Neurocomputing

Diffusion Probabilistic Models

Julien Vitay
Professur für Künstliche Intelligenz - Fakultät für Informatik

1 - Denoising Diffusion probabilistic models

Denoising Diffusion Probabilistic Models

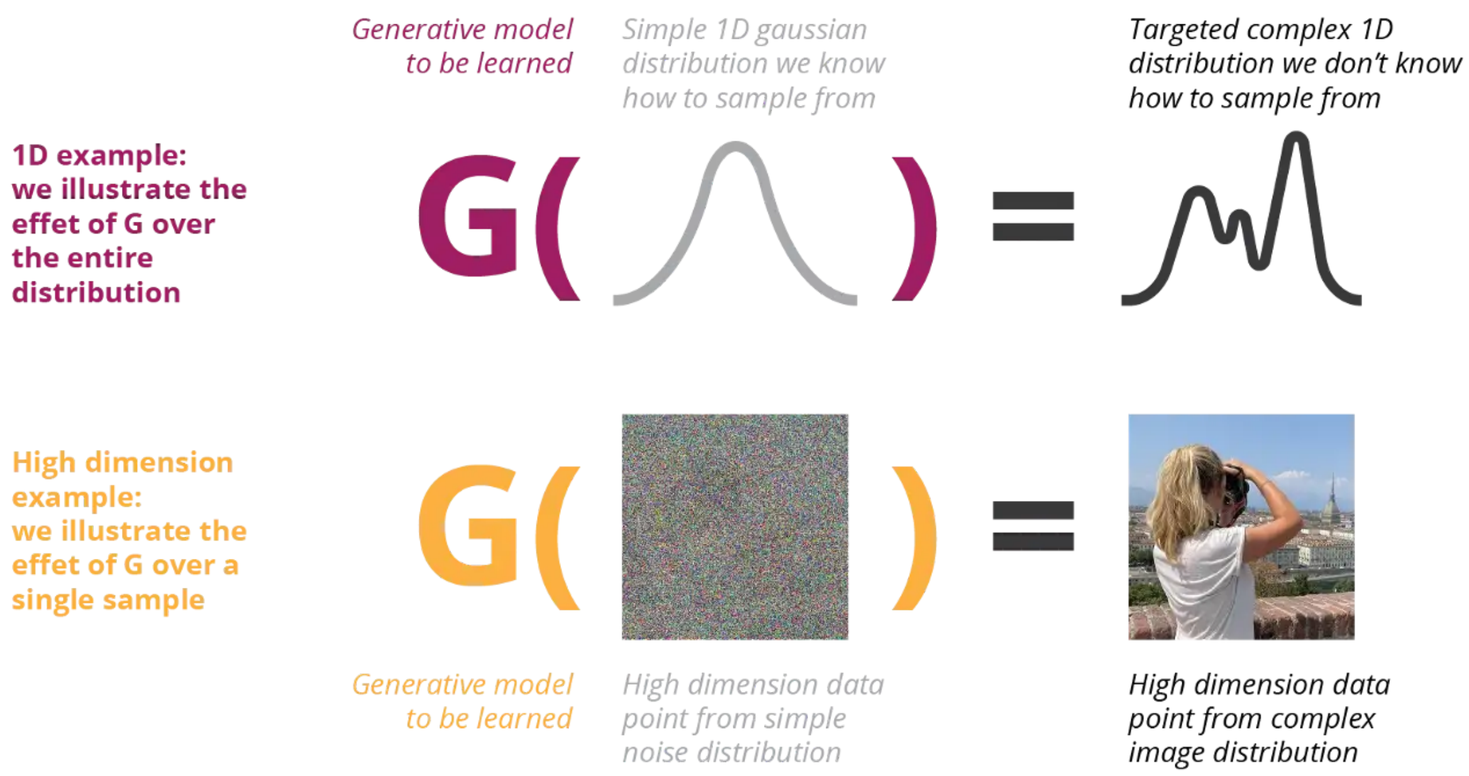
Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

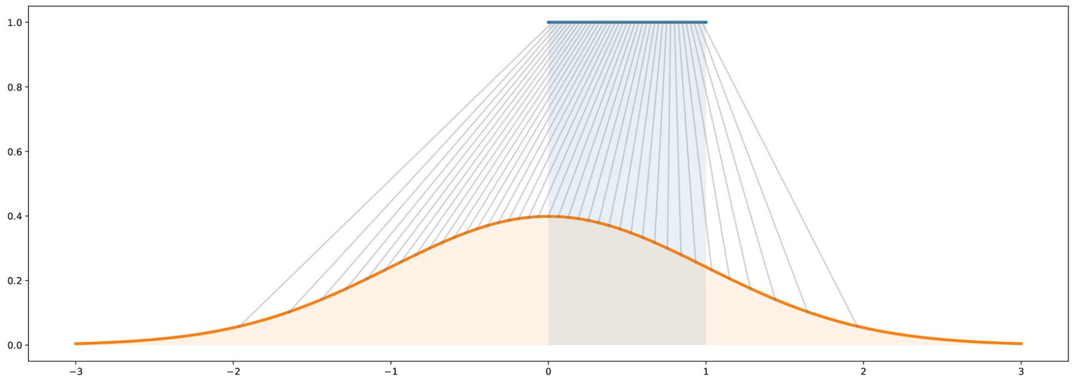
Generative modeling

- Generative modeling consists in transforming a simple probability distribution (e.g. Gaussian) into a more complex one (e.g. images).
- Learning this model allows to easily sample complex images.

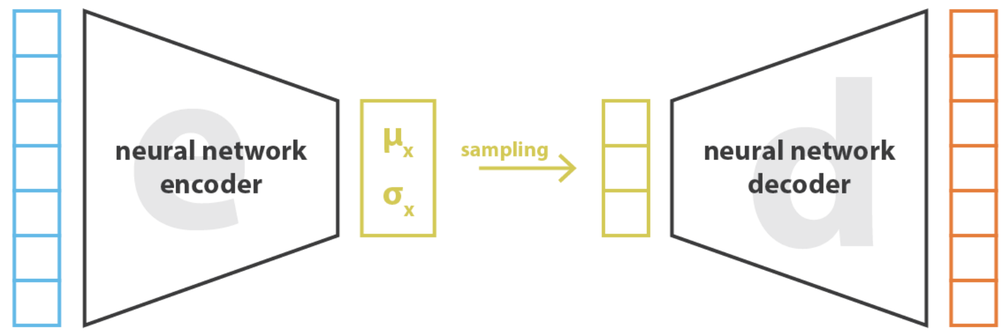


Source: https://towardsdatascience.com/understanding-diffusion-probabilistic-models-dpms-1940329d6048

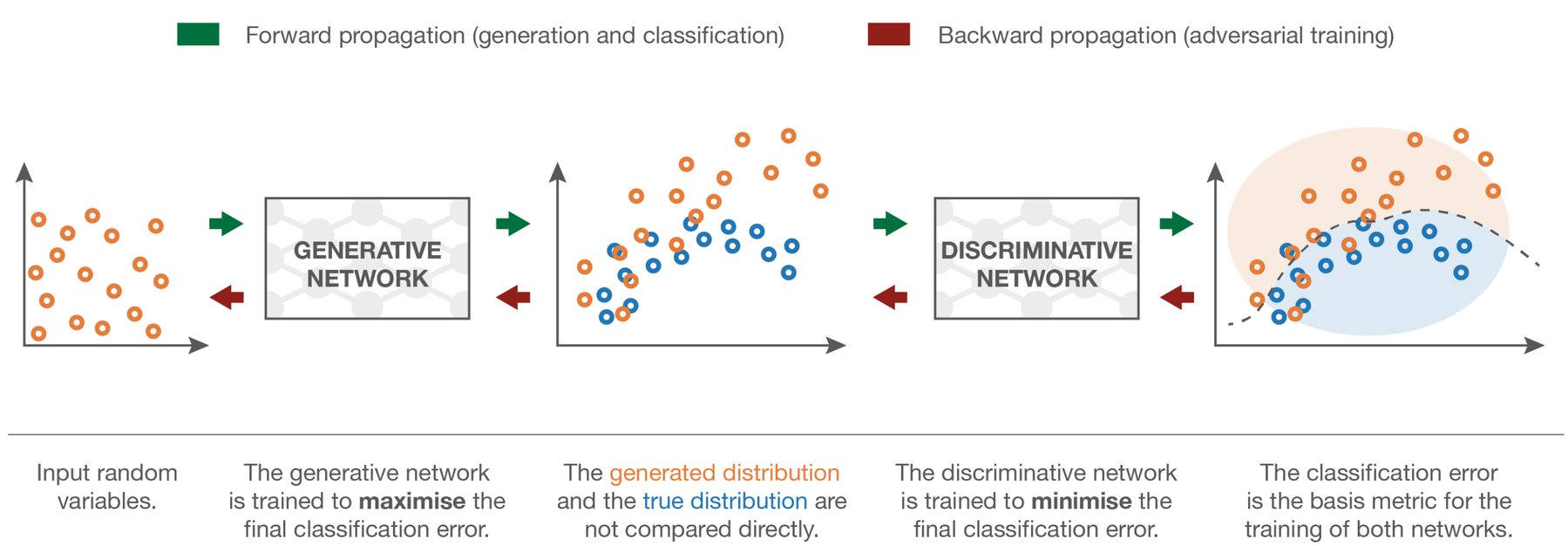
VAE and GAN transform simple noise into complex distributions



Source: https://towardsdatascience.com/understanding-generative-adversarial-networks-gans-cd6e4651a29



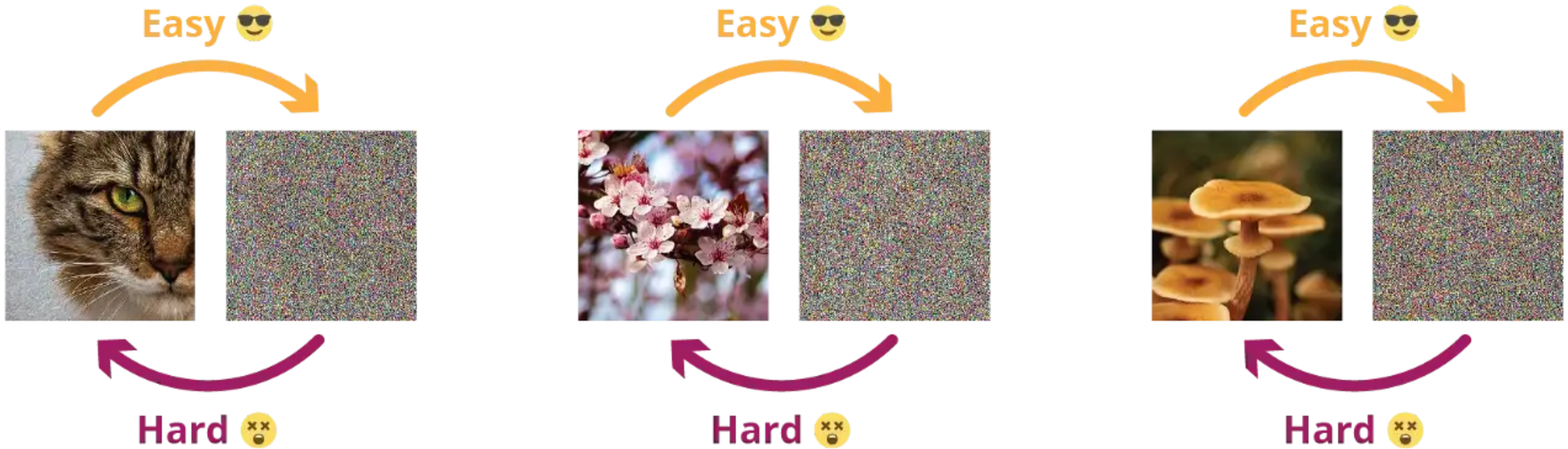
Source: https://ijdykeman.github.io/ml/2016/12/21/cvae.html



Source: https://towardsdatascience.com/understanding-generative-adversarial-networks-gans-cd6e4651a29

Destroying information is easier than creating it

- The task of the generators in GAN or VAE is very hard: going from noise to images in a few layers.
- The other direction is extremely easy.



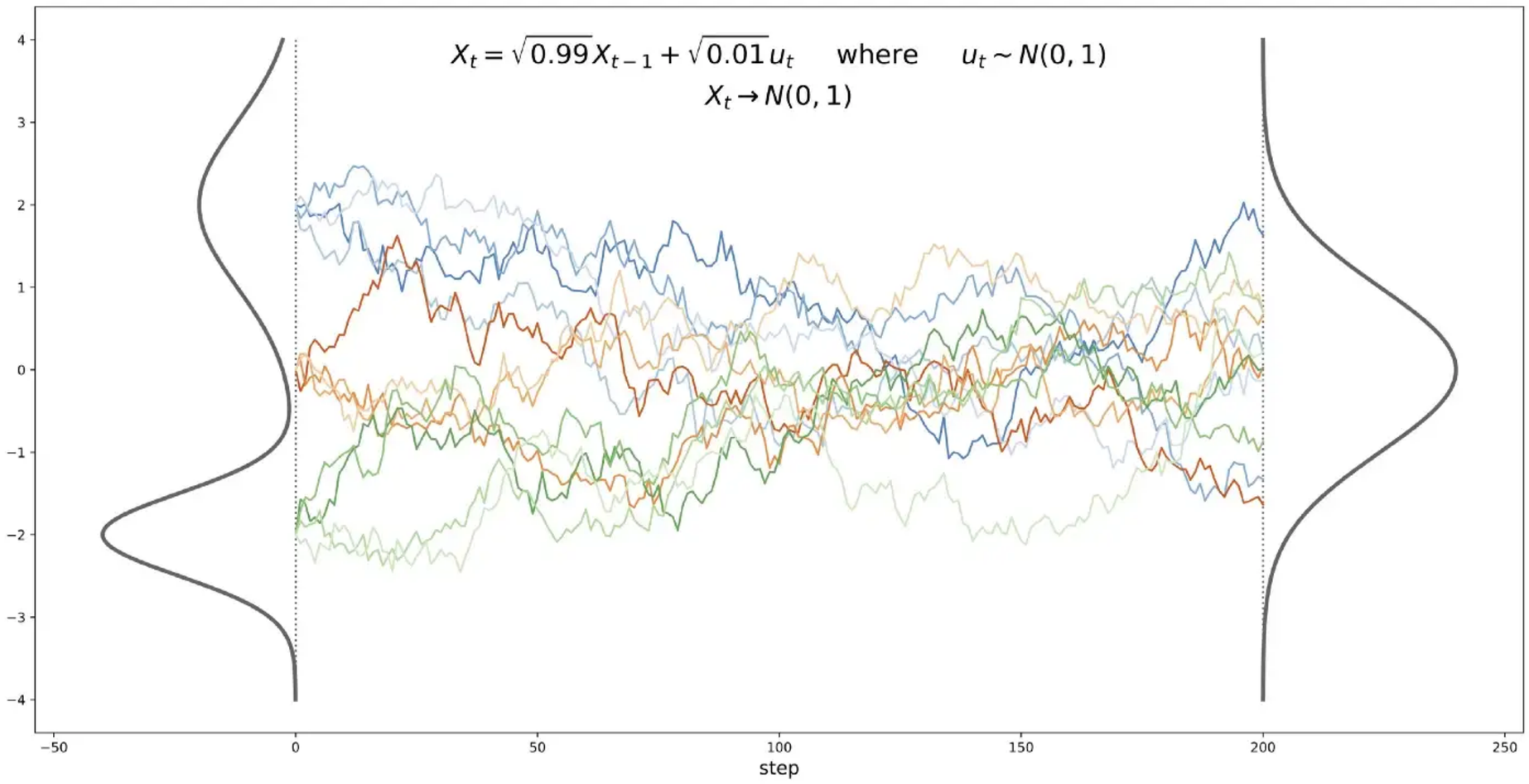
Source: https://towardsdatascience.com/understanding-diffusion-probabilistic-models-dpms-1940329d6048

Stochastic processes can destroy information

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

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

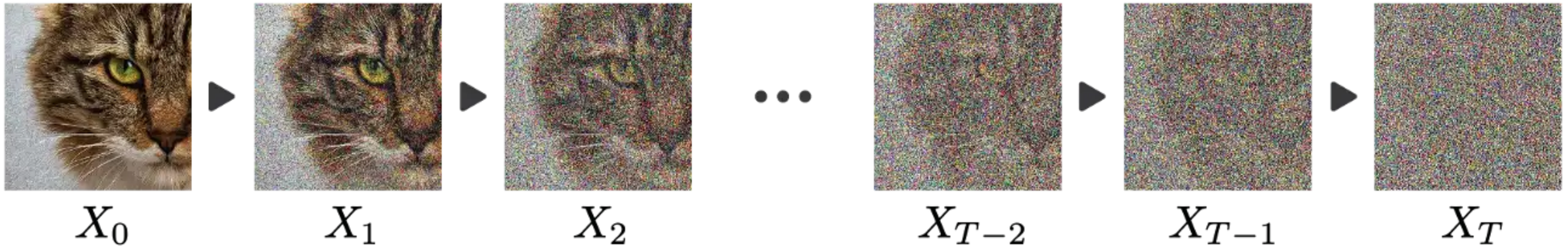
- Under some conditions, any probability distribution converges to a normal distribution.



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.
- A Markov chain implies that each step is independent and governed by a probability distribution $p(X_t|X_{t-1})$.



$X_0 \quad X_1 \quad X_2 \quad \cdots \quad X_{T-2} \quad X_{T-1} \quad X_T$

X_1

$= \sqrt{1-p}$

X_0

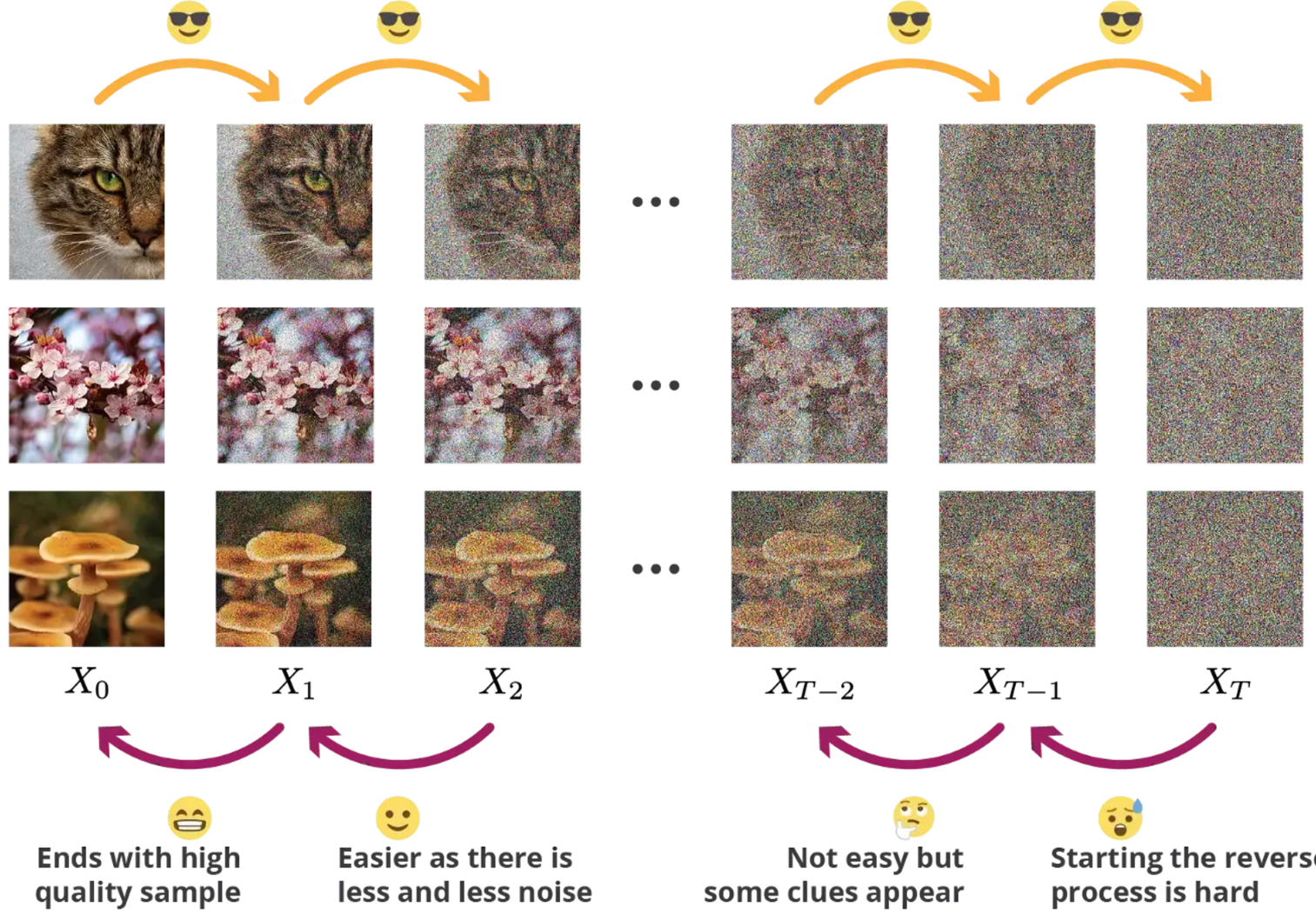
$+ \sqrt{p}$

$u_1 \sim \mathcal{N}(0, I)$

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

Probabilistic diffusion models

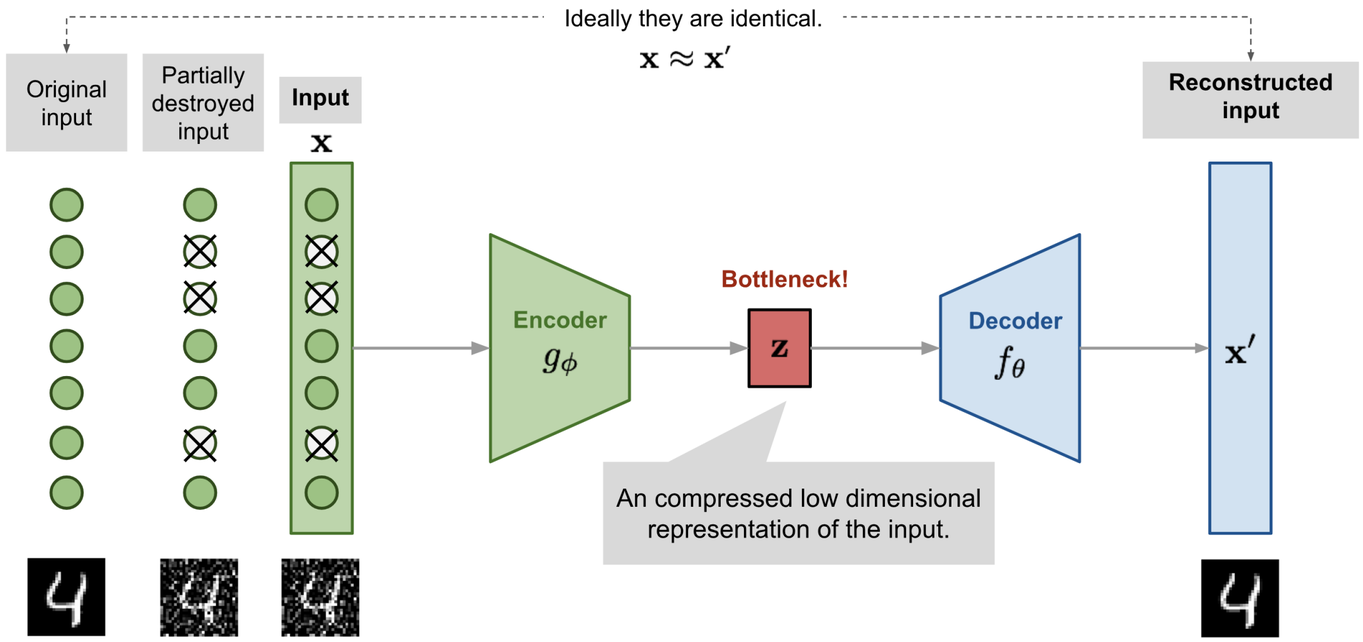
- It should be possible to **reverse** each diffusion step by removing the noise using a form of denoising autoencoder.

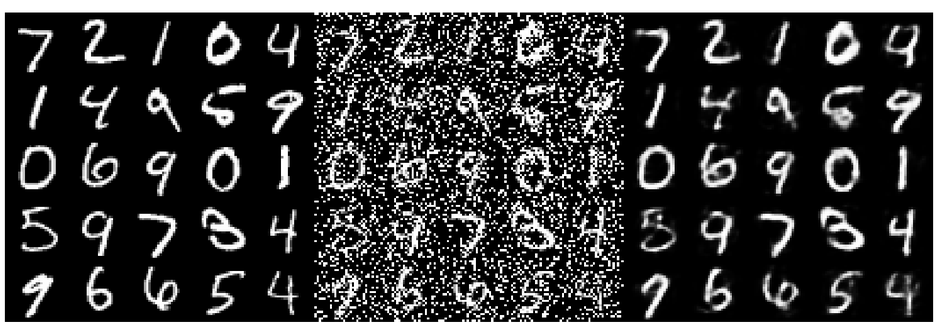


Source: https://towardsdatascience.com/understanding-diffusion-probabilistic-models-dpms-1940329d6048

Reminder: Denoising autoencoder

- A **denoising autoencoder** (DAE) is trained with noisy inputs but perfect desired outputs. It learns to suppress that noise.

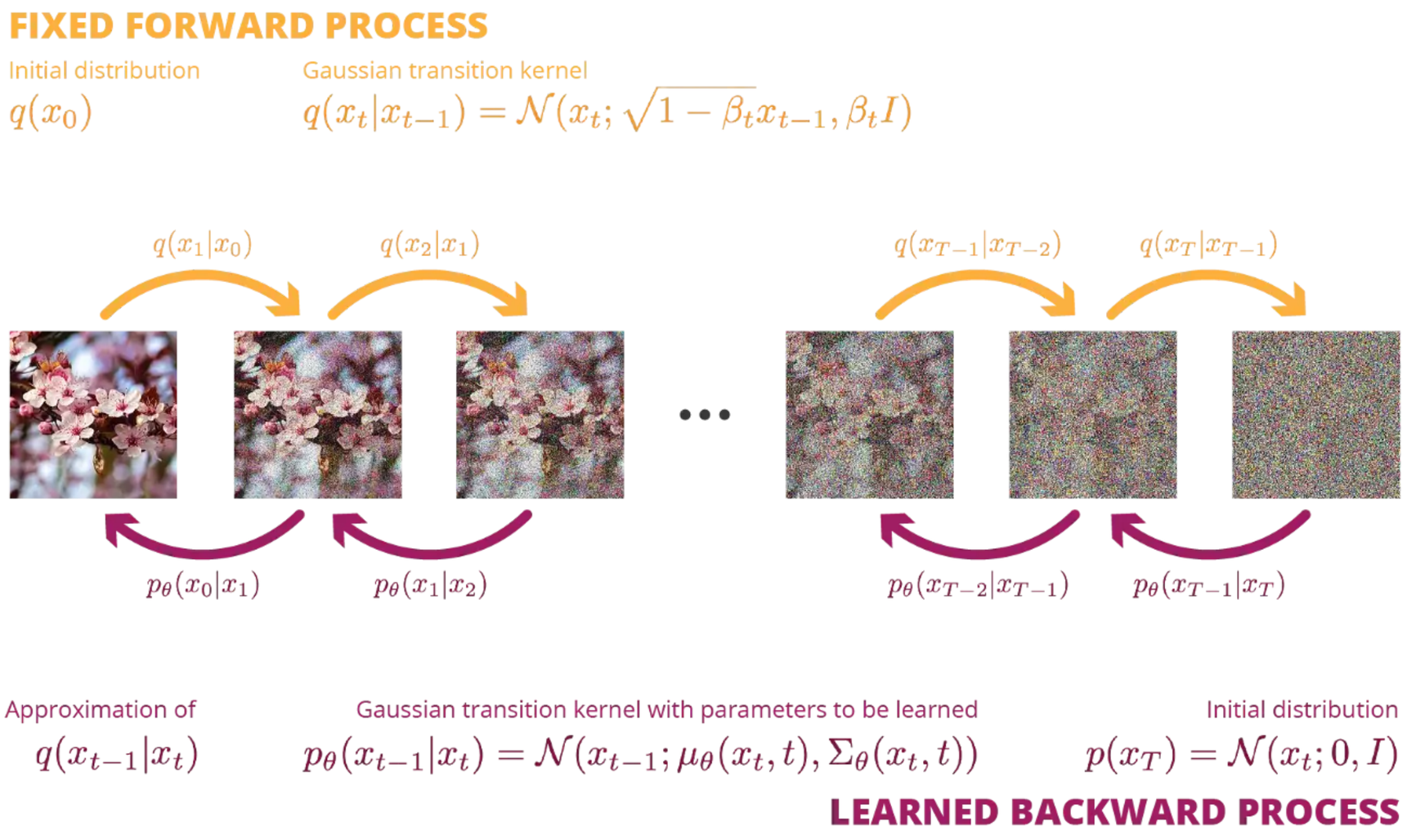




Source : https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

Forward Diffusion process

- The forward process iteratively corrupts the image using $q(x_t|x_{t-1})$ for T steps (e.g. $T = 1000$).
- The goal is to learn a reverse model $p_\theta(x_{t-1}|x_t)$ that approximates the true $q(x_{t-1}|x_t)$.



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

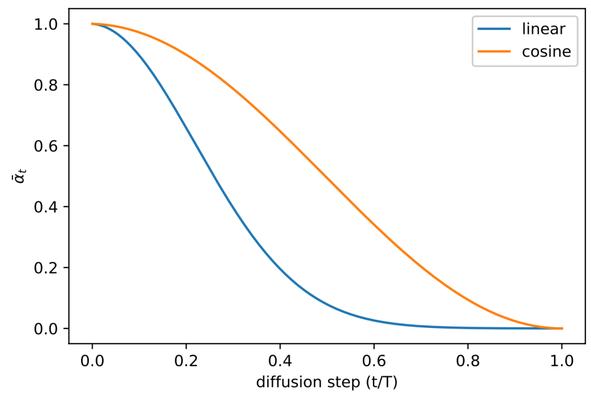
Forward Diffusion process

- The forward diffusion process iteratively adds Gaussian noise with a fixed schedule β_t:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I)$$

$$x_t = \sqrt{1-\beta_t}\, x_t + \beta_t \epsilon \quad \text{where } \epsilon \sim \mathcal{N}(0, I)$$

- $\mu_t = \sqrt{1-\beta_t}\, x_t$ is the mean of the distribution, $\sigma_t = \beta_t I$ its variance.



Source: Nichol and Dhariwal (2021) Improved Denoising Diffusion Probabilistic Models. arXiv:2102.09672

- The parameter β_t is annealed with a decreasing schedule, as adding more noise at the end does not destroy much information.
- Note that each image x_t is also a Gaussian noisy version of the original image x_0:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{1-\bar{\alpha}_t}\, x_0, \bar{\alpha}_t I)$$

$$x_t = \sqrt{1-\bar{\alpha}_t}\, x_0 + \bar{\alpha}_t\, \epsilon_t \quad \text{where } \epsilon_t \sim \mathcal{N}(0, I)$$

with $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ only depending on the history of β_t.

- Given the original image x_0 and a noisy version x_t, we can find the noise ϵ_t that was added.

Probabilistic diffusion models

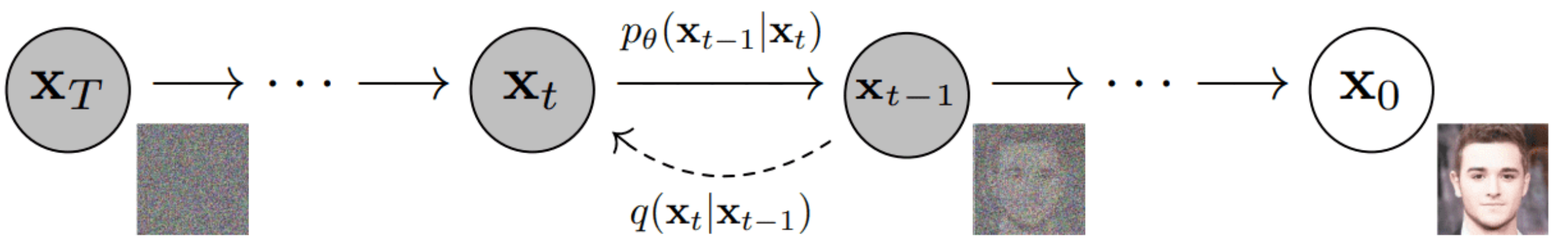
- The goal of the reverse diffusion process is to find a parameterized model p_θ explaining the sequence of images backwards in time:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

where:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

- The reverse process is also normally distributed, given that the noise β_t is not too big.



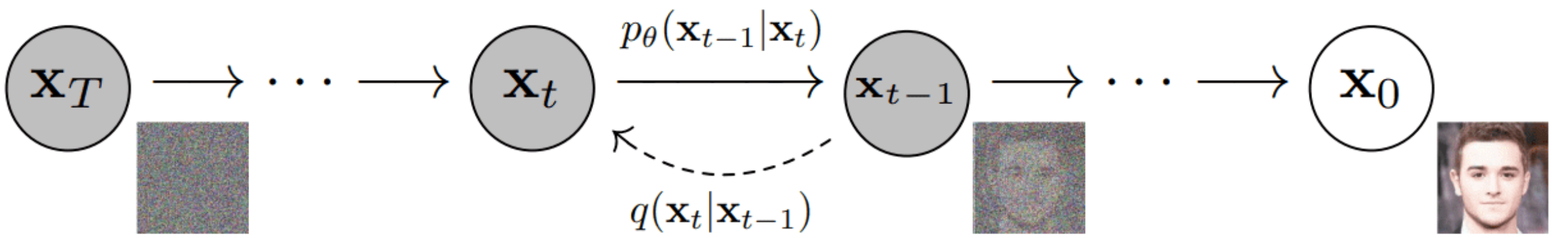
Source: Ho et al. (2020) Denoising Diffusion Probabilistic Models arXiv:2006.11239

Probabilistic diffusion models

- By doing some Bayesian inference on the true posterior $q(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_t, \sigma_t)$, (Ho et al., 2020) could show that:

$$\begin{cases} \mu_t = \frac{1}{\sqrt{\alpha_t}} (x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_t) \\ \\ \sigma_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t \, I = \bar{\beta}_t \, I \end{cases}$$

- The reverse process is also normally distributed, provided the forward noise β_t was not too big.
- The reverse variance only depends on the schedule of β_t, it can be pre-computed.



Source: Ho et al. (2020) Denoising Diffusion Probabilistic Models arXiv:2006.11239

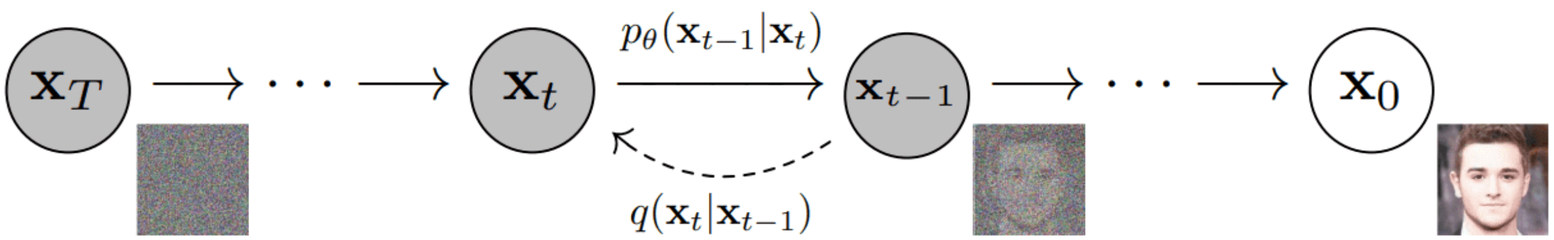
Probabilistic diffusion models

- The reverse model $p_\theta(x_{t-1}|x_t)$ only need to approximate the mean:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t)\right)$$

- x_t is an input to the model, it does not have to predicted.
- All we need to learn is the noise $\epsilon_\theta(x_t, t) \approx \epsilon_t$ that was added to the original image x_0 to obtain x_t:

$$x_t = \sqrt{1-\bar{\alpha}_t}\, x_0 + \bar{\alpha}_t\, \epsilon_t$$



Source: Ho et al. (2020) Denoising Diffusion Probabilistic Models arXiv:2006.11239

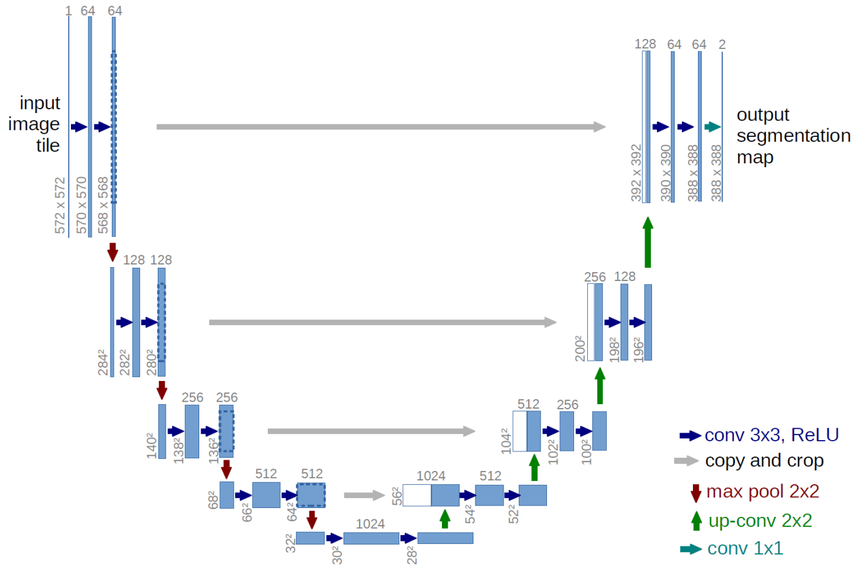
Probabilistic diffusion models

- We want to predict the added noise in the image space:

$$\epsilon_\theta(x_t, t) = \epsilon_\theta(\sqrt{1-\bar{\alpha}_t}\, x_0 + \bar{\alpha}_t\, \epsilon_t, t) \approx \epsilon_t$$

- We can simply minimize the mse with the true noise:

$$\begin{aligned}\mathcal{L}(\theta) &= \mathbb{E}_{t\sim[1,T],x_0,\epsilon_t}\left[(\epsilon_t - \epsilon_\theta(x_t, t))^2\right] \\ &= \mathbb{E}_{t\sim[1,T],x_0,\epsilon_t}\left[(\epsilon_t - \epsilon_\theta(\sqrt{1-\bar{\alpha}_t}\, x_0 + \bar{\alpha}_t\, \epsilon_t, t))^2\right]\end{aligned}$$



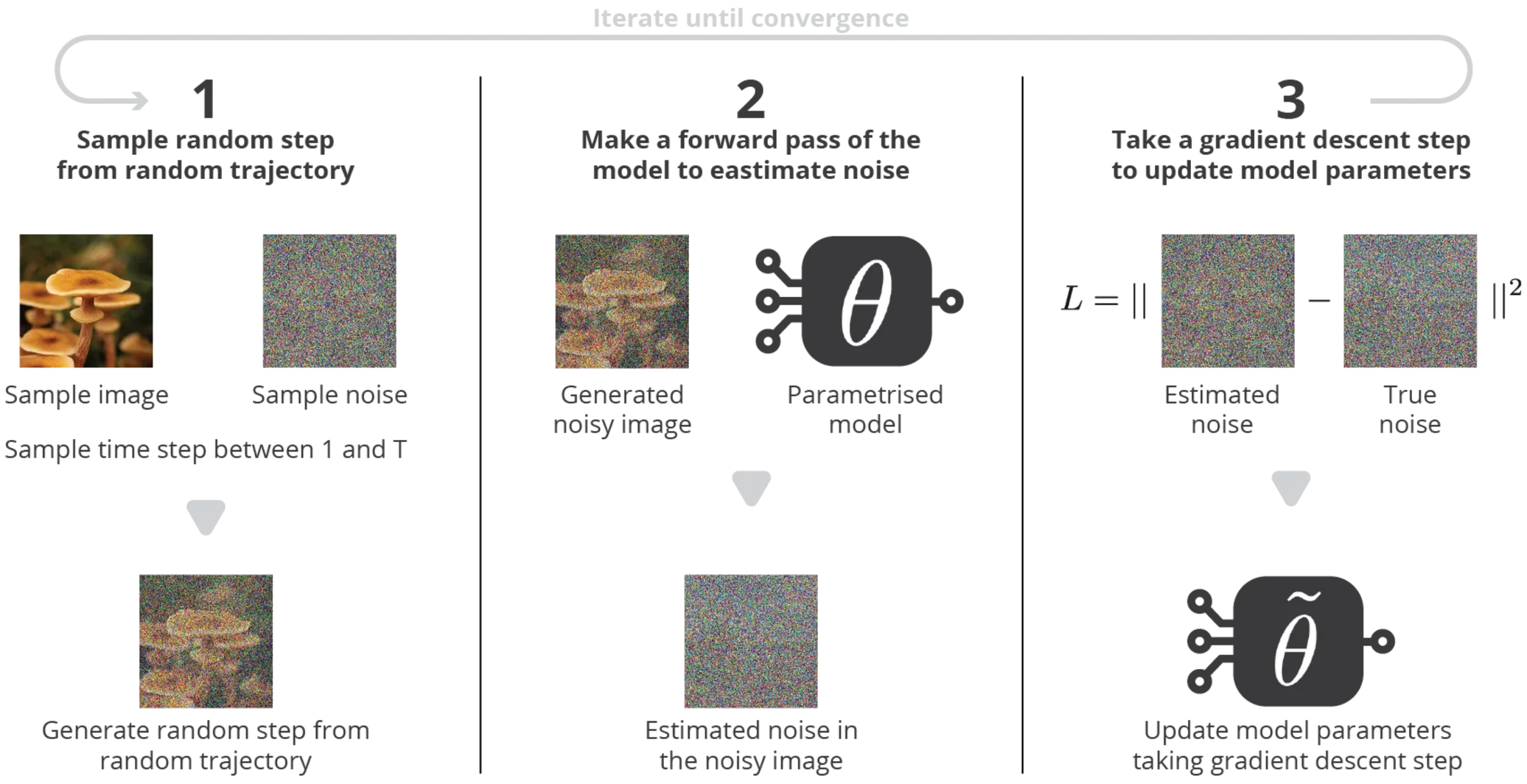
Source: Ronneberger et al. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.04597

- We only need to sample an image x_0, a time step t, a noise $\epsilon_t \sim \mathcal{N}(0, I)$, predict the noise $\epsilon_\theta(x_t, t)$ and minimize the mse!

- The neural network used for the reverse diffusion is usually some kind of U-net, with attentional layers, or even a vision Transformer.

Probabilistic diffusion models

- Training can be done on individual samples, no need for the whole Markov chain to create the minibatches.

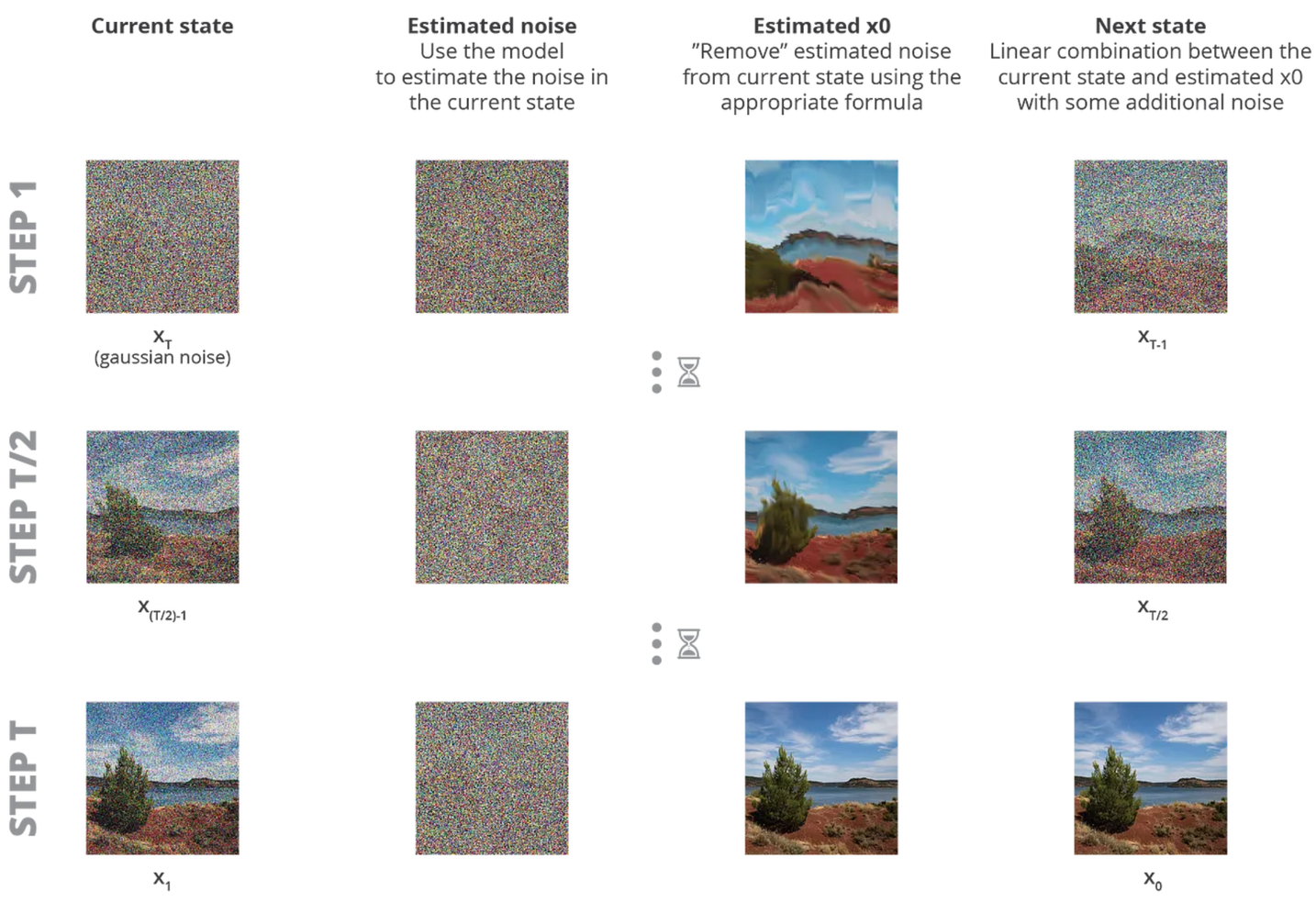


Source: https://towardsdatascience.com/understanding-diffusion-probabilistic-models-dpms-1940329d6048

Probabilistic diffusion models

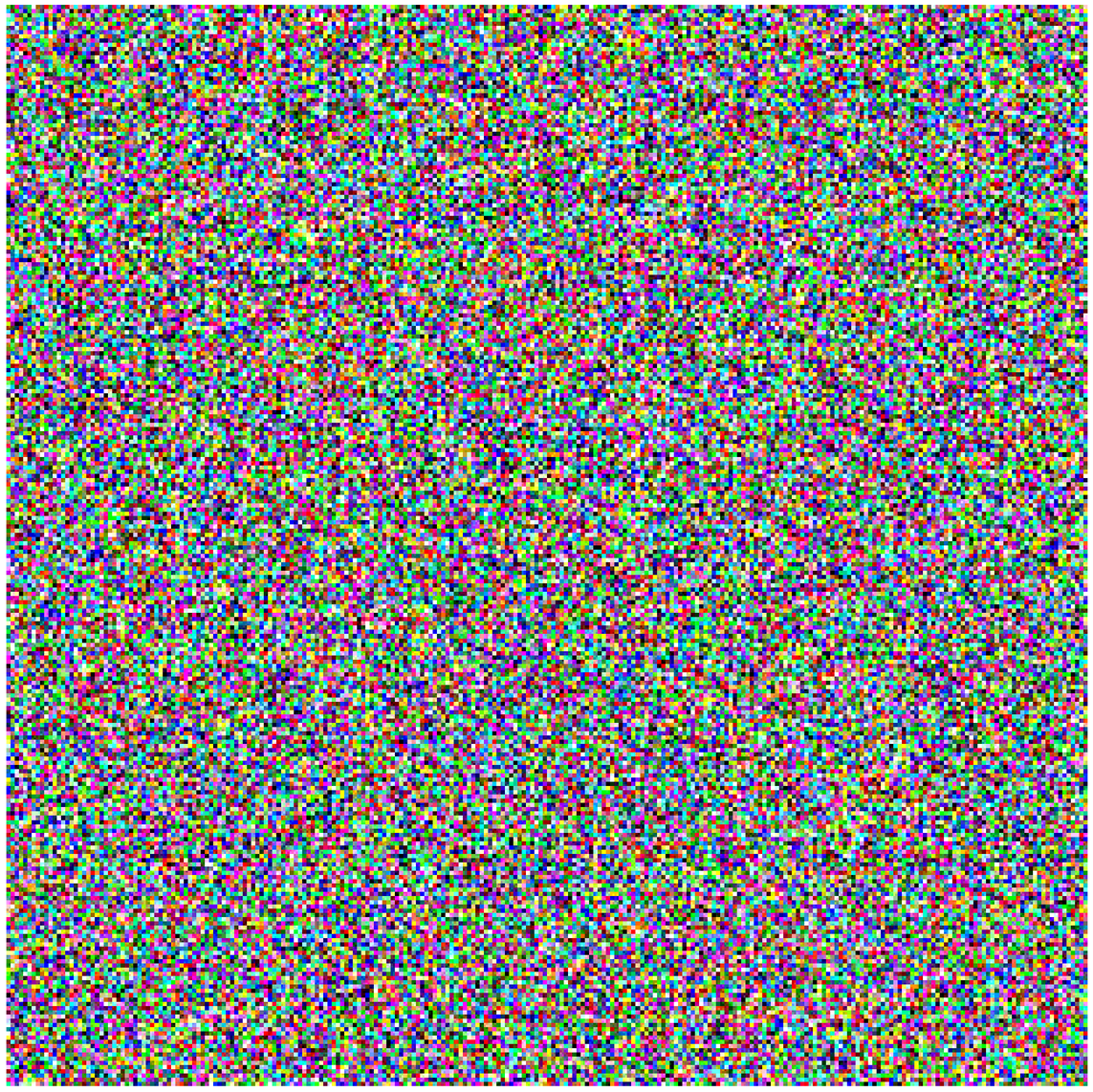
- The reverse diffusion occurs iteratively backwards in time:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t \, z$$



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

Probabilistic diffusion models



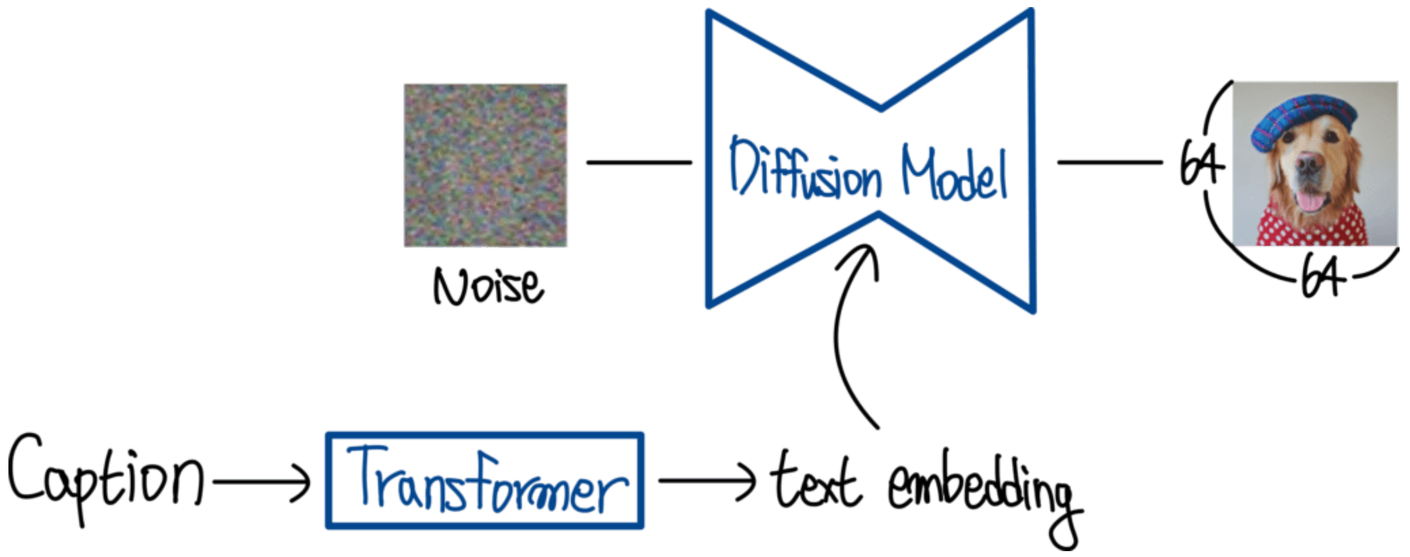
Source: http://adityaramesh.com/posts/dalle2/dalle2.html

GLIDE

- PDMs generate images from raw noise, but there is no control over which image will emerge.
- **GLIDE** (Guided Language to Image Diffusion for Generation and Editing) is a PDM conditioned on a latent representation of a caption c.
- As for cGAN and cVAE, the caption c is provided to the learned model:

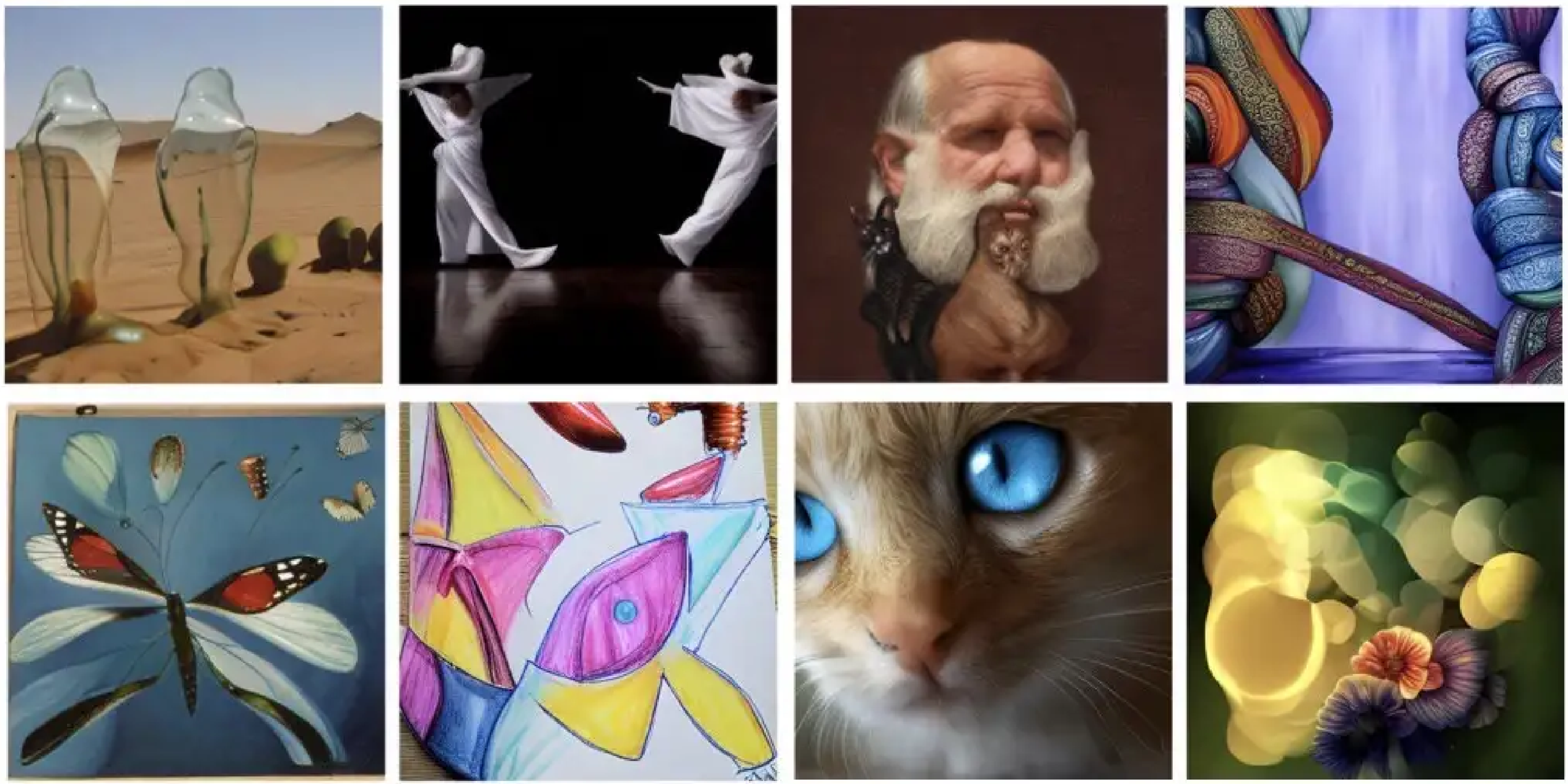
$$\epsilon_\theta(x_t, t, c) \approx \epsilon_t$$

- Text embeddings can be obtained from any NLP model, for example a Transformer.



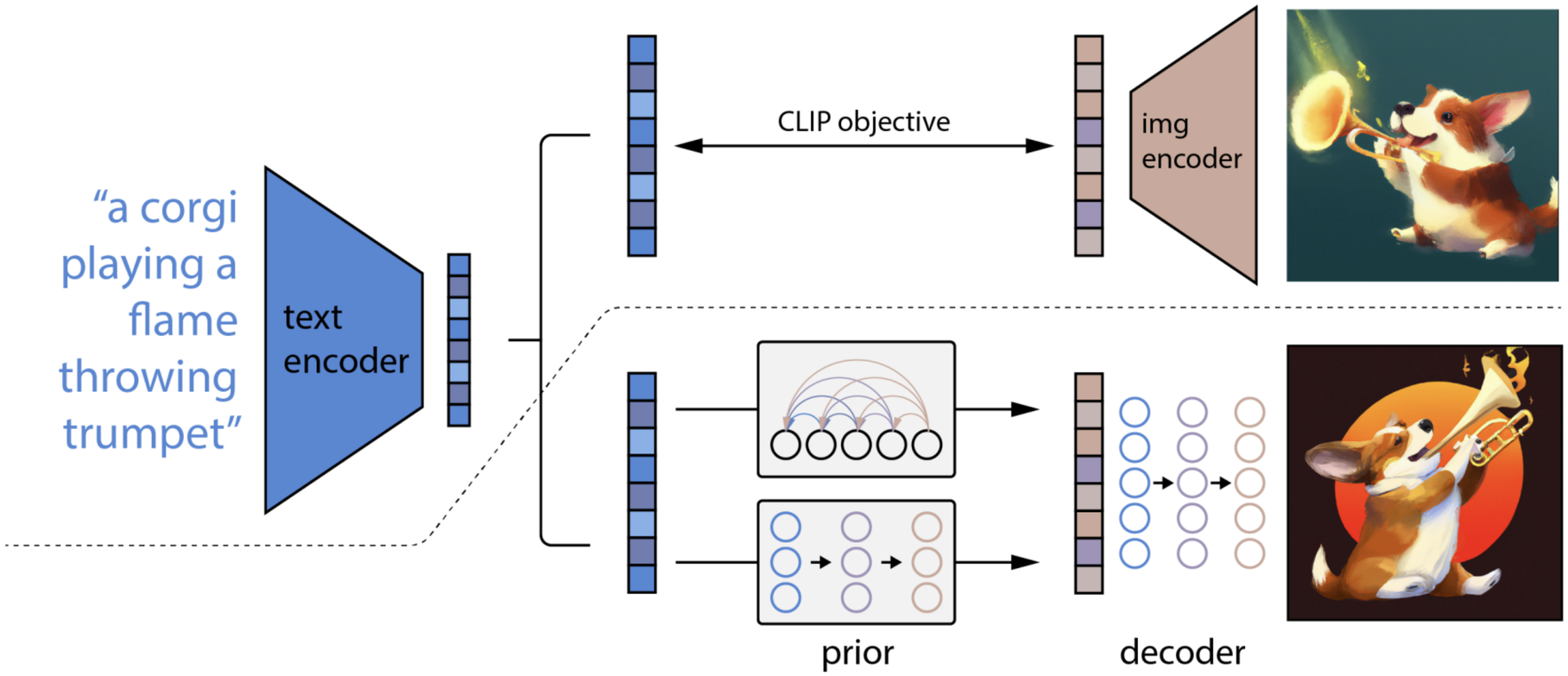
Source: https://ffighting.net/deep-learning-paper-review/diffusion-model/glide/

2 - Dall-e 2



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

Dall-e 2

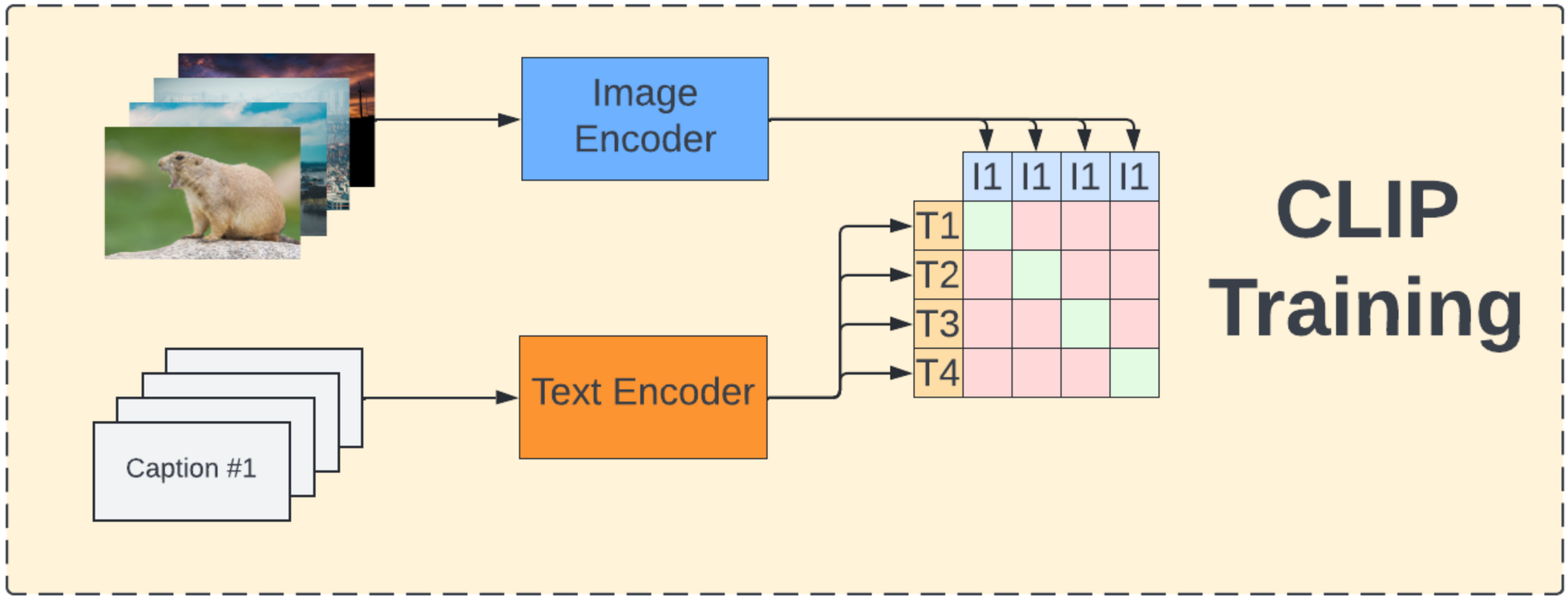


Source: Ramesh et al. (2022)

- CLIP embeddings are first learned using contrastive learning.
- A conditional diffusion process (GLIDE) uses the image embeddings to produce images.
- Dall-e 3, Midjourney, Stable Diffusion, etc., work on similar principles.

CLIP: Contrastive Language-Image Pre-training

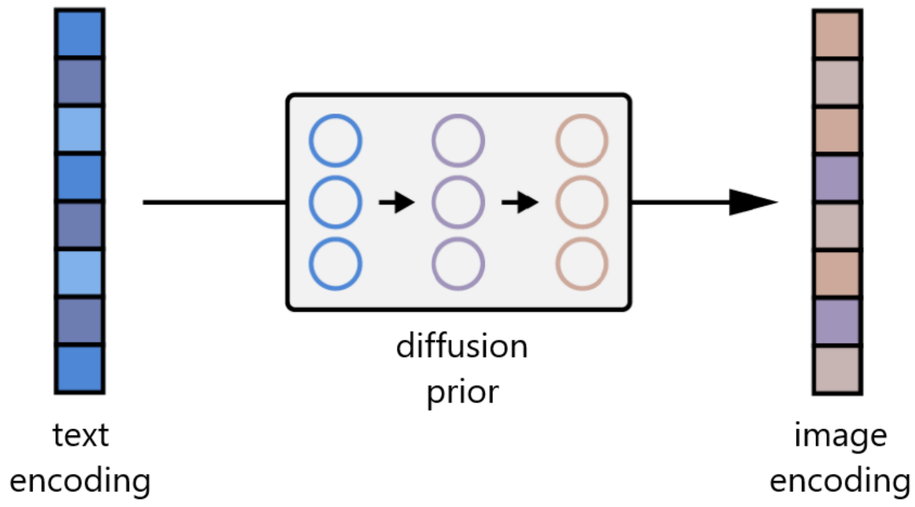
- Embeddings for text and images are learned using **Transformer encoders** and **contrastive learning**.
- For each pair (text, image) in the training set, their representation should be made similar, while being different from the others.



Source: https://towardsdatascience.com/understanding-how-dall-e-mini-works-114048912b3b

Dall-e 2

- The prior network learns to map text embeddings to a sequence of image embeddings:



- After CLIP training, the two embeddings are already close from each other, but the authors find that the diffusion process works better when the image embeddings change during the diffusion.

References

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising Diffusion Probabilistic Models. doi:10.48550/arXiv.2006.11239.

Nichol, A., and Dhariwal, P. (2021). Improved Denoising Diffusion Probabilistic Models. doi:10.48550/arXiv.2102.09672.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., et al. (2022). GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. doi:10.48550/arXiv.2112.10741.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical Text-Conditional Image Generation with CLIP Latents. doi:10.48550/arXiv.2204.06125.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. http://arxiv.org/abs/1505.04597.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *J. Mach. Learn. Res.* 11, 3371–3408.