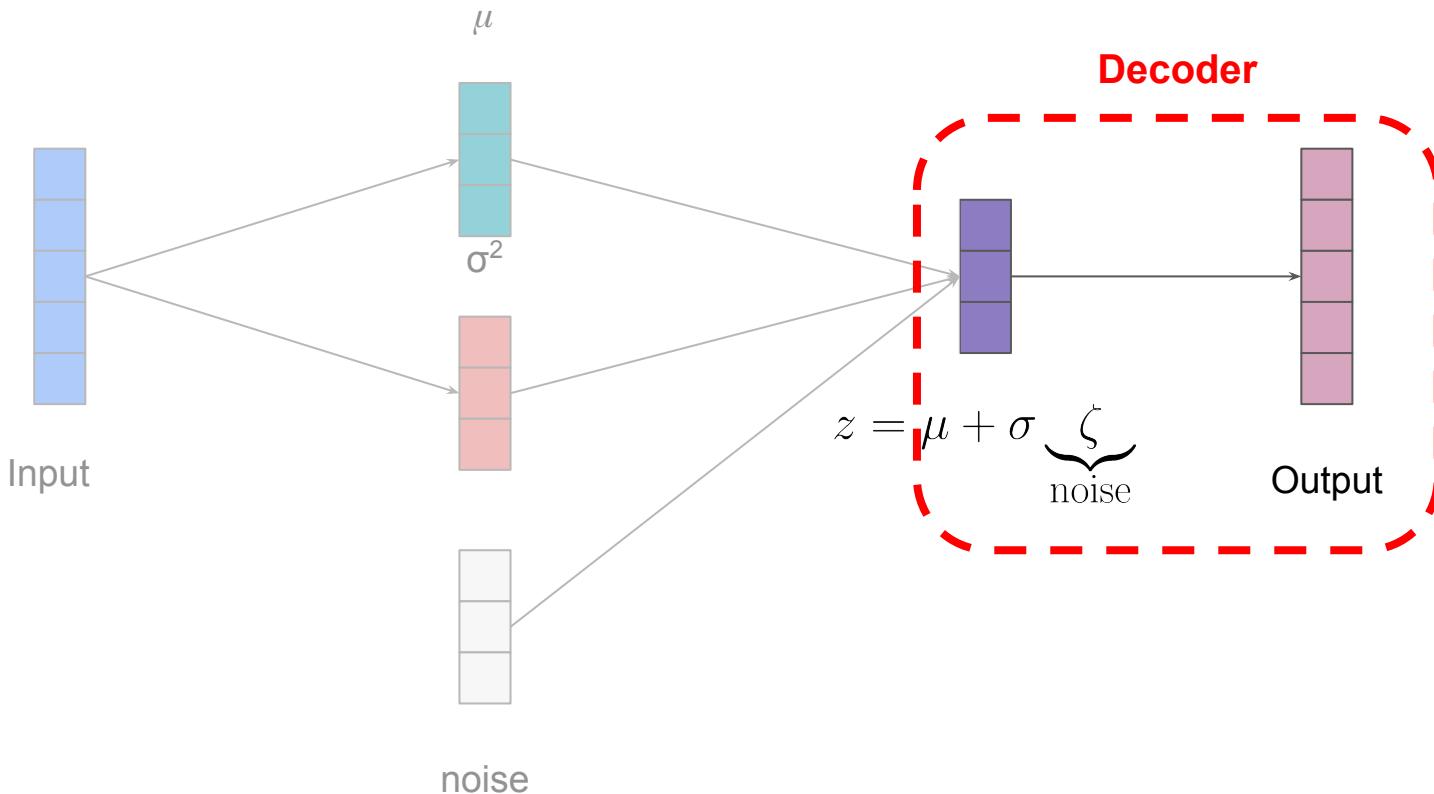


VAE

Encoder



VAE



VAE Pseudocode

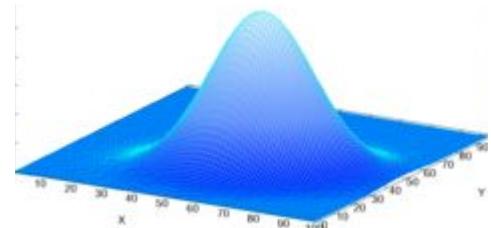
Variational Autoencoder Pseudocode

```
z_mean, z_log_variance = encoder(input_img)  
z = z_mean + exp(z_log_variance) * epsilon  
reconstructed_img = decoder(z)  
model = Model(input_img, reconstructed_img)
```

Variational Autoencoder Pseudocode

```
[z_mean, z_log_variance] = encoder(input_img)  
z = z_mean + exp(z_log_variance) * epsilon  
reconstructed_img = decoder(z)  
model = Model(input_img, reconstructed_img)
```

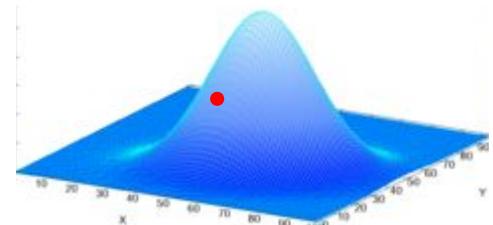
The encoder generates a hidden representation for the input using a multivariate normal *distribution* with two parameters, μ and σ^2



Variational Autoencoder Pseudocode

```
z_mean, z_log_variance = encoder(input_img)  
z = [z_mean + exp(z_log_variance) * epsilon]  
reconstructed_img = decoder(z)  
model = Model(input_img, reconstructed_img)
```

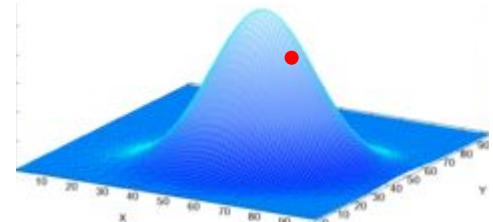
Sample from a normal distribution with mean (z_{mean}) and variance ($\exp(z_{\log \text{variance}})$)



Variational Autoencoder Pseudocode

```
z_mean, z_log_variance = encoder(input_img)  
z = [z_mean + exp(z_log_variance) * epsilon]  
reconstructed_img = decoder(z)  
model = Model(input_img, reconstructed_img)
```

Sample from a normal distribution with mean (z_{mean}) and variance ($\exp(z_{\log \text{variance}})$)



Variational Autoencoder Pseudocode

```
z_mean, z_log_variance = encoder(input_img)  
z = z_mean + exp(z_log_variance) * epsilon  
reconstructed_img = decoder(z)  
model = Model(input_img, reconstructed_img)
```

Decoder takes random sample and tries to re-generate input

Variational Autoencoder Pseudocode

```
z_mean, z_log_variance = encoder(input_img)  
z = z_mean + exp(z_log_variance) * epsilon  
reconstructed_img = decoder(z)  
model = Model(input_img, reconstructed_img)
```

Decoder takes random sample and tries to re-generate input

we expect different samples to create similar images

VAE Loss Function

VAE Loss (Reconstruction)

$$\text{AE loss} = \left\| x - \hat{x} \right\|^2$$

$$\text{VAE loss} = \left\| x - \hat{x} \right\|^2 + KL(N(\mu, \sigma^2), N(0, 1))$$

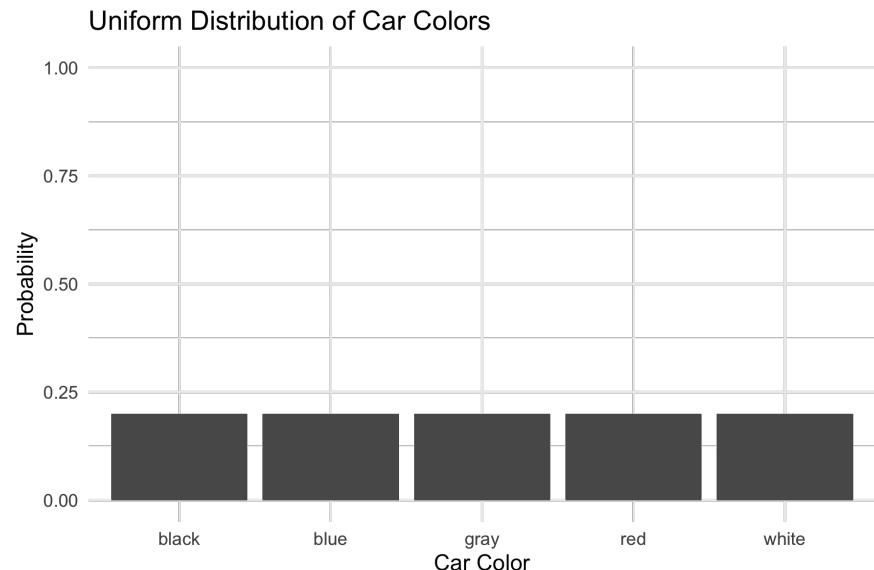
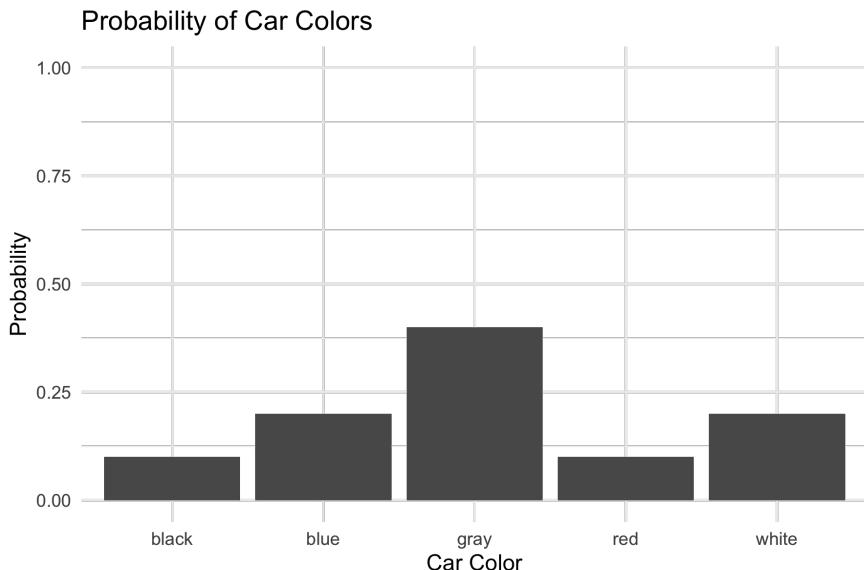
VAE Loss (Regularization)

$$\text{AE loss} = \|x - \hat{x}\|^2$$

$$\text{VAE loss} = \|x - \hat{x}\|^2 + \overbrace{KL(N(\mu, \sigma^2), N(0, 1))}^{\text{Red dashed box}}$$

KL Divergence

$$D_{KL}(p||q) = \sum_{i=1}^N p(x_i) \cdot (\log p(x_i) - \log q(x_i))$$

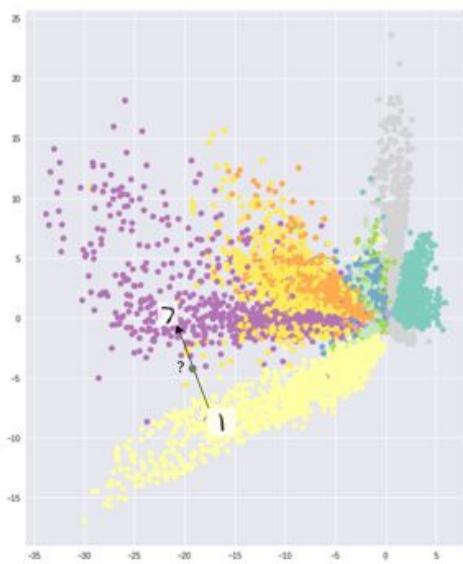


VAE Loss (Regularization)

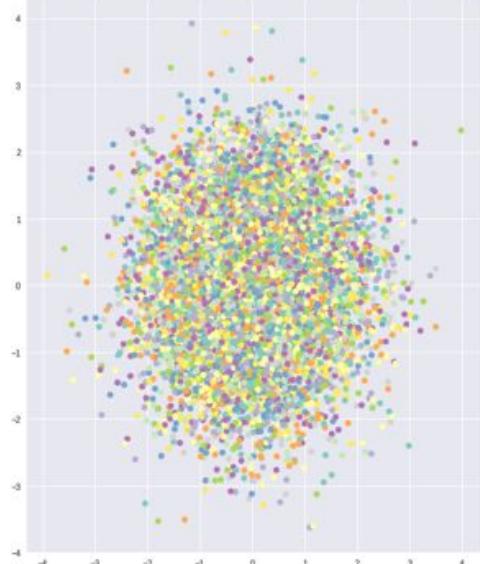
$$\text{AE loss} = \|x - \hat{x}\|^2$$

$$\text{VAE loss} = \|x - \hat{x}\|^2 + \overbrace{KL(N(\mu, \sigma^2), N(0, 1))}^{\text{Red dashed box}}$$

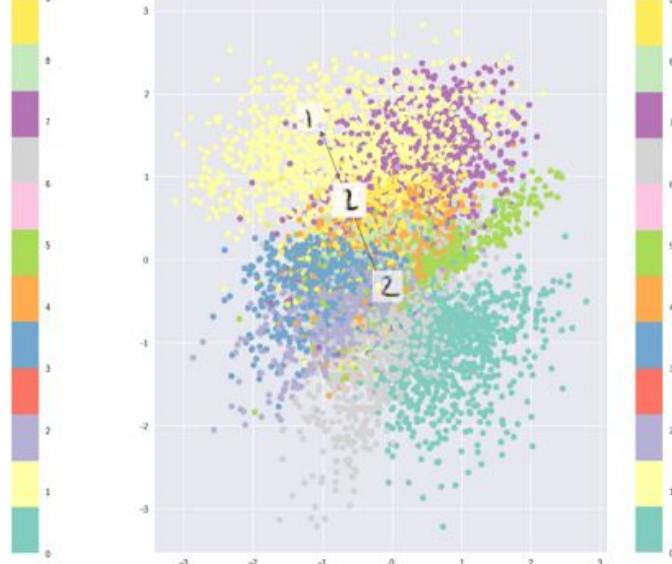
Only reconstruction loss

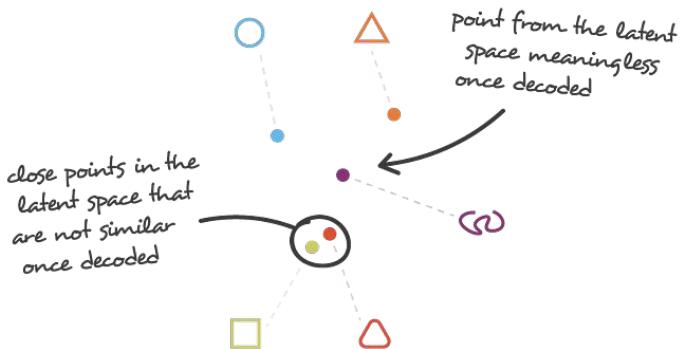


Only KL divergence

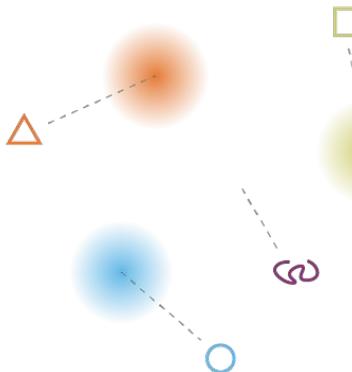


Combination

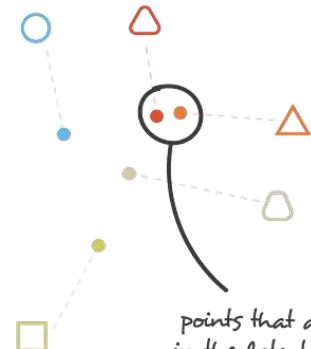




irregular latent space



what can happen without regularisation



regular latent space



what we want to obtain with regularisation

VAE Output

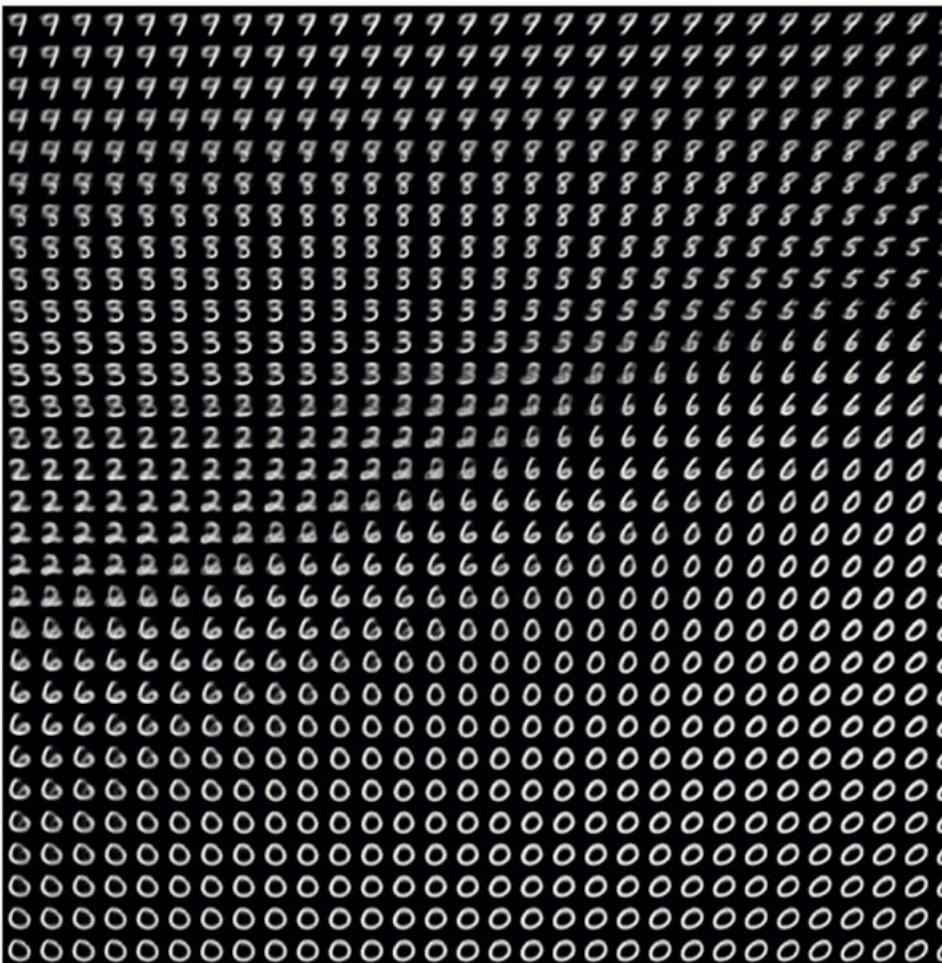


Image from: Deep Learning with Python 2nd Edition (Chollet)

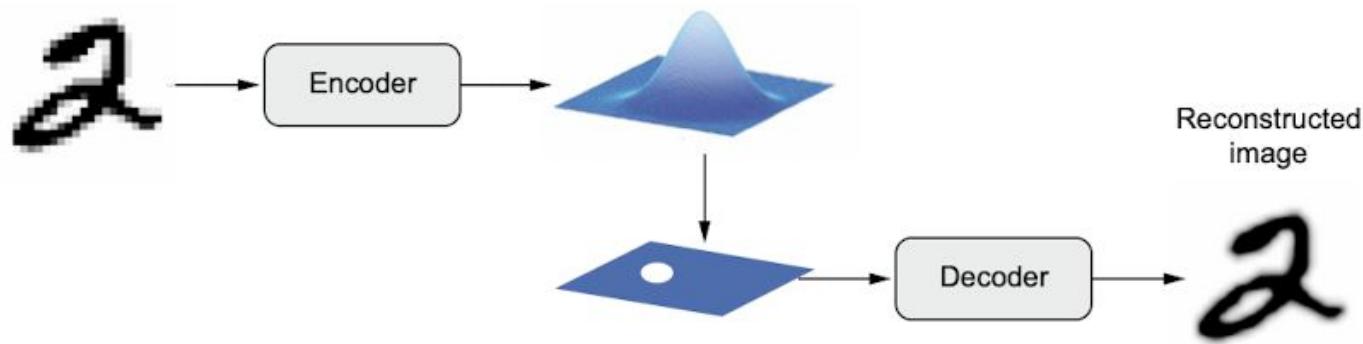
Figure 12.18 Grid of digits decoded from the latent space

Ideal Latent Spaces

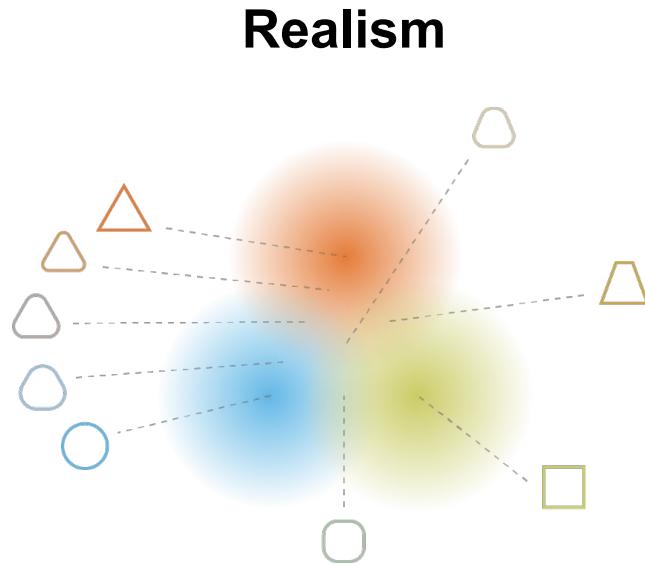
- Representation
- Realism
- Smoothness

Ideal Latent Spaces

Representation



Ideal Latent Spaces

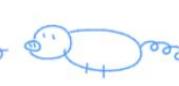


Ideal Latent Spaces

Smoothness



Human Input



Human Input

NOTE

- Convolutional Neural Networks for learning features preferred **max pooling** to strides
- However, max pooling does not preserve location
- In C-VAEs **strides** are used to preserve information location

Strides

Where are things in an image

Max Pooling

What's in an image

Music VAE

Interpolation: a mathematical technique to estimate the values of unknown data points that fall in between existing, known data points.

