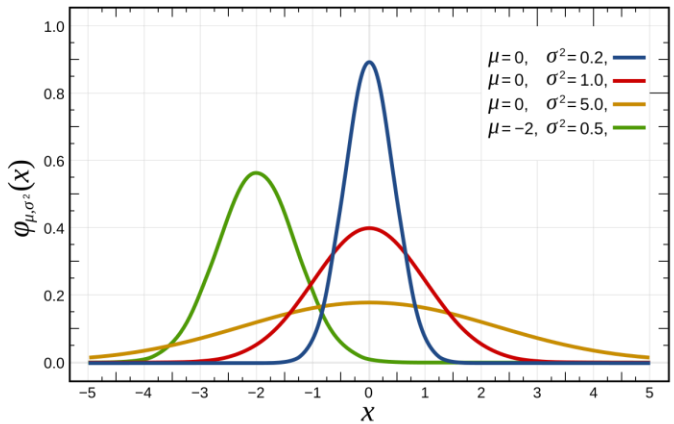
Advanced Machine Learning
Generative Model

Yu Wang
Assistant Professor
Department of Computer Science
University of Oregon

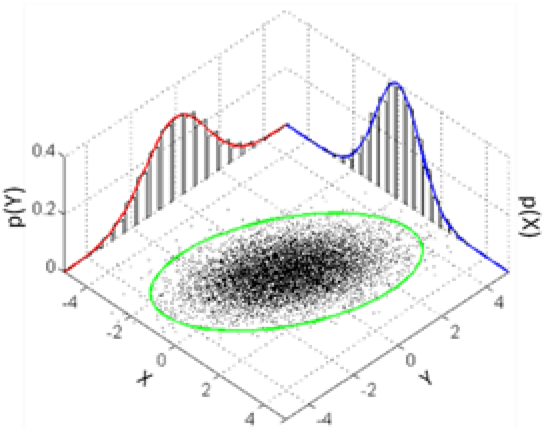
Summary

1D Gaussian Distribution

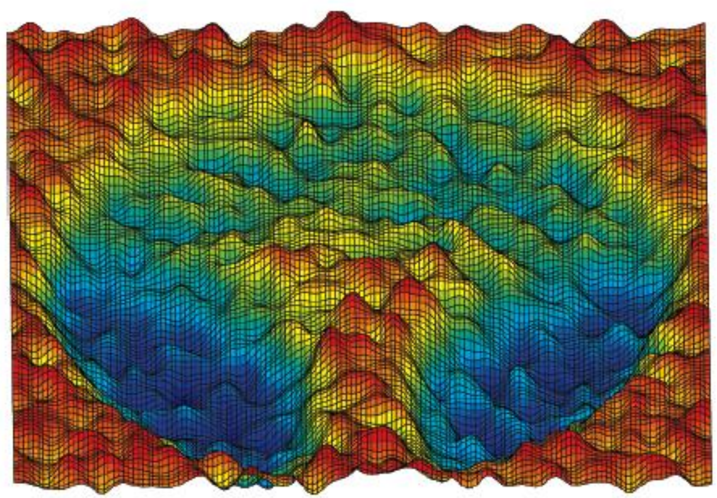


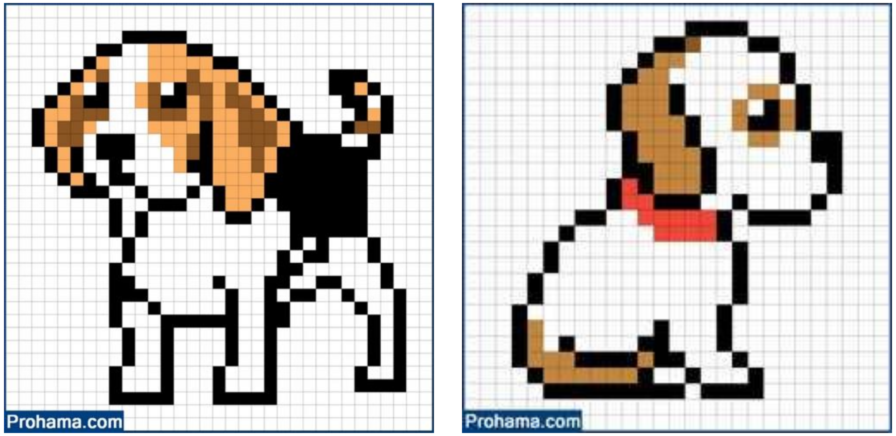
$\mathbb{R}$

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

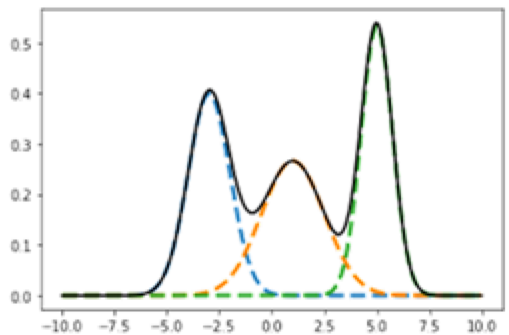
Using existing function to estimate what you do not know that can best fit your observation

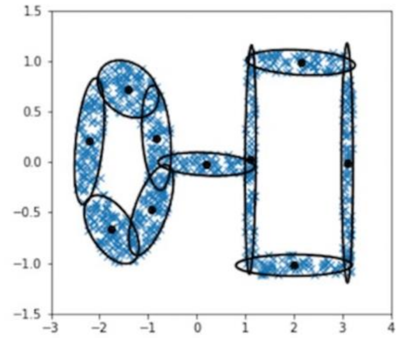
- **Deep Learning Methods**
  - Auto-Encoder (AE)
  - Variational AE (LLM is actually a VAE)
  - Generative Adversarial Network
  - Diffusion Model

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

Problem?

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





$\mathbb{R}^1, \mathbb{R}^2$

$\downarrow$

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

What you have is some low-dimensional data
But what you want to model is some high-dimensional data, how it could be?

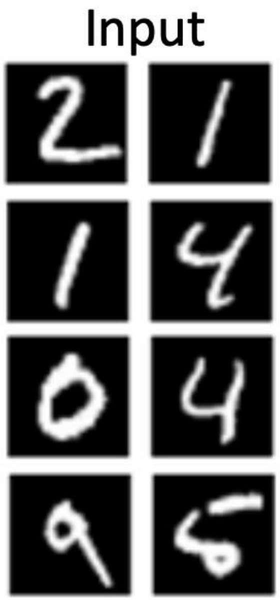
Problem?

What we want: model any data distribution

How to transform any data distribution to low dimensional data?

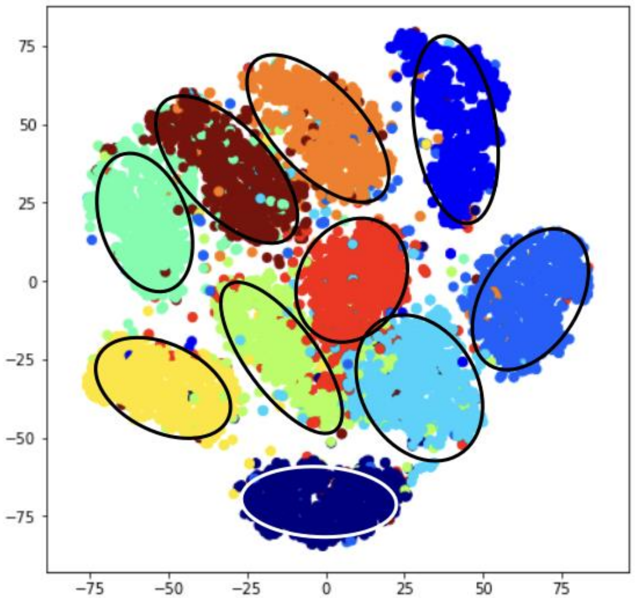
What we have: kernel density estimation to estimate low dimensional PDF

Kernel Density Estimation



Someway to transform

Transform back



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

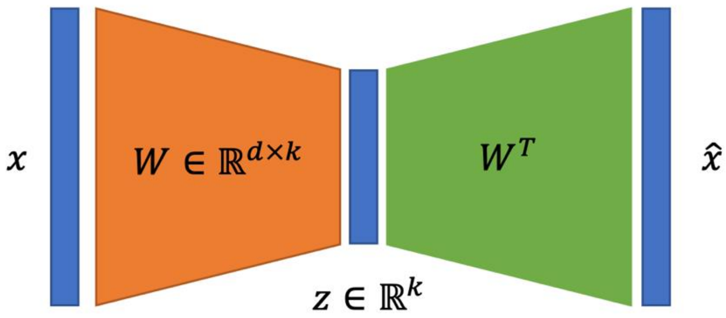
Using existing function to estimate what you do not know that can best fit your observation

↓ **PCA Dimensional Reduction**

- **Deep Learning Methods**
  - Auto-Encoder (AE)
  - Variational AE (LLM is actually a VAE)
  - Generative Adversarial Network
  - Diffusion Model

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

From PCA to Auto-Encoder



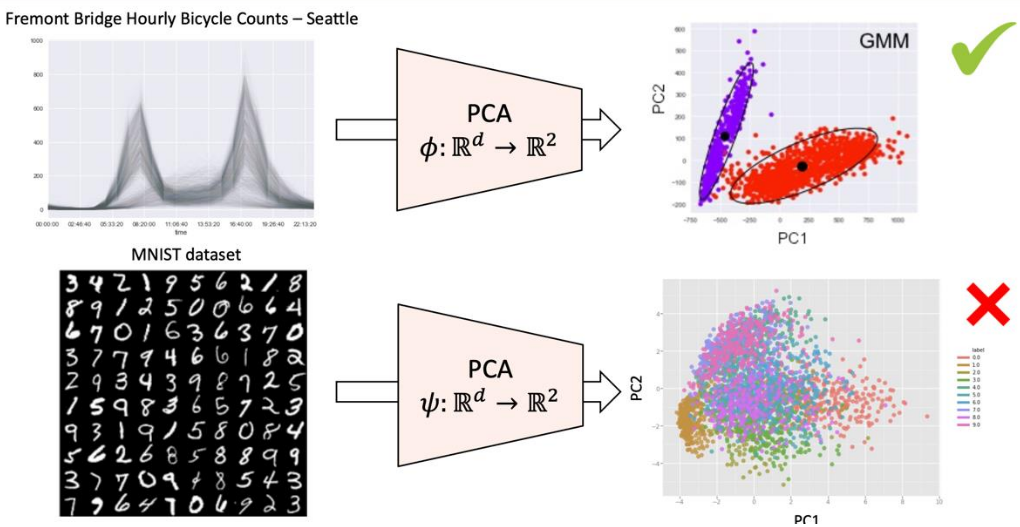
PCA:

- Forward transform: $z = W^T x$
- Inverse transform: $\hat{x} = Wz$

Linear dimensionality Reduction

$$\min_W \mathbb{E}_x[\|x - \hat{x}\|^2] = \mathbb{E}_x[\| x - WW^Tx\|^2]$$
$$s.t. \quad W^TW = I_{k\times k}$$

High-dimensional data often lives on non-linear manifolds that cannot be captured by linear models such as PCA

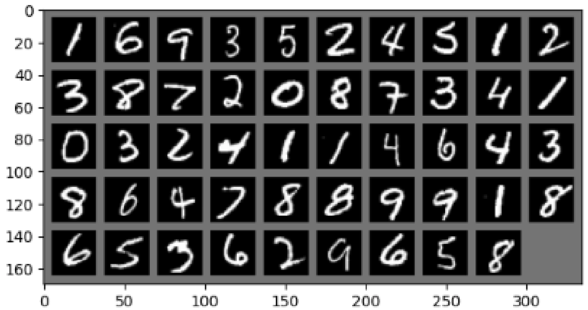


Can we add nonlinearity?

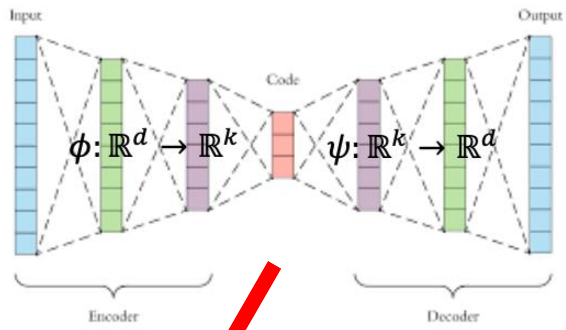
Yes, then it becomes neural network!

Auto-Encoder

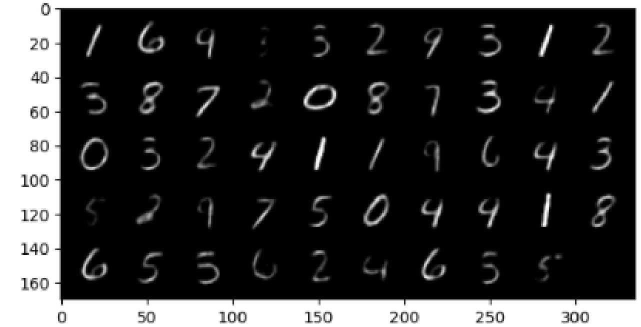
Input

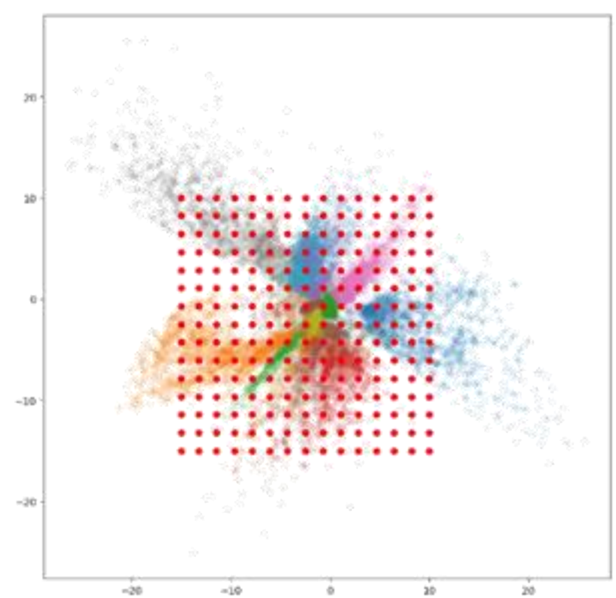


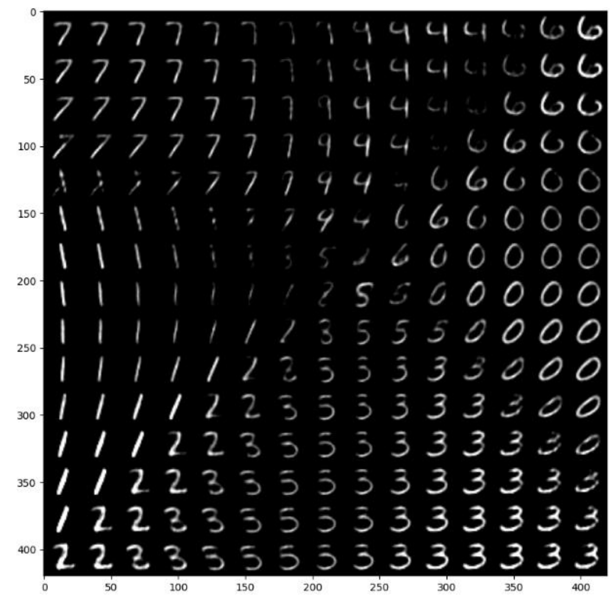
AE



Output

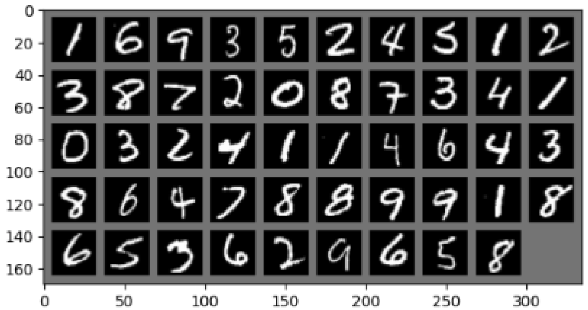




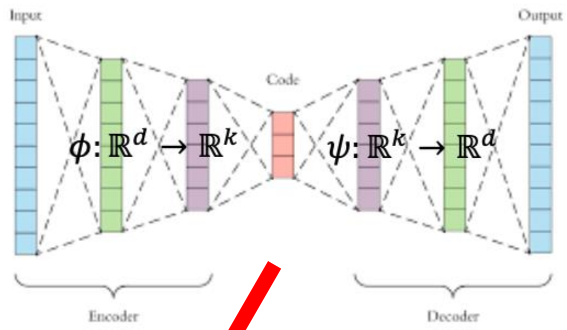


Class-supervised Auto-Encoder

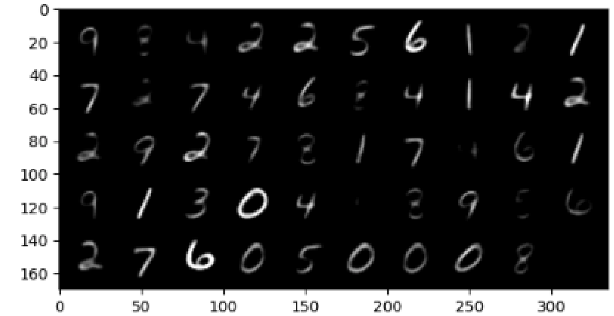
Input

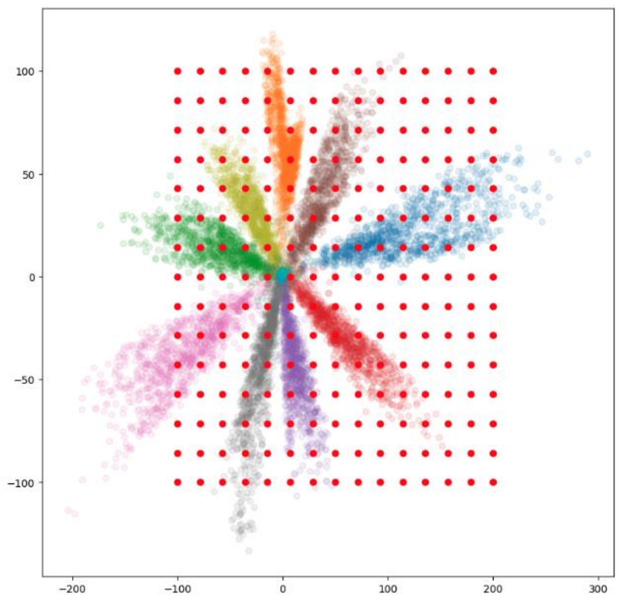


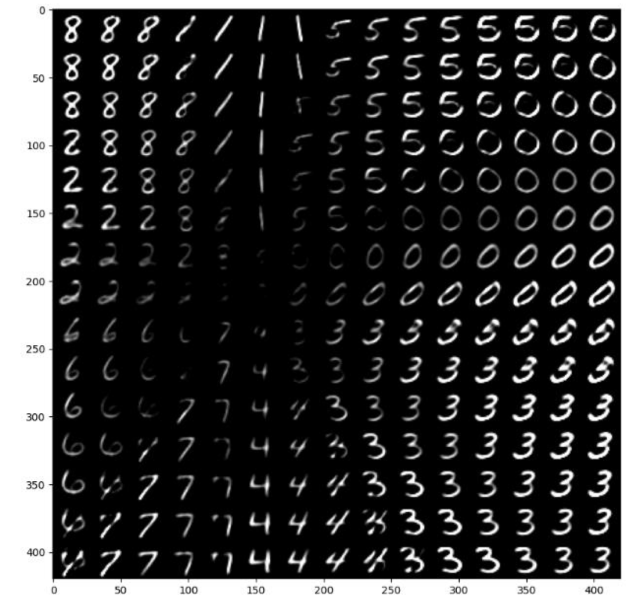
AE



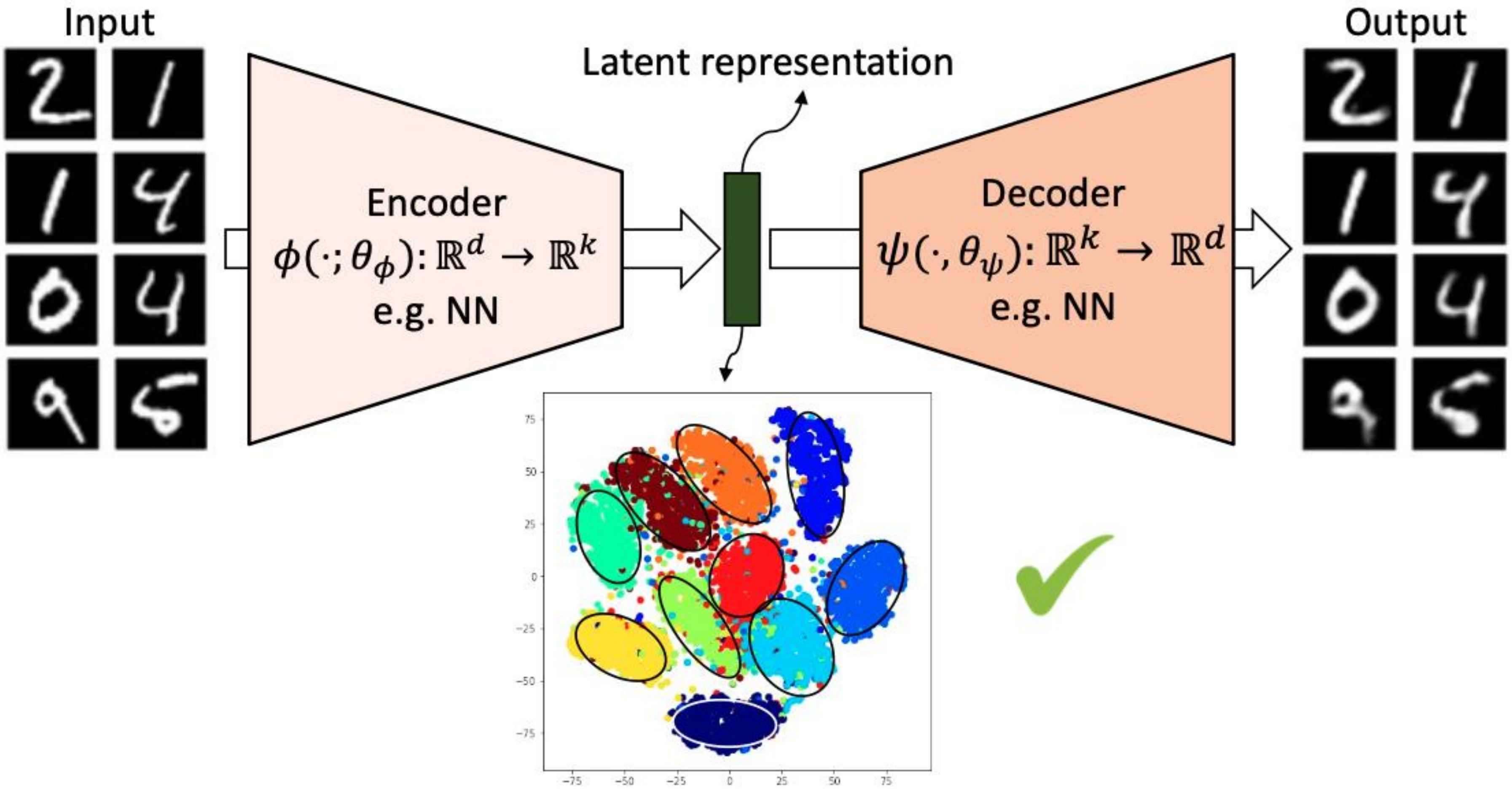
Output







Problem with AE

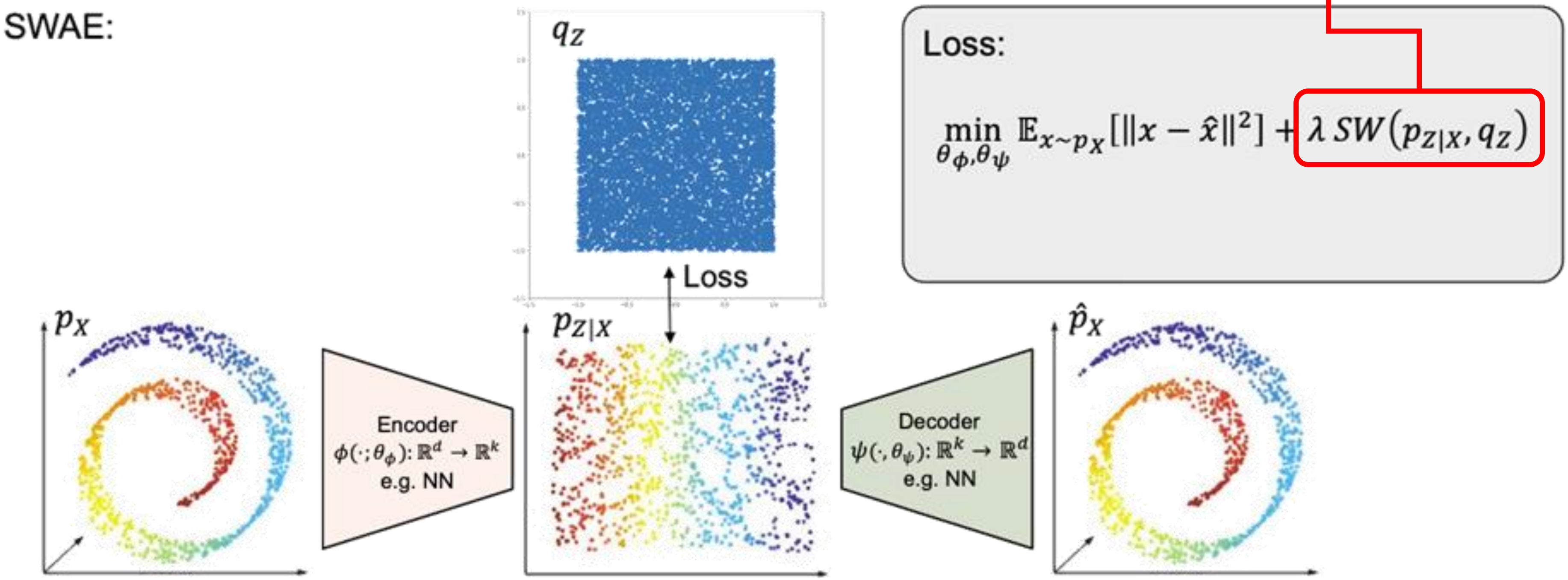


Need to estimate the latent distribution post-hoc!

Solution – Sliced Wasserstein AE

Sliced Wasserstein Distance between two distributions!

SWAE:



Solution – VAE

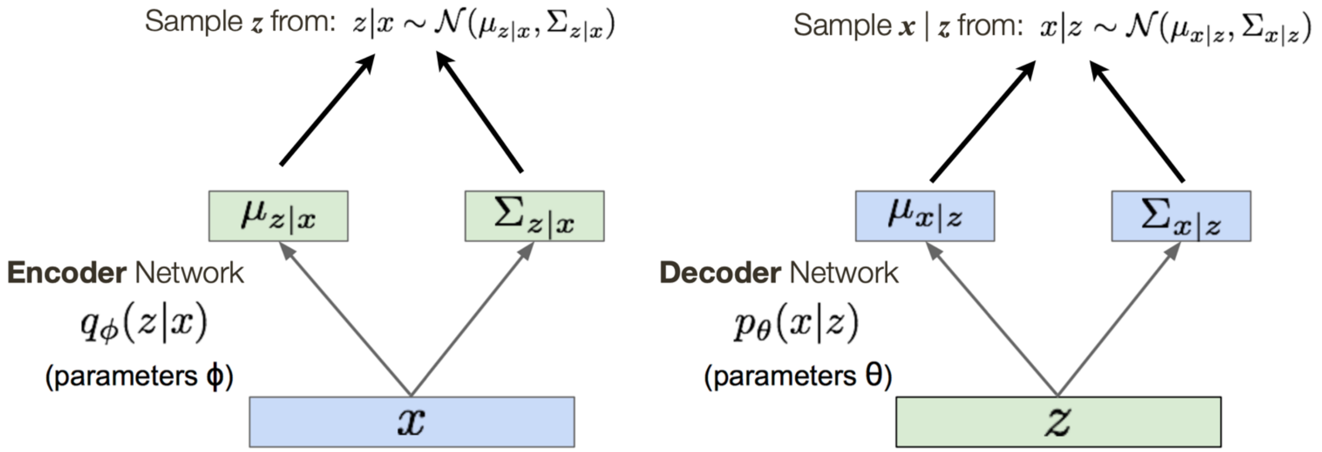
$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)}$$

$$\mathbb{E}_{x\sim p_X}[\|x-\hat{x}\|^2]$$

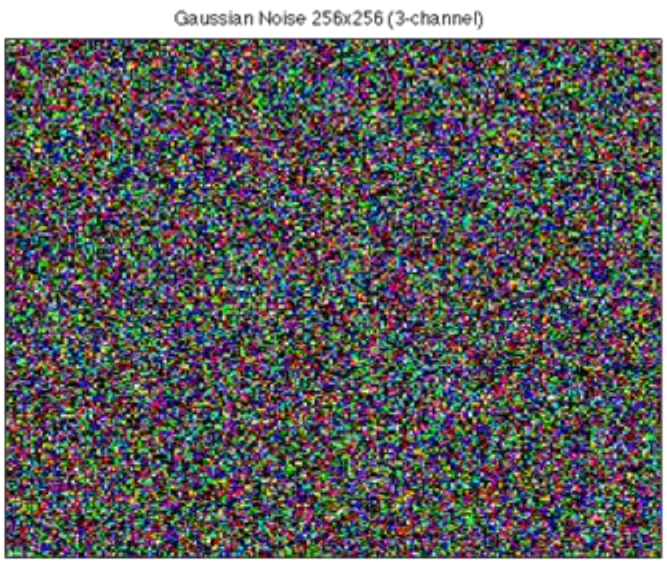
$$D_{\mathrm{KL}}(q_\phi(z|x)\,||\,p(z)) = \frac{1}{2}\sum_{j=1}^{d}\left[\sigma_j^2+\mu_j^2-1-\log\sigma_j^2\right]$$

(1) Reconstruction loss: given z – decoder – x and setup the reconstruction loss

(2) KL divergence: how to optimize the KL divergence between two gaussian distributions?

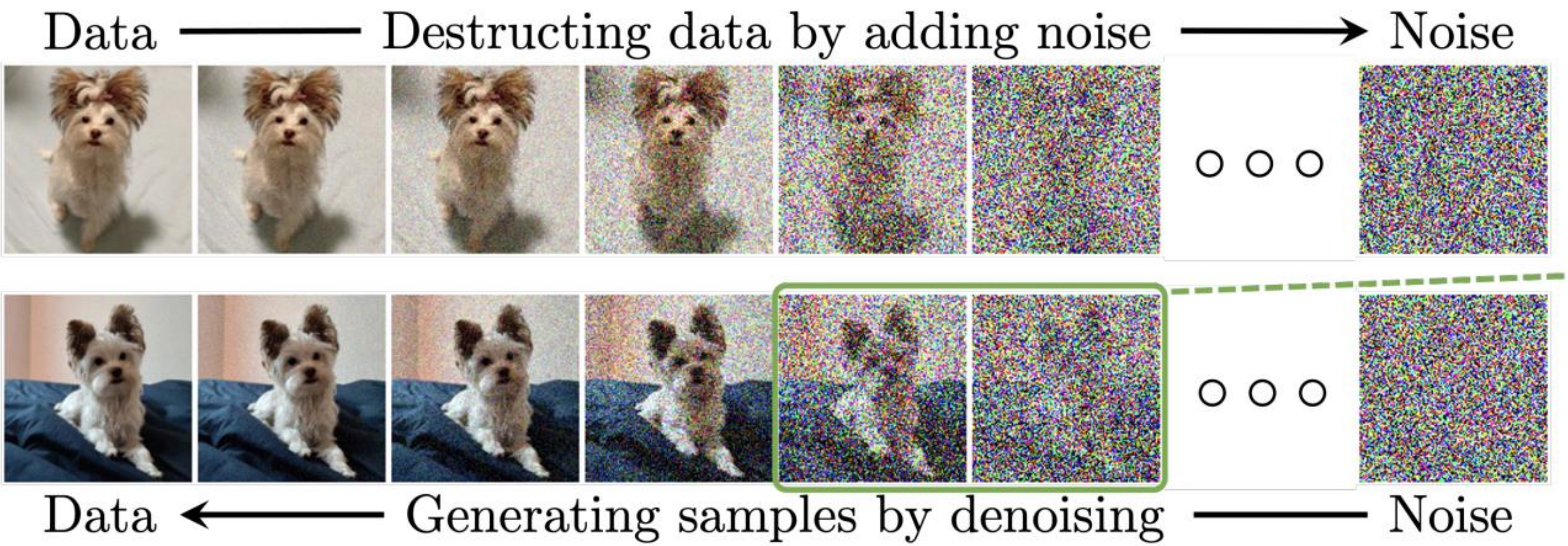


Problem



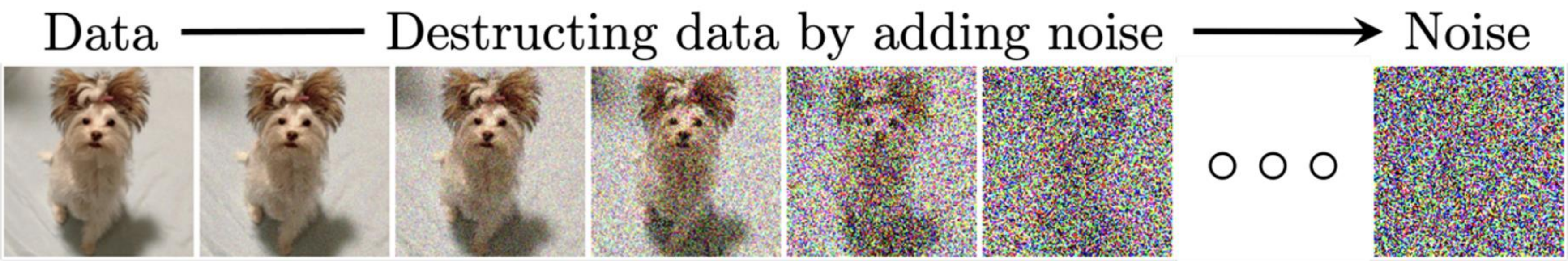
One-shot
Generation

Diffusion



Can we construct the image step by step?

Diffusion



data distribution $\mathbf{x}_0 \sim q(\mathbf{x}_0)$, $\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_T$ with transition kernel $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$

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

$\beta_t \in (0, 1)$ is a hyperparameter

Diffusion

data distribution $\mathbf{x}_0 \sim q(\mathbf{x}_0)$, $\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_T$ with transition kernel $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$

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

$\beta_t \in (0,1)$ is a hyperparameter

Recursive

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

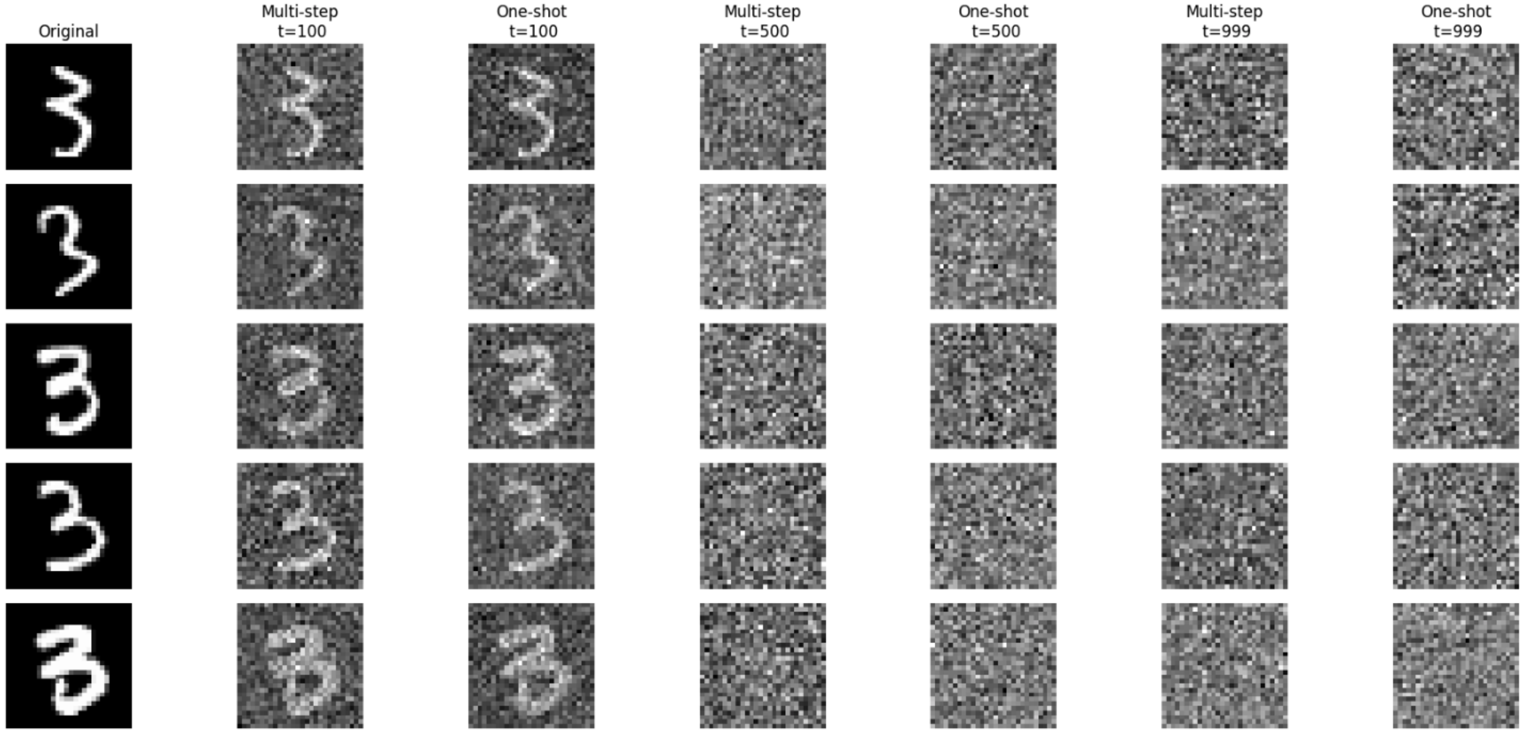
$$p(x_t \mid x_0, x_1, \ldots, x_{t-1}) = p(x_t \mid x_{t-1})$$

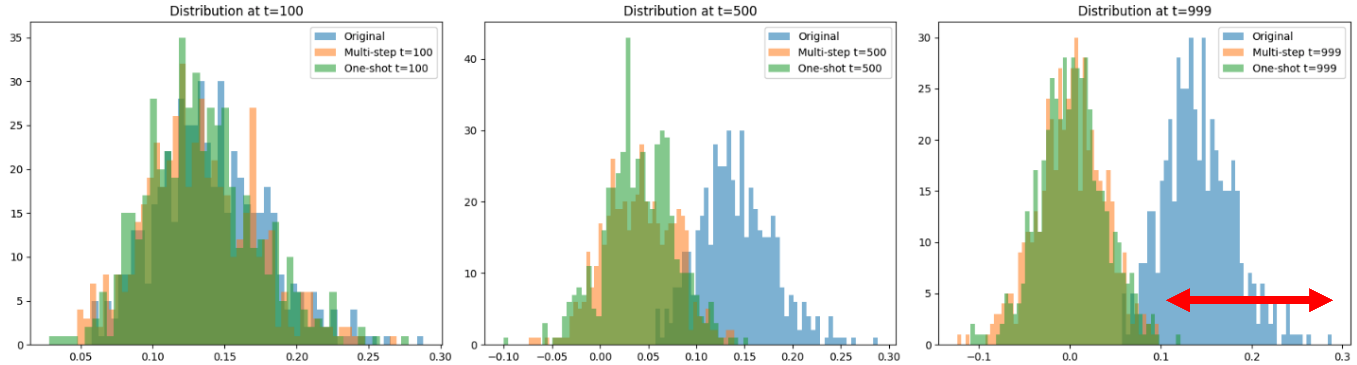
Markov Chain Property

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

with $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$,

Diffusion

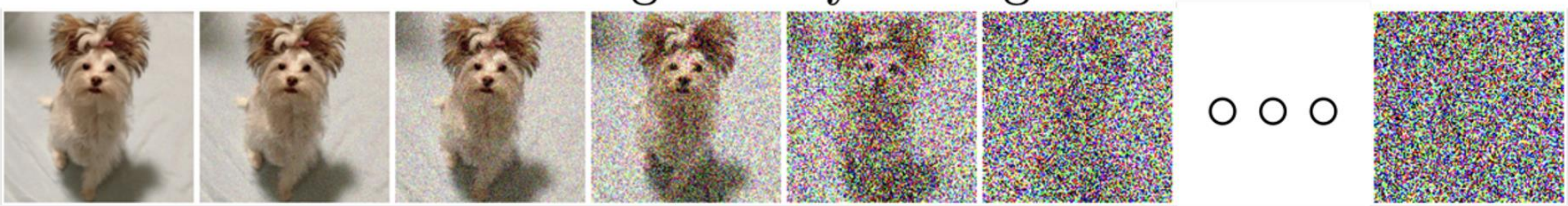




Standard Gaussian

Diffusion

Data —— Destructing data by adding noise ⟶ Noise



data distribution $\mathbf{x}_0 \sim q(\mathbf{x}_0)$. $\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_T$ with transition kernel $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$

$\beta_t \in (0, 1)$ is a hyperparameter

$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I})$.   with $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$,

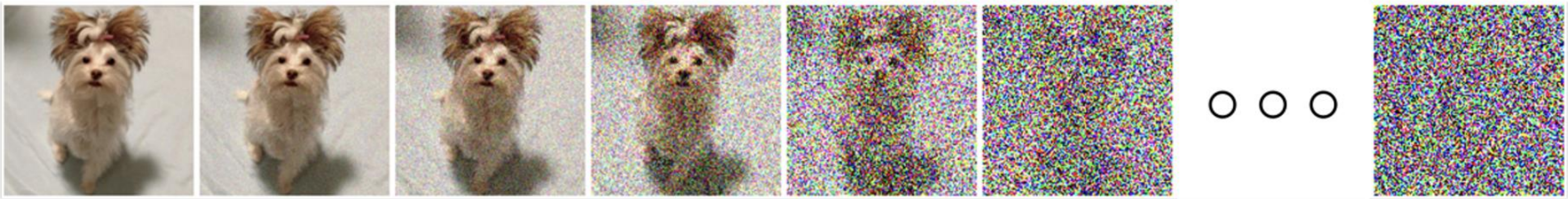
$\boldsymbol{t \to \infty, \alpha_t \to 0,}$

$p(\mathbf{x}_{t-1}|\mathbf{x}_t) \longleftarrow$ $\boldsymbol{q(\mathbf{x}_t|\mathbf{x}_0) \to \mathcal{N}(0, 1)}$

Data ⟵ Generating samples by denoising —— Noise

Diffusion

Data ⟶ Destructing data by adding noise ⟶ Noise



data distribution $\mathbf{x}_0 \sim q(\mathbf{x}_0)$.

$\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_T$ with transition kernel $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$

$\beta_t \in (0, 1)$ is a hyperparameter

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

with $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$,

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \xrightarrow{\text{Approximate}} p(\mathbf{x}_{t-1}|\mathbf{x}_t) \longleftarrow$$

$\boldsymbol{t \to \infty, \alpha_t \to 0,}$

$\boldsymbol{q(\mathbf{x}_t|\mathbf{x}_0) \to \mathcal{N}(0, 1)}$

Data ⟵ Generating samples by denoising ⟵ Noise

Diffusion

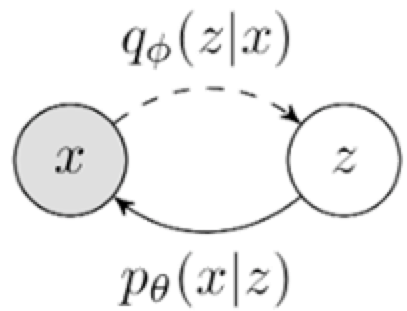
$$p(x) = \int_z p_\theta(x|z)p(z)$$

$$p(x) = \int q_\phi(z|x)\frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}$$

$$\log p(x) = \log \mathbb{E}_{z\sim q_\phi(z|x)}\left[\frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}\right]$$

$$\log p(x) \geq \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}\right]$$

Figure 1 - Graphical Model for VAE



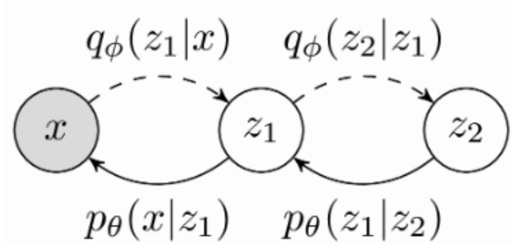
$$p(x) = \int_{z_1}\int_{z_2} p_\theta(x, z_1, z_2)dz_1, dz_2$$

$$p(x) = \int\int q_\phi(z_1, z_2|x)\frac{p_\theta(x, z_1, z_2)}{q_\phi(z_1, z_2|x)}$$

$$p(x) = \mathbb{E}_{z_1,z_2\sim q_\phi(z_1,z_2|x)}\left[\frac{p_\theta(x, z_1, z_2)}{q_\phi(z_1, z_2|x)}\right]$$

$$\log p(x) \geq \mathbb{E}_{z_1,z_2\sim q_\phi(z_1,z_2|x)}\left[\log \frac{p_\theta(x, z_1, z_2)}{q_\phi(z_1, z_2|x)}\right]$$

Figure 2 - A Hierarchical VAE



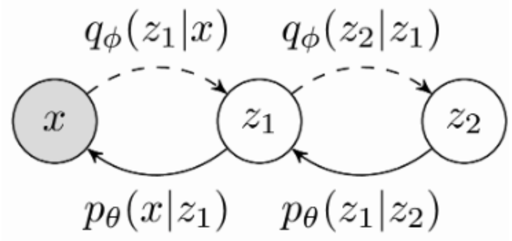
$$p(x, z_1, z_2) = p(x|z_1)p(z_1|z_2)p(z_2)$$

$$q(z_1, z_2|x) = q(z_1|x)q(z_2|z_1)$$

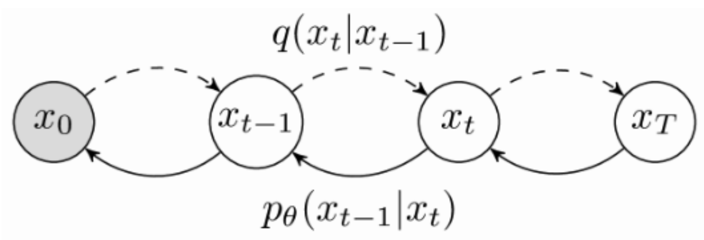
Diffusion

$$\log p(x) \geq \mathbb{E}_{z_1, z_2 \sim q_\phi(z_1, z_2|x)} \left[\log \frac{p_\theta(x, z_1, z_2)}{q_\phi(z_1, z_2|x)} \right]$$

Figure 2 - A Hierarchical VAE



$$\log \mathrm{p(x)} \geq \mathbb{E}_{x_{1:T} \sim q_\phi(x_{1:T}|x_0)} [\log \frac{p_\theta(x_{0:T})}{q_\phi(x_{1:T}|x_0)}]$$



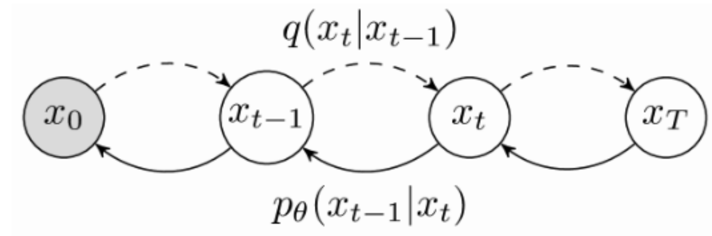
$$= \mathbb{E}_{x_{1:T} \sim q_\phi(x_{1:T}|x_0)} [\log \frac{p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)}{\prod_{t=1}^{T} q_\phi(x_t|x_{t-1})}]$$

$$= \mathbb{E}_{x_{1:T} \sim q_\phi(x_{1:T}|x_0)} [\log p_\theta(x_T) + \sum_{t=1}^{T} \log \frac{p_\theta(x_{t-1}|x_t)}{q_\phi(x_t|x_{t-1})}]$$

Link

Diffusion

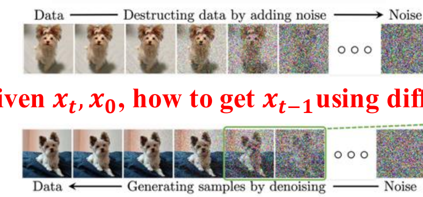
$$\log \mathrm{p(x)} \geq \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[\log p(x_T) + \sum_{t=1}^{T} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right]$$



Link

$$L := \mathbb{E}_q\left[\underbrace{-\log p(x_T) + \log q(x_T|x_0)}_{L_T} - \underbrace{\log p_\theta(x_0|x_1)}_{L_0} - \underbrace{\sum_{t>1}^{T} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)}}_{L_{t-1}}\right]$$

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



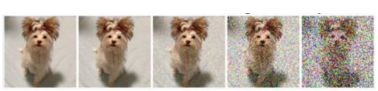
Given x_t, x_0, how to get x_{t-1} using diffusion

Given x_t, how to revert x_{t-1} using decoder

$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)$

$$L := \mathbb{E}_q\left[\underbrace{-\log p(x_T) + \log q(x_T|x_0)}_{L_T} - \underbrace{\log p_\theta(x_0|x_1)}_{L_0} - \underbrace{\sum_{t>1}^{T} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)}}_{L_{t-1}}\right]$$

$$\mathrm{KL}(P \,\|\, Q) = \frac{1}{2}\left[\log \frac{|\Sigma_1|}{|\Sigma_0|} - d + \mathrm{tr}(\Sigma_1^{-1}\Sigma_0) + \boxed{(\mu_1 - \mu_0)^T \Sigma_1^{-1} (\mu_1 - \mu_0)}\right]$$

x_0 

The problem is every-time you need to calculate the target mean value

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

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

$\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$

$$L_{t-1} = \mathbb{E}_{t, x_t, x0}\left[\frac{1}{2\sigma_t^2} \|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\|\right] + C$$

Diffusion

$$L_{t-1} = \mathbb{E}_{t,x_t,x0}\left[\frac{1}{2\sigma_t^2}\|\tilde{\mu}_t(x_t,x_0) - \mu_\theta(x_t,t)\|\right] + C$$

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

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

$\alpha_t := 1-\beta_t$ and $\bar{\alpha}_t := \prod_{s=0}^{t}\alpha_s$

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

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\boldsymbol{\epsilon} \qquad \boldsymbol{x}_0 = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}}{\sqrt{\bar{\alpha}_t}}$$

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

$$\tilde{u}_t(x_t,x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\left(\frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}}{\sqrt{\bar{\alpha}_t}}\right) + \frac{\sqrt{\alpha_t(1-\bar{\alpha}_{t-1})}}{1-\bar{\alpha}_t}\boldsymbol{x_t} = \boxed{\frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x_t} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right)}$$

For a given x_t, add a noise

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

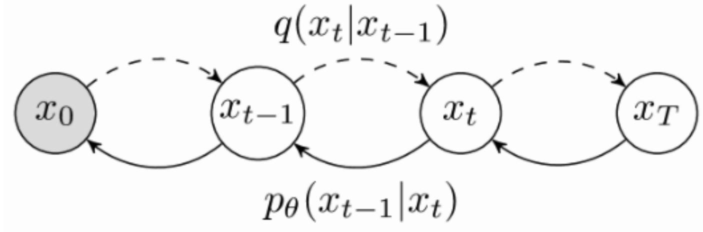
$$= \mathbb{E}_{x_0,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(x_t(x_0,\epsilon),t)\|\right]$$

We do not need to calculate target mean but only do forward diffusion

Diffusion

$$L_{t-1} = \mathbb{E}_{t,x_t,x0}\left[\frac{1}{2\sigma_t^2}\|\tilde{\mu}_t(x_t,x_0) - \mu_\theta(x_t,t)\|\right] + C$$

$$= \mathbb{E}_{x_0,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(x_t(x_0,\epsilon),t)\|\right]$$



Link

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

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

$\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$

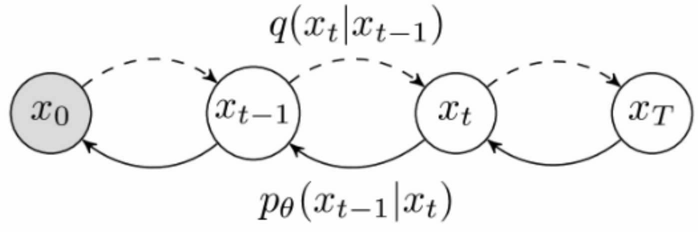
Algorithm 1 Training

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

Diffusion

$$L_{t-1} = \mathbb{E}_{t,x_t,x0}\left[\frac{1}{2\sigma_t^2}\|\tilde{\mu}_t(x_t,x_0) - \mu_\theta(x_t,t)\|\right] + C$$

$$= \mathbb{E}_{x_0,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(x_t(x_0,\epsilon),t)\|\right]$$



Link

Algorithm 2 Sampling

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

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

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

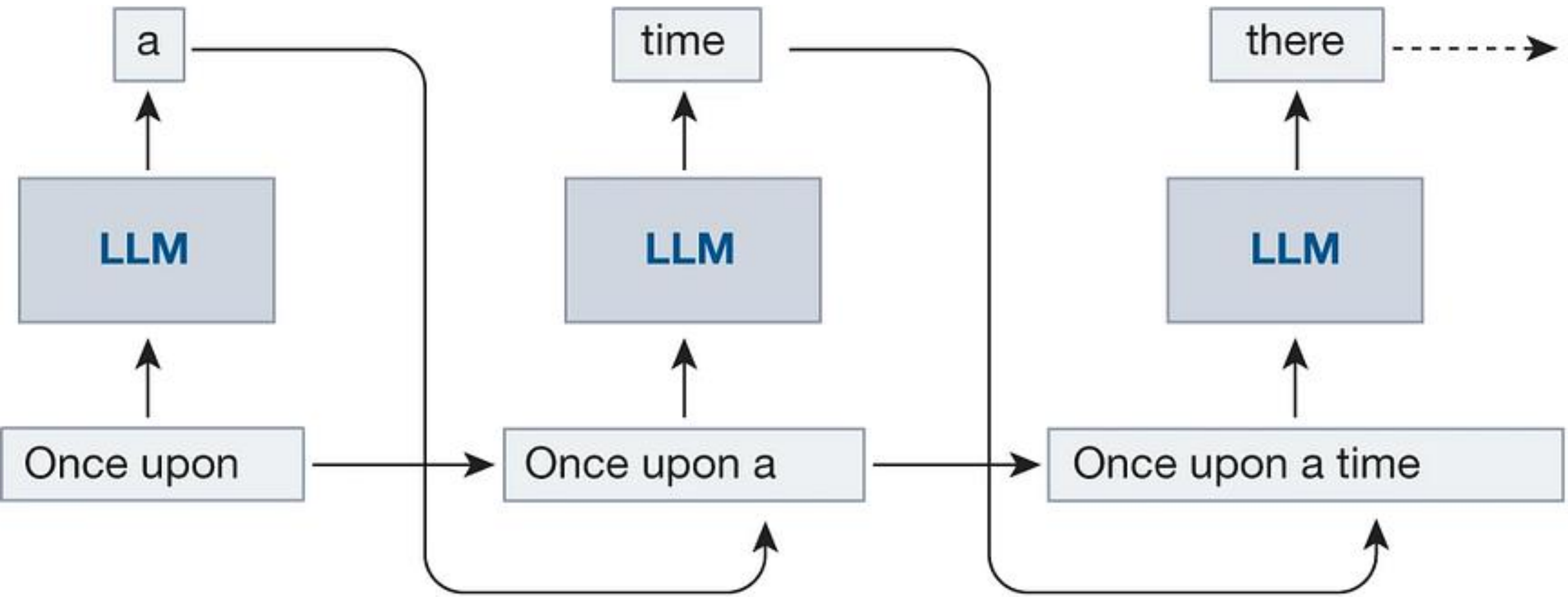
Diffusion

Code Demo

LLM Generation

How can we model the LLM generation under our framework?

LLM Generation



$$x_1^1 \rightarrow x_2^1 \rightarrow x_3^1 \rightarrow \cdots \rightarrow x_n^1$$

$$x_1^2 \rightarrow x_2^2 \rightarrow x_3^2 \rightarrow \cdots \rightarrow x_n^2$$

$$\cdots \qquad \cdots \qquad \cdots$$

$$x_1^S \rightarrow x_2^S \rightarrow x_3^S \rightarrow \cdots \rightarrow x_n^S$$

Padding to be the same length

LLM Generation

$$P(X) = \prod_{s=1}^{|S|} P(X_s) = \prod_{s=1}^{|S|} P(X_1, X_2, \dots, X_{l_S})$$
$$= \prod_{s=1}^{|S|} \prod_{l=2}^{l_S} \boldsymbol{P(X_l | X_{1:l-1})}$$

Different sequences are independent

Given previously observed sequences, what is the probability of observing the ground-truth next token?