Adv ML for Gen-AI

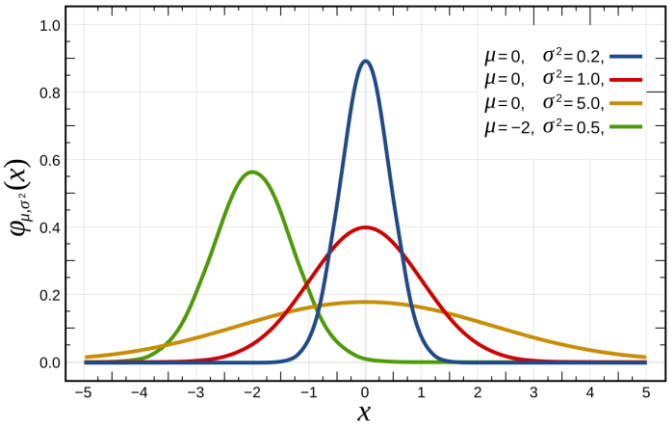
Diffusion

https://ml-graph.github.io/spring-2026/

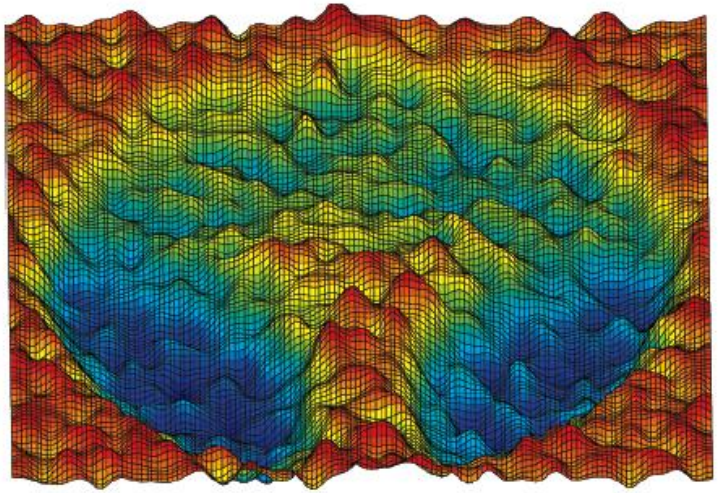
Yu Wang, Ph.D.
Assistant Professor
Department of Computer Science
University of Oregon
Personal: https://yuwang0103.github.io/
Lab: https://kindlab-fly.github.io/

Summary

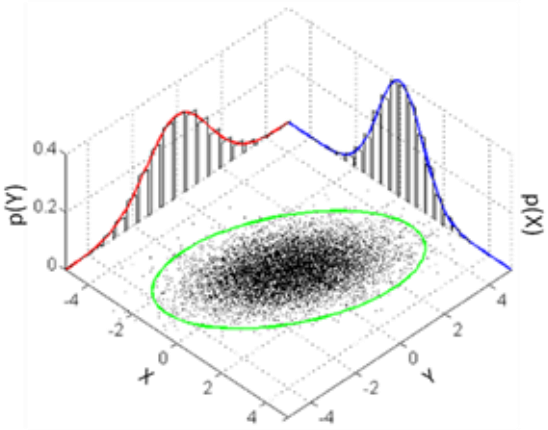
1D Gaussian Distribution



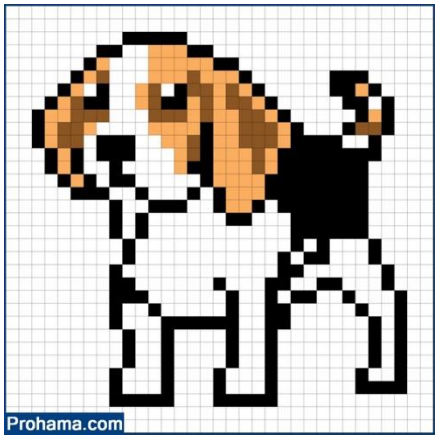
$\mathbb{R}$

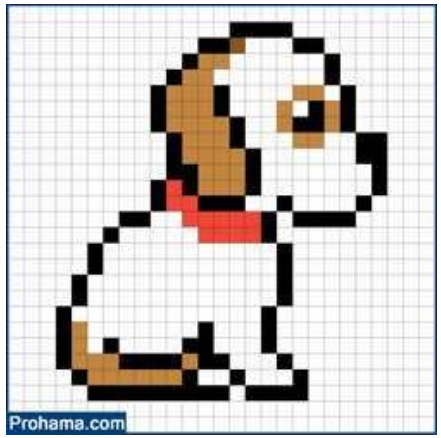


2D Gaussian Distribution



$\mathbb{R}^2$





$\mathbb{R}^{256\times256}$

Summary

Probability distribution of the **objective** based on the **observed data**

$$\{x_i\}_{i=1}^{N} \xrightarrow[\text{Good Model}]{} P(x) \xrightarrow[\text{Good Data}]{} x$$

- **Machine Learning Methods**
  - Gaussian Kernel Density Estimation
  - Gaussian Mixture Models

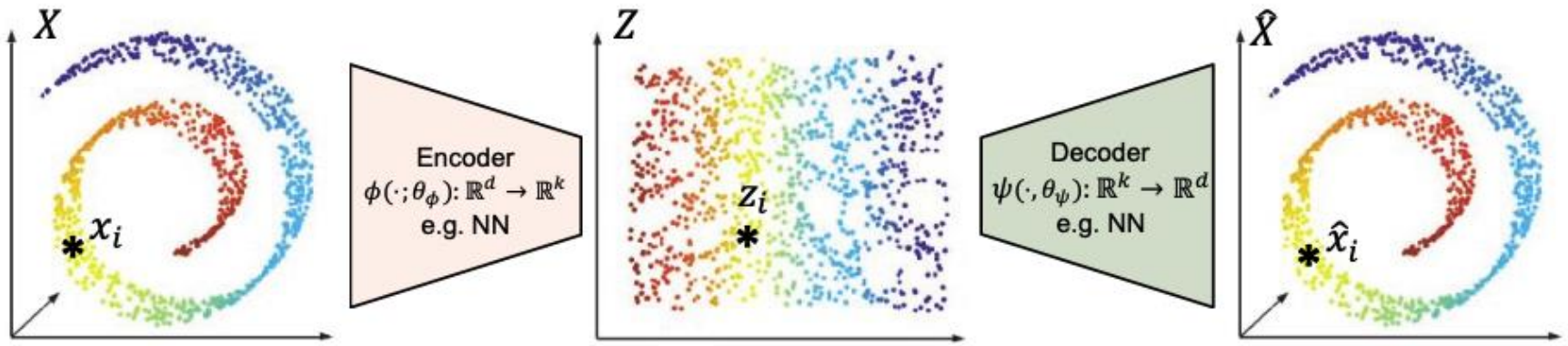
Using existing function to estimate what you do not know that can best fit your observation

- **Deep Learning Methods**
  - Auto-Encoder (AE)
  - Variational AE (VAE)
  - Generative Adversarial Network (GAN)
  - Diffusion Model
  - ……

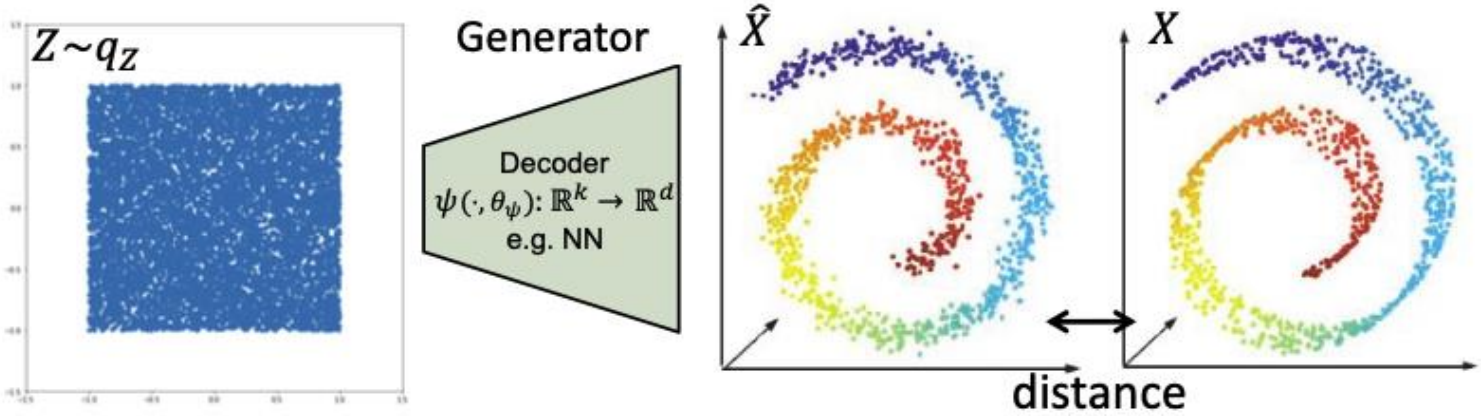
Using learnable function to estimate what you do not know that can best fit your observation

Summary

Auto-encoder based generative modeling

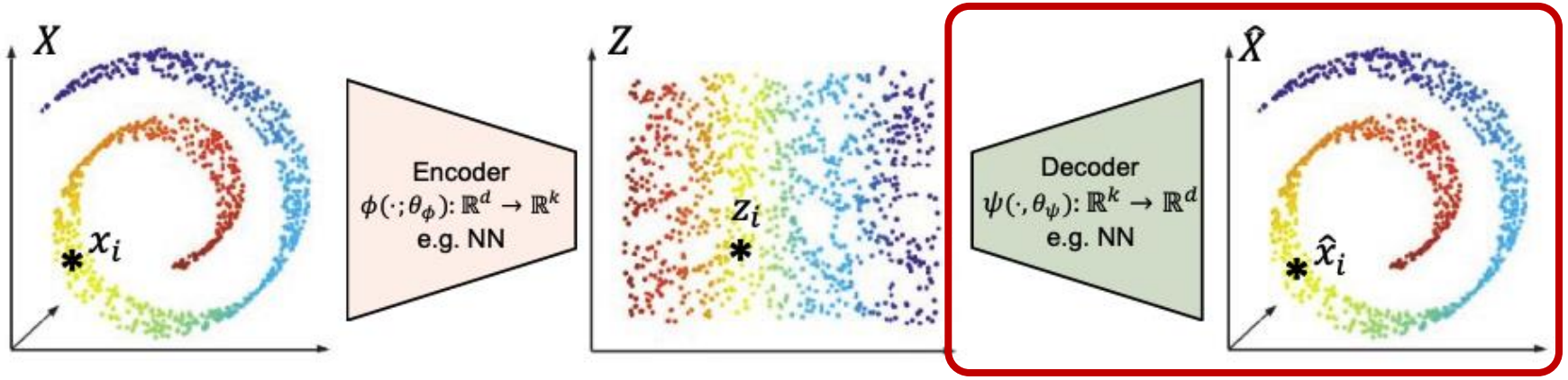


Encoder-less Generative Modeling

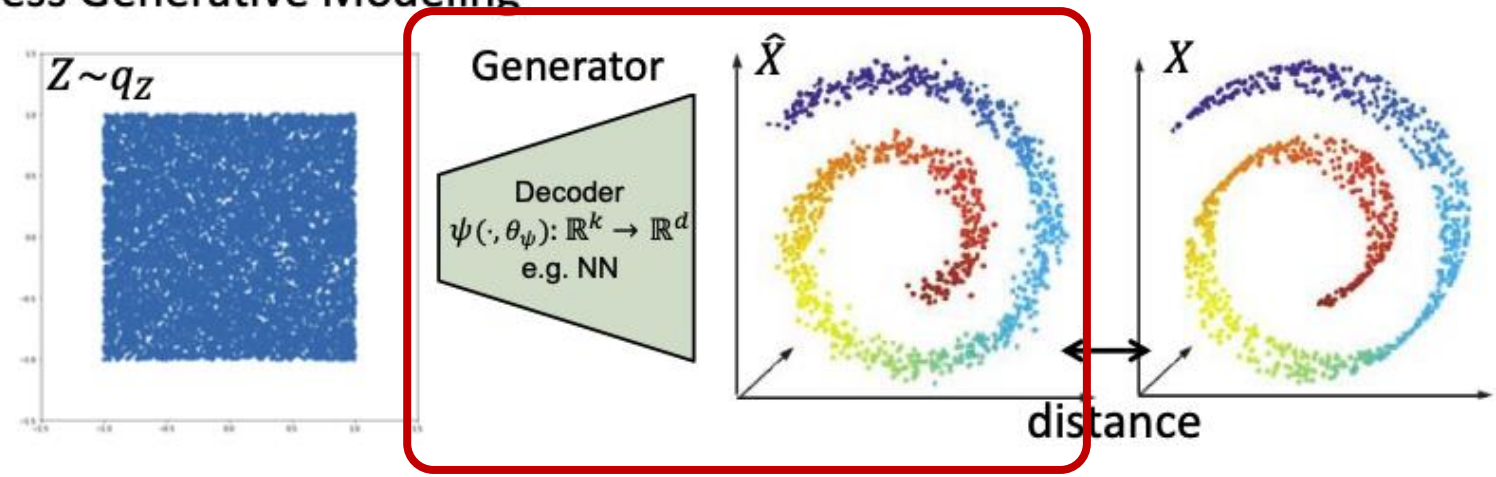


Summary – GAN/AE(VAE) shares the same decoder architecture

Auto-encoder based generative modeling



Encoder-less Generative Modeling



One-step Decoder-based Generation is too Difficult

One-shot Generation

Standard Gaussian —**Too difficulty**→ **Target Distribution**

Motivation of Diffusion

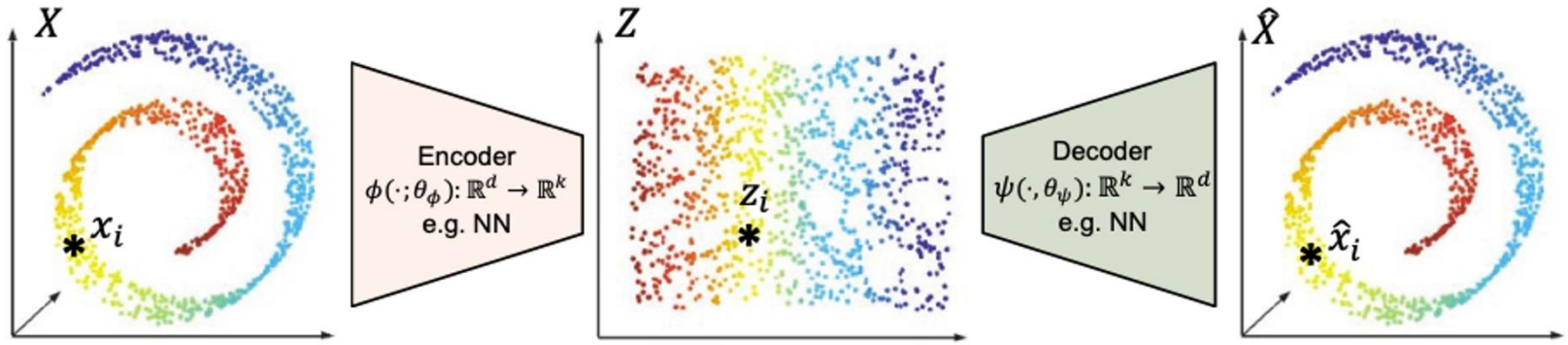
Generation by denoising step by step

Noise → **Data**

x_T x_6 x_5 x_4 x_3 x_2 x_1

Predicted **Ground-truth**

Generation by one step $\mathcal{F}(x)$ $\|\hat{x} - x\|_2^2$



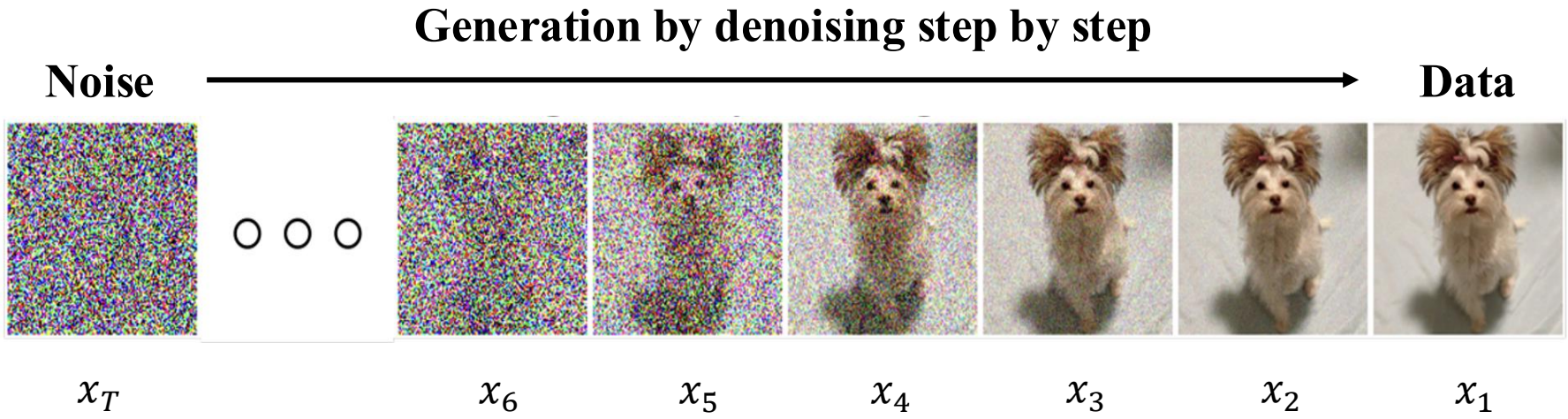
Summary

- **Motivation and Introduction of Diffusion**

- **Forward process of Diffusion**

- **Backward process of Diffusion**
  - **Backward progressive:** $x_t \rightarrow x_{t-1}$
  - **Backward one step:** $x_t \rightarrow x_0$
  - **Backward noise:** $x_t \rightarrow \epsilon$

Motivation of Diffusion

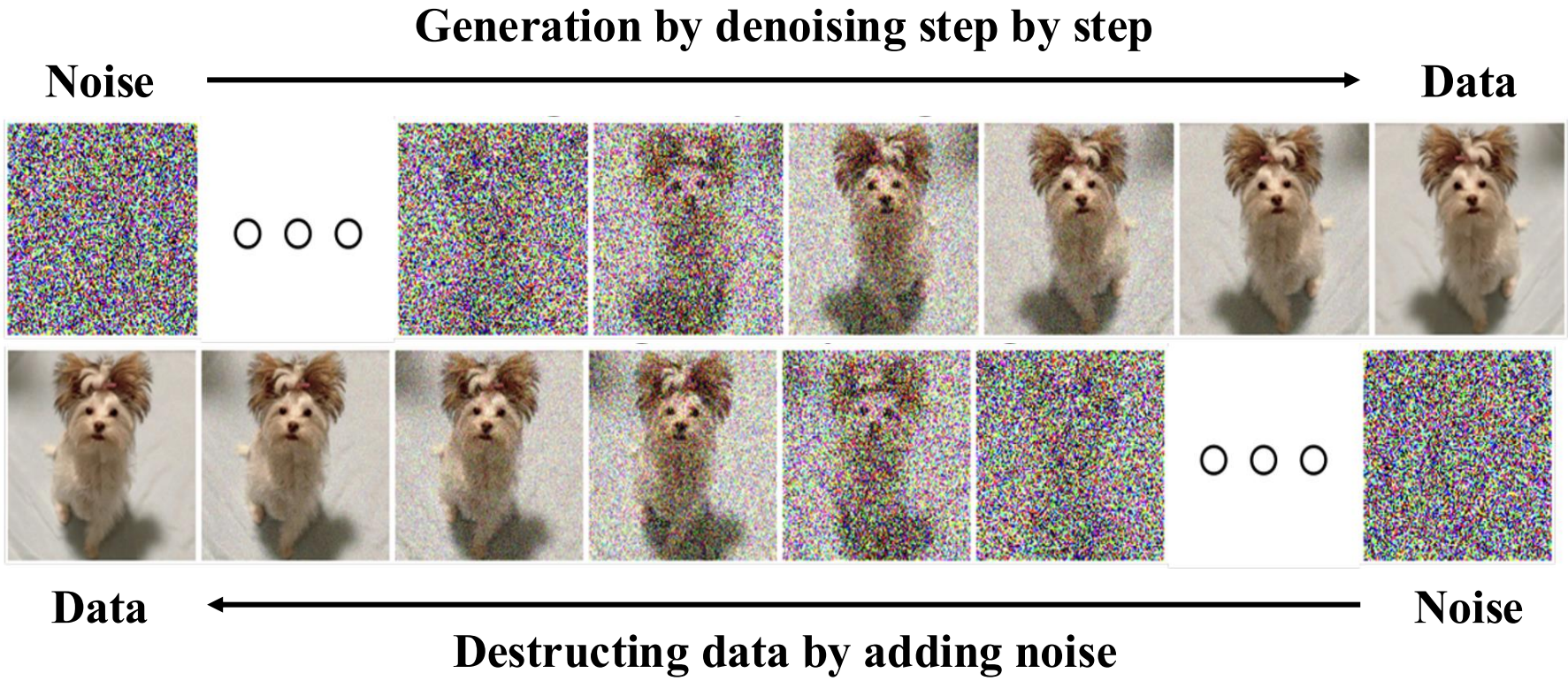


Predicted Ground-truth

$$\|\hat{x} - x_{t-1}\|_2^2 = \|\mathcal{F}(x_t, t) - x_{t-1}\|_2^2$$

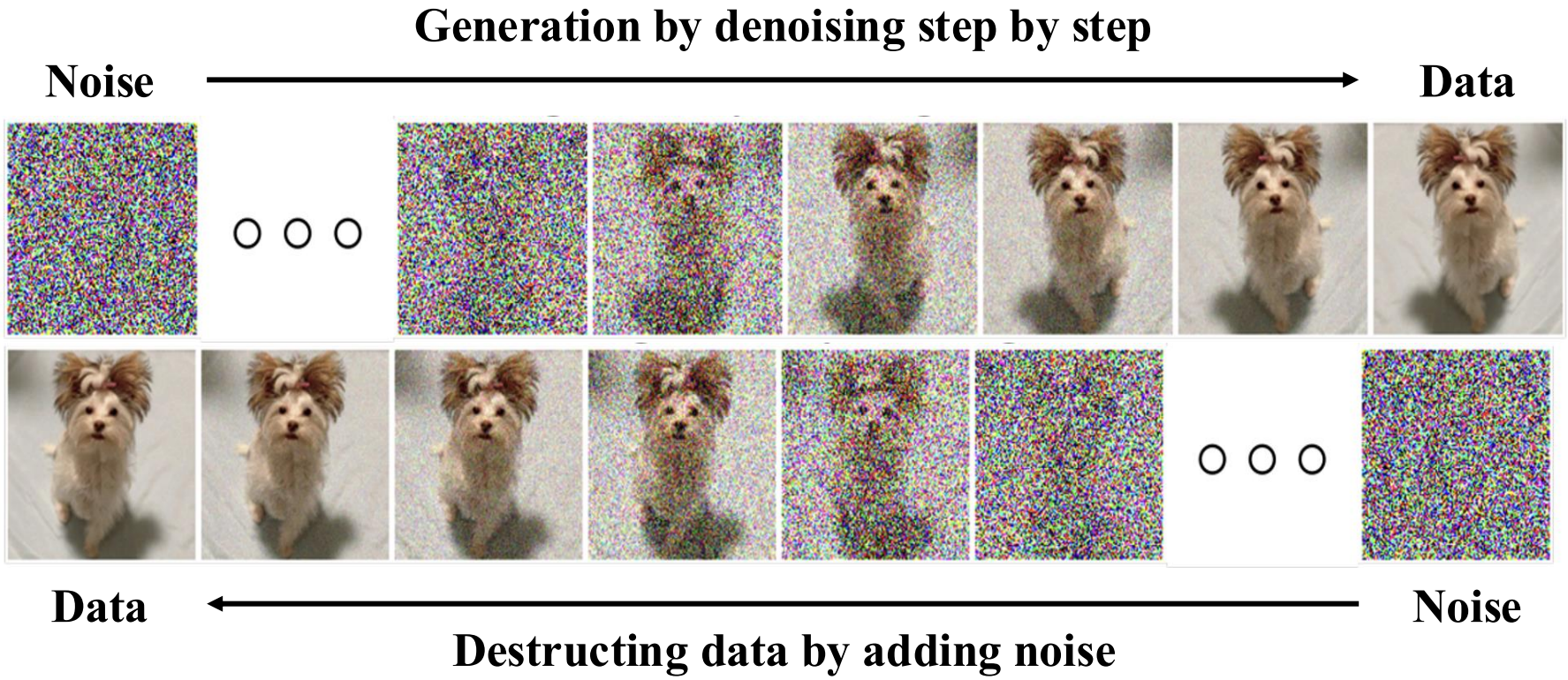
However, how do we obtain $\{x_t\}_{t=1}^T$?

Motivation of Diffusion



However, how do we obtain $\{x_t\}_{t=1}^{T}$? – Forward Process

Motivation of Diffusion



- **Forward Process (Data - Noise):** Gradually add noise to generate intermediate data
- **Backward Process (Noise - Data):** Learn to denoise the intermediate data and recover back

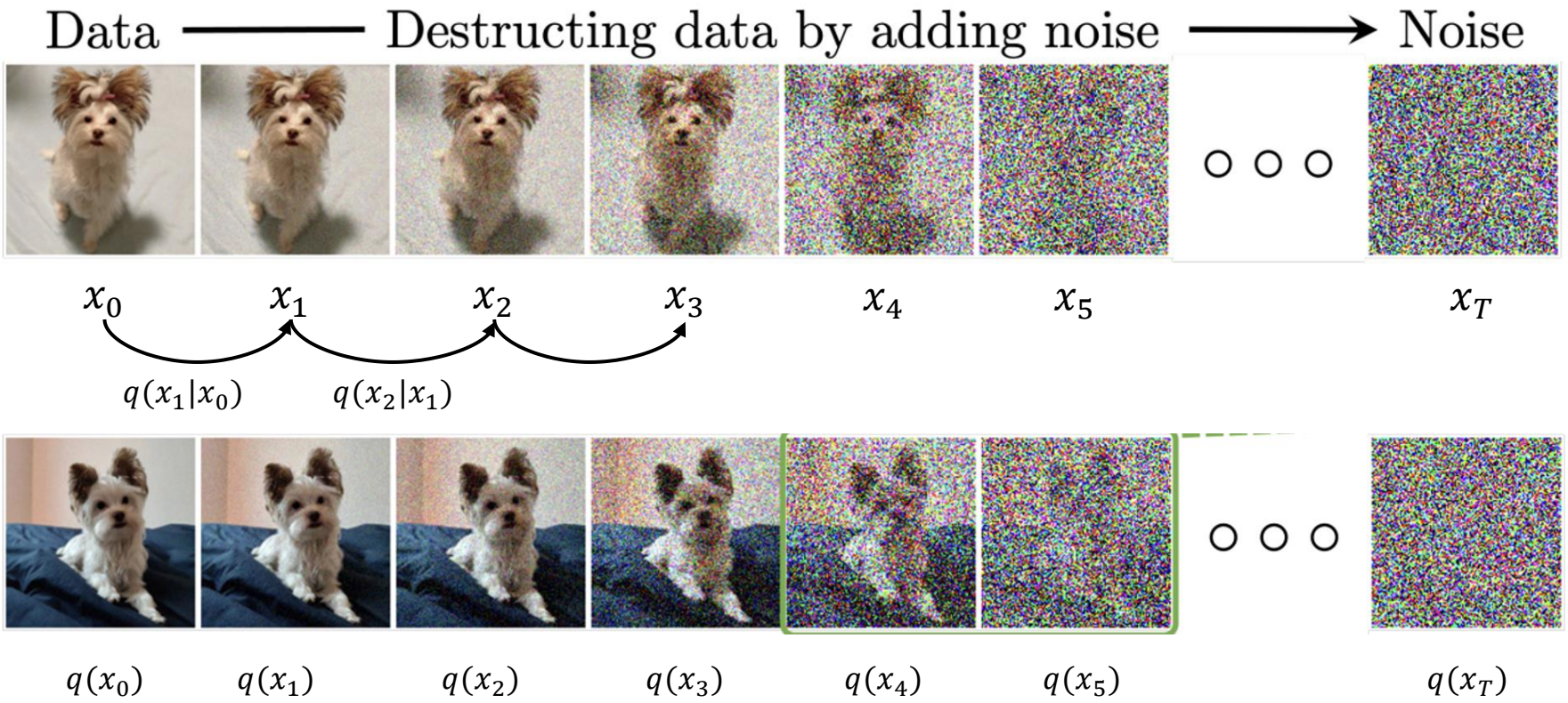
Summary

- **Motivation and Introduction of Diffusion**

- **Forward process of Diffusion**

- **Backward process of Diffusion**
  - **Backward progressive:** $x_t \rightarrow x_{t-1}$
  - **Backward one step:** $x_t \rightarrow x_0$
  - **Backward noise:** $x_t \rightarrow \epsilon$

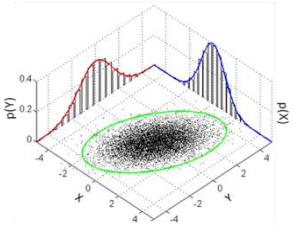
Forward Process



$q(x_t)$ **Data distribution at time step t**

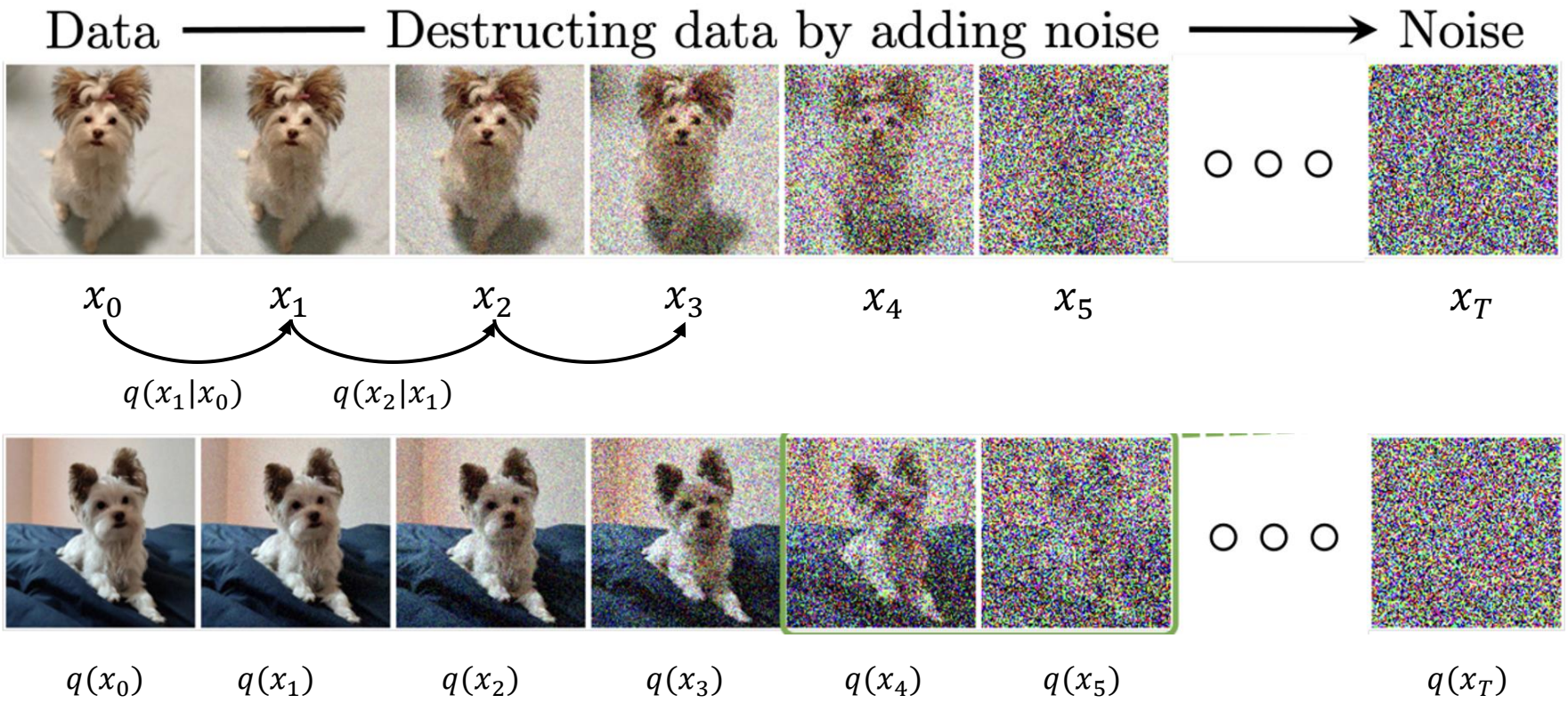
$q(x_t|x_{t-1})$ **Transitional distribution at time step t**

$q(x_0)$ **Start**



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

Forward Process



$q(x_0)$ $\quad x_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

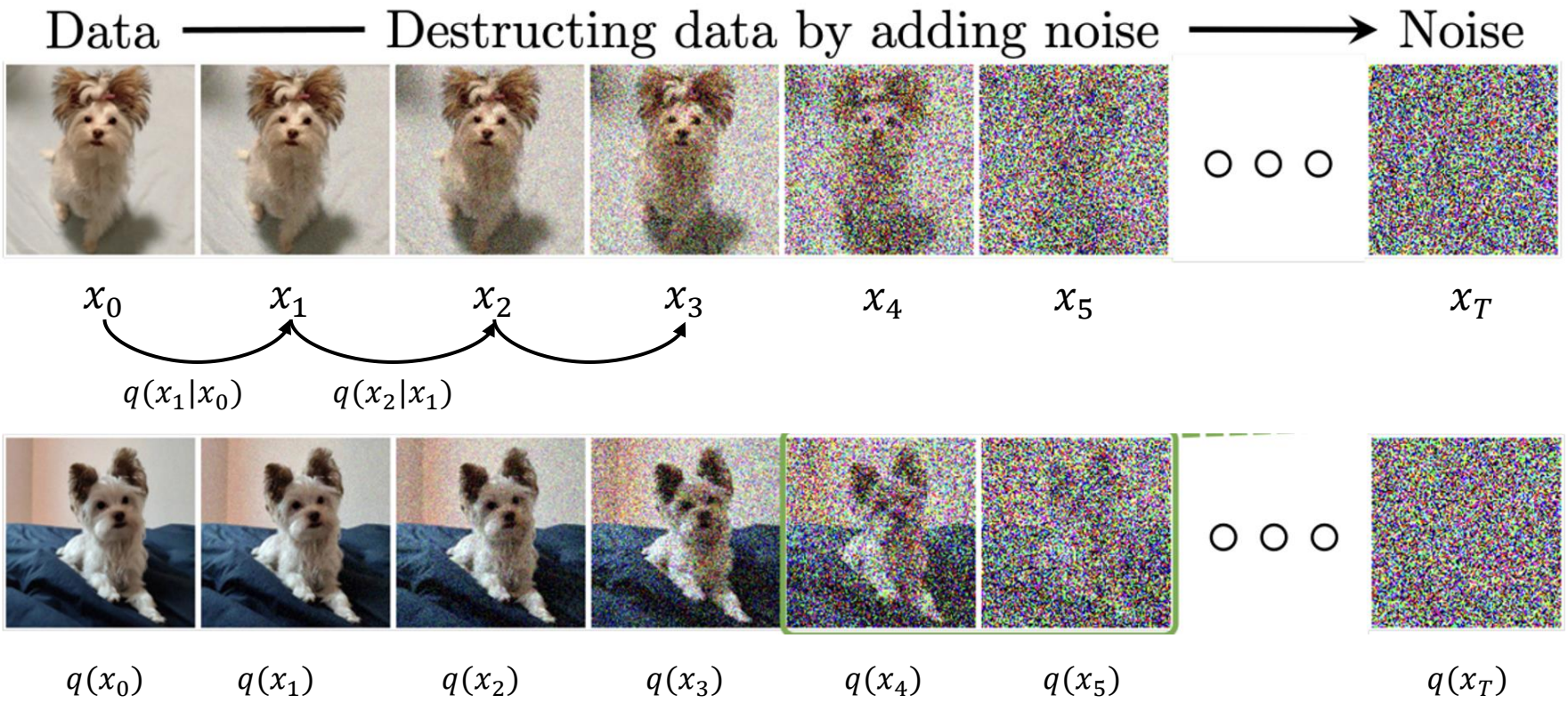
$\beta_t \in [0,1]$

Re-parametric trick (VAE)

$q(x_t|x_{t-1})$ $\quad x_t \sim \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t I)$ $\quad x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{(1-\alpha_t)}\epsilon, \epsilon \in \mathcal{N}(0, I)$

$x_t \sim \mathcal{N}(\sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I)$ $\quad x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{(1-\alpha_t)}\epsilon, \epsilon \in \mathcal{N}(0, I)$

Forward Process

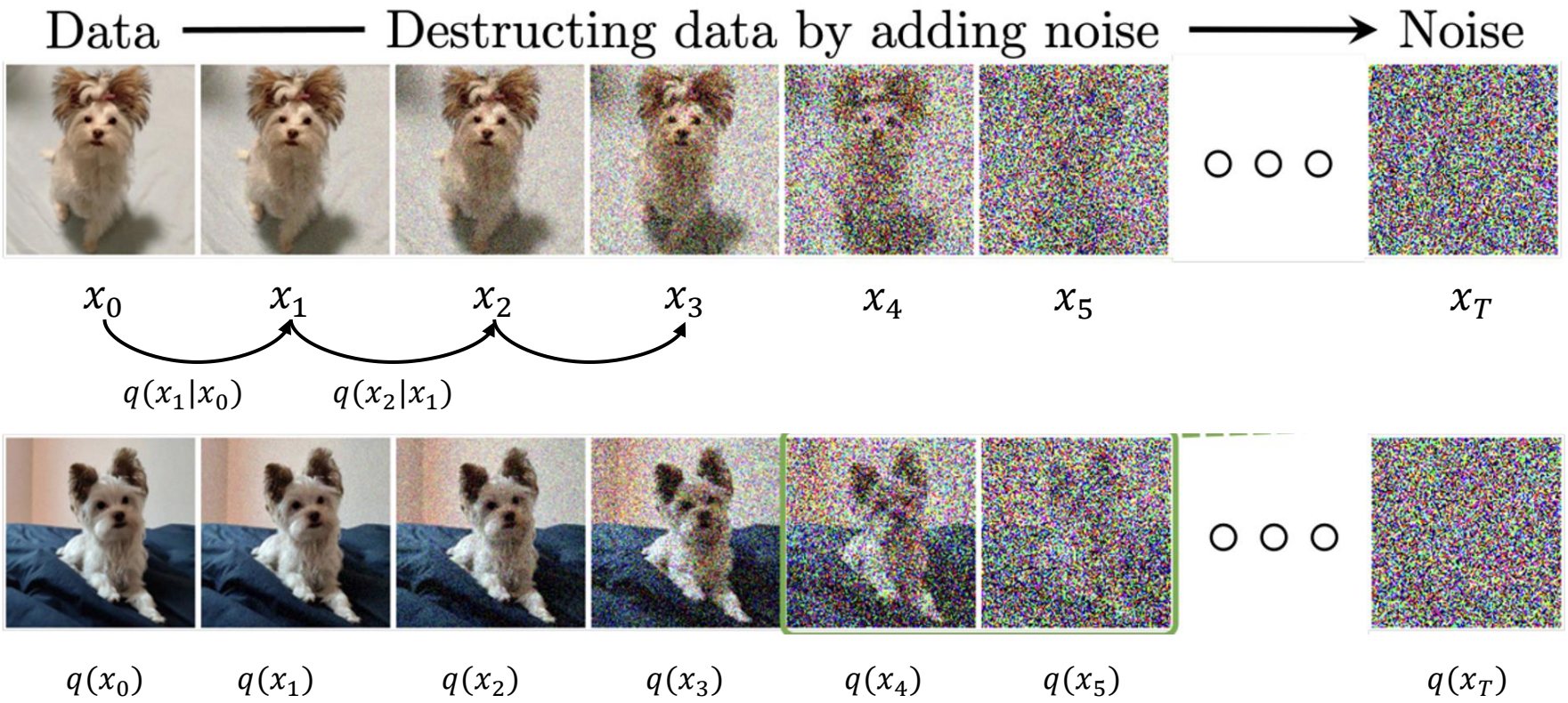


$$x_1 = \sqrt{\alpha_1}x_0 + \sqrt{(1-\alpha_1)}\epsilon_1 \qquad \epsilon_1 \sim \mathcal{N}(0, I) \quad \epsilon_2 \sim \mathcal{N}(0, I)$$

$$x_2 = \sqrt{\alpha_2}x_1 + \sqrt{(1-\alpha_2)}\epsilon_2 \qquad A\epsilon_1 + B\epsilon_2 \sim \mathcal{N}(0, (A^2 + B^2)I)$$

$$x_2 = \sqrt{\alpha_2}\left(\sqrt{\alpha_1}x_0 + \sqrt{(1-\alpha_1)}\epsilon_1\right) + \sqrt{(1-\alpha_2)}\epsilon_2 = \sqrt{\alpha_2\alpha_1}x_0 + \sqrt{\alpha_2}\sqrt{1-\alpha_1}\epsilon_1 + \sqrt{(1-\alpha_2)}\epsilon_2$$

Forward Process

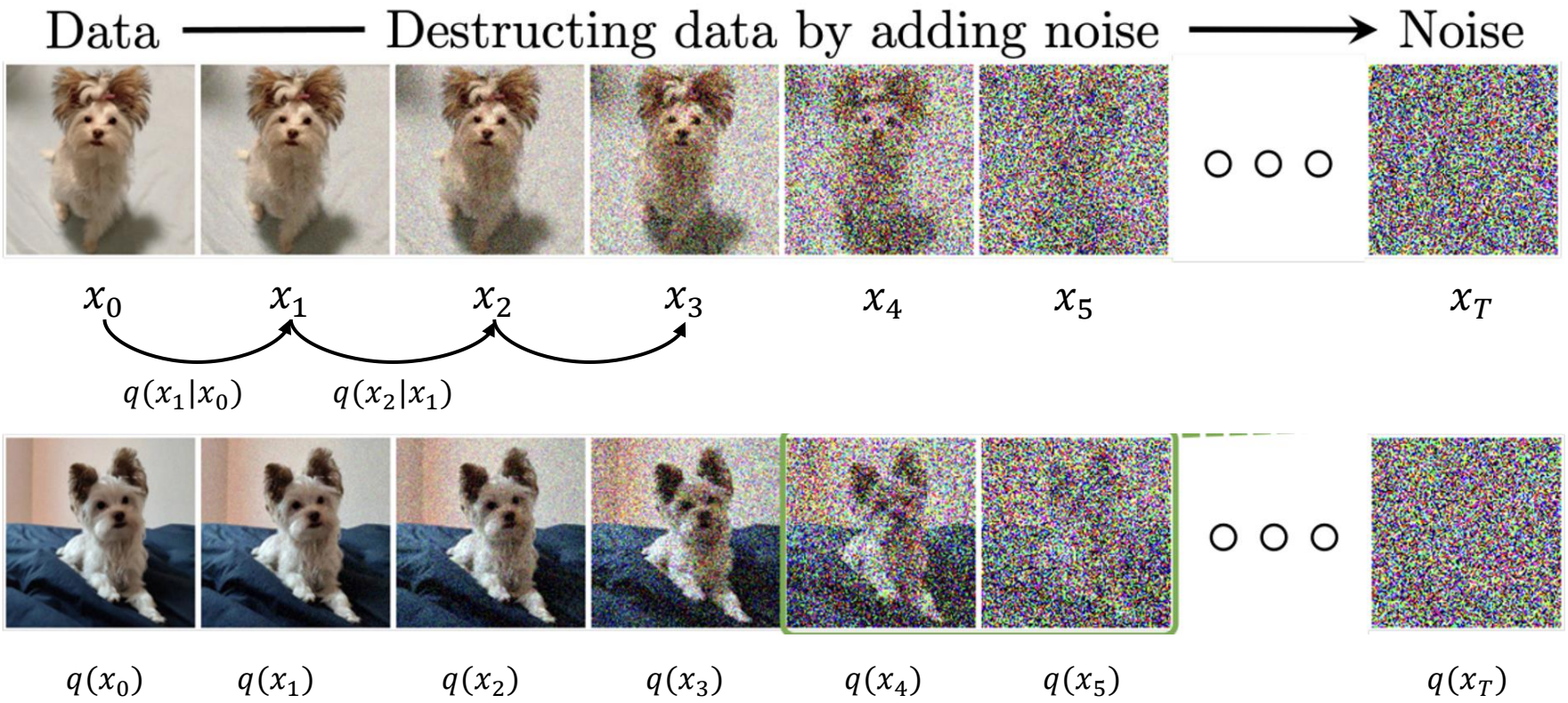


$$x_2 = \sqrt{\alpha_2}\left(\sqrt{\alpha_1}x_0 + \sqrt{(1-\alpha_1)}\epsilon_1\right) + \sqrt{(1-\alpha_2)}\epsilon_2 = \sqrt{\alpha_2\alpha_1}x_0 + \sqrt{\alpha_2}\sqrt{1-\alpha_1}\epsilon_1 + \sqrt{(1-\alpha_2)}\epsilon_2$$

$$\epsilon_1 \sim \mathcal{N}(0, I) \quad \epsilon_2 \sim \mathcal{N}(0, I) \qquad \sqrt{\alpha_2}\sqrt{1-\alpha_1}\epsilon_1 + \sqrt{(1-\alpha_2)}\epsilon_2 \sim \mathcal{N}(0, (1-\alpha_1\alpha_2)I)$$

$$A\epsilon_1 + B\epsilon_2 \sim \mathcal{N}(0, (A^2+B^2)I) \qquad x_2 \sim \mathcal{N}(\sqrt{\alpha_2\alpha_1}x_0, (1-\alpha_1\alpha_2)I)$$

Forward Process

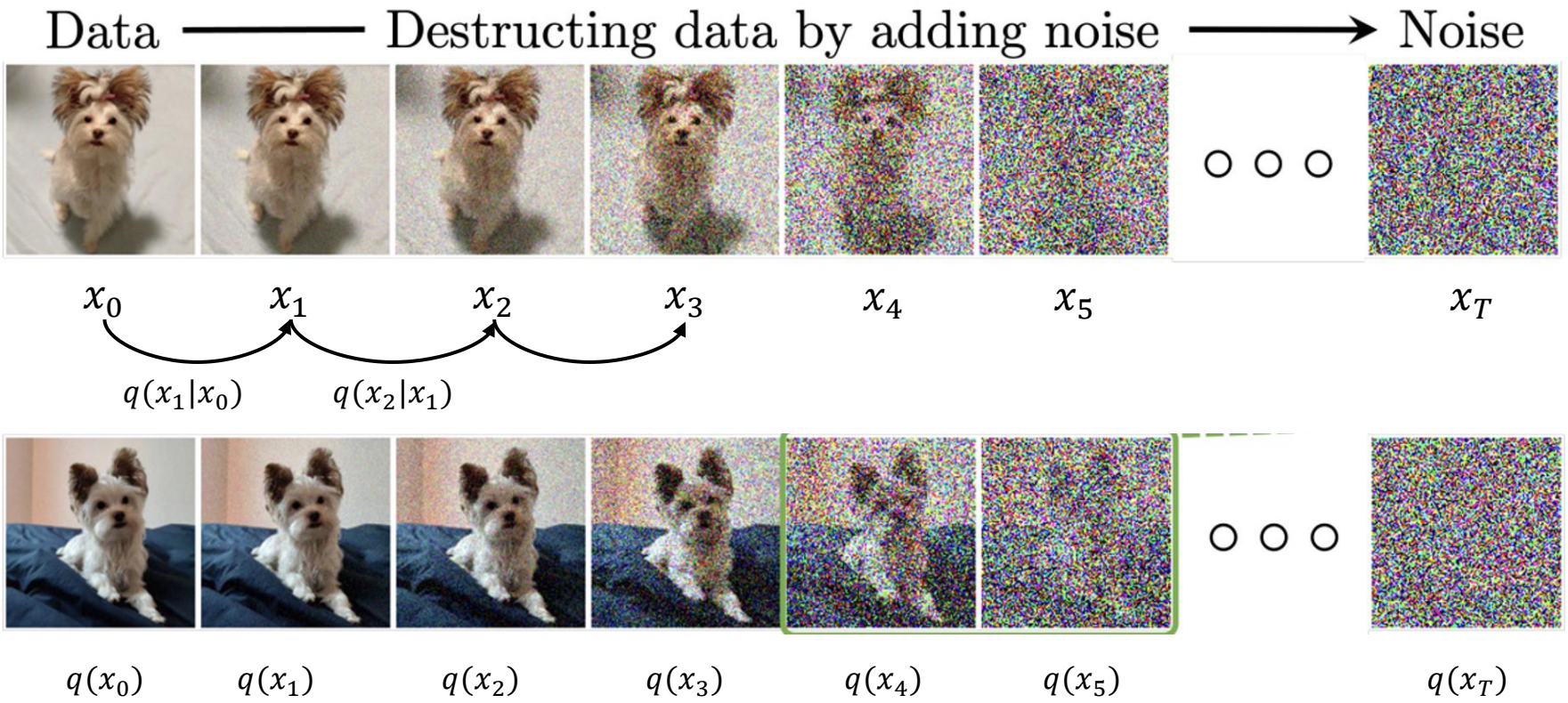


$$x_2 = \sqrt{\alpha_2}\left(\sqrt{\alpha_1}x_0 + \sqrt{(1-\alpha_1)}\epsilon_1\right) + \sqrt{(1-\alpha_2)}\epsilon_2 = \sqrt{\alpha_2\alpha_1}x_0 + \sqrt{\alpha_2}\sqrt{1-\alpha_1}\epsilon_1 + \sqrt{(1-\alpha_2)}\epsilon_2$$

$$\epsilon_1 \sim \mathcal{N}(0, I) \quad \epsilon_2 \sim \mathcal{N}(0, I) \qquad \sqrt{\alpha_2}\sqrt{1-\alpha_1}\epsilon_1 + \sqrt{(1-\alpha_2)}\epsilon_2 \sim \mathcal{N}(0, (1-\alpha_1\alpha_2)I)$$

$$A\epsilon_1 + B\epsilon_2 \sim \mathcal{N}(0, (A^2+B^2)I) \quad x_2 \sim \mathcal{N}(\sqrt{\alpha_2\alpha_1}x_0, (1-\alpha_1\alpha_2)I) \quad x_3 \sim \mathcal{N}(\sqrt{\alpha_3\alpha_2\alpha_1}x_0, (1-\alpha_1\alpha_2\alpha_3)I)$$

Forward Process



$q(x_0)$ $\quad x_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

$q(x_t|x_{t-1})$ $\quad x_t \sim \mathcal{N}(\sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I)$

$x_t \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$

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

Forward Process

$q(x_0)$ $\quad x_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

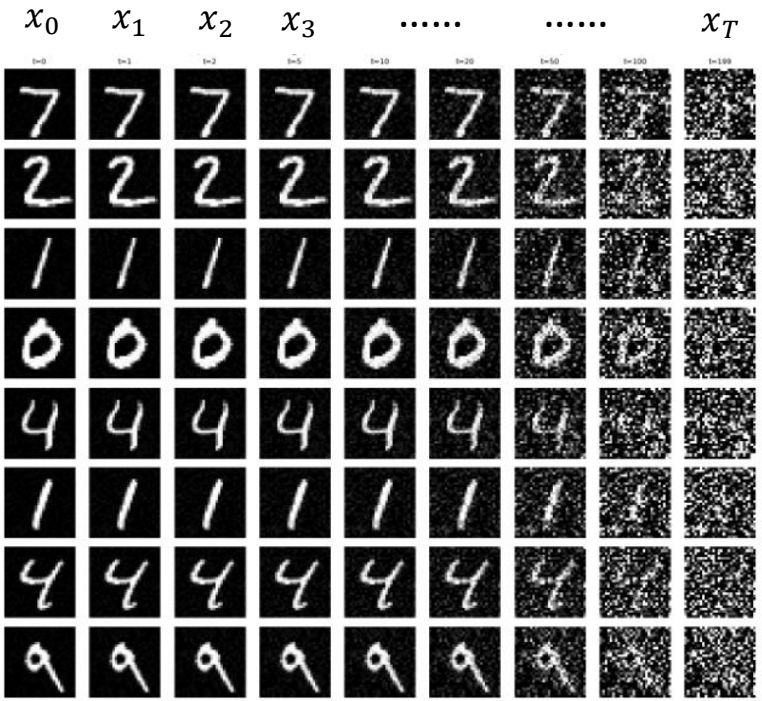
$q(x_t|x_{t-1})$ $\quad x_t \sim \mathcal{N}(\sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I)$

$x_t \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$

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

$\beta_s \in [0,1]$

```python
# Forward process parameters

def linear_beta_schedule(T, beta_start=1e-4, beta_end=2e-2):
    return torch.linspace(beta_start, beta_end, T)

T = 200
betas  = linear_beta_schedule(T).to(device)
alphas = 1.0 - betas
alpha_bar = torch.cumprod(alphas, dim=0)

print('T:', T)
print('beta range:', float(betas.min()), '->', float(betas.max()))
print('alpha_bar[0]:', float(alpha_bar[0]), 'alpha_bar[-1]:', float(alpha_bar[-1]))

# Closed-form sampling from q(x_t | x_0)

def q_sample(x0, t, noise=None):
    # Sample x_t given x_0 at discrete time index t (0..T-1)
    if noise is None:
        noise = torch.randn_like(x0)
    a_bar = alpha_bar[t].view(-1, 1, 1, 1)  # (B,1,1,1)
    return torch.sqrt(a_bar) * x0 + torch.sqrt(1.0 - a_bar) * noise
```



Forward Process

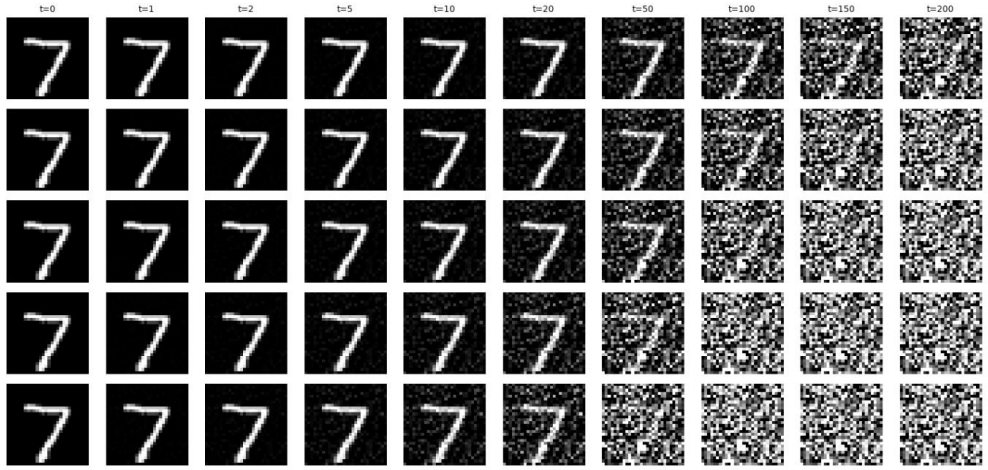
$q(x_0)$ $\quad x_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

$q(x_t|x_{t-1})$ $\quad x_t \sim \mathcal{N}(\sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I)$

$$x_t \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$$

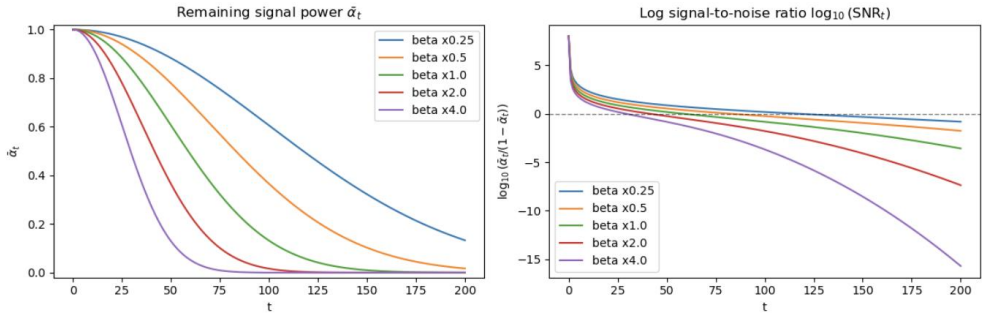
$$\bar{\alpha}_t = \prod_{s=1}^{T}\alpha_s = \prod_{s=1}^{T}(1-\beta_s)$$

$\beta_s \in [0,1]$



$$\mathrm{SNR}_t = \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}$$

Signal Noise Ratio



Forward Process

$q(x_0)$ $\quad x_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

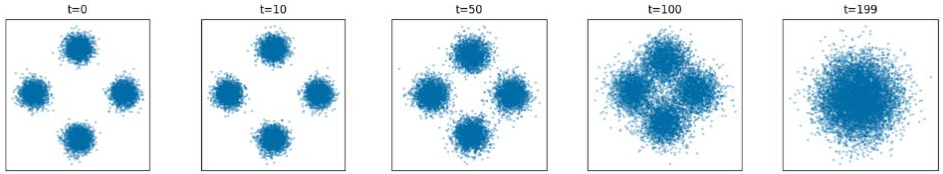
$q(x_t|x_{t-1})$ $\quad x_t \sim \mathcal{N}(\sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I)$

$$x_t \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$$

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

$\beta_s \in [0,1]$

```python
# Forward process parameters

def linear_beta_schedule(T, beta_start=1e-4, beta_end=2e-2):
    return torch.linspace(beta_start, beta_end, T)

T = 200
betas  = linear_beta_schedule(T).to(device)
alphas = 1.0 - betas
alpha_bar = torch.cumprod(alphas, dim=0)

print('T:', T)
print('beta range:', float(betas.min()), '->', float(betas.max()))
print('alpha_bar[0]:', float(alpha_bar[0]), 'alpha_bar[-1]:', float(alpha_bar[-1]))

# Closed-form sampling from q(x_t | x_0)

def q_sample(x0, t, noise=None):
    # Sample x_t given x_0 at discrete time index t (0..T-1)
    if noise is None:
        noise = torch.randn_like(x0)
    a_bar = alpha_bar[t].view(-1, 1, 1, 1)  # (B,1,1,1)
    return torch.sqrt(a_bar) * x0 + torch.sqrt(1.0 - a_bar) * noise
```



Forward Process

$q(x_0)$ $\quad x_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

$q(x_t|x_{t-1})$ $\quad x_t \sim \mathcal{N}(\sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I)$

$$x_t \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$$

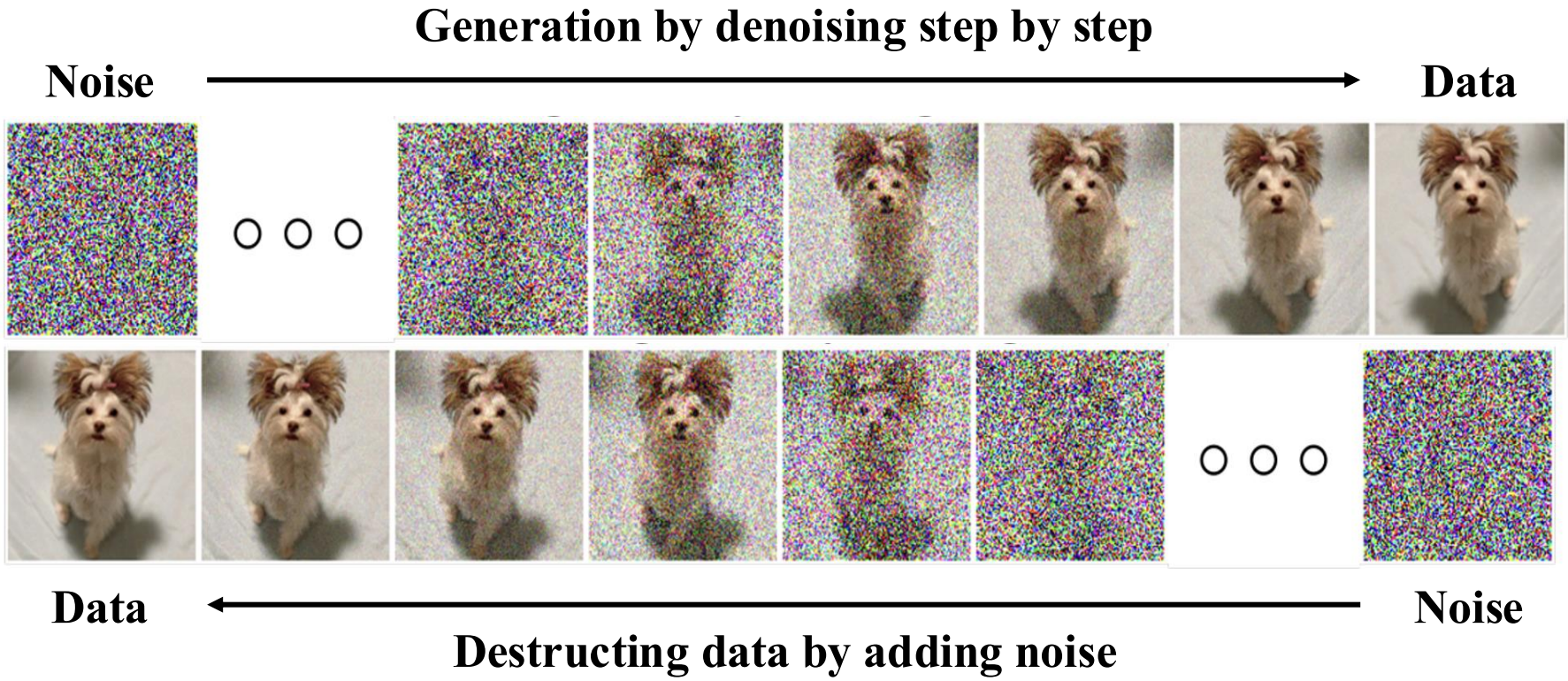
$$\bar{\alpha}_t = \prod_{s=1}^{T}\alpha_s = \prod_{s=1}^{T}(1-\beta_s)$$

$\beta_s \in [0,1]$

```python
# Forward process parameters

def linear_beta_schedule(T, beta_start=1e-4, beta_end=2e-2):
    return torch.linspace(beta_start, beta_end, T)

T = 200
betas  = linear_beta_schedule(T).to(device)
alphas = 1.0 - betas
alpha_bar = torch.cumprod(alphas, dim=0)

print('T:', T)
print('beta range:', float(betas.min()), '->', float(betas.max()))
print('alpha_bar[0]:', float(alpha_bar[0]), 'alpha_bar[-1]:', float(alpha_bar[-1]))

# Closed-form sampling from q(x_t | x_0)

def q_sample(x0, t, noise=None):
    # Sample x_t given x_0 at discrete time index t (0..T-1)
    if noise is None:
        noise = torch.randn_like(x0)
    a_bar = alpha_bar[t].view(-1, 1, 1, 1)  # (B,1,1,1)
    return torch.sqrt(a_bar) * x0 + torch.sqrt(1.0 - a_bar) * noise
```

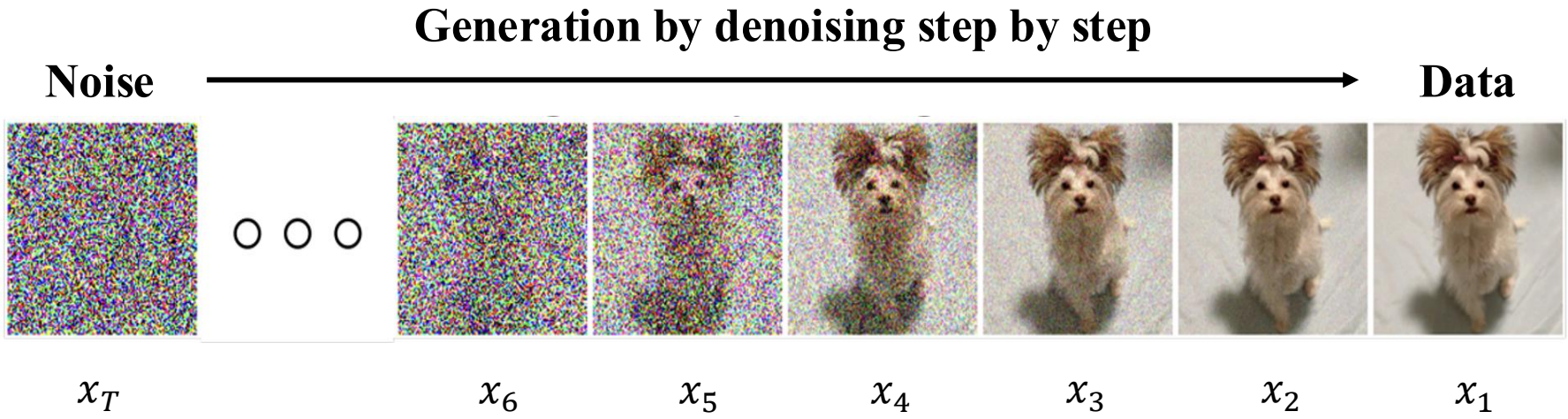
Dog Image

Forward/Backward Process of Diffusion



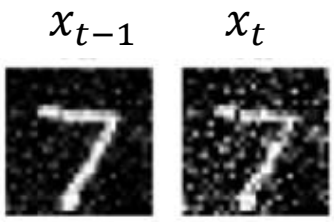
- **Forward Process (Data - Noise):** Gradually add noise to generate intermediate data
- **Backward Process (Noise - Data):** Learn to denoise the intermediate data and recover back

We now can use forward process to obtain data pairs



Predicted Ground-truth

$$\|\hat{x} - x_{t-1}\|_2^2 = \|\mathcal{F}(x_t, t) - x_{t-1}\|_2^2$$



$$\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_{t-1} \sim q(x_{t-1}|x_0), x_t \sim q(x_t|x_{t-1})} \|\mathcal{F}(x_t, t) - x_{t-1}\|_2^2$$

Summary

- **Motivation and Introduction of Diffusion**

- **Forward process of Diffusion**

- **Backward process of Diffusion**
  - **Backward progressive:** $x_t \rightarrow x_{t-1}$
  - **Backward one step:** $x_t \rightarrow x_0$
  - **Backward noise:** $x_t \rightarrow \epsilon$

Backward Optimization

Generation by denoising step by step

Noise → **Data**

x_T x_6 x_5 x_4 x_3 x_2 x_1

Predicted **Ground-truth**

$$\|\hat{x} - x_{t-1}\|_2^2 = \|\mathcal{F}(x_t, t) - x_{t-1}\|_2^2$$

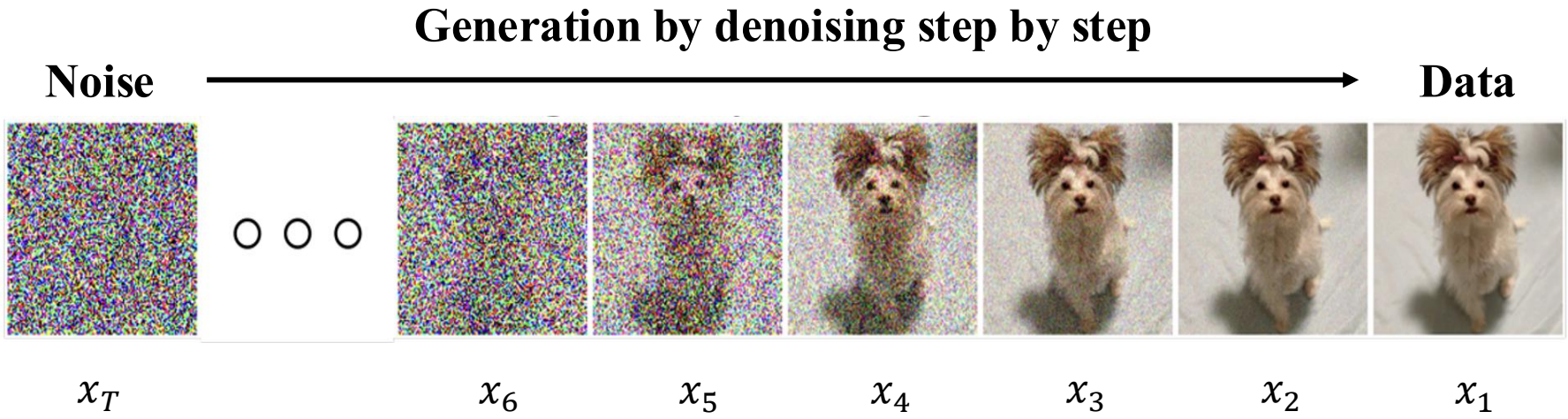
$$\mathbb{E}_{x_0 \sim p_{\text{data}}(x), t \sim U(1,T), x_{t-1} \sim q(x_{t-1}|x_0), x_t \sim q(x_t|x_{t-1})} \|\mathcal{F}(x_t, t) - x_{t-1}\|_2^2$$

$p_{\text{data}}(x)$

Sample

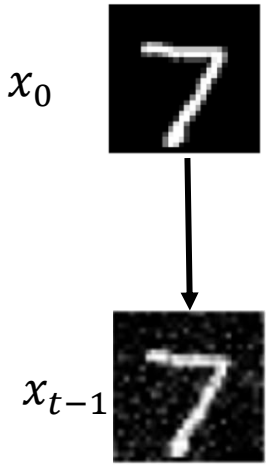
x_0

Backward Optimization



Predicted Ground-truth

$$\|\hat{x} - x_{t-1}\|_2^2 = \|\mathcal{F}(x_t, t) - x_{t-1}\|_2^2$$



$$\mathbb{E}_{x_0 \sim p_{data}(x), \boldsymbol{t \sim U(1,T)}, \boldsymbol{x_{t-1} \sim q(x_{t-1}|x_0)}, x_t \sim q(x_t|x_{t-1})} \|\mathcal{F}(x_t, t) - x_{t-1}\|_2^2$$

Backward Optimization

Generation by denoising step by step

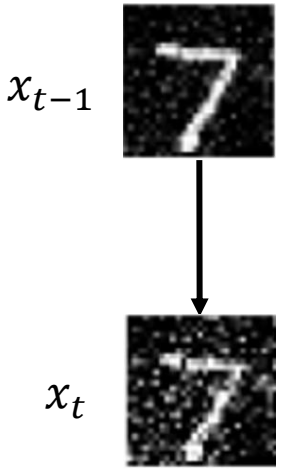
Noise → **Data**

x_T x_6 x_5 x_4 x_3 x_2 x_1

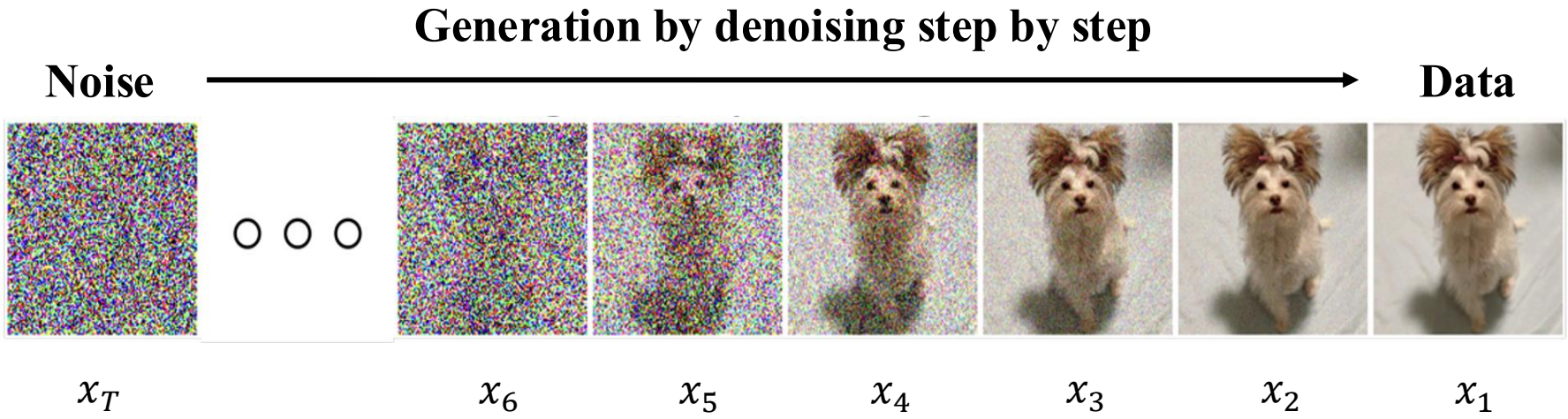
Predicted **Ground-truth**

$$\|\hat{x} - x_{t-1}\|_2^2 = \|\mathcal{F}(x_t, t) - x_{t-1}\|_2^2$$

$$\mathbb{E}_{x_0 \sim p_{data}(x), \boldsymbol{t \sim U(1,T)}, \boldsymbol{x_{t-1} \sim q(x_{t-1}|x_0)}, x_t \sim q(x_t|x_{t-1})} \|\mathcal{F}(x_t, t) - x_{t-1}\|_2^2$$

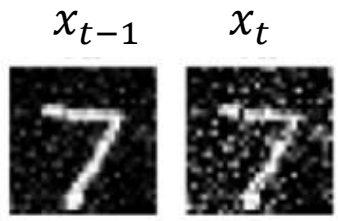


Backward Optimization



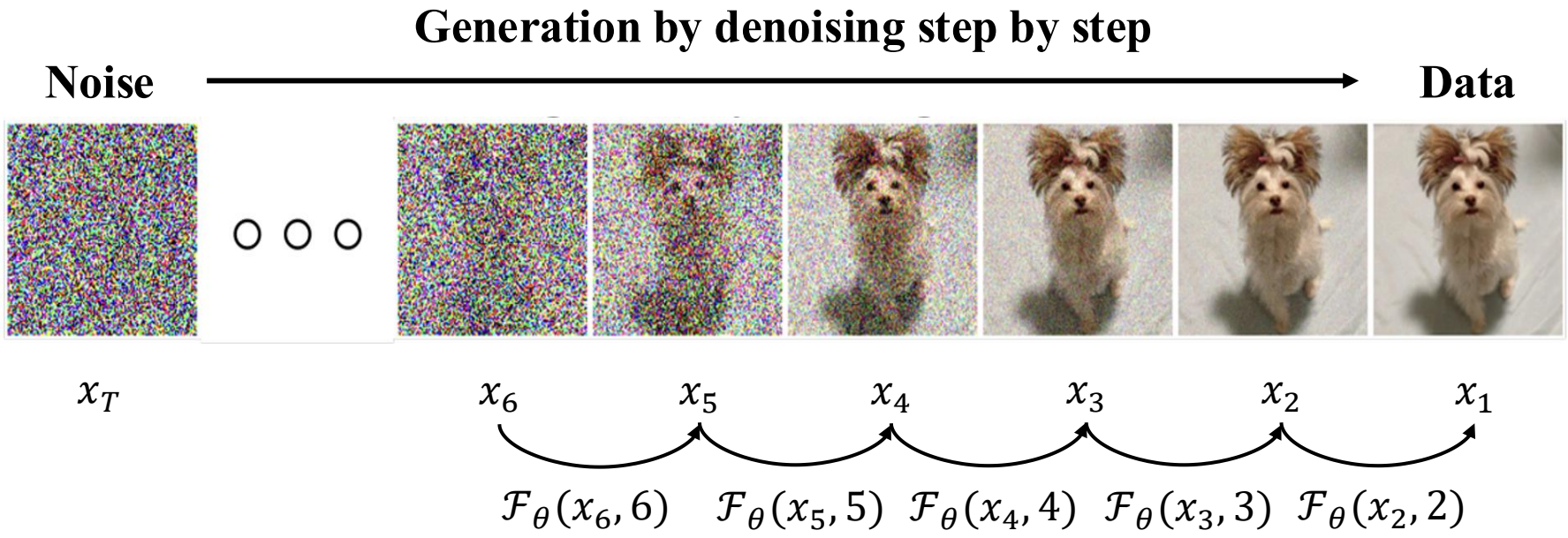
Predicted **Ground-truth**

$$\|\hat{x} - x_{t-1}\|_2^2 = \|\mathcal{F}(x_t, t) - x_{t-1}\|_2^2$$



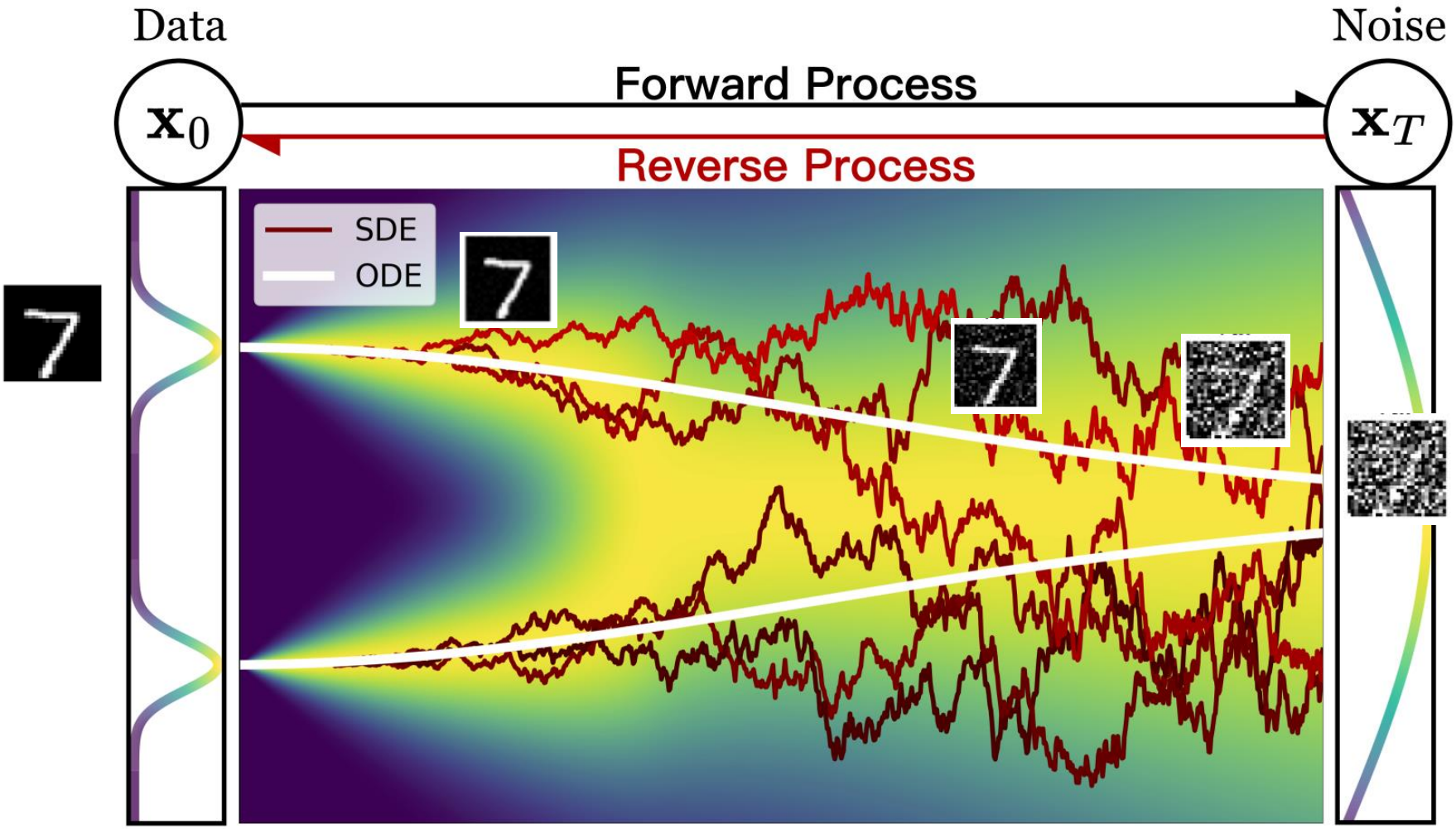
$$\theta^* = argmin\, \mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_{t-1} \sim q(x_{t-1}|x_0), x_t \sim q(x_t|x_{t-1})} \|\mathcal{F}_\theta(x_t, t) - x_{t-1}\|_2^2$$

Generation



Recursively generate back the original image

Generation


Data
Noise
$\mathbf{x}_0$
$\mathbf{x}_T$
Forward Process
Reverse Process
SDE
ODE

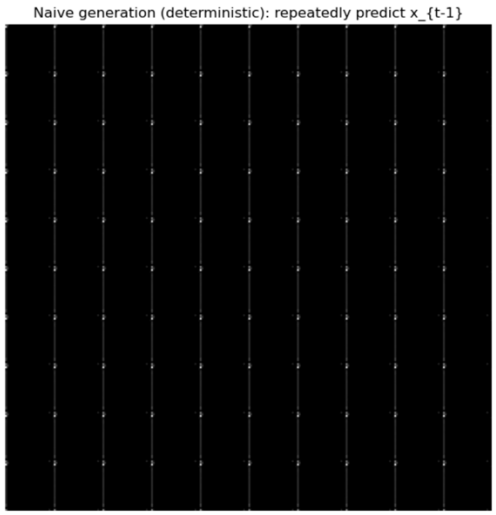
Generation - Problem

Generation by denoising step by step

Noise → Data

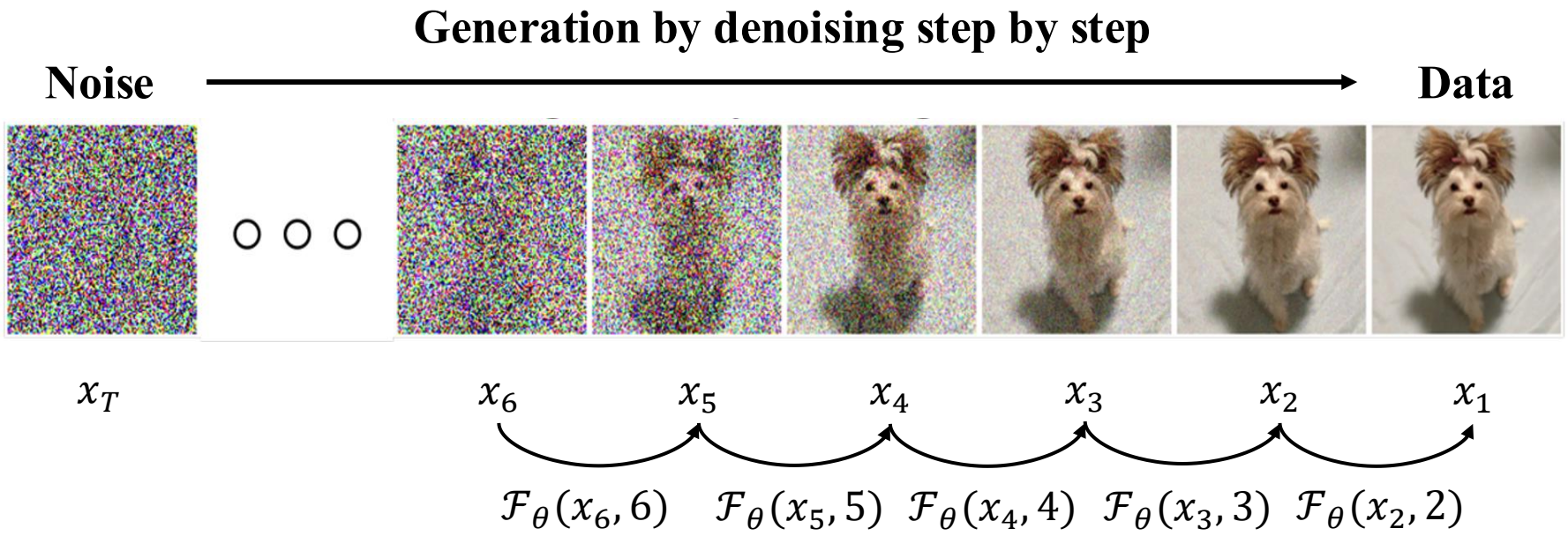
x_T x_6 x_5 x_4 x_3 x_2 x_1

$\mathcal{F}_\theta(x_6, 6)$ $\mathcal{F}_\theta(x_5, 5)$ $\mathcal{F}_\theta(x_4, 4)$ $\mathcal{F}_\theta(x_3, 3)$ $\mathcal{F}_\theta(x_2, 2)$

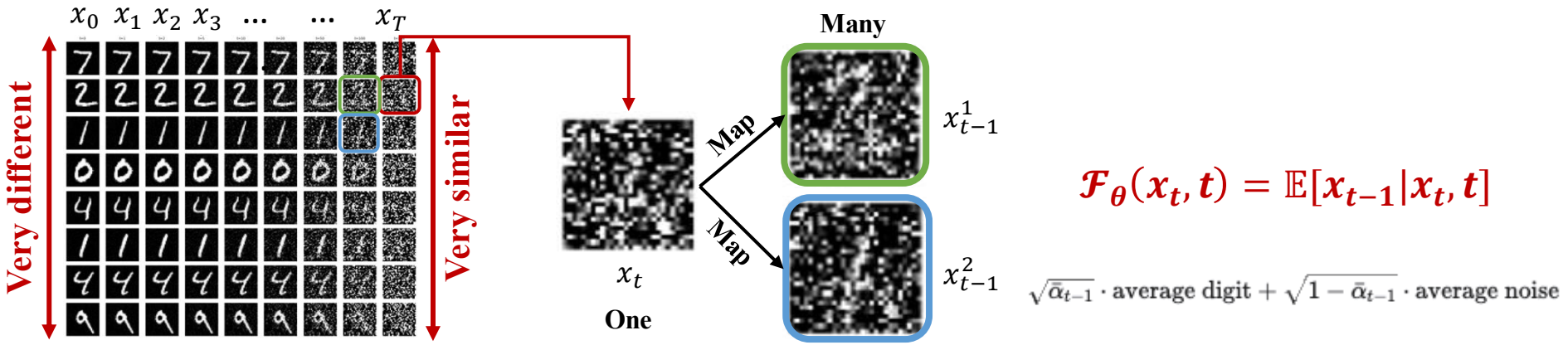


$$\theta^* = argmin\ \mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_{t-1} \sim q(x_{t-1}|x_0), x_t \sim q(x_t|x_{t-1})} \|\mathcal{F}_\theta(x_t, t) - x_{t-1}\|_2^2$$

Generation - Problem

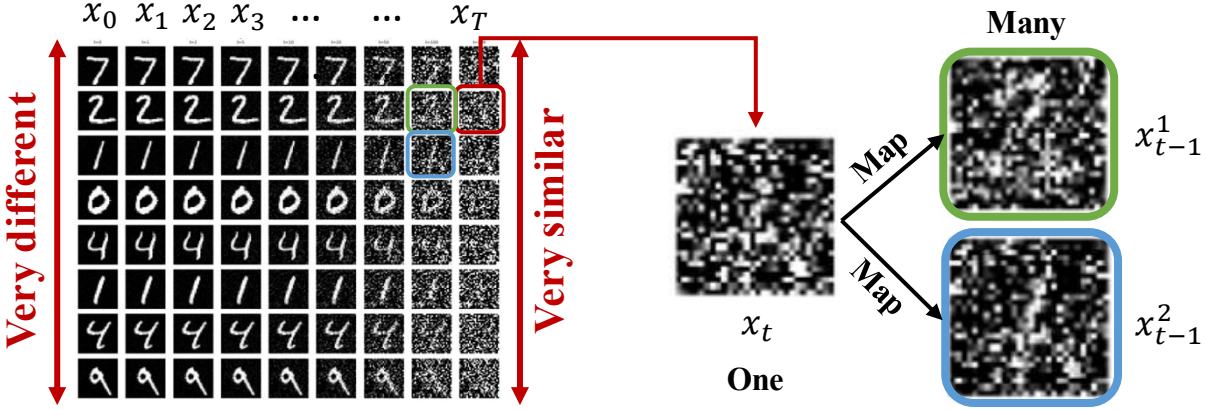


$$\theta^* = argmin\ \mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_{t-1} \sim q(x_{t-1}|x_0), x_t \sim q(x_t|x_{t-1})} \|\boldsymbol{\mathcal{F}_\theta(x_t, t) - x_{t-1}}\|_2^2$$



Generation - Problem

$$\theta^* = argmin\ \mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_{t-1} \sim q(x_{t-1}|x_0), x_t \sim q(x_t|x_{t-1})} \| \boldsymbol{\mathcal{F}_\theta(x_t, t) - x_{t-1}} \|_2^2$$



$$\boldsymbol{\mathcal{F}_\theta(x_t, t) = \mathbb{E}[x_{t-1}|x_t, t]}$$

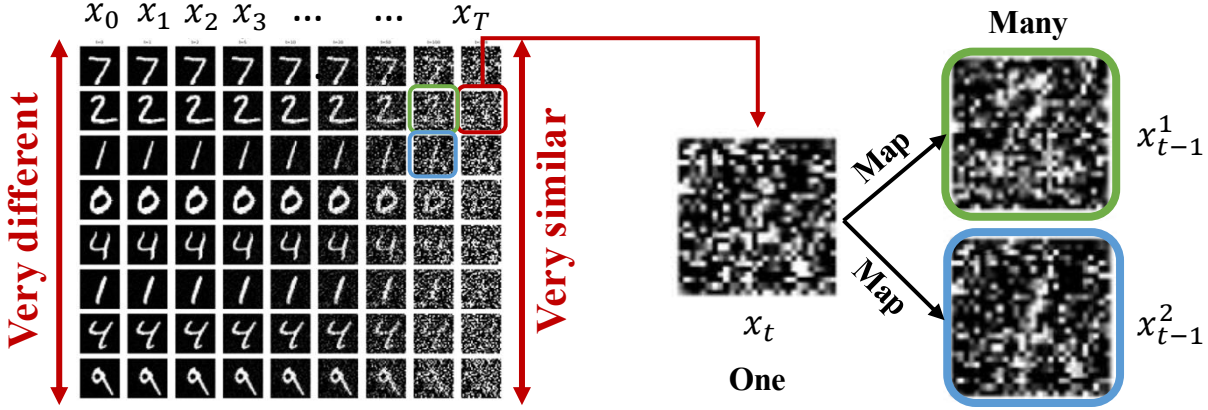
$$\frac{1}{K}\sum_i \epsilon^{(i)} \sim \mathcal{N}\left(0, \frac{1}{K} I\right)$$

$$\frac{1}{K}\sum_{i=1}^{K} x_{t-1}^{(i)} = \sqrt{\bar{\alpha}_{t-1}} x_0 + \sqrt{1-\bar{\alpha}_{t-1}} \frac{1}{K}\sum_{i=1}^{K} \varepsilon^{(i)}.$$

```
early t:    average looks like a somewhat meaningful blurry digit
middle t:   average looks like a faint mixture of digit shapes
late t:     average approaches gray / zero, not a new Gaussian sample
```

Generation - Problem

$$\theta^* = argmin\ \mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_{t-1} \sim q(x_{t-1}|x_0), x_t \sim q(x_t|x_{t-1})} \| \boldsymbol{\mathcal{F}_\theta(x_t, t) - x_{t-1}} \|_2^2$$



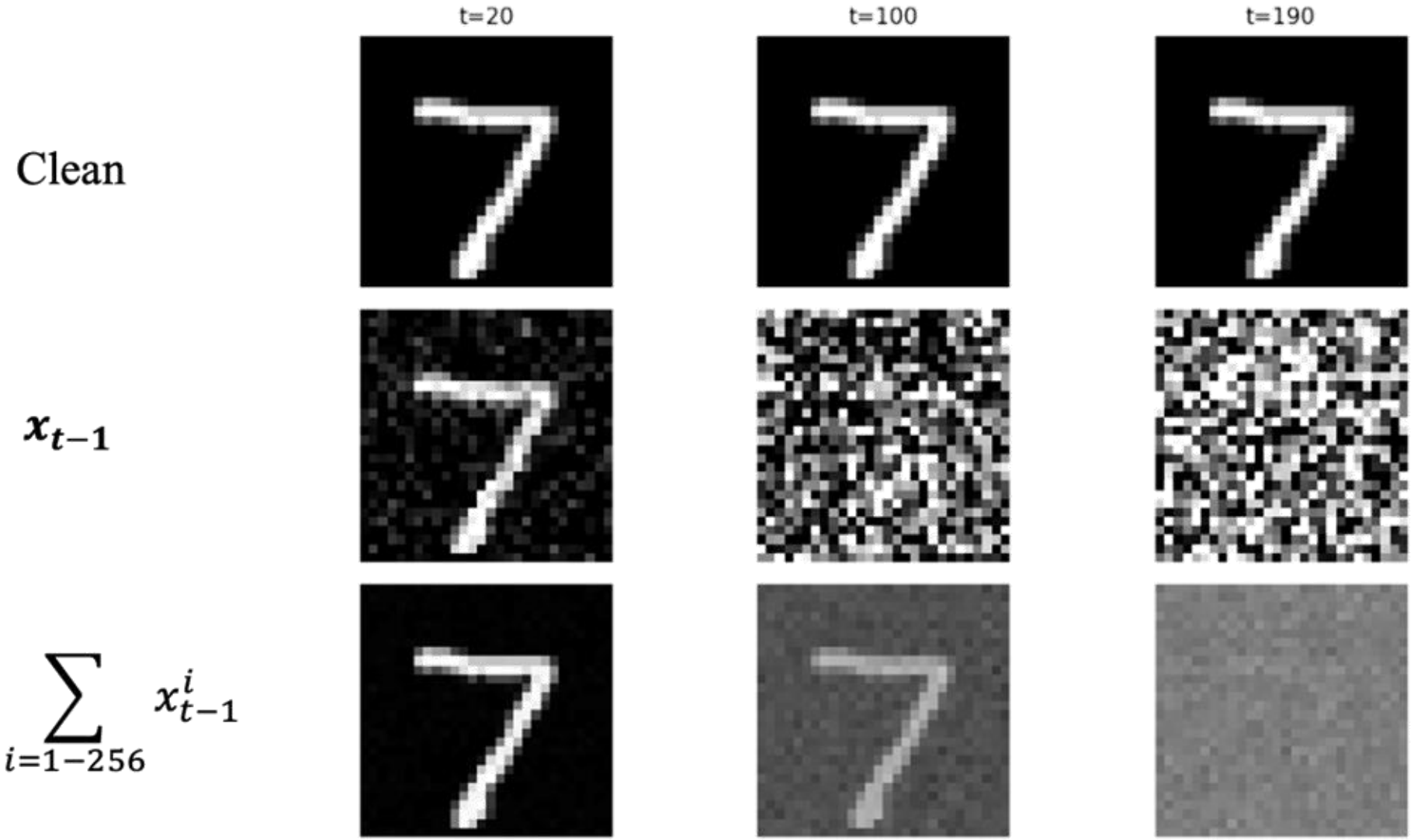
$$\boldsymbol{\mathcal{F}_\theta(x_t, t) = \mathbb{E}[x_{t-1}|x_t, t]}$$

$$\frac{1}{K}\sum_i \epsilon^{(i)} \sim \mathcal{N}\left(0, \frac{1}{K} I\right)$$

$$\frac{1}{K}\sum_{i=1}^{K} x_{t-1}^{(i)} = \sqrt{\bar{\alpha}_{t-1}} x_0 + \sqrt{1-\bar{\alpha}_{t-1}} \frac{1}{K}\sum_{i=1}^{K} \varepsilon^{(i)}.$$

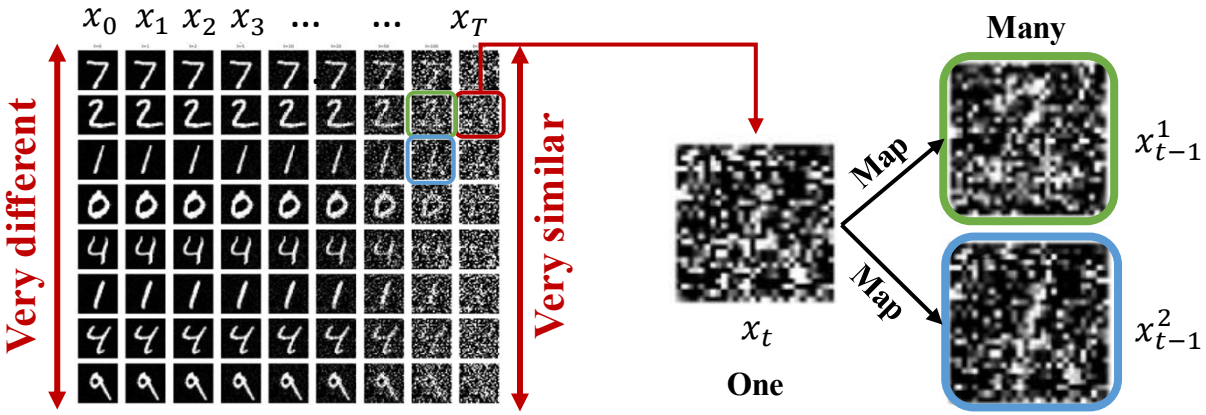
```
early t:    average looks like a somewhat meaningful blurry digit
middle t:   average looks like a faint mixture of digit shapes
late t:     average approaches gray / zero, not a new Gaussian sample
```

Generation - Problem



Generation - Problem

$$\theta^* = argmin\ \mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_{t-1} \sim q(x_{t-1}|x_0), x_t \sim q(x_t|x_{t-1})} \|\mathcal{F}_\theta(x_t, t) - x_{t-1}\|_2^2$$



$$\mathcal{F}_\theta(x_t, t) = \mathbb{E}[x_{t-1}|x_t, t]$$

Train a neural network to predict the nearly grey image

Summary

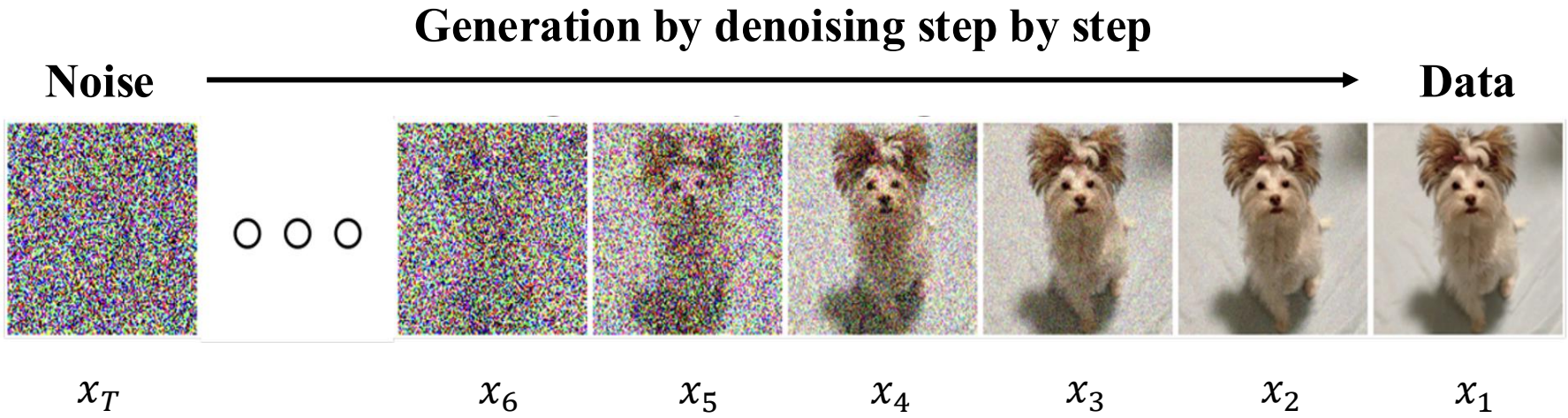
- **Motivation and Introduction of Diffusion**

- **Forward process of Diffusion**

- **Backward process of Diffusion**

  - **Backward progressive:** $x_t \rightarrow x_{t-1}$
  - **Backward one step:** $x_t \rightarrow x_0$
  - **Backward noise:** $x_t \rightarrow \epsilon$

Backward Optimization 2

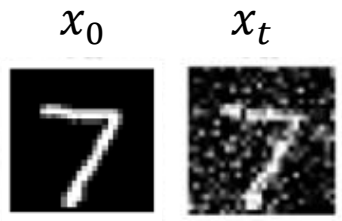


Predicted **Ground-truth**

$$\|\hat{x} - x_{t-1}\|_2^2 = \|\mathcal{F}(x_t, t) - x_{t-1}\|_2^2$$

$$\|\hat{x} - \boldsymbol{x_0}\|_2^2 = \|\mathcal{F}(x_t, t) - \boldsymbol{x_0}\|_2^2$$

No randomness



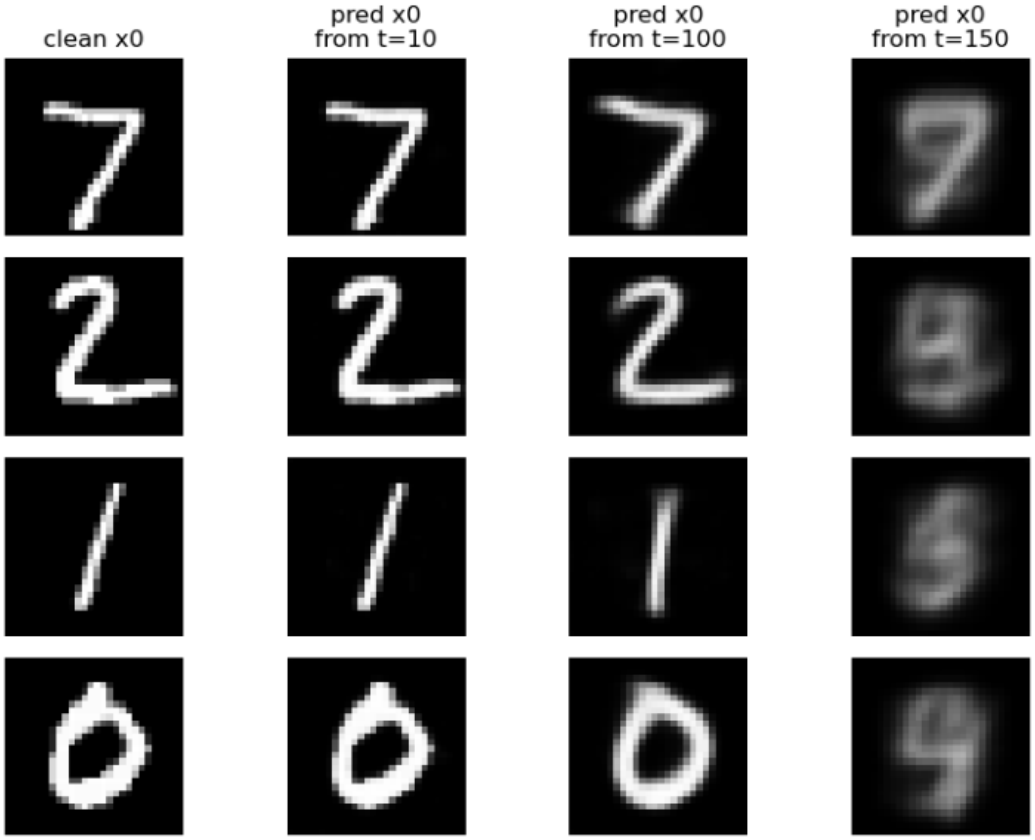
$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_t \sim q(x_t|x_0)} \|\mathcal{F}_\theta(x_t, t) - x_0\|_2^2$$

Given this noisy image and noise level, recover the clean digit.

Backward Optimization 2

$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_t \sim q(x_t|x_0)} \|\mathcal{F}_\theta(x_t, t) - x_0\|_2^2$$

$$x_T \sim \mathcal{N}(0, I) \qquad \mathcal{F}_{\theta^*}(x_T, t) = \bar{x}_0$$

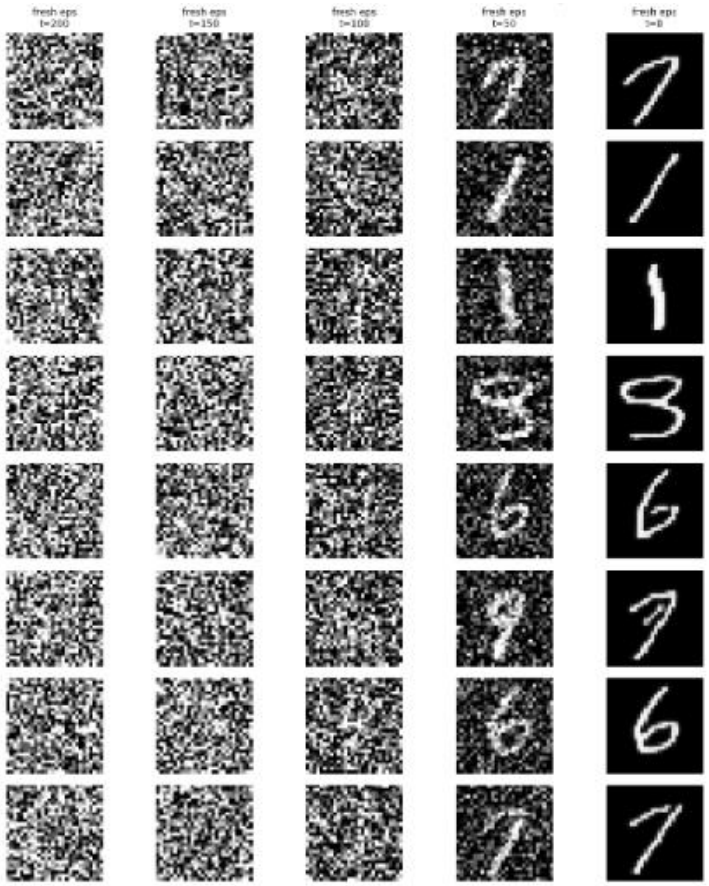


Predict from later time step gives more noisy images

Backward Optimization 2

$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_t \sim q(x_t|x_0)} \|\mathcal{F}_\theta(x_t, t) - x_0\|_2^2$$

$x_T \sim \mathcal{N}(0, I)$ $\mathcal{F}_{\theta^*}(x_T, t) = \bar{x}_0$



$$x_T \xrightarrow{\mathcal{F}_{\theta^*}} \bar{x}_0 \xrightarrow{q(x_{T-1}|\bar{x}_0)} x_{T-1} \xrightarrow{\mathcal{F}_{\theta^*}} \bar{x}_0 \xrightarrow{q(x_{T-1}|\bar{x}_0)} x_{T-2} \cdots$$

$$x_t \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I)$$

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

Inconsistency

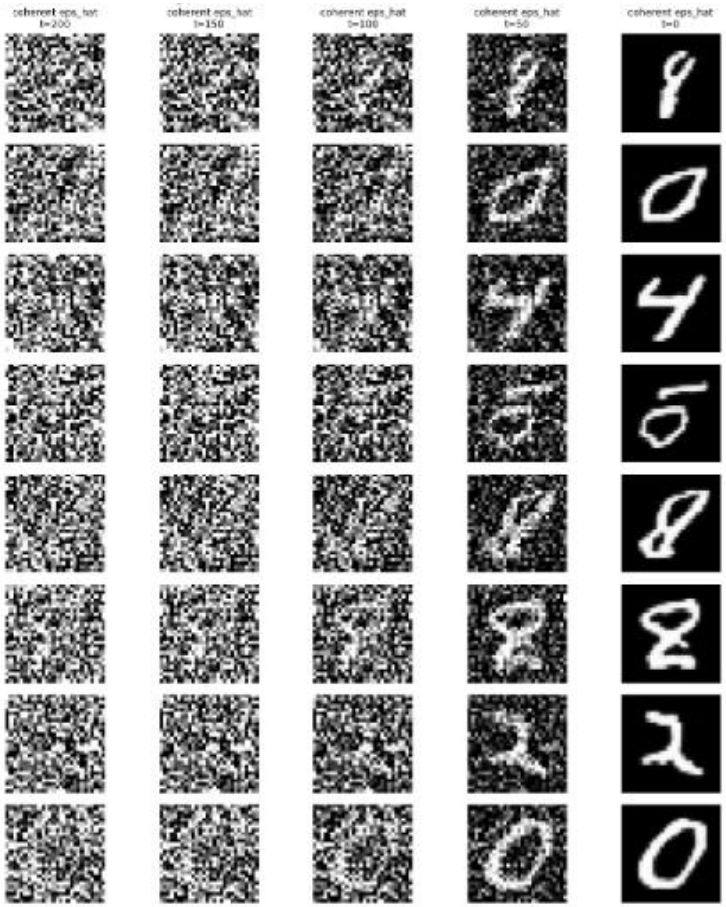
since

resample ϵ

Backward Optimization 2

$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_t \sim q(x_t|x_0)} \|\mathcal{F}_\theta(x_t, t) - x_0\|_2^2$$

$x_T \sim \mathcal{N}(0, I)$ $\mathcal{F}_{\theta^*}(x_T, t) = \bar{x}_0$



$$x_T \rightarrow \bar{x}_0 \rightarrow \epsilon \rightarrow x_{T-1} \rightarrow$$

$$x_t \sim \mathcal{N}(\sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I)$$

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

$$\hat{\epsilon} = \frac{x_t - \sqrt{\bar{\alpha}_t} \hat{x}_0}{\sqrt{1 - \bar{\alpha}_t}}$$

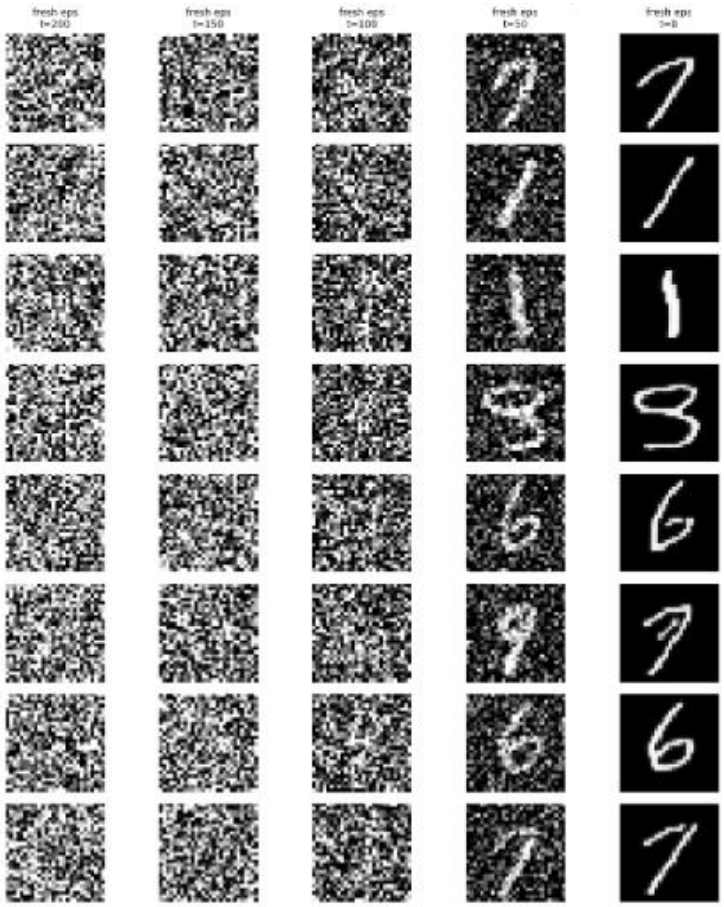
$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \hat{x}_0 + \sqrt{1 - \bar{\alpha}_{t-1}} \hat{\epsilon}$$

Consistency

since

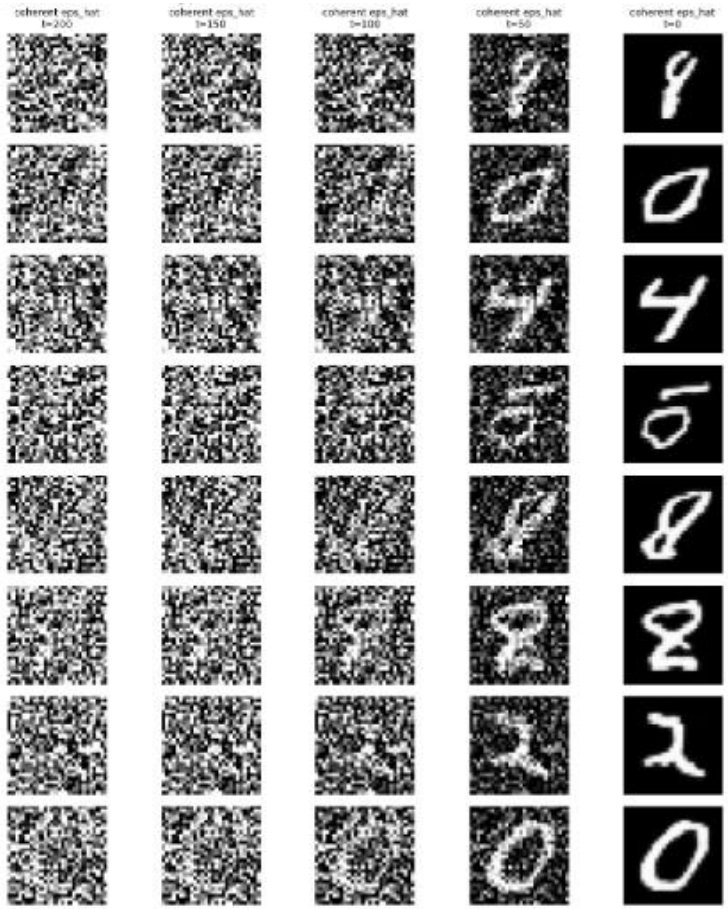
Solve ϵ

Backward Optimization 2



$$x_T \rightarrow \bar{x}_0 \rightarrow x_{T-1} \rightarrow \bar{x}_0 \rightarrow x_{T-2} \cdots$$

Inconsistency



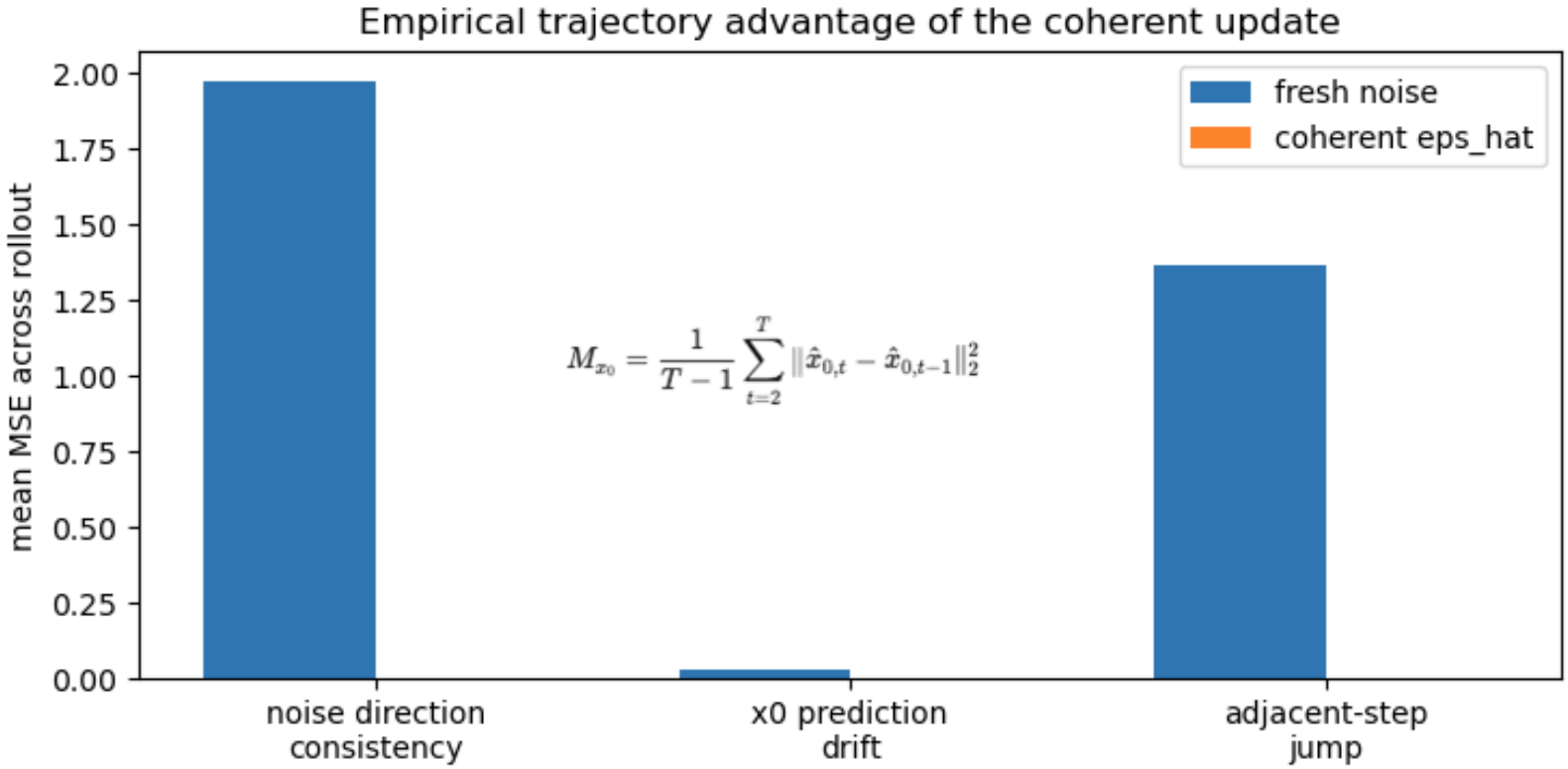
$$x_T \rightarrow \bar{x}_0 \rightarrow \epsilon \rightarrow x_{T-1} \rightarrow$$

Consistency

Backward Optimization 2

$$M_\varepsilon = \frac{1}{T-1}\sum_{t=2}^{T} \|\hat{\varepsilon}_{t-1}^{\text{implied}} - \hat{\varepsilon}_t\|_2^2$$

$$M_{\Delta x} = \frac{1}{T}\sum_{t=1}^{T} \|x_{t-1} - x_t\|_2^2$$



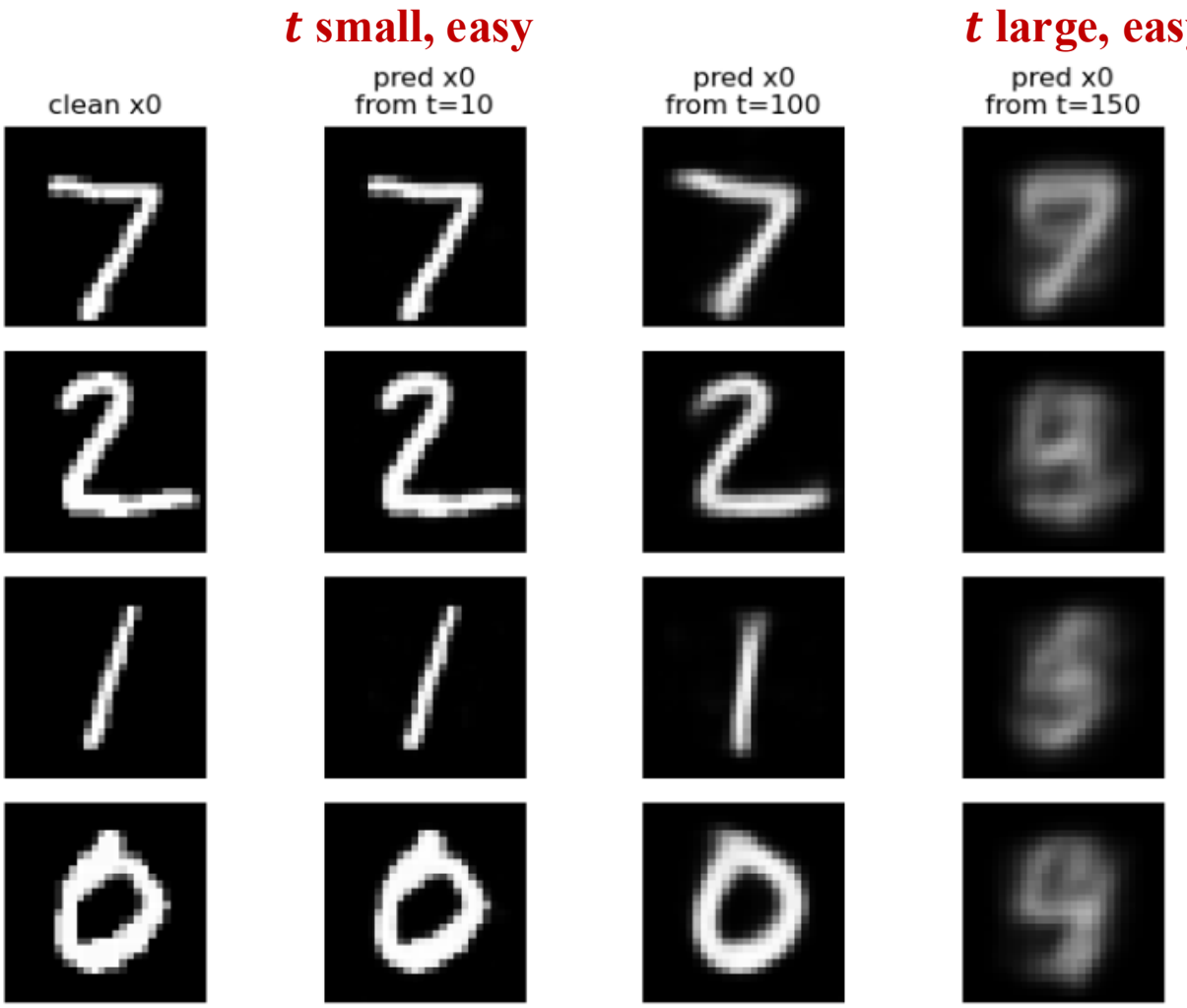
$x_T \to \bar{x}_0 \to x_{T-1} \to \bar{x}_0 \to x_{T-2} \ldots$

$x_T \to \bar{x}_0 \to \epsilon \to x_{T-1} \to$

Both are good

Backward Optimization 2 - Problem

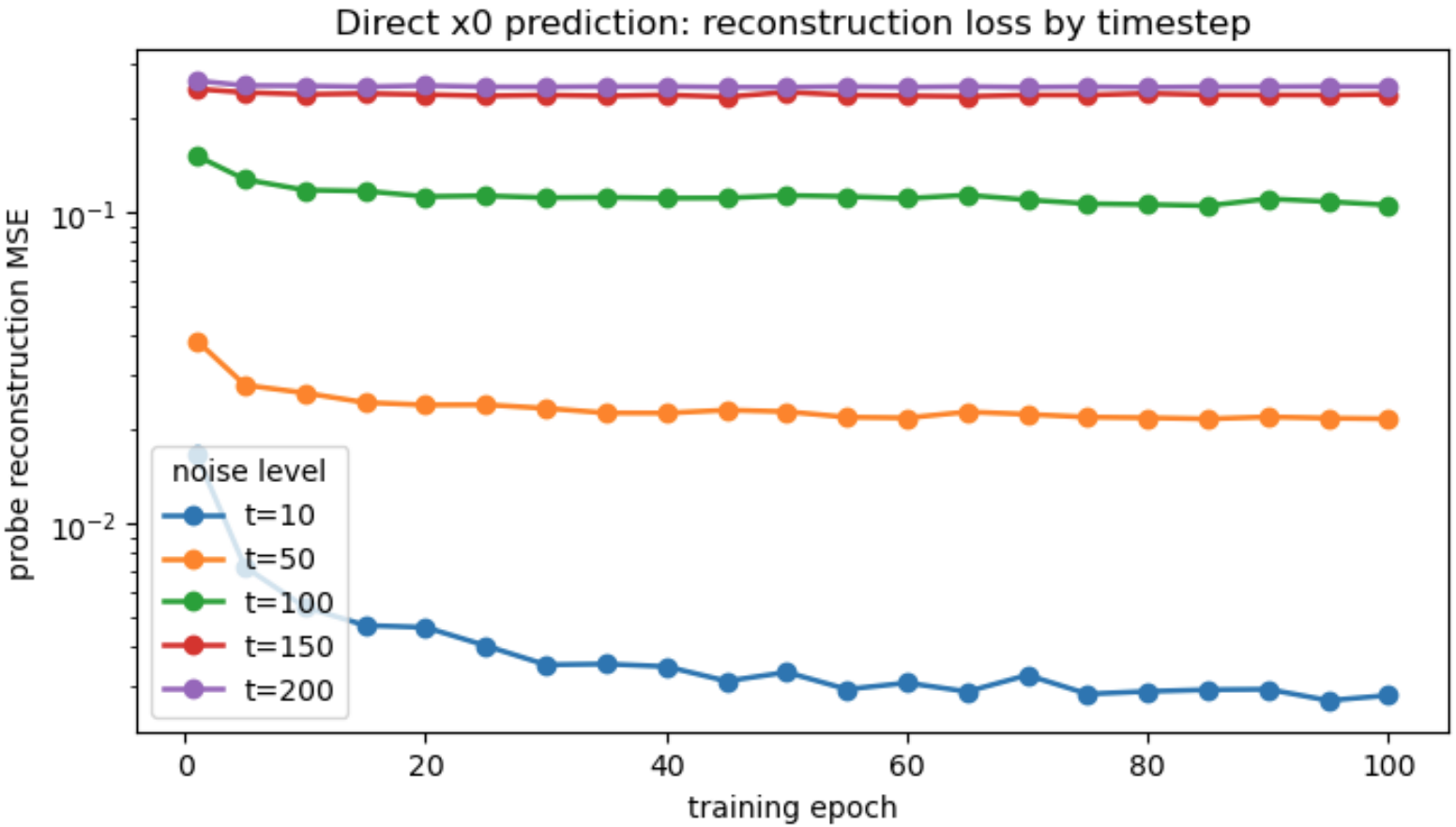
$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_t \sim q(x_t|x_0)} \| \mathcal{F}_\theta(x_t, t) - x_0 \|_2^2$$



Backward Optimization 2 - Problem

$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_t \sim q(x_t|x_0)} \|\mathcal{F}_\theta(x_t, t) - x_0\|_2^2$$

***t* small, easy** ***t* large, easy**



Summary

- **Motivation and Introduction of Diffusion**

- **Forward process of Diffusion**

- **Backward process of Diffusion**
  - **Backward progressive:** $x_t \rightarrow x_{t-1}$
  - **Backward one step:** $x_t \rightarrow x_0$
  - **Backward noise:** $x_t \rightarrow \epsilon$

Backward Optimization 3

$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_t \sim q(x_t|x_0)} \|\mathcal{F}_\theta(x_t, t) - x_0\|_2^2$$

$$\mathcal{F}_{\theta^*}(x_T, t) = \bar{x}_0 \longrightarrow \hat{\epsilon} = \frac{x_t - \sqrt{\bar{\alpha}_t}\,\hat{x}_0}{\sqrt{1 - \bar{\alpha}_t}}.$$

Can we instead directly predict noise?

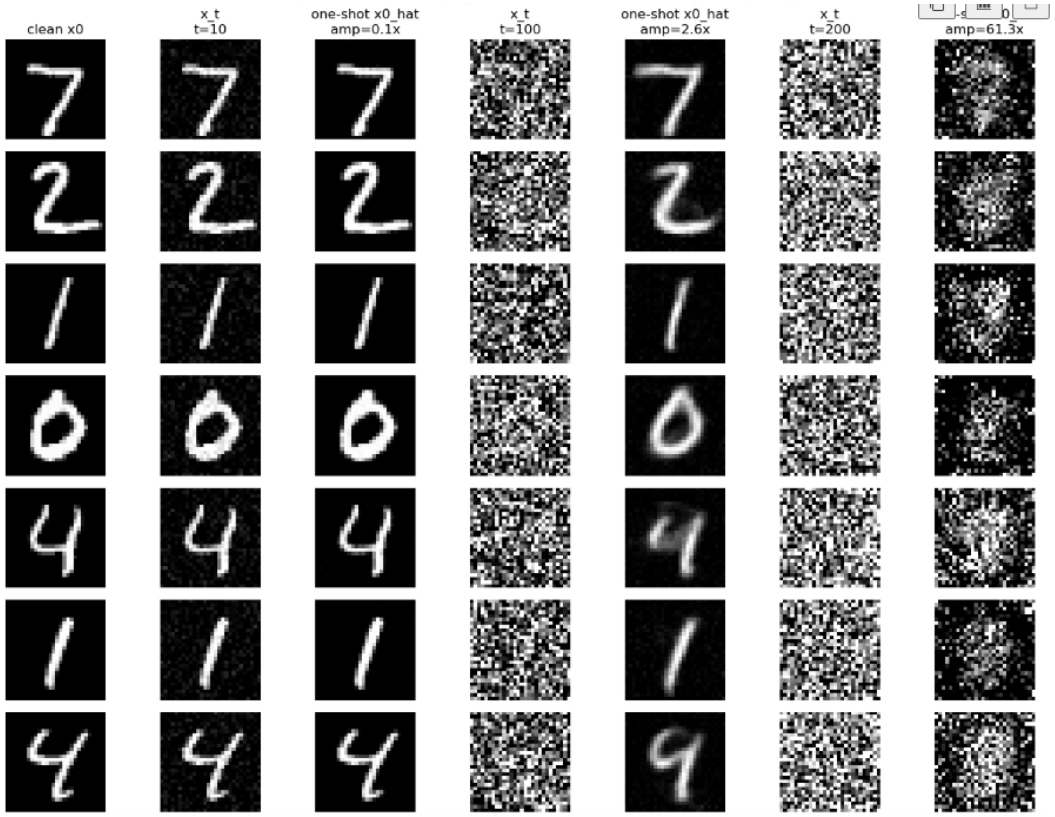
$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), \epsilon \sim \mathcal{N}(0,I)} \|\epsilon_\theta(x_t, t) - \epsilon\|_2^2$$

$$\hat{x}_0 = \frac{x_t - \sqrt{1 - \bar{\alpha}_t}\,\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}}.$$

Backward Optimization 3

$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), \epsilon \sim \mathcal{N}(0,I)} \|\epsilon_\theta(x_t, t) - \epsilon\|_2^2$$

$$x_T \sim \mathcal{N}(0, I) \qquad \hat{\epsilon} = \epsilon_{\theta^*}(x_t, t) \longrightarrow \hat{x}_0 = \frac{x_t - \sqrt{1 - \bar{\alpha}_t}\,\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}}.$$

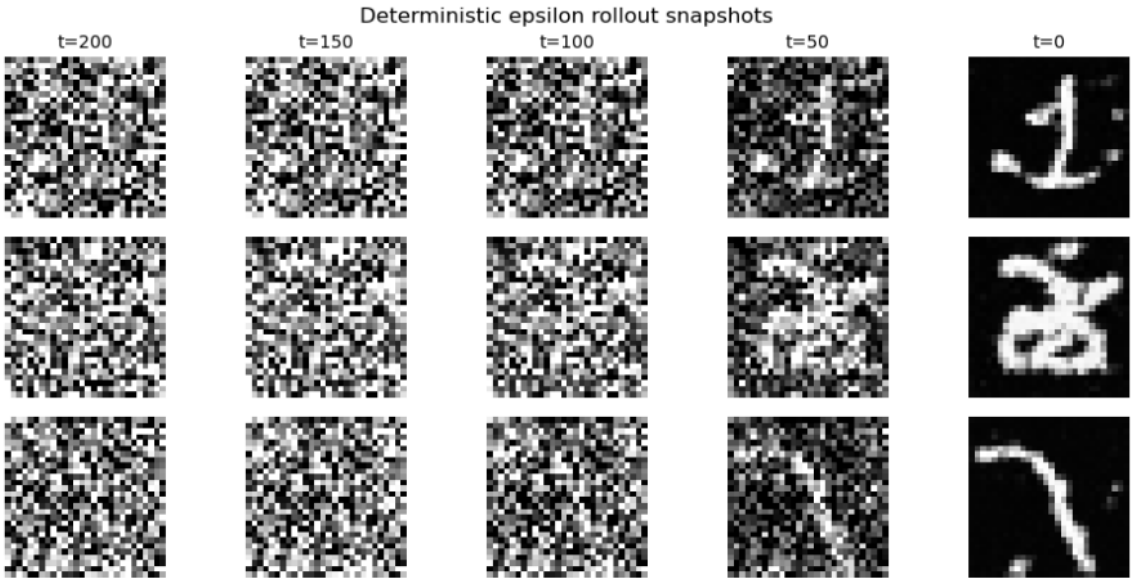


Backward Optimization 3

$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), \epsilon \sim \mathcal{N}(0,I)} \|\epsilon_\theta(x_t, t) - \epsilon\|_2^2$$

$$x_T \sim \mathcal{N}(0, I) \qquad \hat{\epsilon} = \epsilon_{\theta^*}(x_t, t)$$

$$\hat{x}_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\,\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}}. \qquad x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\epsilon}$$



Backward Optimization 3

$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), \epsilon \sim \mathcal{N}(0,I)} \|\epsilon_\theta(x_t, t) - \epsilon\|_2^2$$

$$x_T \sim \mathcal{N}(0, I) \qquad \hat{\epsilon} = \epsilon_{\theta^*}(x_t, t)$$

$$\tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t.$$

$$\hat{x}_0 = \frac{x_t - \sqrt{1 - \bar{\alpha}_t}\,\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}}. \qquad q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(\tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I).$$

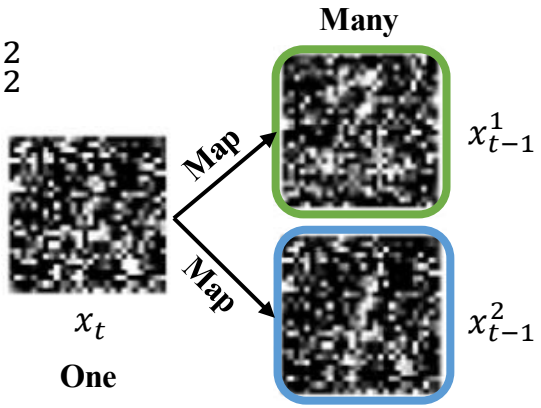
$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t$$

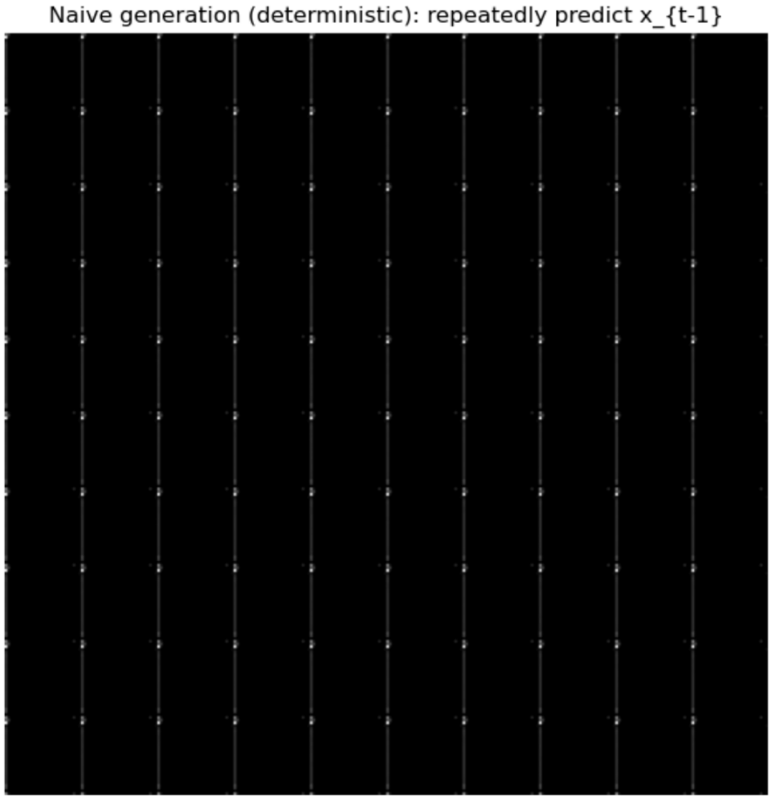
$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \varepsilon_\theta(x_t, t) \right)$$

$$x_{t-1} = \mu_\theta(x_t, t) + \sqrt{\tilde{\beta}_t}\, z, \qquad z \sim \mathcal{N}(0, I)$$

Summary of three backward optimization - 1

$$\mathbb{E}_{\boldsymbol{x_0 \sim p_{data}}(\boldsymbol{x}), t \sim U(1,T), x_{t-1} \sim q(x_{t-1}|x_0), x_t \sim q(x_t|x_{t-1})} \|\mathcal{F}(x_t, t) - x_{t-1}\|_2^2$$





Forward Process

cat ⟶ c#a$t%

Backward Process

cat ⟵ c#a$t%

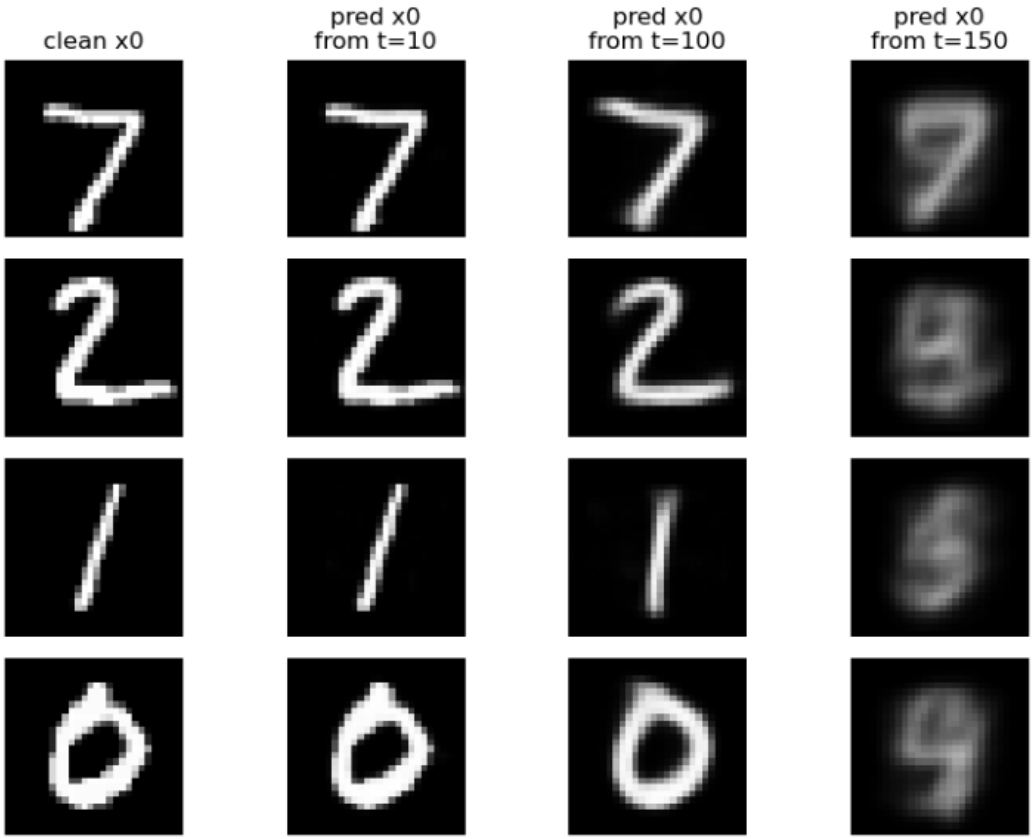
What was the exact previous corrupted string?
Was it "c#a$t", "ca$t%", "c#at%", ... ?

$$x_t \rightarrow x_{t-1}$$

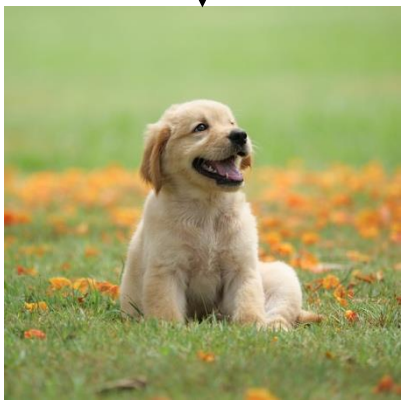
Summary of three backward optimization - 2

$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_t \sim q(x_t|x_0)} \|\mathcal{F}_\theta(x_t, t) - x_0\|_2^2$$

$$x_T \sim \mathcal{N}(0, I) \qquad \mathcal{F}_{\theta^*}(x_T, t) = \bar{x}_0$$



One-shot Generation



Summary of three backward optimization - 2

$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_t \sim q(x_t|x_0)} \|\mathcal{F}_\theta(x_t, t) - x_0\|_2^2$$

$$x_T \sim \mathcal{N}(0, I) \qquad \mathcal{F}_{\theta^*}(x_T, t) = \bar{x}_0$$

Inconsistency

$$x_T \xrightarrow{\mathcal{F}_{\theta^*}} \bar{x}_0 \xrightarrow{} x_{T-1} \xrightarrow{\mathcal{F}_{\theta^*}} \bar{x}_0 \xrightarrow{} x_{T-2} \cdots$$

$$q(x_{T-1}|\bar{x}_0) \qquad q(x_{T-1}|\bar{x}_0)$$

$$x_t \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$$

Consistency

$$x_T \rightarrow \bar{x}_0 \rightarrow \epsilon \rightarrow x_{T-1} \rightarrow$$

$$x_t \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$$

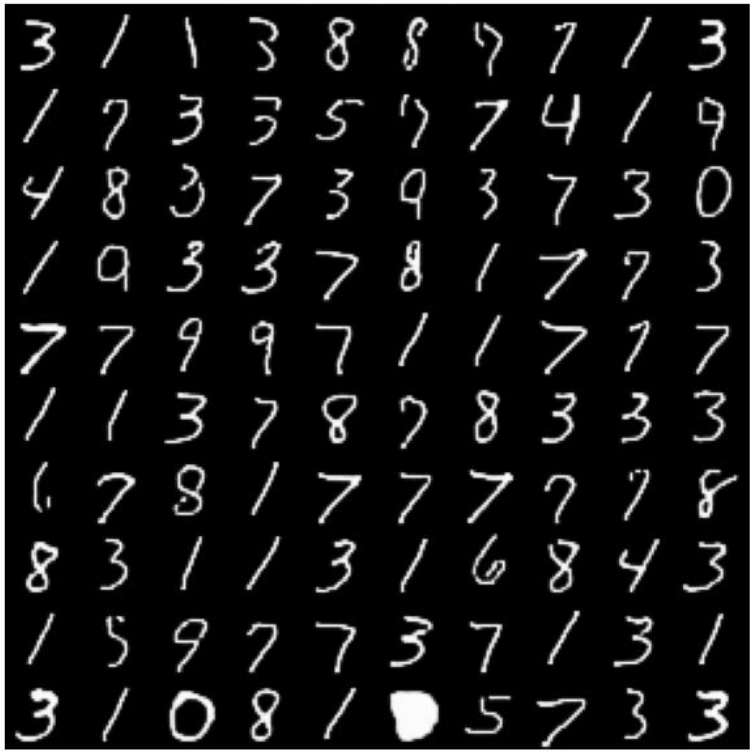
$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

$$\hat{\epsilon} = \frac{x_t - \sqrt{\bar{\alpha}_t}\hat{x}_0}{\sqrt{1-\bar{\alpha}_t}} \qquad x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\epsilon}$$

Summary of three backward optimization - 2

$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_t \sim q(x_t|x_0)} \|\mathcal{F}_\theta(x_t, t) - x_0\|_2^2$$

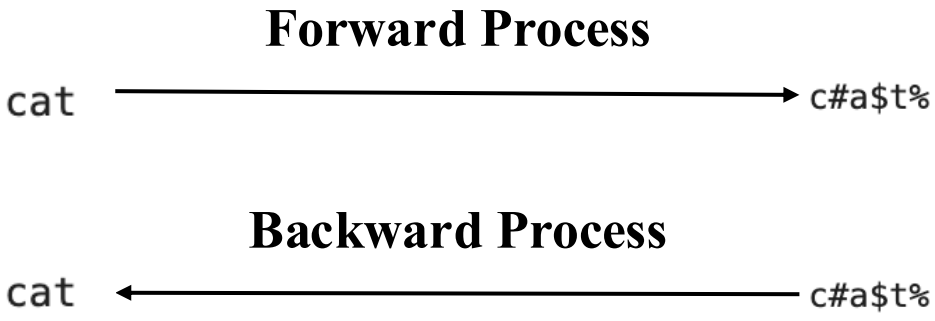
$$x_T \sim \mathcal{N}(0, I) \qquad \mathcal{F}_{\theta^*}(x_T, t) = \bar{x}_0$$



Summary of three backward optimization - 2

$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), x_t \sim q(x_t|x_0)} \|\mathcal{F}_\theta(x_t, t) - x_0\|_2^2$$

$$x_T \sim \mathcal{N}(0, I) \qquad \mathcal{F}_{\theta^*}(x_T, t) = \bar{x}_0$$



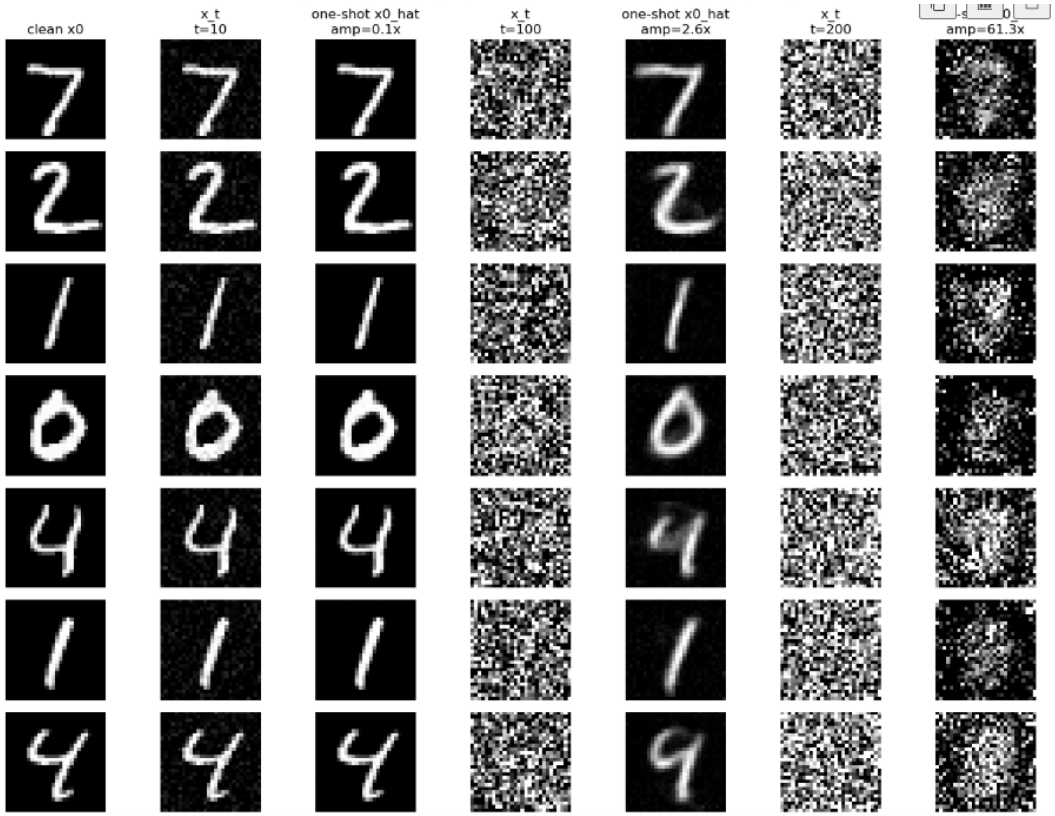
What was the clean string?

$x_t \rightarrow x_0$

Summary of three backward optimization - 3

$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), \epsilon \sim \mathcal{N}(0,I)} \|\epsilon_\theta(x_t, t) - \epsilon\|_2^2$$

$$x_T \sim \mathcal{N}(0, I) \qquad \hat{\epsilon} = \epsilon_{\theta^*}(x_t, t) \longrightarrow \hat{x}_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\,\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}}.$$



Summary of three backward optimization - 3

$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), \epsilon \sim \mathcal{N}(0,I)} \|\epsilon_\theta(x_t, t) - \epsilon\|_2^2$$

$$x_T \sim \mathcal{N}(0, I) \qquad \hat{\epsilon} = \epsilon_{\theta^*}(x_t, t) \longrightarrow \hat{x}_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\,\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}}.$$

$$x_T \sim \mathcal{N}(0, I) \qquad \hat{\epsilon} = \epsilon_{\theta^*}(x_t, t)$$

$$\hat{x}_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\,\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}}.$$

Introduce some noise

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\epsilon}$$

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(\tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I).$$

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

$$x_{t-1} = \mu_\theta(x_t, t) + \sqrt{\tilde{\beta}_t}\, z, \qquad z \sim \mathcal{N}(0, I)$$

Summary of three backward optimization - 3

$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), \epsilon \sim \mathcal{N}(0,I)} \|\epsilon_\theta(x_t, t) - \epsilon\|_2^2$$

$$x_T \sim \mathcal{N}(0, I) \qquad \hat{\epsilon} = \epsilon_{\theta^*}(x_t, t) \longrightarrow \hat{x}_0 = \frac{x_t - \sqrt{1 - \bar{\alpha}_t}\,\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}}.$$

Summary of three backward optimization - 3

$$\theta^* = argmin\mathbb{E}_{x_0 \sim p_{data}(x), t \sim U(1,T), \epsilon \sim \mathcal{N}(0,I)} \|\epsilon_\theta(x_t, t) - \epsilon\|_2^2$$

$$x_T \sim \mathcal{N}(0, I) \qquad \hat{\epsilon} = \epsilon_{\theta^*}(x_t, t)$$

Forward Process

cat ⟶ c#a$t%

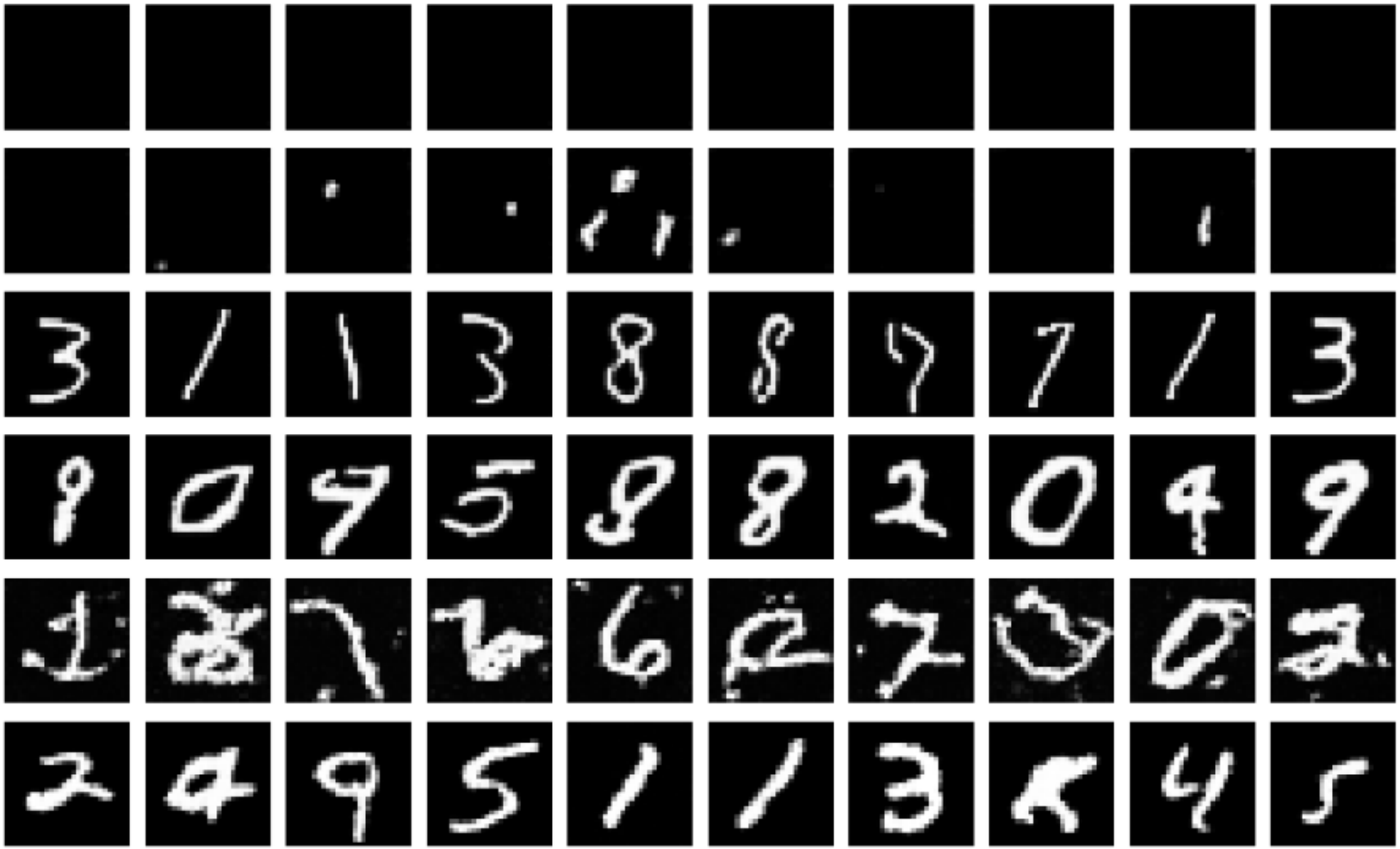
Backward Process

cat ⟵ c#a$t%

Which symbols look like noise?
Answer: #, $, %

$x_t \to \sigma$

Summary



Summary

method	NN-MSE	divMSE	div/real	meanGap	stdGap	cover	entropy
A1 xprev det	0.1223	0.0000	0.000	0.3298	0.5671	1	-0.000
A1 xprev +noise	0.1557	0.0874	0.171	0.3088	0.3843	2	0.141
A2 x0 fresh	0.1299	0.2566	0.501	0.1117	0.1677	4	0.530
A2 x0 coherent	0.2295	0.5247	1.025	0.0571	0.0554	6	0.759
A3 eps det	0.3334	0.6679	1.304	0.1542	0.0730	5	0.676
A3 eps DDPM	0.1930	0.5187	1.013	0.0194	0.0262	7	0.819

Nearest-real MSE measures whether generated images lie close to the real MNIST manifold:

$$M_{\mathrm{NN}} = \frac{1}{N}\sum_{i=1}^{N} \min_{1\le j\le M} \frac{1}{D}\left\|\hat{x}_i - x_j^{\mathrm{real}}\right\|_2^2 .$$

Summary

method	NN-MSE	divMSE	div/real	meanGap	stdGap	cover	entropy
A1 xprev det	0.1223	0.0000	0.000	0.3298	0.5671	1	-0.000
A1 xprev +noise	0.1557	0.0874	0.171	0.3088	0.3843	2	0.141
A2 x0 fresh	0.1299	0.2566	0.501	0.1117	0.1677	4	0.530
A2 x0 coherent	0.2295	0.5247	1.025	0.0571	0.0554	6	0.759
A3 eps det	0.3334	0.6679	1.304	0.1542	0.0730	5	0.676
A3 eps DDPM	0.1930	0.5187	1.013	0.0194	0.0262	7	0.819

Diversity MSE measures spread among generated samples:

$$M_{\text{div}} = \frac{2}{N(N-1)} \sum_{1 \le i < k \le N} \frac{1}{D} \|\hat{x}_i - \hat{x}_k\|_2^2 .$$

Too low suggests collapse; too high suggests noisy/unstructured samples. The notebook also reports

$$R_{\text{div}} = \frac{M_{\text{div}}}{M_{\text{div}}^{\text{real}}},$$

where values near 1 mean the generated diversity roughly matches the real reference batch.

Summary

method	NN-MSE	divMSE	div/real	meanGap	stdGap	cover	entropy
A1 xprev det	0.1223	0.0000	0.000	0.3298	0.5671	1	-0.000
A1 xprev +noise	0.1557	0.0874	0.171	0.3088	0.3843	2	0.141
A2 x0 fresh	0.1299	0.2566	0.501	0.1117	0.1677	4	0.530
A2 x0 coherent	0.2295	0.5247	1.025	0.0571	0.0554	6	0.759
A3 eps det	0.3334	0.6679	1.304	0.1542	0.0730	5	0.676
A3 eps DDPM	0.1930	0.5187	1.013	0.0194	0.0262	7	0.819

Pixel mean gap compares global brightness:

$$M_{\text{mean}} = \left| \frac{1}{ND} \sum_{i=1}^{N} \sum_{d=1}^{D} \hat{x}_{i,d} - \frac{1}{MD} \sum_{j=1}^{M} \sum_{d=1}^{D} x_{j,d}^{\text{real}} \right|.$$

Lower means the generated images have a similar average intensity to real MNIST.

Summary

method	NN–MSE	divMSE	div/real	meanGap	stdGap	cover	entropy
A1 xprev det	0.1223	0.0000	0.000	0.3298	0.5671	1	-0.000
A1 xprev +noise	0.1557	0.0874	0.171	0.3088	0.3843	2	0.141
A2 x0 fresh	0.1299	0.2566	0.501	0.1117	0.1677	4	0.530
A2 x0 coherent	0.2295	0.5247	1.025	0.0571	0.0554	6	0.759
A3 eps det	0.3334	0.6679	1.304	0.1542	0.0730	5	0.676
A3 eps DDPM	0.1930	0.5187	1.013	0.0194	0.0262	7	0.819

Centroid coverage and entropy are a lightweight class-spread diagnostic. Let c_k be the centroid of real images with digit label k. Assign each generated sample to the nearest centroid:

$$a_i = \arg\min_{k \in \{0,\dots,9\}} \frac{1}{D} \|\hat{x}_i - c_k\|_2^2 .$$

Coverage is the number of centroids used:

$$C = |\{k : \exists i, a_i = k\}| .$$

Entropy measures how evenly samples spread across these centroid assignments:

$$H = -\frac{1}{\log 10} \sum_{k=0}^{9} p_k \log p_k, \qquad p_k = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[a_i = k].$$

Higher coverage and entropy suggest better digit-class diversity, but this is only a rough proxy, not a trained classifier evaluation.

QA