



UNIVERSITY OF TECHNOLOGY  
IN THE EUROPEAN CAPITAL OF CULTURE  
CHEMNITZ

# Neurocomputing

Diffusion Probabilistic Models

Julien Vitay

Professur für Künstliche Intelligenz - Fakultät für Informatik

# **1 - Diffusion probabilistic models**

# Generative modeling

**1D example:**  
we illustrate the  
effet of G over  
the entire  
distribution

*Generative model  
to be learned*

*Simple 1D gaussian  
distribution we know  
how to sample from*

*Targeted complex 1D  
distribution we don't know  
how to sample from*

$$G(\text{---}) = \text{---}$$

**High dimension  
example:**  
we illustrate the  
effet of G over a  
single sample

*Generative model  
to be learned*

*High dimension data  
point from simple  
noise distribution*

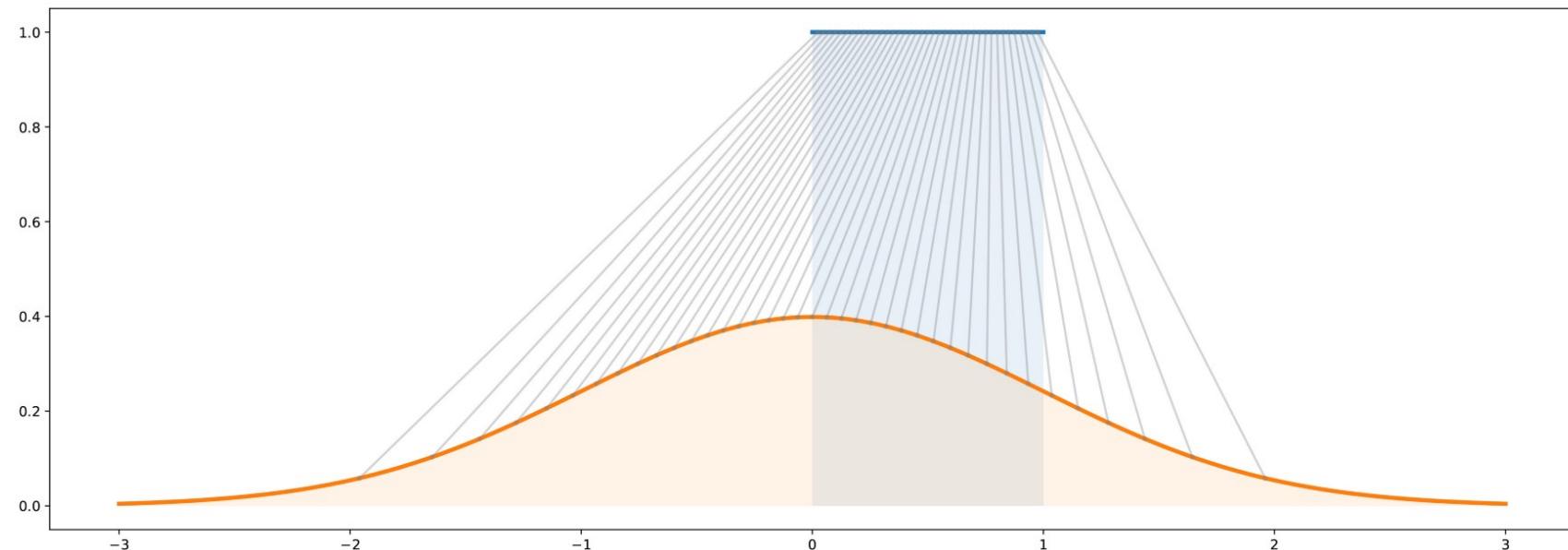
*High dimension data  
point from complex  
image distribution*

$$G(\text{---}) = \text{---}$$



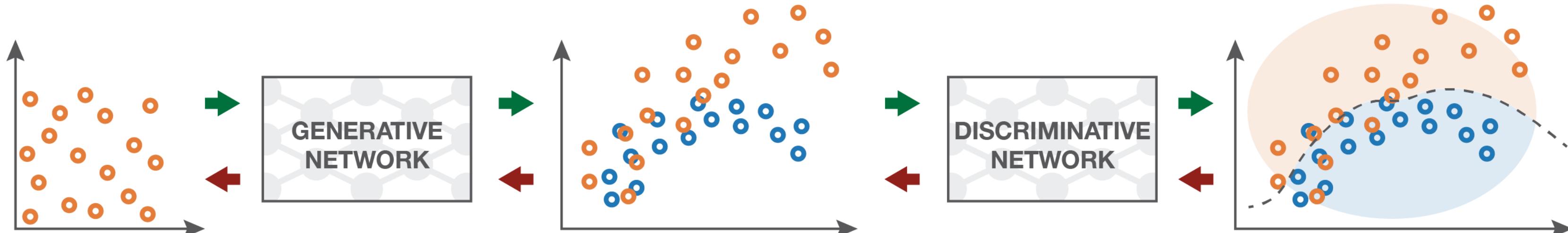
Source: <https://towardsdatascience.com/understanding-diffusion-probabilistic-models-dpms-1940329d6048>

# VAE and GAN generators transform simple noise to complex distributions



Forward propagation (generation and classification)

Backward propagation (adversarial training)



Input random variables.

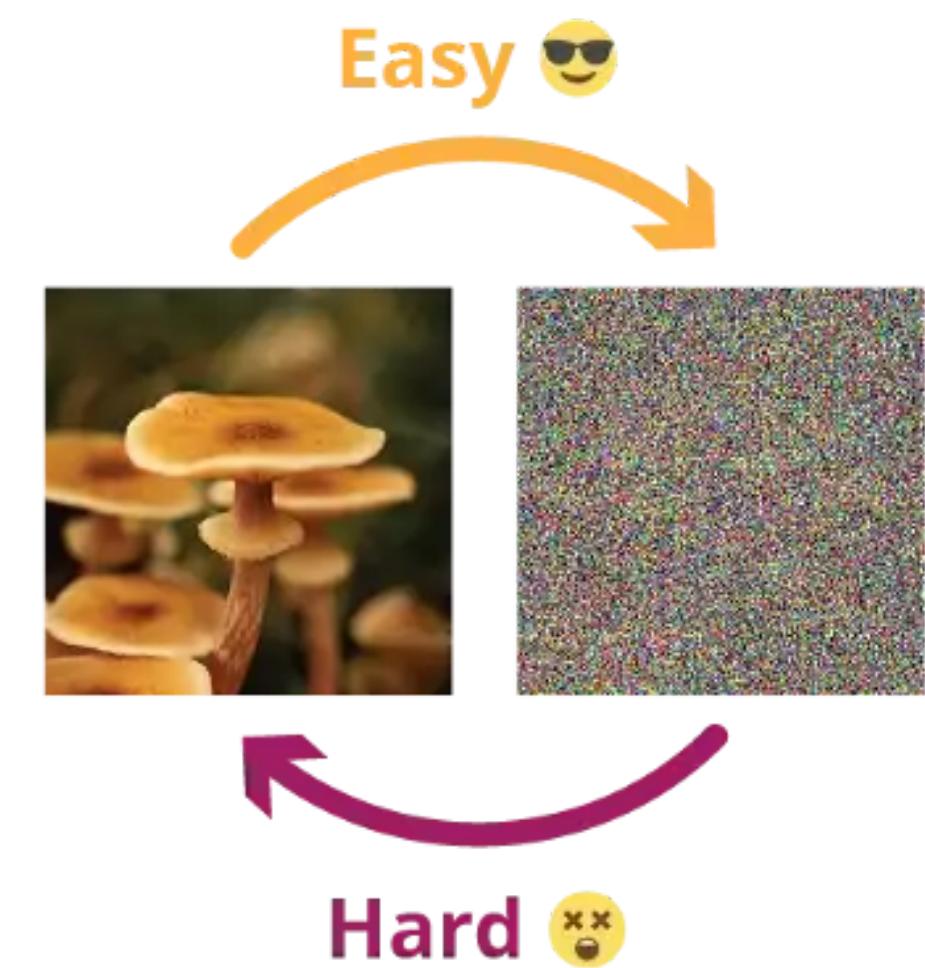
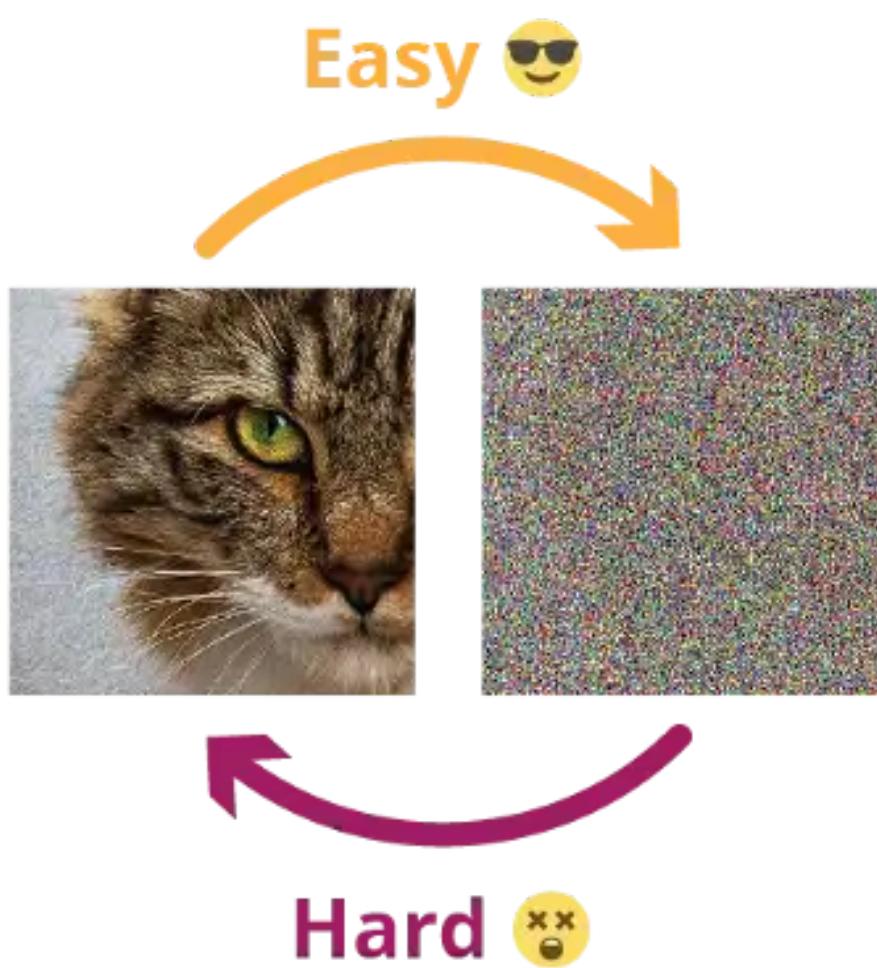
The generative network is trained to **maximise** the final classification error.

The **generated distribution** and the **true distribution** are not compared directly.

The discriminative network is trained to **minimise** the final classification error.

The classification error is the basis metric for the training of both networks.

# Destroying information is easier than creating it



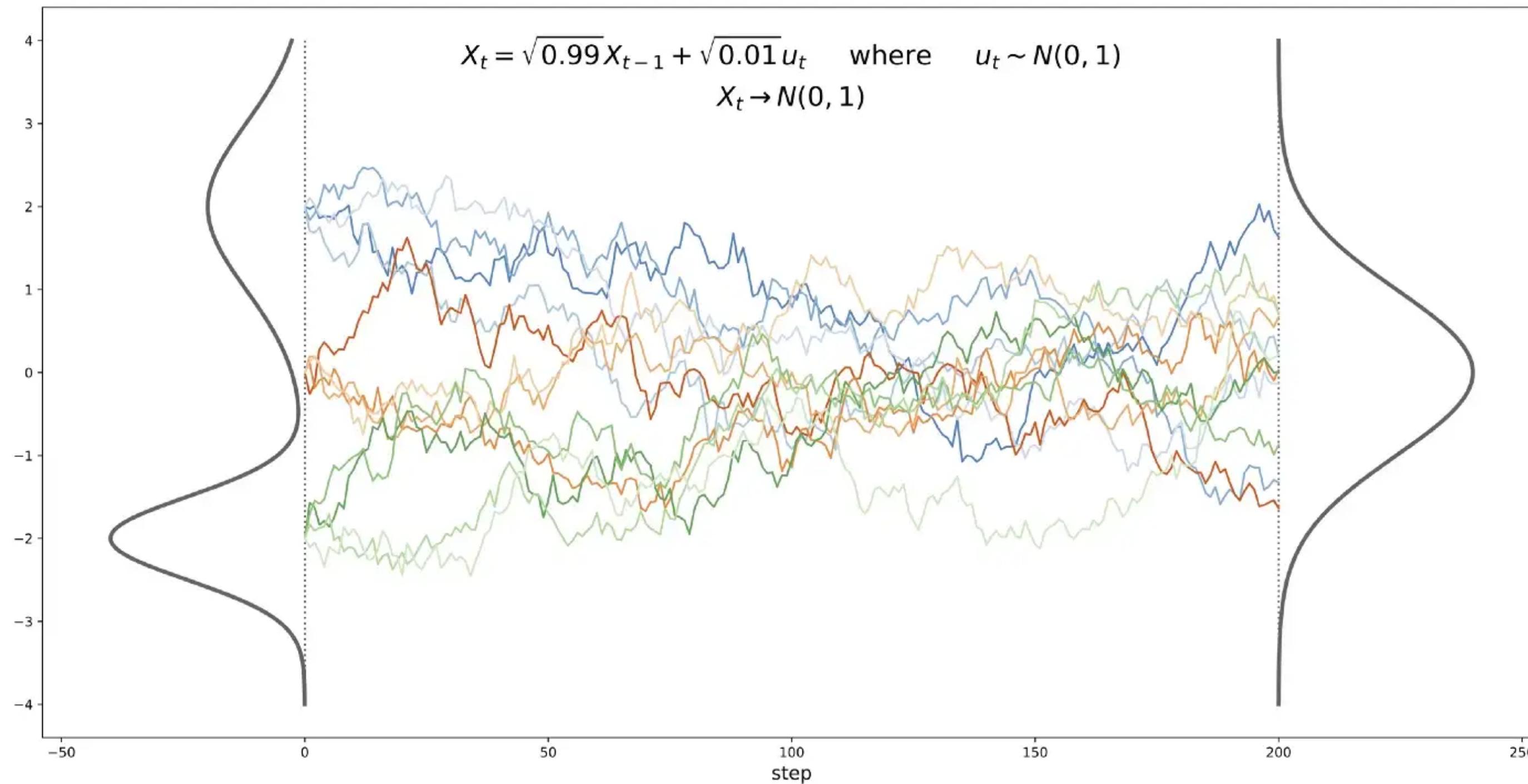
Source: <https://towardsdatascience.com/understanding-diffusion-probabilistic-models-dpms-1940329d6048>

# Stochastic processes can destroy information

- Iteratively adding normal noise to a signal creates a **stochastic differential equation** (SDE).

$$X_t = \sqrt{1 - p} X_{t-1} + \sqrt{p} \sigma \quad \text{where} \quad \sigma \sim \mathcal{N}(0, 1)$$

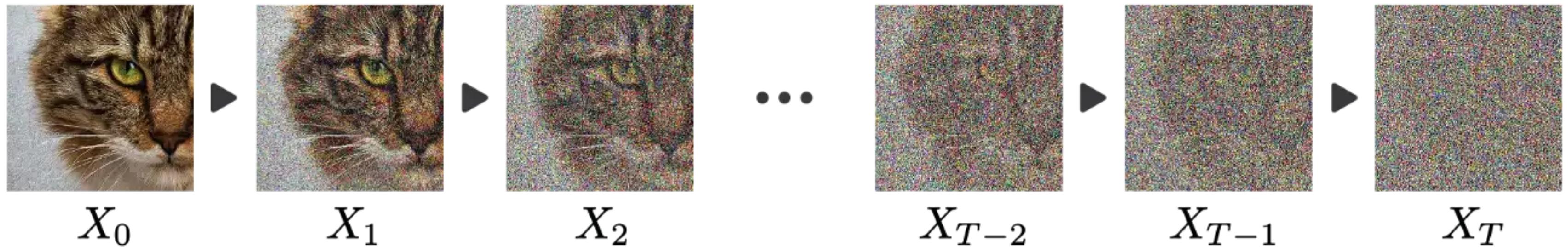
- Under some conditions, any probability distribution converges to a normal distribution.



Source: <https://towardsdatascience.com/understanding-diffusion-probabilistic-models-dpms-1940329d6048>

# Diffusion process

- A **diffusion process** can iteratively destruct all information in an image through a Markov chain.



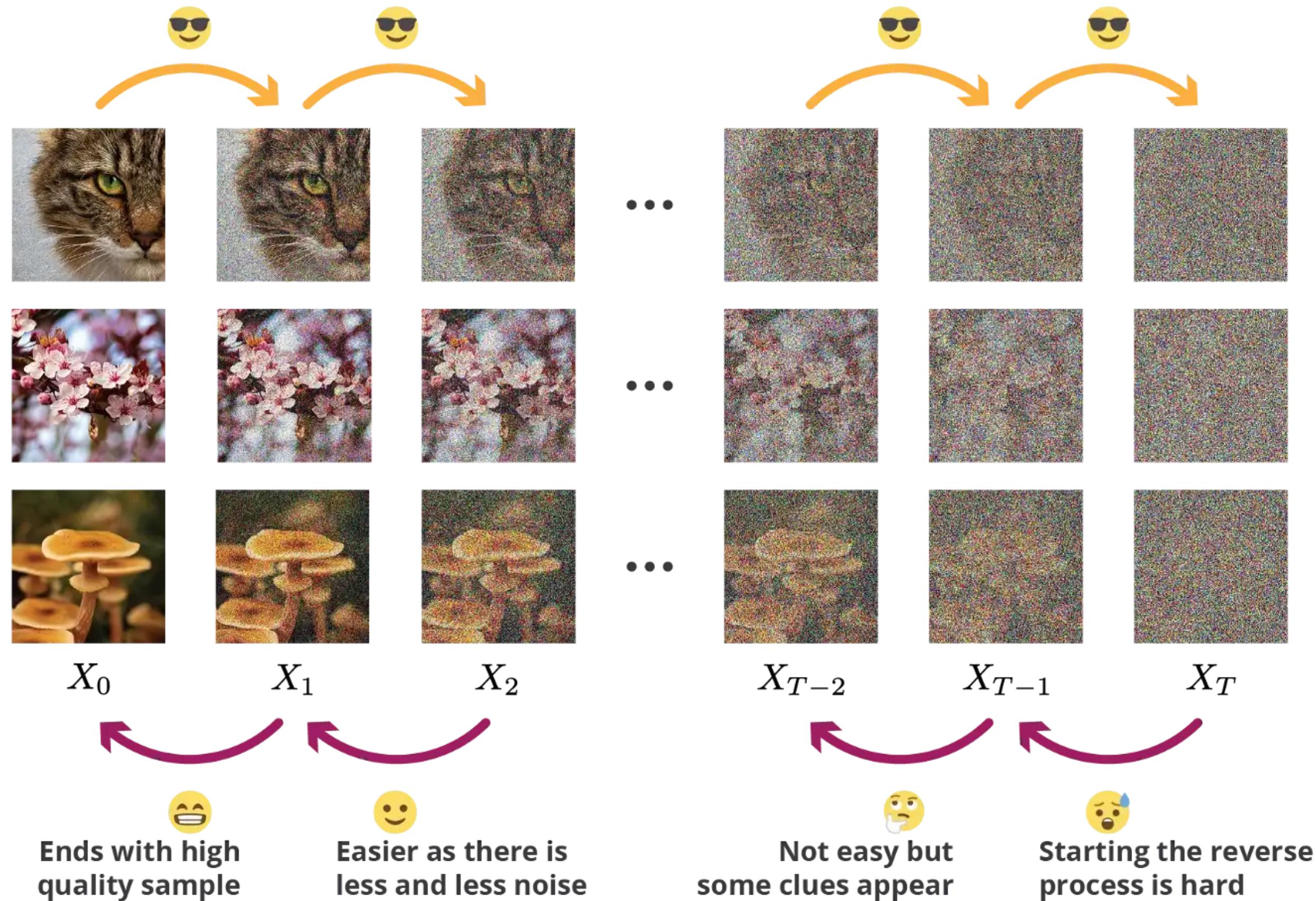
$$X_1 = \sqrt{1-p} X_0 + \sqrt{p} u_1 \sim \mathcal{N}(0, I)$$

The equation shows the mathematical formulation of the diffusion process. An image  $X_1$  is generated by adding a scaled noise term  $\sqrt{p} u_1$  to the original image  $X_0$ , where  $u_1 \sim \mathcal{N}(0, I)$ . The scaling factor  $\sqrt{1-p}$  indicates that the noise is added to the original image while maintaining its overall structure.

Source: <https://towardsdatascience.com/understanding-diffusion-probabilistic-models-dpms-1940329d6048>

# Reverse Diffusion process

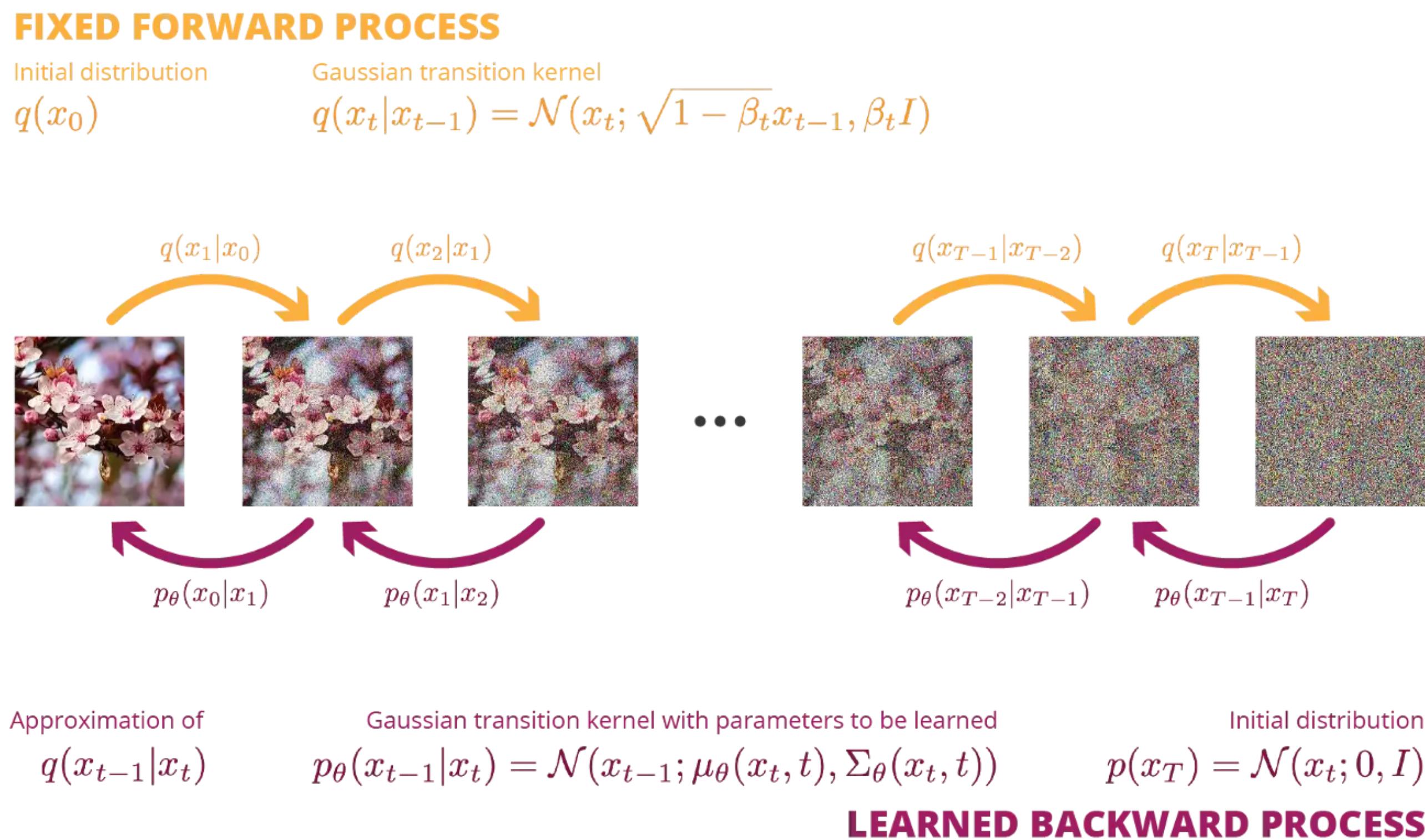
- It should be possible to **reverse** each diffusion step by removing the noise using a form of denoising autoencoder.



Source: <https://towardsdatascience.com/understanding-diffusion-probabilistic-models-dpms-1940329d6048>

# Reverse Diffusion process

- We will not get into details, but learning the reverse diffusion step implies Bayesian inference, KL divergence and so on.
- As we have the images at  $t$  and  $t + 1$ , it should be possible to learn, right?



Source: <https://towardsdatascience.com/understanding-diffusion-probabilistic-models-dpms-1940329d6048>

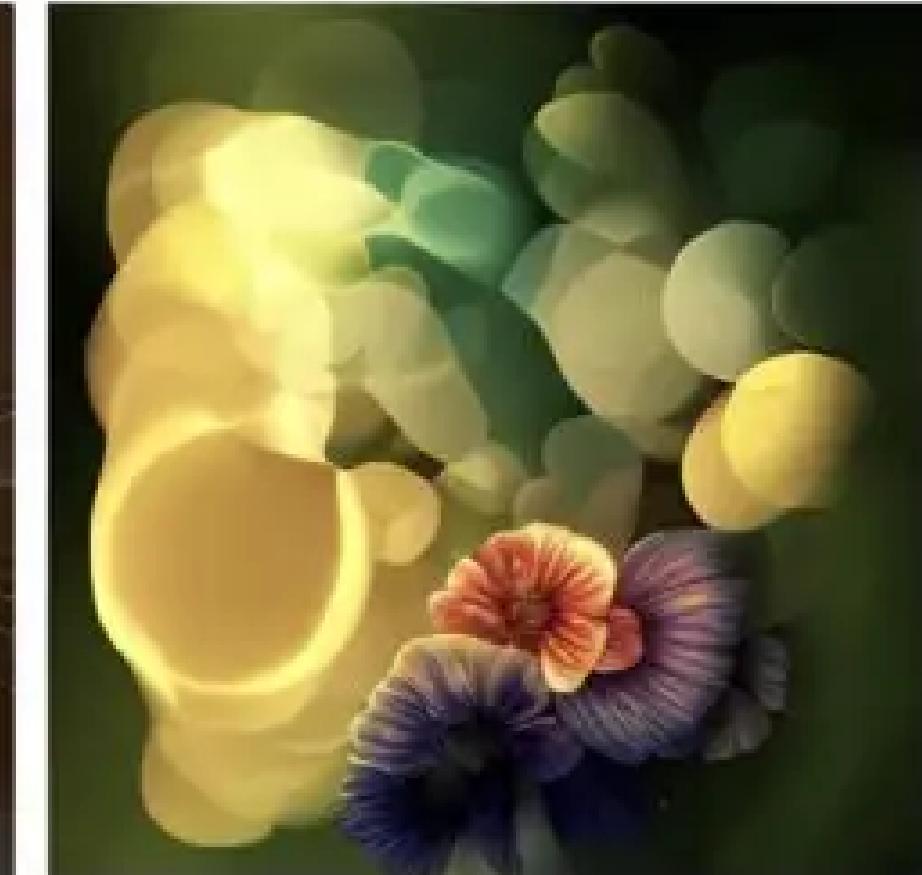
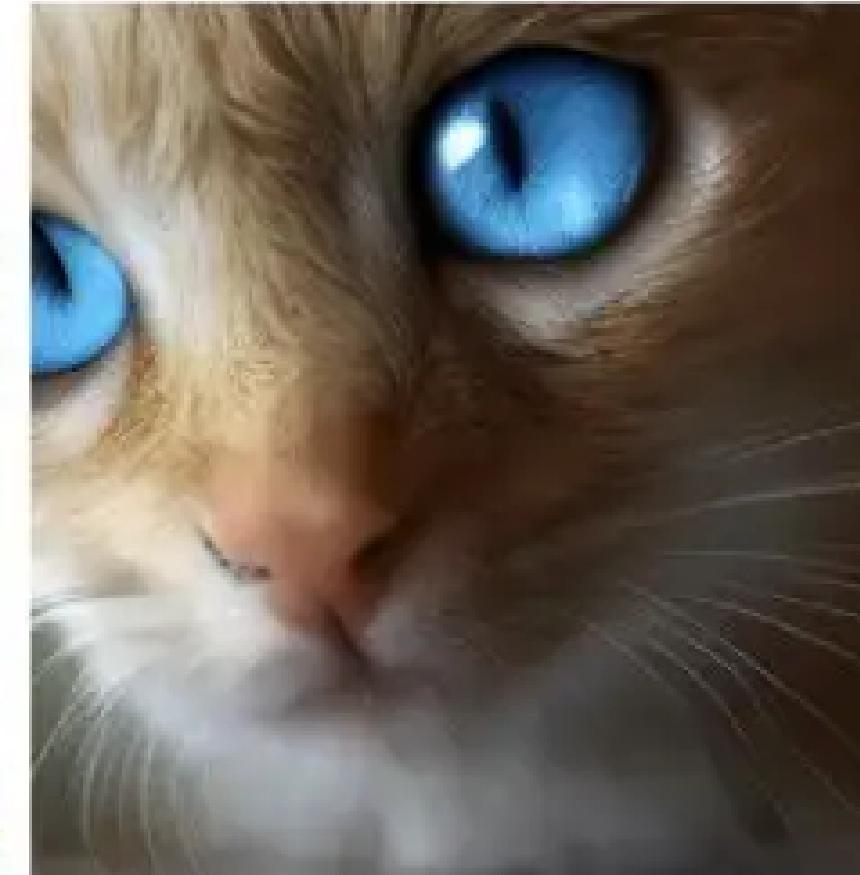
# Probabilistic diffusion models



Source: <http://adityaramesh.com/posts/dalle2/dalle2.html>

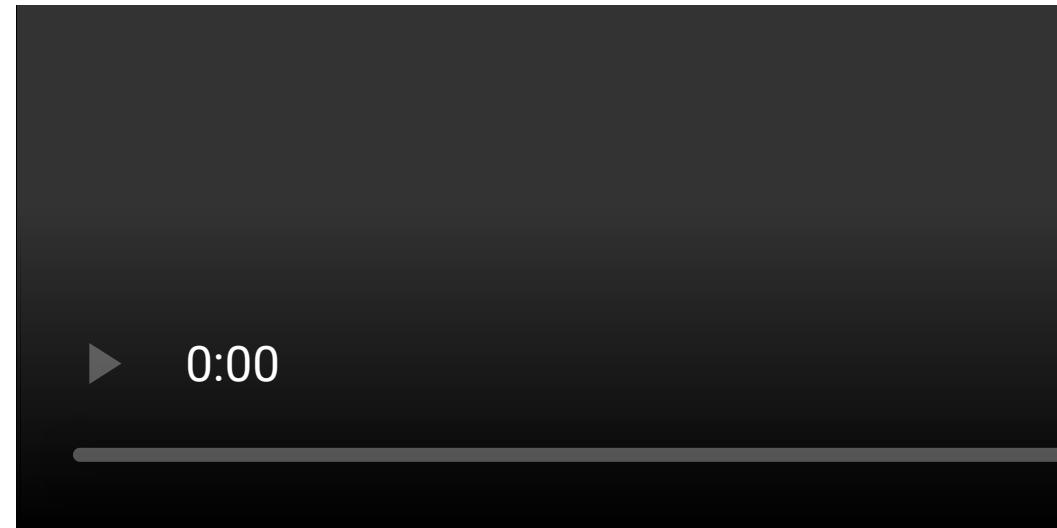
## **2 - Dall-e**

# Dall-e



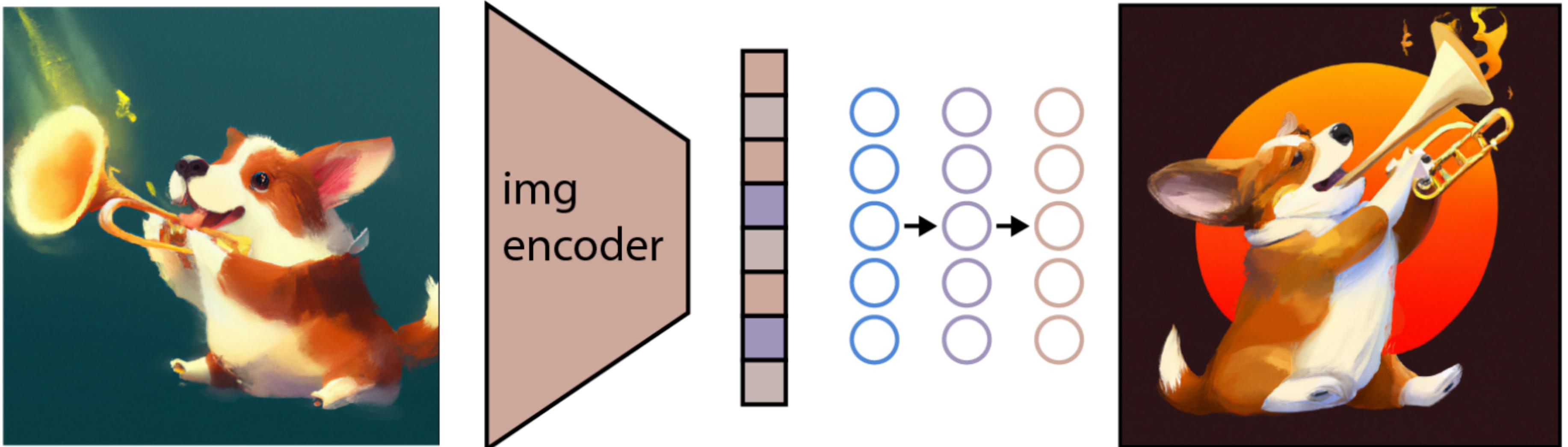
Source: <https://towardsdatascience.com/understanding-diffusion-probabilistic-models-dpms-1940329d6048>

# CLIP: Contrastive Language-Image Pre-training



# GLIDE

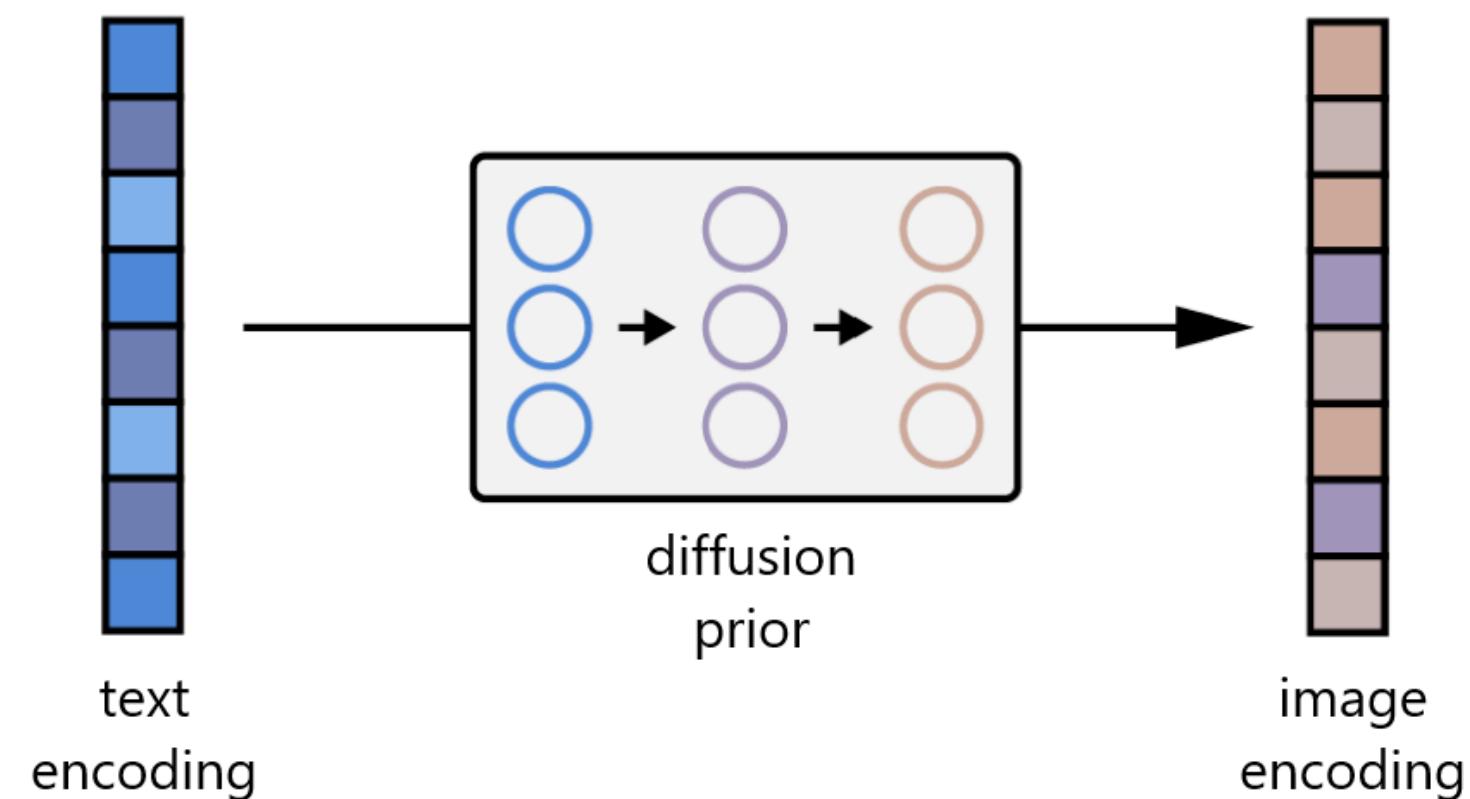
- GLIDE is a **reverse diffusion process** conditioned on the encoding of an image.



Source: <https://www.assemblyai.com/blog/how-dall-e-2-actually-works/>

# Dall-e

- A prior network learns to map text embeddings to image embeddings:



Source: <https://www.assemblyai.com/blog/how-dall-e-2-actually-works/>

- Complete Dall-e architecture:

