

# Deep Learning: Denoising Diffusion Probabilistic Models

**Ozan Özdenizci**

Institute of Theoretical Computer Science

[ozan.ozdenizci@igi.tugraz.at](mailto:ozan.ozdenizci@igi.tugraz.at)

Deep Learning VO - WS 23/24  
Lecture 11 - January 8th, 2024

# Today

---

- Deep Generative Models
  - Variational Autoencoders
  - Generative Adversarial Networks
- Denoising Diffusion Probabilistic Models
- Idea & Training Objective
- Sampling
- Further Extensions & Applications

# Today's lecture is based on...

---

## 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

---

International Conference on Machine Learning (ICML) 2015

---

## 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

---

Advances in Neural Information Processing Systems (NeurIPS) 2020

---

## Diffusion Models Beat GANs on Image Synthesis

---

Prafulla Dhariwal\*  
OpenAI  
prafulla@openai.com

Alex Nichol\*  
OpenAI  
alex@openai.com

---

Advances in Neural Information Processing Systems (NeurIPS) 2021

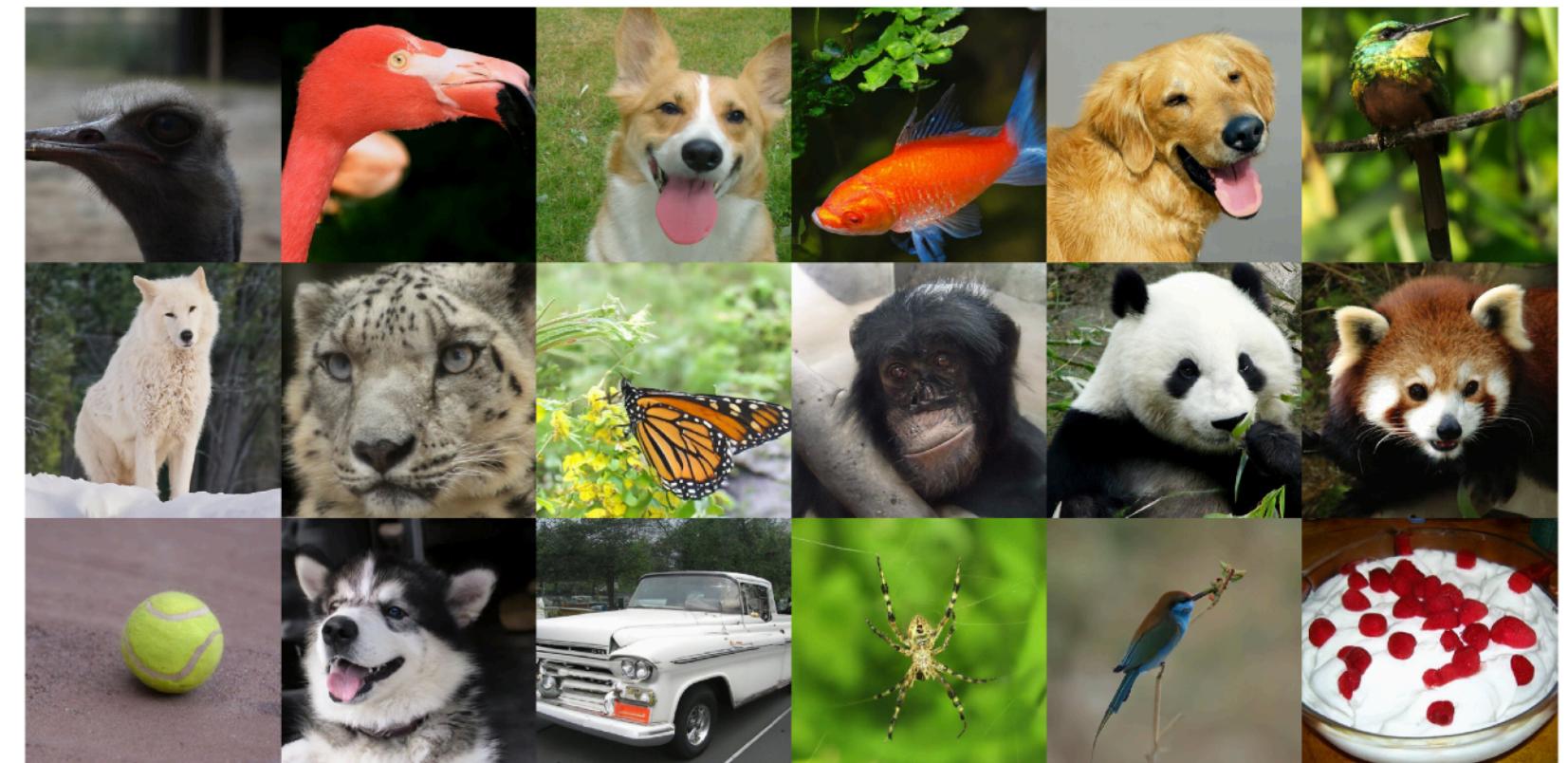
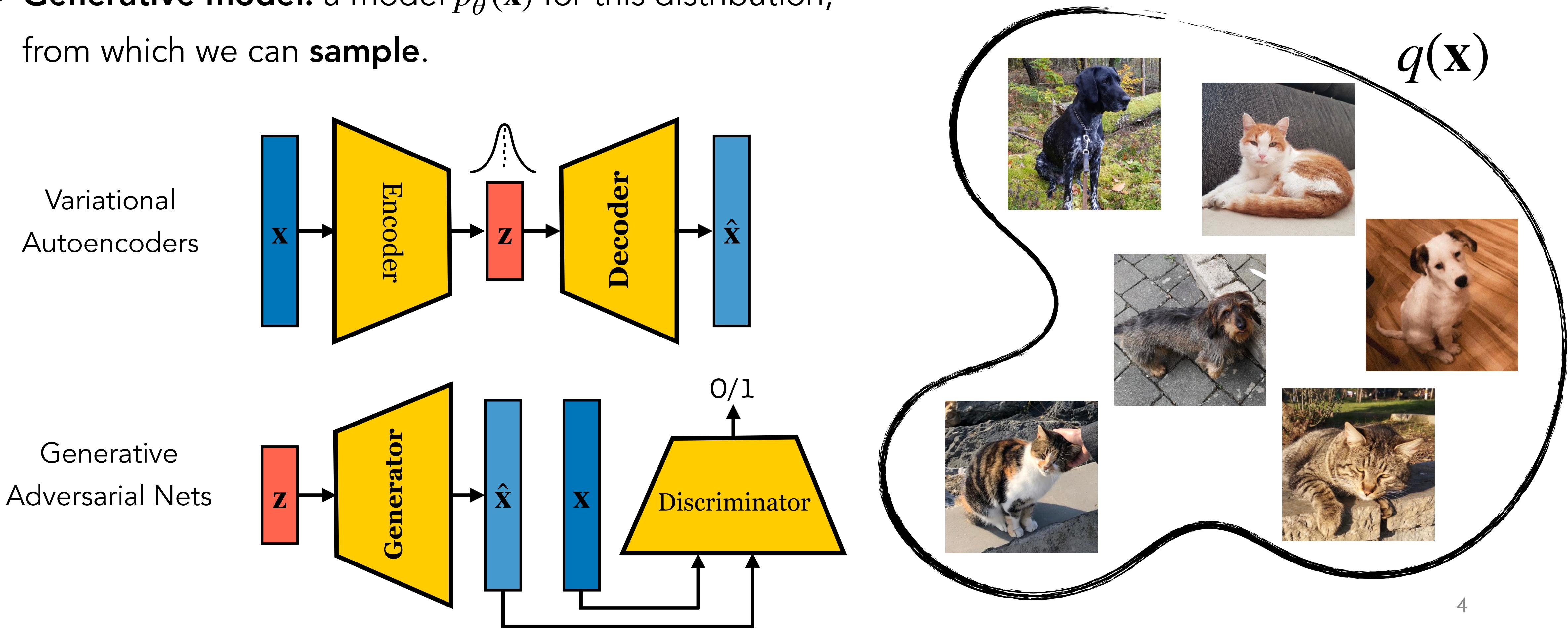


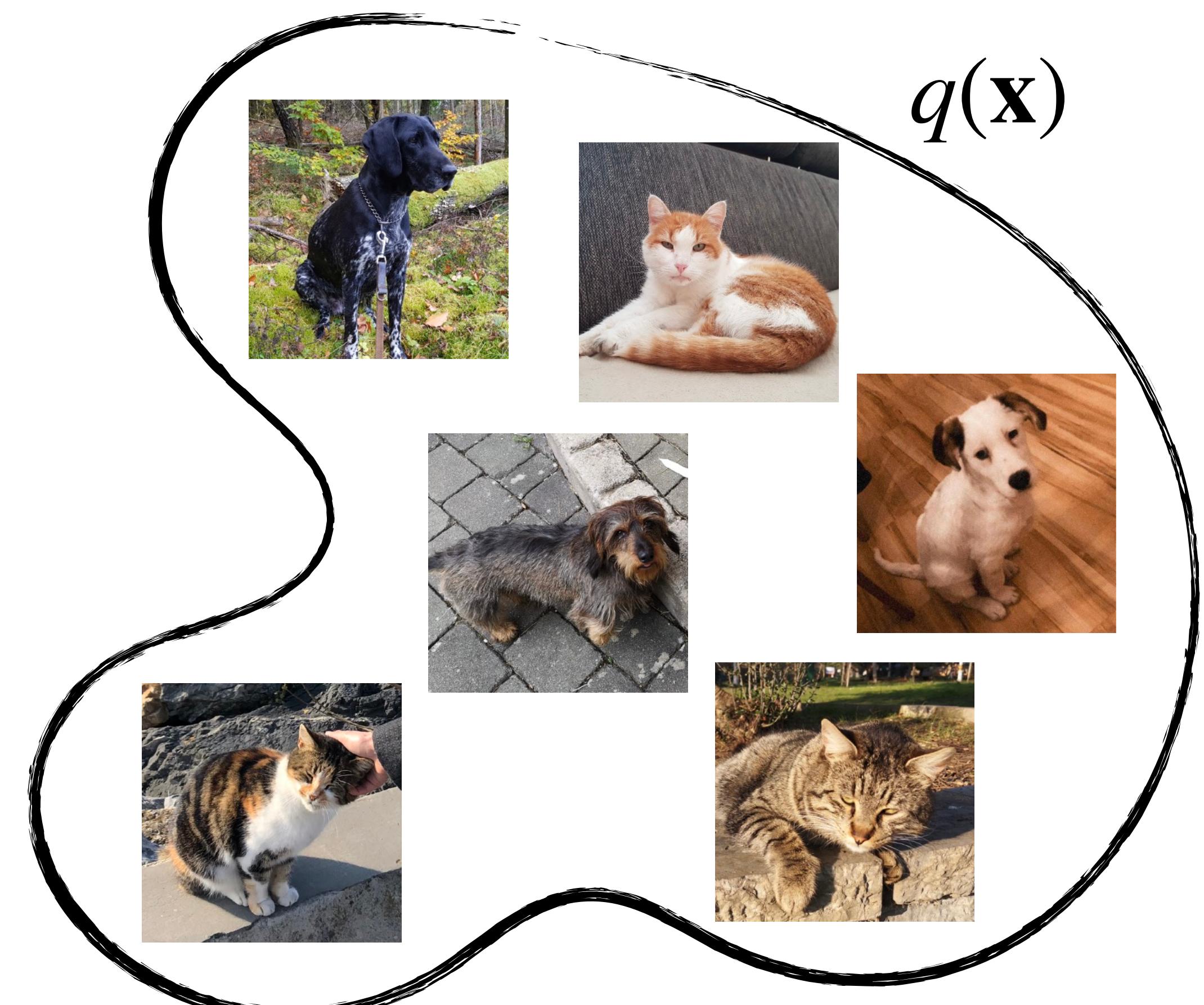
Figure 1: Selected samples from our best ImageNet 512×512 model (FID 3.85)

# Recap: Deep Generative Models

- Given a list of data points  $X = \langle \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)} \rangle$  coming from a data distribution  $q(\mathbf{x})$ .
- Generative model:** a model  $p_\theta(\mathbf{x})$  for this distribution, from which we can **sample**.



# Denoising Diffusion Probabilistic Models (DDPMs)



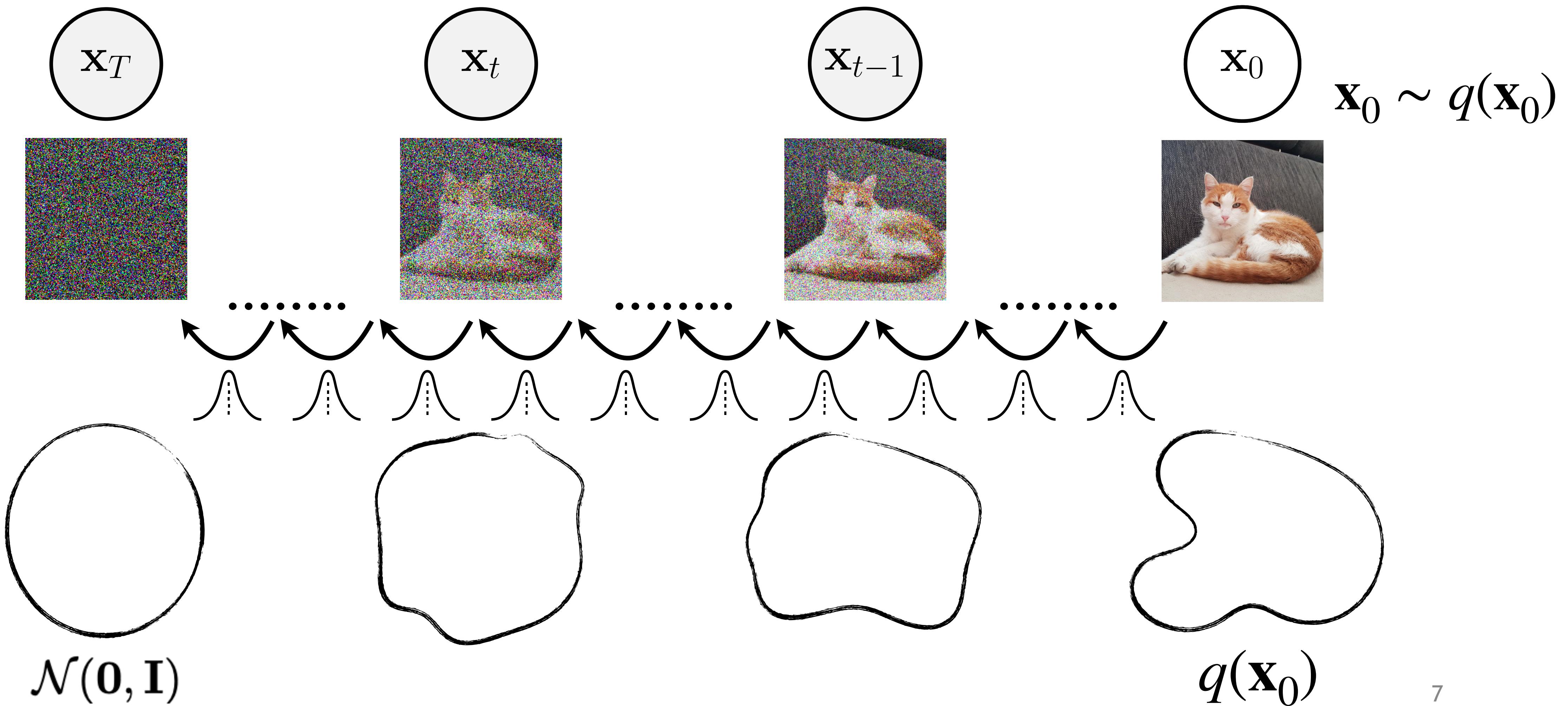
# Denoising Diffusion Probabilistic Models (DDPMs)

---

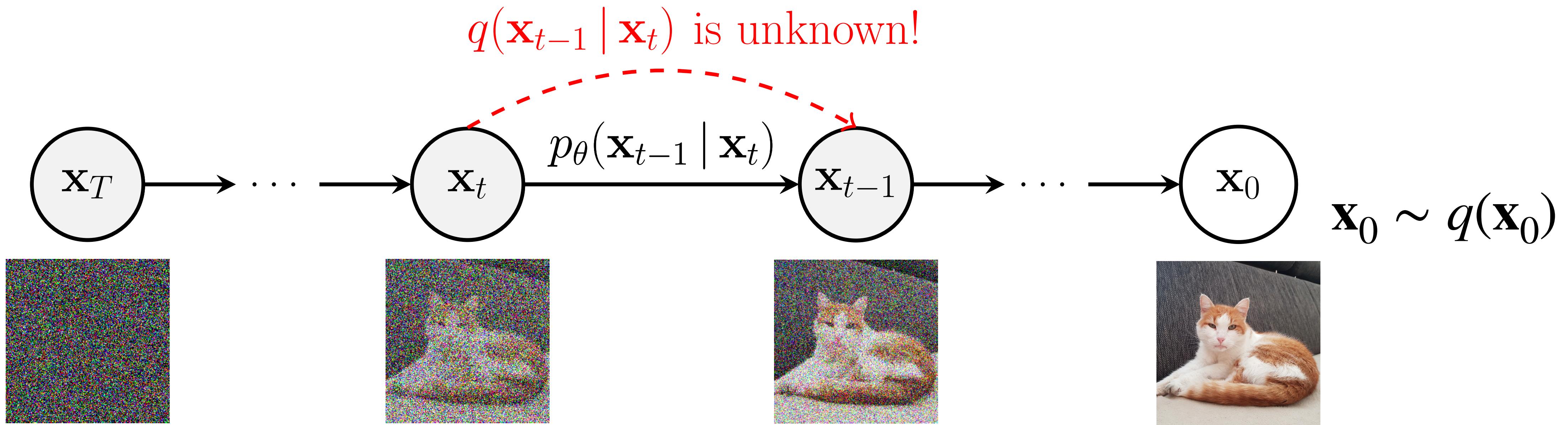
$\mathbf{x}_0 \sim q(\mathbf{x}_0)$



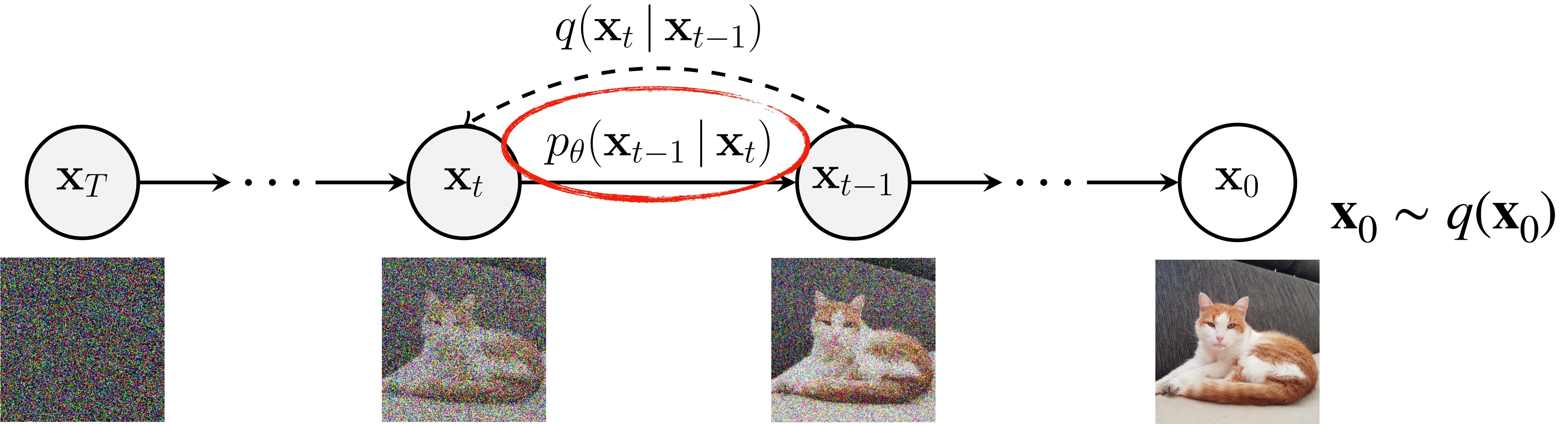
# Denoising Diffusion Probabilistic Models (DDPMs)



# Denoising Diffusion Probabilistic Models (DDPMs)



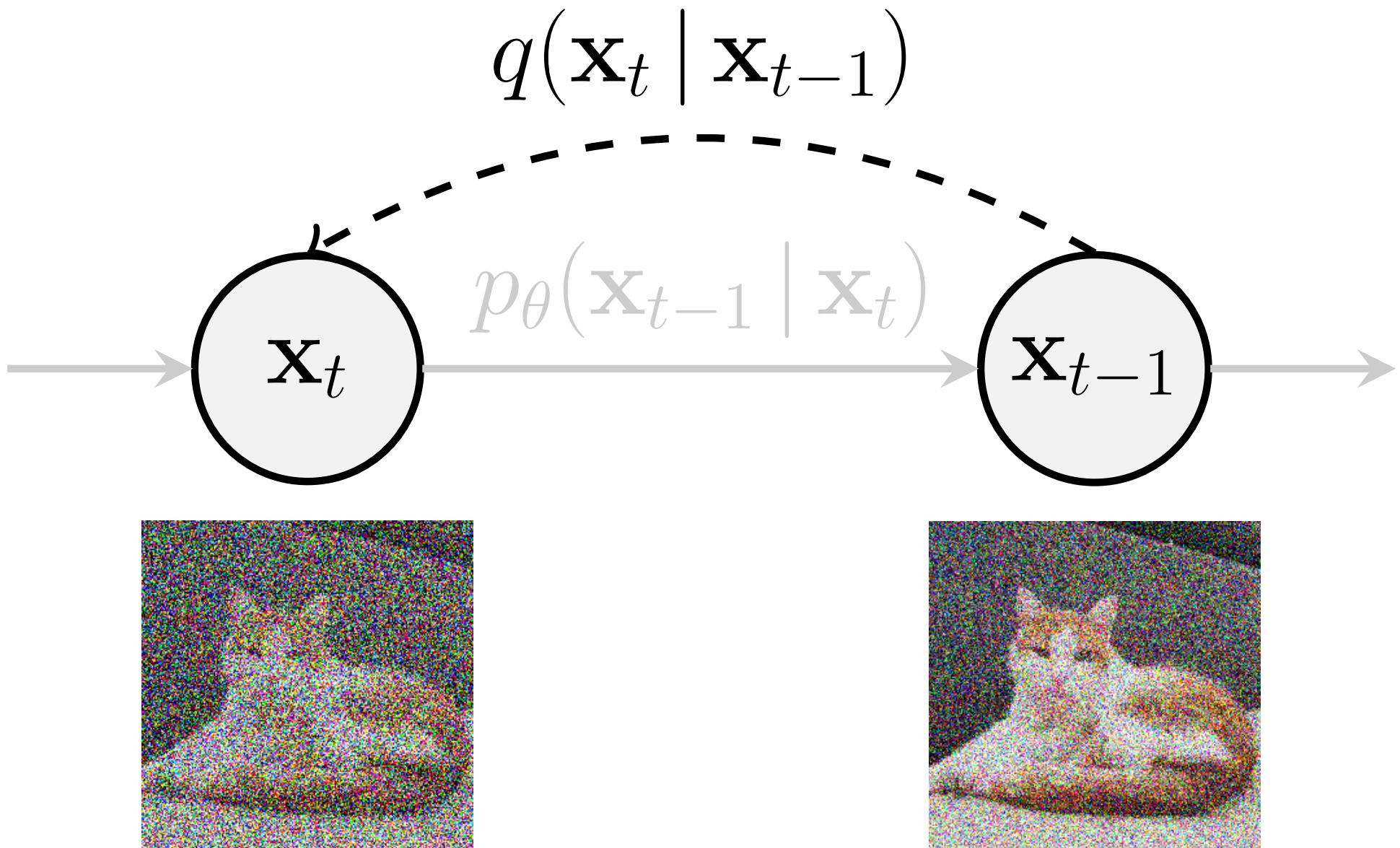
# Denoising Diffusion Probabilistic Models (DDPMs)



**Central idea:** Training a deep neural network that iteratively denoises an image through a reverse process that corresponds to a diffusion process which destroys the data structure by adding noise.

# Fixed Forward (Diffusion) Process

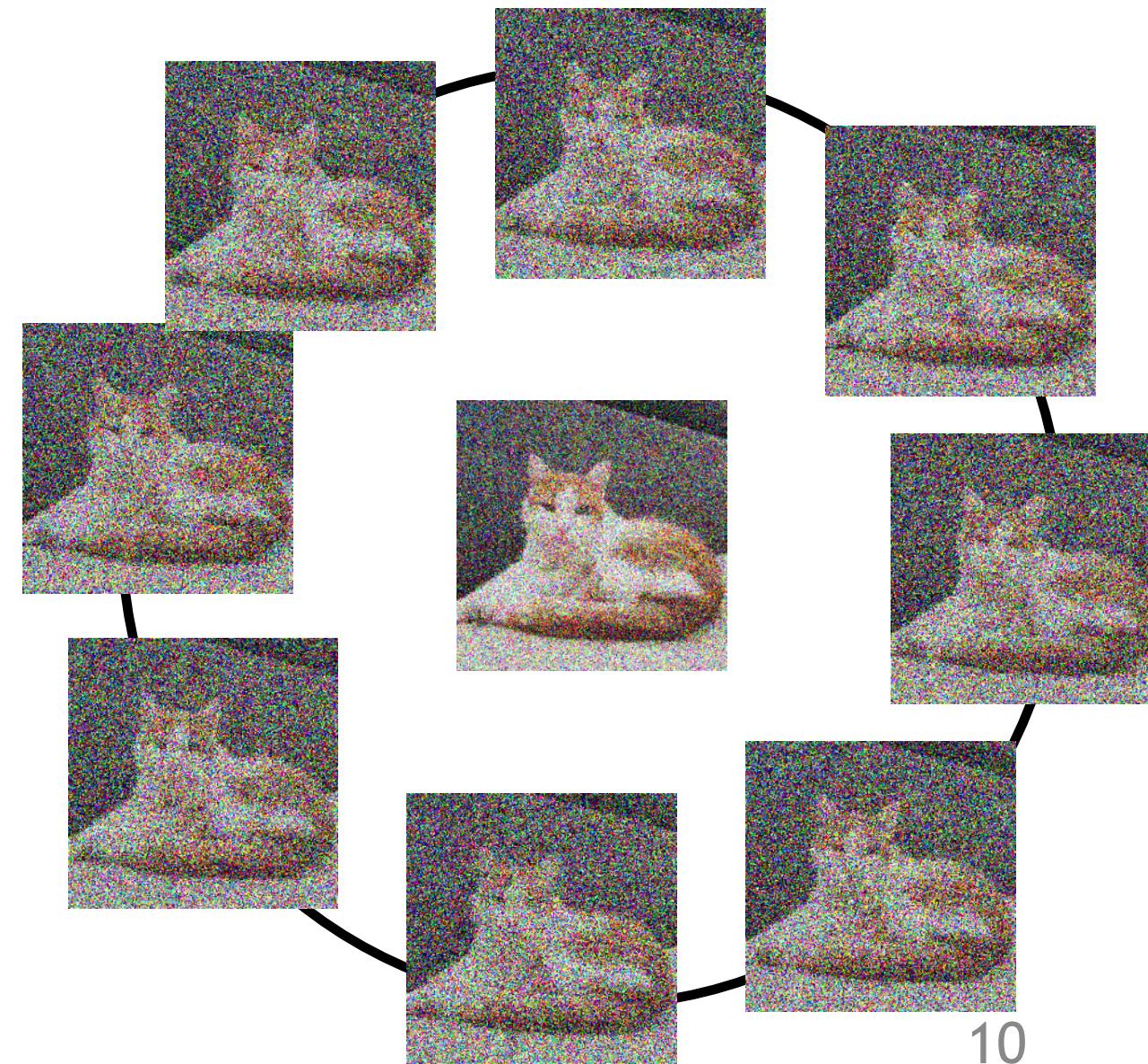
forward process



- The **forward** process (i.e., ***diffusion*** process) gradually adds Gaussian noise according to a known variance schedule  $\beta_1 < \beta_2 < \dots < \beta_T$ .

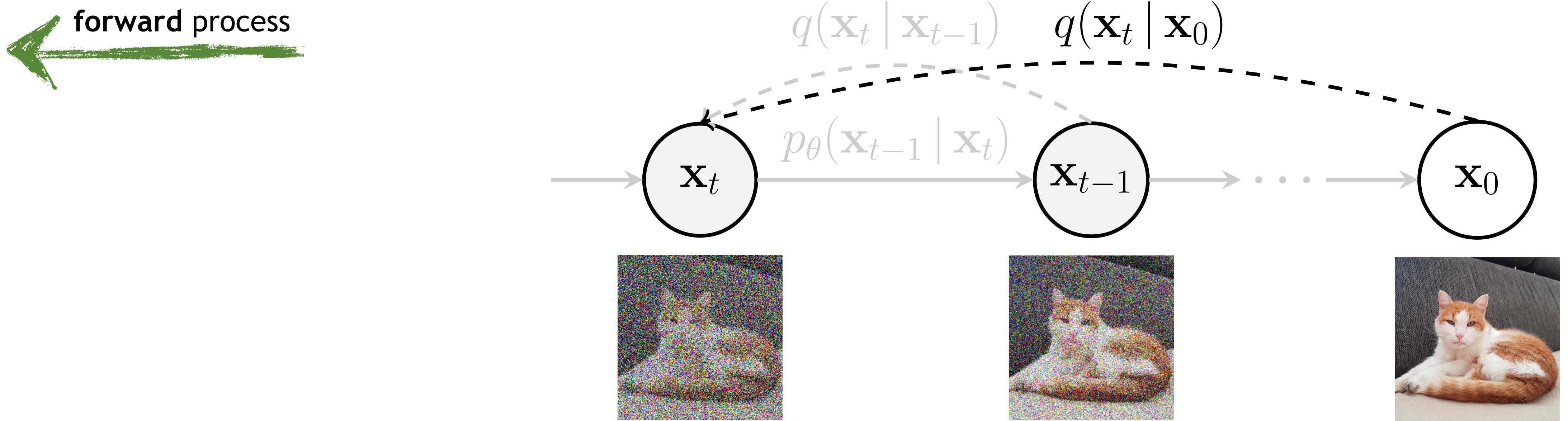
$$q(x_t | x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t} x_{t-1}, \beta_t \mathbf{I})$$

$$q(\mathbf{x}_{1:T} | \mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1}) \rightarrow \text{joint distribution}$$



$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

# Fixed Forward (Diffusion) Process



- We can also directly jump to any time-step using:

$$\alpha_t = 1 - \beta_t \quad \text{and} \quad \bar{\alpha}_t = \prod_{s=1}^t \alpha_s$$

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I})$$

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon} \quad \text{where} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

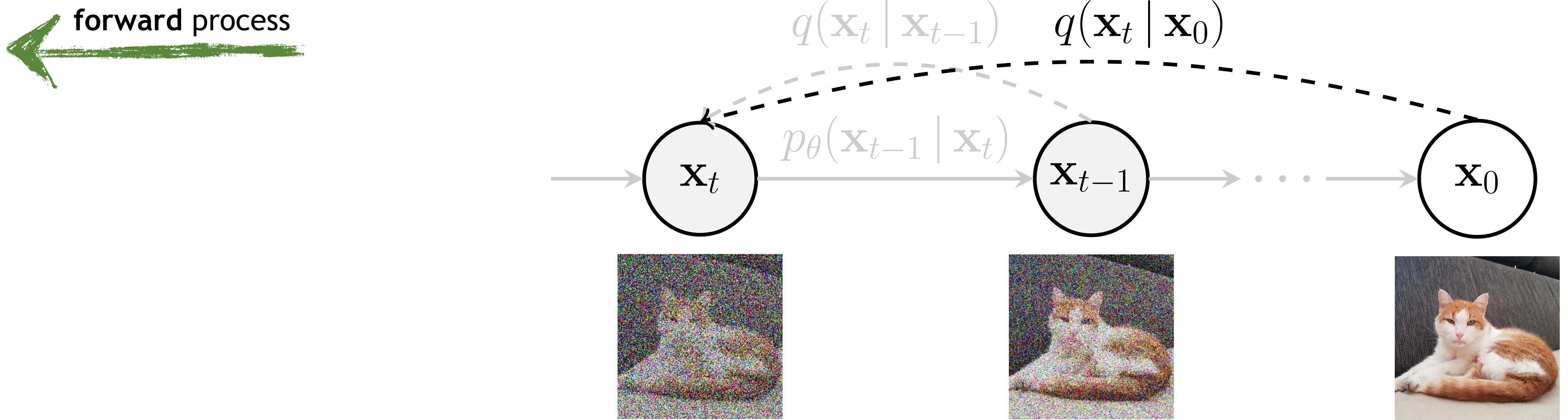
the noise schedule is designed such that:

$$\bar{\alpha}_T \rightarrow 0$$

$$\bar{\alpha}_1 > \dots > \bar{\alpha}_T$$

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

# Fixed Forward (Diffusion) Process



- We can also directly jump to any time-step using:

$$\alpha_t = 1 - \beta_t \quad \text{and} \quad \bar{\alpha}_t = \prod_{s=1}^t \alpha_s$$

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I})$$

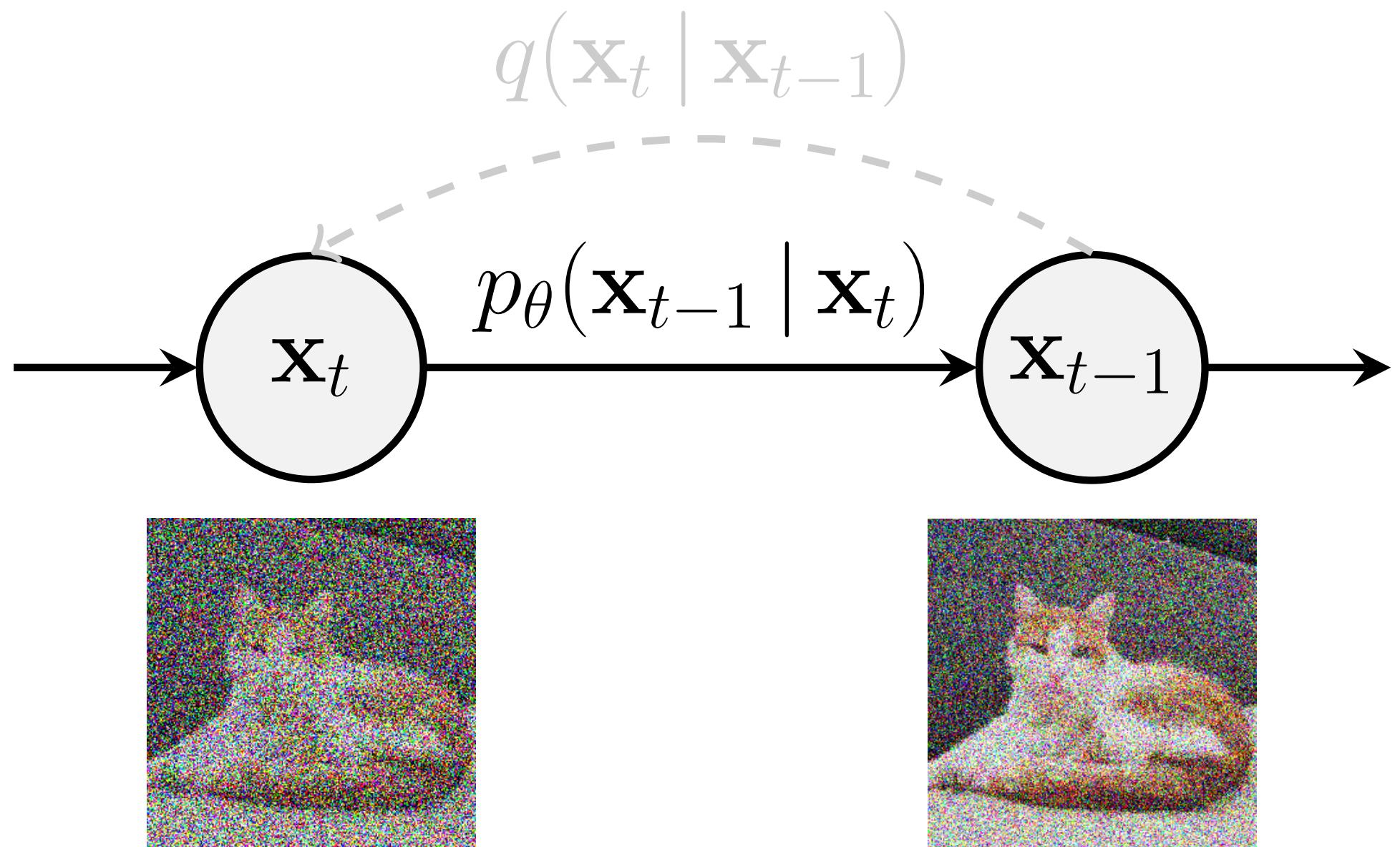
$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon} \quad \text{where} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

consecutively merging Gaussians:

$$\begin{aligned} \mathbf{x}_t &= \sqrt{\alpha_t} \mathbf{x}_{t-1} + \sqrt{1 - \alpha_t} \boldsymbol{\epsilon}_{t-1} \\ &= \sqrt{\alpha_t \alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{1 - \alpha_t \alpha_{t-1}} \boldsymbol{\epsilon}_{t-2} \\ &= \dots \\ &= \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon} \end{aligned}$$

# Generative Reverse (Denoising) Process

reverse process 



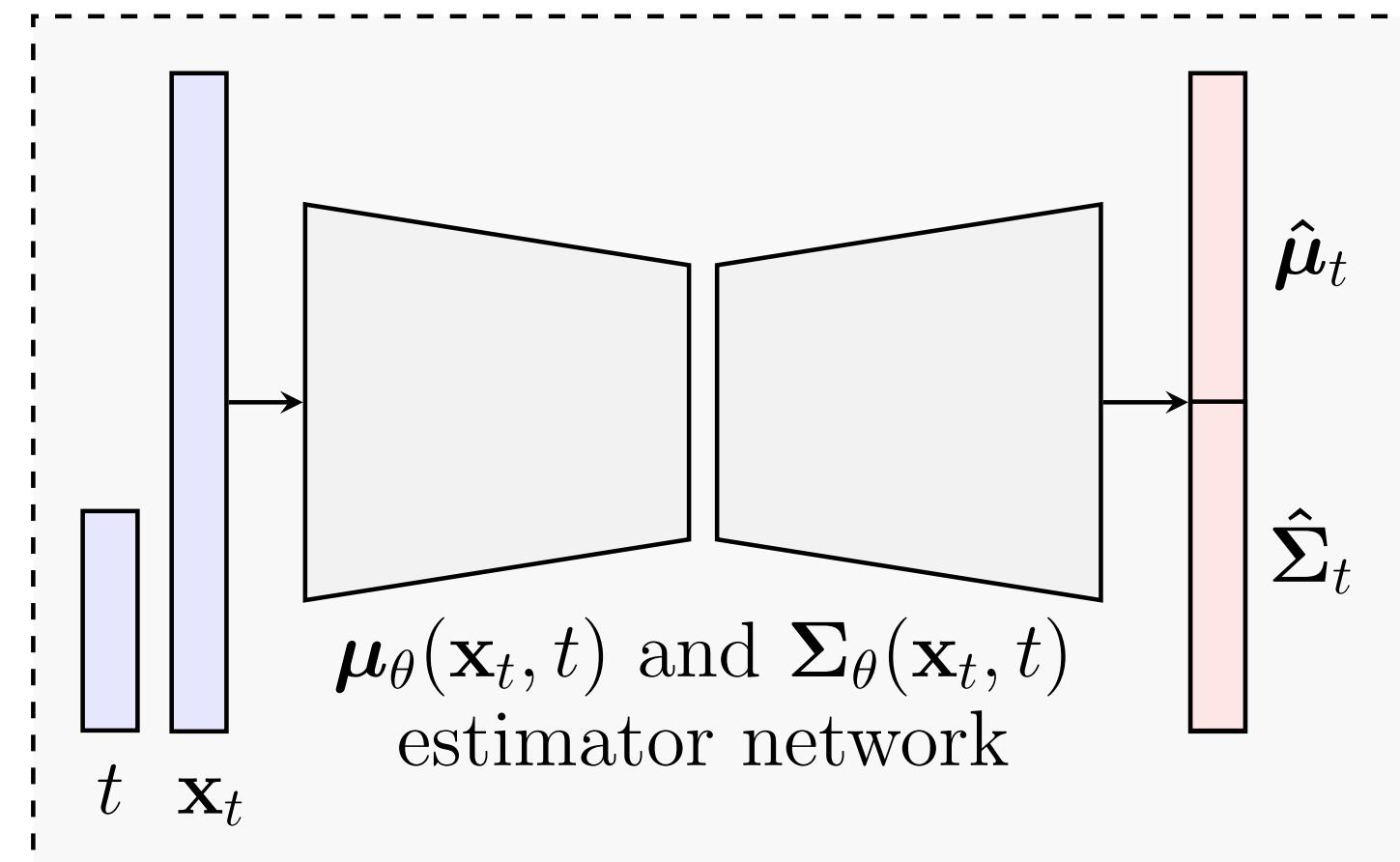
- The reverse process is the joint distribution, with learned Gaussian transitions starting from noise.

$$p_\theta(\mathbf{x}_{0:T}) = p_\theta(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$$

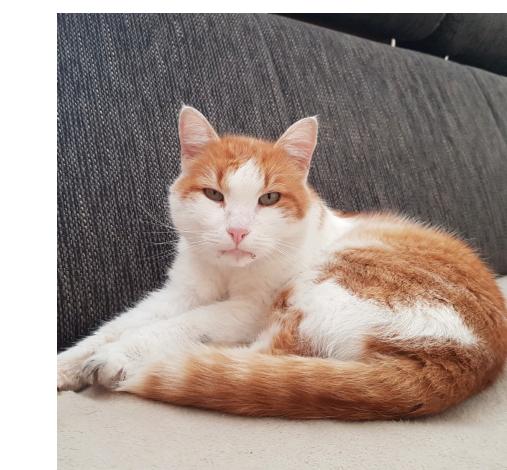
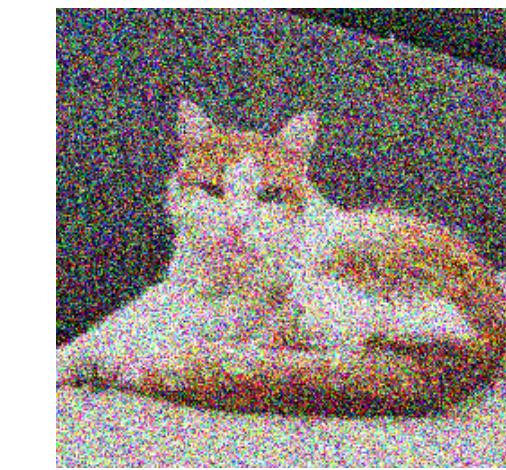
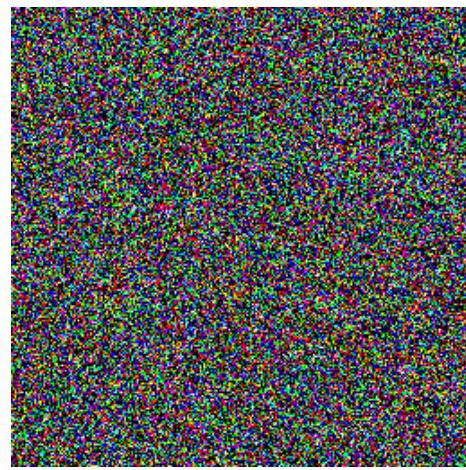
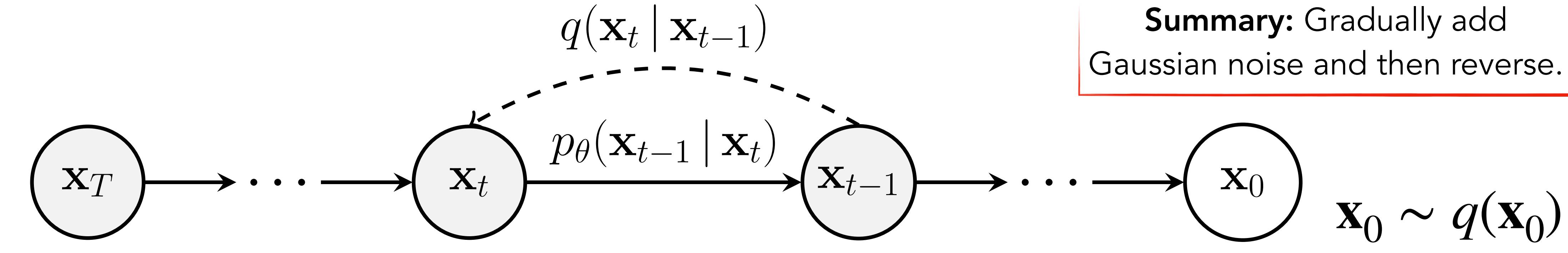
known

$$\underline{p(\mathbf{x}_T)} = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}, \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$$



# Denoising Diffusion Probabilistic Models (DDPMs)



$$\mathbf{x}_0 \sim q(\mathbf{x}_0)$$

forward process

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I})$$

$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I})$$

reverse process

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$$

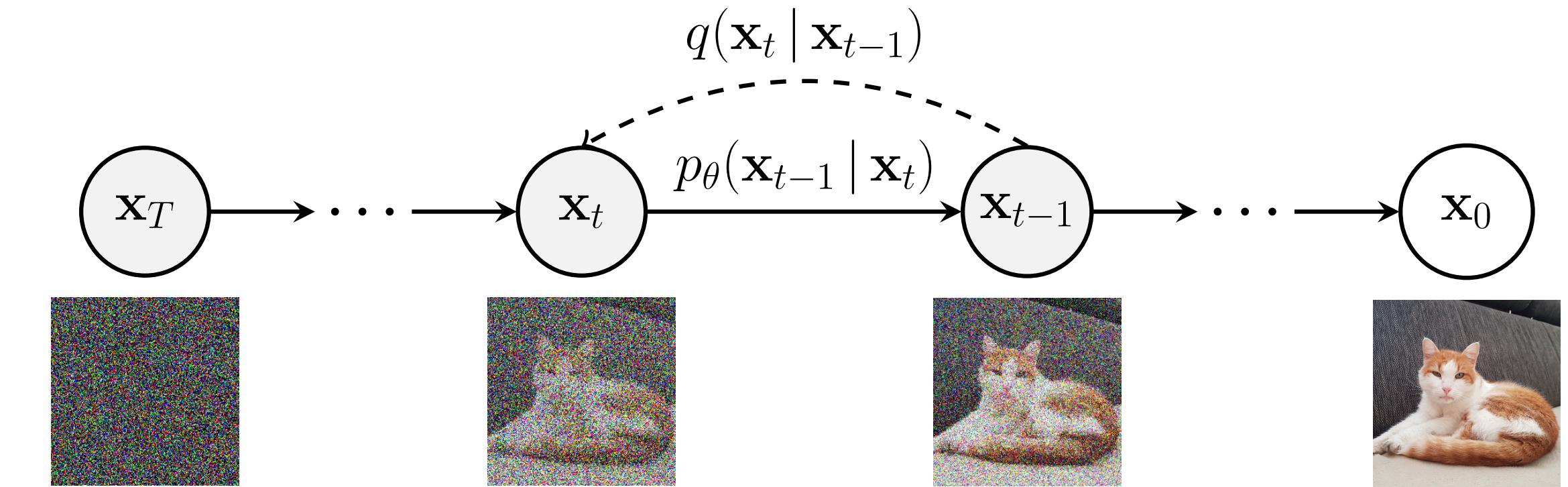
# Training: Denoising Diffusion Probabilistic Models

Maximize:  $\mathbb{E}_{q(\mathbf{x}_0)}[\log p_{\theta}(\mathbf{x}_0)]$

$$\mathbb{E}_{q(\mathbf{x}_0)}[\log p_{\theta}(\mathbf{x}_0)] \geq \mathbb{E}_q \left[ \log p(\mathbf{x}_T) + \sum_{t=1}^T \log \frac{p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t)}{q(\mathbf{x}_t | \mathbf{x}_{t-1})} \right]$$

$\hat{L}$ , where  $\hat{L}$  is the ELBO

Training is similar to VAEs, by maximizing an evidence lower-bound (ELBO) on the log-likelihood.



reverse process:  $p_{\theta}(\mathbf{x}_{0:T}) = p_{\theta}(\mathbf{x}_T) \prod_{t=1}^T p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t)$

$$p_{\theta}(\mathbf{x}_0) = \int p_{\theta}(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}$$

$$\log p_{\theta}(\mathbf{x}_0) = \log \left[ \int d\mathbf{x}_{1:T} q(\mathbf{x}_{1:T} | \mathbf{x}_0) \frac{p_{\theta}(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \right]$$

$$\geq \int d\mathbf{x}_{1:T} q(\mathbf{x}_{1:T} | \mathbf{x}_0) \log \left( p(\mathbf{x}_T) \prod_{t=1}^T \frac{p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t)}{q(\mathbf{x}_t | \mathbf{x}_{t-1})} \right)$$

# Training: Denoising Diffusion Probabilistic Models

Maximize:  $\mathbb{E}_{q(\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0)]$

$$\mathbb{E}_{q(\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0)] \geq \mathbb{E}_q \left[ \log p(\mathbf{x}_T) + \sum_{t=1}^T \log \frac{p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)}{q(\mathbf{x}_t | \mathbf{x}_{t-1})} \right]$$

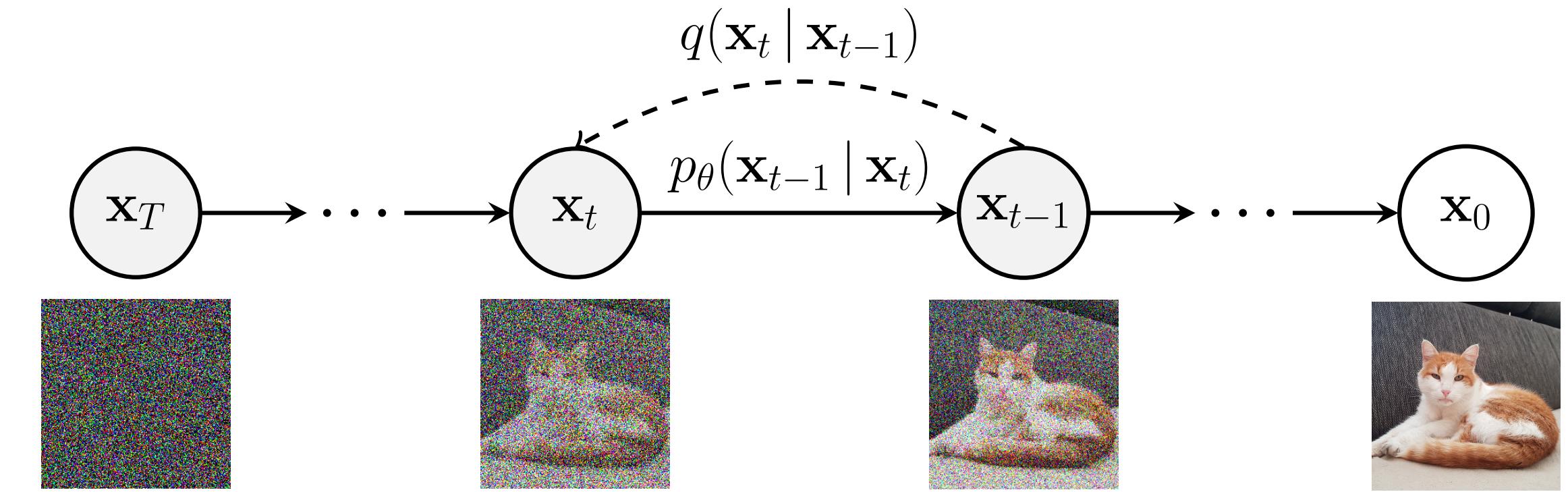
$\mathcal{L}$ , where  $\mathcal{L}$  is the ELBO

Training is similar to VAEs, by maximizing an evidence lower-bound (ELBO) on the log-likelihood.

plug in the following:  $q(\mathbf{x}_t | \mathbf{x}_{t-1}) = q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_0) = \frac{q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)q(\mathbf{x}_t | \mathbf{x}_0)}{q(\mathbf{x}_{t-1} | \mathbf{x}_0)}$

...

$$\mathcal{L} = -\mathbb{E}_{q(\mathbf{x}_0)} \left[ \underbrace{D_{\text{KL}}(q(\mathbf{x}_T | \mathbf{x}_0) \| p(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^T \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t))}_{L_{t-1}} - \underbrace{\log p_\theta(\mathbf{x}_0 | \mathbf{x}_1)}_{L_0} \right]$$



# Training: Denoising Diffusion Probabilistic Models

Maximize:  $\mathbb{E}_{q(\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0)]$

$$\mathbb{E}_{q(\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0)] \geq \mathbb{E}_q \left[ \log p(\mathbf{x}_T) + \sum_{t=1}^T \log \frac{p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)}{q(\mathbf{x}_t | \mathbf{x}_{t-1})} \right]$$

$\mathcal{L}$ , where  $\mathcal{L}$  is the ELBO

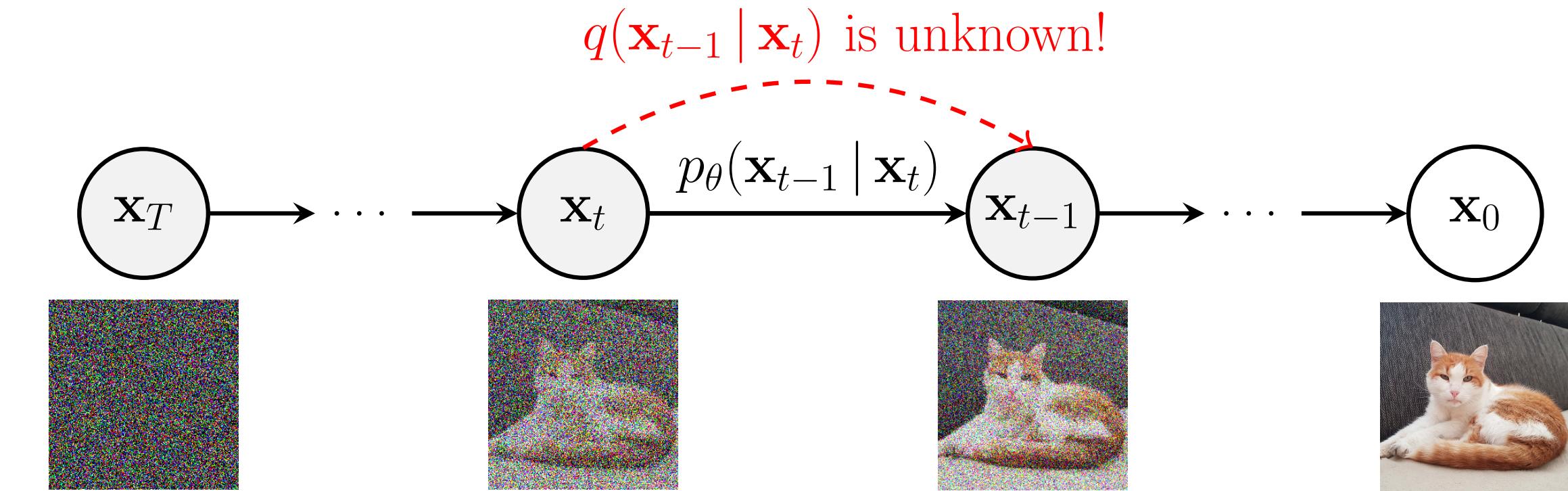
Training is similar to VAEs, by maximizing an evidence lower-bound (ELBO) on the log-likelihood.

plug in the following:  $q(\mathbf{x}_t | \mathbf{x}_{t-1}) = q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_0) = \frac{q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)q(\mathbf{x}_t | \mathbf{x}_0)}{q(\mathbf{x}_{t-1} | \mathbf{x}_0)}$

...

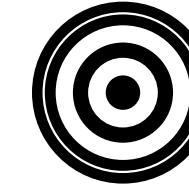
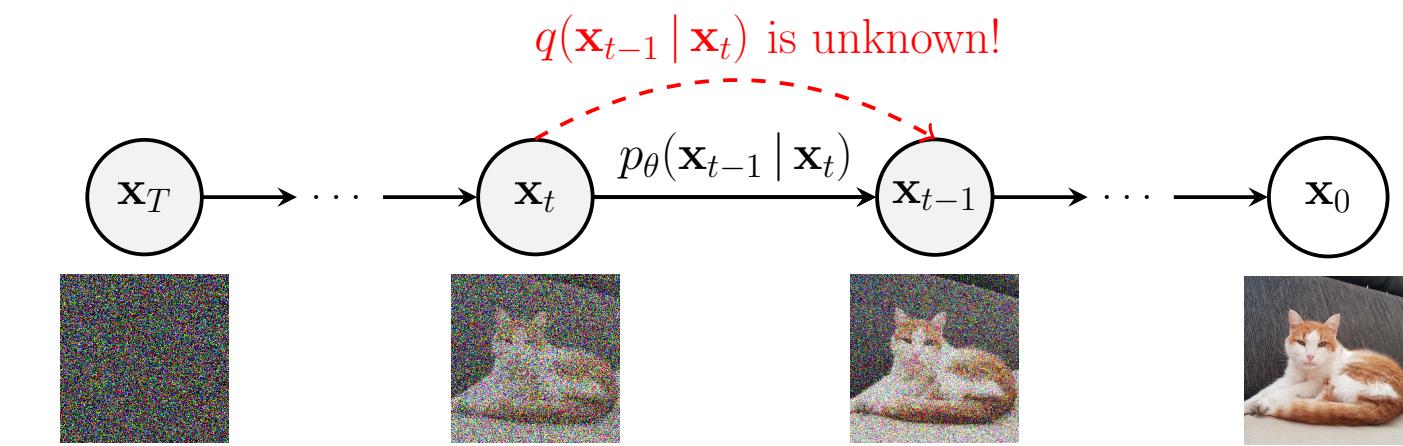
$$\mathcal{L} = -\mathbb{E}_{q(\mathbf{x}_0)} \left[ \underbrace{D_{\text{KL}}(q(\mathbf{x}_T | \mathbf{x}_0) \| p(\mathbf{x}_T))}_{L_T} + \underbrace{\sum_{t=2}^T D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t))}_{L_{t-1}} - \underbrace{\log p_\theta(\mathbf{x}_0 | \mathbf{x}_1)}_{L_0} \right]$$

We can compute an approximation to the overall objective by sampling random time points and optimizing a single  $L_t$  each time.



# Training: Denoising Diffusion Probabilistic Models

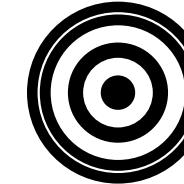
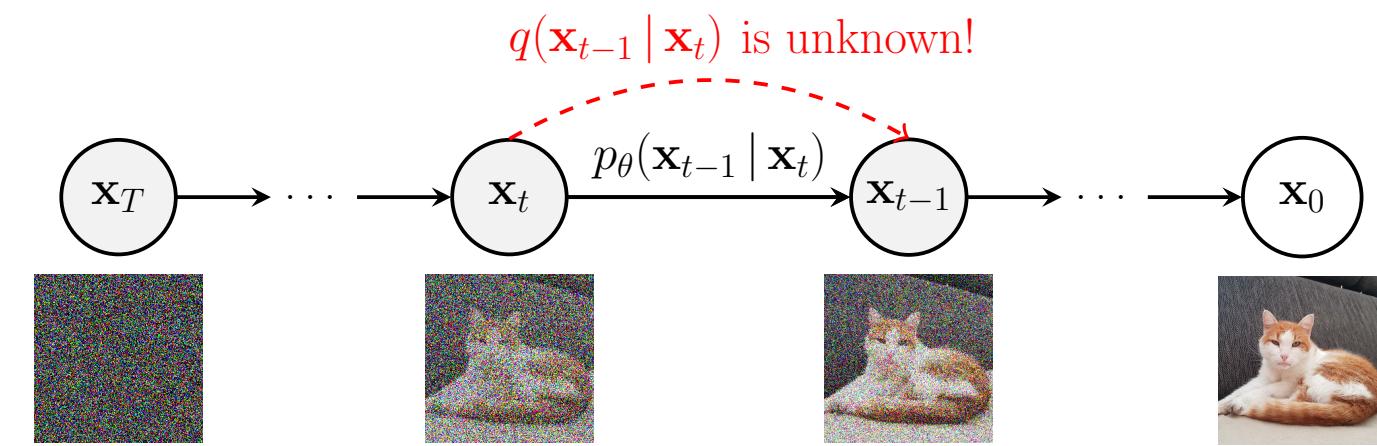
$$\hat{L} = -\mathbb{E}_{q(\mathbf{x}_0)} \left[ \underbrace{D_{\text{KL}}(q(\mathbf{x}_T | \mathbf{x}_0) \| p(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^T \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t))}_{L_{t-1}} - \log p_{\theta}(\mathbf{x}_0 | \mathbf{x}_1) \underbrace{}_{L_0} \right]$$



Minimizing the KL divergence between the modeled reverse process transition probability and the true (unknown) reverse process conditioned on the input.

# Training: Denoising Diffusion Probabilistic Models

$$\hat{L} = -\mathbb{E}_{q(\mathbf{x}_0)} \left[ \underbrace{D_{\text{KL}}(q(\mathbf{x}_T | \mathbf{x}_0) \| p(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^T \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t))}_{L_{t-1}} - \log p_{\theta}(\mathbf{x}_0 | \mathbf{x}_1) \underbrace{}_{L_0} \right]$$



Minimizing the KL divergence between the modeled reverse process transition probability and the true (unknown) reverse process conditioned on the input.

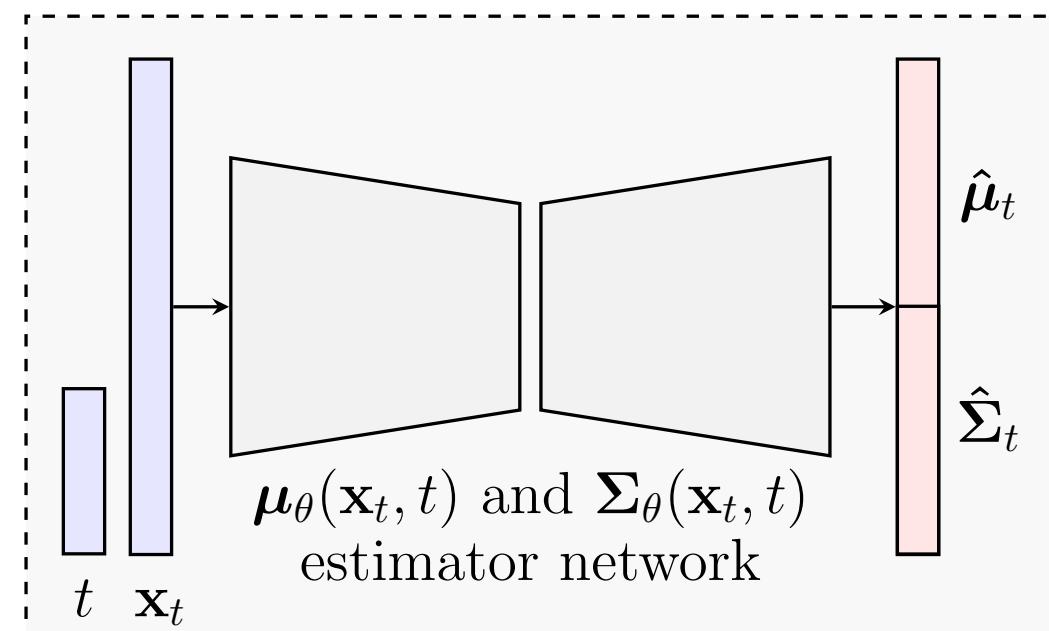
This is tractable when conditioned on  $\mathbf{x}_0$ :

$$\begin{aligned} q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) &\propto q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_0) q(\mathbf{x}_{t-1} | \mathbf{x}_0) \\ &= \mathcal{N}(\mathbf{x}_{t-1} | \frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1 - \bar{\alpha}_t} \mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{x}_t, \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t \mathbf{I}) \end{aligned}$$

Using our forward process properties:

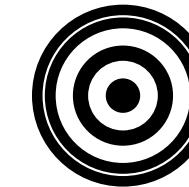
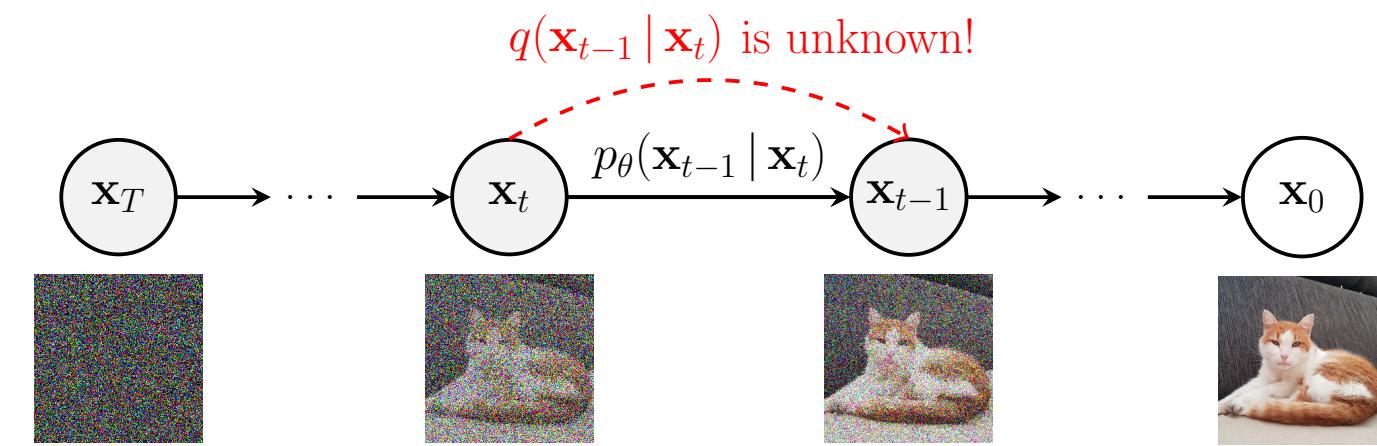
$$\begin{aligned} q(\mathbf{x}_t | \mathbf{x}_{t-1}) &= \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}) \\ q(\mathbf{x}_t | \mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}) \end{aligned}$$

Optimize  $p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_{\theta}(\mathbf{x}_t, t), \Sigma_{\theta}(\mathbf{x}_t, t))$  to follow  $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$  via KL-divergence between Gaussians.



# Training: Denoising Diffusion Probabilistic Models

$$\hat{L} = -\mathbb{E}_{q(\mathbf{x}_0)} \left[ \underbrace{D_{\text{KL}}(q(\mathbf{x}_T | \mathbf{x}_0) \| p(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^T \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t))}_{L_{t-1}} - \log p_{\theta}(\mathbf{x}_0 | \mathbf{x}_1) \underbrace{}_{L_0} \right]$$



Minimizing the KL divergence between the modeled reverse process transition probability and the true (unknown) reverse process conditioned on the input.

This is tractable when conditioned on  $\mathbf{x}_0$ :

$$\begin{aligned} q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) &\propto q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_0) q(\mathbf{x}_{t-1} | \mathbf{x}_0) \\ &= \mathcal{N}(\mathbf{x}_{t-1} | \frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1 - \bar{\alpha}_t} \mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{x}_t, \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t \mathbf{I}) \end{aligned}$$

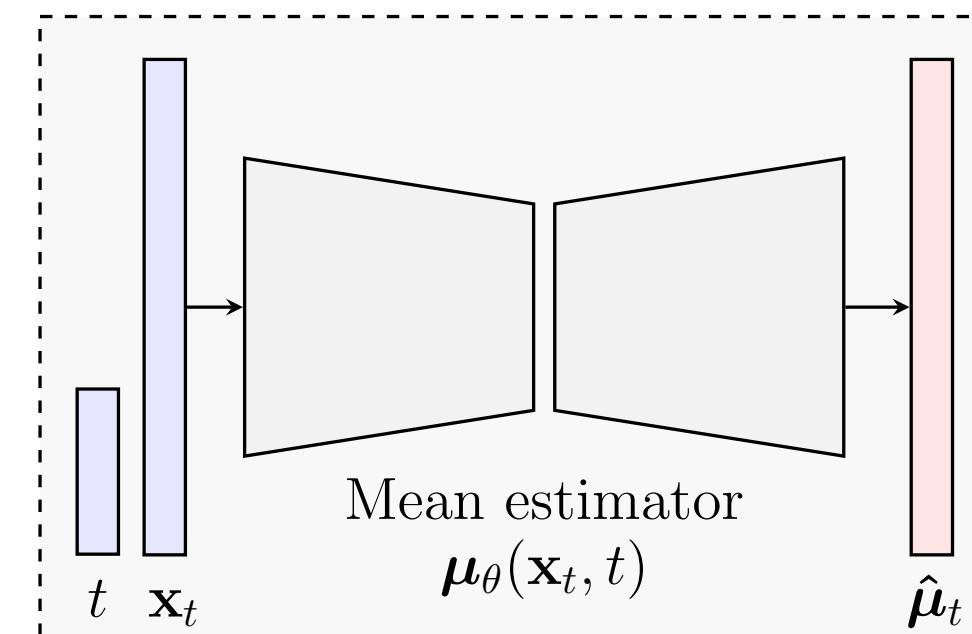
Using our forward process properties:

$$\begin{aligned} q(\mathbf{x}_t | \mathbf{x}_{t-1}) &= \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}) \\ q(\mathbf{x}_t | \mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}) \end{aligned}$$

Optimize  $p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_{\theta}(\mathbf{x}_t, t), \Sigma_{\theta}(\mathbf{x}_t, t))$  to follow  $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$  via KL-divergence between Gaussians.

**Simplify (1):** We often set  $\Sigma_{\theta}(\mathbf{x}_t, t) = \sigma_t^2 \mathbf{I}$ , as untrained time-dependent constants, and optimize a network to only predict the mean:

$$L_{t-1} = \mathbb{E}_q \left[ \frac{1}{2\sigma_t^2} \left\| \frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1 - \bar{\alpha}_t} \mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{x}_t - \mu_{\theta}(\mathbf{x}_t, t) \right\|^2 \right] + C$$



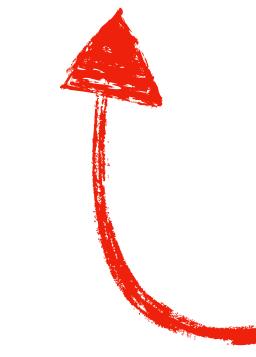
# Training: Denoising Diffusion Probabilistic Models

Optimize  $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I})$  to follow  $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$  via KL-divergence between Gaussians.

$$L_{t-1} = \mathbb{E}_q \left[ \frac{1}{2\sigma_t^2} \left\| \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} \mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \mathbf{x}_t - \mu_\theta(\mathbf{x}_t, t) \right\|^2 \right] + C$$

**Simplify (2):** We can re-parameterize the objective to predict the sampled noise  $\epsilon$  for time  $t$ , instead of predicting the mean.

$$q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1} | \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} \mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \mathbf{x}_t, \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t \mathbf{I})$$



Using forward process properties:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \epsilon \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

# Training: Denoising Diffusion Probabilistic Models

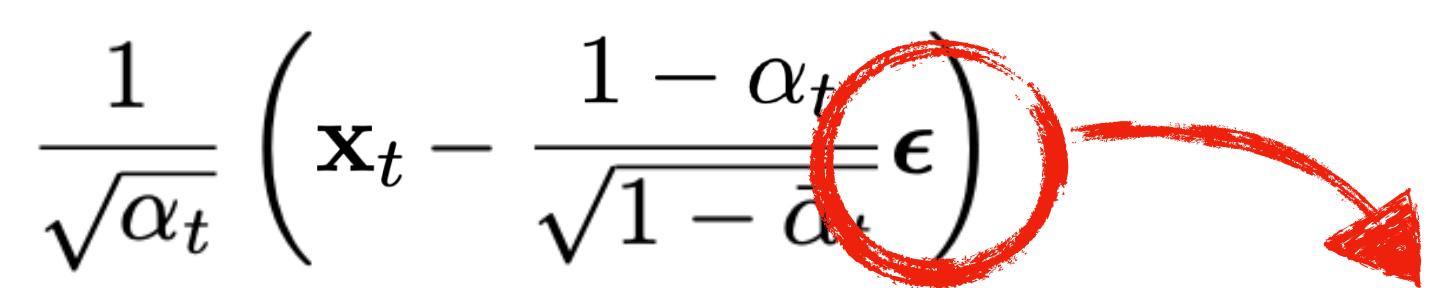
Optimize  $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I})$  to follow  $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$  via KL-divergence between Gaussians.

$$L_{t-1} = \mathbb{E}_q \left[ \frac{1}{2\sigma_t^2} \| \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} \mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \mathbf{x}_t - \mu_\theta(\mathbf{x}_t, t) \|^2 \right] + C$$

**Simplify (2):** We can re-parameterize the objective to predict the sampled noise  $\epsilon$  for time  $t$ , instead of predicting the mean.

$$q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1} | \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} \mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \mathbf{x}_t, \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t \mathbf{I})$$

||

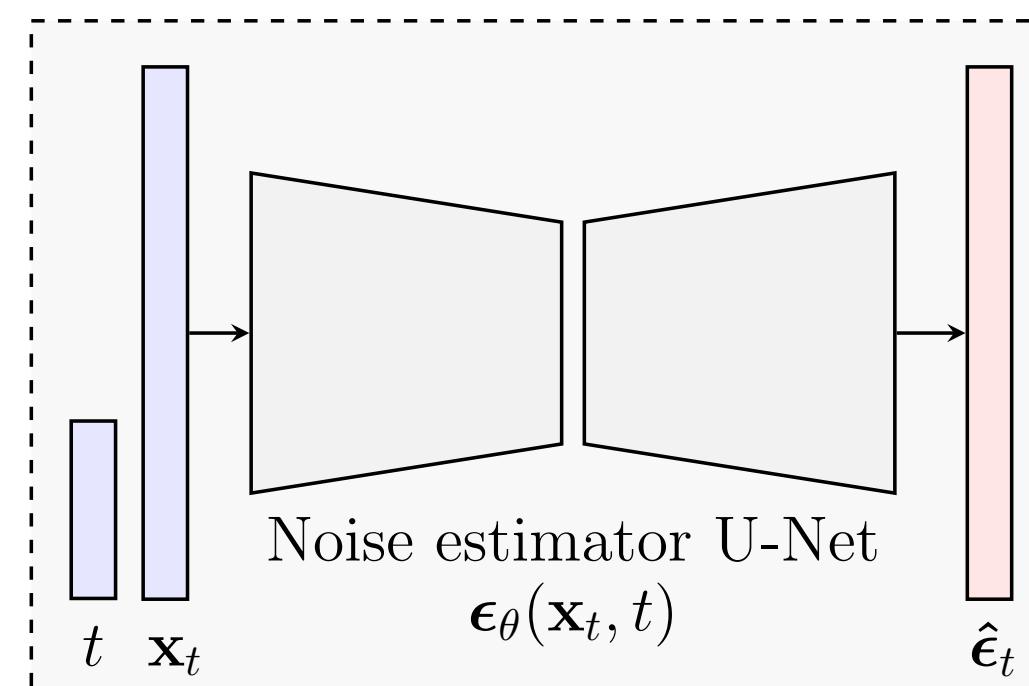
$$\frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon \right)$$


$\epsilon_\theta(\mathbf{x}_t, t)$

Optimize a noise estimator  $\epsilon_\theta(\mathbf{x}_t, t)$  to predict the noise added to  $\mathbf{x}_0$  at time  $t$ :

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[ \| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \epsilon, t) \|^2 \right]$$

such that one can predict  $\mathbf{x}_0$  via:  $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$   $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$



# Today

---

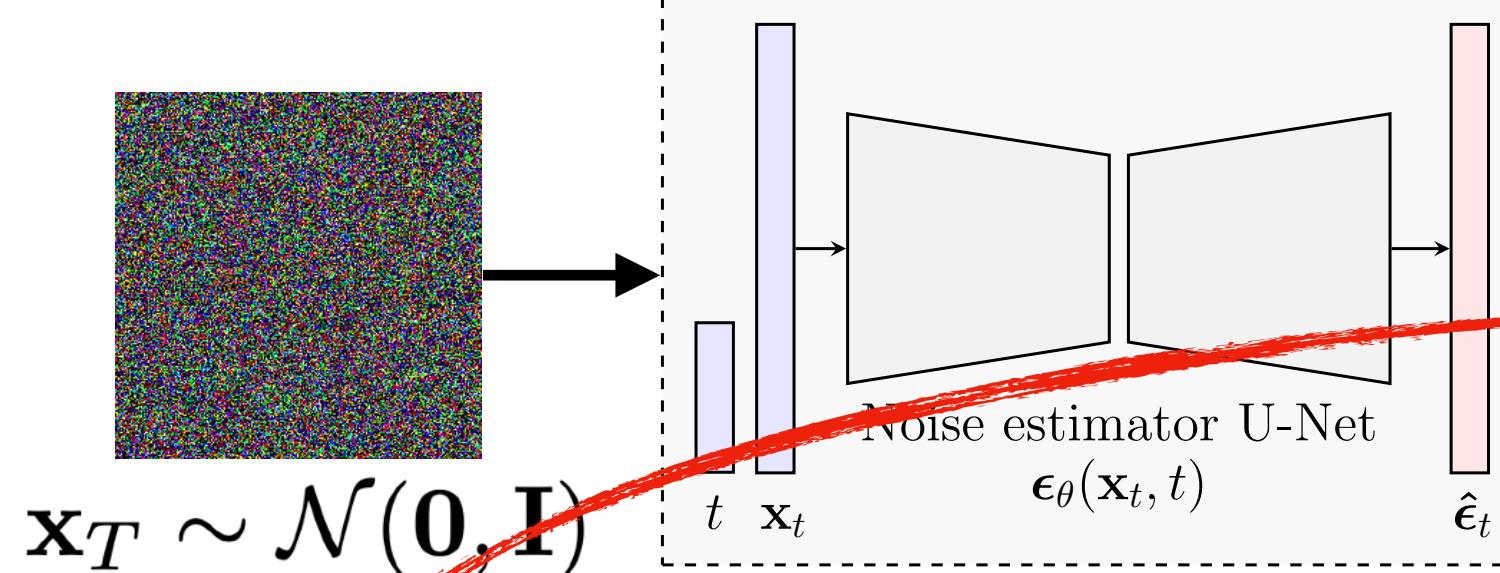
- Deep Generative Models
  - Variational Autoencoders
  - Generative Adversarial Networks
- Denoising Diffusion Probabilistic Models
  - Idea & Training Objective
  - Sampling
  - Further Extensions & Applications

# Sampling: Denoising Diffusion Probabilistic Models

$$\beta_1 < \beta_2 < \dots < \beta_T$$

$$T = 1000$$

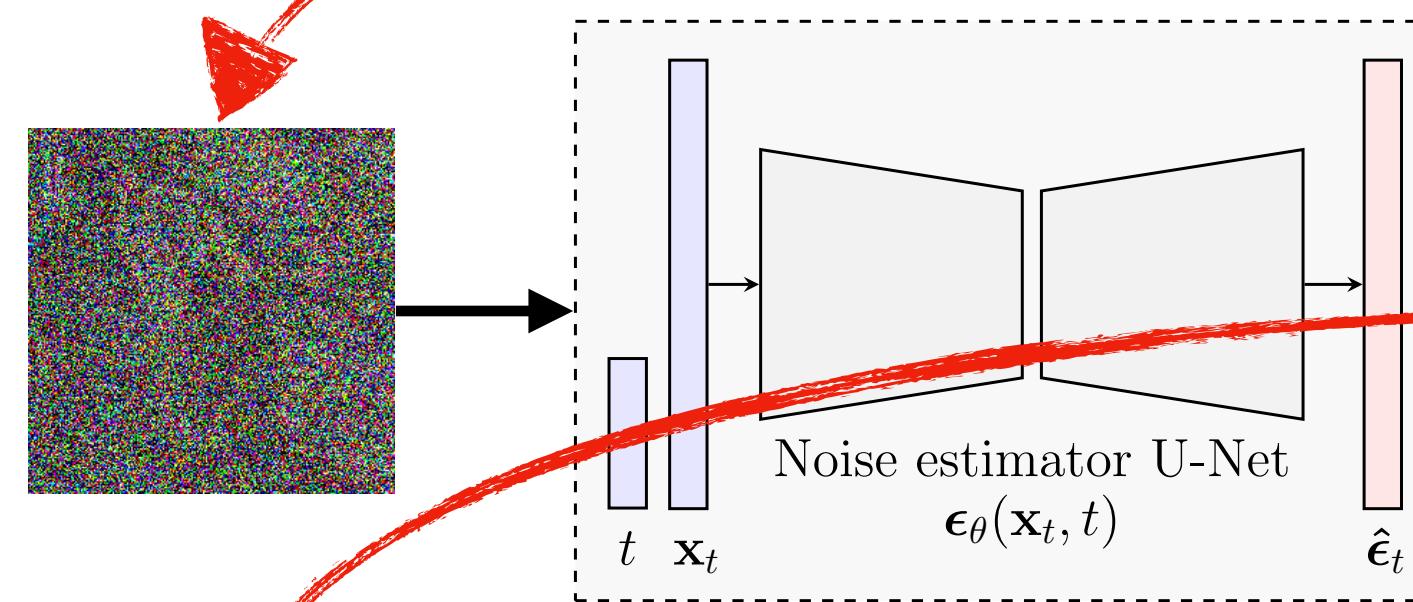
$$t = 1000$$



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

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

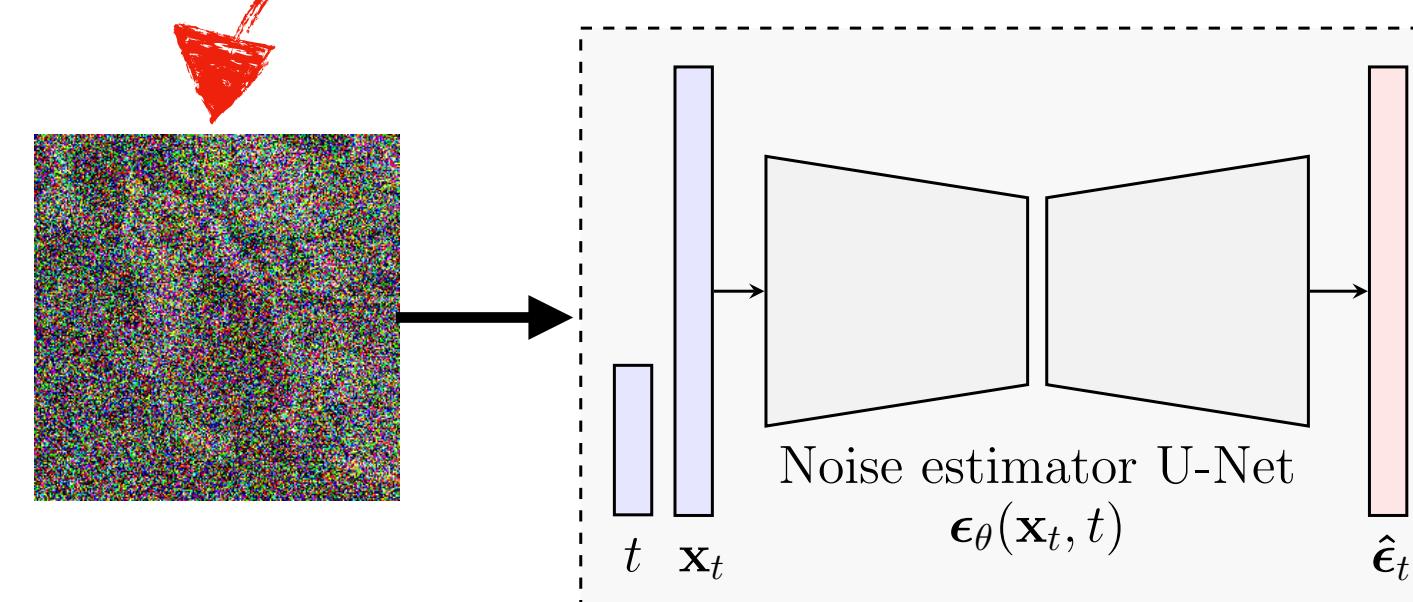
$$t = 999$$



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

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

$$t = 998$$



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

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

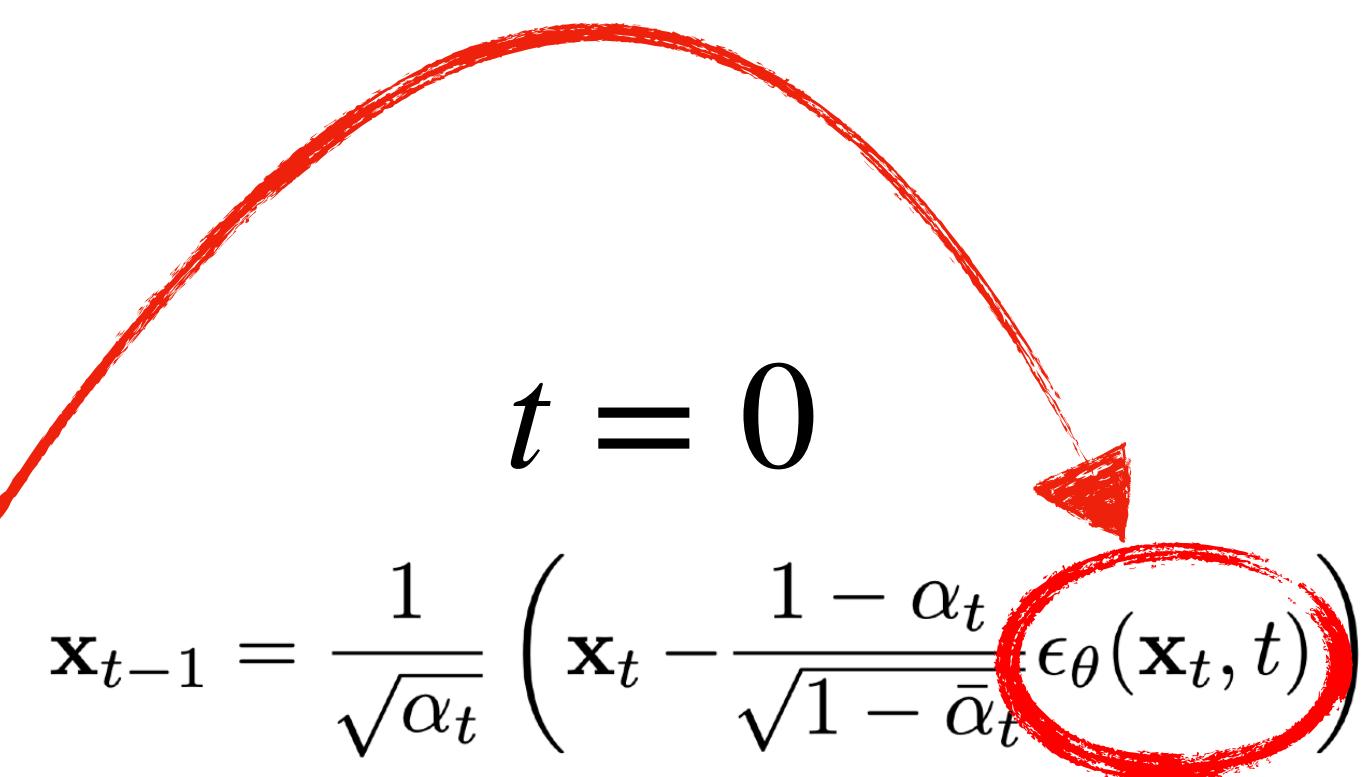
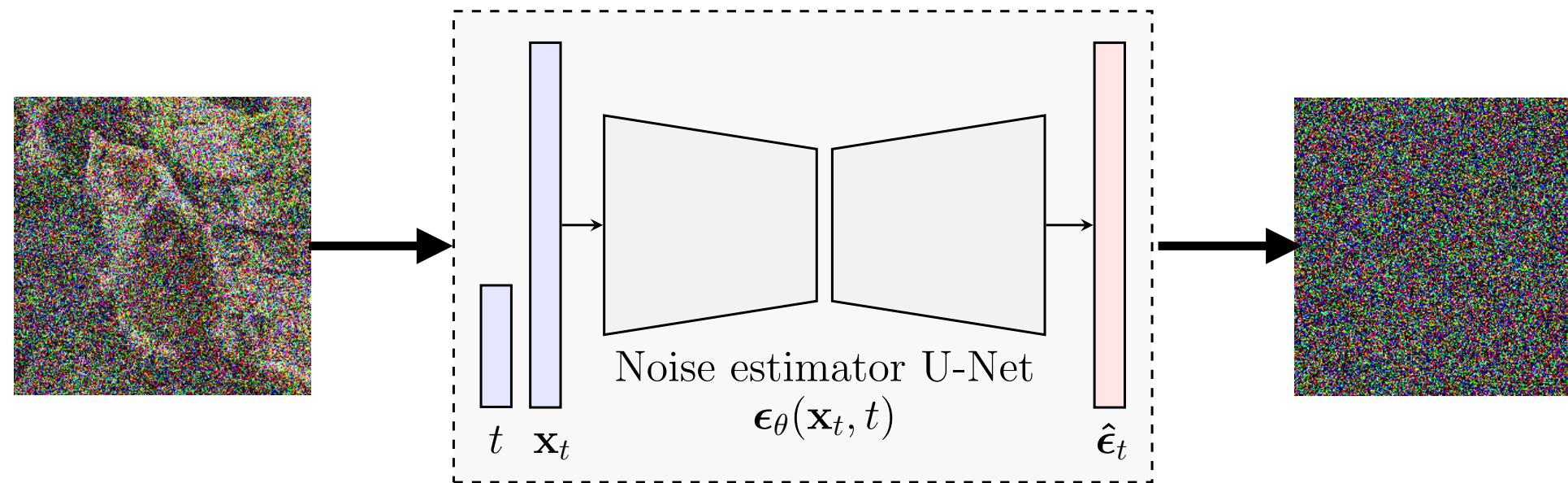
.....

# Sampling: Denoising Diffusion Probabilistic Models

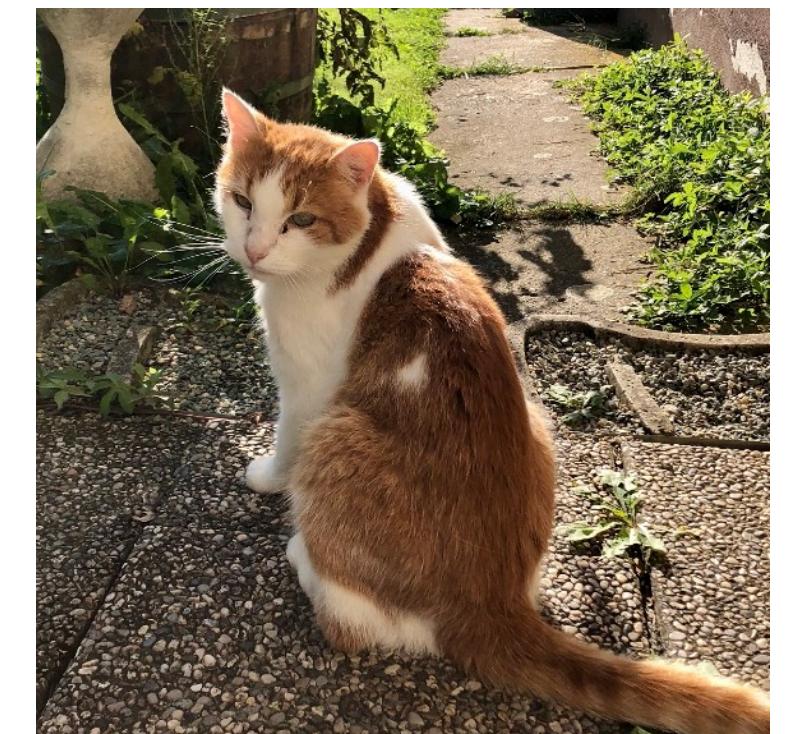
$$\beta_1 < \beta_2 < \dots < \beta_T$$

$$T = 1000$$

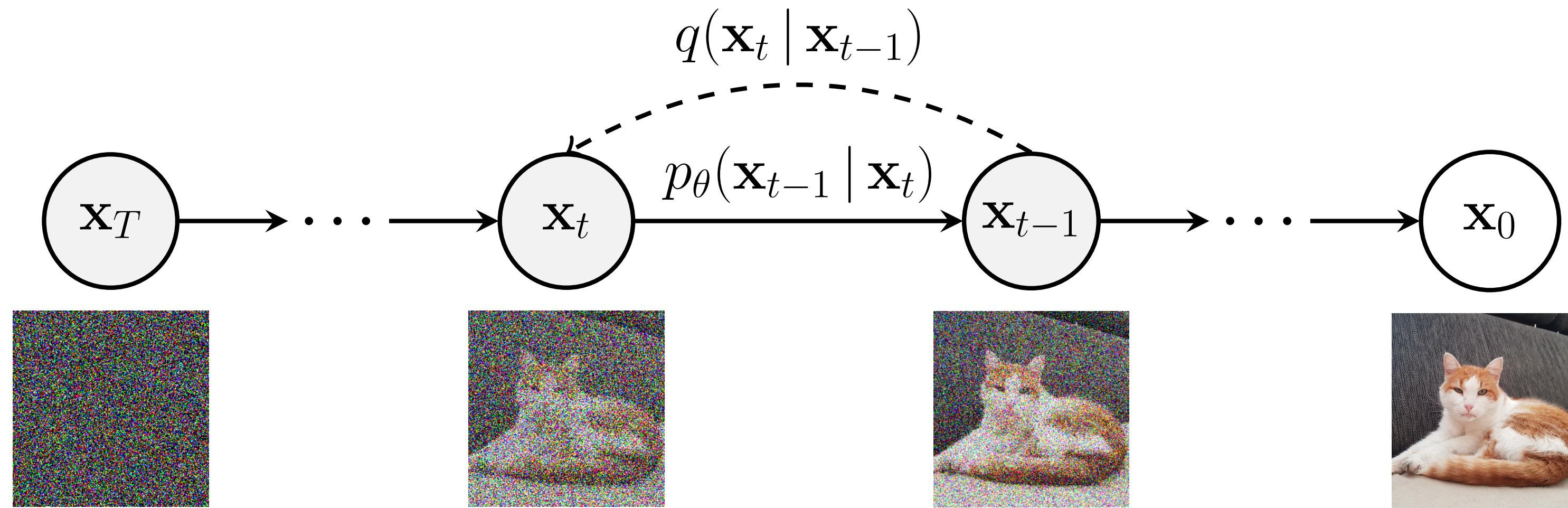
$$t = 100$$



$$\mathbf{x}_0 \sim p_\theta(\mathbf{x}_0)$$



# Algorithm: Denoising Diffusion Probabilistic Models

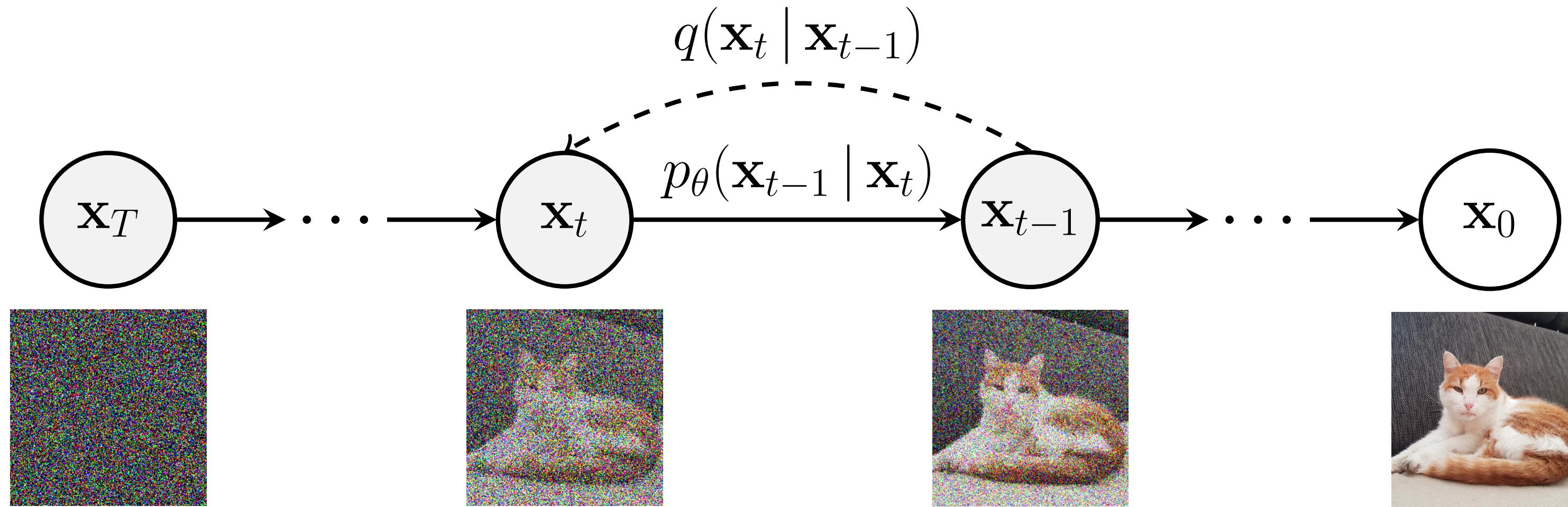


## Algorithm 1 Training

- 1: **repeat**
- 2:  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
- 3:  $t \sim \text{Uniform}(\{1, \dots, T\})$
- 4:  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 5: Take gradient descent step on  
$$\nabla_{\theta} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}, t)\|^2$$
- 6: **until converged**

Optimizing model parameters that predicts the “added noise” that was sampled, using  $\mathbf{t}$  and  $\mathbf{x}_t$

# Algorithm: Denoising Diffusion Probabilistic Models



## Algorithm 1 Training

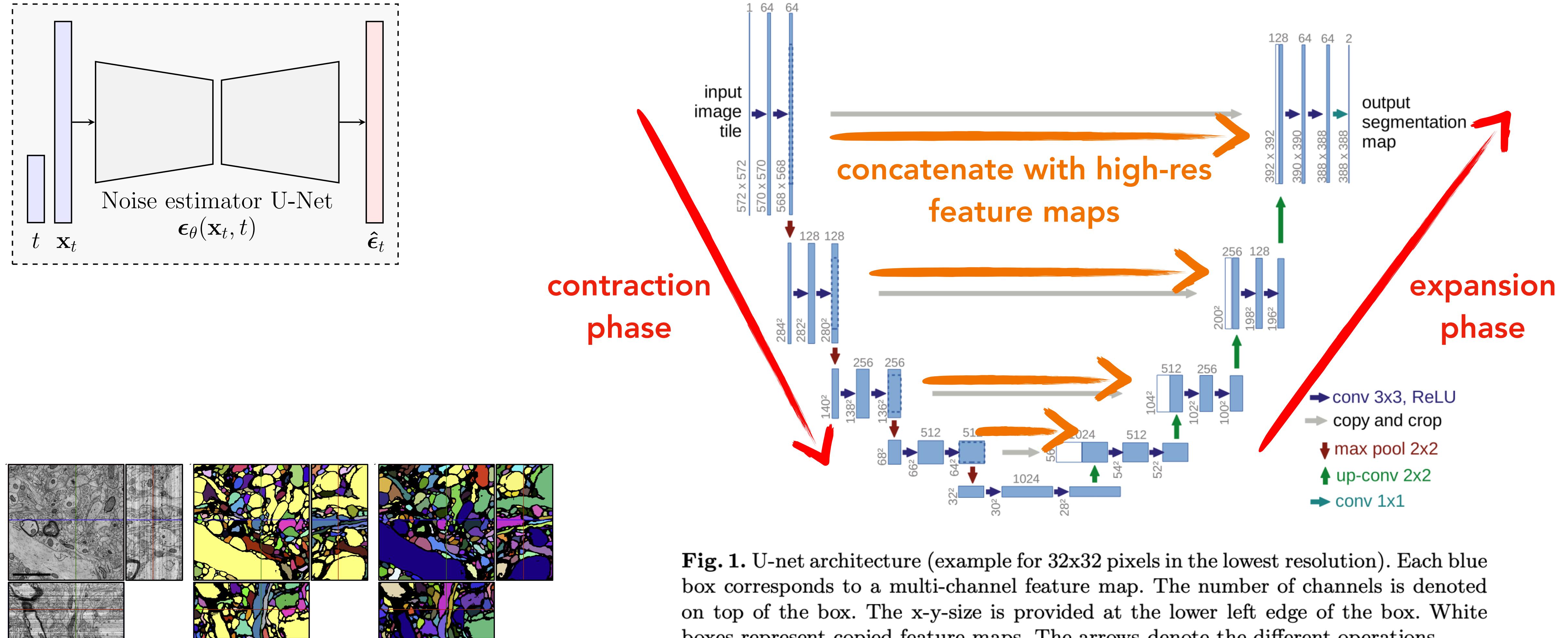
```
1: repeat
2:    $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ 
3:    $t \sim \text{Uniform}(\{1, \dots, T\})$ 
4:    $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
5:   Take gradient descent step on
         $\nabla_{\theta} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}, t)\|^2$ 
6: until converged
```

## Algorithm 2 Sampling

```
1:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$ , else  $\mathbf{z} = \mathbf{0}$ 
4:    $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_{\theta}(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$ 
5: end for
6: return  $\mathbf{x}_0$ 
```

Sequentially remove the estimated “added noise” starting from an image sampled from a Gaussian noise distribution.

# Image-to-Image Network Architecture: U-Net



# Today

---

- Deep Generative Models
  - Variational Autoencoders
  - Generative Adversarial Networks
- Denoising Diffusion Probabilistic Models
  - Idea & Training Objective
  - Sampling
- Further Extensions & Applications

# Applications: Conditional Image Diffusion

## Palette: Image-to-Image Diffusion Models

Chitwan Saharia  
Google Research, Brain Team  
Toronto, ON, Canada  
sahariac@google.com

Chris A. Lee  
Google Research  
Mountain View, CA, USA  
chrisalee@google.com

David Fleet  
Google Research, Brain Team  
Toronto, ON, Canada  
davidfleet@google.com

William Chan  
Google Research, Brain Team  
Toronto, ON, Canada  
williamchan@google.com

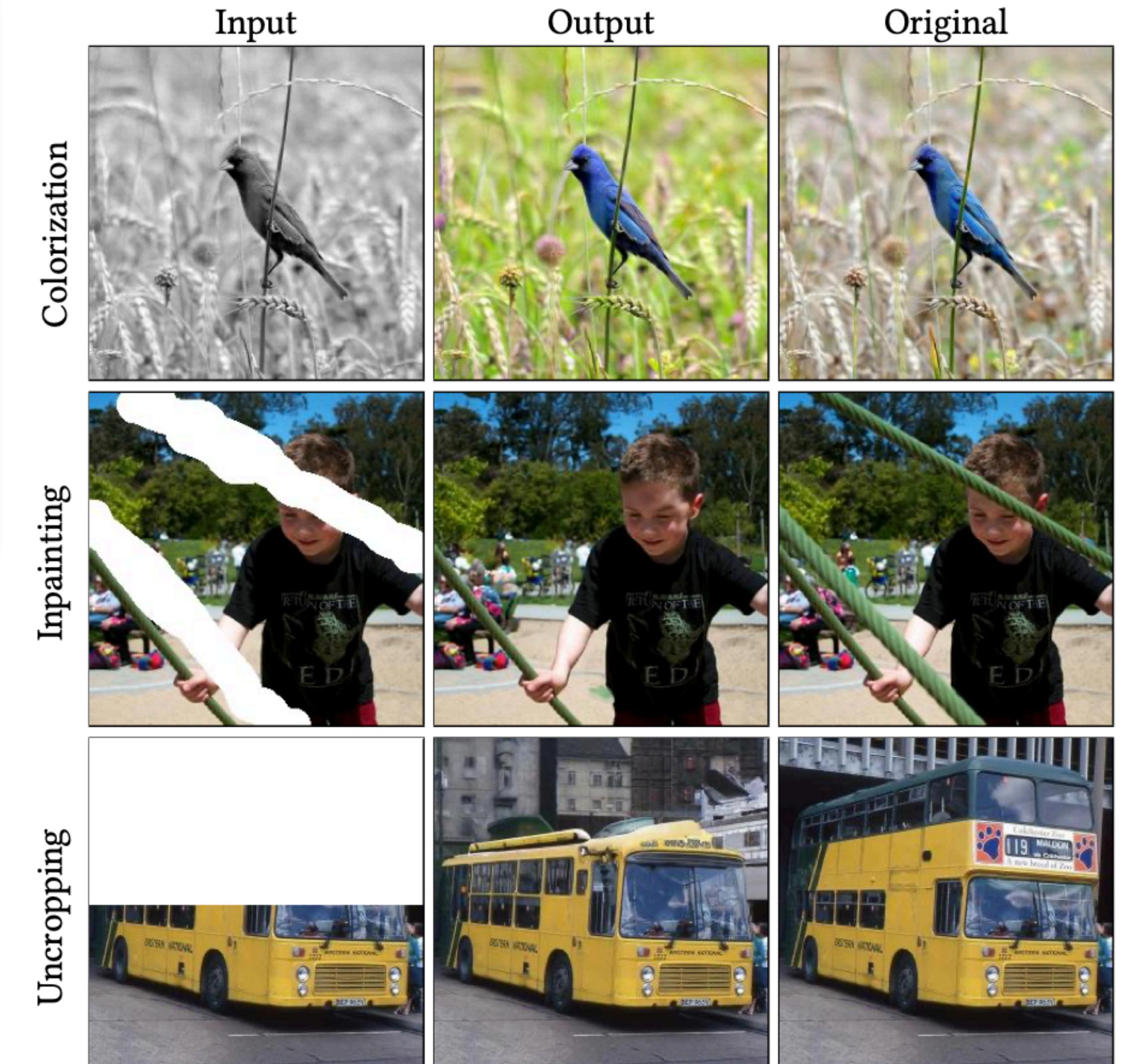
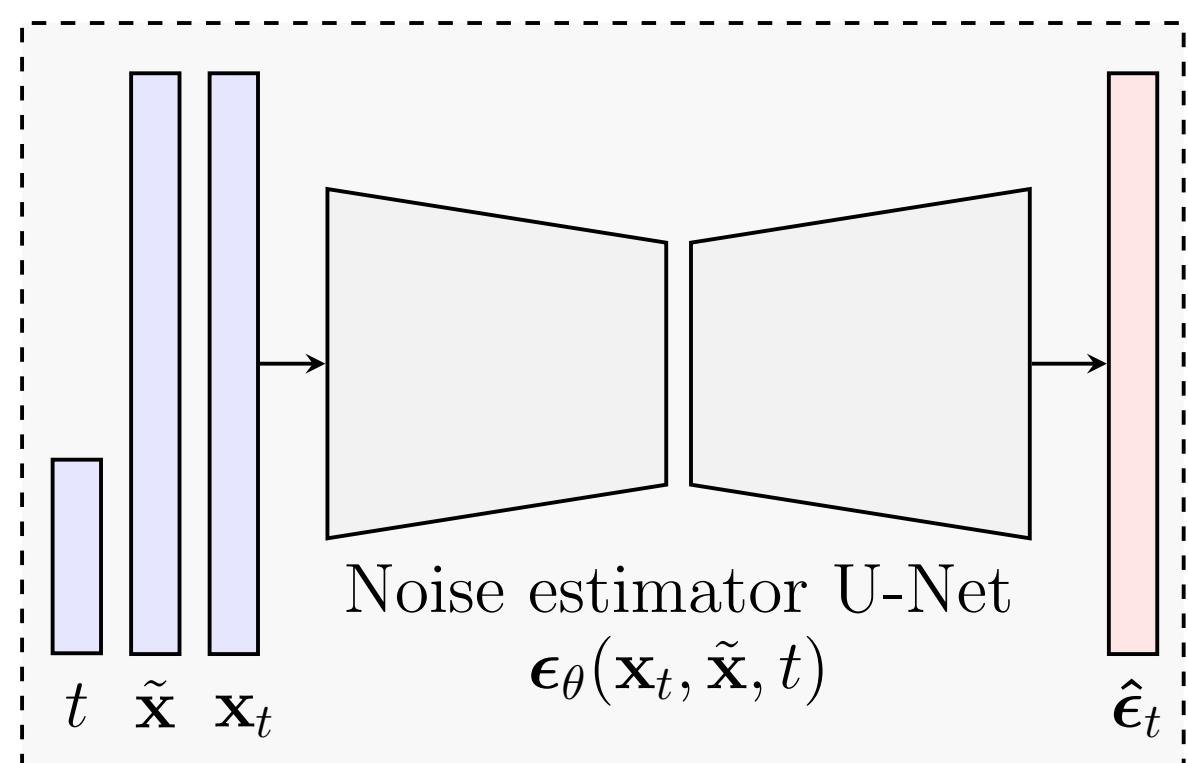
Jonathan Ho  
Google Research, Brain Team  
New York, NY, USA  
jonathanho@google.com

Mohammad Norouzi  
Google Research, Brain Team  
Toronto, ON, Canada  
mnorouzi@google.com

Huiwen Chang  
Google Research  
New York, NY, USA  
huiwenchang@google.com

Tim Salimans  
Google Research, Brain Team  
Amsterdam, Netherlands  
salimans@google.com

Training a conditional diffusion model by concatenating a conditioning image at the input.



# Applications: Conditional Image Diffusion

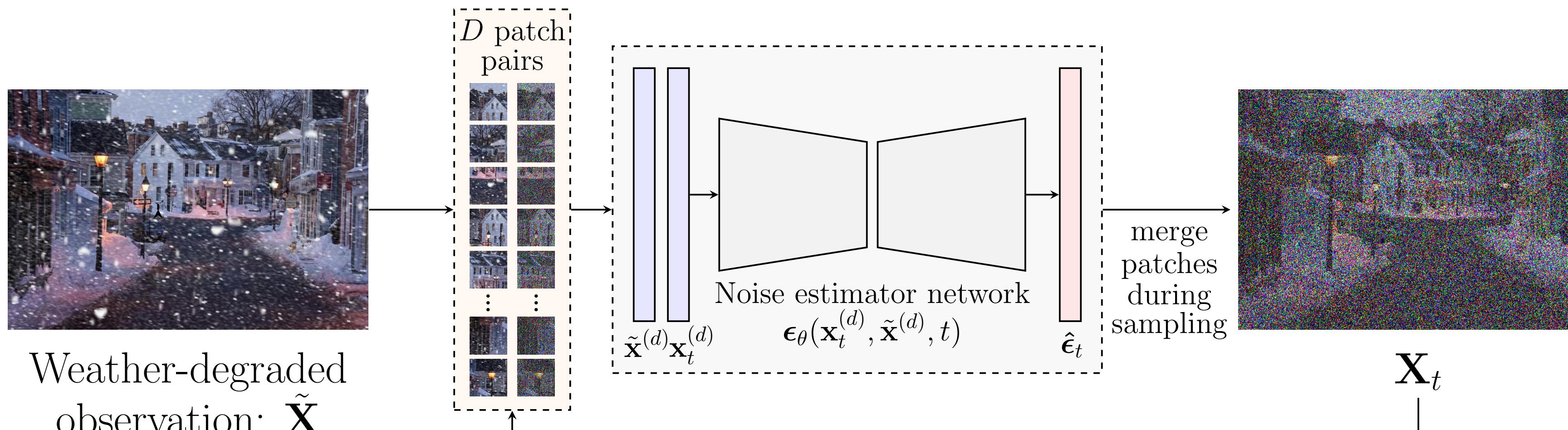


## Restoring Vision in Adverse Weather Conditions with Patch-Based Denoising Diffusion Models

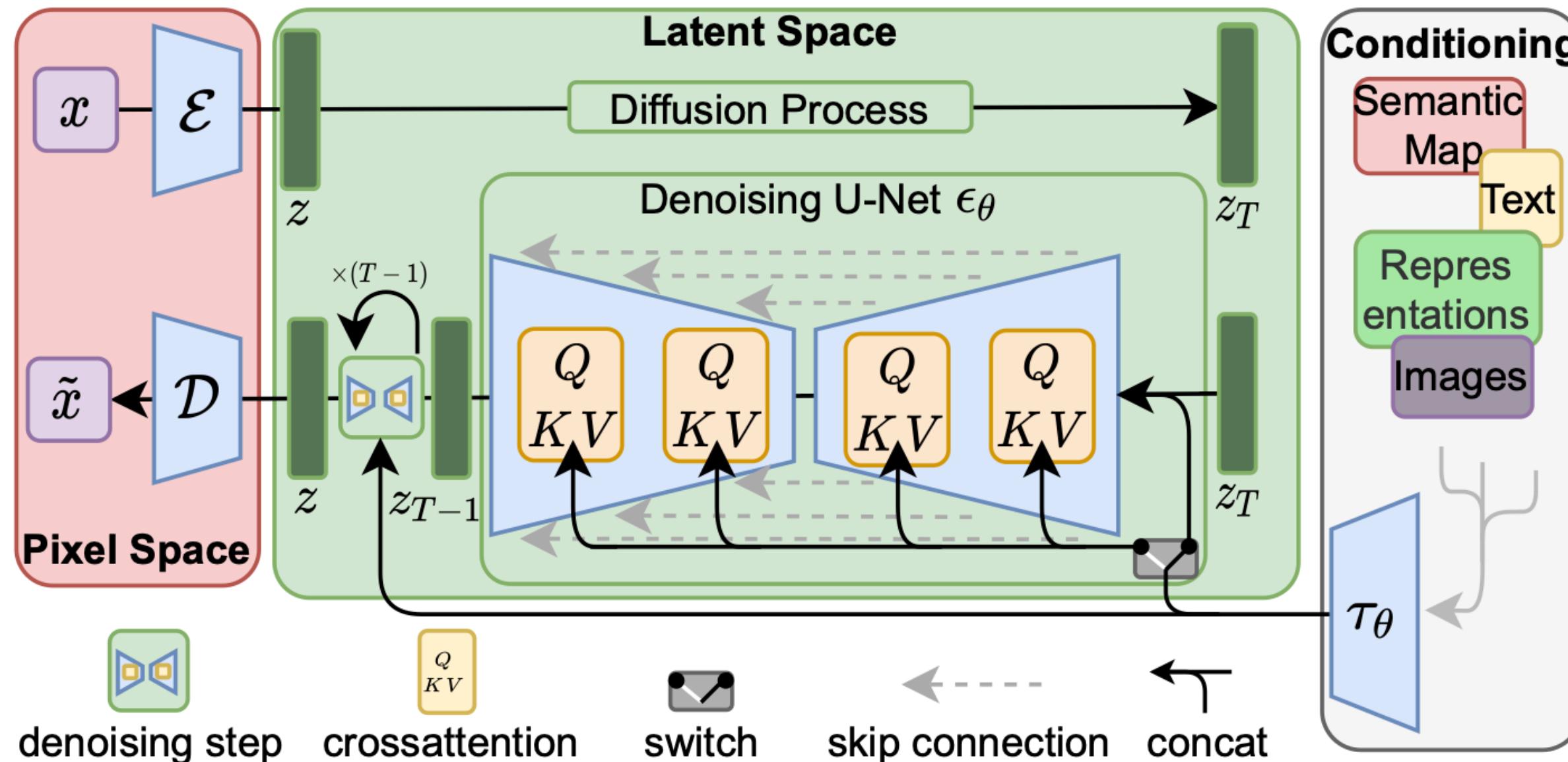
Ozan Özdenizci and Robert Legenstein

**Abstract**—Image restoration under adverse weather conditions has been of significant interest for various computer vision applications. Recent successful methods rely on the current progress in deep neural network architectural designs (e.g., with vision transformers). Motivated by the recent progress achieved with state-of-the-art conditional generative models, we present a novel patch-based image restoration algorithm based on denoising diffusion probabilistic models. Our patch-based diffusion modeling approach enables size-agnostic image restoration by using a guided denoising process with smoothed noise estimates across overlapping patches during inference. We empirically evaluate our model on benchmark datasets for image desnowing, combined deraining and dehazing, and raindrop removal. We demonstrate our approach to achieve state-of-the-art performances on both weather-specific and multi-weather image restoration, and qualitatively show strong generalization to real-world test images.

**Index Terms**—denoising diffusion models, patch-based image restoration, deraining, desnowing, dehazing, raindrop removal.



# Applications: Text-to-Image Diffusion Models



“a teddy bear in a suit riding a bike on the streets of Graz”  
by Stable Diffusion

# More recent applications...

- Improvements in higher resolution image synthesis
- Generating and editing 3D shapes
- Video diffusion models
- Text-to-music diffusion models (e.g., Riffusion, Moûsai, ...)
- ...

## SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis

Dustin Podell   Zion English   Kyle Lacey   Andreas Blattmann   Tim Dockhorn

Jonas Müller

Joe Penna

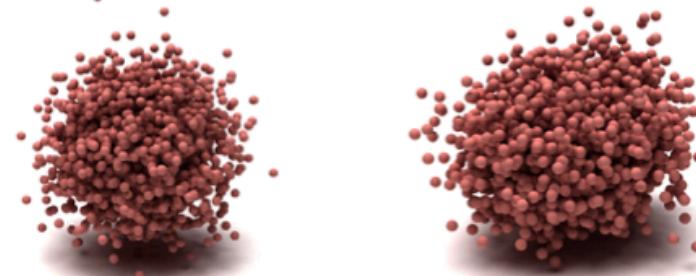
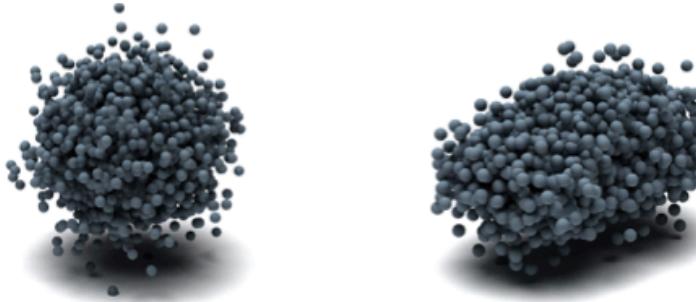
Robin Rombach

Stability AI, Applied Research

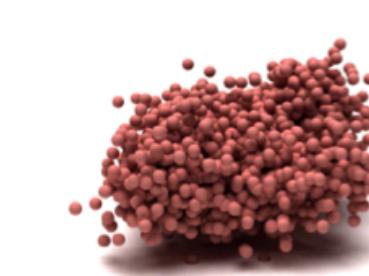


## 3D Shape Generation and Completion through Point-Voxel Diffusion

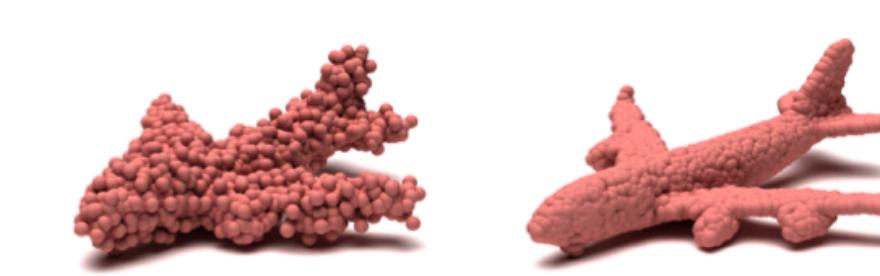
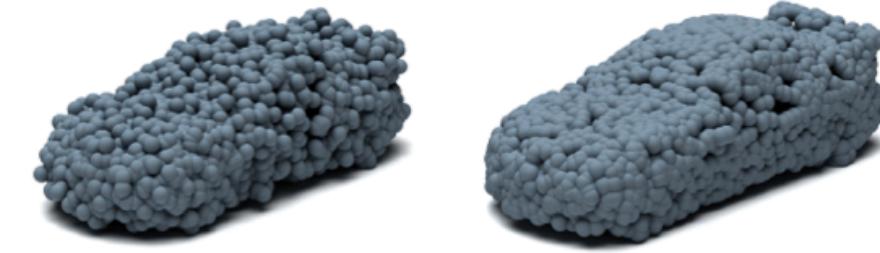
Linqi Zhou  
Stanford University



Yilun Du  
MIT



Jiajun Wu  
Stanford University



## Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets

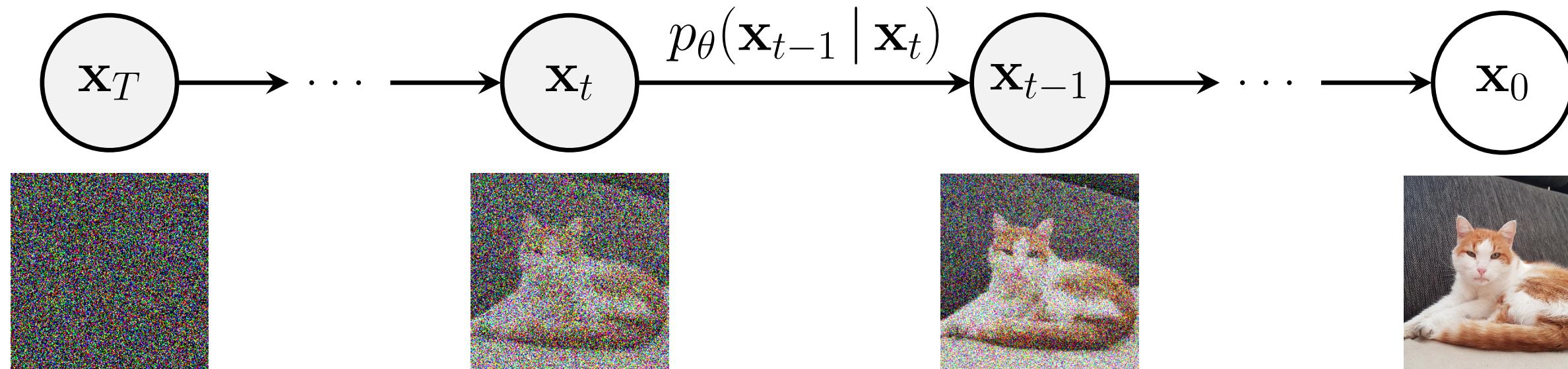
Andreas Blattmann\*   Tim Dockhorn\*   Sumith Kulal\*   Daniel Mendelevitch  
Maciej Kilian   Dominik Lorenz   Yam Levi   Zion English   Vikram Voleti  
Adam Letts   Varun Jampani   Robin Rombach  
Stability AI



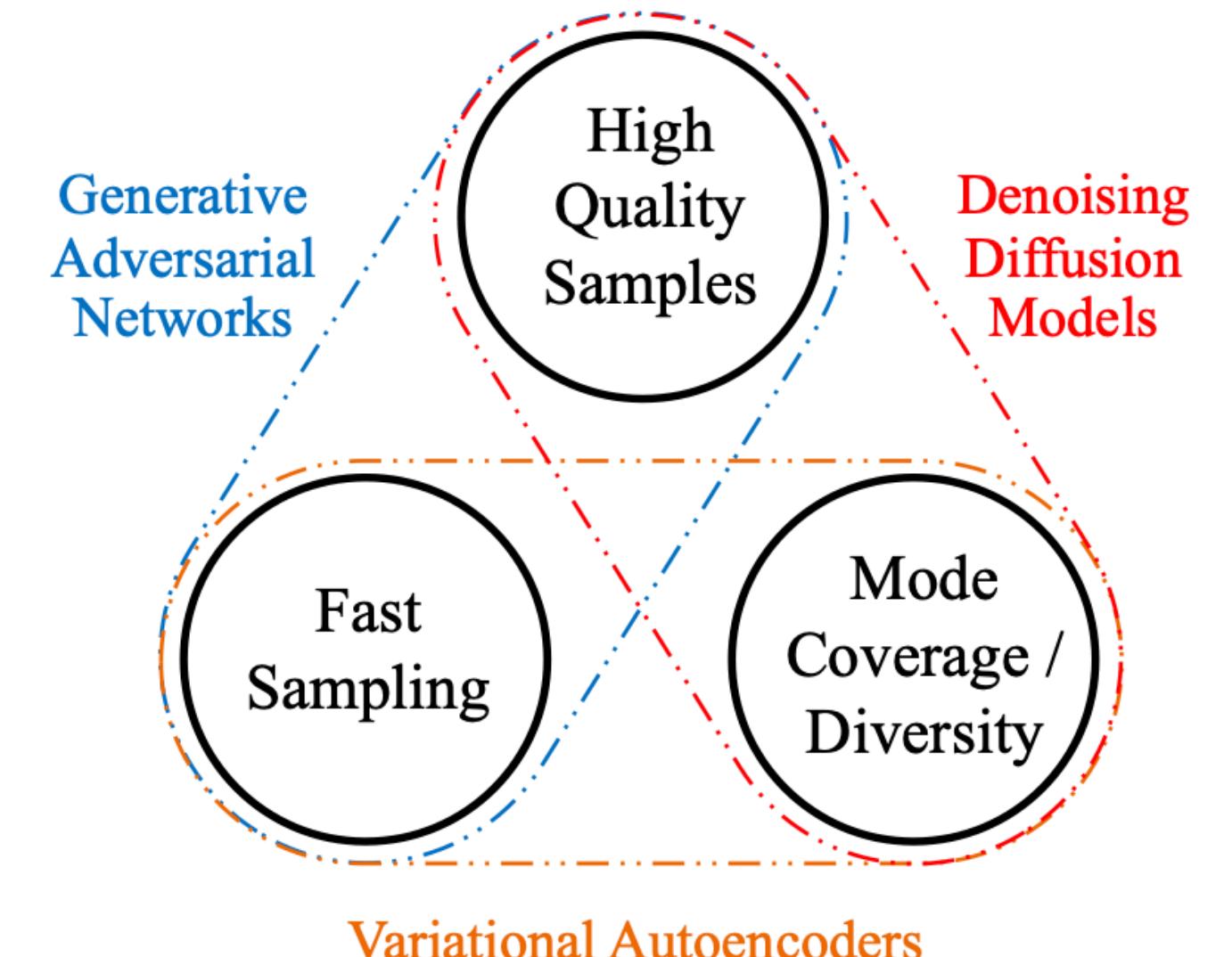
"A robot dj is playing the turntables, in heavy raining futuristic tokyo, rooftop, sci-fi, fantasy"

# Summary

- ▶ Iteratively **denoising** an image with a neural network by inverting a **diffusion** process which destroys the data structure by gradually adding noise.
- ▶ Denoising diffusion models are easy to train:  $\mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[ \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t)\|^2 \right]$
- ▶ Image-to-image (U-Net) or vector-to-vector (latent-space) network architectures can be used.
- ▶ Sampling from a diffusion model is relatively slow.  
(Note: recent accelerated sampling techniques exist.)



## The Generative Learning Trilemma



- ▶ State-of-the-art: More powerful than existing generative models.

# Today

---

Deep Generative Models

Variational Autoencoders

Generative Adversarial Networks

Denoising Diffusion Probabilistic Models

Idea & Training Objective

Sampling

Further Extensions & Applications



*"audience raising hands for questions during a lecture"*  
by Stable Diffusion

# Questions?