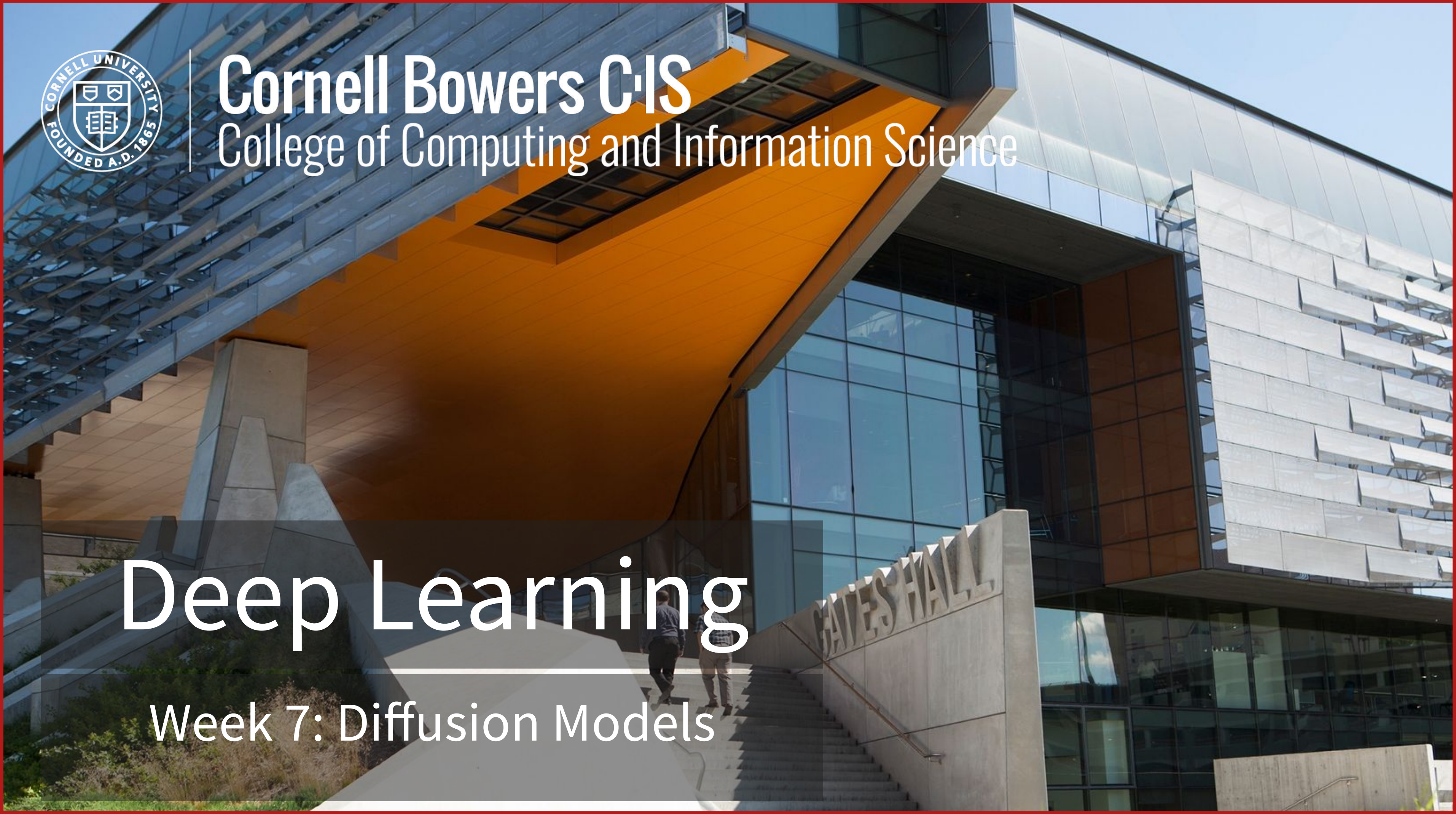
Cornell Bowers C·IS

College of Computing and Information Science

Deep Learning

Week 7: Diffusion Models

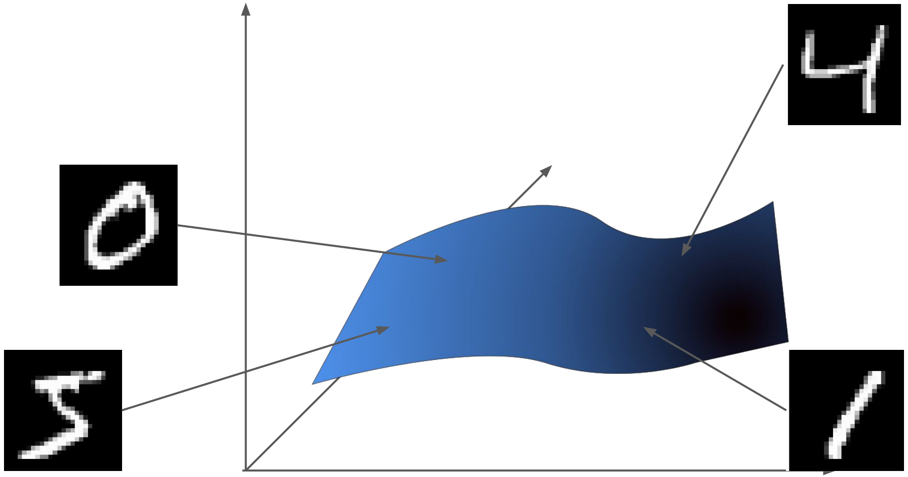


Overview

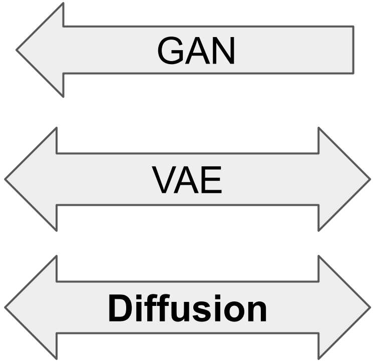
- Recap
- Diffusion model overview
- Forward
- Reverse
- Training Objective

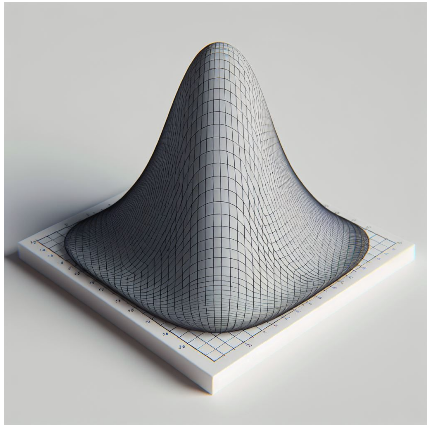
Recall: Data Manifold

- Data distribution **P(X)** defines a manifold of valid images
- Problem: data manifold takes up **tiny** volume of ambient space
- Naive random samples (e.g. within $[0,1]^d$) are always **off manifold**
- Solution: Sample from a Gaussian, then learn mapping to and from manifold



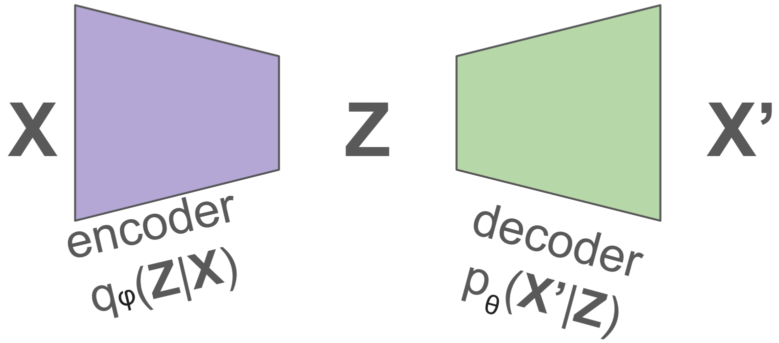
We want to sample from the elusive Image Manifold





We **can** sample from a Gaussian Distribution

Recall: VAE

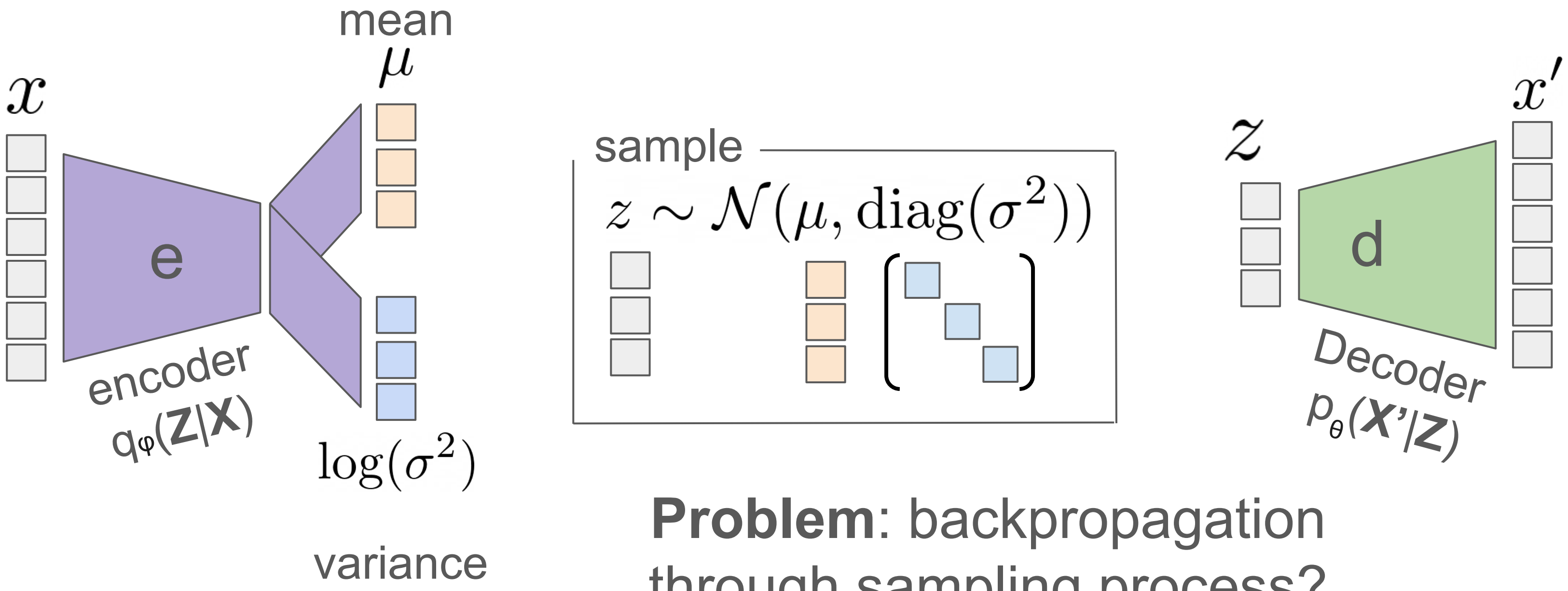


Back to our AutoEncoder, but this time we make everything **probabilistic**!

$$\max_{\phi,\theta} \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\left(p_\theta(\mathbf{x}|\mathbf{z})\right)\right]$$

How likely would it be to encode x, decode the result, and recover x?

Probabilistic **Encoder** (Gaussian)

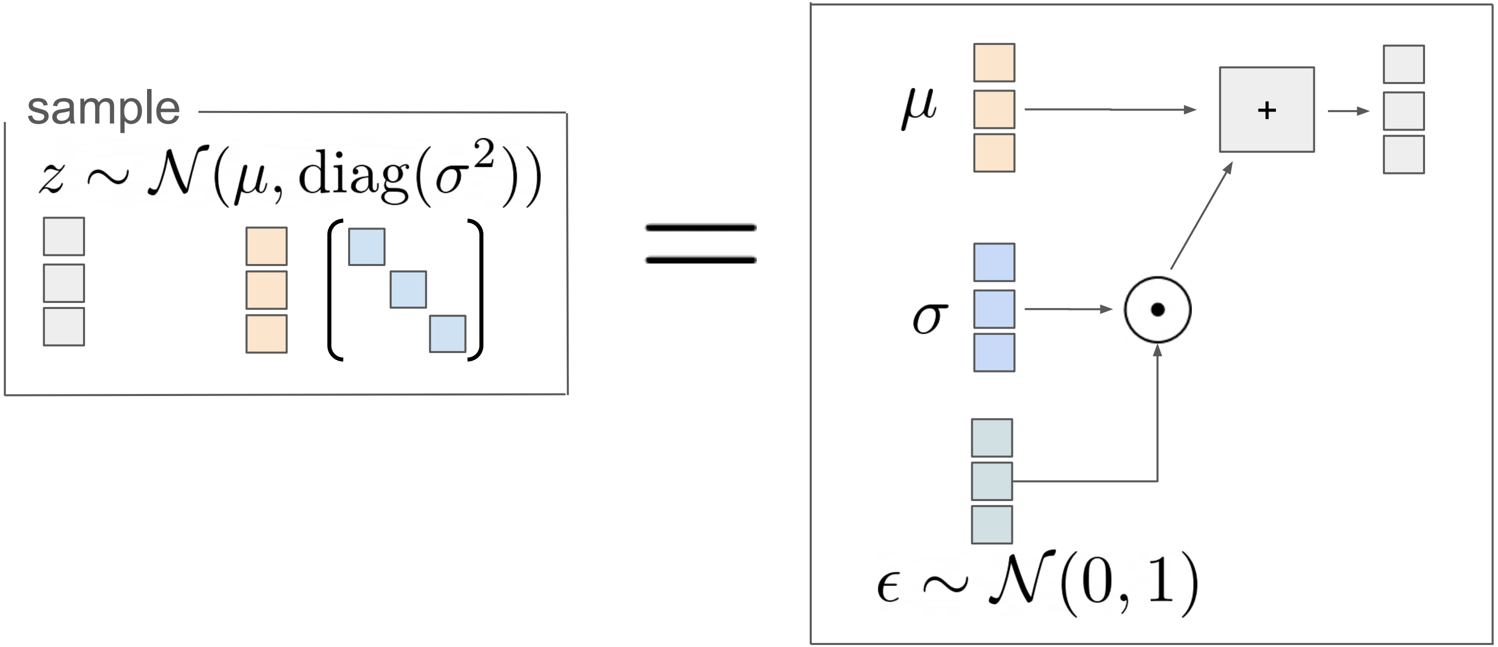


Problem: backpropagation through sampling process?

$$\max_{\phi,\theta} \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\left(p_\theta(\mathbf{x}|\mathbf{z})\right)\right]$$

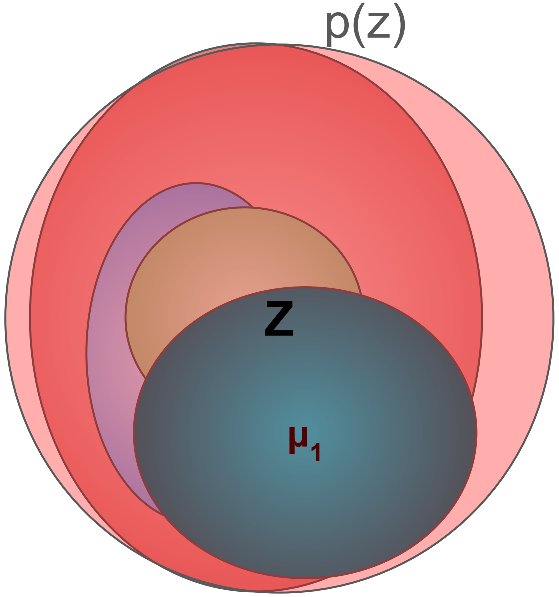
Recall: The Reparameterization Trick

$$\mathcal{N}(\mu, \mathrm{diag}(\sigma^2)) = \mu + \sigma \odot \mathcal{N}(0, I)$$


sample
$z \sim \mathcal{N}(\mu, \mathrm{diag}(\sigma^2))$
=
μ
+
σ
$\odot$
$\epsilon \sim \mathcal{N}(0, 1)$

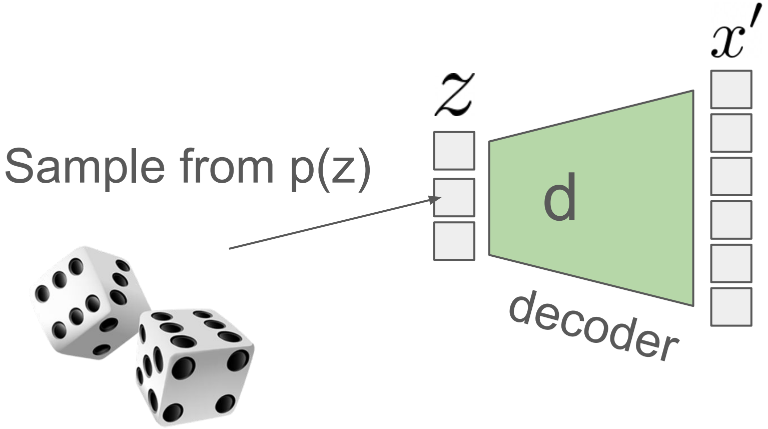
Recall: How do we sample in latent space?

Solution: Regularize all distributions to be close to the standard normal ***N(0;I)*.**



maximize

$$\underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z}))}_{\text{prior matching term}}$$



Sample from p(z)

KL Divergence (a.k.a. relative entropy)

$$\boldsymbol{D}(\,p\,\|\,q\,) := \mathop{\mathbb{E}}_{x\sim p}\left[\log\frac{p(x)}{q(x)}\right]$$

Distribution 1

Distribution 2

$$= \underbrace{\mathop{\mathbb{E}}_{x\sim p}\Big[\log\frac{1}{q(x)}}_{\text{Cross Entropy!}} - \underbrace{\log\frac{1}{p(x)}\Big]}_{\text{(constant wrt q)}}$$

- non-negative $\boldsymbol{D}(\,p\,\|\,q\,) \geq 0$
- zero means same $\boldsymbol{D}(\,p\,\|\,q\,) = 0 \iff p = q$
- not symmetric
- has many other, uniquely nice properties …

Cornell Bowers CIS

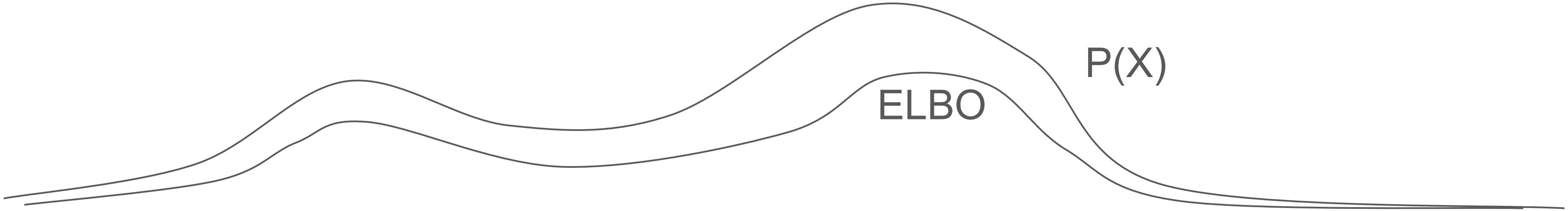
Recall: Evidence Lower Bound (ELBO)

Data likelihood ≥ Reconstruction – KL Divergence

$$\log p(\boldsymbol{x}) \geq \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z}))}_{\text{prior matching term}}$$

(We are **maximizing** this lower bound.)

If we maximize ELBO, we get closer to max to P(**x**).

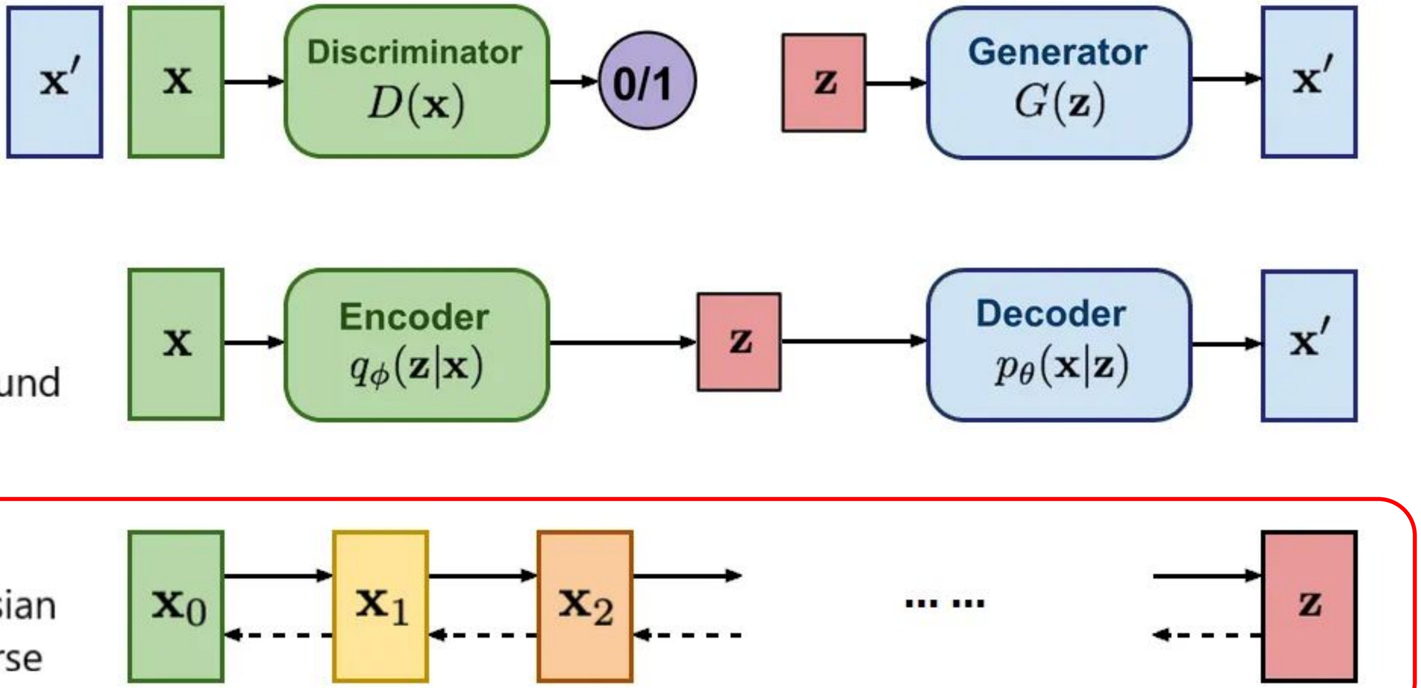


[Calvin Luo https://arxiv.org/abs/2208.11970]

GAN: Adversarial training

VAE: maximize variational lower bound

Diffusion models: Gradually add Gaussian noise and then reverse

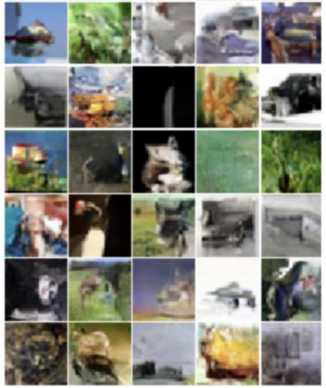


https://miro.medium.com/v2/resize:fit:1400/format:webp/1*_5GpdejeOvt61ew4aPtT_g.png

Progress In Generative Modeling

VAEs, 2013

GANs, 2014



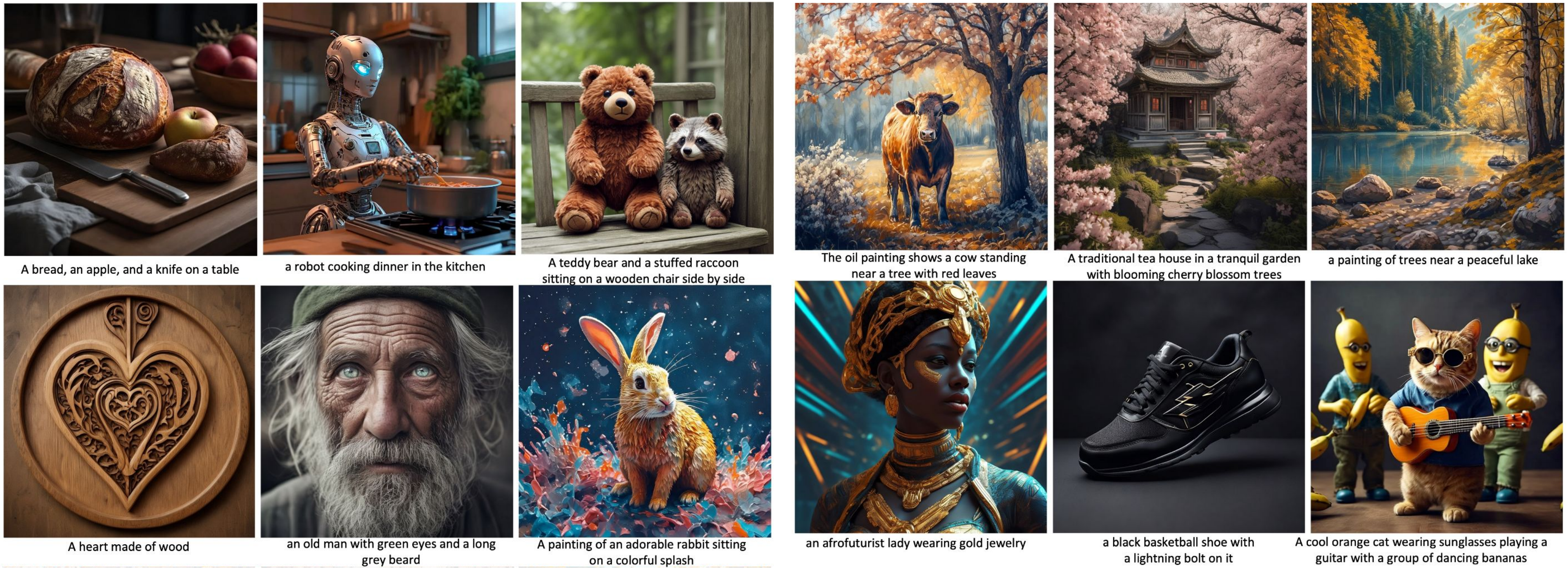
PixelCNN, 2016

BigGAN, 2019

Imagen, 2022

Imagen
Imagen

Text-to-Image Diffusion Models



A bread, an apple, and a knife on a table

a robot cooking dinner in the kitchen

A teddy bear and a stuffed raccoon sitting on a wooden chair side by side

The oil painting shows a cow standing near a tree with red leaves

A traditional tea house in a tranquil garden with blooming cherry blossom trees

a painting of trees near a peaceful lake

A heart made of wood

an old man with green eyes and a long grey beard

A painting of an adorable rabbit sitting on a colorful splash

an afrofuturist lady wearing gold jewelry

a black basketball shoe with a lightning bolt on it

A cool orange cat wearing sunglasses playing a guitar with a group of dancing bananas

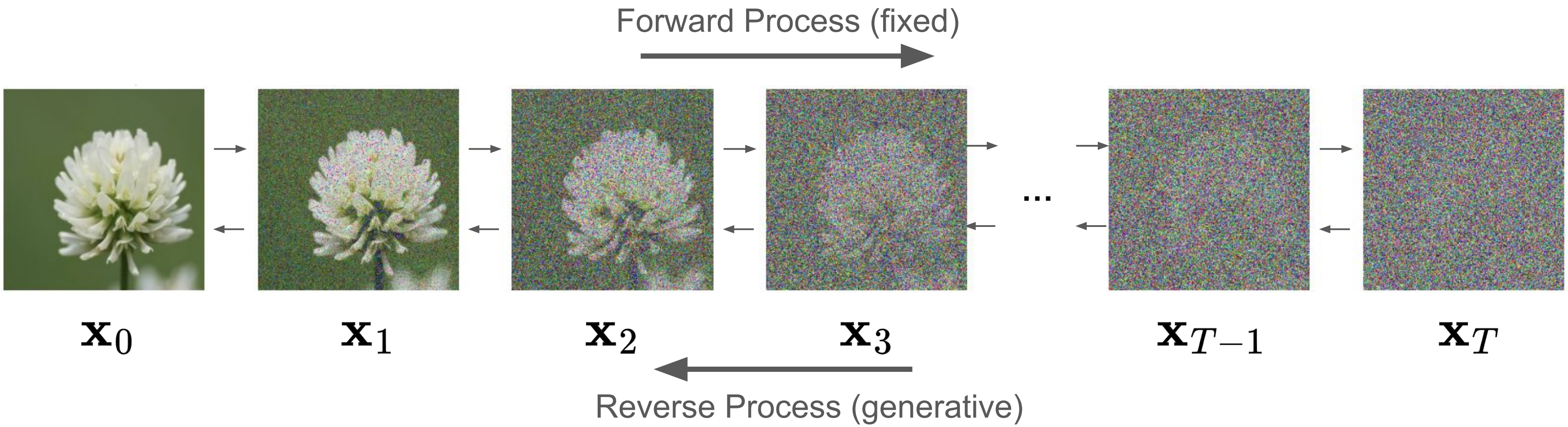
Dai, Xiaoliang, et al. "Emu: Enhancing image generation models using photogenic needles in a haystack." arXiv preprint arXiv:2309.15807 (2023).

Diffusion Overview

Denoising Diffusion Models

Denoising diffusion models consist of two processes:

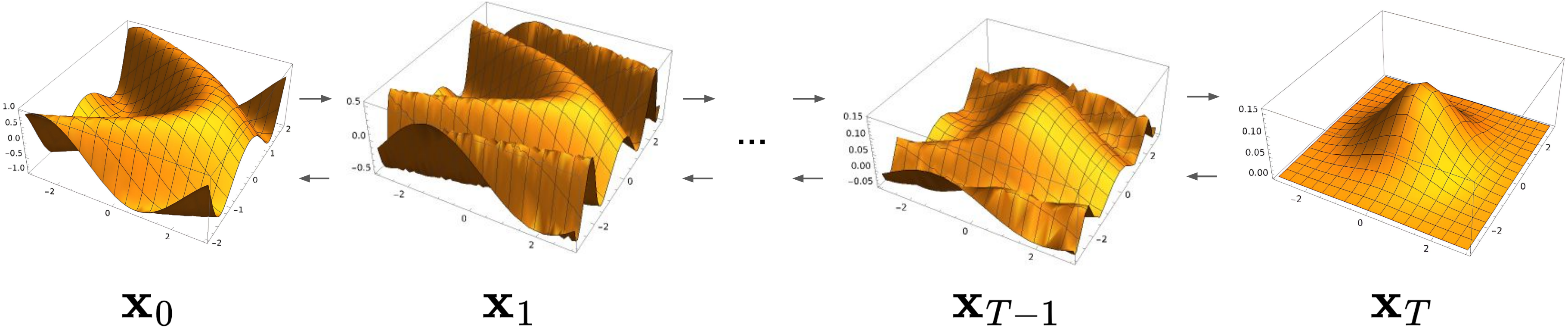
- Forward diffusion process that gradually adds noise to input
- Reverse denoising process that learns to generate data by denoising



Diffusion Models

We define a mapping to Gaussian noise (forward process)

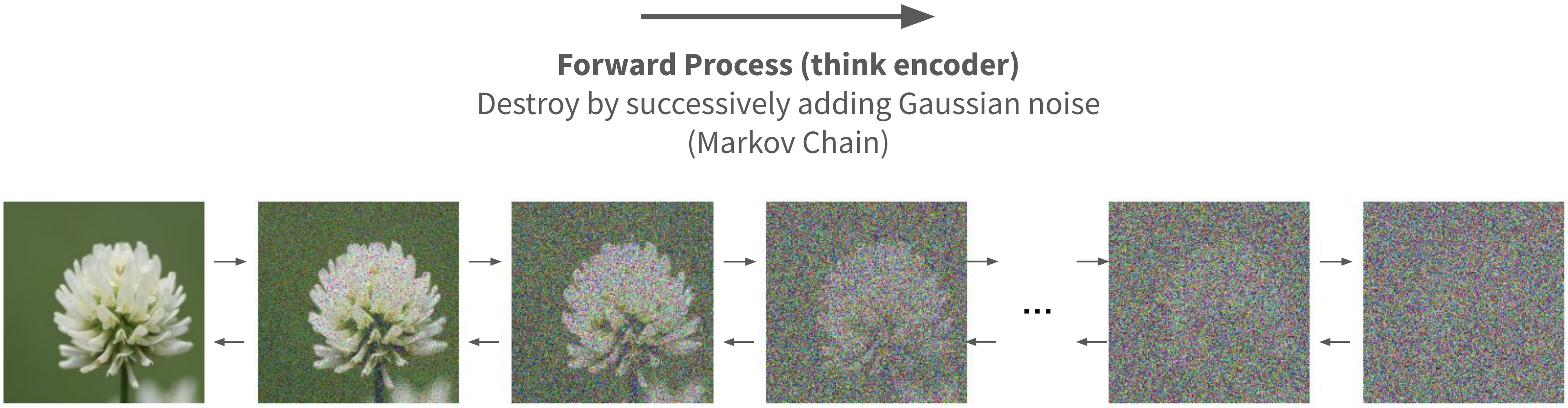
Want to **learn the reverse mapping to generate data** (reverse process)



Forward Process: high level idea

Forward Process (think encoder)

Destroy by successively adding Gaussian noise
(Markov Chain)

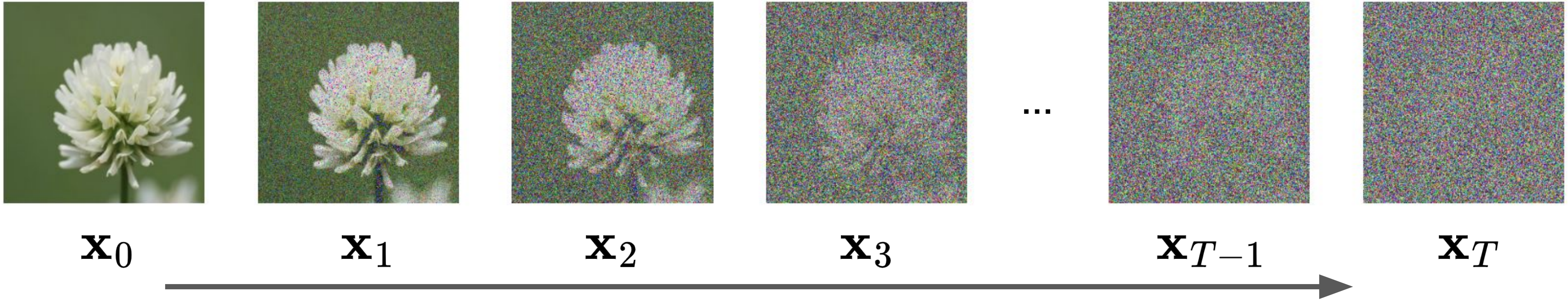


$\mathbf{x}_0$ $\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{x}_3$ ... $\mathbf{x}_{T-1}$ $\mathbf{x}_T$

Training
Sample

Gaussian Noise
(distribution)

Markov Chain Implications

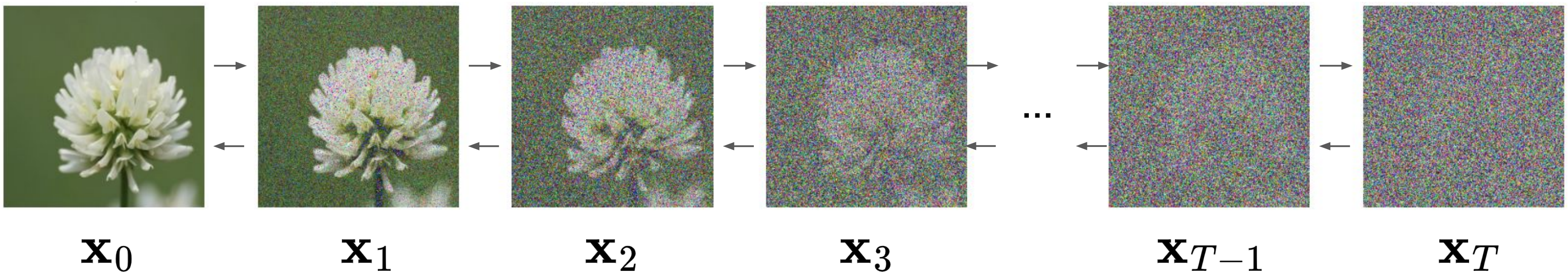


Direction of dependence

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) = q(\mathbf{x}_t|\mathbf{x}_{t-1})$$ T or F?

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = q(\mathbf{x}_{t-1}|\mathbf{x}_t)$$ T or F?

Reverse Process: high level idea



Training Sample

Gaussian Noise (distribution)

Reverse Process (think decoder)

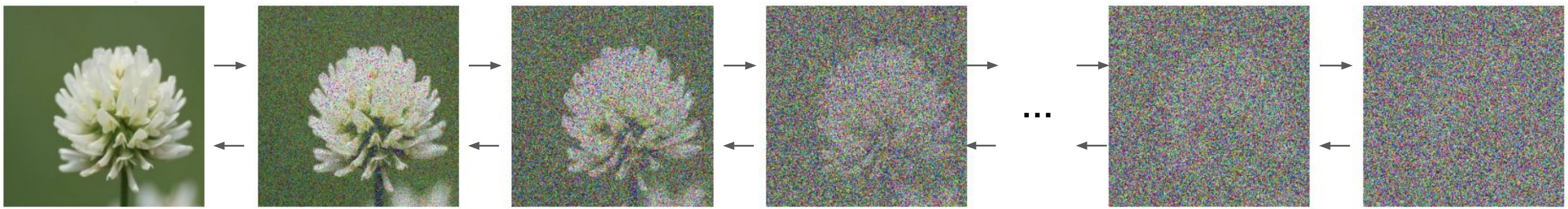
Learn to recover original image

(i.e. learn to reverse the Markov Chain)

Putting it together

Forward Process (think encoder)

Destroy by successively adding Gaussian noise
(Markov Chain)



$\mathbf{x}_0$ $\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{x}_3$ $\mathbf{x}_{T-1}$ $\mathbf{x}_T$

Training
Sample

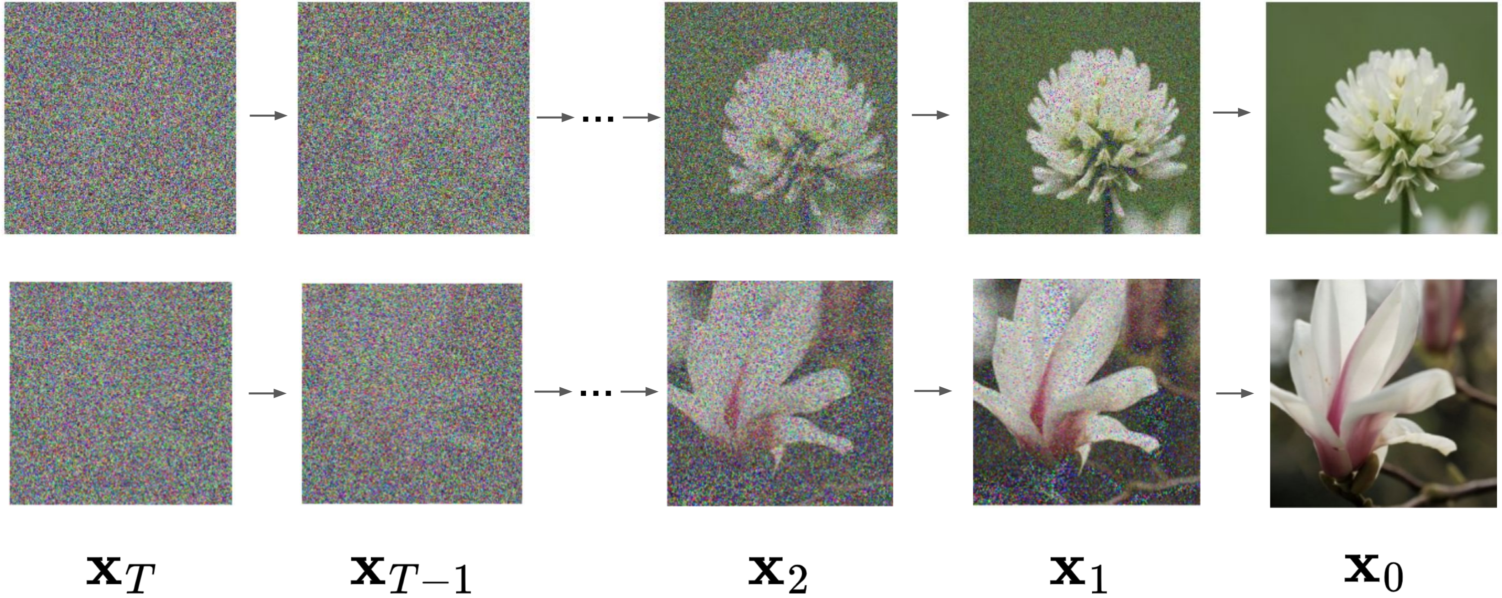
Gaussian Noise
(distribution)

Reverse Process (think decoder)

Learn to recover original image
(i.e. learn to reverse the Markov Chain)

Diffusion Sampling

Different draws of initial noise lead to diverse of outputs

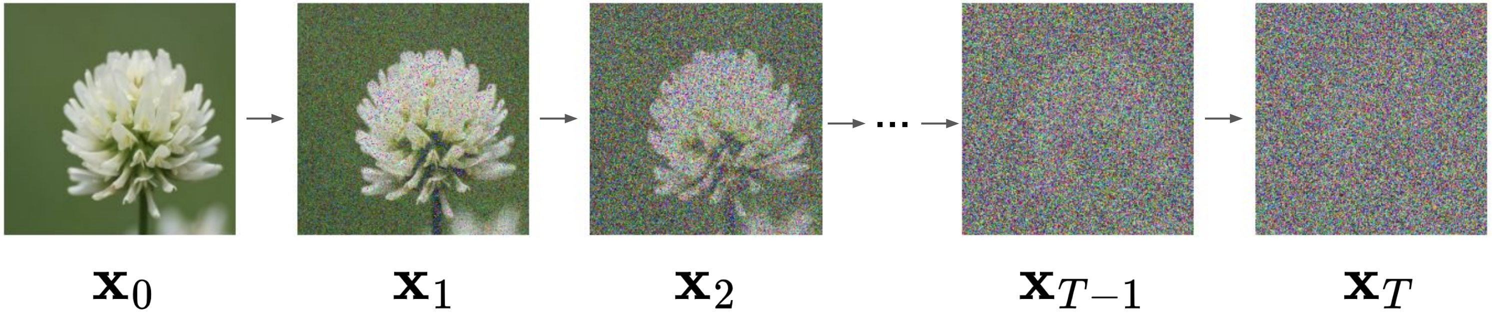


Forward Process

Forward Process Overview

- Destroys original image $\mathbf{x}_0$ by **successively adding Gaussian noise**
- Desired outcome: At step T , $\mathbf{x}_T$ is a **pure Gaussian noise**
  - i.e. the distribution we map the data manifold to

No training yet!!!



Details: Forward Process

Start from $\mathbf{x}_0$ sampled from some real-world distribution of images

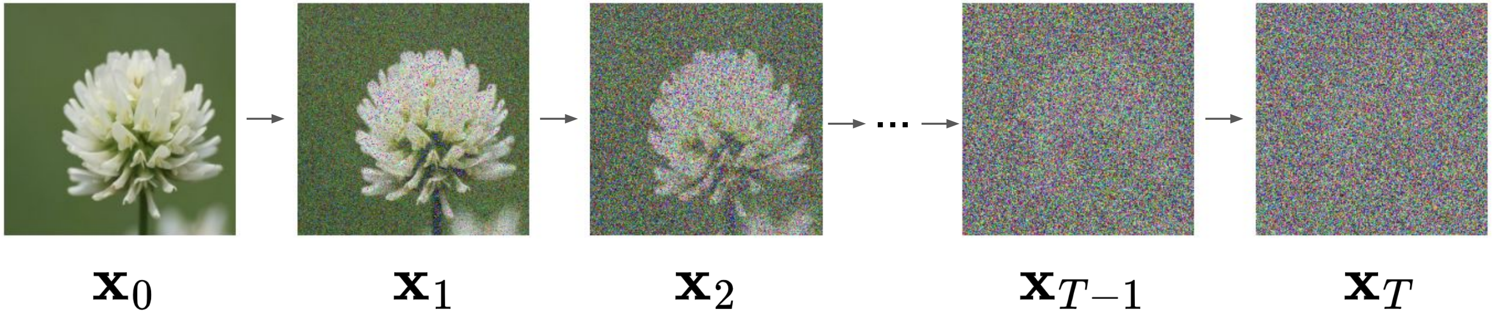
For timestamps until T:

$\mathbf{x}_t$ sampled from normal distribution conditioned on $\mathbf{x}_{t-1}$

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \qquad \{\beta_t \in (0,1)\}_{t=1}^{T}$$

$$q(\mathbf{x}_T) \approx \mathcal{N}(0, \mathbf{I})$$

noise schedule: how fast we move towards Gaussian noise



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

Details: Forward Process

Can we extend this to sampling $\mathbf{x}_t$ in a closed form?

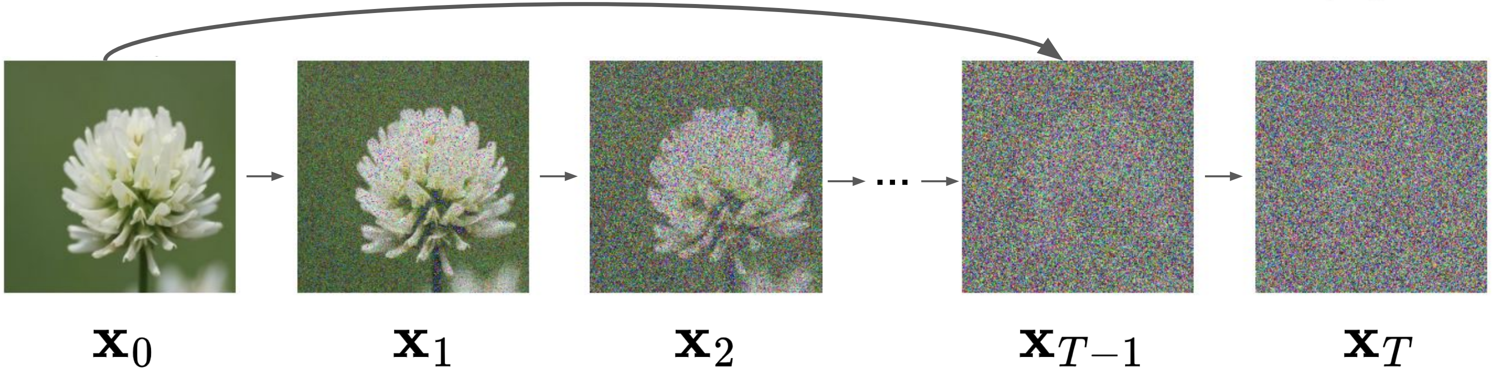
Let $\alpha_t := 1 - \beta_t$

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

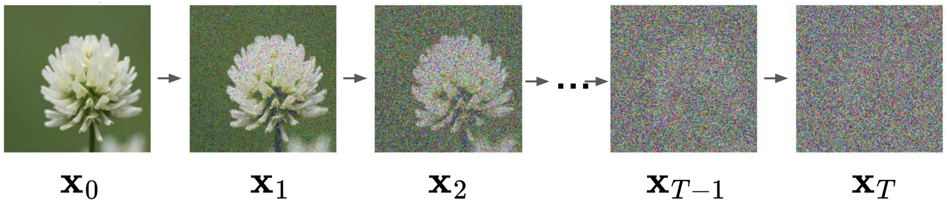
Re-parametrization trick!

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}$$

$$\boldsymbol{\epsilon}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$



Details: Forward Process



Inductively, we can say

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

Merged noise.
epsilon is still $\sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

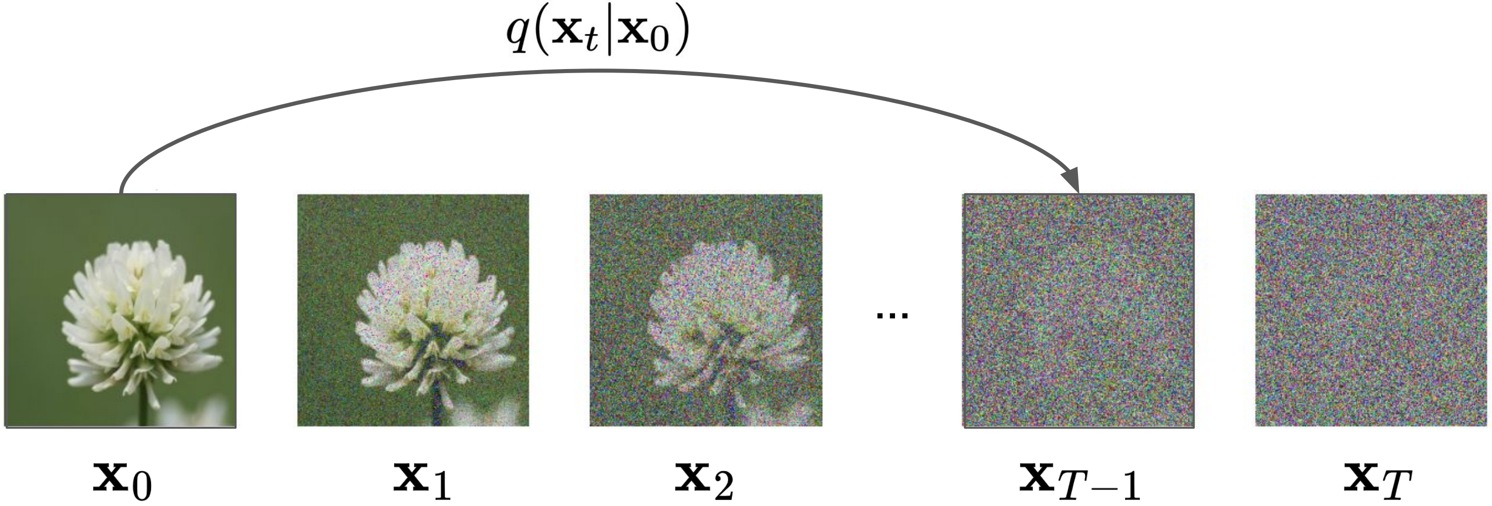
$$\boxed{\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i}$$

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

Details: Forward Process

Can sample $\mathbf{x}_t$ in closed-form as $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$

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



Aside: Noise Schedules

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

- Define the noise schedule in terms of $\bar{\alpha}_t \in (0,1)$
  - Some monotonically decreasing function from 1 to 0

- Cosine Noise schedule:

$$\bar{\alpha}_t = \cos(.5\pi t/T)^2$$

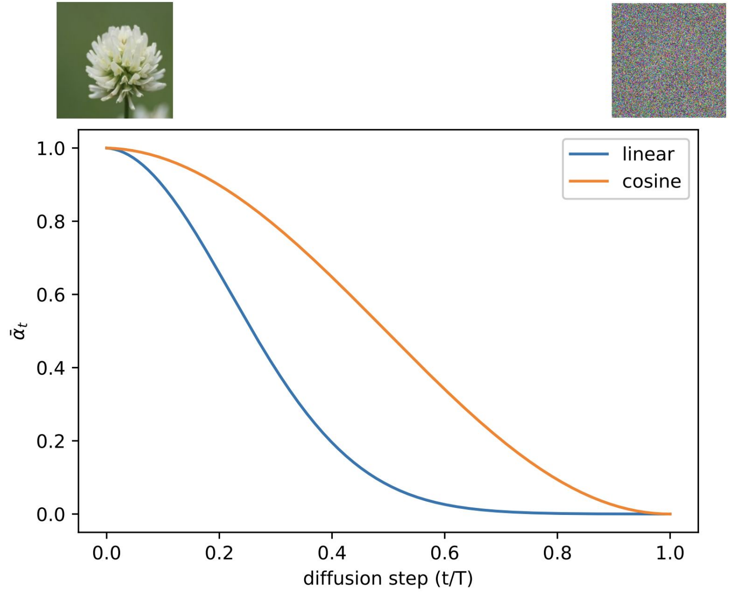


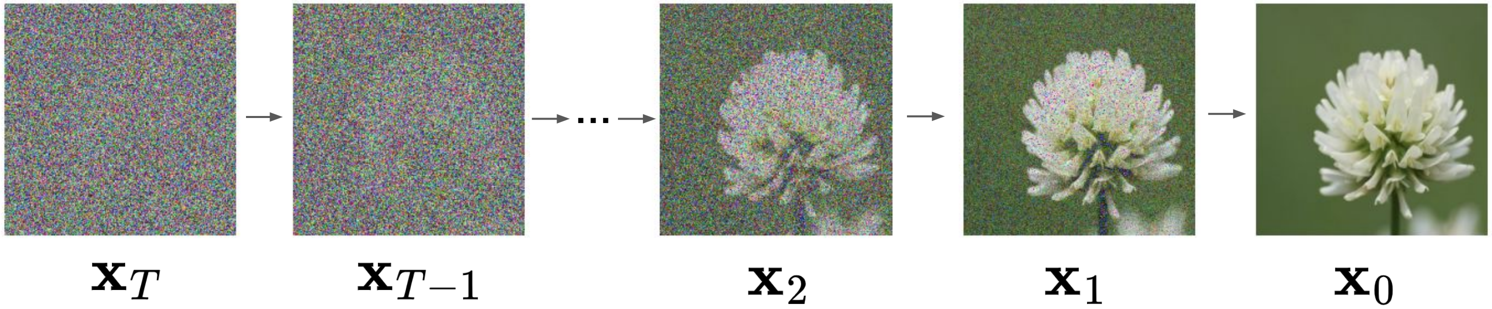
Figure 5. $\bar{\alpha}_t$ throughout diffusion in the linear schedule and our proposed cosine schedule.

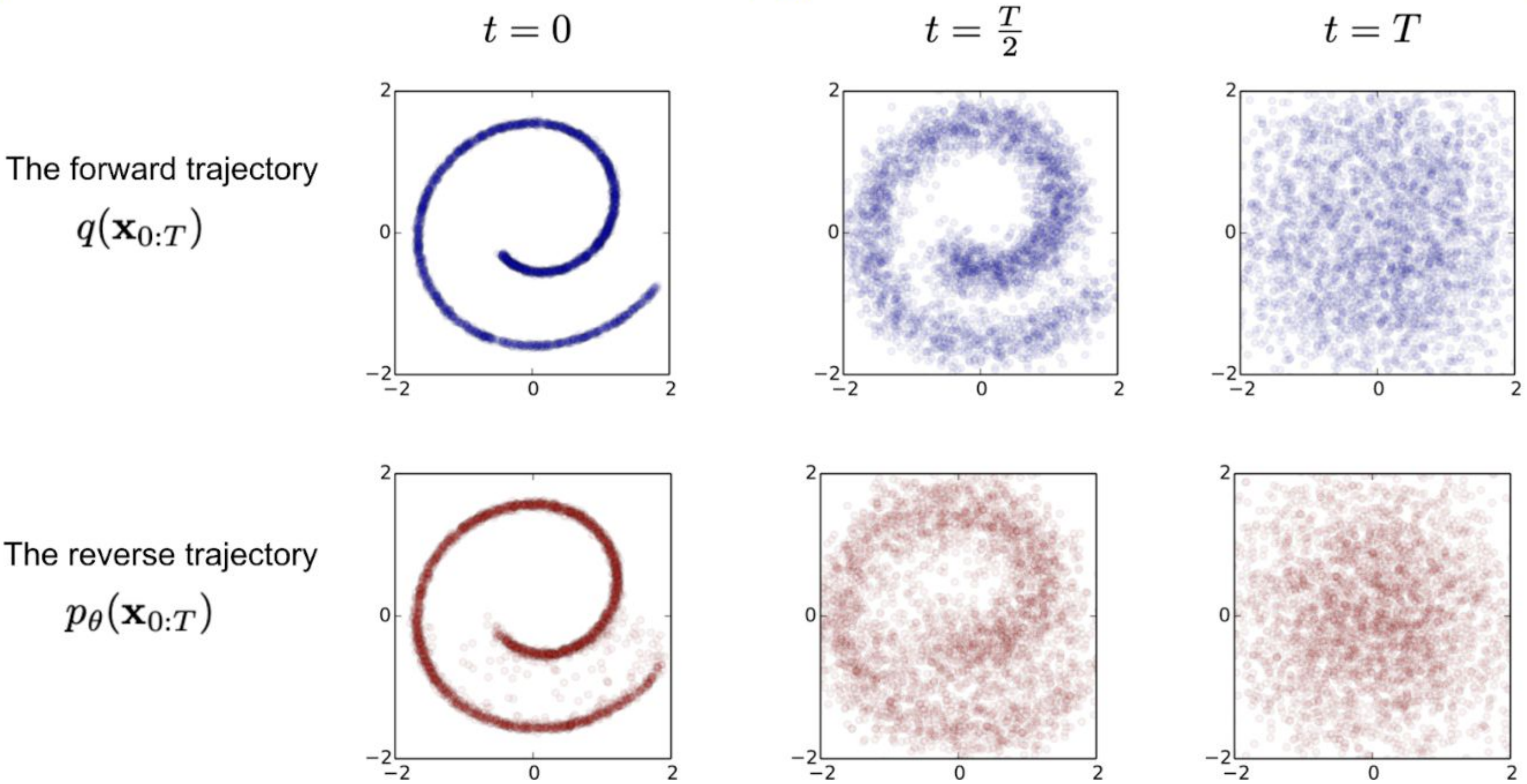
Nichol, Alexander Quinn, and Prafulla Dhariwal. "Improved denoising diffusion probabilistic models." International conference on machine learning. PMLR, 2021.

Reverse Process

Reverse Process Overview

- "Learn to reverse what we just destroyed"
  - Learn time reversal of Markov Chain; we **train a model for this**
- Desired outcome: some $\mathbf{x}_0$ close to the original data distribution





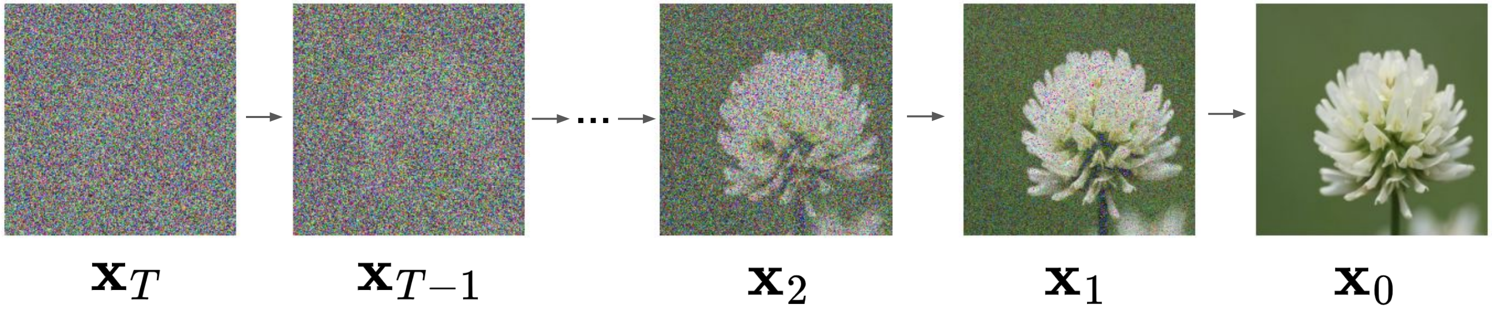
Sohl-Dickstein et al., 2015

Details: Reverse Process

Start from $q(\mathbf{x}_T) = \mathcal{N}(0, \mathbf{I})$

Ideally, sample from reversed conditional distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$

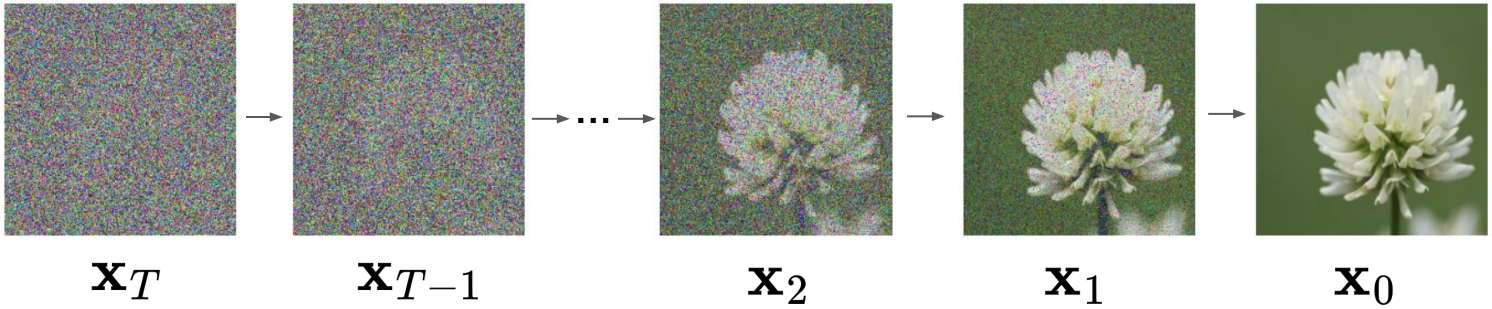
How to compute $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$?



Details: Reverse Process

How to compute $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$?

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})q(\mathbf{x}_{t-1})}{q(\mathbf{x}_t)}$$



Recall: forward

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

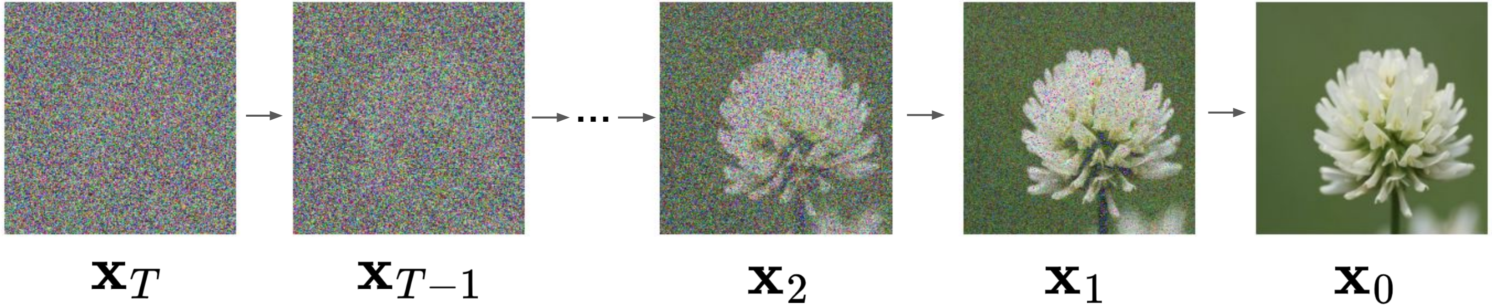
$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

Details: Reverse Process

$q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ Is **not tractable**. Is $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ tractable?

I.e. Can we reverse the forward process given the original data?

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}$$



Cornell Bowers C·IS

Details: Reverse Process

Recall: forward

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

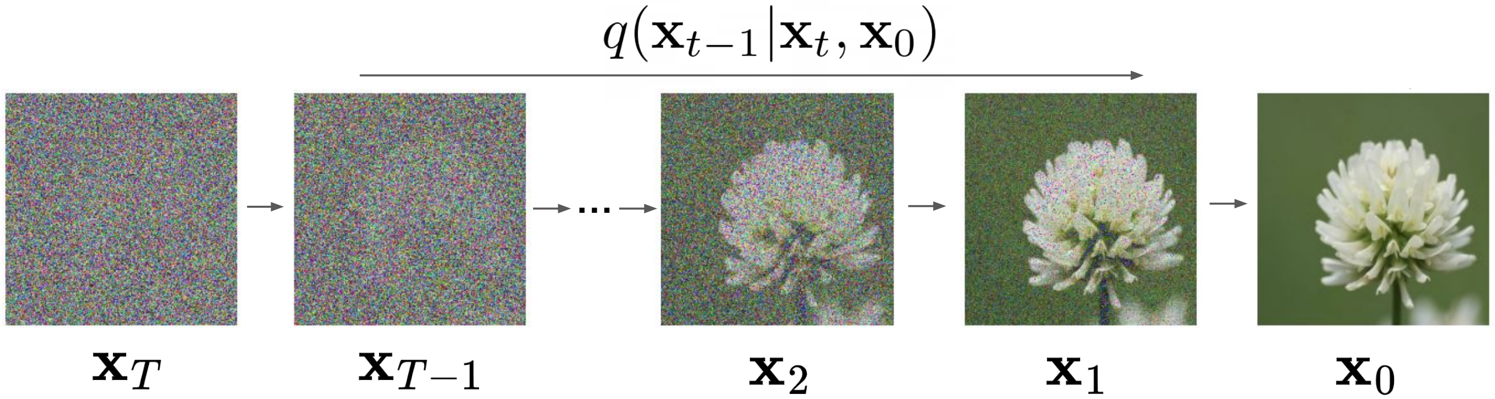
$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ is **tractable**

Can reverse the forward process given the original data!

Problem: Don’t have any “original data $\mathbf{x}_0$” during **inference**

We have $\mathbf{x}_0$ during **training**; train a **generative model**



Key Idea

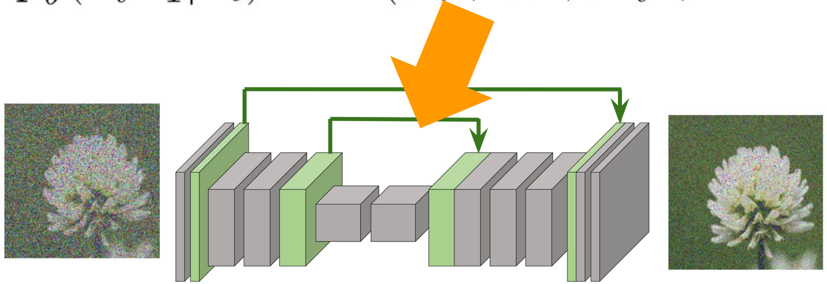
We introduce a generative model to approximate the reverse process $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$

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

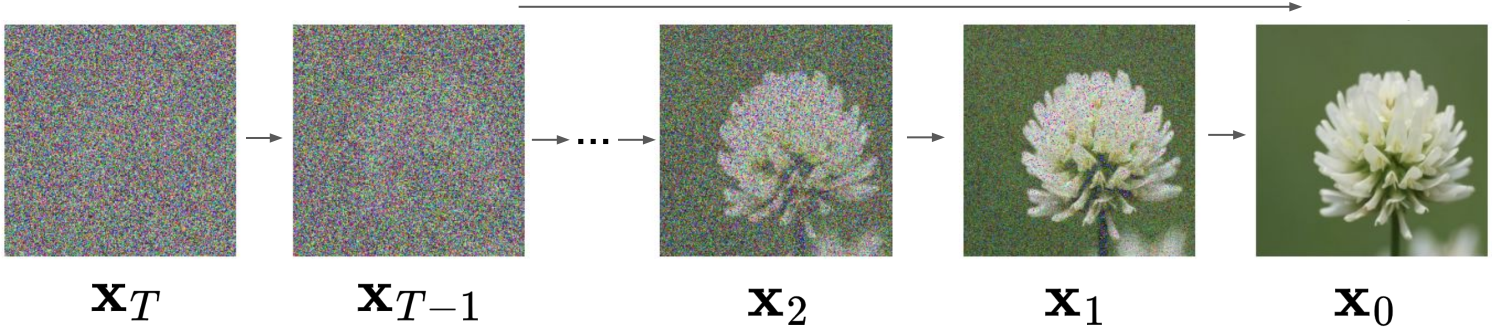
$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$$

$$\boxed{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))\right]}$$

Learning Objective!



$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$



Diffusion
Training Objective

Training

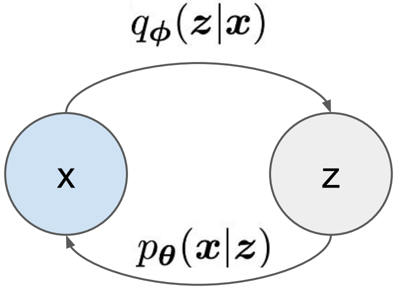
Find the model that maximizes the likelihood of the training data

i.e. same as VAEs, variational inference; approximate the true posterior

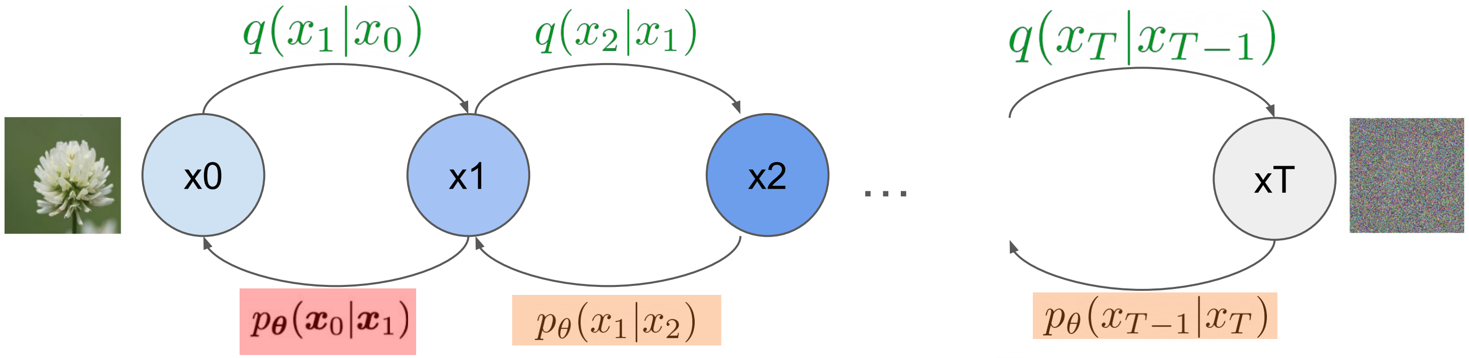
max $\log p(\boldsymbol{x})$

Training Objective

- Bound the likelihood with the ELBO
  - Exactly like VAEs


$q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$
x
z
$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$

$$\log p(\boldsymbol{x}) \geq \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z}))}_{\text{prior matching term}}$$



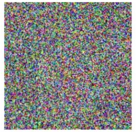
$$\log p(\boldsymbol{x}) \geq \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))]}_{\text{denoising matching term}}$$

http://cs231n.stanford.edu/slides/2023/lecture_15.pdf

Parameterizing the Denoising Model

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

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



$$-\sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))\right]}_{\text{denoising matching term}}$$

$$\mathbb{E}_q\left[\frac{1}{2\sigma_t^2}||\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_t, t)||^2\right] + C$$

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right)$$

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boxed{\epsilon_\theta(\mathbf{x}_t, t)}\right)$$

$$L(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}[||\epsilon - \epsilon_\theta(\mathbf{x}_t, t)||^2]$$

Simplifying KL Divergence to MSE of means, as distributions are Gaussians with same variance!

Simplifying Bayes Rule…

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}$$

Re-parametrize $\mu_\theta(x_t, t)$

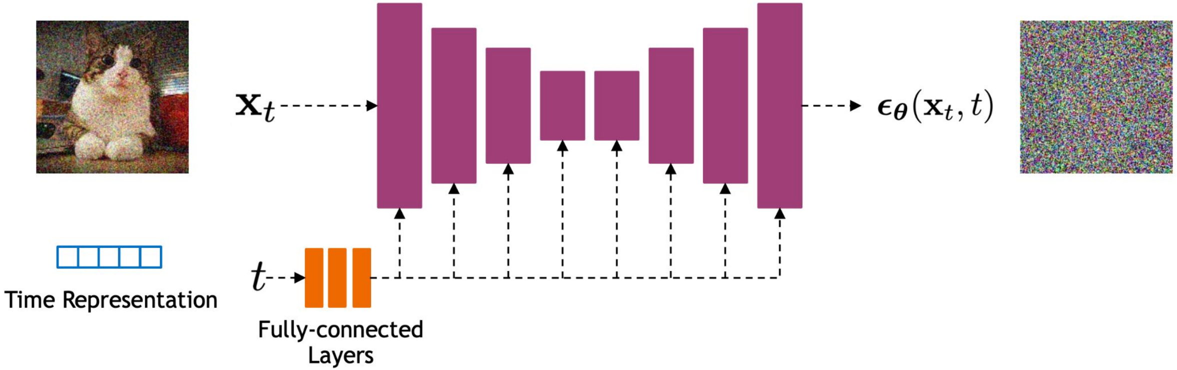
Loss is MSE of actual to predicted loss!

http://cs231n.stanford.edu/slides/2023/lecture_15.pdf

What Network Architecture to Use For ϵ_θ ?

People often use U-Nets with residual blocks and self-attention layers at low resolutions

Has same input and output image dimensions



Time representation: sinusoidal positional embeddings

Inject time embedding throughout the network (e.g. additive positional embedding)

http://cs231n.stanford.edu/slides/2023/lecture_15.pdf

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

Training Algorithm

```
Repeat until convergence
1. $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
2. $t \sim U\{1, 2, \ldots, T\}$
3. $\epsilon \sim \mathcal{N}(0, 1)$
4. Optimizer step on $L(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}[||\epsilon - \epsilon_\theta(\mathbf{x}_t, t)||^2]$
```

← Sample original image from image distribution

← Sample random time step uniformly

← Sample Gaussian noise

← Model predicts noise applied at time step t and calculate loss

Sampling Algorithm

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

← Sample pure Gaussian noise

$$\text{For } t = T, T-1 \ldots, 1$$

$$\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \text{ if } t > 1, \text{ else } \mathbf{z} = \mathbf{0}$$

← Sample Gaussian noise to apply to image

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

← Predict noise applied to image and remove that noise

$$\text{Return } \mathbf{x}_0$$

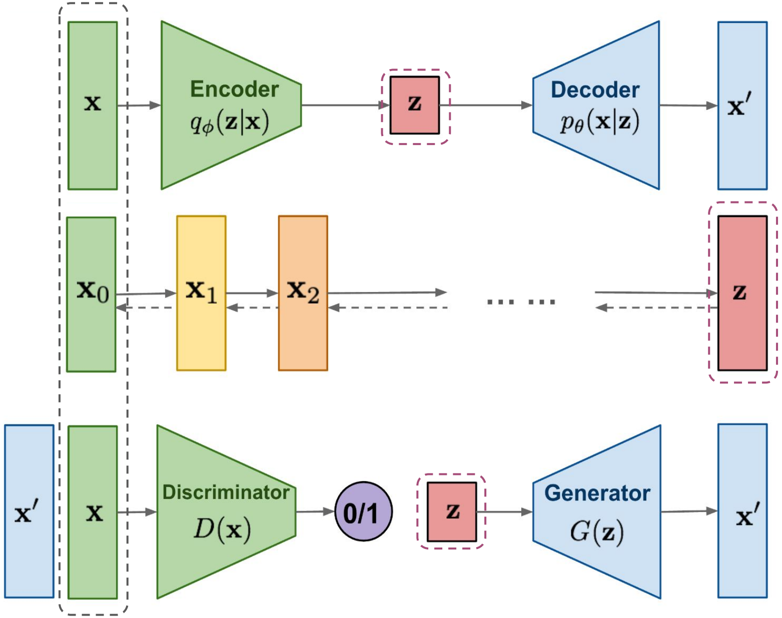
$$\boxed{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \mathbf{x}_\theta(\mathbf{x}_t, t))}$$

Generative Modeling

VAE: maximize variational lower bound

Diffusion models: Gradually add Gaussian noise and then reverse

GAN: Adversarial training



VAEs typically have a smaller latent dimension, while diffusion models do not

Latent Distribution

Target Distribution

Image Source

Recap

- Can bound the likelihood of observed data (i.e. the evidence) with the Evidence Lower Bound (i.e. the ELBO)
- Can learn generative models by maximizing the ELBO
    - VAEs, hierarchical VAEs, Diffusion models
- Learning objective decomposed to each timestep
    - Can be made extremely deep!
    - Can focus on higher noise levels to improve perceptual quality!
- Limitation:
    - Can require many sampling steps for good quality