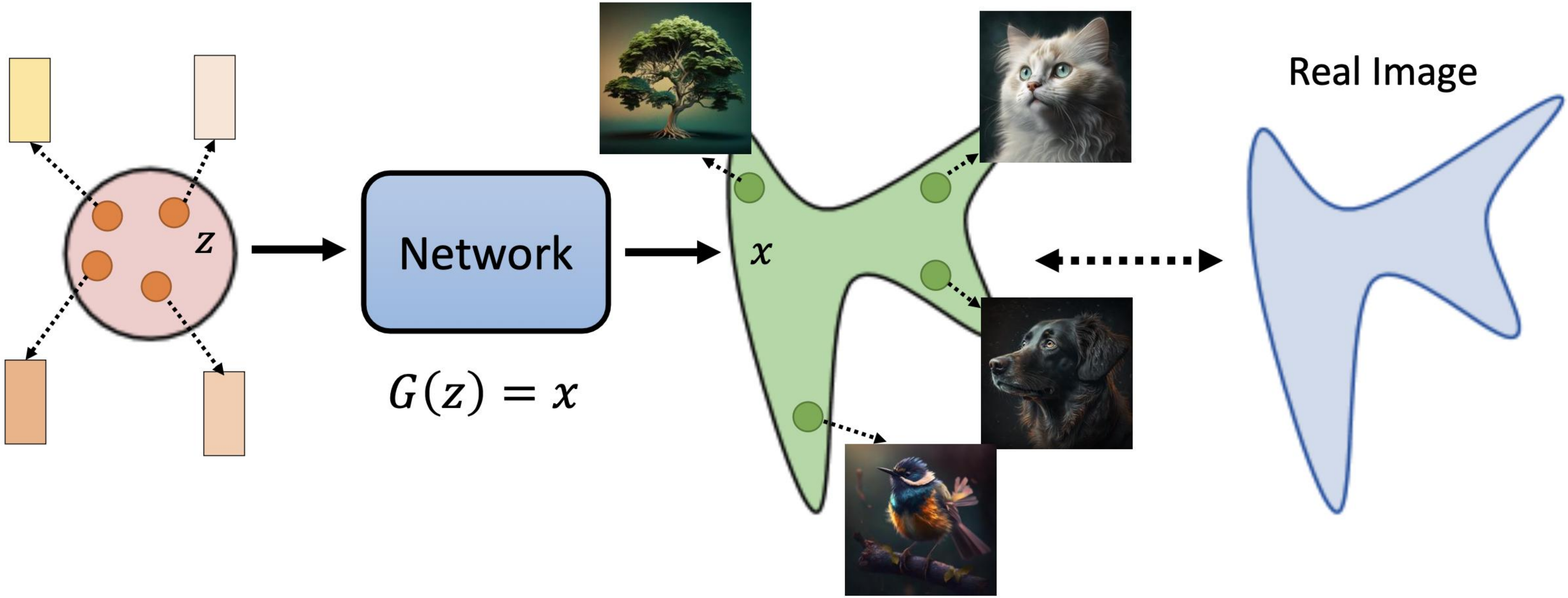
Diffusion Models and Scientific Generative AI

—— Lecture 4 & 5: Fundamentals of Diffusion Models

Professor Siyu Zhu
AI3 institute, Fudan University

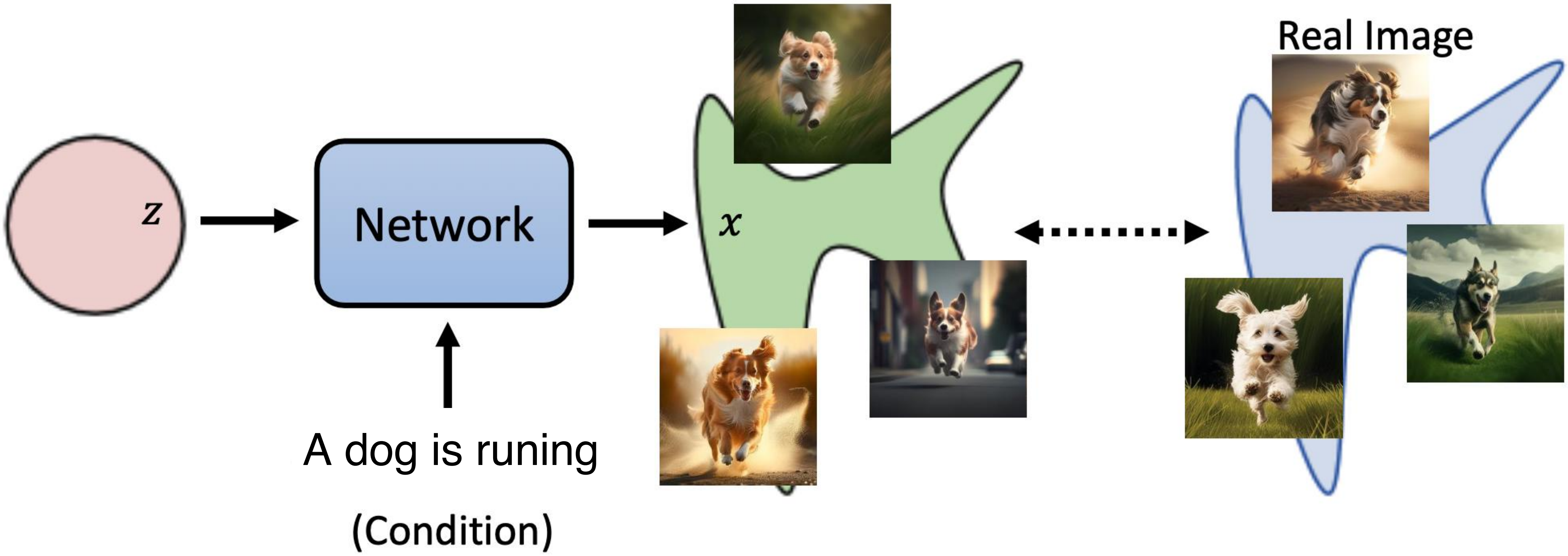
Contents

Generative AI Model



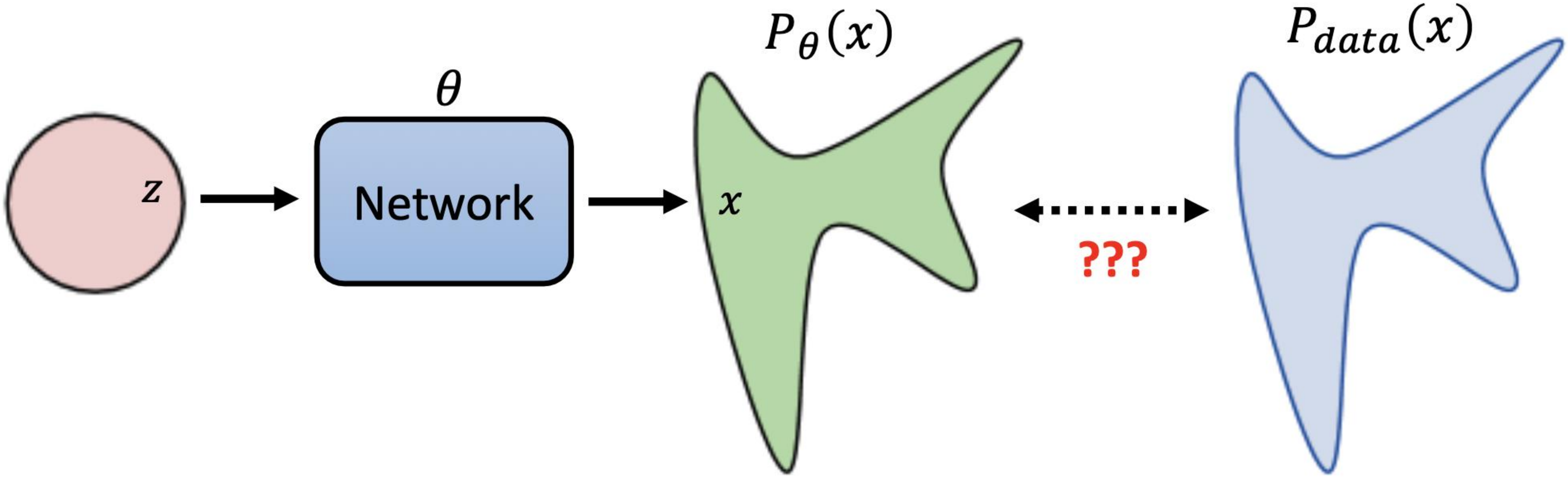
slide from https://speech.ee.ntu.edu.tw/~hylee/ml/2023-spring.php

Generative AI Model



slide from https://speech.ee.ntu.edu.tw/~hylee/ml/2023-spring.php

Generative AI Model



Sample $\{x^1, x^2, \dots, x^m\}$ from $P_{data}(x)$

We can compute $P_\theta(x^i)$

???

$$\theta^* = arg \max_\theta \prod_{i=1}^{m} P_\theta(x^i)$$

slide from https://speech.ee.ntu.edu.tw/~hylee/ml/2023-spring.php

Generative AI Model

Sample $\{x^1, x^2, \ldots, x^m\}$ from $P_{data}(x)$

$$\theta^* = arg\max_\theta \prod_{i=1}^{m} P_\theta(x^i) \quad = arg\max_\theta log \prod_{i=1}^{m} P_\theta(x^i)$$

$$= arg\max_\theta \sum_{i=1}^{m} logP_\theta(x^i) \quad \approx arg\max_\theta E_{x\sim P_{data}}[logP_\theta(x)]$$

(not related to θ)

$$= arg\max_\theta \int_x P_{data}(x) logP_\theta(x)dx \quad - \int_x P_{data}(x) logP_{data}(x)dx$$

Difference between P_{data} and P_θ

$$= arg\max_\theta \int_x P_{data}(x) log\frac{P_\theta(x)}{P_{data}(x)}dx \quad = arg\min_\theta KL(P_{data}||P_\theta)$$

Maximum Likelihood = Minimize KL Divergence

slide from https://speech.ee.ntu.edu.tw/~hylee/ml/2023-spring.php

Supervised vs Unsupervised Learning

Supervised Learning

Data: (x, y)
x is data, y is label

Goal: Learn a function to map x -> y

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.

Unsupervised Learning

Data: x
Just data, no labels!

Goal: Learn some underlying hidden structure of the data

Examples: Clustering, dimensionality reduction, density estimation, etc.

Self-Supervised Learning

Data: (x, pseudo generated y)
No manual labels!

Goal: Learn to generate good features (reduce the data to useful/generic features)

Example: Classification in downstream applications where we have limited data

slide from https://cs231n.stanford.edu/slides/2024/lecture_13.pdf

Autoregressive Models

- PixelCNN [van der Oord et al. 2016]

Still generate image pixels starting from corner

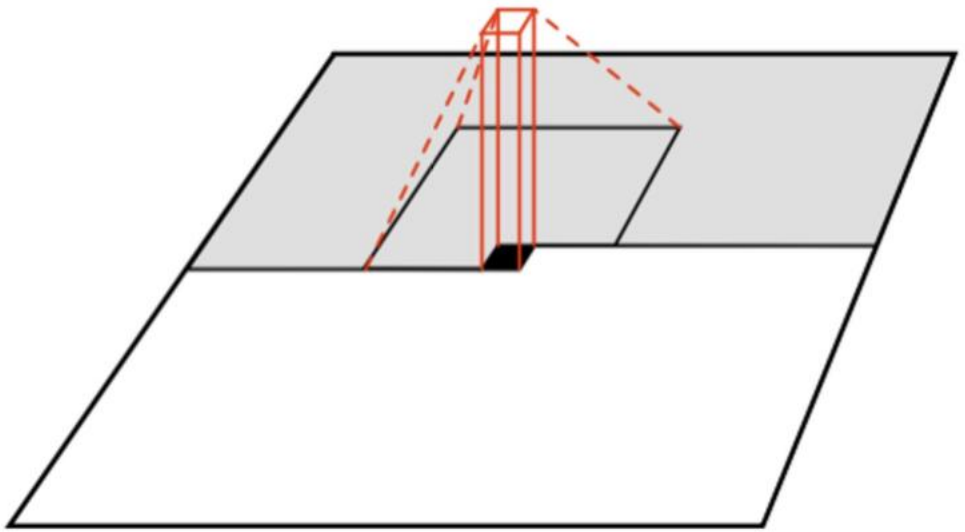
Dependency on previous pixels now modeled using a CNN over context region

(masked convolution)

$$p(x) = p(x_1, x_2, \ldots, x_n)$$

Likelihood of image x

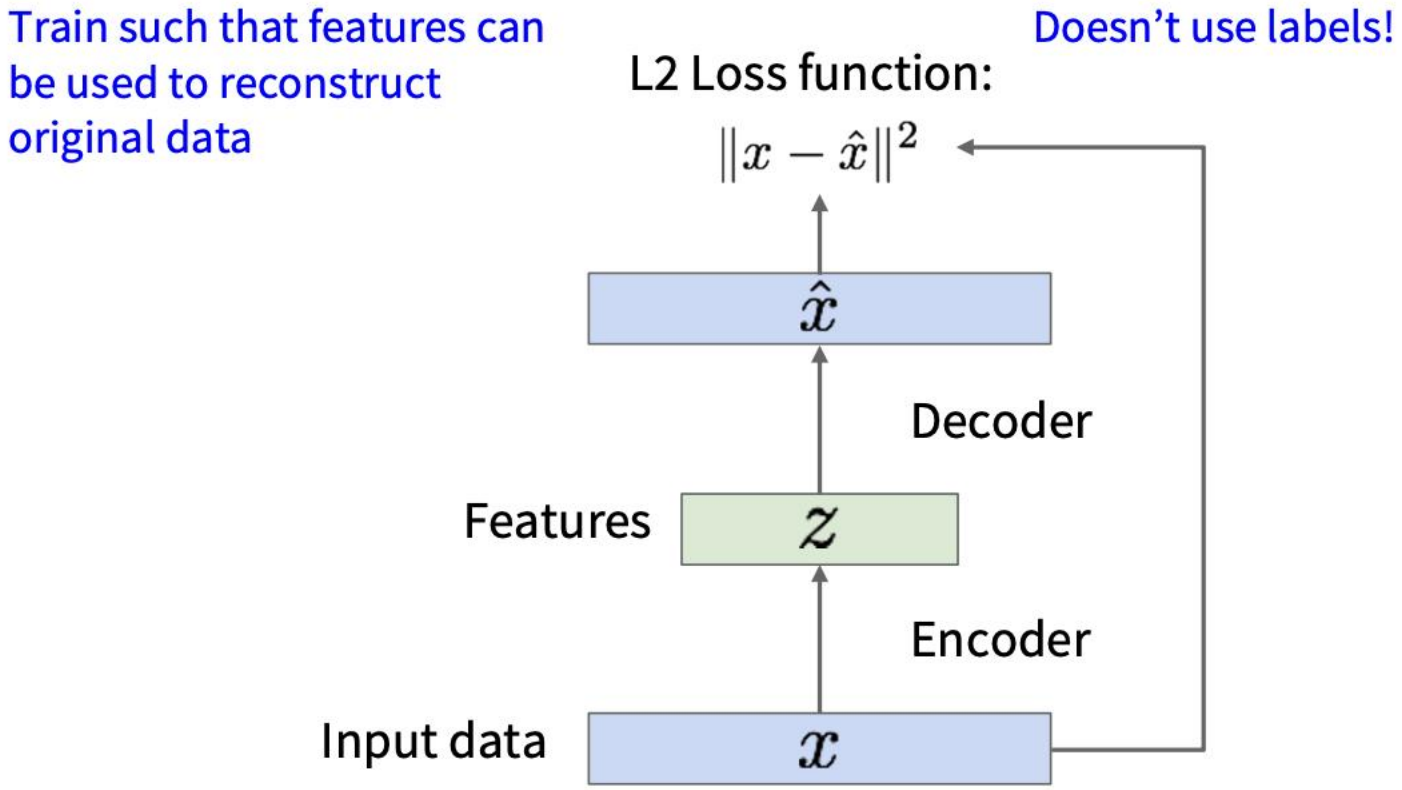
Joint likelihood of each pixel in the image

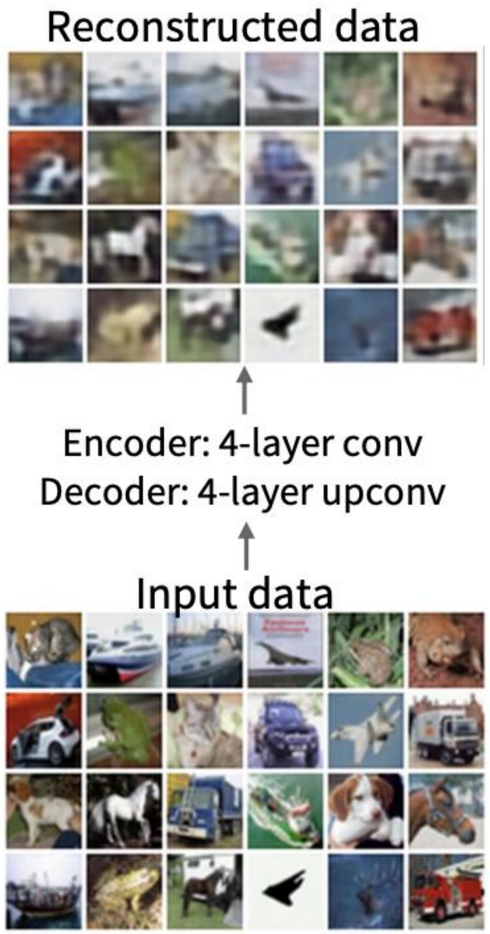


slide from https://cs231n.stanford.edu/slides/2024/lecture_13.pdf

Variational Autoencoder (VAE)

- Train such that features can be used to reconstruct original data





slide from https://cs231n.stanford.edu/slides/2024/lecture_13.pdf

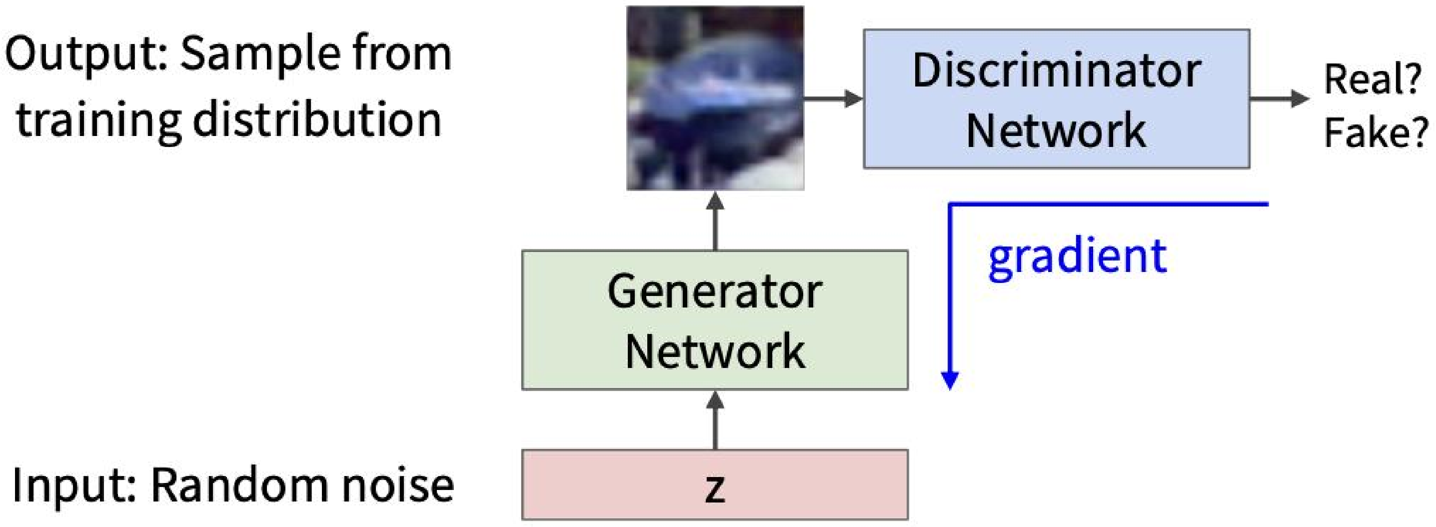
Generative Adversarial Networks (GAN)

Problem: Want to sample from complex, high-dimensional training distribution. No direct way to do this!

Solution: Sample from a simple distribution we can easily sample from, e.g. random noise. Learn transformation to training distribution.

But we don't know which sample z maps to which training image -> can't learn by reconstructing training images

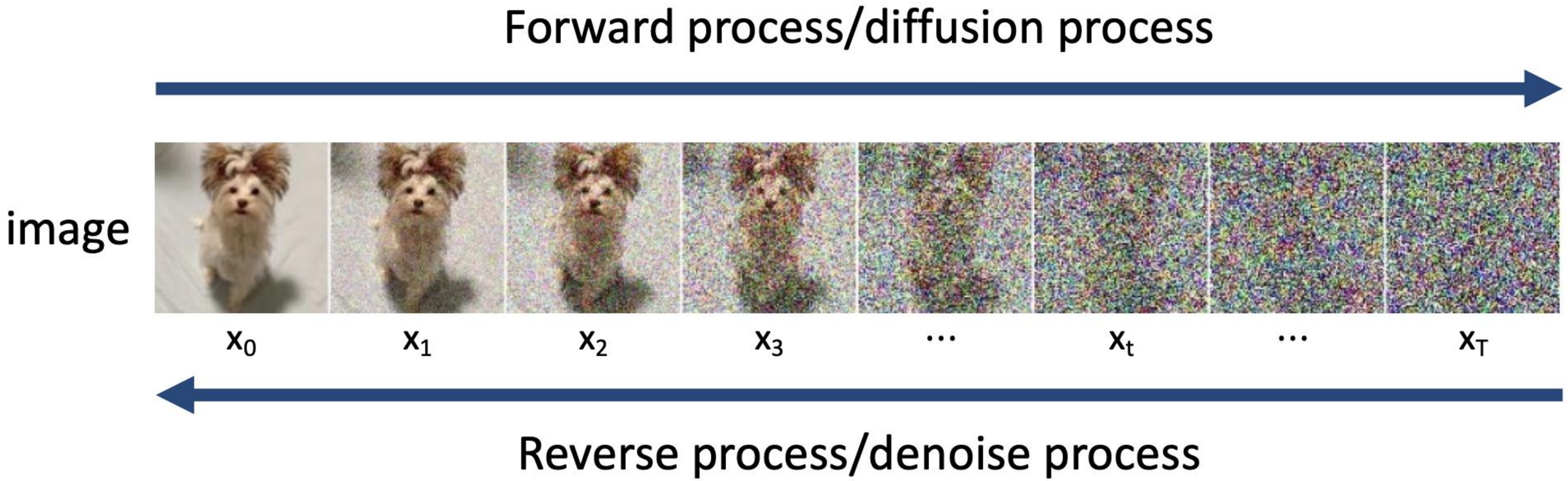
Solution: Use a discriminator network to tell whether the generate image is within data distribution ("real") or not



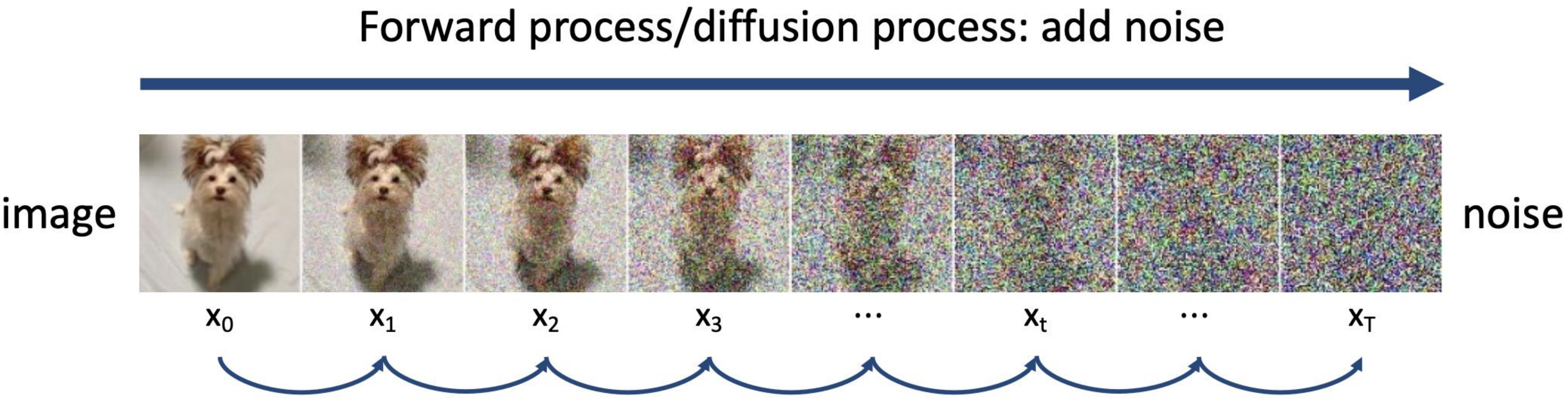
slide from https://cs231n.stanford.edu/slides/2024/lecture_13.pdf

Contents

DDPM (Denoising Diffusion Probabilistic Models)



DDPM (Denoising Diffusion Probabilistic Models)

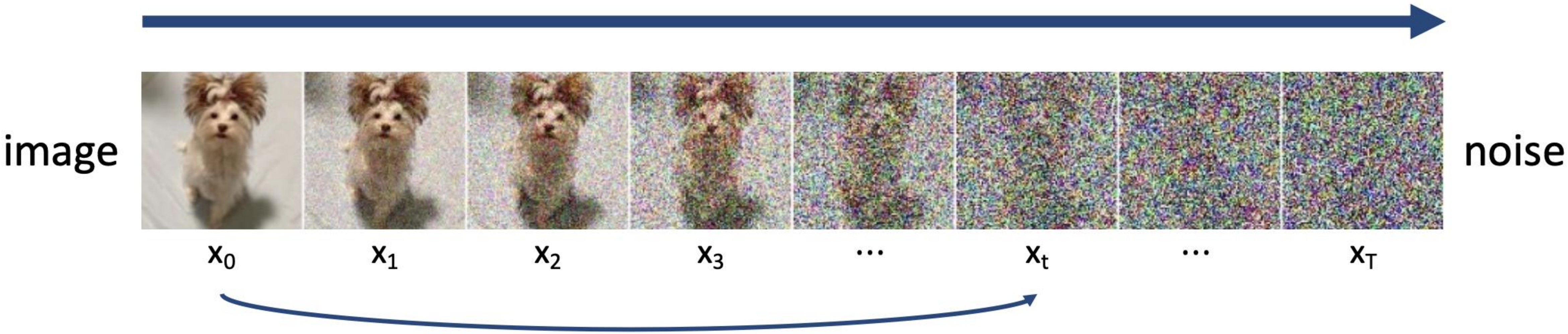


$$\Pr(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\left(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}\right)$$

β_t is the variance (strength) of noise

DDPM (Denoising Diffusion Probabilistic Models)

Forward process/diffusion process: add noise

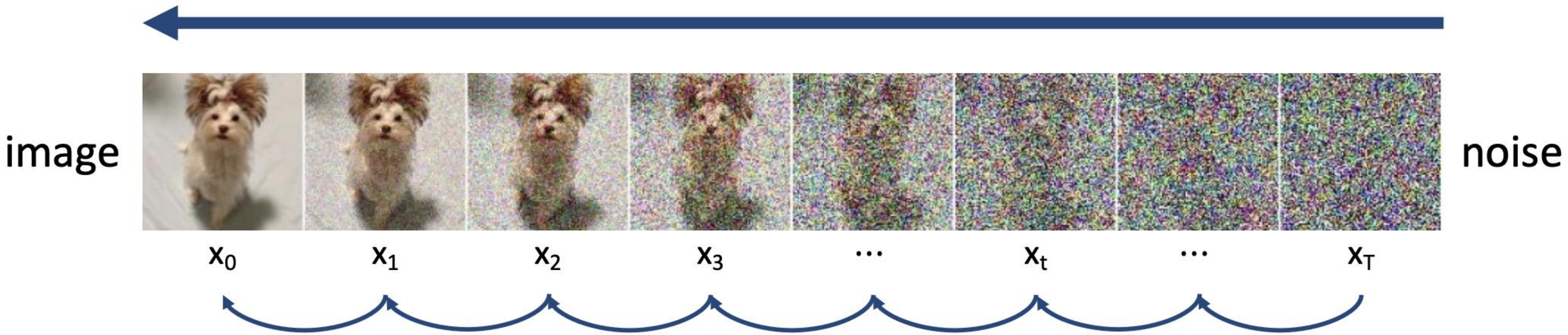


$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$$

DDPM (Denoising Diffusion Probabilistic Models)

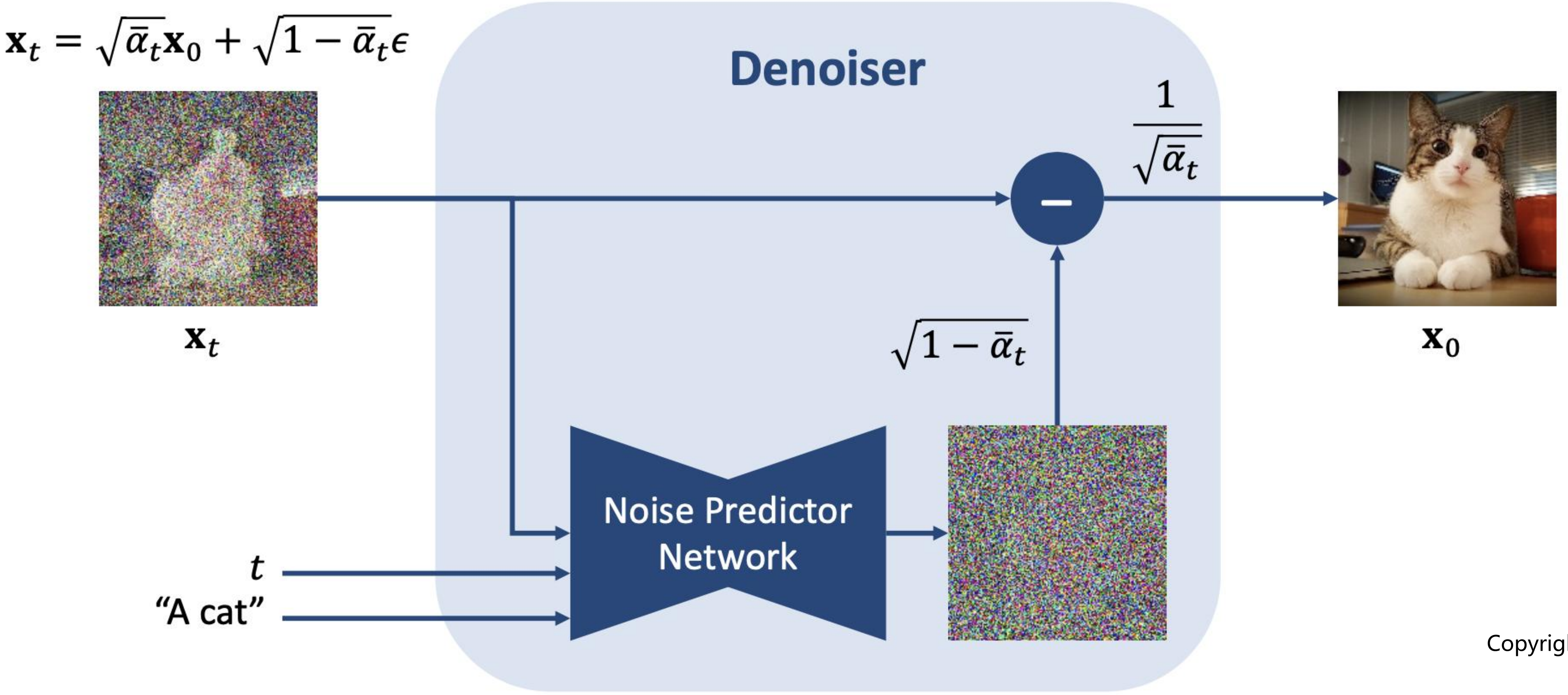
Reverse process/denoise process: remove noise



Train a neural network model to remove noise gradually

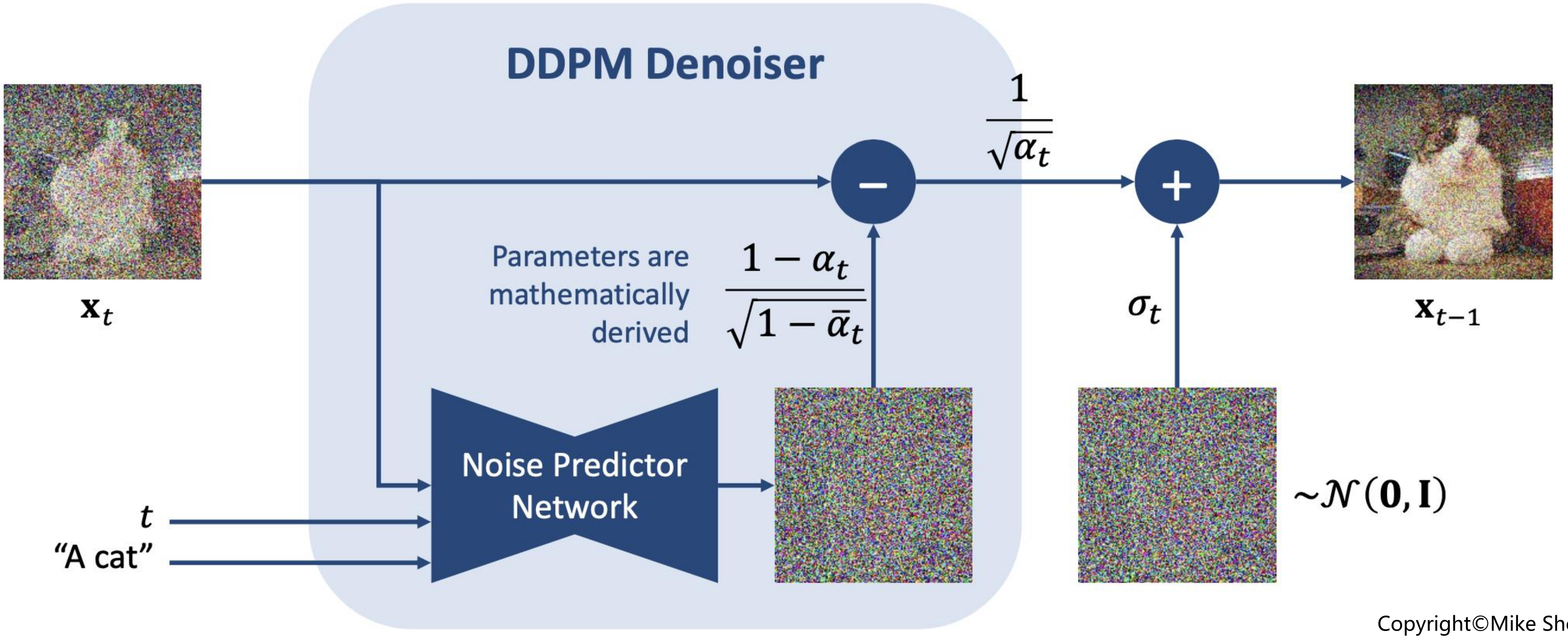
DDPM (Denoising Diffusion Probabilistic Models)

Training objective: one-step predict the noise w.r.t. the original image

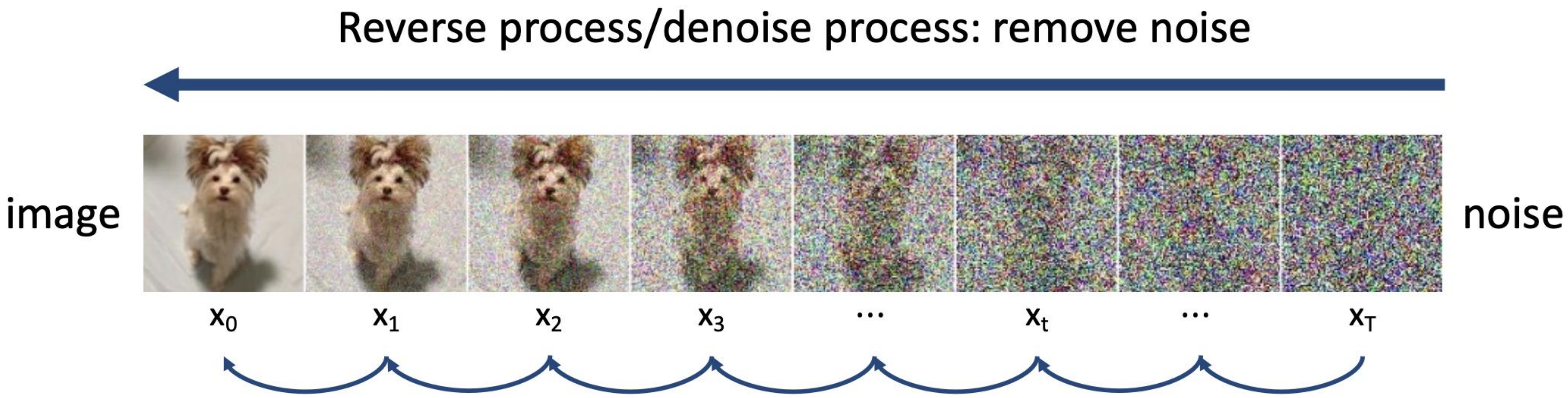


DDPM (Denoising Diffusion Probabilistic Models)

During generation: denoise step-by-step, in each step, add noise after noise removal



DDIM (Denoising Diffusion Implicit Models)

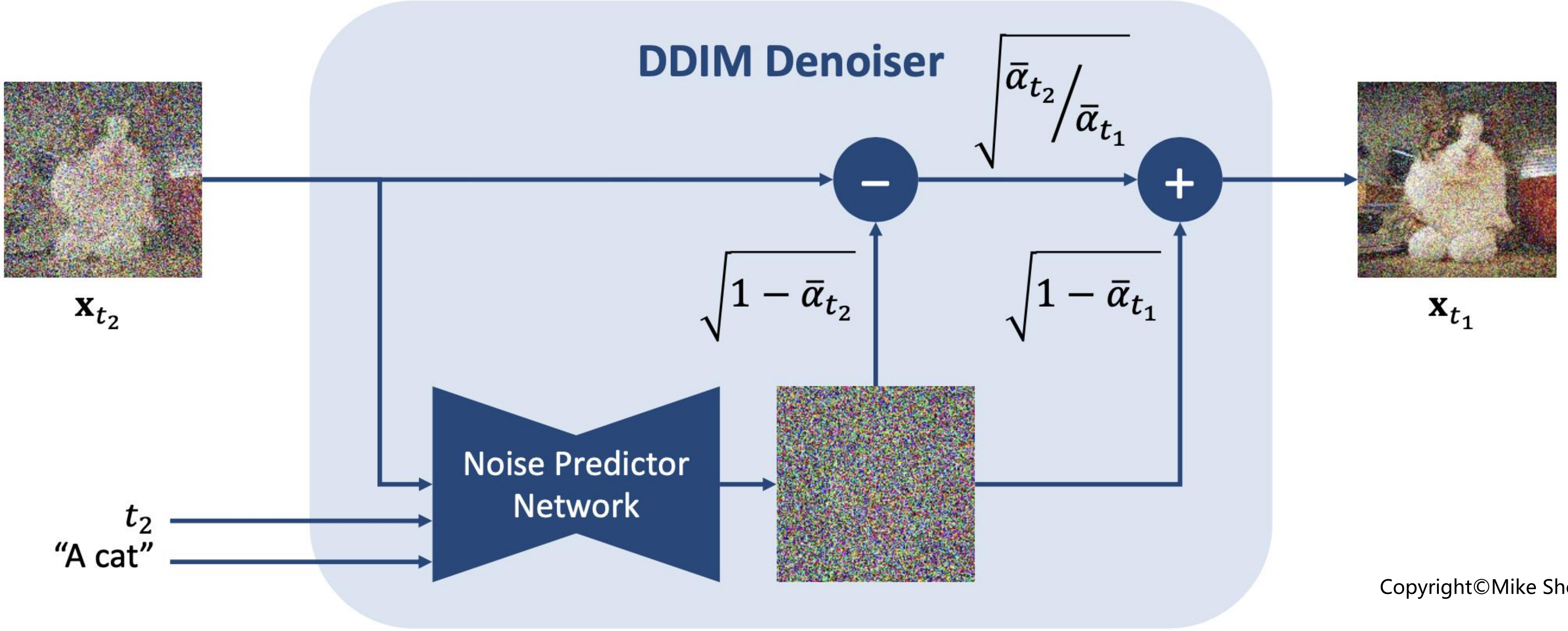


Train a neural network model to remove noise gradually

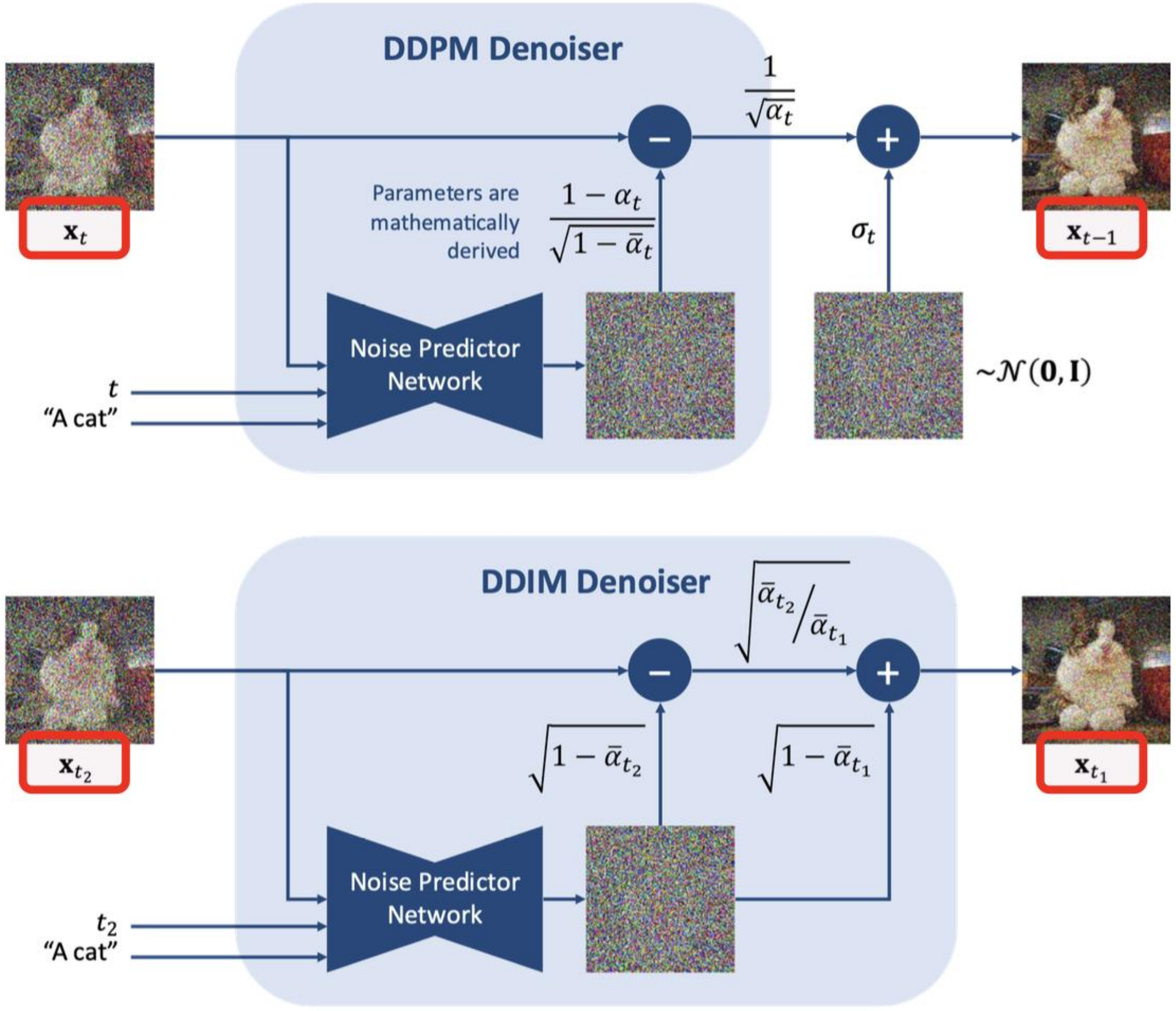
Speed up? Skip steps during denoising!

DDIM (Denoising Diffusion Implicit Models)

During generation: can skip steps from t_2 directly to t_1



DDPM vs. DDIM



DDPM cannot skip timesteps

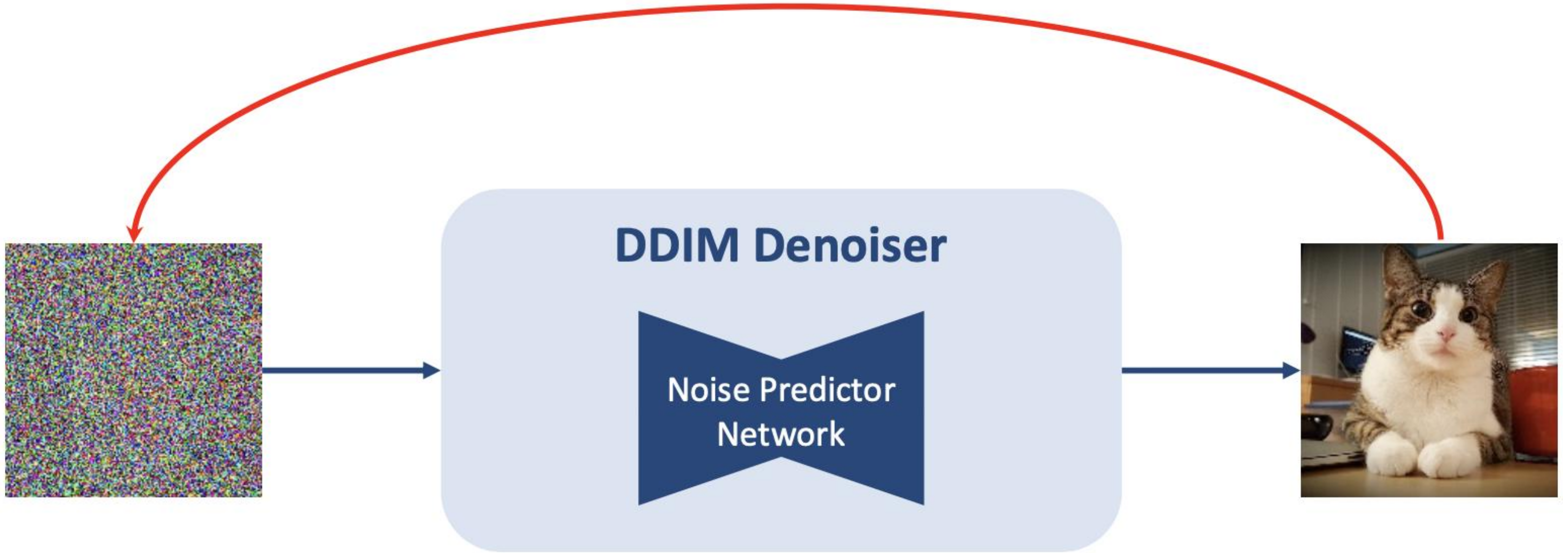
A few hundreds steps to generate an image

DDIM can skip timesteps

Say 50 steps to generate an image

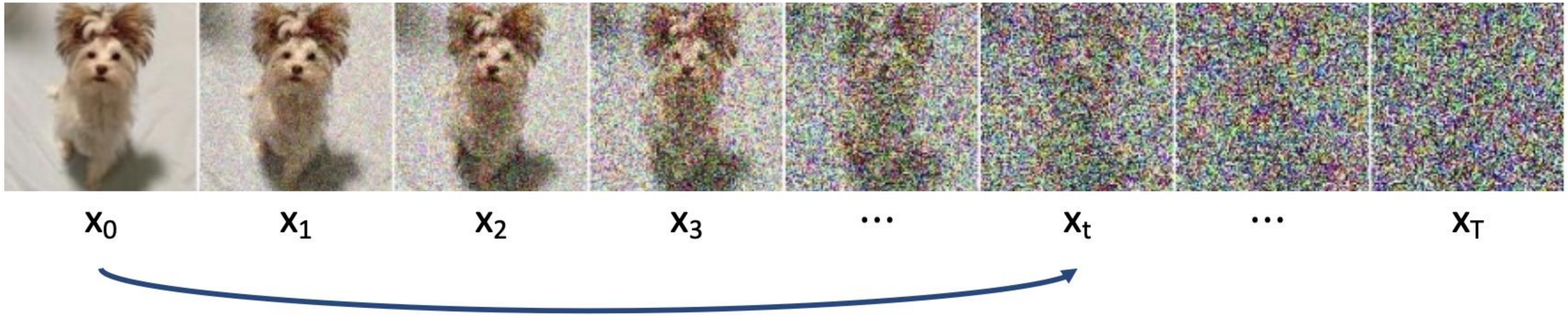
DDIM (Denoising Diffusion Implicit Models)

Given an image & trained denoiser, find out which initial noise can be denoised into this image?



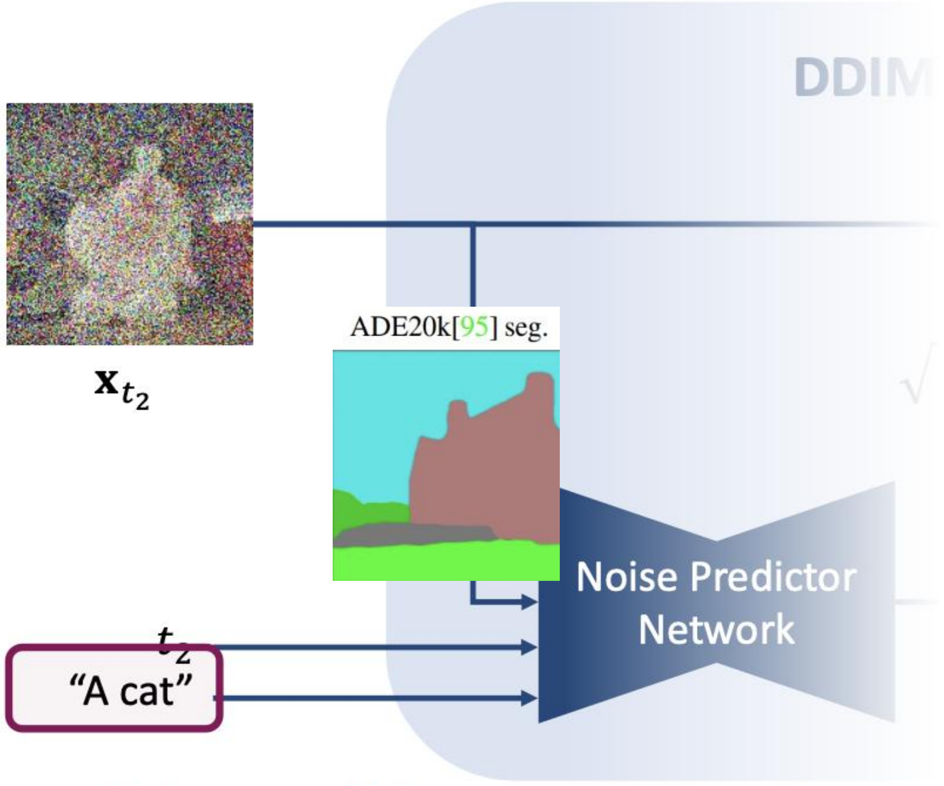
DDIM (Denoising Diffusion Implicit Models)

Forward Diffusion Process: Add $\mathcal{N}(0, \mathbf{I})$ Noise

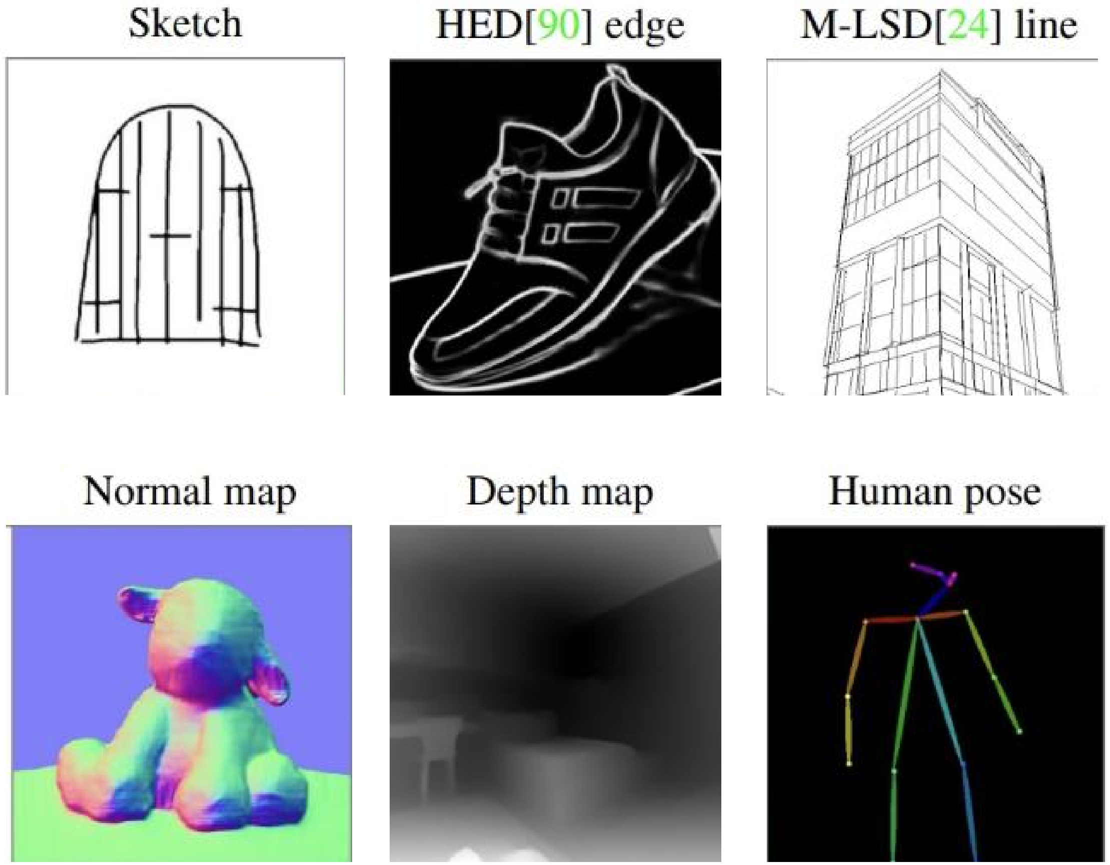


DDIM Inversion Process: Add Noise inverted by the trained DDIM denoiser

Conditional Generation: Explicit conditions

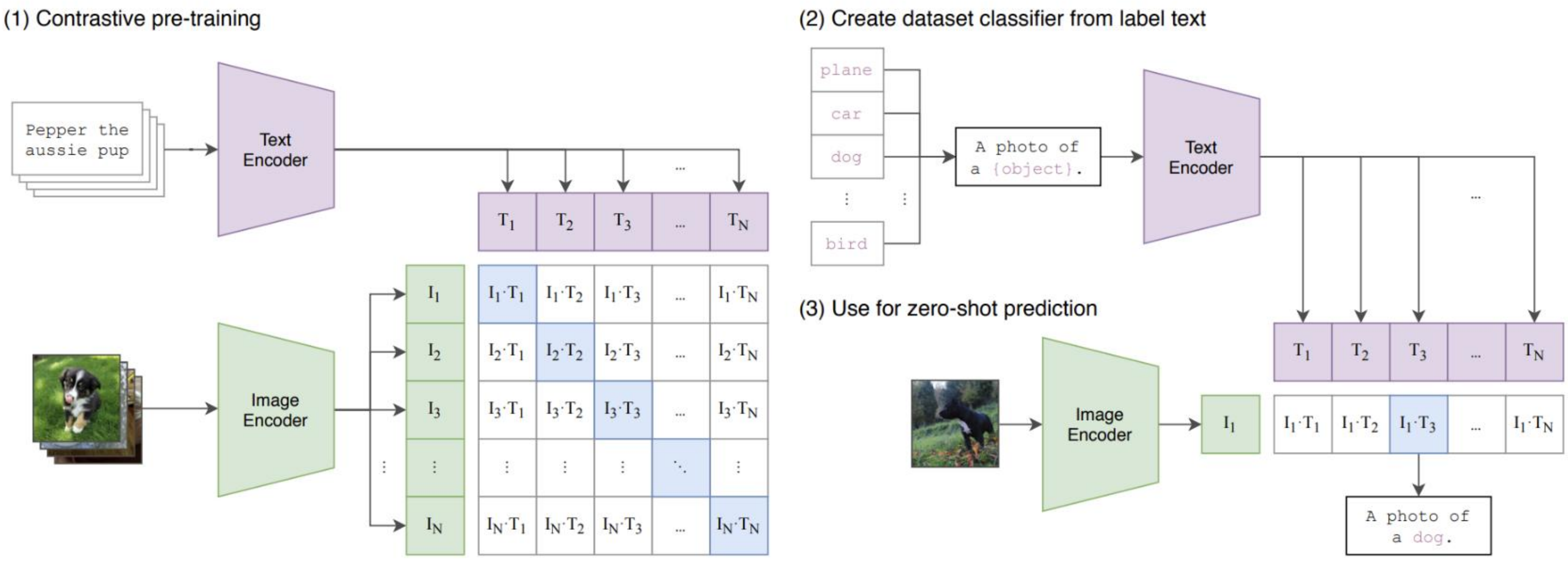


explicit conditions



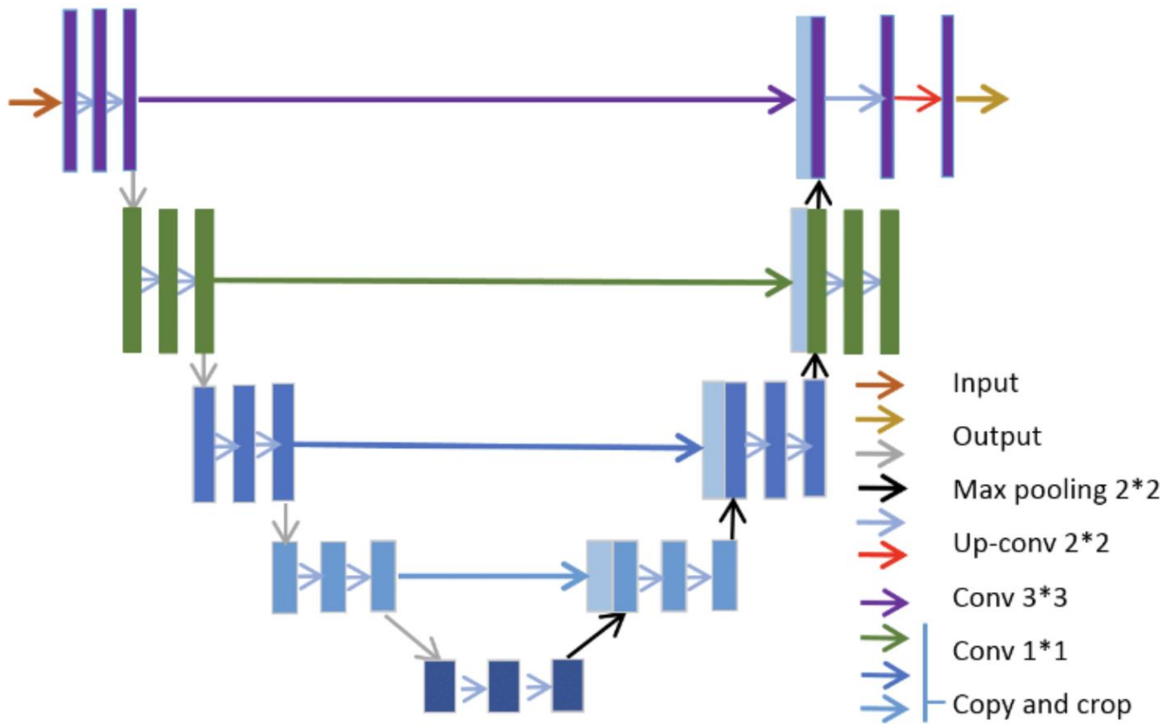
CLIP: Encoders bridge vision and language

- CLIP text-/image-embeddings are commonly used in diffusion models for conditional generation

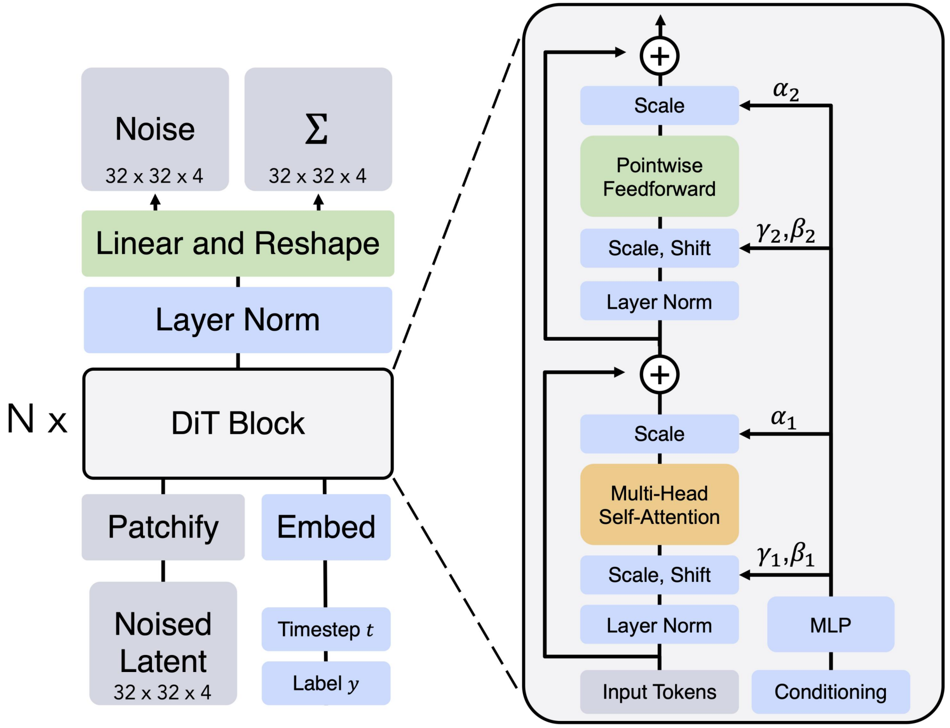


Denoiser

Unet



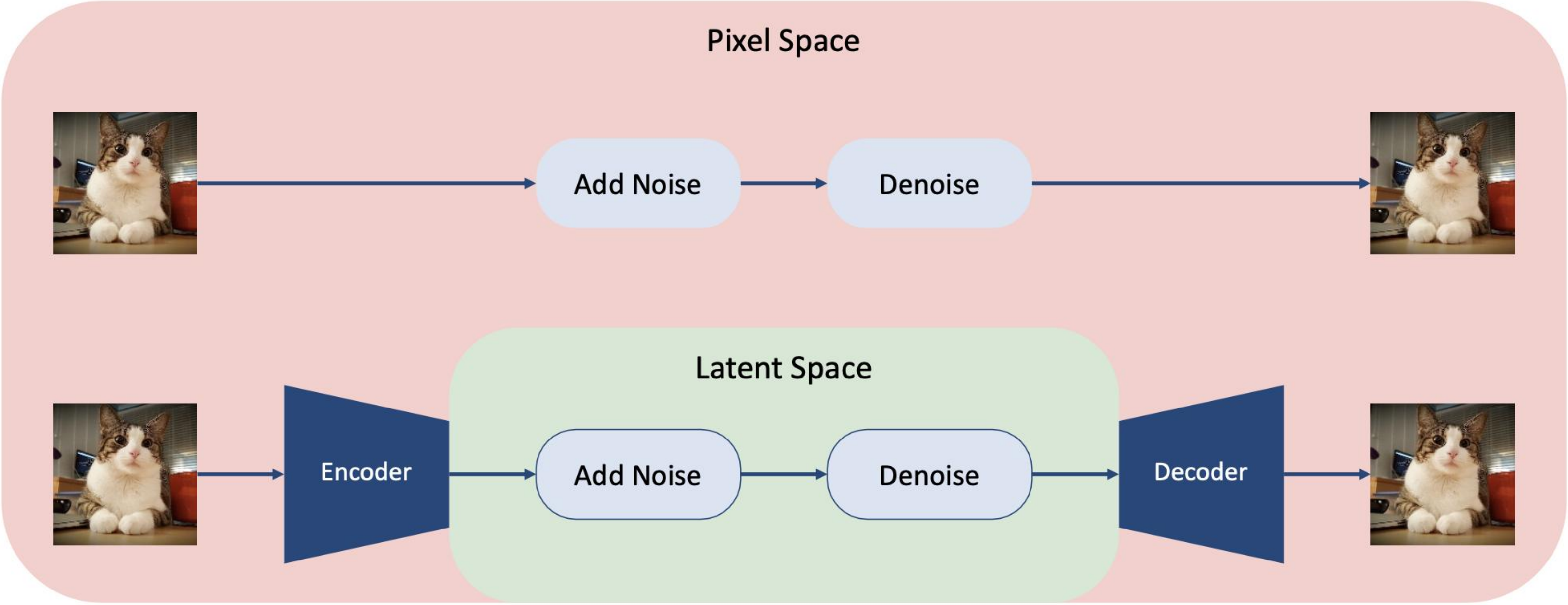
DiT



Latent Diffusion Transformer

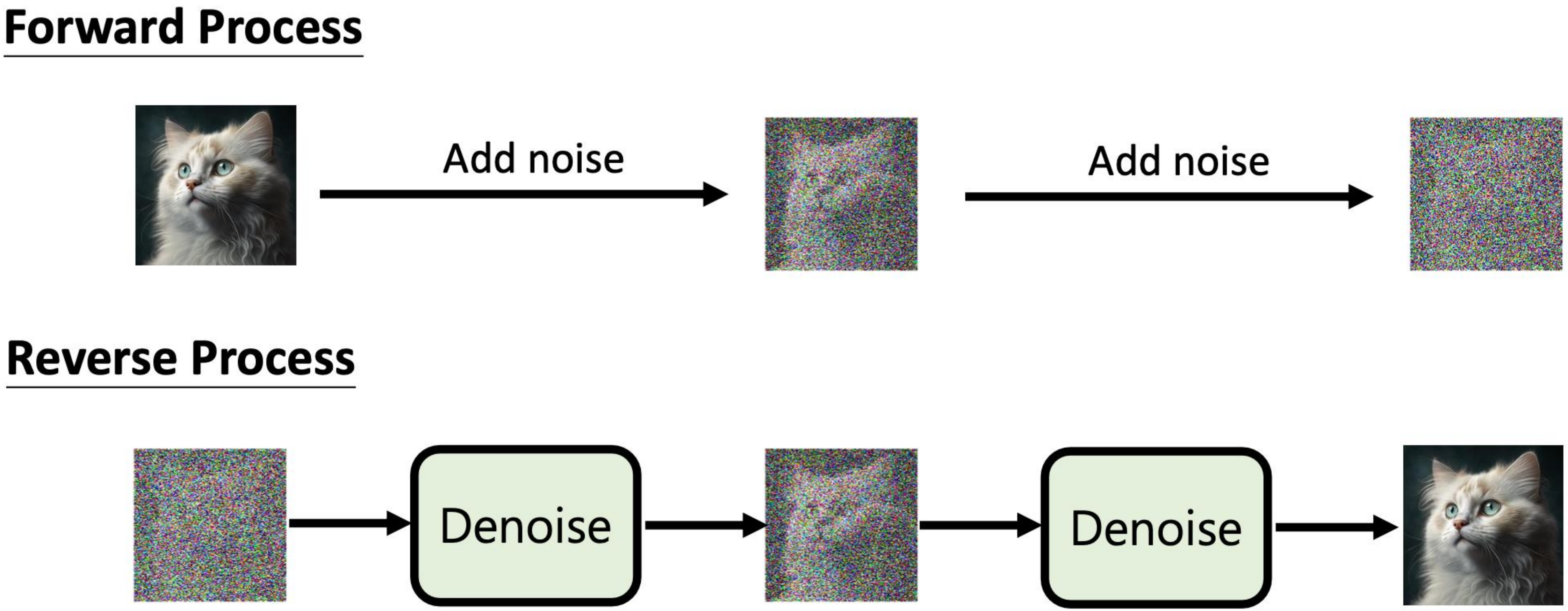
DiT Block with adaLN-Zero

Latent Diffusion



Contents

Diffusion Model



VAE vs. Diffusion Model

Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
until converged

Algorithm 2 Sampling

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

Training

x_0: clean image

ε: noise

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ ← sample clean image
3: $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ← sample a noise
5: Take gradient descent step on

$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\underbrace{\boxed{\sqrt{\bar\alpha_t}\mathbf{x}_0 + \sqrt{1-\bar\alpha_t}\boldsymbol{\epsilon}}}_{\text{Noisy image}}, t) \right\|^2$$

6: **until** converged

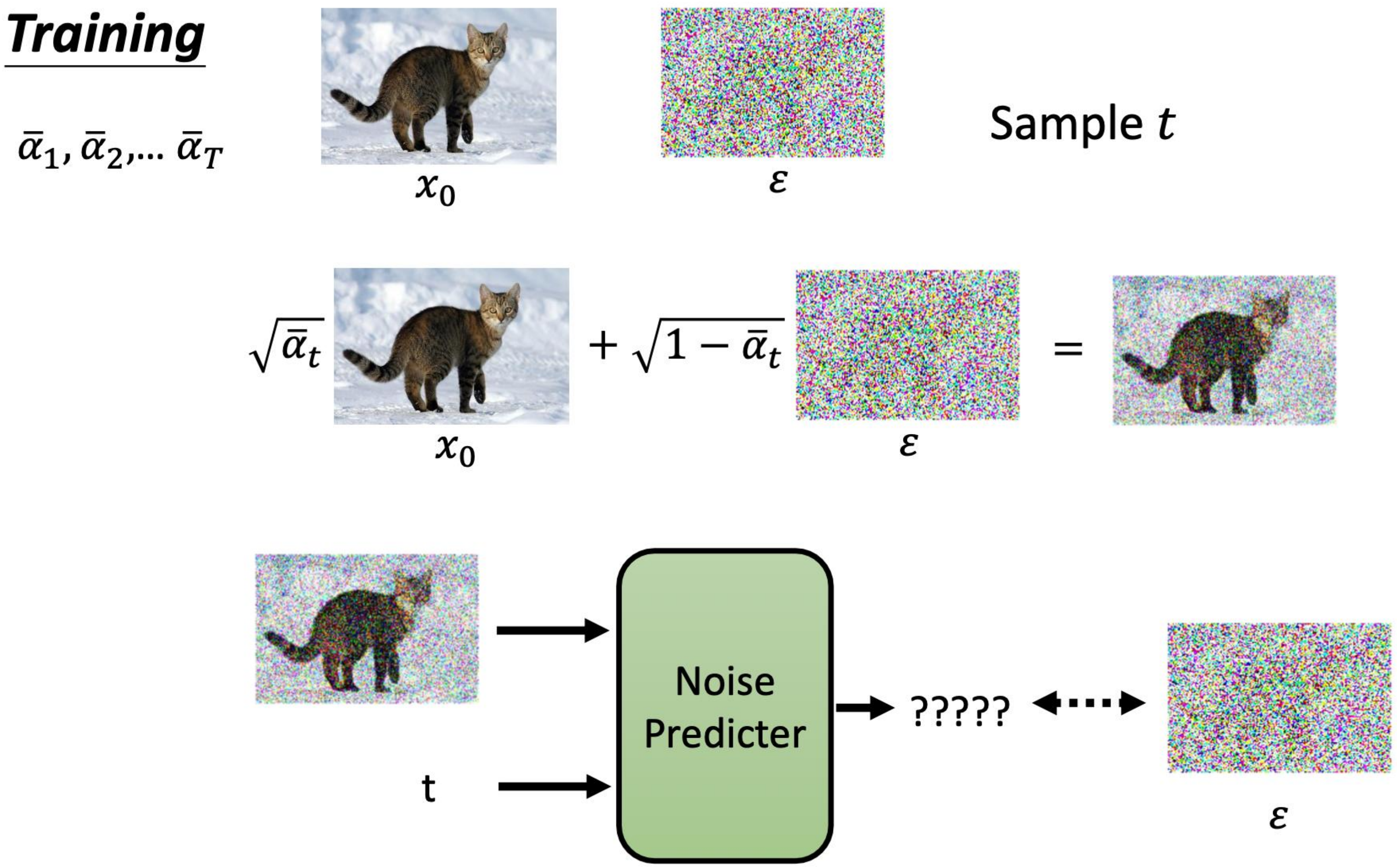
$\bar\alpha_1, \bar\alpha_2, \ldots \bar\alpha_T$ → smaller

Target Noise

Noise predictor

slide from https://speech.ee.ntu.edu.tw/~hylee/ml/2023-spring.php

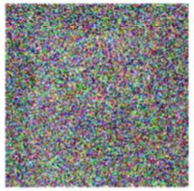
Training



Training
$\bar{\alpha}_1, \bar{\alpha}_2, \ldots \bar{\alpha}_T$
x_0
ε
Sample t
$\sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \varepsilon =$
x_0
ε
t
Noise Predicter
?????
ε

Inference

Inference



x_T

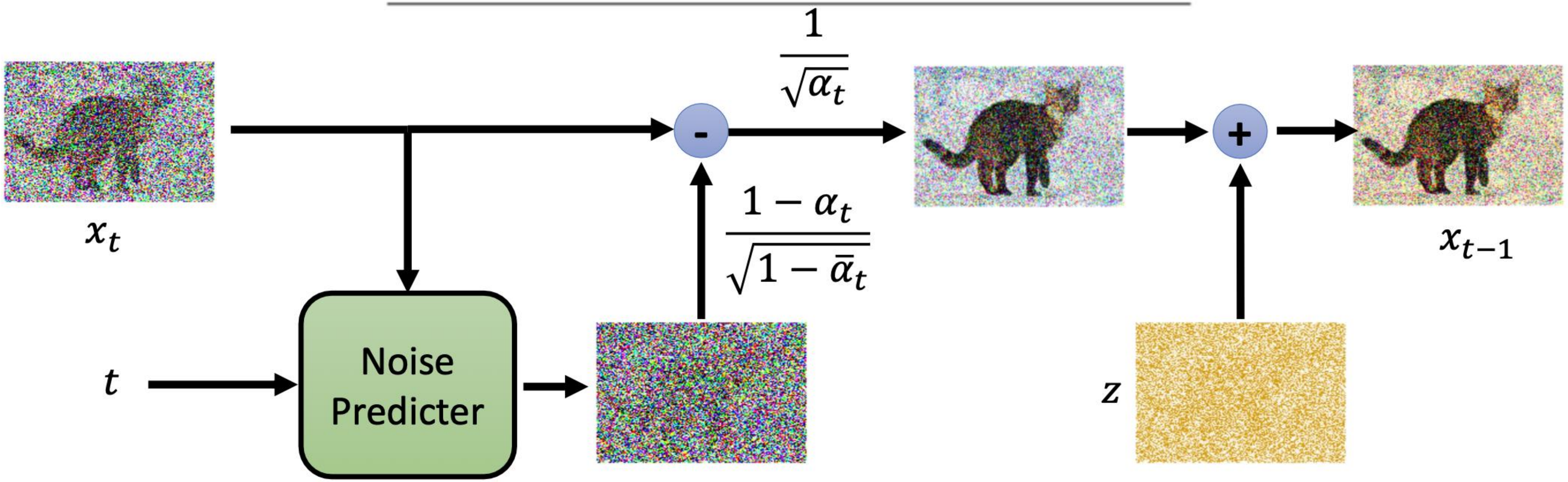
Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\boxed{\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})}$ 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** $\mathbf{x}_0$

sample a noise?!

$\bar{\alpha}_1, \bar{\alpha}_2, \ldots \bar{\alpha}_T$

$\alpha_1, \alpha_2, \ldots \alpha_T$



VAE: Lower bound of log P(x)

$$logP_\theta(x) = \int_z q(z|x)logP(x)dz$$

$q(z|x)$ can be any distribution

$$= \int_z q(z|x)log\left(\frac{P(z,x)}{P(z|x)}\right)dz = \int_z q(z|x)log\left(\frac{P(z,x)}{q(z|x)}\frac{q(z|x)}{P(z|x)}\right)dz$$

$$= \int_z q(z|x)log\left(\frac{P(z,x)}{q(z|x)}\right)dz + \underbrace{\int_z q(z|x)log\left(\frac{q(z|x)}{P(z|x)}\right)dz}_{KL\left(q(z|x)||P(z|x)\right)}$$

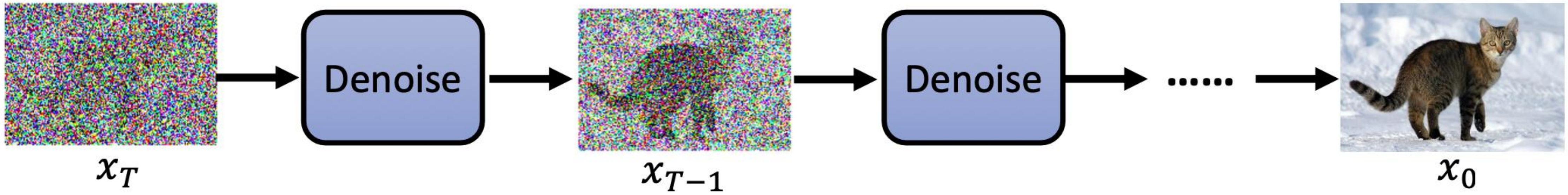
≥ 0

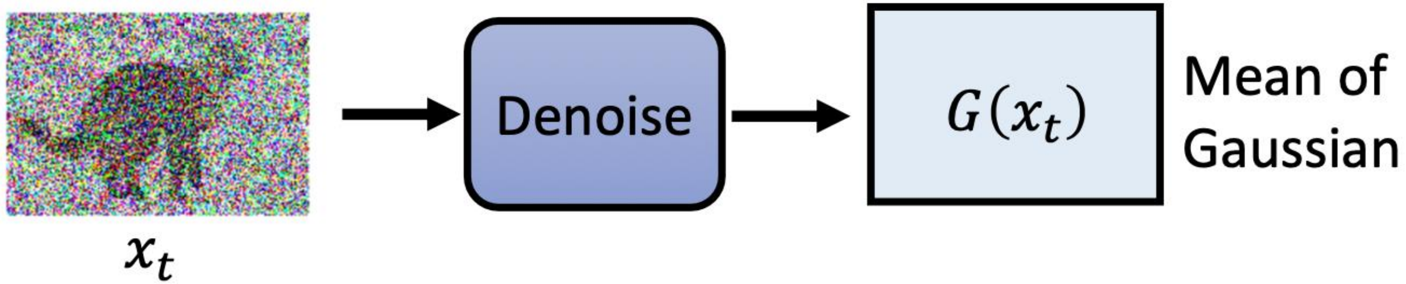
$$\geq \int_z q(z|x)log\left(\frac{P(z,x)}{q(z|x)}\right)dz = \mathrm{E}_{\underline{q(z|x)}}[log\left(\frac{P(x,z)}{q(z|x)}\right)]$$

lower bound

Encoder

DDPM: Compute log P(x)





$$P_\theta(x_{t-1}|x_t)$$
$$\propto \exp(-\|G(x_t) - x_{t-1}\|_2)$$

$$P_\theta(x_0) = \int_{x_1:x_T} P(x_T)P_\theta(x_{T-1}|x_T) \dots P_\theta(x_{t-1}|x_t) \dots P_\theta(x_0|x_1)dx_1{:}x_T$$

VAE: Lower bound of P(x)

VAE Maximize $logP_\theta(x)$ $\longrightarrow$ Maximize $E_{\boxed{q(z|x)}}\left[log\left(\frac{P(x,z)}{q(z|x)}\right)\right]$

Encoder

DDPM Maximize $logP_\theta(x_0)$ $\longrightarrow$ Maximize $E_{\boxed{q(x_1:x_T|x_0)}}\left[log\left(\frac{P(x_0:x_T)}{q(x_1:x_T|x_0)}\right)\right]$

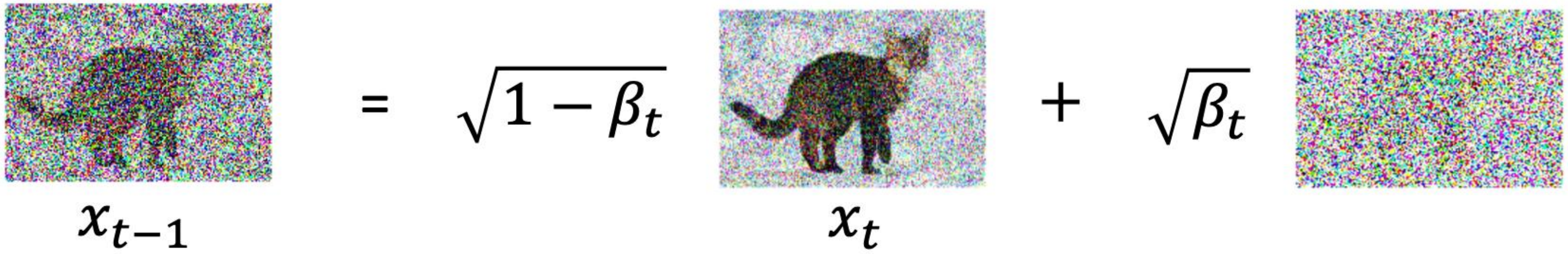
Forward Process
(Diffusion Process)

$$q(x_1:x_T|x_0) = q(x_1|x_0)q(x_2|x_1)\ldots q(x_T|x_{T-1})$$

slide from https://speech.ee.ntu.edu.tw/~hylee/ml/2023-spring.php

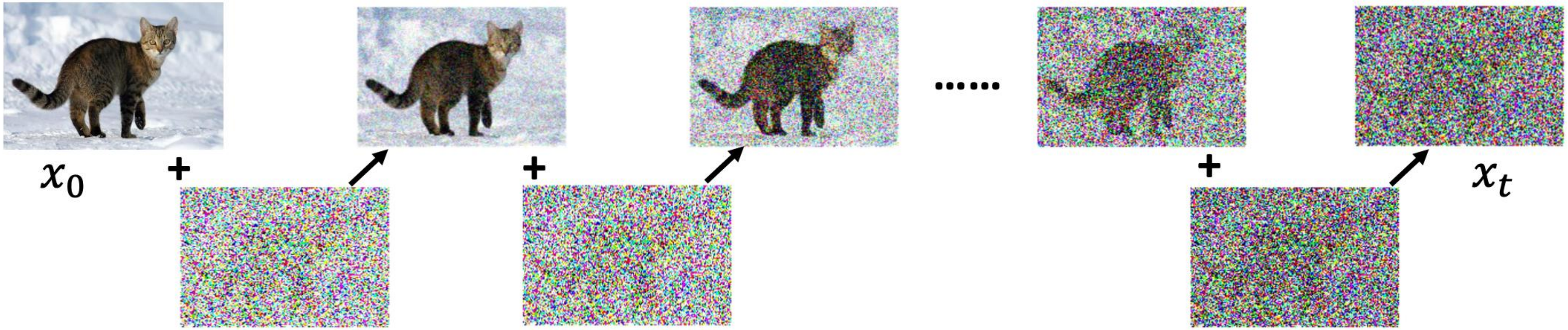
Diffusion Model

$q(x_t|x_{t-1})$

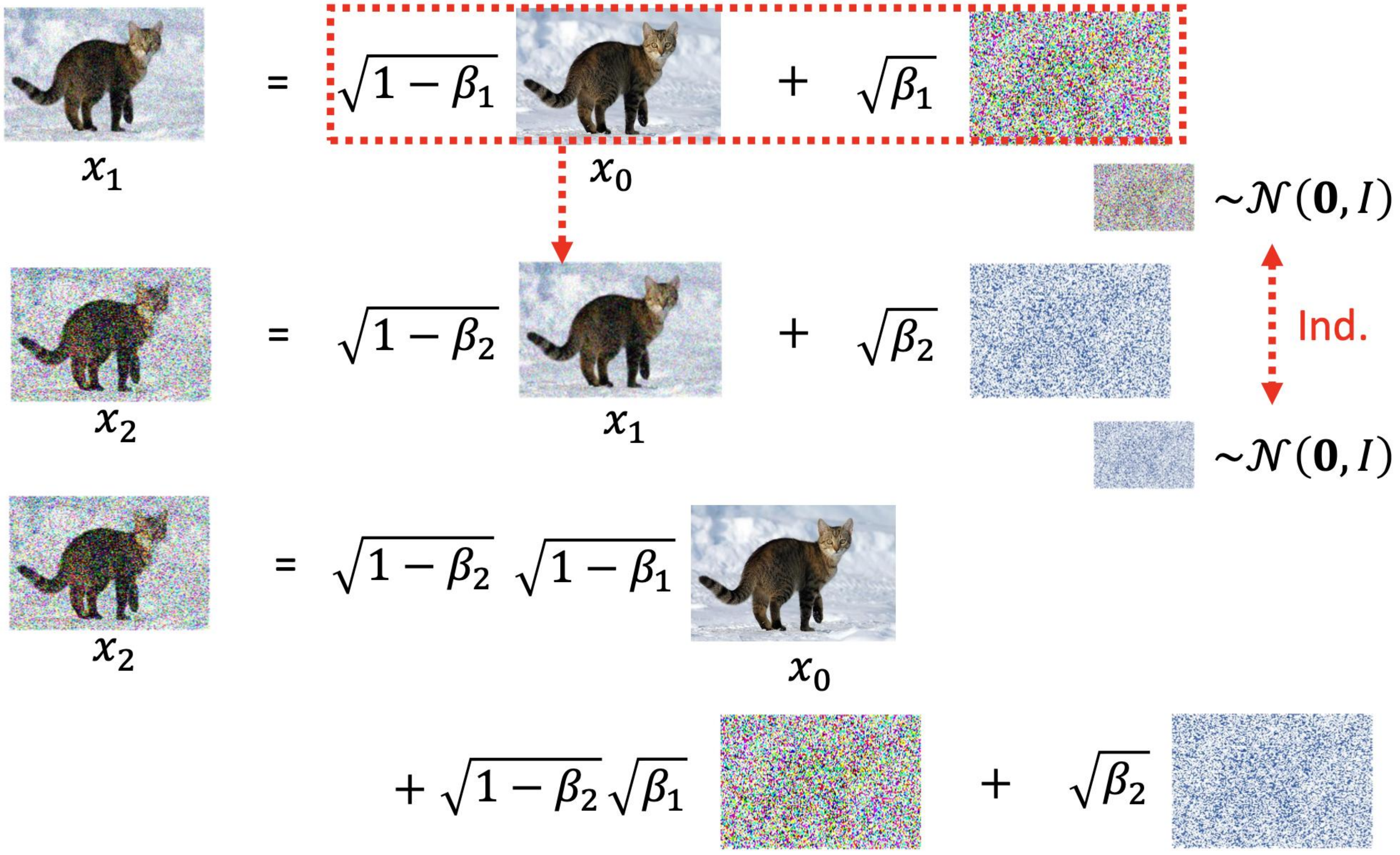


$\beta_1, \beta_2, \ldots, \beta_T$

$q(x_t|x_0)$



Diffusion Model



Diffusion Model

x_2

$= \sqrt{1-\beta_2}\,\sqrt{1-\beta_1}$

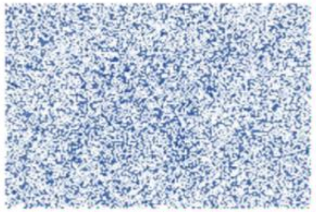
x_0

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

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

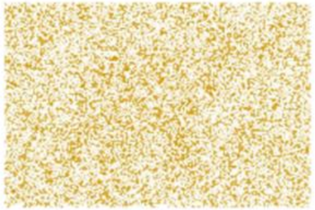
$+ \sqrt{1-\beta_2}\sqrt{\beta_1}$

$+ \sqrt{\beta_2}$



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

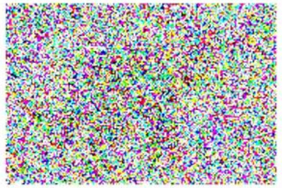
$+ \sqrt{1-(1-\beta_2)(1-\beta_1)}$



Diffusion Model

$q(x_t|x_0)$

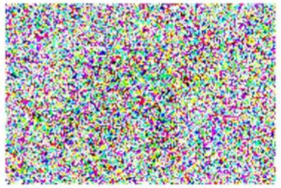
$\beta_1, \beta_2, \dots, \beta_T$

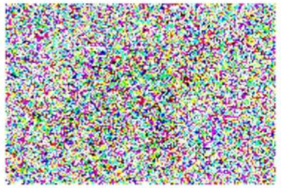


$\sim \mathcal{N}(\mathbf{0}, I)$

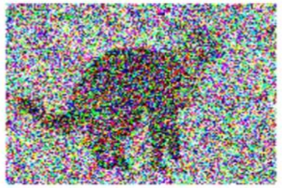
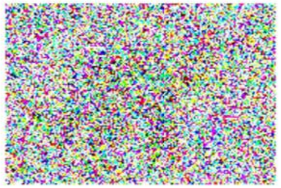
$\alpha_t = 1 - \beta_t$

$\bar{\alpha}_t = \alpha_1 \alpha_2 \dots \alpha_t$

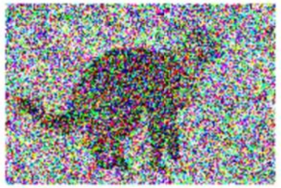
 $= \sqrt{1-\beta_1}$  $+ \sqrt{\beta_1}$ 

 $= \sqrt{1-\beta_2}$  $+ \sqrt{\beta_2}$ 

⋮

 $= \sqrt{1-\beta_t}$  $+ \sqrt{\beta_t}$ 

|||

 $= \underbrace{\boxed{\sqrt{1-\beta_1} \dots \sqrt{1-\beta_t}}}_{\sqrt{\bar{\alpha}_t}}$  $+ \underbrace{\boxed{\sqrt{1-(1-\beta_1)\dots(1-\beta_t)}}}_{\sqrt{1-\bar{\alpha}_t}}$ 

Diffusion Model

$$
\begin{aligned}
\log p(\boldsymbol{x}) &\geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right] && (47)\\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{\prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right] \\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)\prod_{t=2}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right] \\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)\prod_{t=2}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{x}_0)}\right] && (50)\\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{x}_0)}\right] && (51)\\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{\frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)q(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}}\right] && (52)\\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{\frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)\cancel{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}}{\cancel{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}}}\right] && (53)\\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{\cancel{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}} + \log \frac{\cancel{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)}\right] && (54)\\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)} + \sum_{t=2}^{T} \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)}\right] && (55)\\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\right] + \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)}\right] + \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)}\right] && (56)\\
&= \mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\right] + \mathbb{E}_{q(\boldsymbol{x}_T|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)}\right] + \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t,\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)}\right] && (57)\\
&= \boxed{\underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))\right]}_{\text{denoising matching term}} \quad (58)}
\end{aligned}
$$

$$\boxed{\text{Maximize } \mathrm{E}_{q(x_1:x_T|x_0)}\left[log\left(\frac{P(x_0:x_T)}{q(x_1:x_T|x_0)}\right)\right]}$$

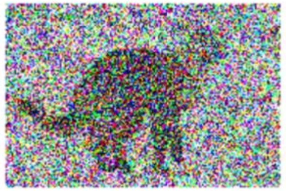
Understanding Diffusion Models:
A Unified Perspective

https://arxiv.org/pdf/2208.11970.pdf

Diffusion Model

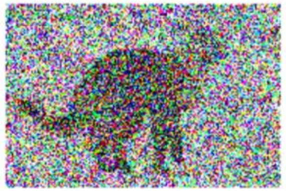
$$\mathrm{E}_{q(x_1|x_0)}[logP(x_0|x_1)] - KL\big(q(x_T|x_0)||P(x_T)\big) - \sum_{t=2}^{T} \boxed{\mathrm{E}_{q(x_t|x_0)}\big[KL\big(q(x_{t-1}|x_t, x_0)||P(x_{t-1}|x_t)\big)\big]}$$

Sample x_0



Sample x_t

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

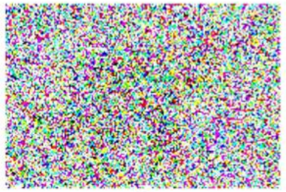


x_t

$= \sqrt{\bar{\alpha}_t}$

x_0

$+ \sqrt{1-\bar{\alpha}_t}$



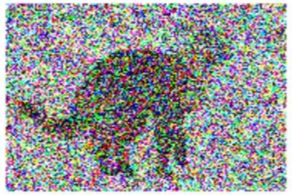
ε

Diffusion Model

$$\mathrm{E}_{q(x_1|x_0)}[logP(x_0|x_1)] - KL\big(q(x_T|x_0)||P(x_T)\big)$$

$$-\sum_{t=2}^{T} \boxed{\mathrm{E}_{q(x_t|x_0)}\big[KL\big(q(x_{t-1}|x_t, x_0)||P(x_{t-1}|x_t)\big)\big]}$$

x_0

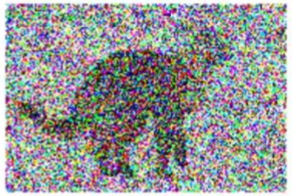


x_t

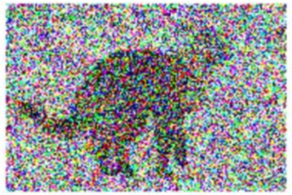
t

Denoise

?



$$\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t x_0 + \sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t}{1-\bar{\alpha}_t}$$



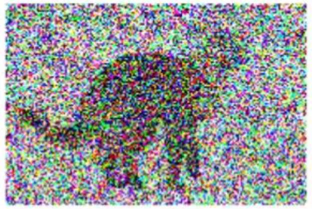
Sample x_t

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

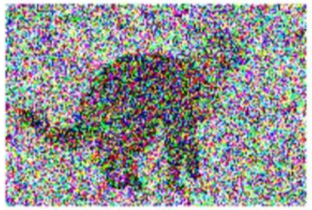
slide from https://speech.ee.ntu.edu.tw/~hylee/ml/2023-spring.php

Diffusion Model

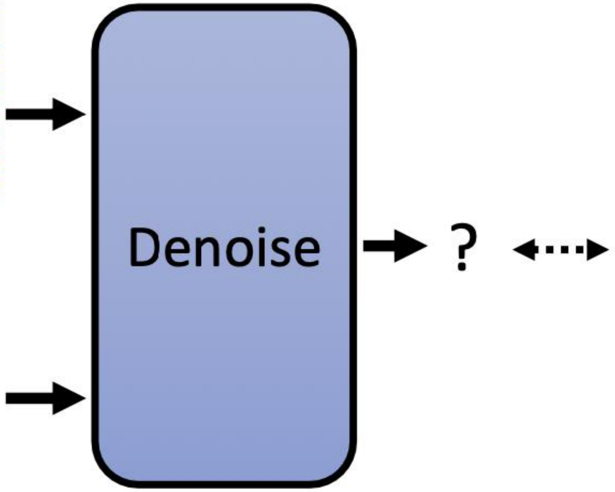
x_0



x_t



Sample x_t



t

$$? \longleftrightarrow \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t \boxed{x_0} + \sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t}{1-\bar{\alpha}_t}$$

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

$$x_t - \sqrt{1-\bar{\alpha}_t}\varepsilon = \sqrt{\bar{\alpha}_t}x_0$$

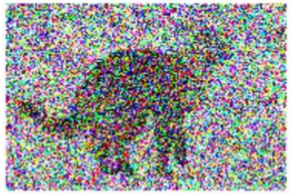
$$\frac{x_t - \sqrt{1-\bar{\alpha}_t}\varepsilon}{\sqrt{\bar{\alpha}_t}} = x_0$$

$$= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t \boxed{\frac{x_t - \sqrt{1-\bar{\alpha}_t}\varepsilon}{\sqrt{\bar{\alpha}_t}}} + \sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t}{1-\bar{\alpha}_t}$$

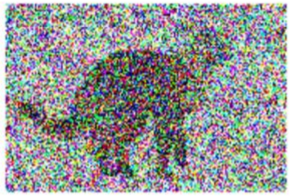
$$= \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon\right)$$

Diffusion Model

x_0



x_t



Sample x_t

$x_t \rightarrow$ Denoise $\leftarrow t$

Denoise $\rightarrow$? $\leftrightarrow$ $\frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boxed{\varepsilon}\right)$

實際需要 network predict 的部分

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

$$x_t - \sqrt{1-\bar{\alpha}_t}\varepsilon = \sqrt{\bar{\alpha}_t}x_0$$

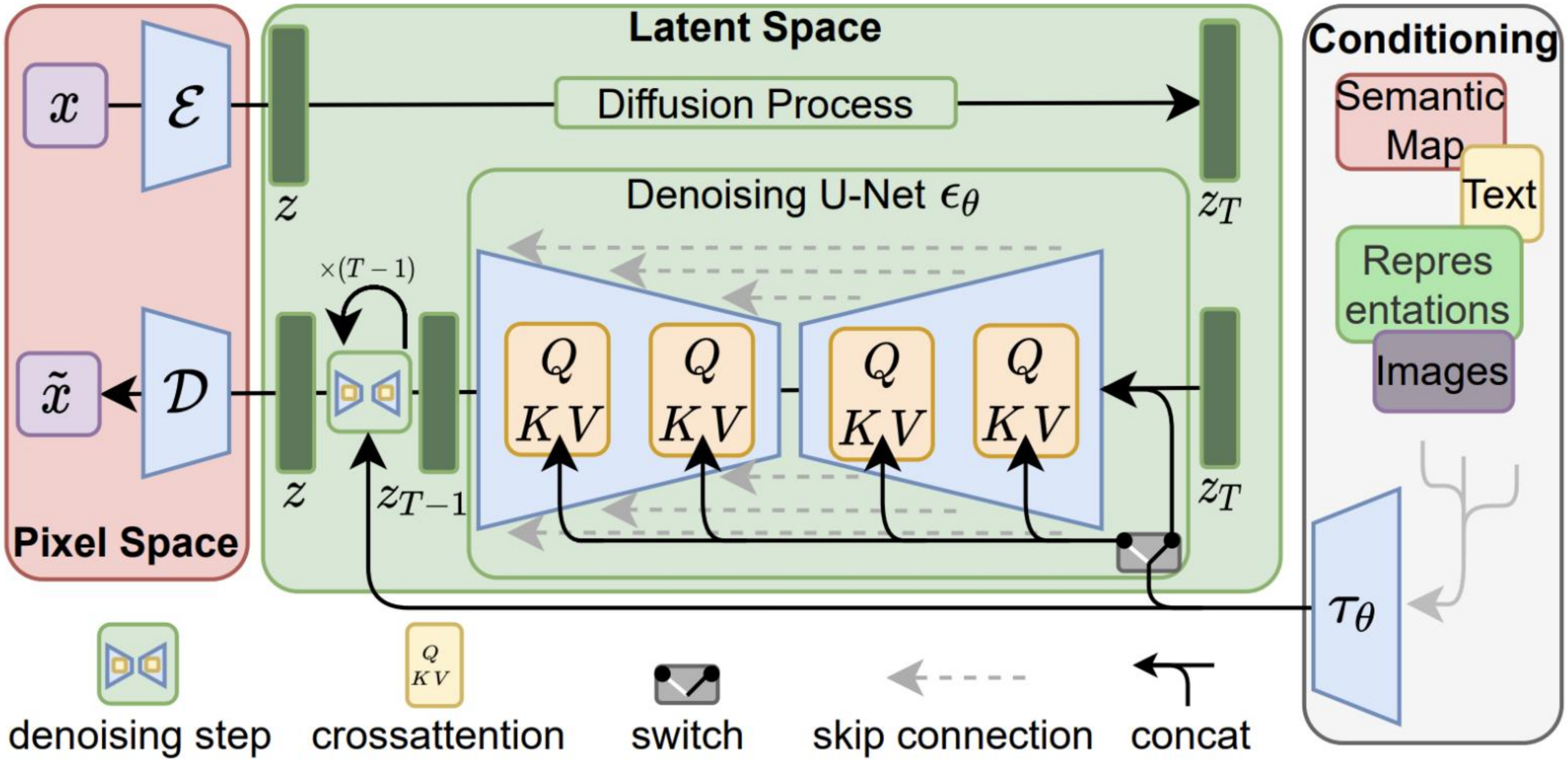
$$\frac{x_t - \sqrt{1-\bar{\alpha}_t}\varepsilon}{\sqrt{\bar{\alpha}_t}} = x_0$$

Algorithm 2 Sampling

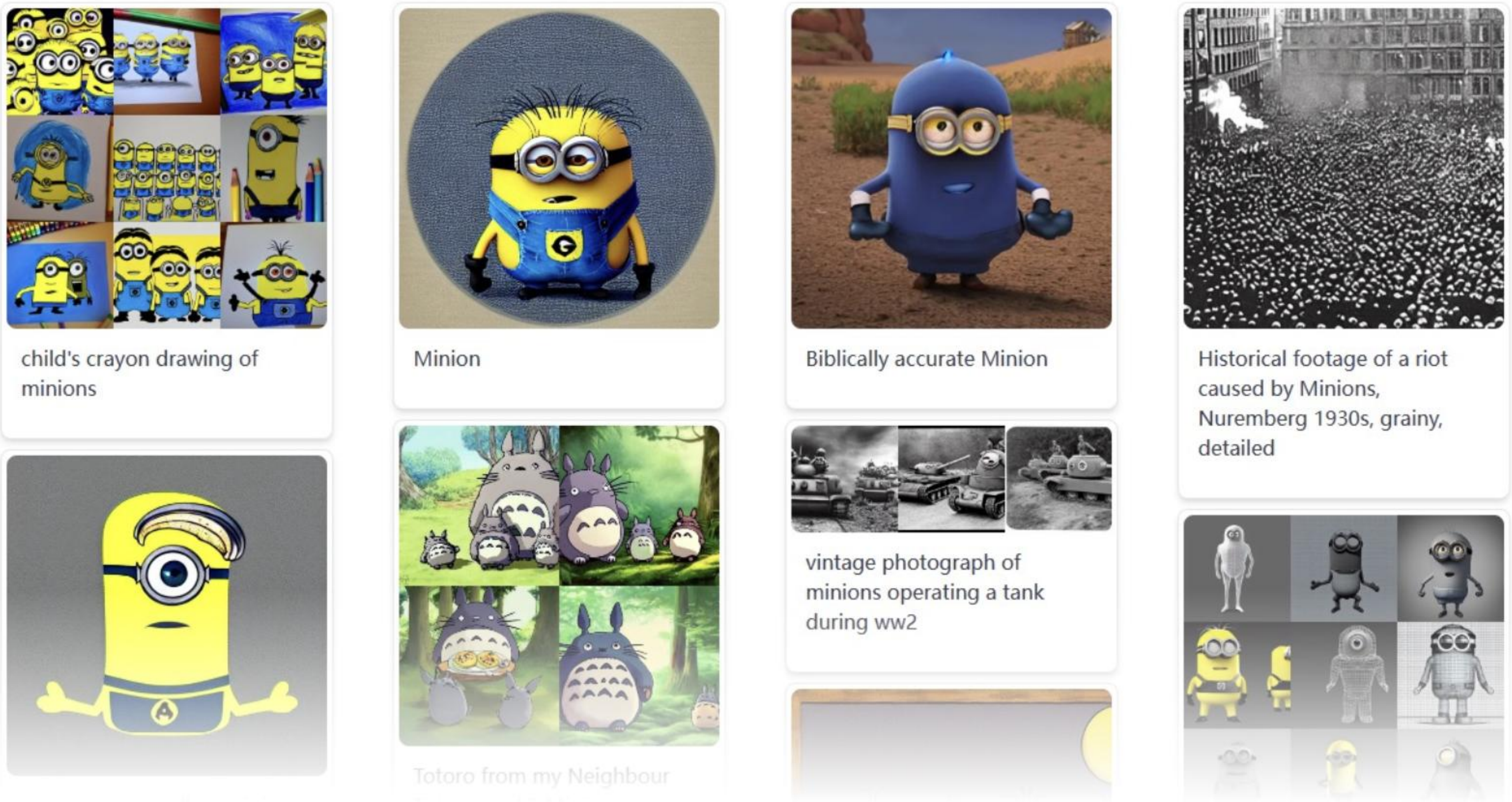
$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $\mathbf{x}_{t-1} = \boxed{\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right)} + \sigma_t \mathbf{z}$
end for
return $\mathbf{x}_0$

- Last Class Review
- Overview of Diffusion Models
- Theoretical Derivation of Diffusion Models
- Image Generation Applications
- Basic Concepts in Neural Network

Stable Diffusion: Conditional/unconditional image generation

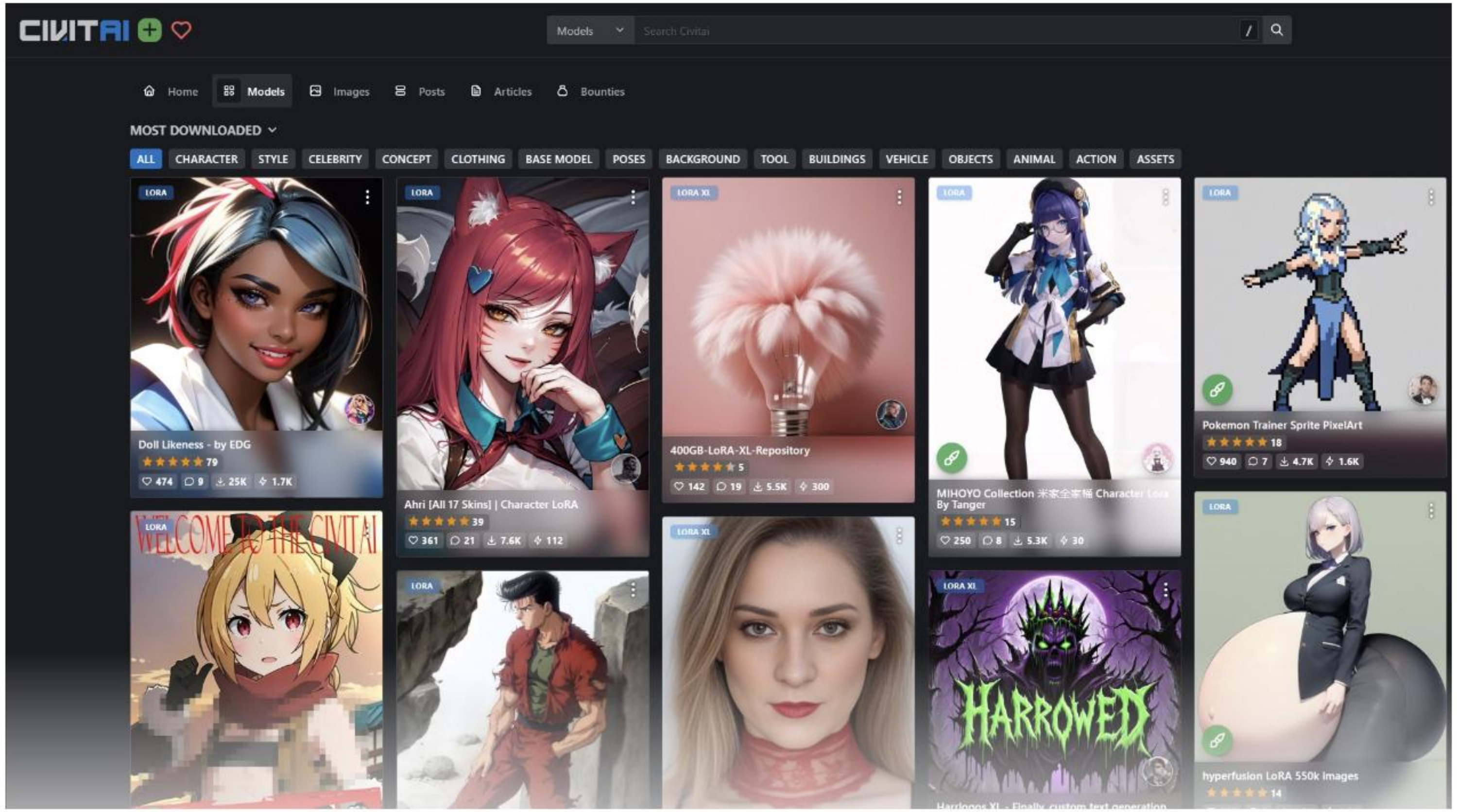


Stable Diffusion: Conditional/unconditional image generation



LoRA: Low-Rank Adaptation

Few-shot finetuning of large models for personalized generation.



A Parameter-Efficient Fine-Tuning Method

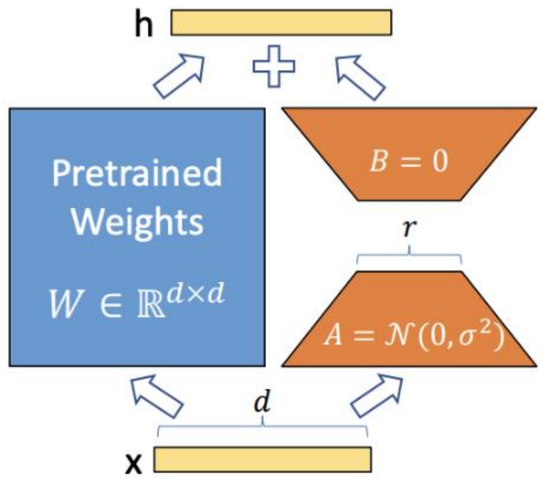
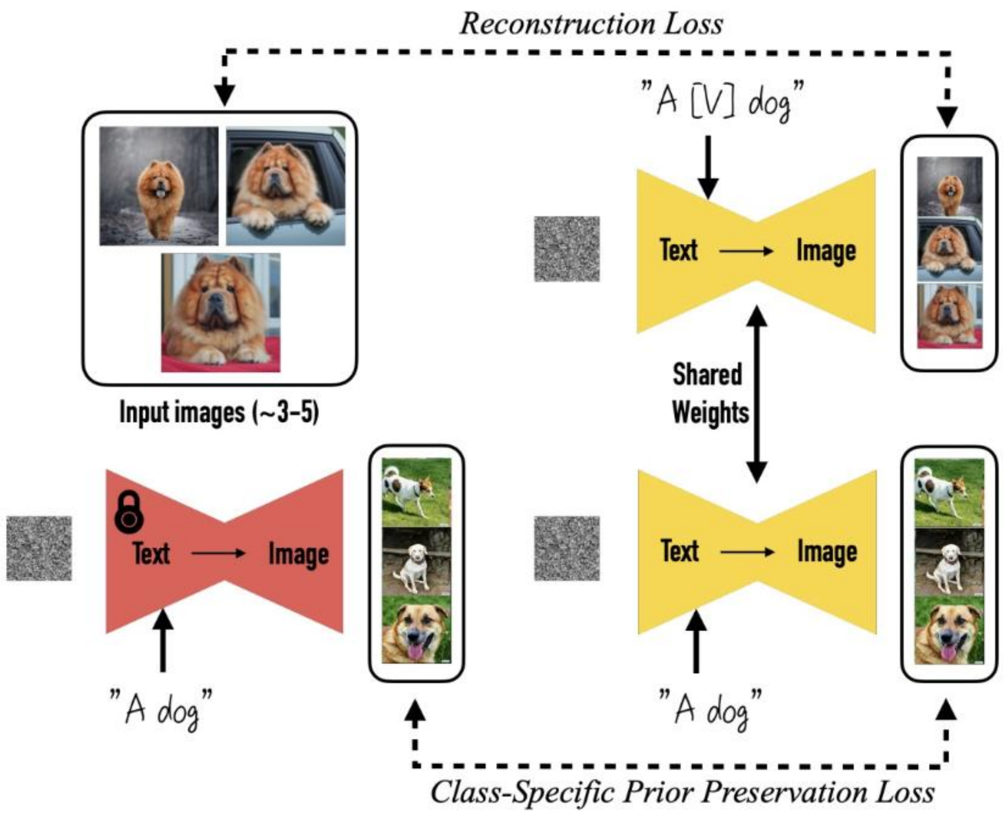


Figure 1: Our reparametrization. We only train A and B.

DreamBooth

Input images
A [V] backpack in the Grand Canyon
A wet [V] backpack in water
A [V] backpack in Boston
A [V] backpack with the night sky
Input images
A [V] teapot floating in milk
A transparent [V] teapot with milk inside
A [V] teapot pouring tea
A [V] teapot floating in the sea



Reconstruction Loss
"A [V] dog"
Input images (~3-5)
Text → Image
Shared Weights
Text → Image
Text → Image
"A dog"
"A dog"
Class-Specific Prior Preservation Loss

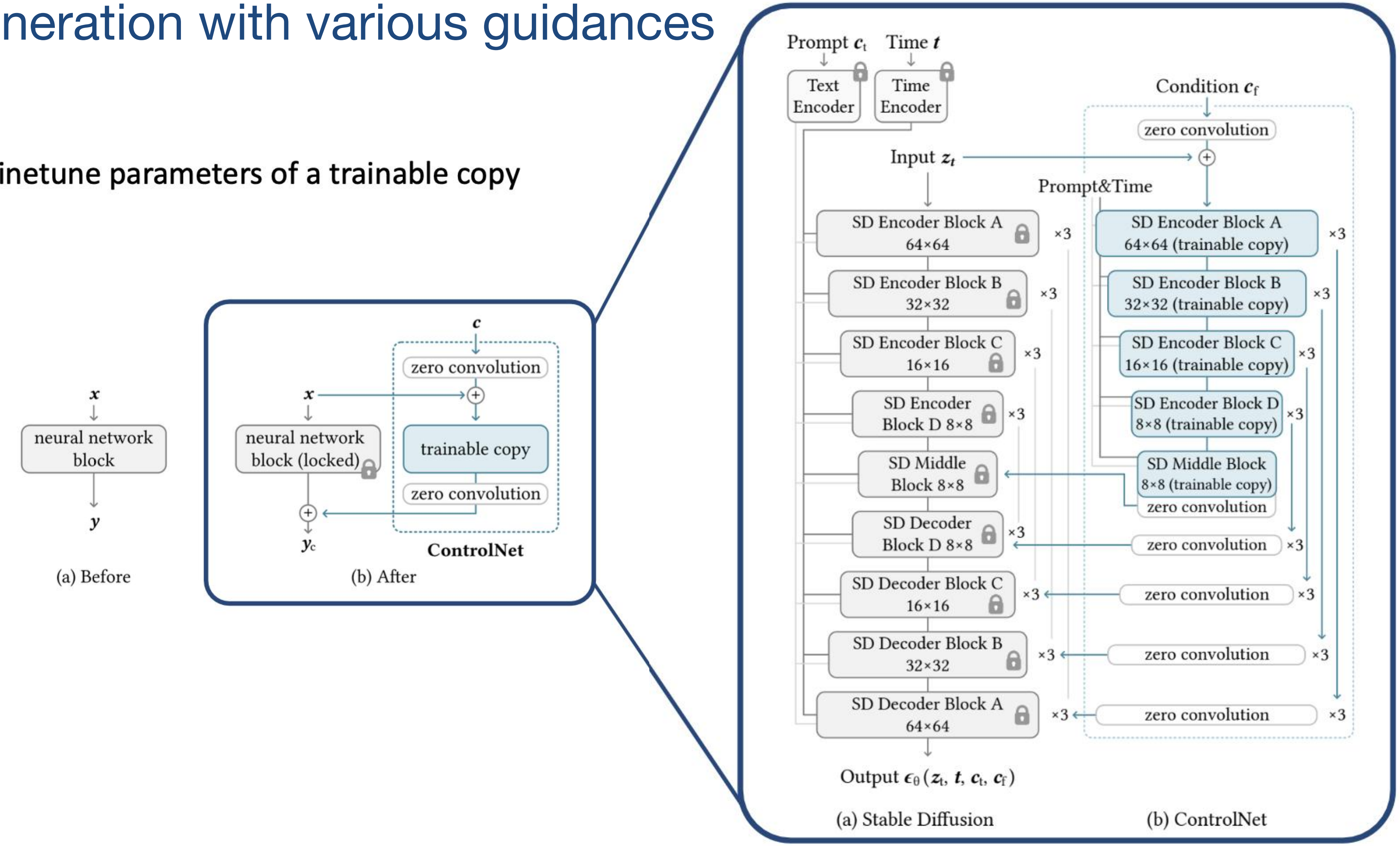
ControlNet

Conditional generation with various guidances

ControlNet

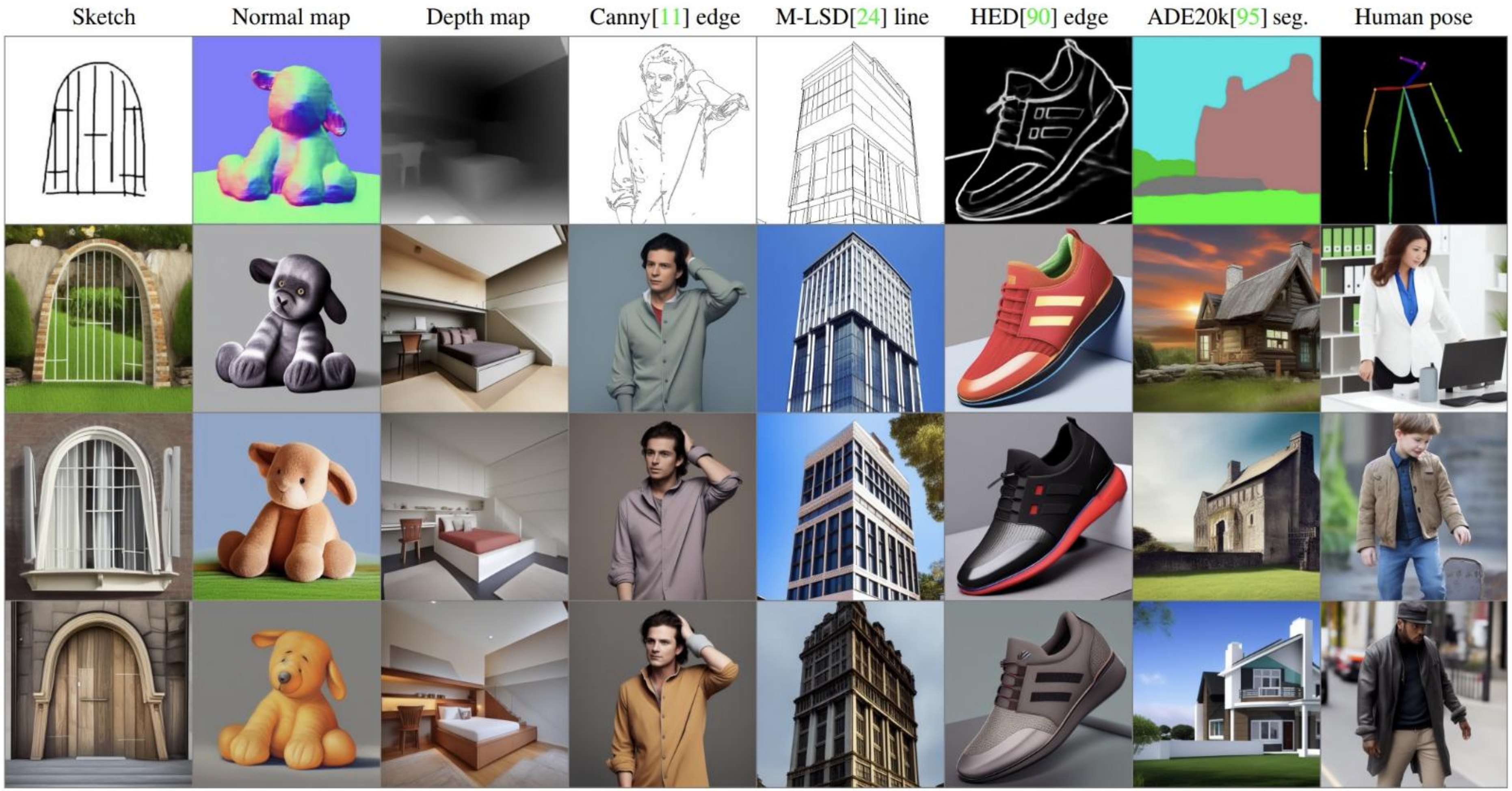
Conditional generation with various guidances

- Finetune parameters of a trainable copy



ControlNet

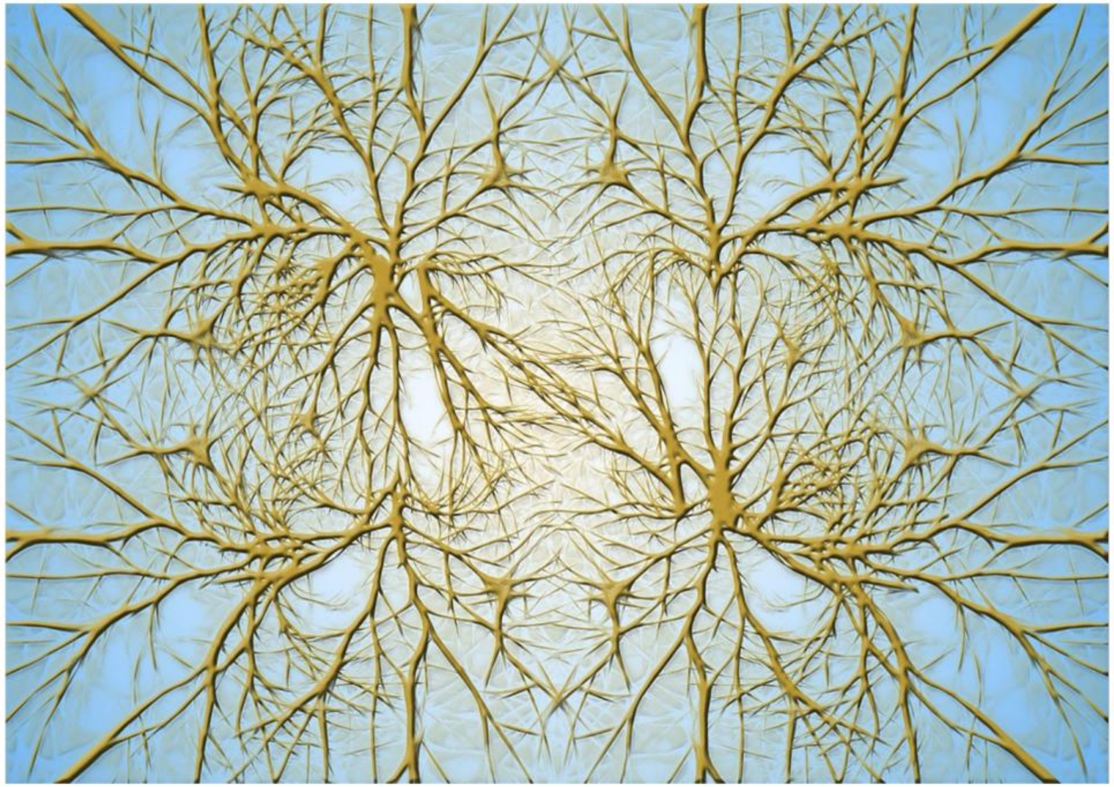
Conditional generation with various guidances



- Last Class Review
- Overview of Diffusion Models
- Theoretical Derivation of Diffusion Models
- Image Generation Applications
- Basic Concepts in Neural Network

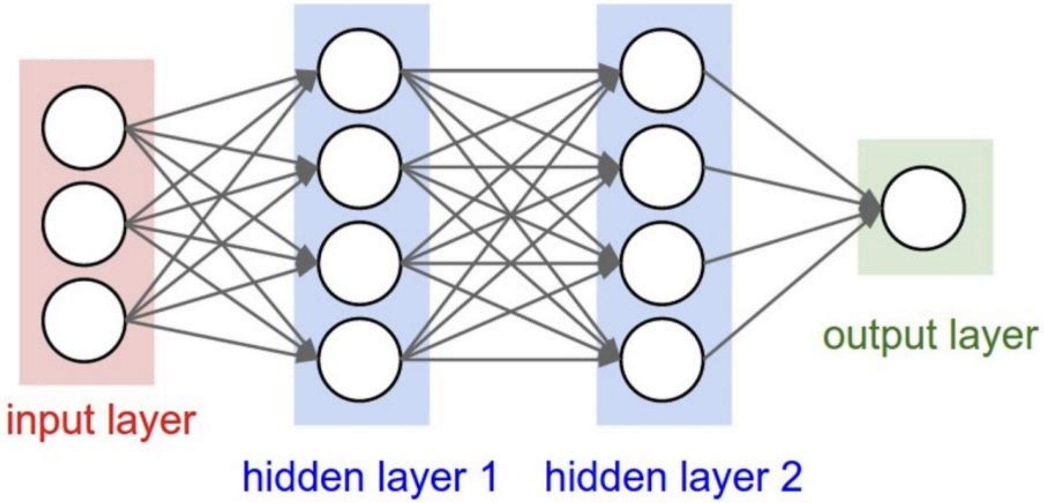
Neural Network: multilayer perceptron

Biological Neurons:
Complex connectivity patterns

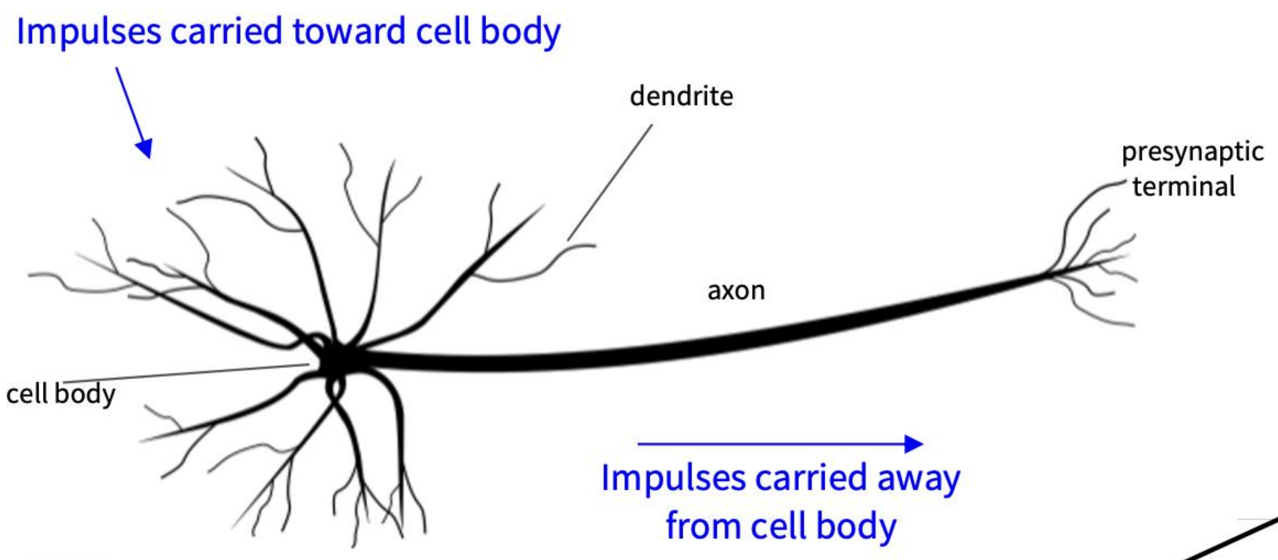


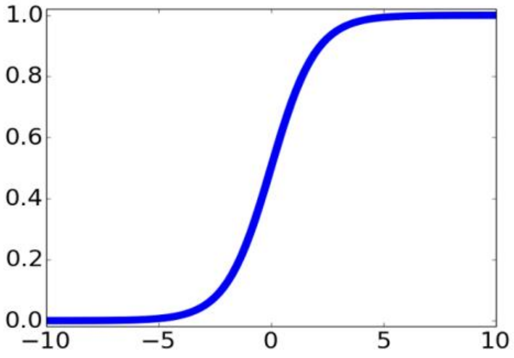
This image is CC0 Public Domain

Neurons in a neural network:
Organized into regular layers for computational efficiency



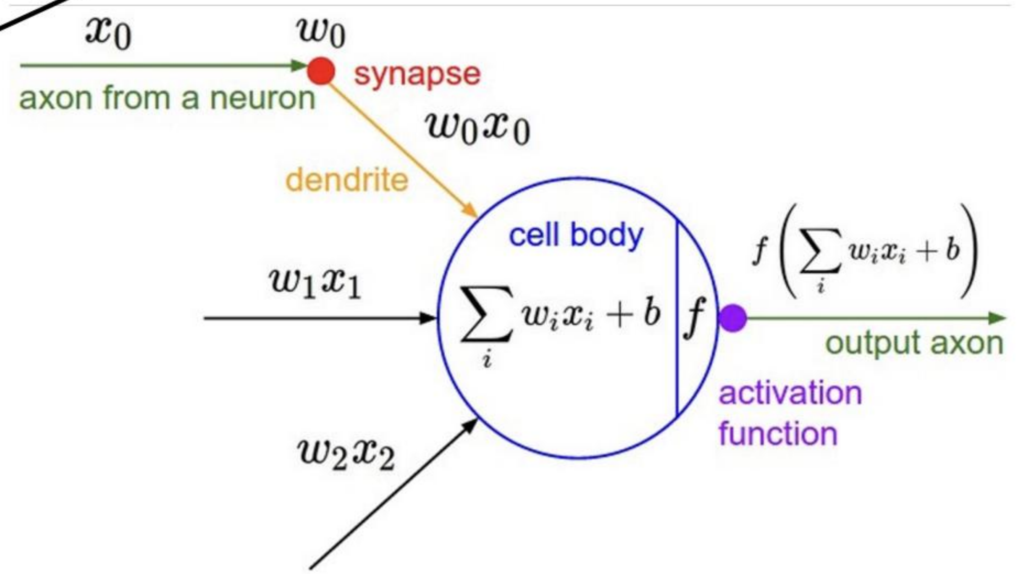
Neural Network: Neuron





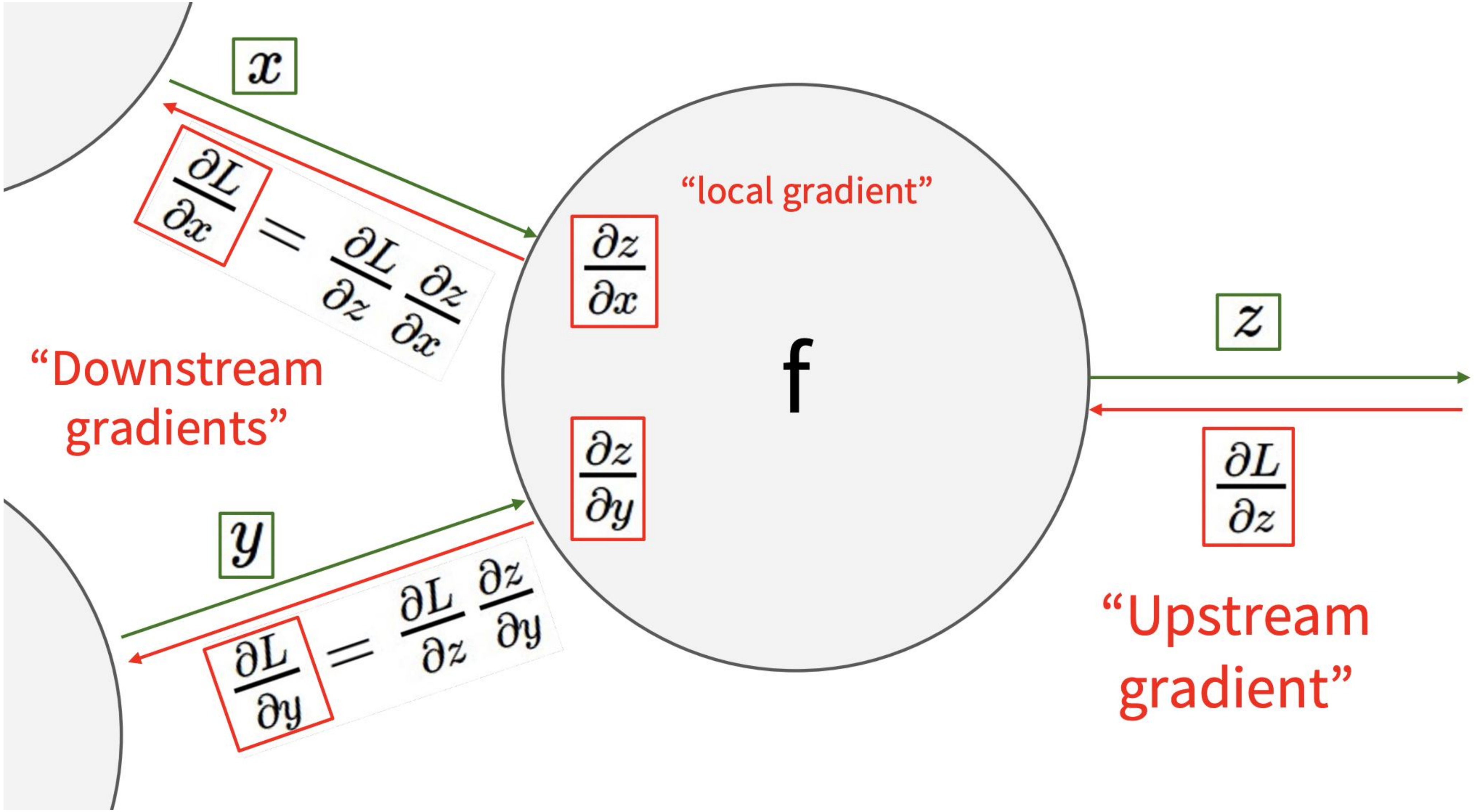
sigmoid activation function

$$\frac{1}{1+e^{-x}}$$



slide from https://cs231n.stanford.edu/slides/2024/lecture_4.pdf

Neural Network: Backpropagation



slide from https://cs231n.stanford.edu/slides/2024/lecture_4.pdf

Thanks!