

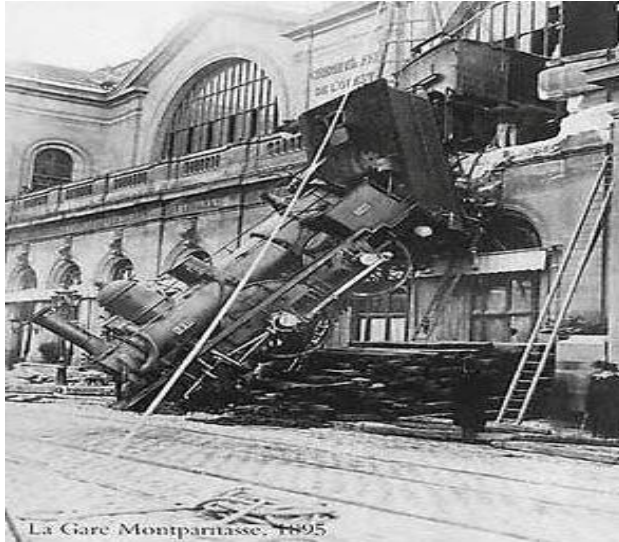
Ralf Möller, Sylvia Melzer

Generative Vision Models, Mental Imagery

We've seen how to compute representations of words and sentences. What about images?

<http://people.cs.umass.edu/~miyyer/cs685/>

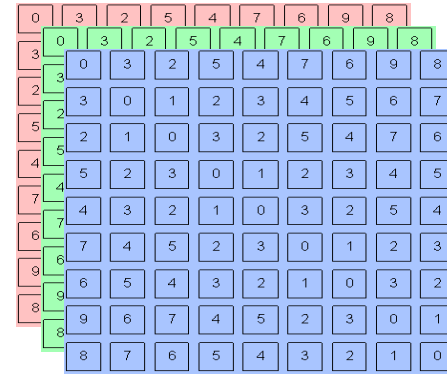
Grayscale images are matrices



0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

<http://people.cs.umass.edu/~miyyer/cs685/>

Color images are tensors



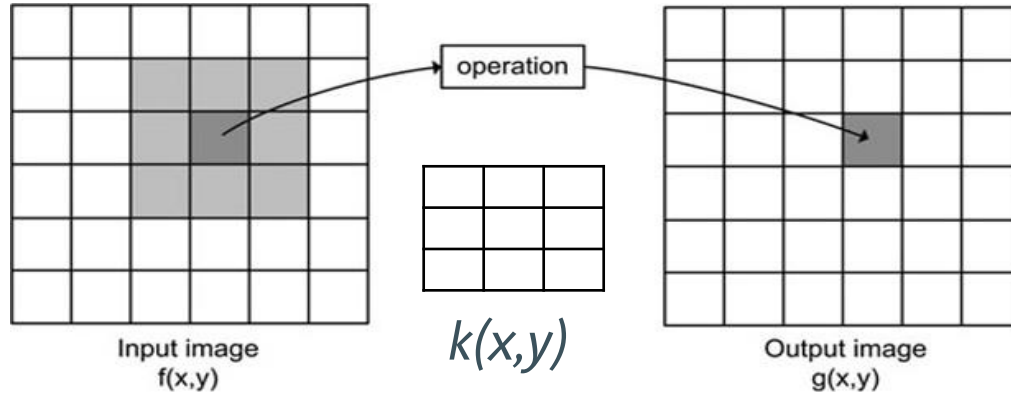
Channel x height x width x

Channels are usually RGB: Red, Green, and Blue

Other color spaces: HSV, HSL, LUV, XYZ, Lab, CMYK, etc

<http://people.cs.umass.edu/~miyyer/cs685/>

Convolution operator



$$g(x,y) = \sum_v \sum_u k(u,v)f(x-u, y-u)$$

<http://people.cs.umass.edu/~miyyer/cs685/>; Image Credits: <http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/>

Image Filtering

Input image

*

(Filter, Kernel)

Weights



Output image

4	5	7	6	6
3	2	8	0	7
6	7	7	1	5
3	0	1	1	1
4	3	2	1	7

*

0	0	0
1	0	1
0	0	0



	11	2	15	
	13	8	12	
	4			

<http://people.cs.umass.edu/~miyyer/cs685/>

Demo:
<http://setosa.io/ev/image-kernels/>

<http://people.cs.umass.edu/~miyyer/cs685/>

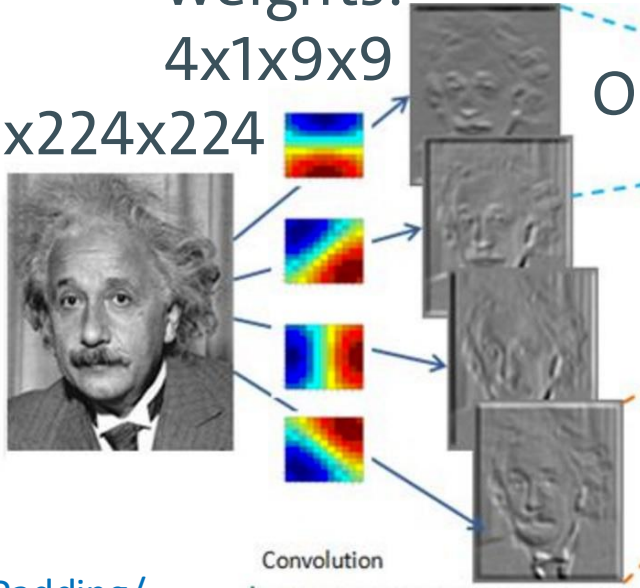
Convolutional Layer (with 4 filters)

weights:
4x1x9x9

Input: 1x224x224

Output: 4x224x224
if stride = 1

In summary, to convolve a $n \times n \times c$ with a $f \times f \times c$ filter with the stride s and padding p , the generated output size should be $(\frac{n+2p-f}{s} + 1) \times (\frac{n+2p-f}{s} + 1) \times 1$. However, in some cases, the value of $\frac{n+2p-f}{s}$ sometimes is not integer, thus we will take the floor value, which is $\lfloor \frac{n+2p-f}{s} \rfloor$.



Example: <https://guandi1995.github.io/Padding/>

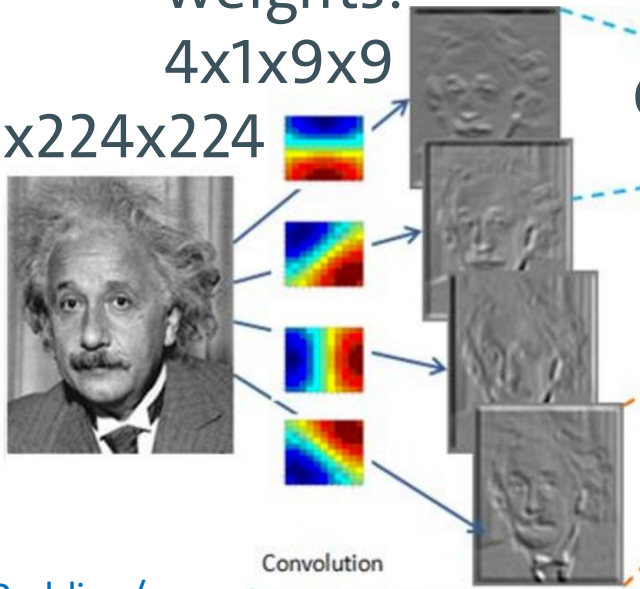
Convolutional Layer (with 4 filters)

weights:
4x1x9x9

Input: 1x224x224

Output: 4x112x112
if stride = 2

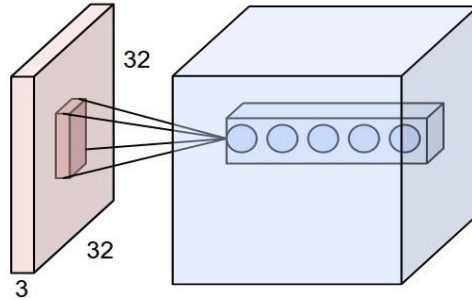
In summary, to convolve a $n \times n \times c$ with a $f \times f \times c$ filter with the stride s and padding p , the generated output size should be $(\frac{n+2p-f}{s} + 1) \times (\frac{n+2p-f}{s} + 1) \times 1$. However, in some cases, the value of $\frac{n+2p-f}{s}$ sometimes is not integer, thus we will take the floor value, which is $\lfloor \frac{n+2p-f}{s} \rfloor$.



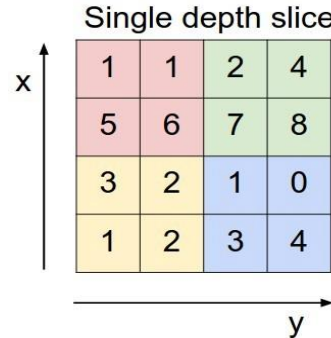
Example: <https://guandi1995.github.io/Padding/>

Pooling layers to reduce dimensionality

Convolutional Layers: slide a set of small filters over the image



Pooling Layers: reduce dimensionality of representation



max pool with 2x2 filters and stride 2

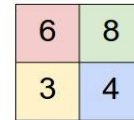


image: <https://cs231n.github.io/convolutional-networks/>

Alexnet

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

The paper that started the
deep learning revolution!

Image classification

Classify an image into 1000 possible classes:

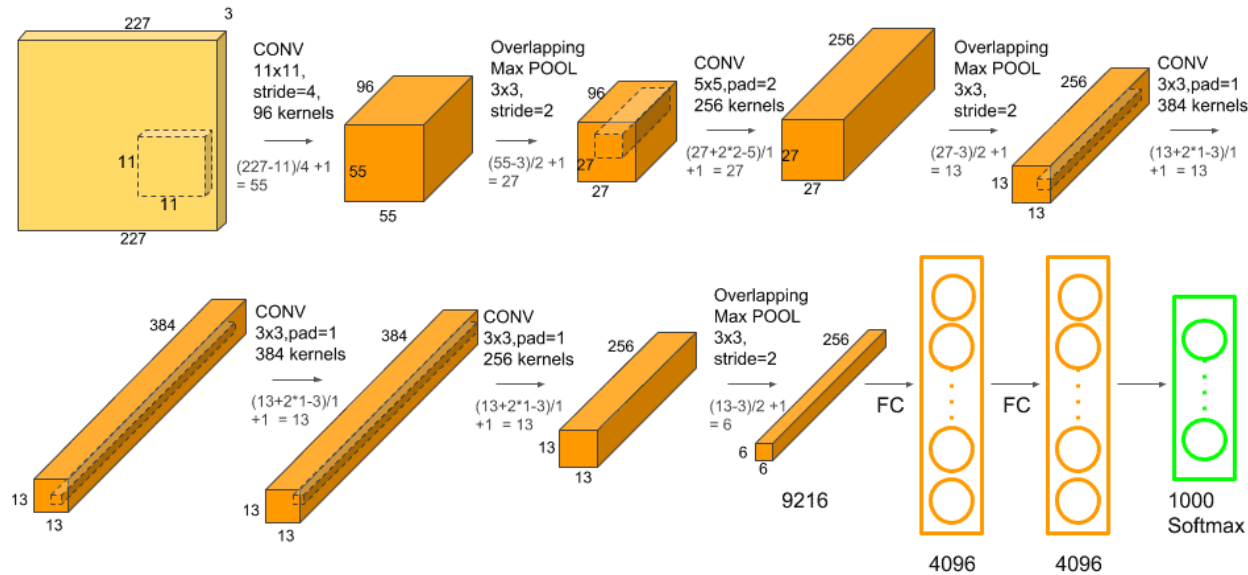
e.g. Abyssinian cat, Bulldog, French Terrier, Cormorant, Chickadee,
Red fox, banjo, barbell, hourglass, knot, maze, viaduct, etc.



cat, tabby cat (0.71)
Egyptian cat (0.22)
red fox (0.11)

Train on ImageNet challenge
dataset, ~1.2 million images

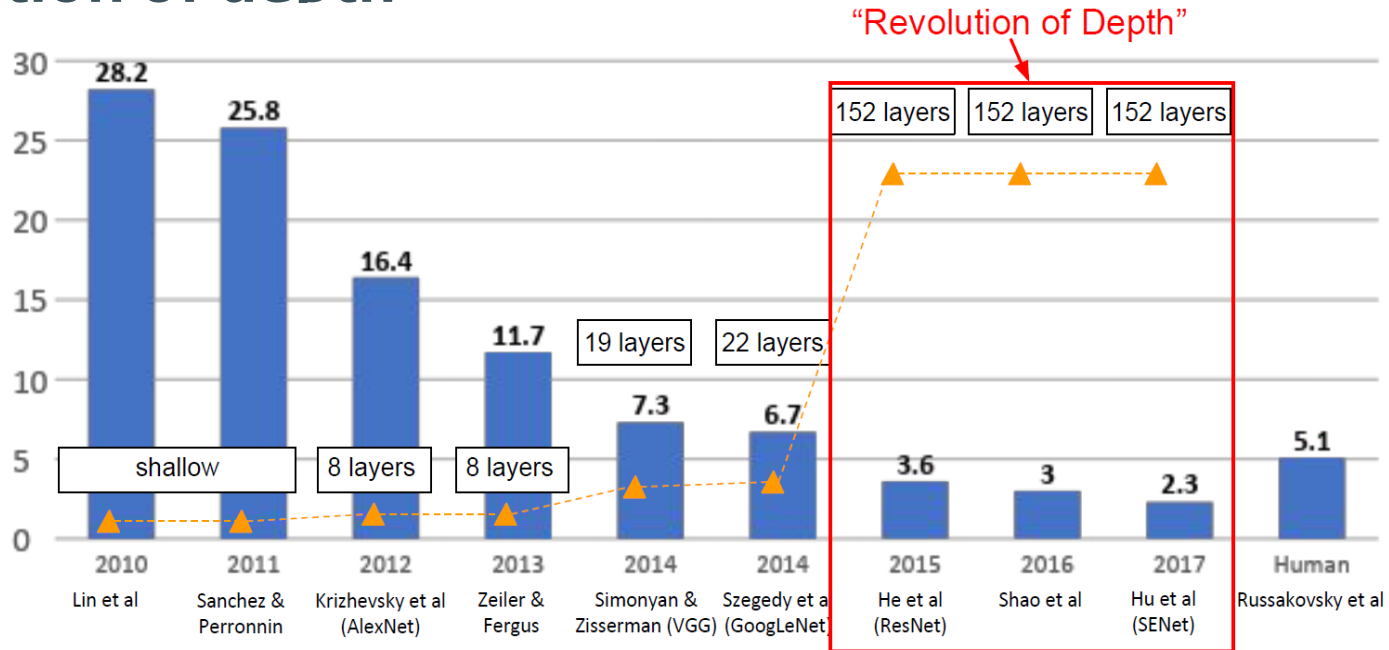
Alexnet



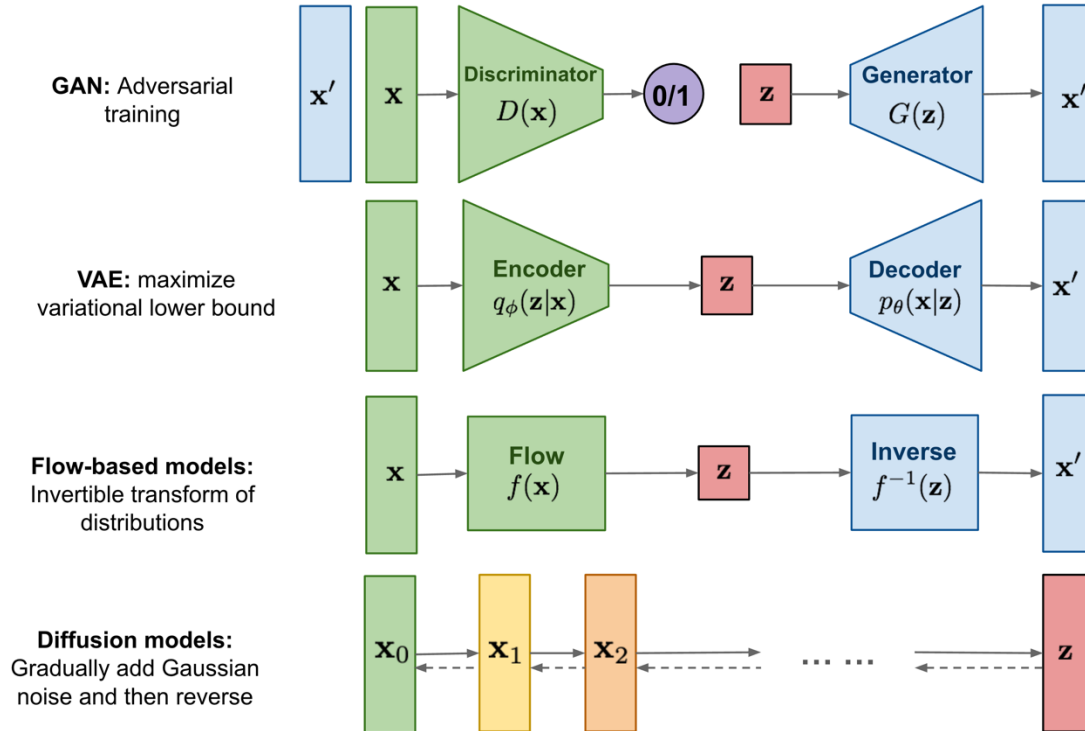
- Initially vectors of $227 \times 227 \times 3 = 154\,587$ features).
- Represented as a vector of 4096 features

- The two fully connected and softmax layers are similar to a multi layer perceptron and could actually be replaced by other kinds of classifiers such as Random Forests or SVMs. However, they are really important for the training phase of the neural net.

Revolution of depth



Generative Models



Evolution of Generative Vision Models

VAEs, 2013



GANs, 2014



PixelCNN, 2016



BigGAN, 2019



Imagen, 2022

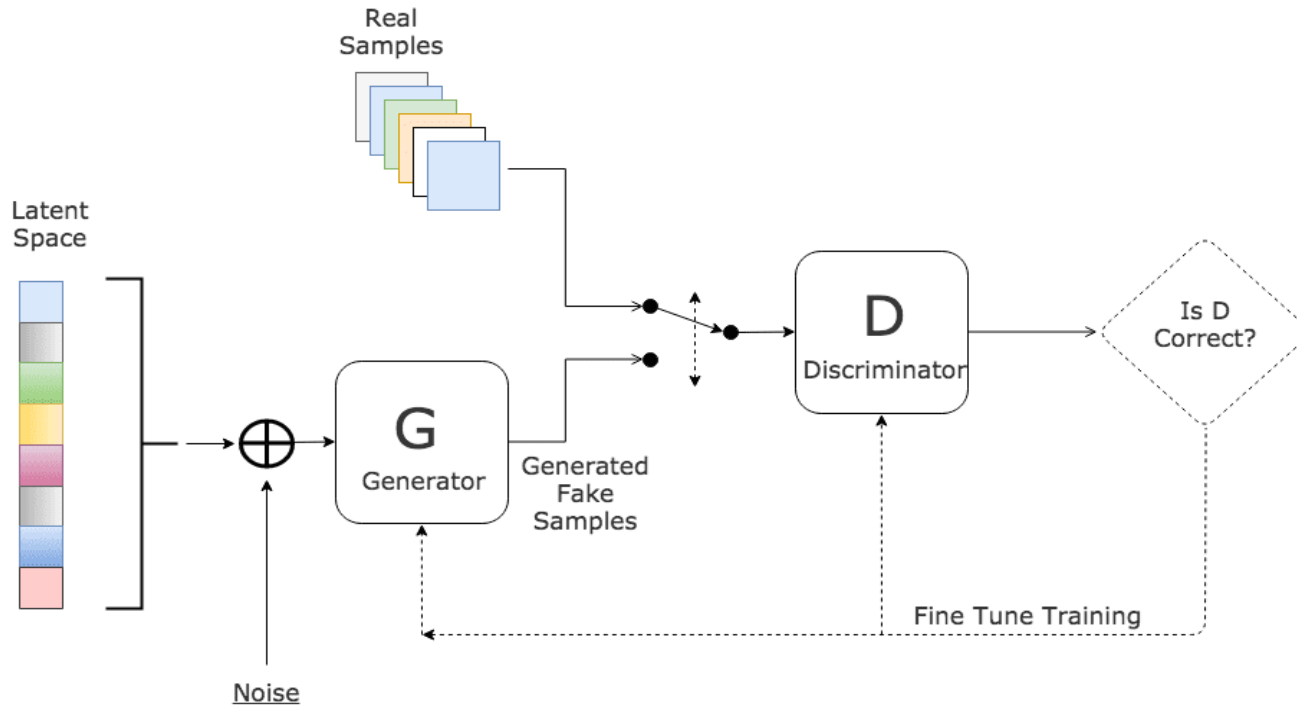


SORA 2024



<https://deeplearning.cs.cmu.edu/F24/document/slides/lec24.diffusion.pdf>

Generative Adversarial Network (GAN)



<https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html>

Metrics for Quantifying the Similarity between two probability distributions

Kullback–Leibler Divergence

$$D_{KL}(p||q) = \int_x p(x) \log \frac{p(x)}{q(x)} dx$$

Jensen–Shannon Divergence

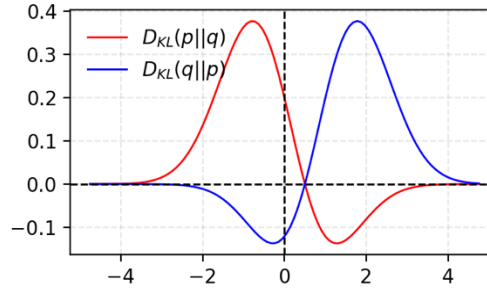
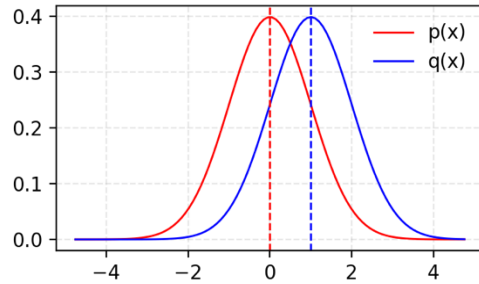
$$D_{JS}(p||q) = \frac{1}{2} D_{KL}(p||\frac{p+q}{2}) + \frac{1}{2} D_{KL}(q||\frac{p+q}{2})$$

<https://lilianweng.github.io/posts/2017-08-20-gan/>

The distributions are similar, but slightly shifted.

KL measures: How bad is q , if the true distribution is p ?

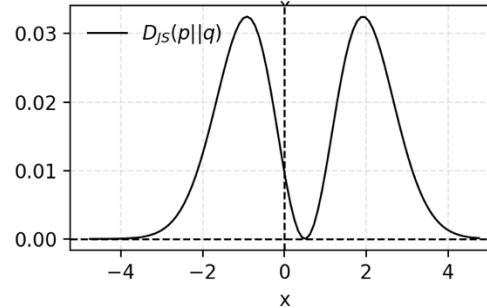
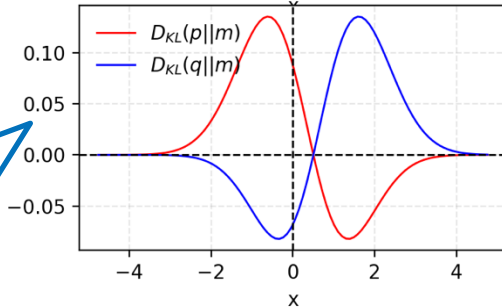
red: $p(x)$
 blue: $q(x)$



red: $D_{KL}(p||q)$
 blue: $D_{KL}(q||p)$

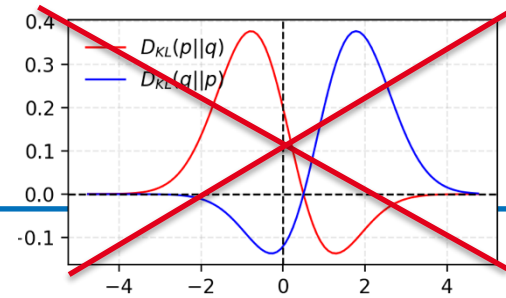
$D_{KL}(p||q) \neq D_{KL}(q||p)$

$m = \frac{p+q}{2}$
 red: $D_{KL}(p||m)$
 blue: $D_{KL}(q||m)$

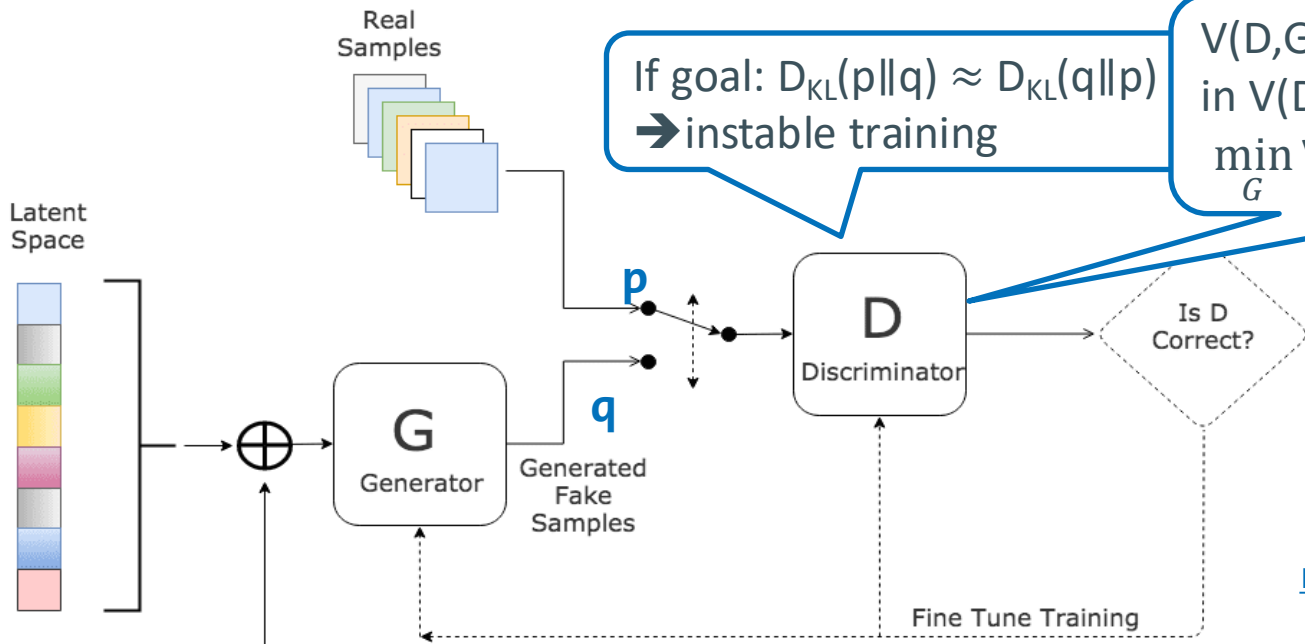


black: $D_{JS}(p||q)$

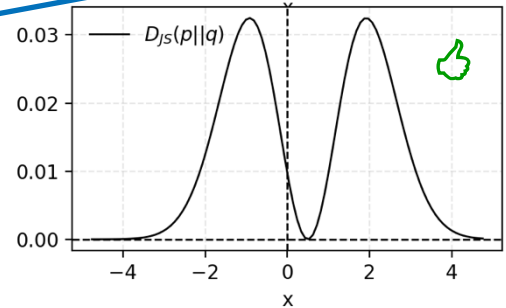
<https://lilianweng.github.io/posts/2017-08-20-gan/>



Generative Adversarial Network (GAN)



$V(D,G)$ value function; set D^* in $V(D,G)$:
 $\min_G V(D^*,G) \Leftrightarrow \min_G D_{JS}(p||q)$

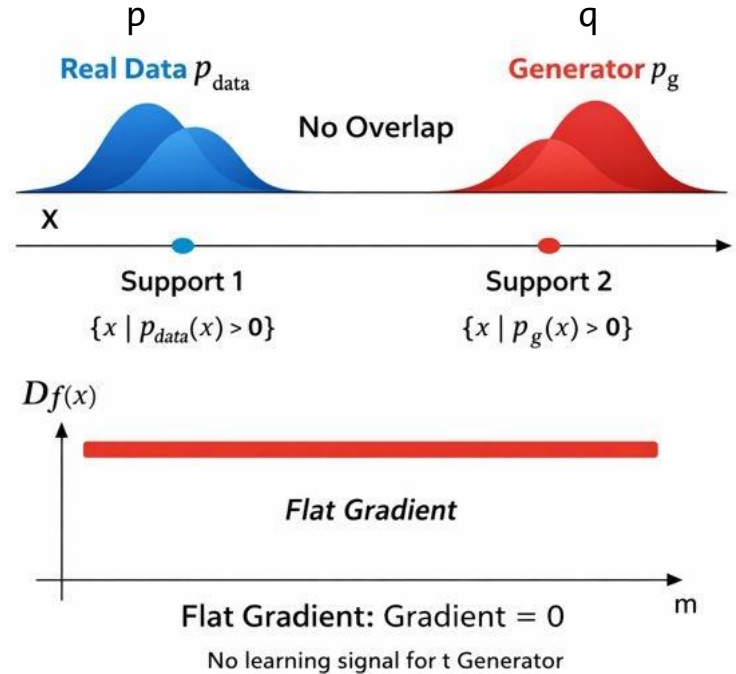


<https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html>

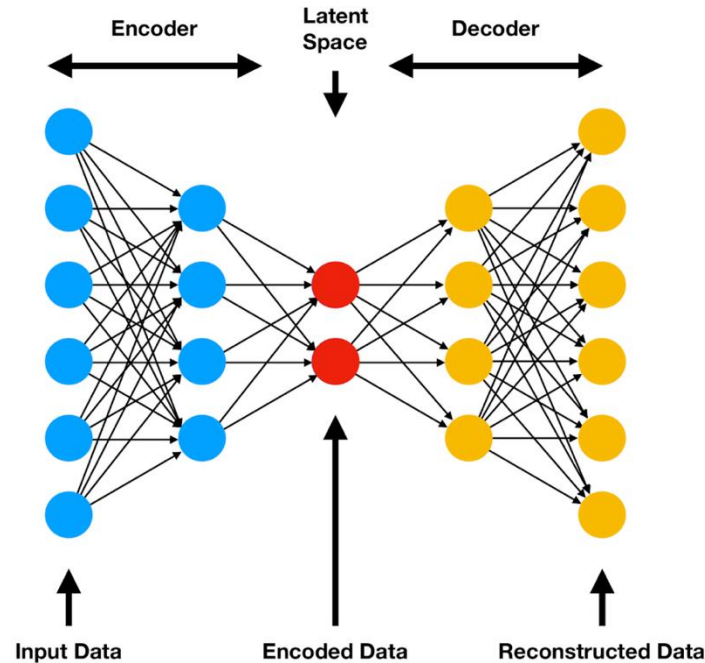
G generates data, D distinguishes between them, $V(D,G)$ couples both in such a way that the JSD is minimized at the optimum.

The GAN Problem of 2016–2017

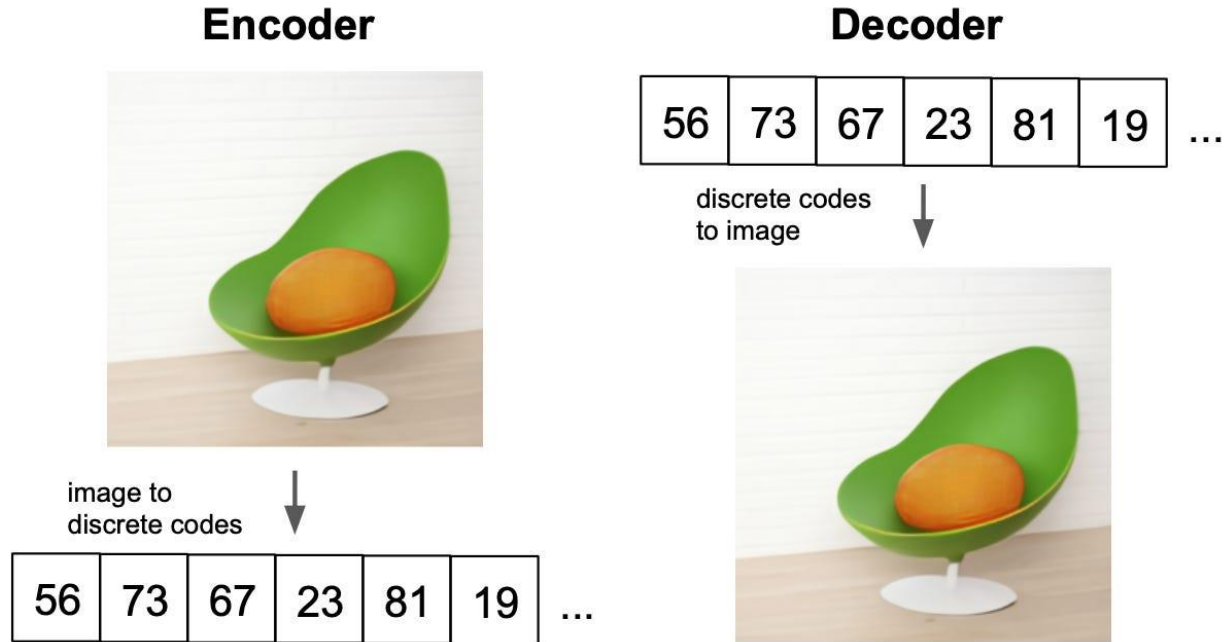
$D_{JS}(p||q) = \log 2$



Autoencoder

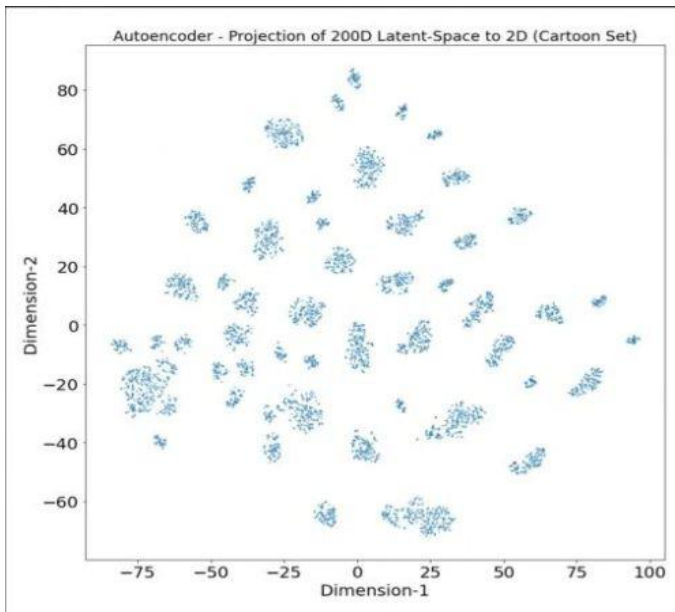


Autoencoder - Example

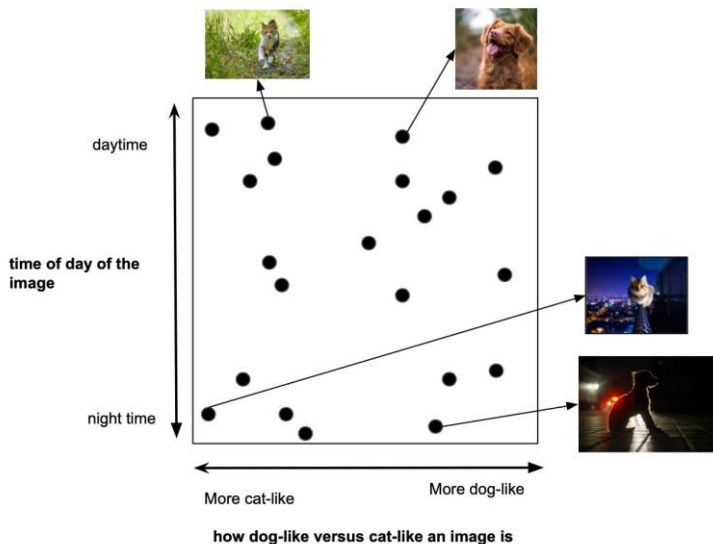


Continuous latent space, but...

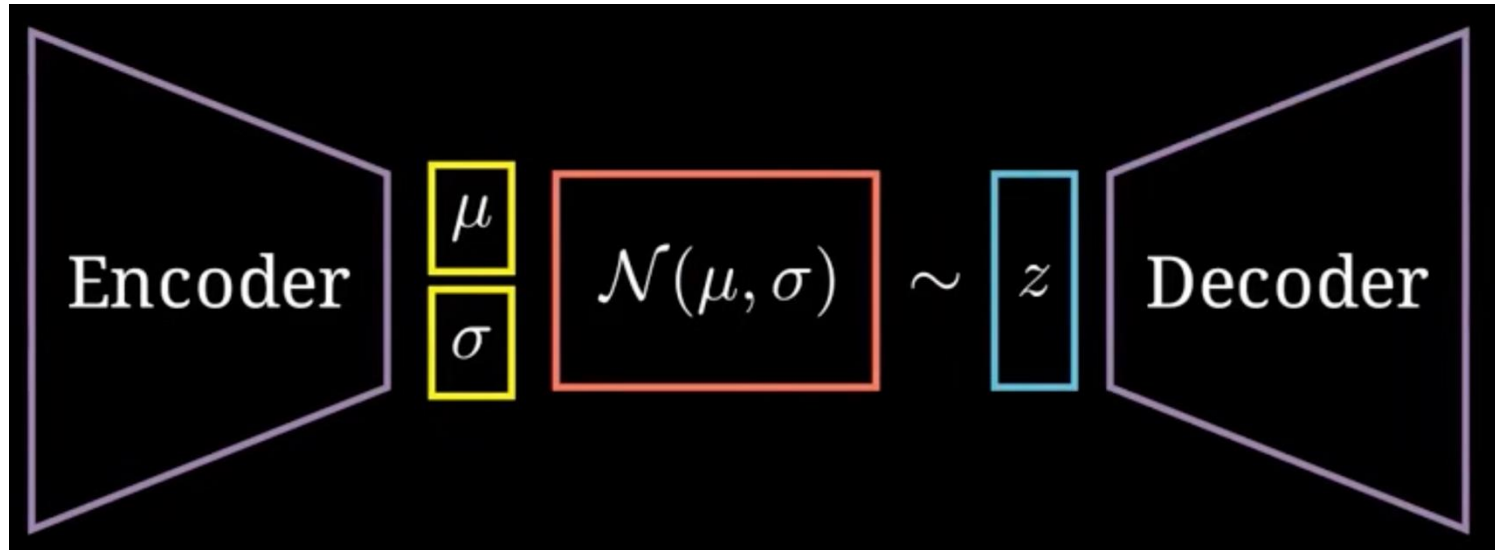
Related Work - Autoencoder problem



An Oversimplified Example of a Cat/Dog Image Latent Space



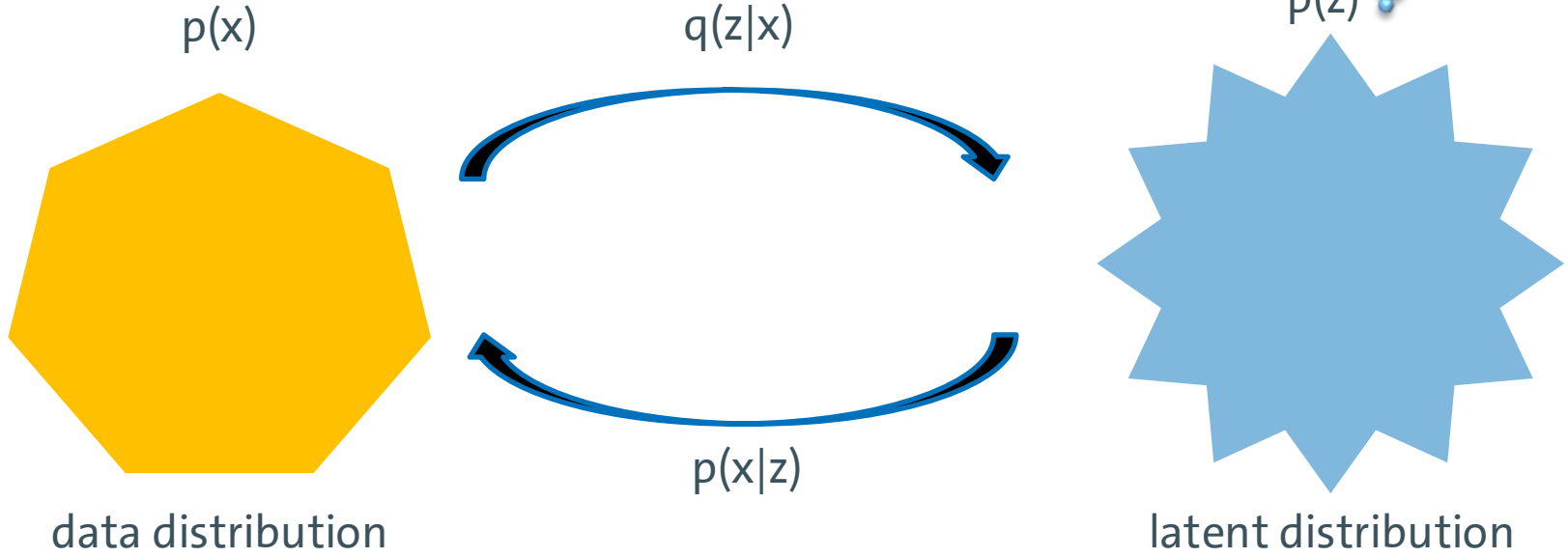
Variational Autoencoder (VAE)



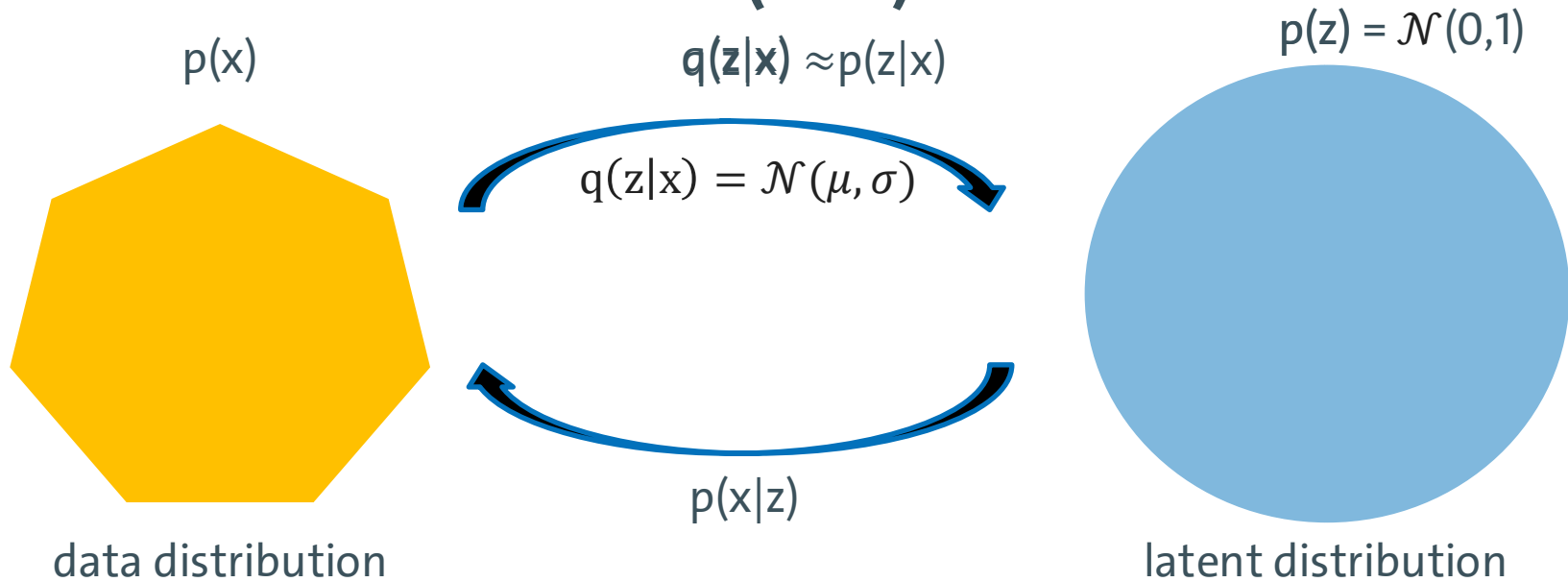
<https://www.youtube.com/watch?v=qleaCHQ1k2w>

We don't know
the latent
distribution of $p(z)$

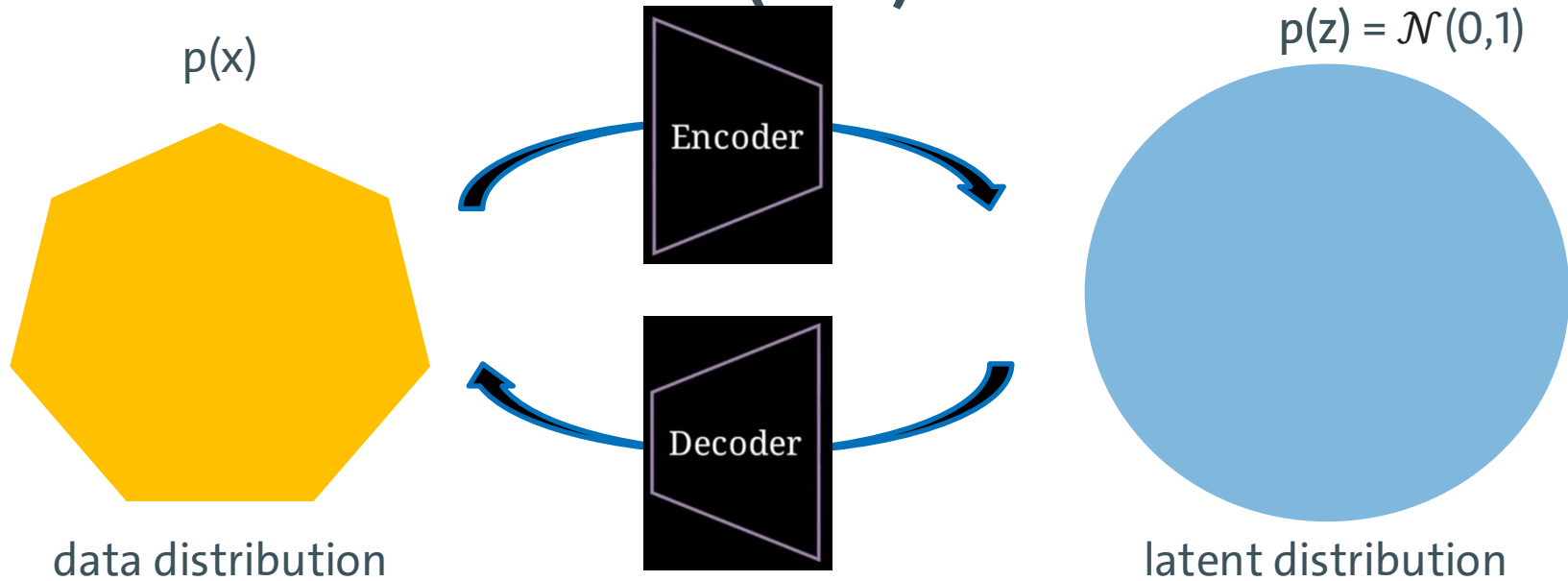
Variational Autoencoder (VAE)



Variational Autoencoder (VAE)



Variational Autoencoder (VAE)



Variational Autoencoder (VAE)

How to train this autoencoder?

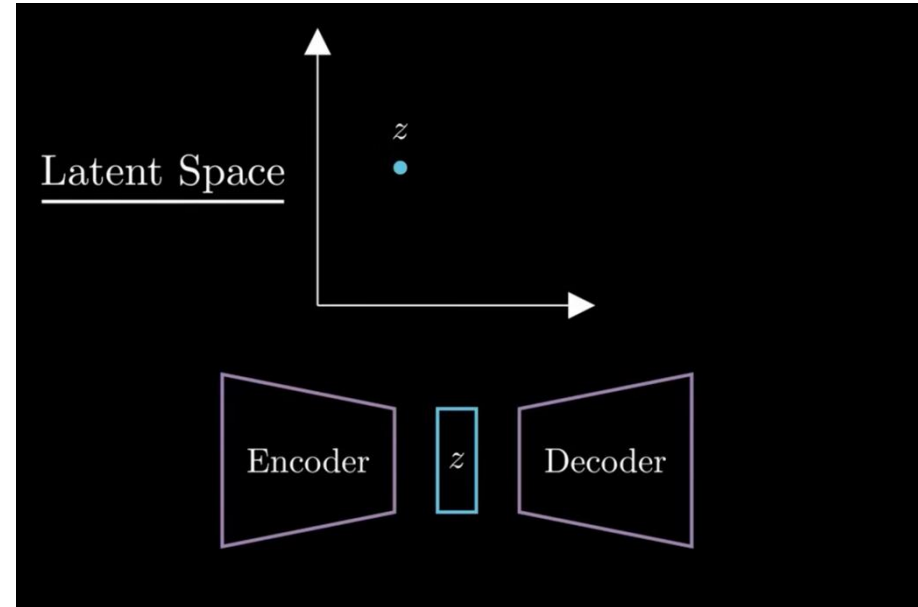
$$\underbrace{-E_{z \sim q(z|x)}[\log(p(x|z))]}_{\text{Data consistency (usual loss)}} + \underbrace{KL(q(z|x) || p(z))}_{\text{Regularization (imposing a normal distribution on the latent space)}}$$

Data consistency
(usual loss)

Regularization
(imposing a
normal
distribution on
the latent space)

Variational Autoencoder (VAE)

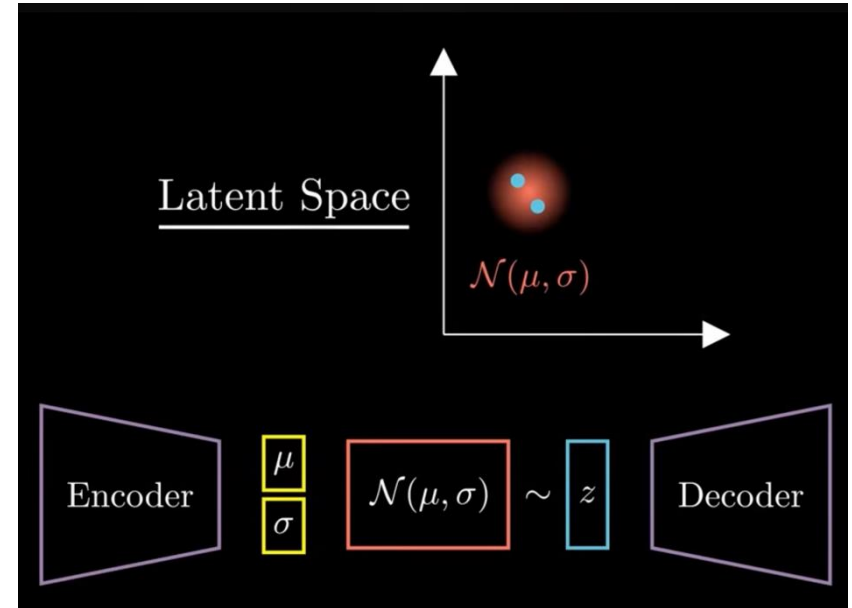
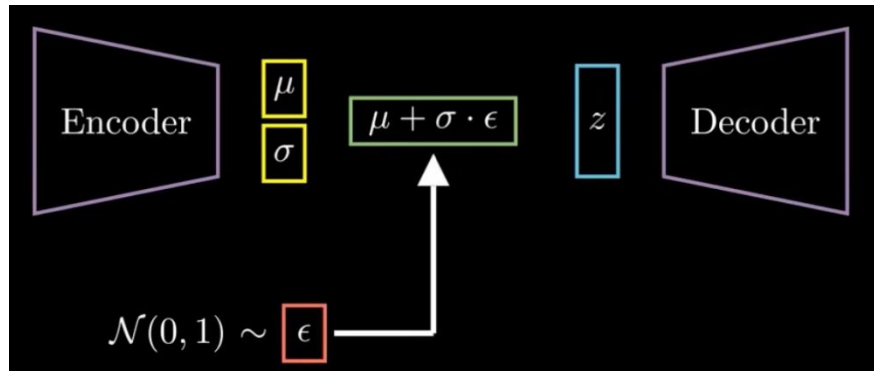
How to train this autoencoder in practice?



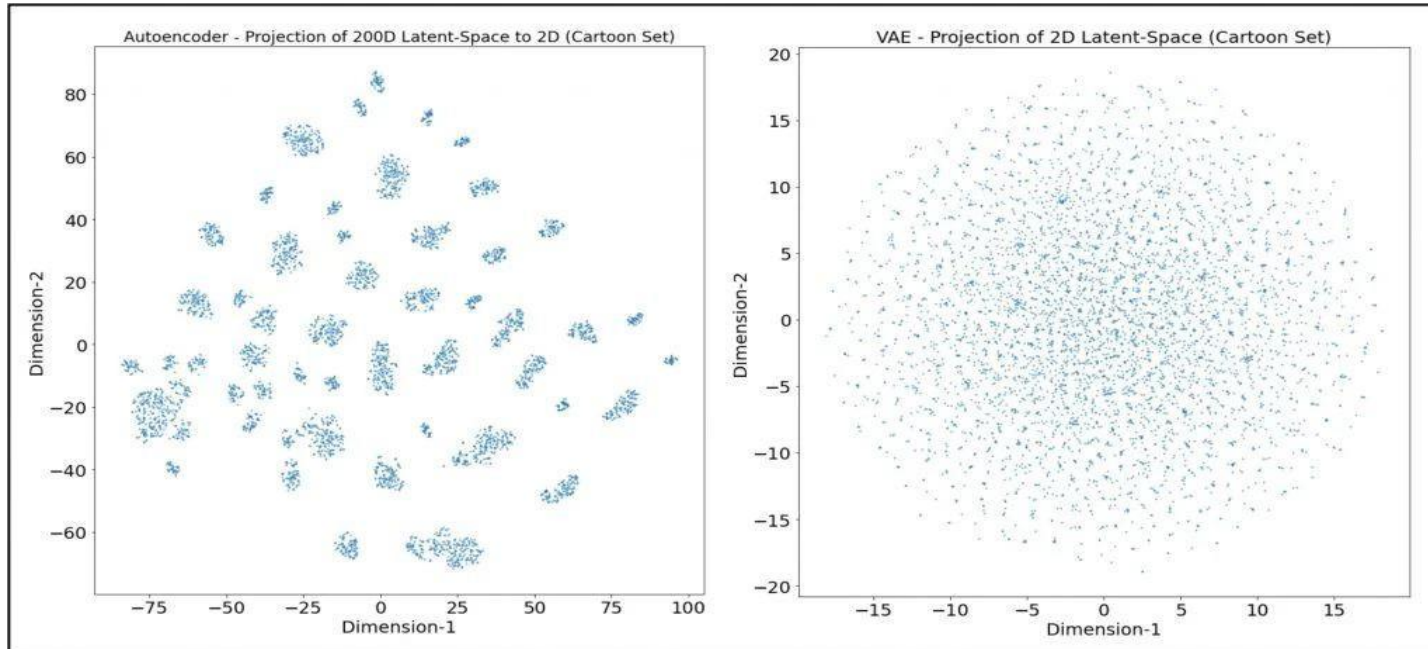
Variational Autoencoder (VAE)

How to backpropagate to the sampling process?

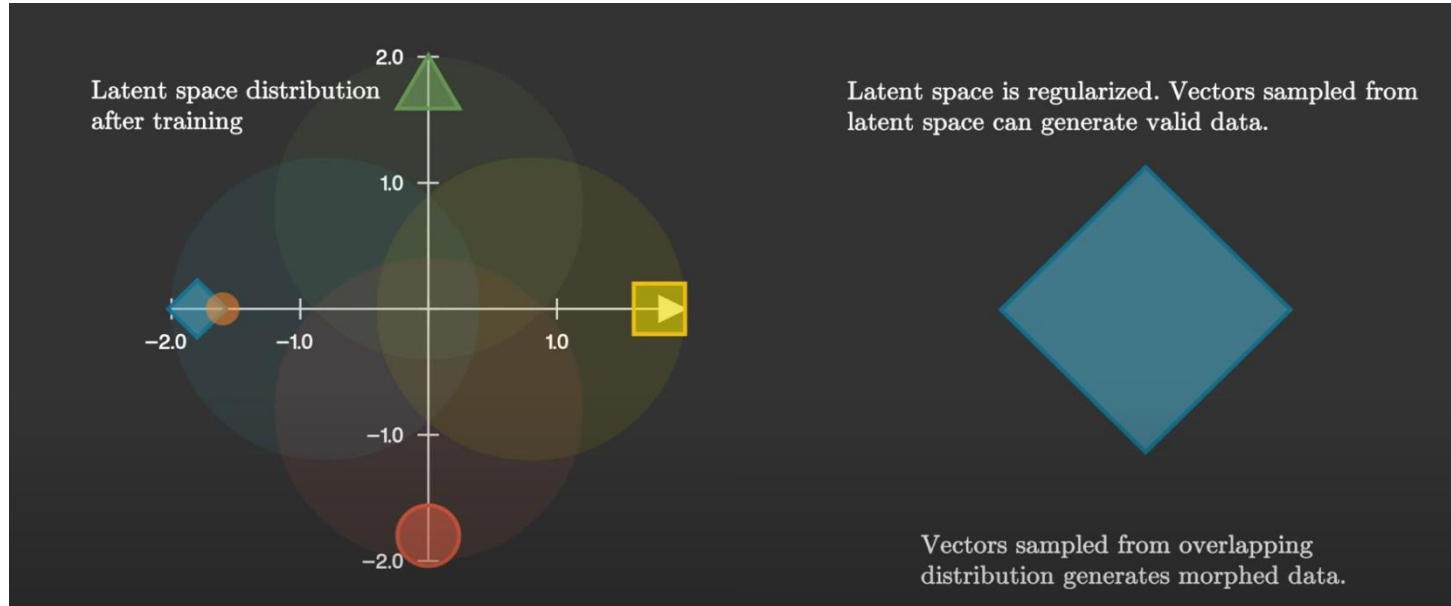
-->Reparameterization Trick



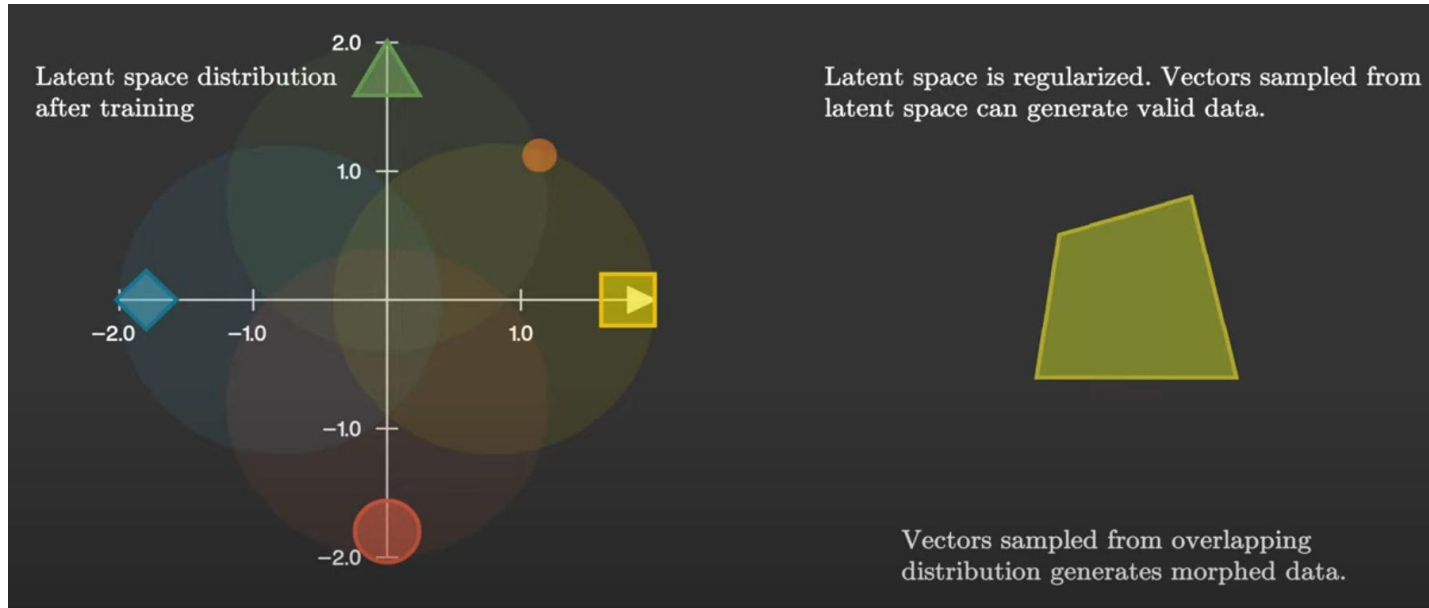
Autoencoder vs. VAE



VAE

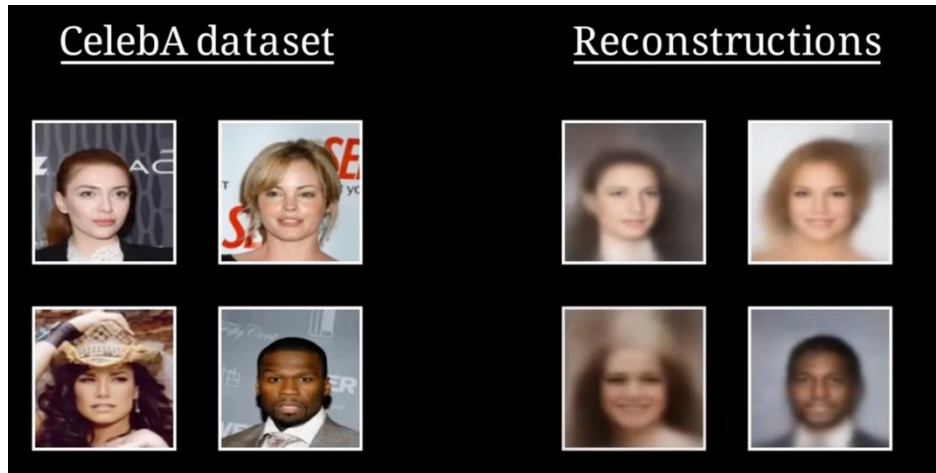


VAE



VAE results

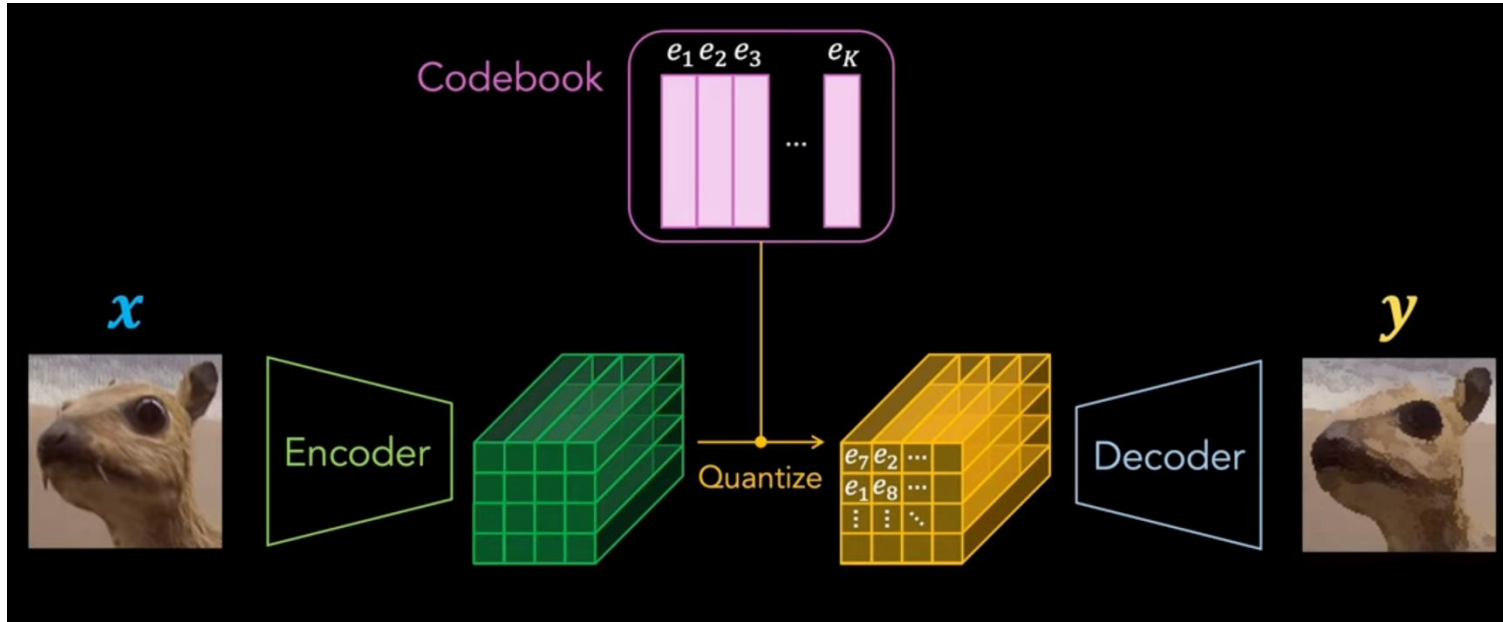
- Creating new images, though the results are blurry



Vector Quantized VAEs, which provide a latent space for sharper reconstructions.

<https://doi.org/10.48550/arXiv.1711.00937>

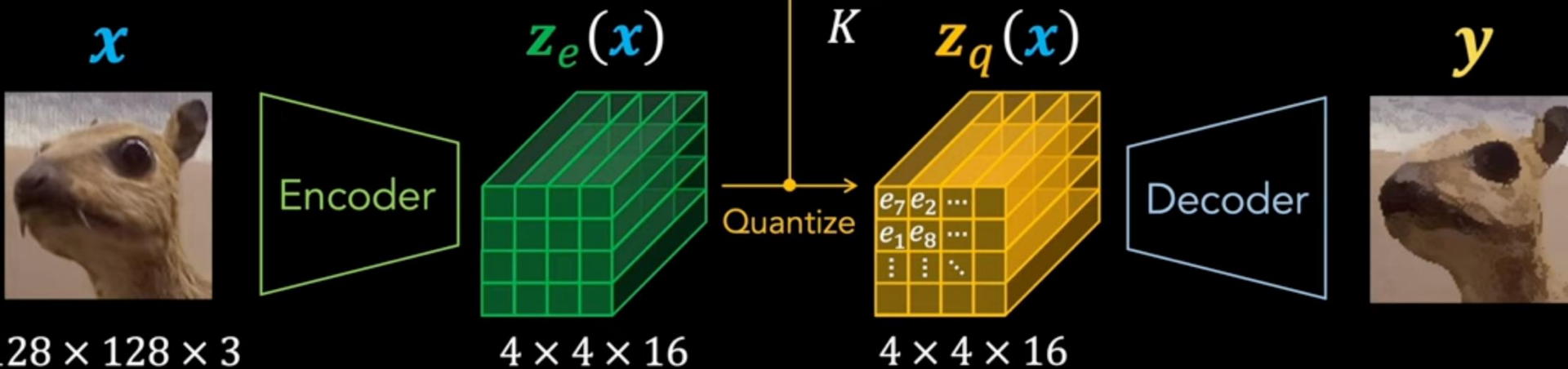
Vector Quantized (VQ)-VAE



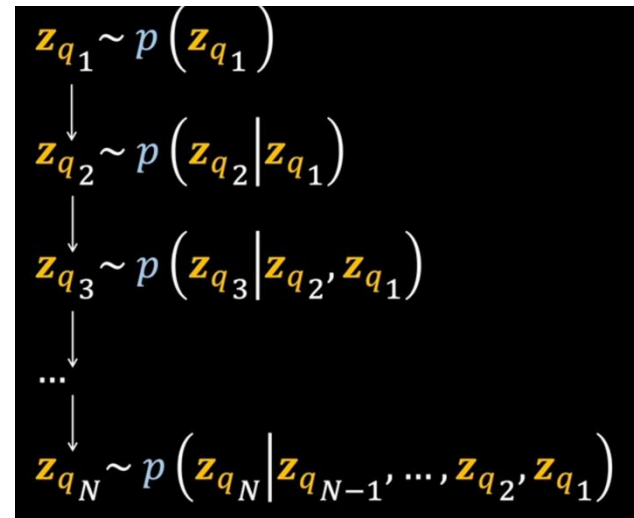
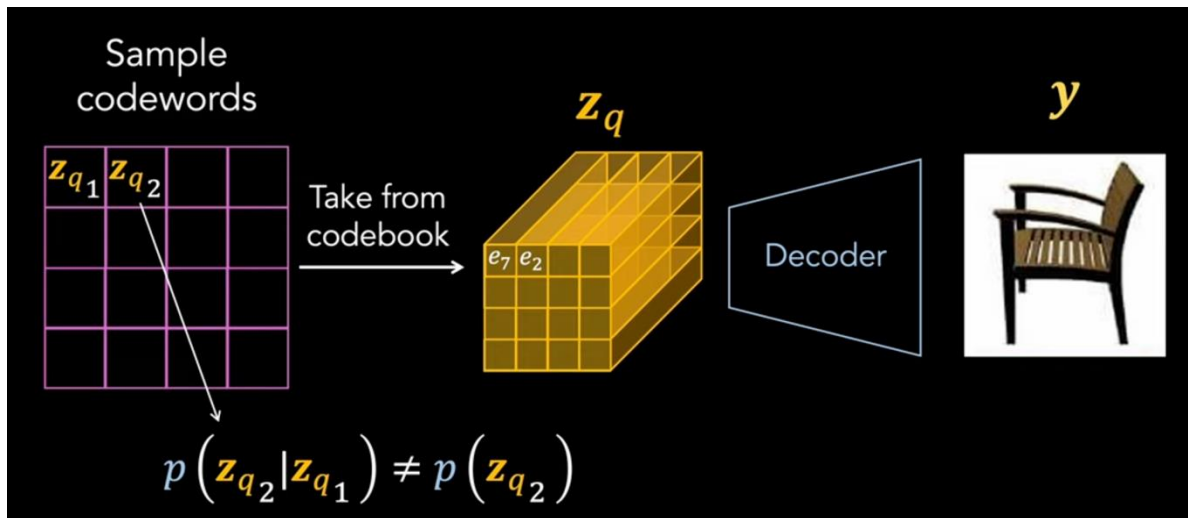
$$\mathbf{z}_q(\mathbf{x}) = e_k$$

where

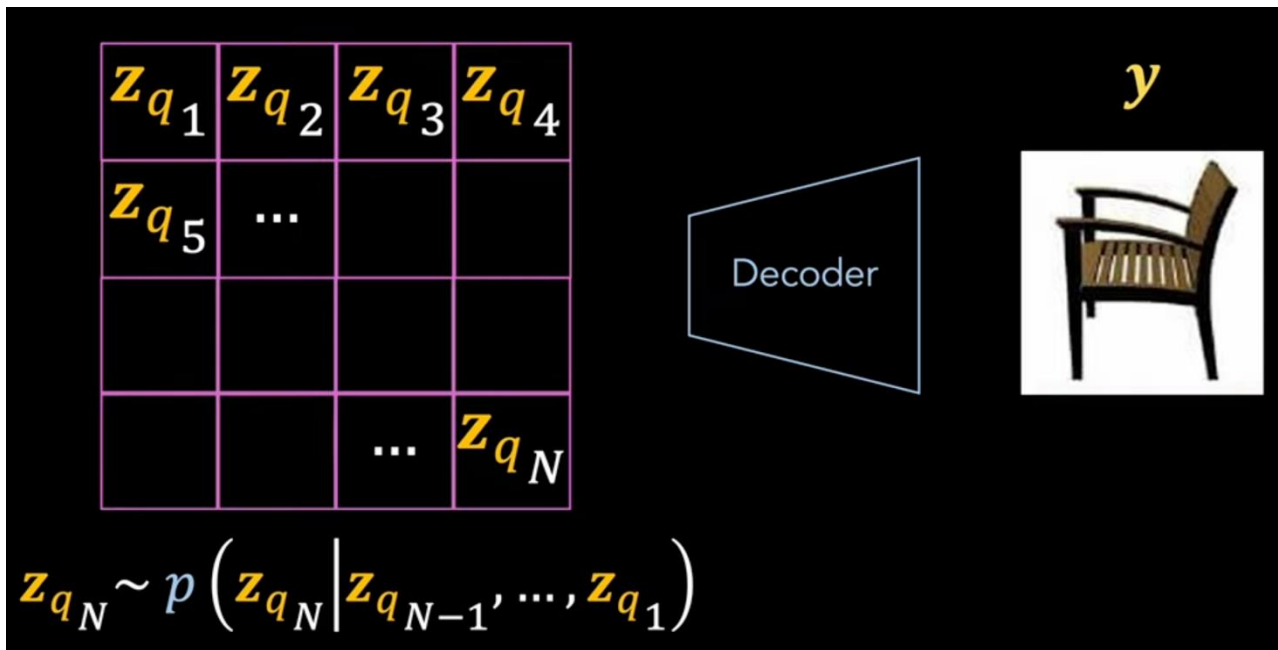
$$k = \operatorname{argmin}_j \|\mathbf{z}_e(\mathbf{x}) - e_j\|_2$$



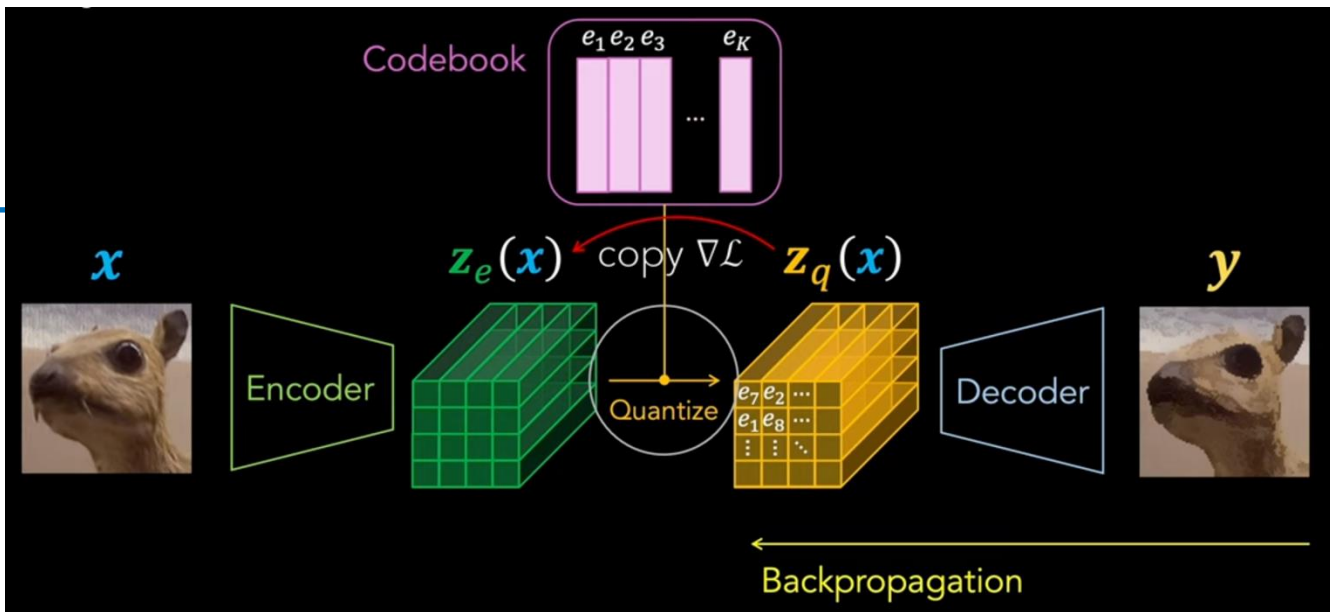
VQ-VAE



VQ-VAE



VQ-VAE



(typically weighted by γ)

$$\mathcal{L}(\theta, \phi, C; x, z_q) = -\log p_{\theta}(x|z_q) + \|\text{sg}[z_e(x)] - e\|_2^2 + \gamma \|z_e(x) - \text{sg}[e]\|_2^2$$

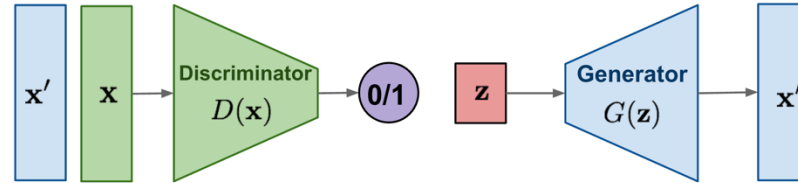
Reconstruction loss (optimizes ϕ, θ) Codebook loss (optimizes C) Commitment loss

VQ-VAE

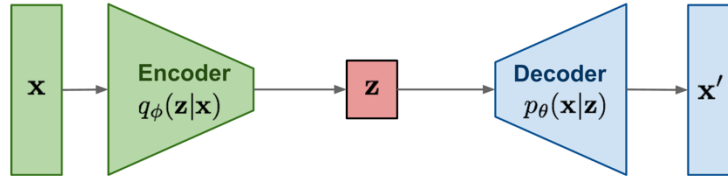
- DALL-E: The original DALL-E model used VQ-VAE to compress images into a discrete set of tokens that could then be modeled by a transformer.
- Jukebox: Used for high-fidelity music generation.
- Voice Conversion: Applied in research for acoustic unit discovery and voice conversion tasks.
- <https://xnought.github.io/vq-vae-explainer/>

Generative Models

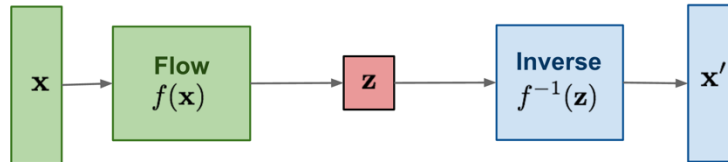
GAN: Adversarial training



VAE: maximize variational lower bound



Flow-based models:
Invertible transform of distributions



Diffusion models:
Gradually add Gaussian noise and then reverse



Diffusion Models

Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

Abstract

We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decompression scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset, we obtain an Inception score of 9.46 and a state-of-the-art FID score of 3.17. On 256x256 LSUN, we obtain sample quality similar to ProgressiveGAN. Our implementation is available at <https://github.com/hojonathanho/diffusion>

Jonathan Ho, Ajay Jain, Pieter Abbeel, **Denoising Diffusion Probabilistic Models, 2020**, <https://doi.org/10.48550/arXiv.2006.11239>

Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Jascha Sohl-Dickstein
Stanford University

JASCHA@STANFORD.EDU

Eric A. Weiss
University of California, Berkeley

EWEISS@BERKELEY.EDU

Niru Maheswaranathan
Stanford University

NIRUM@STANFORD.EDU

Surya Ganguli
Stanford University

SGANGULI@STANFORD.EDU

Abstract

A central problem in machine learning involves modeling complex data-sets using highly flexible families of probability distributions in which learning, sampling, inference, and evaluation are still analytically or computationally tractable. Here, we develop an approach that simultaneously achieves both flexibility and tractability. The essential idea, inspired by non-equilibrium statistical physics, is to systematically and slowly destroy structure in a data distribution through an iterative forward diffusion process. We then learn a reverse diffusion process that restores structure in data, yielding a highly flexible and tractable generative model of the data. This approach allows us to rapidly learn, sample from, and evaluate probabilities in deep generative models with thousands of layers or time steps, as well as to compute conditional and posterior probabilities under the learned model. We additionally release an open source reference implementation of the algorithm.

these models are unable to aptly describe structure in rich datasets. On the other hand, models that are *flexible* can be molded to fit structure in arbitrary data. For example, we can define models in terms of any (non-negative) function $q(\mathbf{x})$ yielding the flexible distribution $p(\mathbf{x}) = \frac{q(\mathbf{x})}{Z}$, where Z is a normalization constant. However, computing this normalization constant is generally intractable. Evaluating, training, or drawing samples from such flexible models typically requires a very expensive Monte Carlo process.

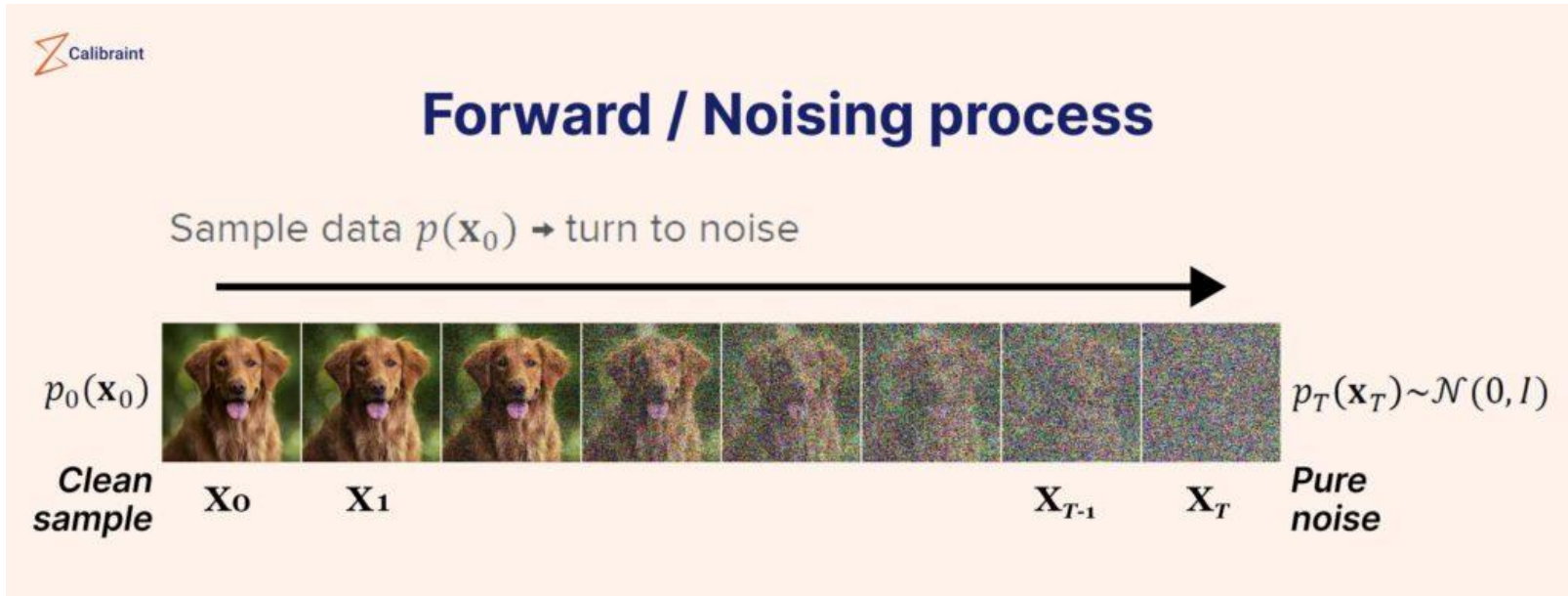
A variety of analytic approximations exist which ameliorate, but do not remove, this tradeoff—for instance mean field theory and its expansions (T, 1982; Tanaka, 1998), variational Bayes (Jordan et al., 1999), contrastive divergence (Welling & Hinton, 2002; Hinton, 2002), minimum probability flow (Sohl-Dickstein et al., 2011b), minimum KL contraction (Lyu, 2011), proper scoring rules (Gneiting & Raftery, 2007; Parry et al., 2012), score matching (Hyvriinen, 2005), pseudolikelihood (Besag, 1975), loopy belief propagation (Murphy et al., 1999), and many, many more. Non-parametric methods (Gershman & Blei, 2012) can also be very effective!

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, Surya Ganguli, **Deep Unsupervised Learning using Nonequilibrium Thermodynamics, 2015**, <https://doi.org/10.48550/arXiv.1503.03585>

Why Diffusion Models?

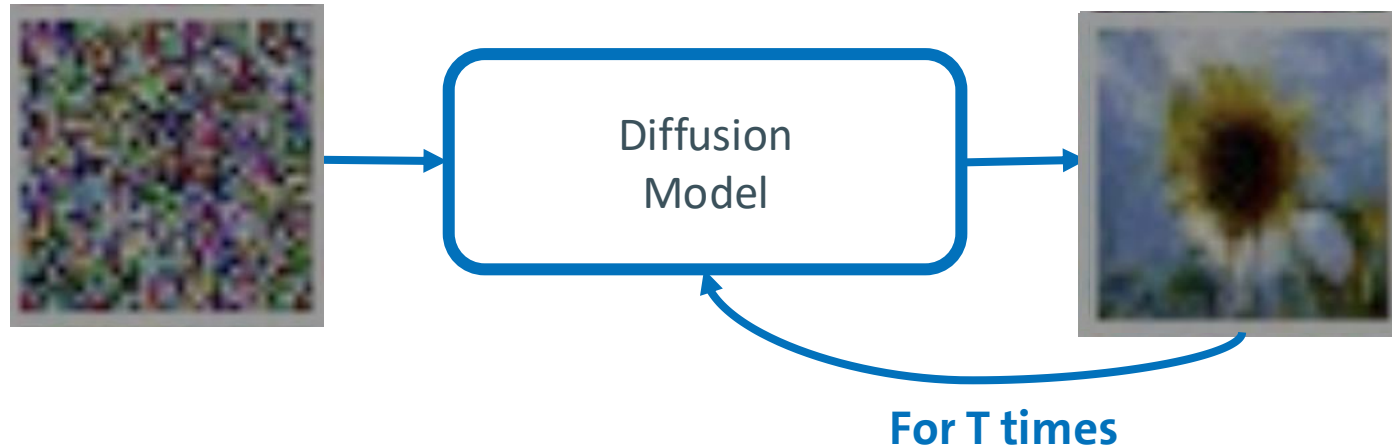
- Aim: Learning the data distribution (e.g., images)
- Traditional approaches:
 - GANs → unstable training
 - VAEs → blurry outputs
- Diffusion models:
 - Stable training
 - High-quality outputs
 - Representing the density (likelihood)

Diffusion Models



Diffusion Models

Key idea: Generating images through reversing a gradual noising process



Images/adapted process from: <https://learnopencv.com/denoising-diffusion-probabilistic-models/>

Diffusion Models

Two processes:

- Forward diffusion (noising)
- Reverse diffusion (denoising, learned)

„Slowly destroy data with noise, then learn how to undo it.“

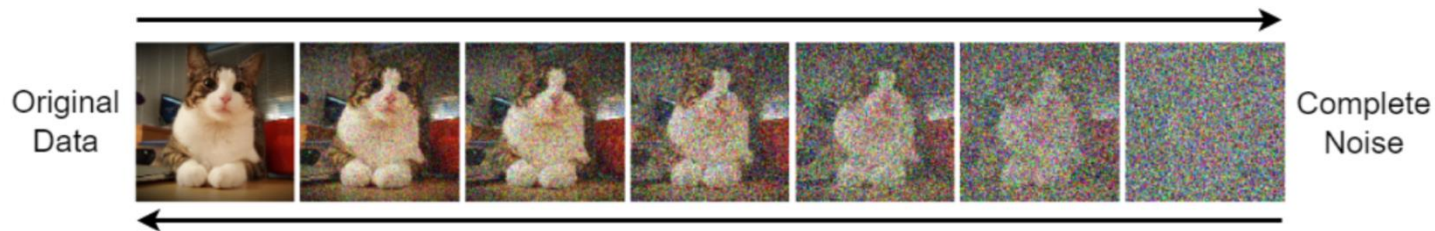


Image: Benyamin Ghojogh, Ali Ghodsi, Diffusion Models: Tutorial and Survey, 2024. <https://doi.org/10.31219/osf.io/w7jcm>

Diffusion Models

The Forward Process

$$\mathbf{x}_0 \rightarrow \mathbf{x}_1 \rightarrow \dots \rightarrow \mathbf{x}_T$$

Original
Data



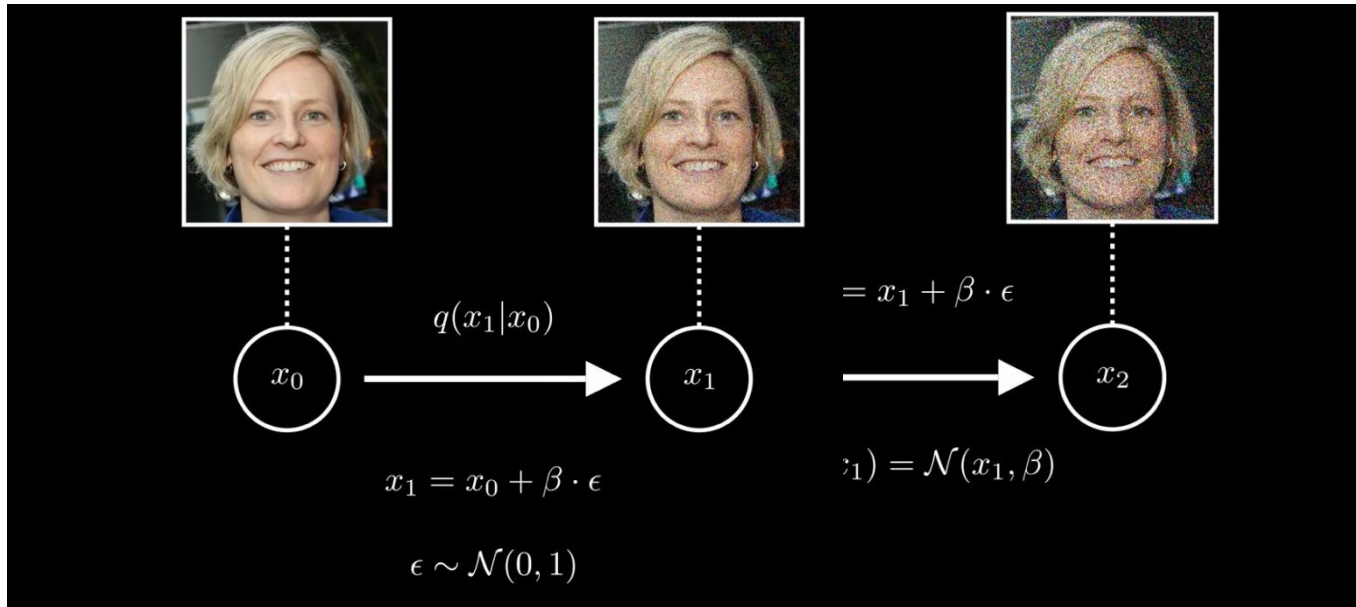
Complete
Noise

$$\mathbf{x}_0 \leftarrow \mathbf{x}_1 \leftarrow \dots \leftarrow \mathbf{x}_T$$

The Generative Backward Process

Image: Benyamin Ghojogh, Ali Ghodsi, Diffusion Models: Tutorial and Survey, 2024. <https://doi.org/10.31219/osf.io/w7jcm>

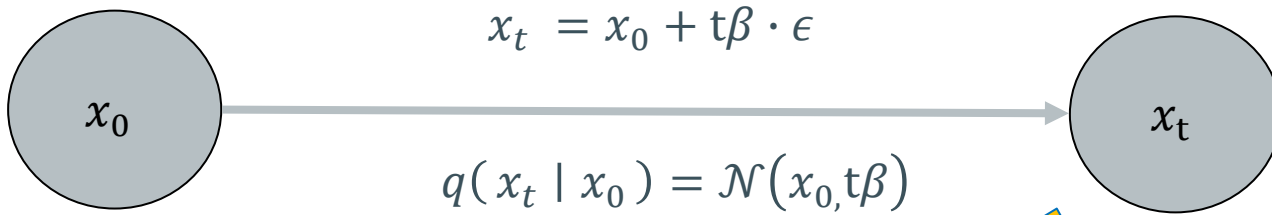
Diffusion Models – Forward Process



<https://www.youtube.com/watch?v=EhndHhIvWWw>

GenAI | Ralf Möller, Sylvia Melzer

Diffusion Models

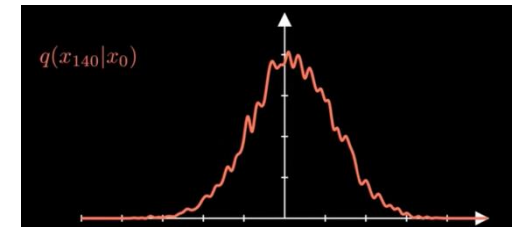
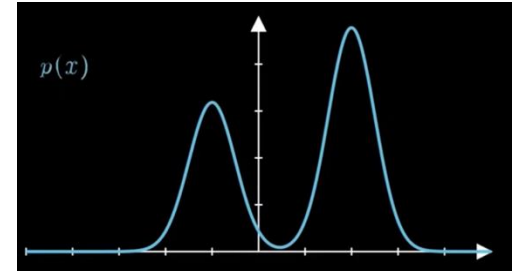
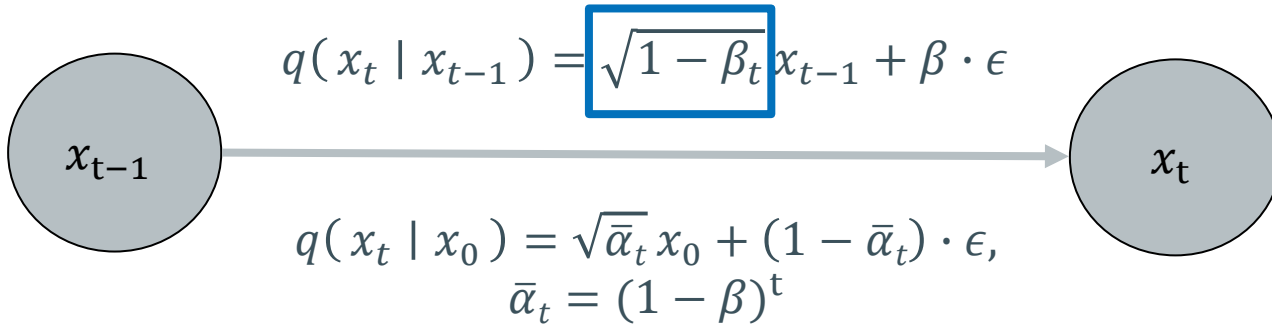


The diffusion process does not converge to the normal distribution.

Goal:
 $q(x_t | x_0) \xrightarrow{t \rightarrow \infty} \mathcal{N}(0, 1)$

Goal:
 $q(x_t | x_0) \xrightarrow{t \rightarrow \infty} \mathcal{N}(0,1)$

Diffusion Models



Diffusion Models

- Forward diffusion: Add Gaussian noise over t steps

$q(x_t | x_{t-1})$ defines the probability density function of an image at timestep t in the forward diffusion process x_t given the image x_{t-1}

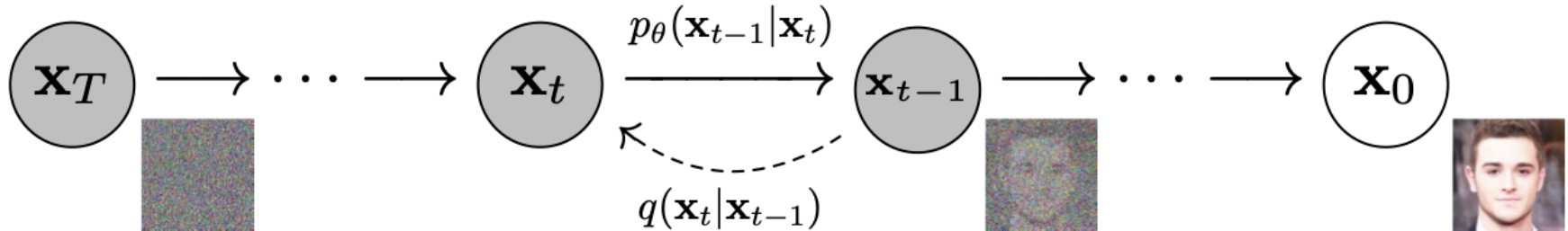
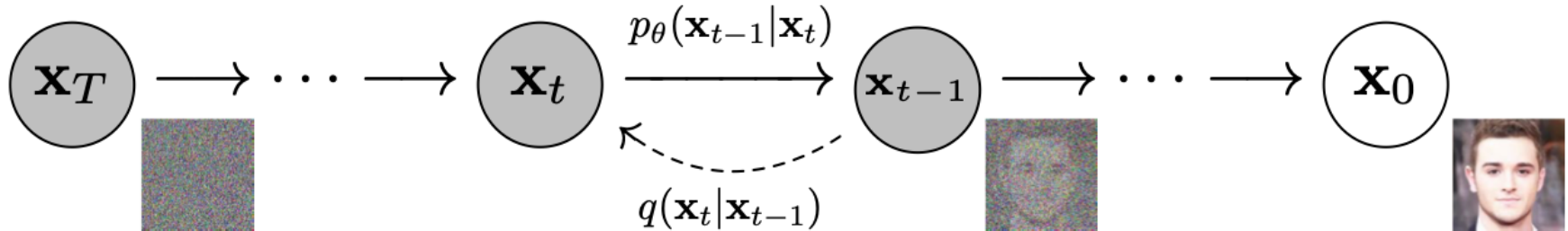


Image: Benyamin Ghojogh, Ali Ghodsi, Diffusion Models: Tutorial and Survey, 2024. <https://doi.org/10.31219/osf.io/w7jcm>

Diffusion Models

Reverse Diffusion: Removing noise

- Goal: $p_{\theta}(x_{t-1} | x_t)$
- Approach: Minimize $-\log p_{\theta}(x_0)$

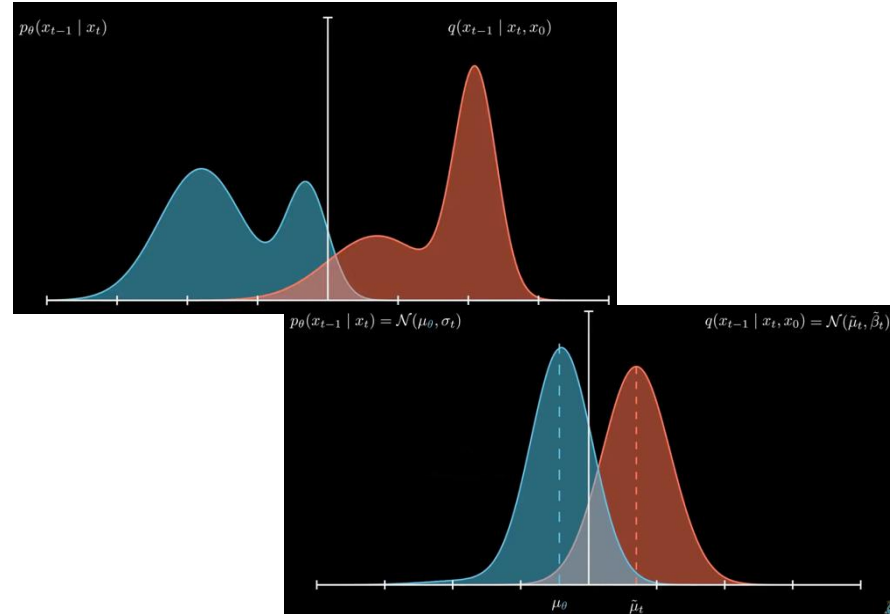
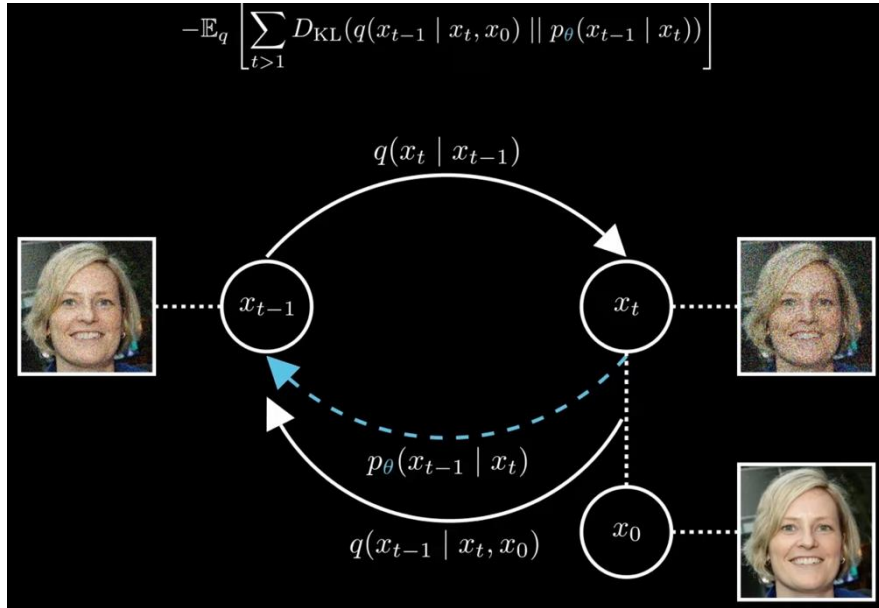


Jonathan Ho, Ajay Jain, Pieter Abbeel, **Denoising Diffusion Probabilistic Models**, 2020, <https://doi.org/10.48550/arXiv.2006.11239>

Detail
 computation:
 see paper

Diffusion Models

<https://www.youtube.com/watch?v=EhndHhivWWw>



1 Negative log-likelihood

$$-\log p_{\theta}(x_0) = -\log \int p_{\theta}(x_0, x_t) dx_t$$


2 Evidence lower bound

$$-\log p_{\theta}(x_0) \leq \mathbb{E}_q \left[-\log \frac{p_{\theta}(x_0, x_t)}{q(x_t | x_0)} \right]$$


3 Rewrite the ELBO

$$\mathbb{E}_q [D_{\text{KL}}(q(x_T | x_0) || p(x_T)) + \sum_{t>1} D_{\text{KL}}(q(x_{t-1} | x_t, x_0) || p_{\theta}(x_{t-1} | x_t)) - \log p_{\theta}(x_0 | x_1)]$$


Predicting Noise Instead of Images

6 Reparameterization trick

$$\mathbb{E}_q \left[\sum_{t>1} \frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1 - \bar{\alpha}_t)} \|\epsilon - \epsilon_{\theta}(x_t, t)\|^2 \right]$$

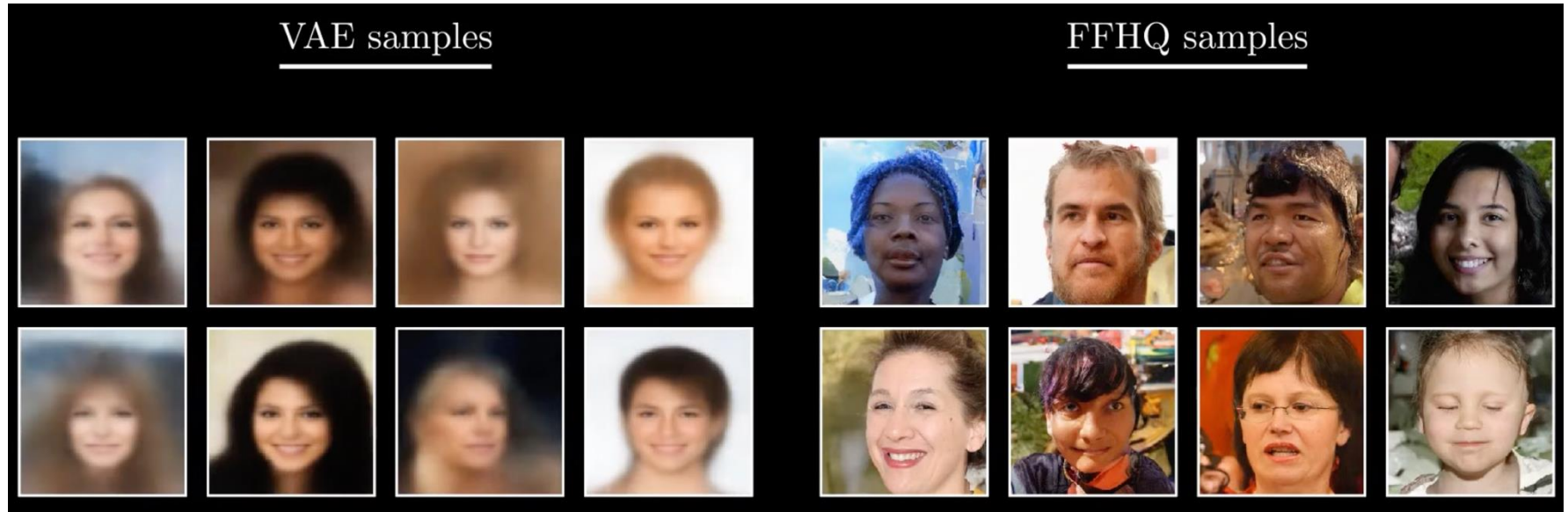

5 Use Gaussians

$$\mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \sum_{t>1} \|\tilde{\mu}_t - \mu_{\theta}(x_t, t)\|^2 \right]$$


4 Simplify the ELBO

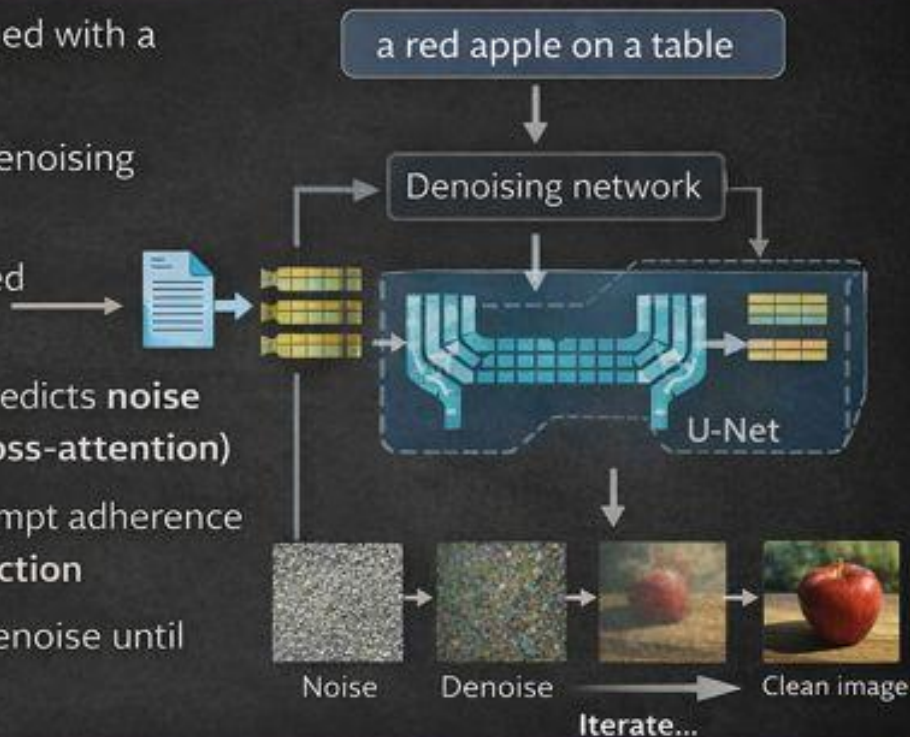
$$\mathbb{E}_q \left[\sum_{t>1} D_{\text{KL}}(q(x_{t-1} | x_t, x_0) || p_{\theta}(x_{t-1} | x_t)) \right]$$

Diffusion Models - Results



Text-to-Image Generation (Diffusion Models)

- **Goal:** Generate an image that is semantically aligned with a natural-language prompt
- **Model:** Diffusion-based generative model with a denoising network (U-Net)
- **Text Encoding:** The prompt is converted into a fixed semantic embedding using a **transformer**
- **Conditioning:** At each denoising step, the model predicts **noise** conditioned on the image, timestep, and **text** (via **cross-attention**)
- **Guidance:** Classifier-free guidance strengthens prompt adherence by steering denoising toward the **conditional prediction**
- **Generation:** Start from pure noise and iteratively denoise until a clean image matching the prompt emerges



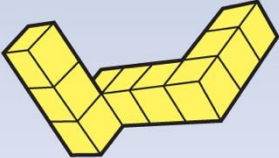
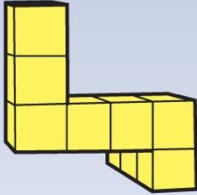
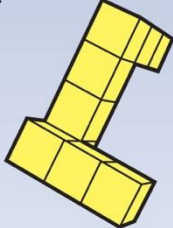
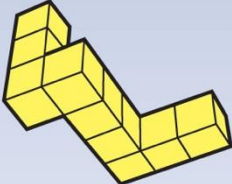
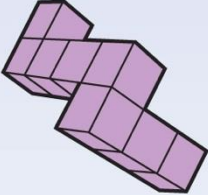
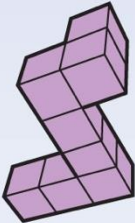
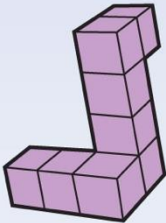
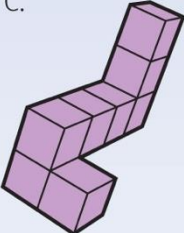
Key intuition: "Language does not create pixels directly—it guides the denoising trajectory toward semantically consistent images."

Image created with ChatGPT 5.2

Nice Applications. But...

- How can agents use text-descriptive image generation?
- Agent could generate “internal images” and interpret them to optimally carry out tasks
 - Planning might get easier with “mental imagery”
- Mental imagery (aka visual imagery) has a long tradition in cognitive psychology
- → Imagery Debate
 - Propositional or visual/”perceptive” reasoning
 - ”Pylyshyn vs. Kosslyn”

Mental Rotation

Standard	Comparison shapes		
1. 	A. 	B. 	C. 
2. 	A. 	B. 	C. 

Mental Scanning



Island stimulus for mental scanning used in (Kosslyn, Ball, & Reiser, 1978). The island contains different locations that differ in their distance to each other. In the lower left corner a hut, a well, a lake, and a tree are visible. On the top is a rock and further locations include grass and a beach.

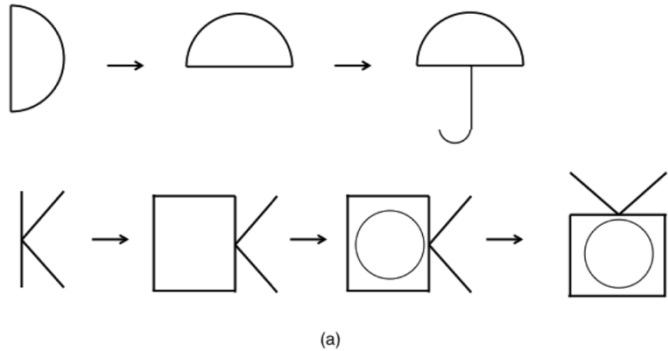
Kosslyn, S. M., Ball, T. M., & Reiser, B. J., Visual images preserve metric spatial information: evidence from studies of image scanning. *Journal of experimental psychology: Human Perception and Performance*, 4(1), 47, 1978.

Mental Scanning

- Using their mental image, participants are asked to shift their attention from one entity in the image to another entity.
- It turned out that participants take significantly longer for attention shift between, for example, the hut and the rock, ...
- ... than they do for a shift between the hut and the well

- Strong linear correlation between the time it takes to scan between two entities in the mental image and the distance between these two entities in the original stimulus

Mental Reinterpretation



(a)



(b)

(a) The first figure in each line is described to the participants verbally who then mentally transform their mental images according to verbal instructions so that the depicted intermediate figures should result.

(b) The respectively left one is briefly shown to the participants who then have to find an alternative interpretation of just the right side of the stimulus using their mental image.

Dual coding theory

- Human cognition divided into two processing systems: visual and verbal.
 - The visual system deals with graphical information processing and the verbal system deals with linguistic processing
 - These two systems are separate and are activated by different information
- GenAI: Text → Image/Video
- GenAI: Image/Video → Caption as text

Sadoski, Mark; Paivio, Allan, "A Dual Coding Theoretical Model of Reading", *Theoretical Models and Processes of Reading*, DE: International Reading Association, *Cognitive psychology*. Cengage Learning. pp. 1329–1362, 2016.

Counterfactuals and Causality

