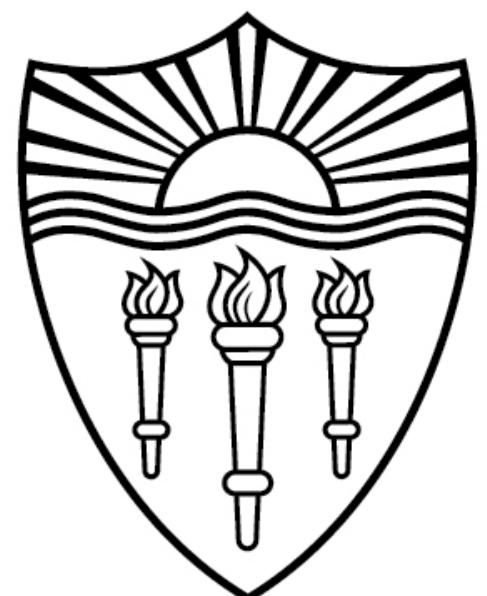


CSCI 544: Applied Natural Language Processing

Deep Generative Models for Image Generation

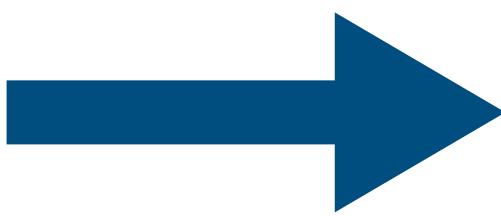
Xuezhe Ma (Max)



USC University of
Southern California

Generative Models

- What are generative models
 - Learning to generate new data from samples



Generative Models

- Why generative models
 - Requiring no labeled data
 - A good way to learn knowledge from data



What I cannot create, I do not understand.

— *Richard P. Feynman* —

AZ QUOTES

Deep Generative Models

- **Distribution-based Generative Models**
 - Auto-regressive Models
 - Generative (Normalizing) Flows
 - Variational Auto-Encoders (VAEs)
 - Diffusion Models
- **Non-distribution Generative Models**
 - Generative Adversarial Networks (GANs)

Distribution-based Generative Models

- Goal: learn to generate new data from samples
 - How?
 - To model the data distribution $P(X)$
 - Closed-form analytic solution
 - Exact density estimation via “black-box” deep neural networks
 - Density/distribution approximation

Distribution-based Generative Models

- Goal: learn to generate new data from samples
 - How?
 - To model the data distribution $P(X)$
 - Closed-form analytic solution
 - Exact density estimation via “black-box” deep neural networks
 - Density/distribution approximation

Closed-form Analytic Solution

- Providing a closed-form analytic solution of $P(X)$

- Kernel-based approaches
- Gaussian process
- ...

- Pros

- Theoretically grounded
- Analytic solution for future derivations

- Cons

- Limited capacity
- Unable to model complex data/distributions

Distribution-based Generative Models

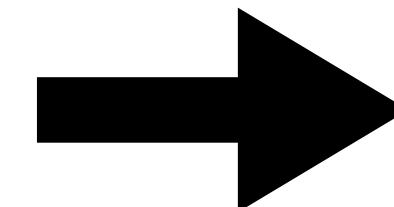
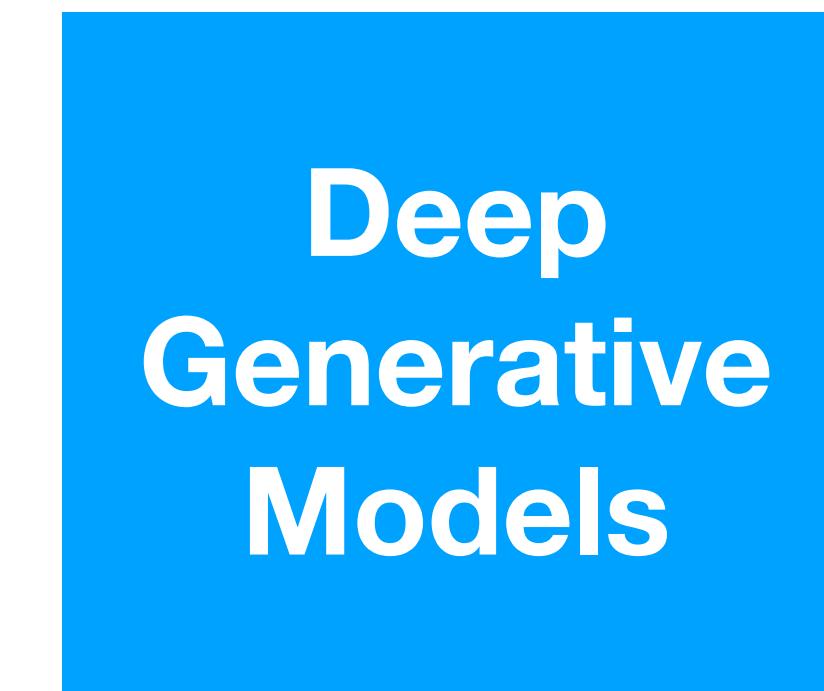
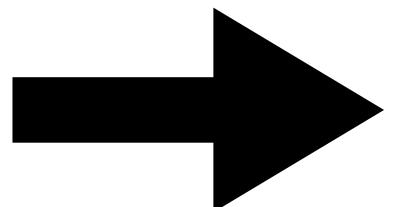
- Goal: learn to generate new data from samples
 - How?
 - To model the data distribution $P(X)$
 - Closed-form analytic solution
 - Exact density estimation via “black-box” deep neural networks
 - Density/distribution approximation

Deep Generative Models w. Exact Density Estimation

- Exact density estimation via deep neural networks
 - Autoregressive models
 - Generative (normalizing) flows



X



$P(X)$
Value only!

Autoregressive Models

- **Sequential data**

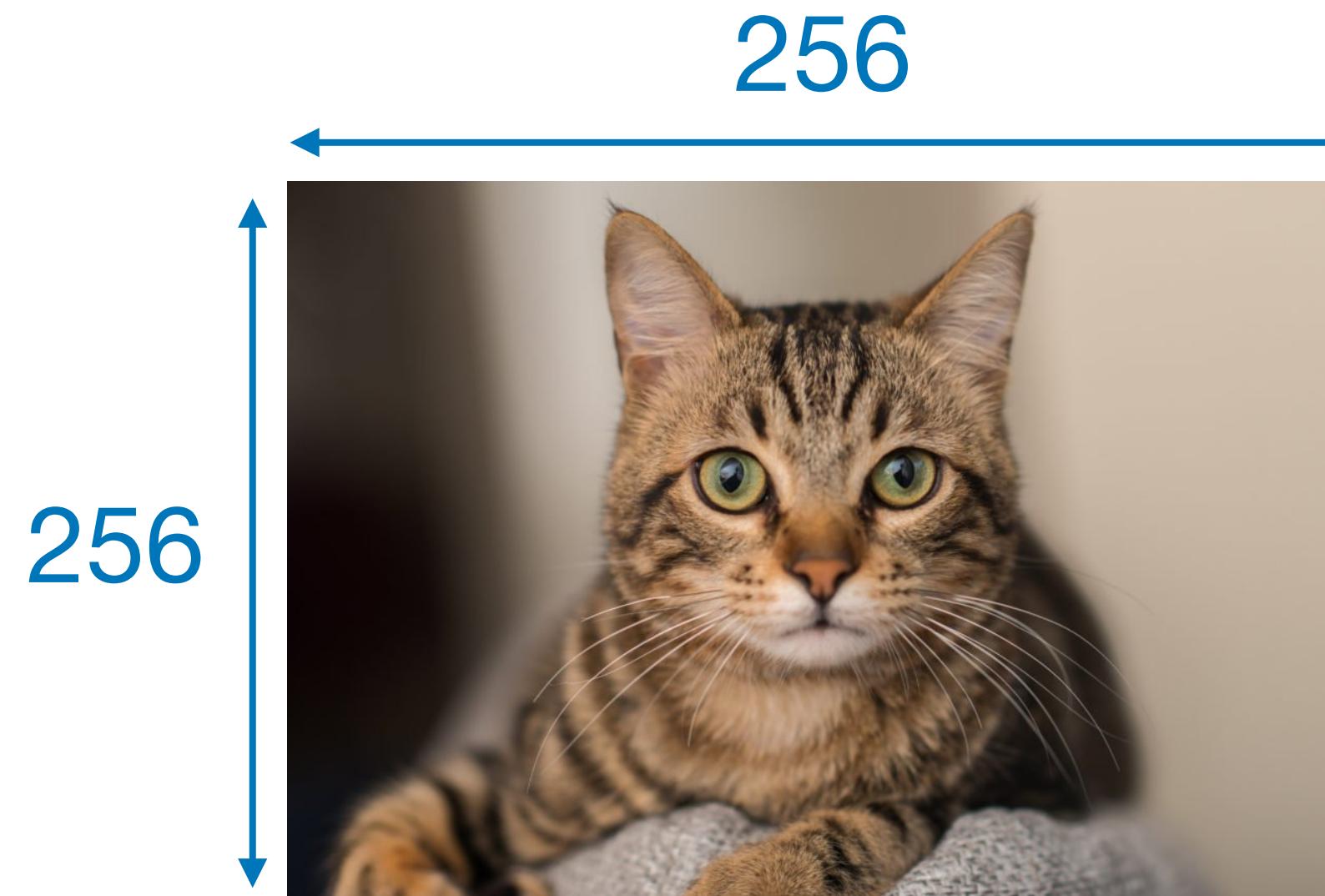
- $X = \{x_1, x_2, \dots, x_L\}$
- **Languages:** sequences of words
- **Images:** sequences of pixels
- **Audios:** sequences of signals

- **Autoregressive Factorization**

$$p_{\theta}(X) = \prod_{t=1}^T p_{\theta}(x_t | x_{<t})$$

- Autoregressive models
 - RNN, Transformer, ...

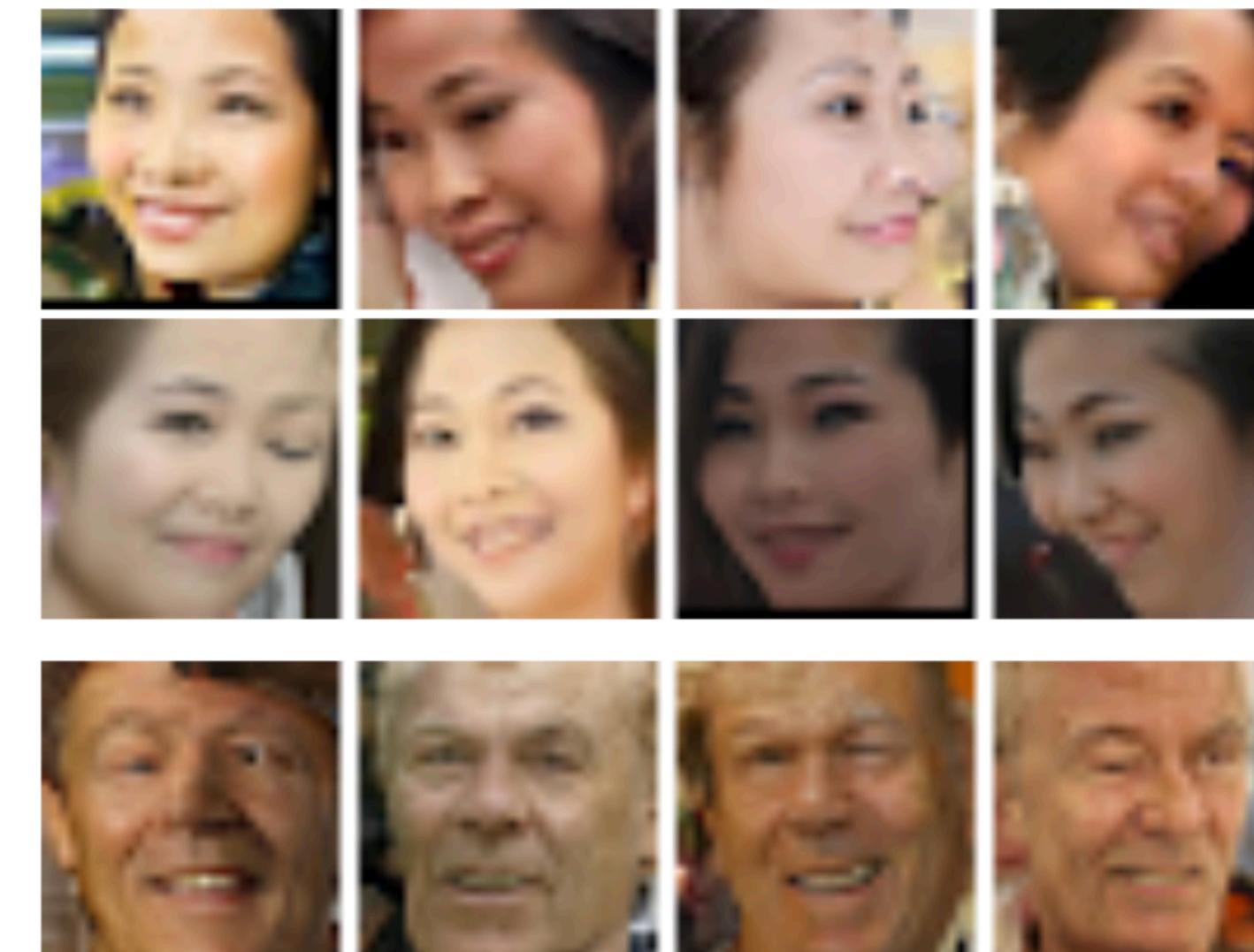
Problems on Autoregressive Models for Image



$256 \times 256 \times 3 = 131072$ pixels

Problems:

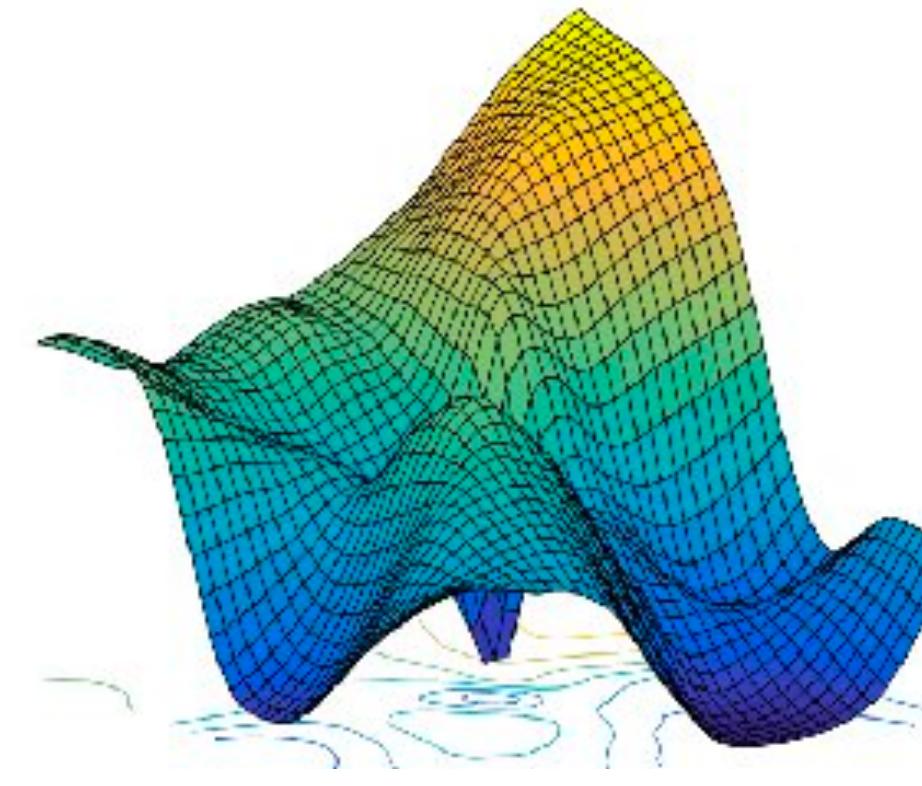
- One pre-defined order
 - No clear order for data like images
- Error propagation
 - Limited context at beginning



Generative (Normalizing) Flows

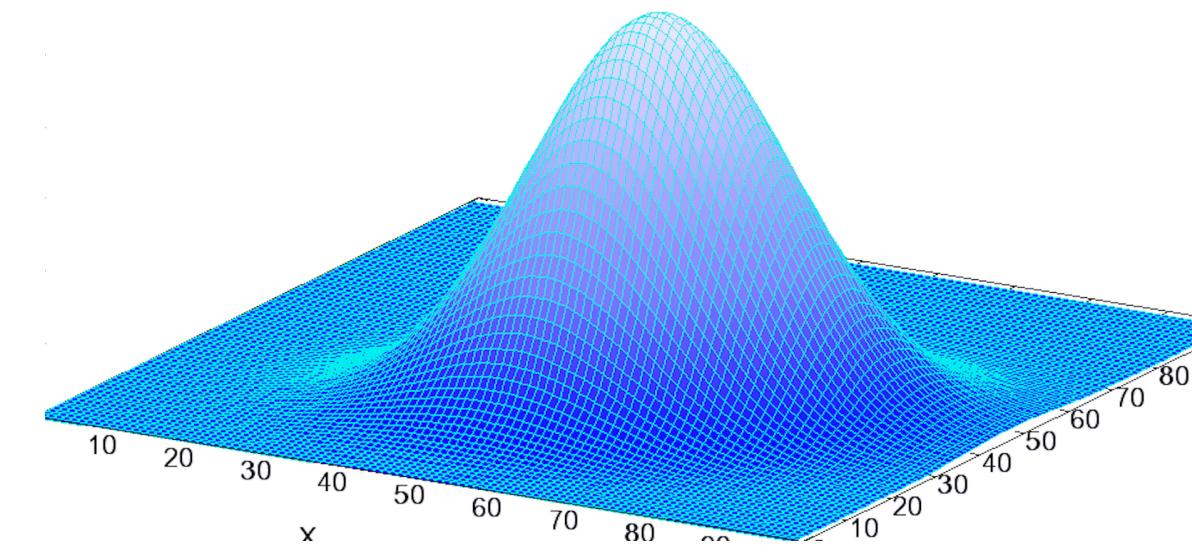
- **Modeling density via invertible mapping**
 - Directly modeling the joint distribution of all variates in X
 - Exact density estimation (no approximation)

Generative (Normalizing) Flows



$$X \sim p_\theta(X)$$

$$\begin{array}{c} \Gamma = f_\theta(X) \\ \longleftrightarrow \\ X = f_\theta^{-1}(\Gamma) \end{array}$$



$$\Gamma \sim \text{Normal}(0, I)$$

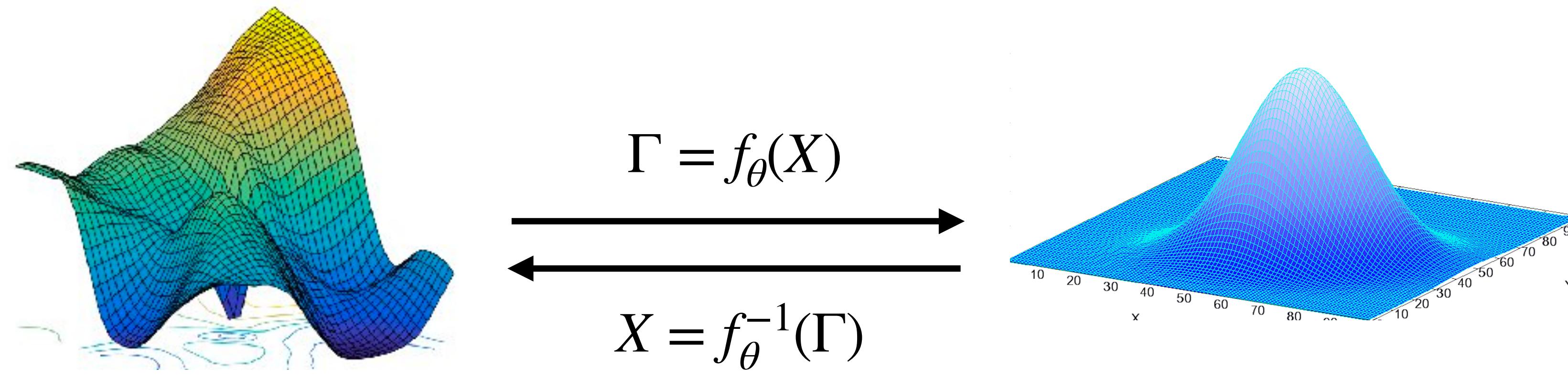
Change of Variable formula:

$$p_\theta(x) = p_\Gamma(f_\theta(x)) \left| \det \left(\frac{\partial f_\theta(x)}{\partial x} \right) \right|$$

Normal

Jacobian Matrix

Generative (Normalizing) Flows



$$X \sim p_\theta(X)$$

$$\Gamma \sim \text{Normal}(0, I)$$

Change of Variable formula:

$$p_\theta(x) = p_\Gamma(f_\theta(x)) \left| \det \left(\frac{\partial f_\theta(x)}{\partial x} \right) \right|$$

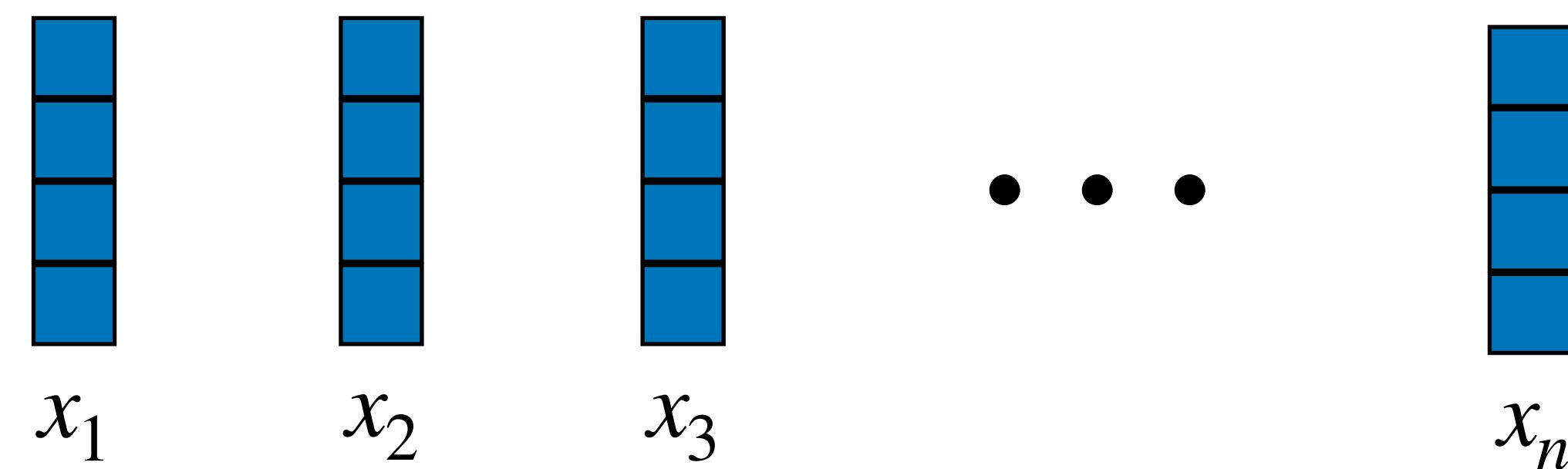
Generative Flow: A series of such f

$$X \xrightarrow[g_1]{f_1} H_1 \xrightarrow[g_2]{f_2} H_2 \xrightarrow[g_3]{f_3} \dots \xrightarrow[g_K]{f_K} \Gamma$$

Generative (Normalizing) Flows

- ActNorm

$$y_i = s \odot x_i + b$$



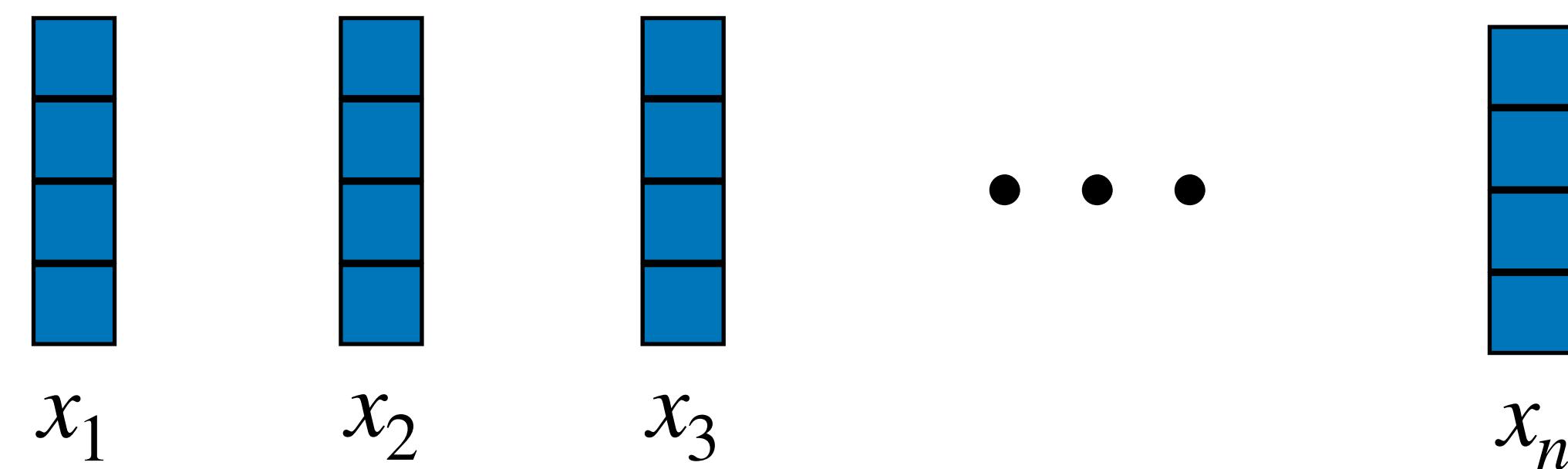
Generative (Normalizing) Flows

- ActNorm

$$y_i = s \odot x_i + b$$

- Invertible Linear

$$y_i = Wx_i$$



Generative (Normalizing) Flows

- ActNorm

$$y_i = s \odot x_i + b$$

- Invertible Linear

$$y_i = Wx_i$$

- Affine Coupling

$$x_a, x_b = \text{split}(x)$$

$$y_a = x_a$$

$$y_b = s(x_a) \odot x_b + b(x_a)$$

$$y = \text{concat}(y_a, y_b),$$



x_1



x_2

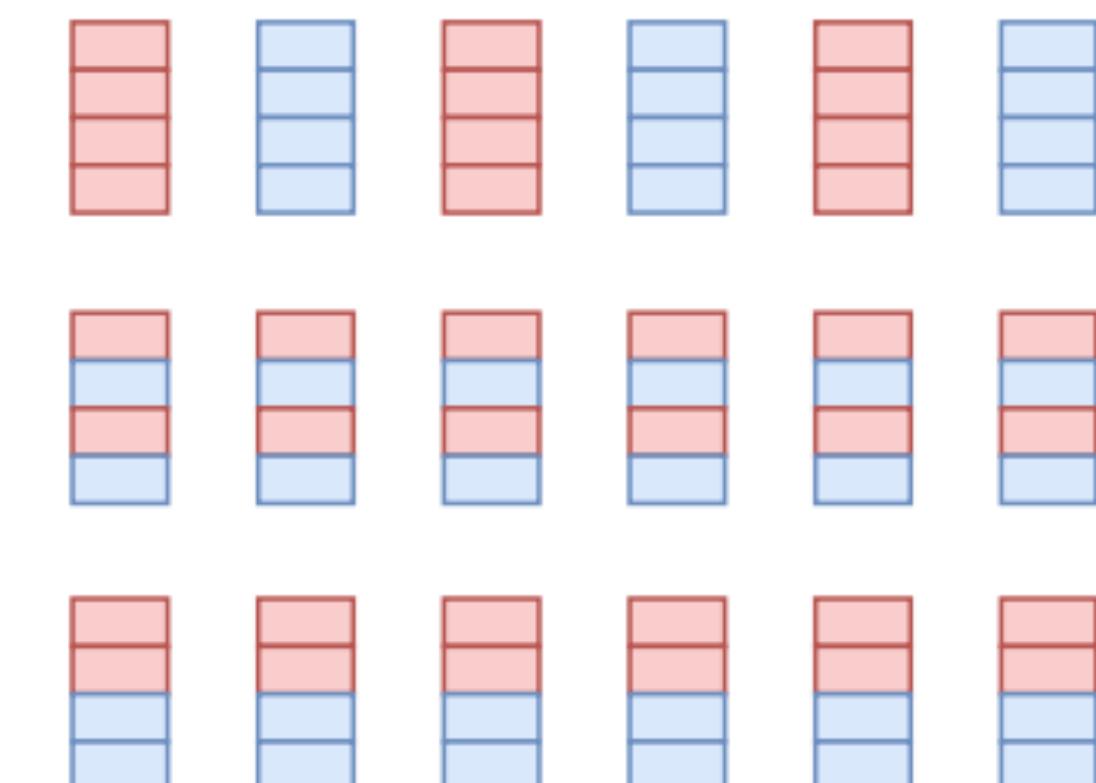


x_3

• • •



x_n



Generative (Normalizing) Flows: Pros and Cons

- Modeling the exact distribution $P(X)$
- No auto-regressive factorization
- A large number of layers: invertible function f_i is very weak
- Determinant calculation is expensive



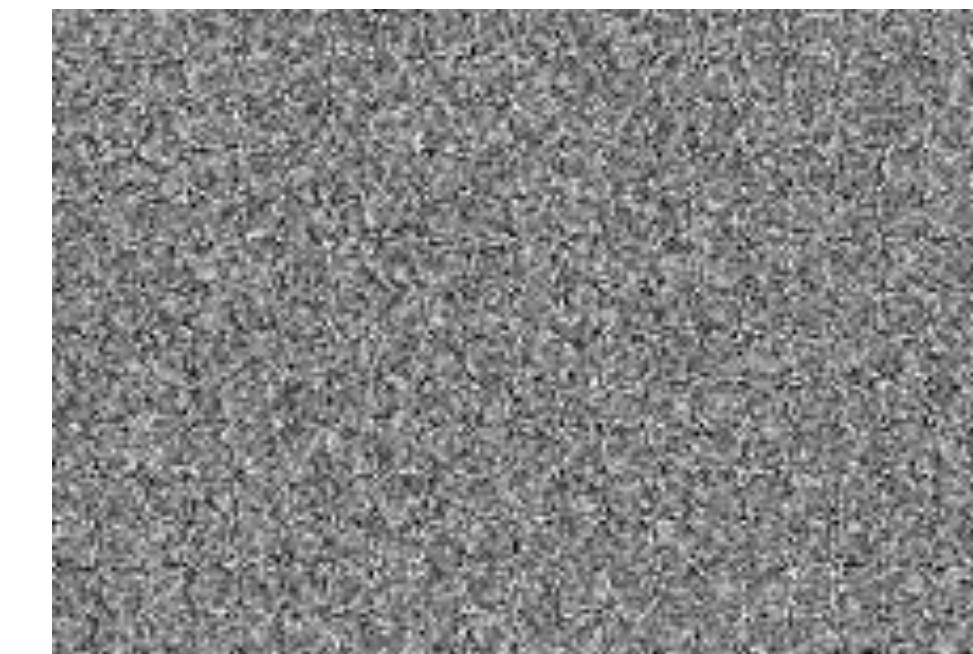
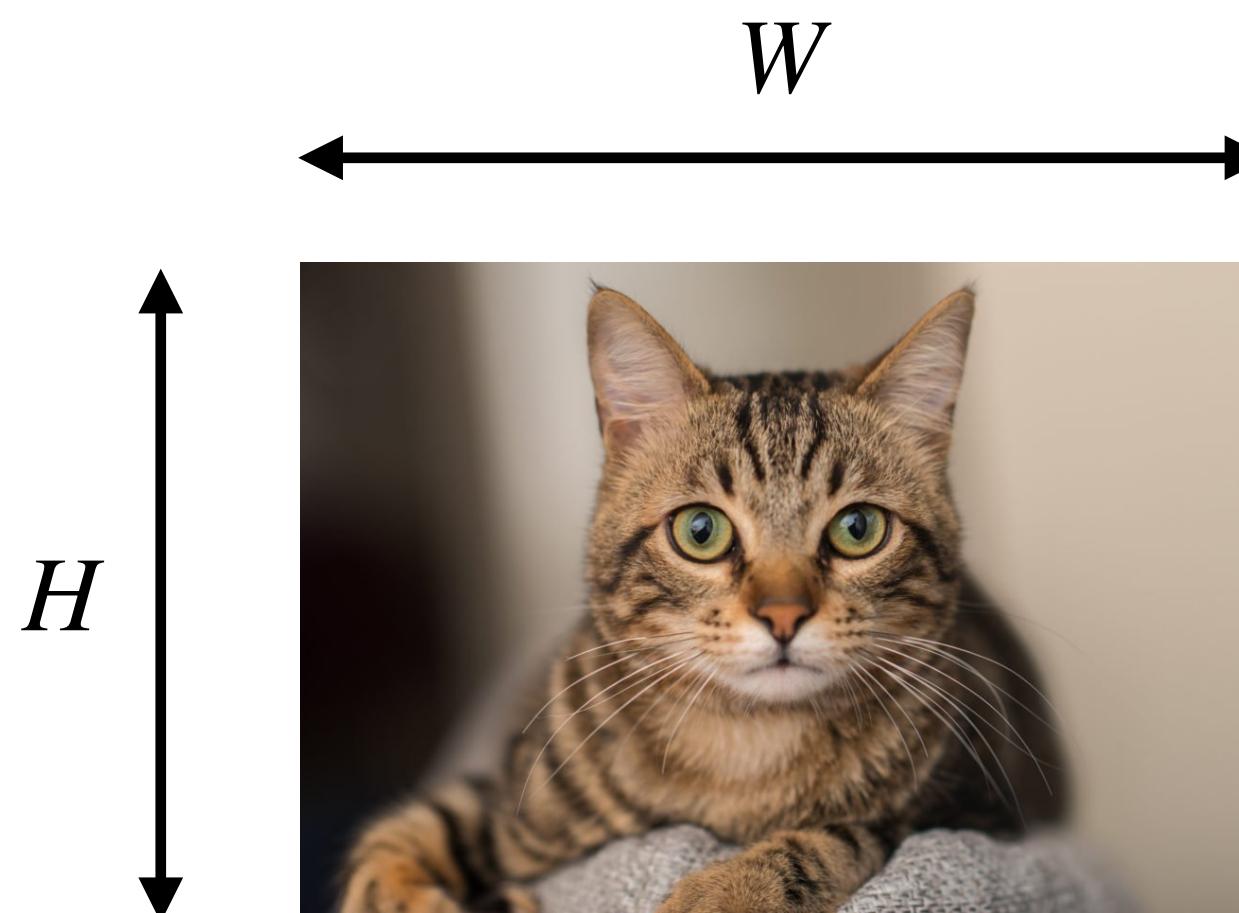
Distribution-based Generative Models

- Goal: learn to generate new data from samples
 - How?
 - To model the data distribution $P(X)$
 - Closed-form analytic solution
 - Exact density estimation via “black-box” deep neural networks
 - Density/distribution approximation

Problems of Exact Density Estimation

- What are the problems of exact density estimation?

- The space of pixels is **huge** $|V| = 256^{H \times W \times 3}$
- The manifold/sub-space of natural images is sparse w.r.t the whole space
$$|V'|/|V| \approx 0$$
- Waste too much model capacity on garbage images/noises



Variational Auto-Encoders (VAEs)

- Learning a (low-dimensional) latent representation
 - The manifold/sub-space of natural images is sparse w.r.t the whole space
$$|V'|/|V| \approx 0$$
 - After down-project to low-dimension space of Z , natural images are **less sparse**

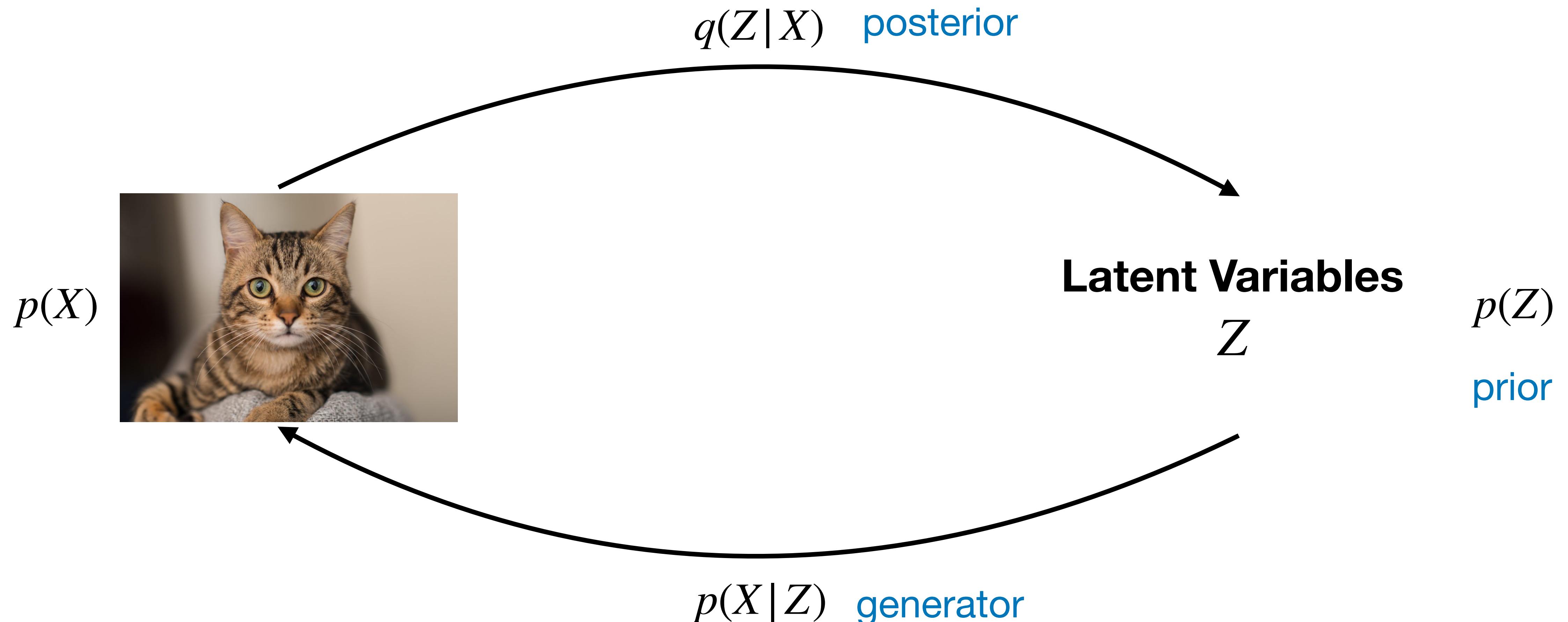


X

low-dimensional

Deep Generative Models w. **Approx.** Density Estimation

- Variational Auto-Encoders (VAEs)
- Diffusion Models



Variational Auto-Encoders

- Low-dimensional latent variable $Z \in \mathbb{R}^d$
- Marginal distribution

$$p(X) = \int_Z p(X|Z)p(Z)dz,$$

- How to compute/approximate the integral?
 - Variational Inference

Variational Inference

$$\underbrace{\log p(X)}_{\text{LL}} = \log \int_Z p(X|Z)p(Z)dz$$

Evidence Lower Bound (ELBO)

$$\geq \underbrace{\mathbb{E}_{q(Z|X)}[\log p(X|Z)]}_{\text{Posterior}} - \underbrace{\text{KL}(q(Z|X) || p(Z))}_{\text{Prior}}$$

ELBO Generator

The diagram illustrates the Evidence Lower Bound (ELBO) as a sum of two terms. The first term is the expectation under the posterior distribution $q(Z|X)$ of the log likelihood $\log p(X|Z)$. The second term is the Kullback-Leibler divergence $\text{KL}(q(Z|X) || p(Z))$. Red arrows point from the labels "Posterior", "Generator", and "Prior" to their respective components in the ELBO formula.

Variational Inference

$$\underbrace{\log p(X)}_{\text{LL}} = \log \int_Z p(X|Z)p(Z)dz$$

Evidence Lower Bound (ELBO)

$$\geq \underbrace{\mathbb{E}_{q(Z|X)}[\log p(X|Z)] - \text{KL}(q(Z|X) || p(Z))}_{\text{ELBO}}$$

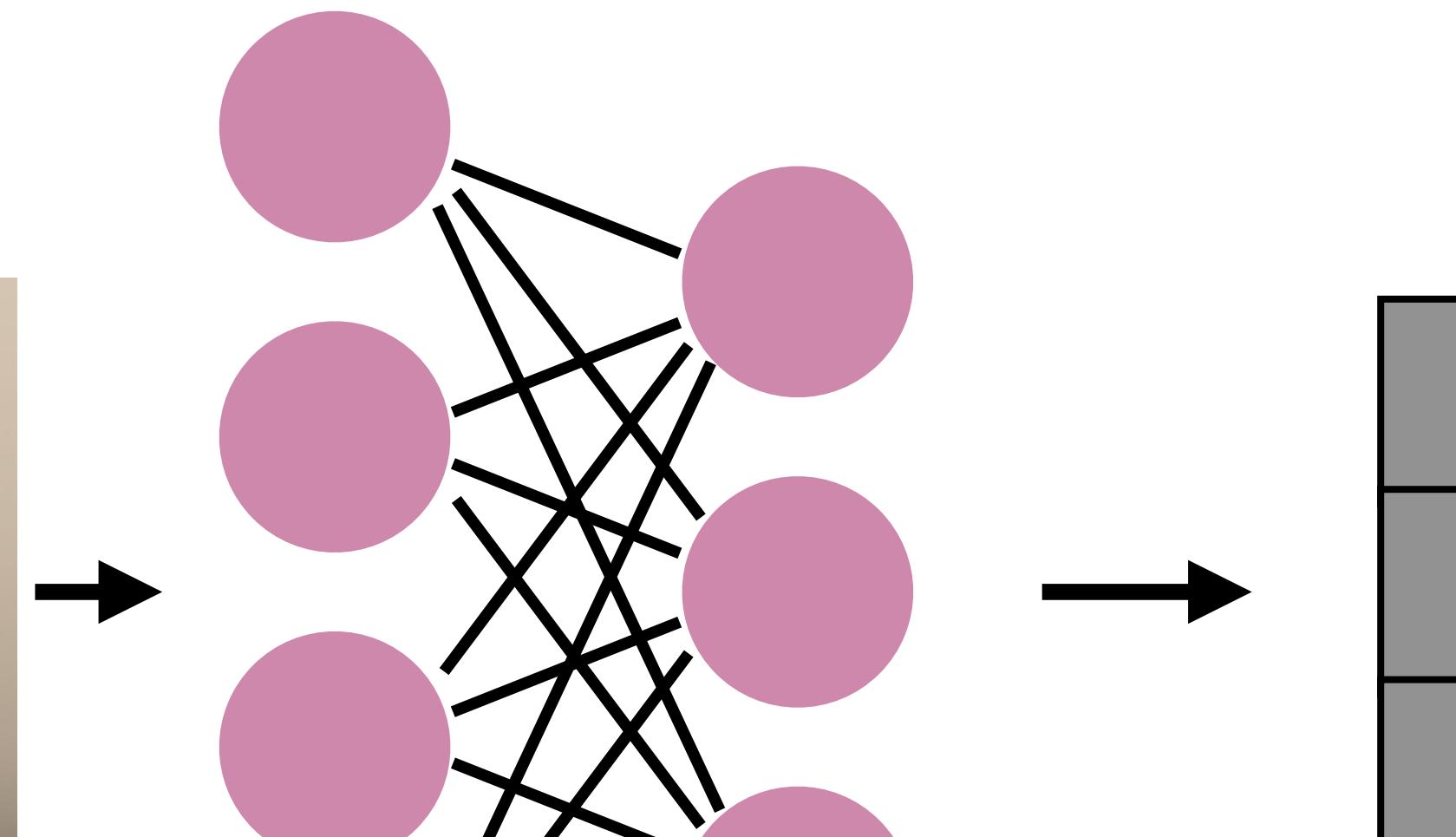
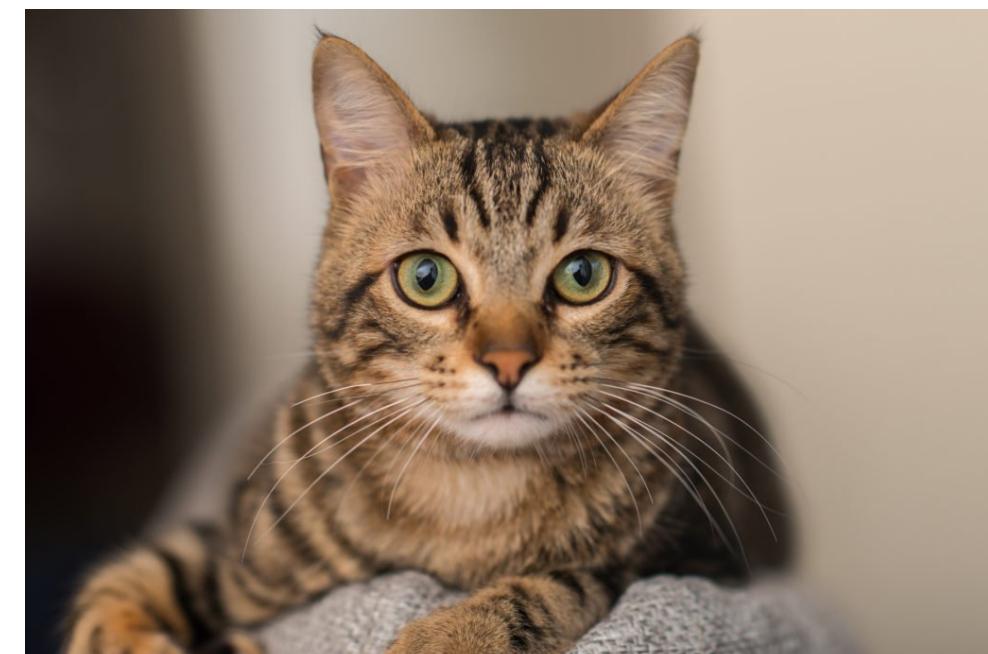
$$= \underbrace{\mathbb{E}_{q(Z|X)}[\log p(X|Z)]}_{\text{Reconstruction}} - \underbrace{\text{KL}(q(Z|X) || p(Z))}_{\text{KL Regularizer}}$$

Variational Inference

Evidence Lower Bound (ELBO)

$$\underbrace{\log p_{\theta}(X)}_{\text{LL}} \geq \underbrace{\mathbb{E}_{q_{\phi}(Z|X)}[\log p_{\theta}(X|Z)] - \text{KL}(q_{\phi}(Z|X) || p_{\theta}(Z))}_{\text{ELBO}}$$

Posterior



X

$q_{\phi}(Z|X)$

Z

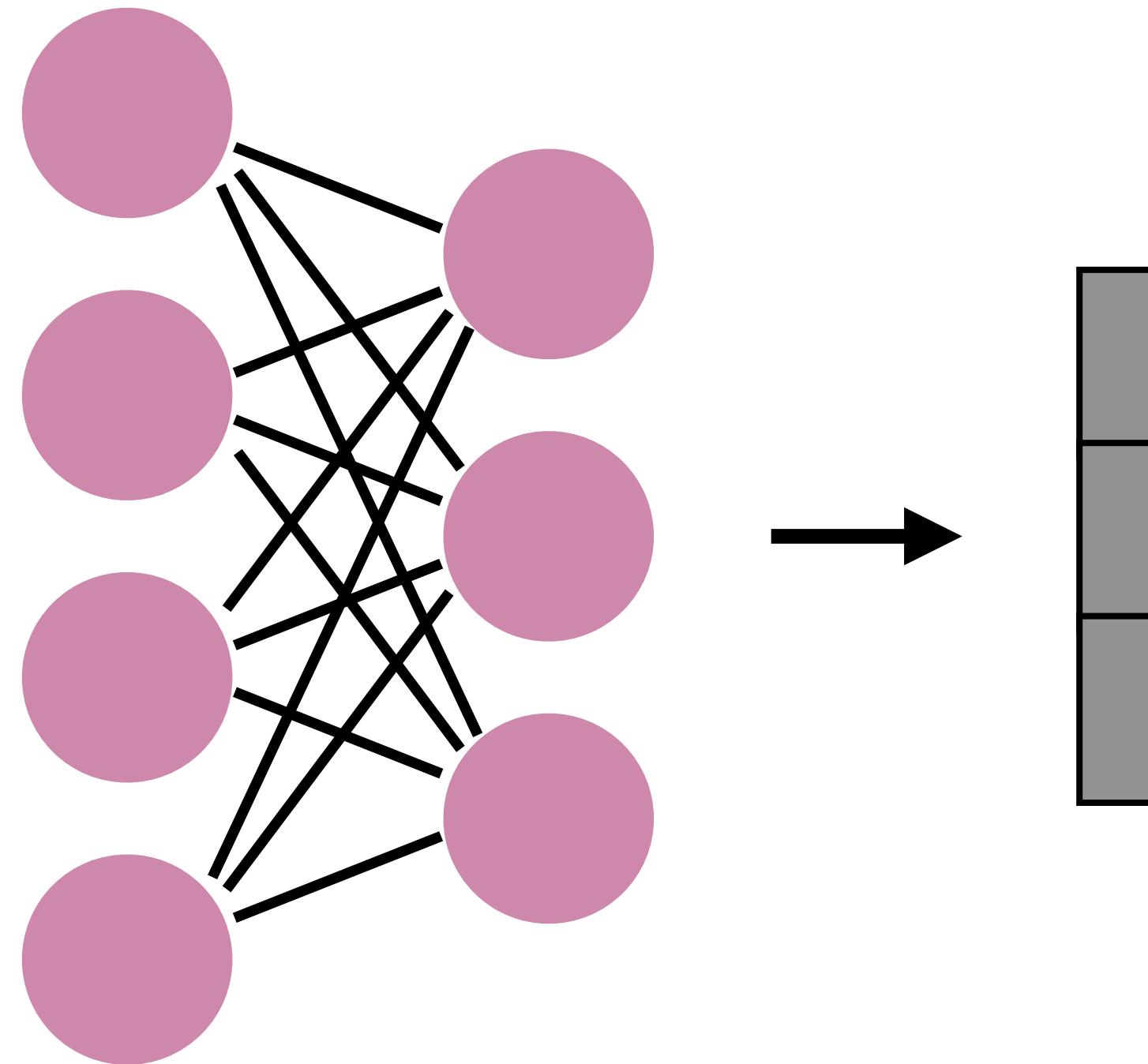
Variational Inference

Evidence Lower Bound (ELBO)

$$\underbrace{\log p_{\theta}(X)}_{\text{LL}} \geq \underbrace{\mathbb{E}_{q_{\phi}(Z|X)}[\log p_{\theta}(X|Z)] - \text{KL}(q_{\phi}(Z|X) || p_{\theta}(Z))}_{\text{ELBO}}$$

Posterior

Prior



X

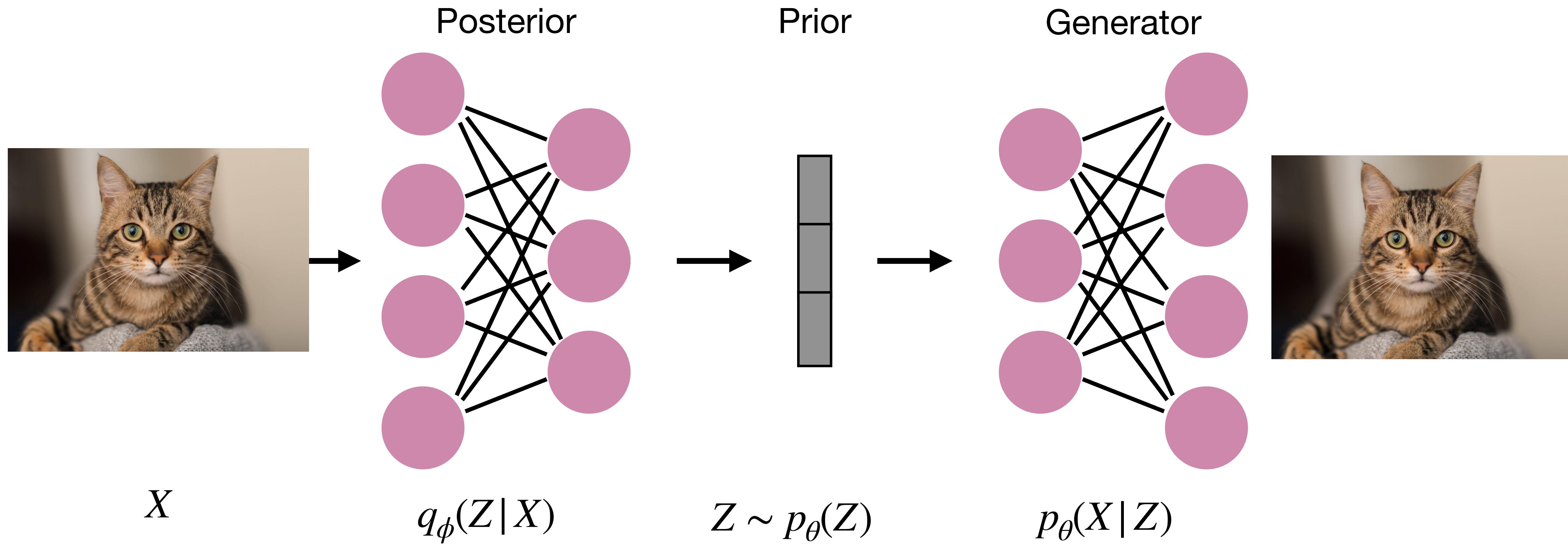
$q_{\phi}(Z|X)$

$Z \sim p_{\theta}(Z)$

Variational Inference

Evidence Lower Bound (ELBO)

$$\underbrace{\log p_{\theta}(X)}_{\text{LL}} \geq \underbrace{\mathbb{E}_{q_{\phi}(Z|X)}[\log p_{\theta}(X|Z)] - \text{KL}(q_{\phi}(Z|X) || p_{\theta}(Z))}_{\text{ELBO}}$$



Variational Auto-Encoders

- Prior

$$P(Z)$$

- Posterior

$$q(Z|X)$$

Depends on tasks:
Gaussian
Auto-regressive model for sequences
Generative flows
...

- Generator

$$P(X|Z)$$

Variational Auto-Encoders

- Prior

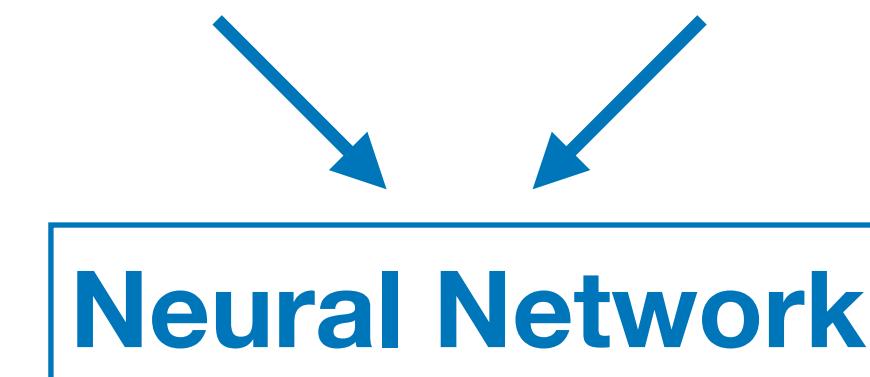
$$P(Z) \sim \mathcal{N}(0, I)$$

Low-dimensional Gaussian

- Posterior

$$q(Z|X) \sim \mathcal{N}(\underline{\mu(X)}, \underline{\sigma^2(X)})$$

Low-dimensional Gaussian



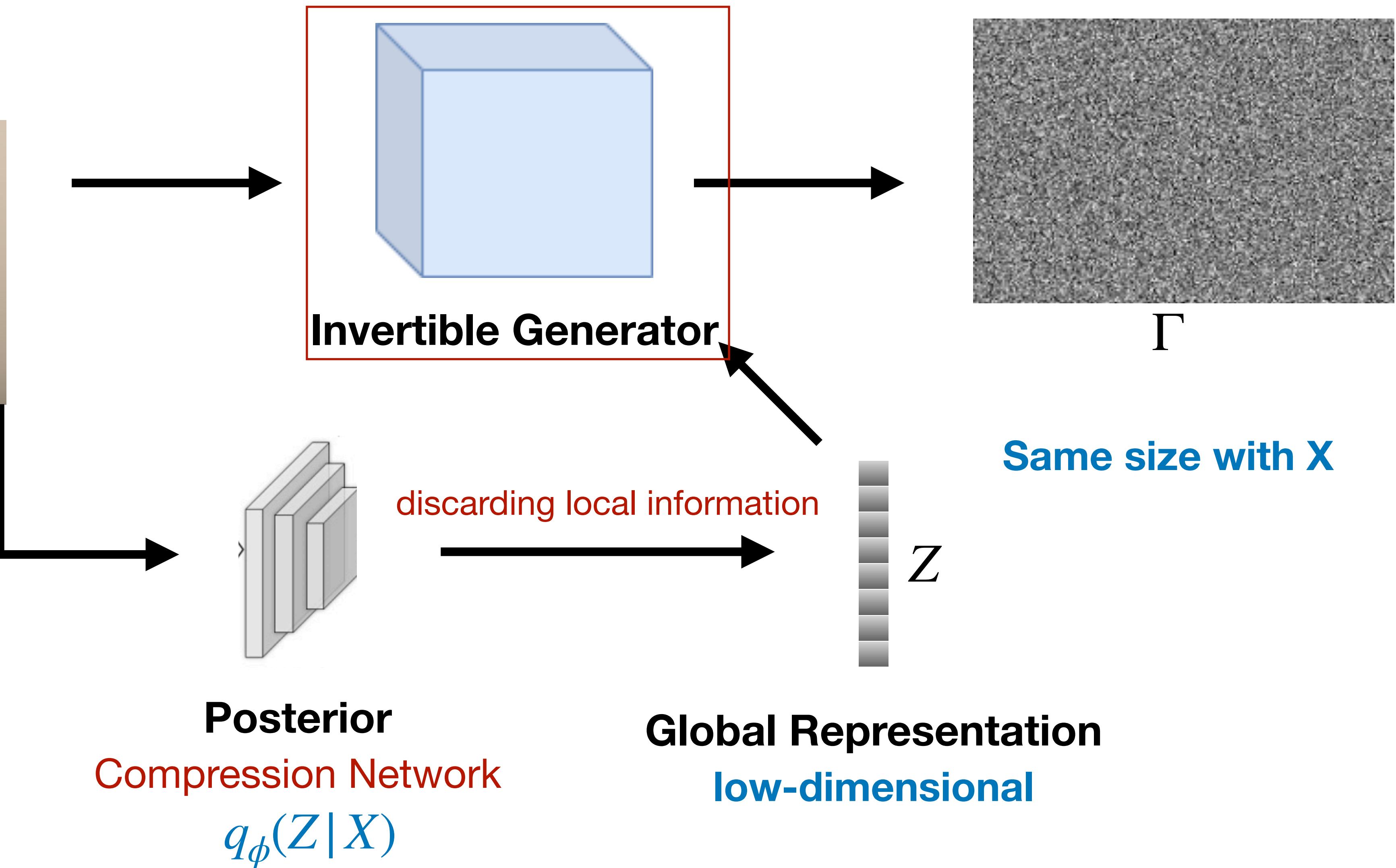
- Generator

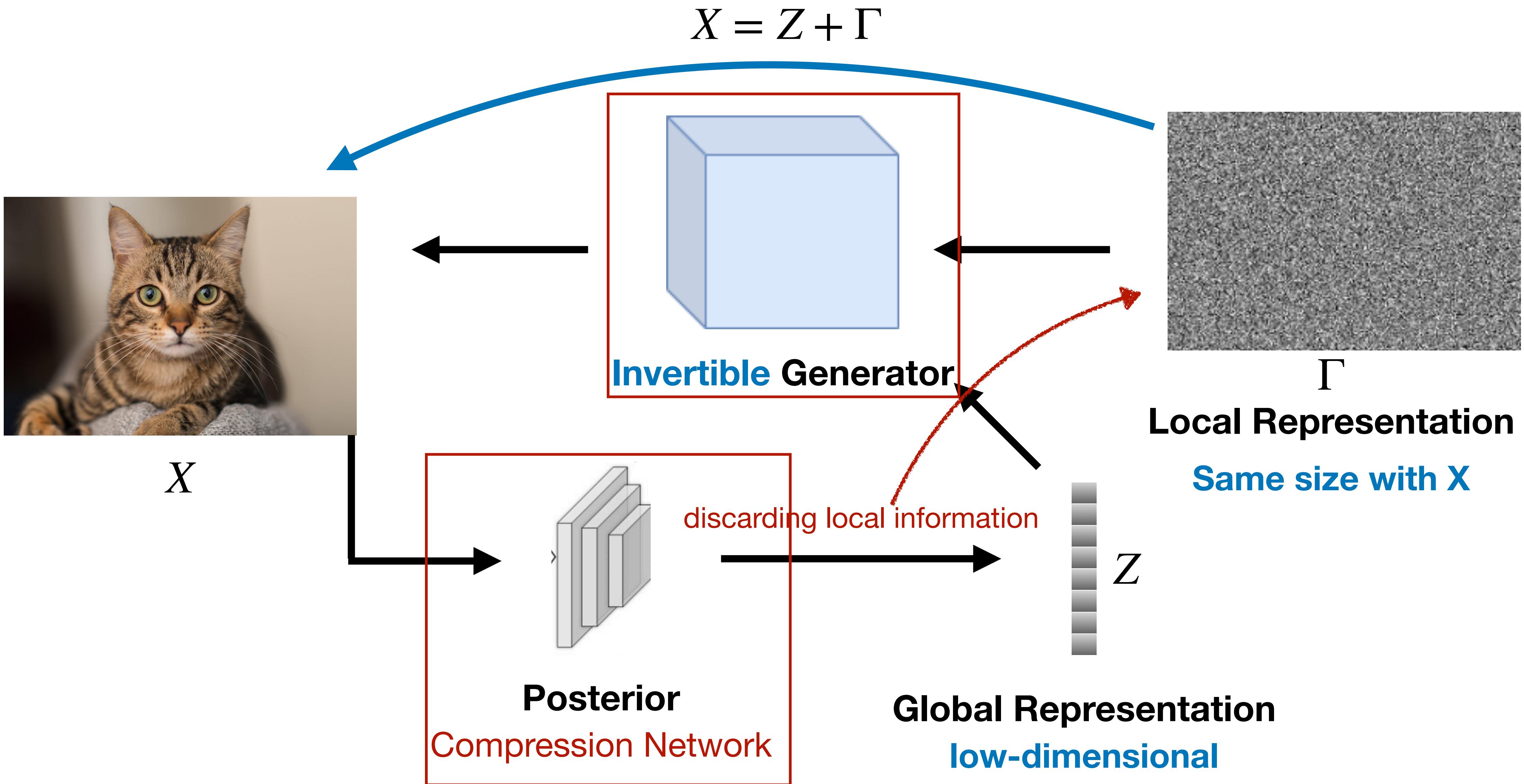
$$P(X|Z)$$

Generative Flow

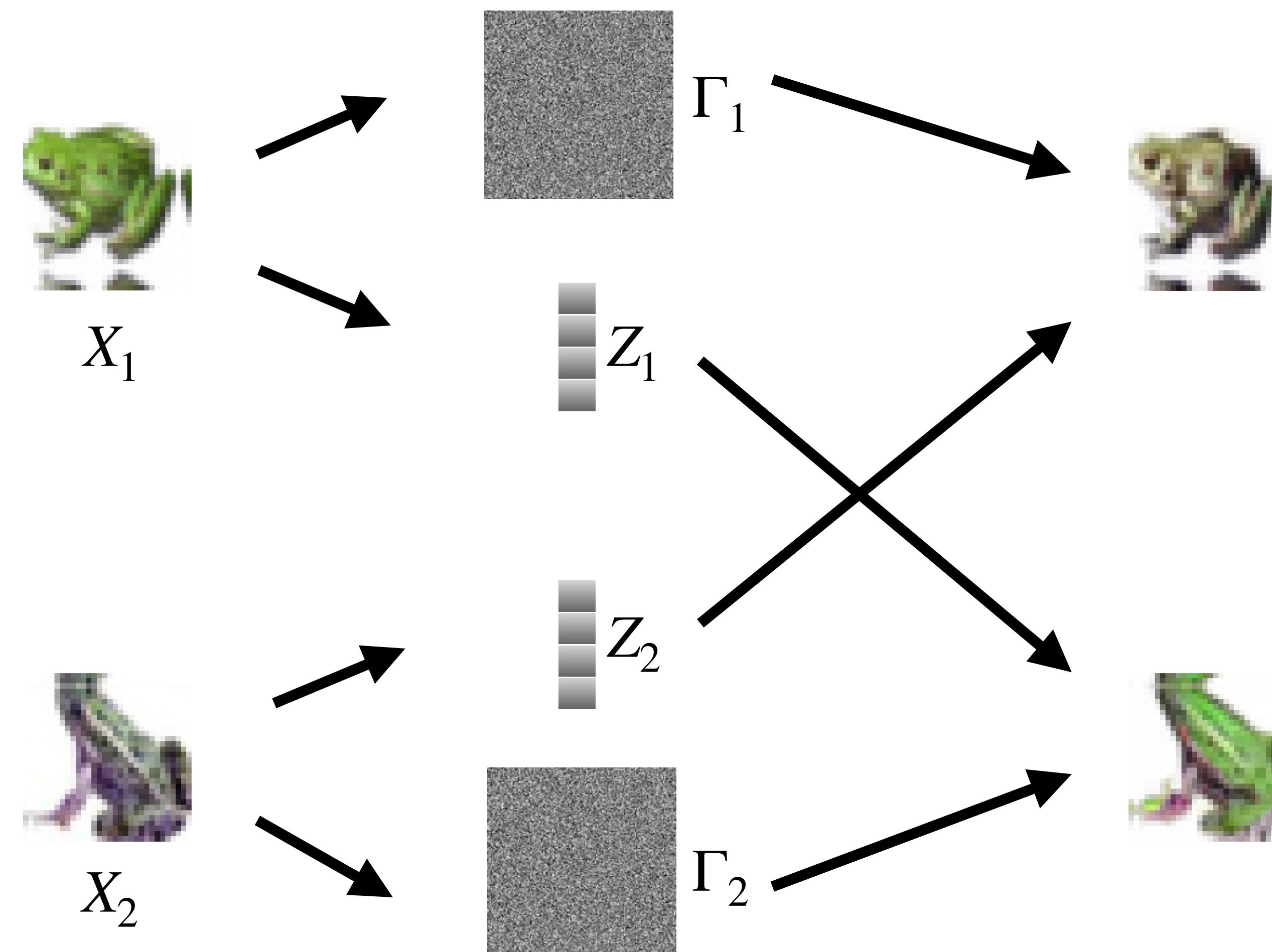


Generative Flow

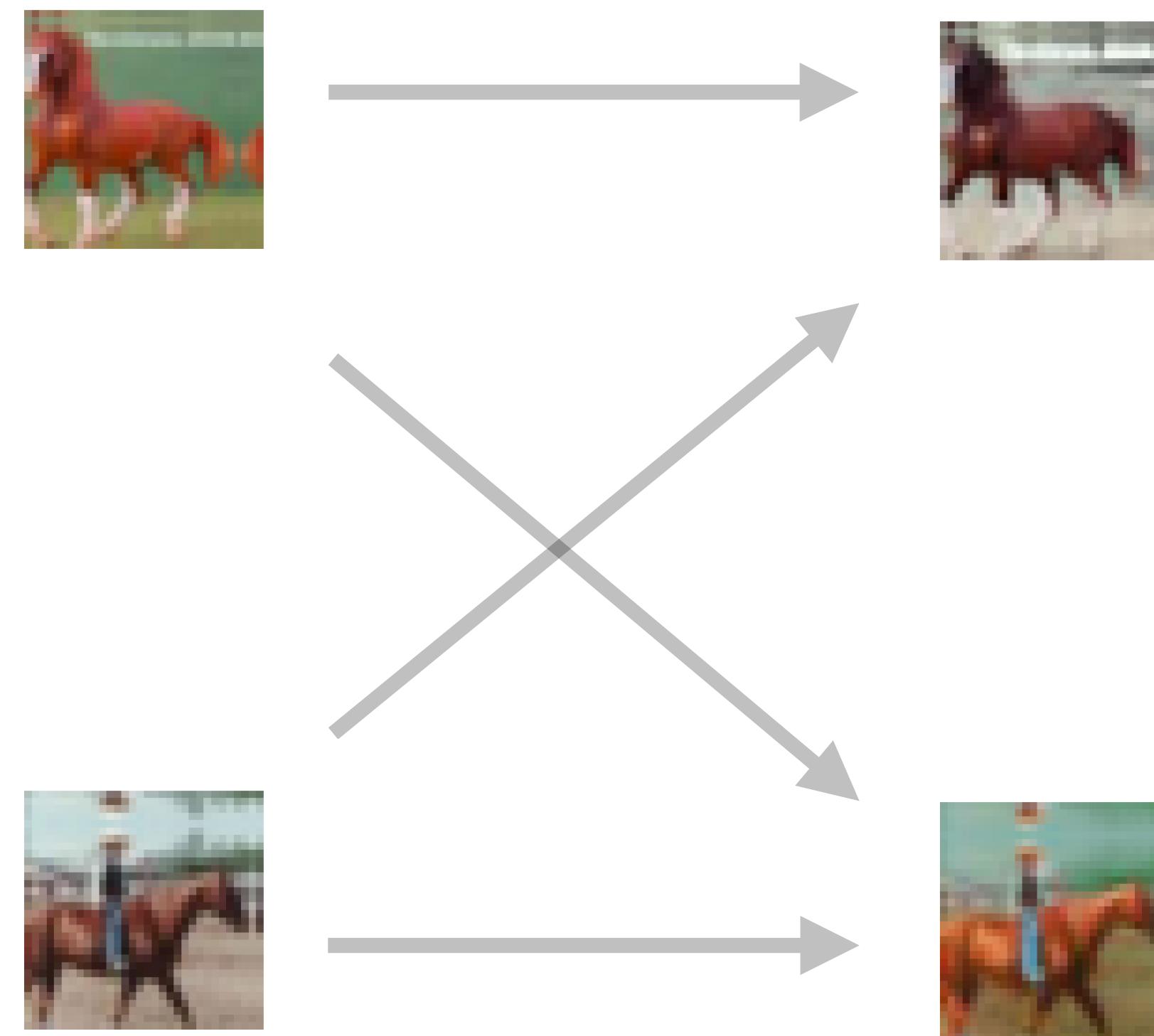




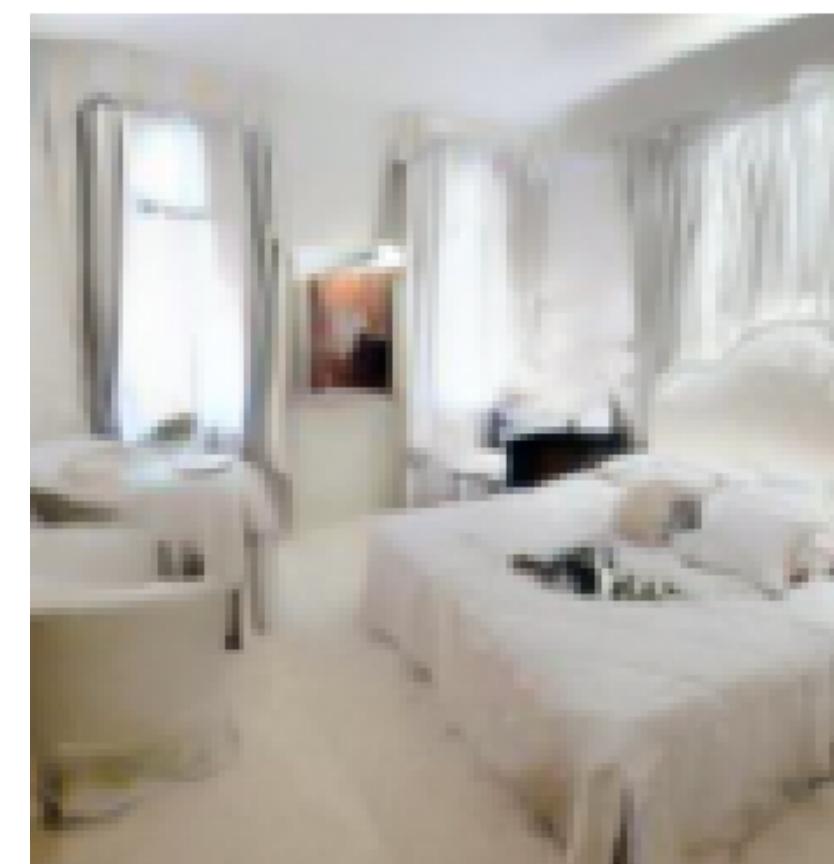
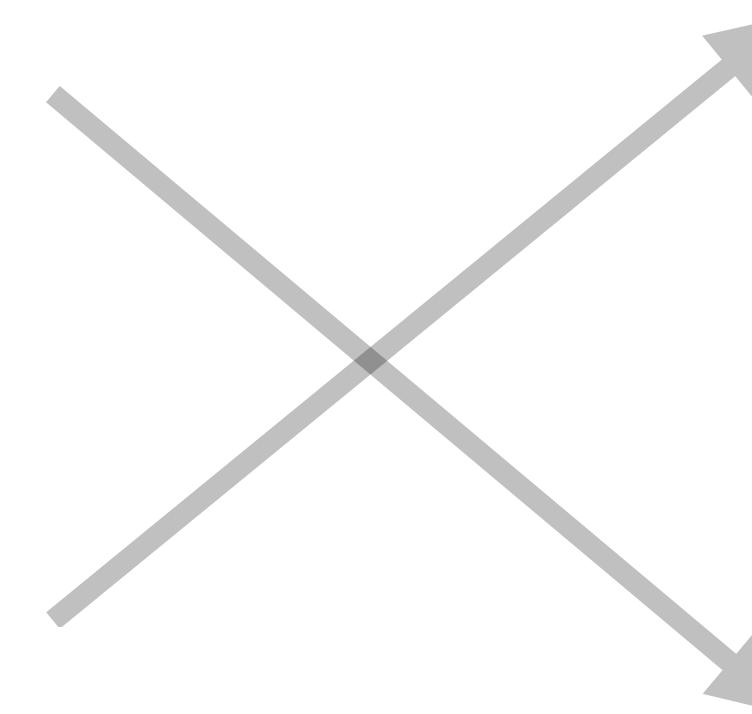
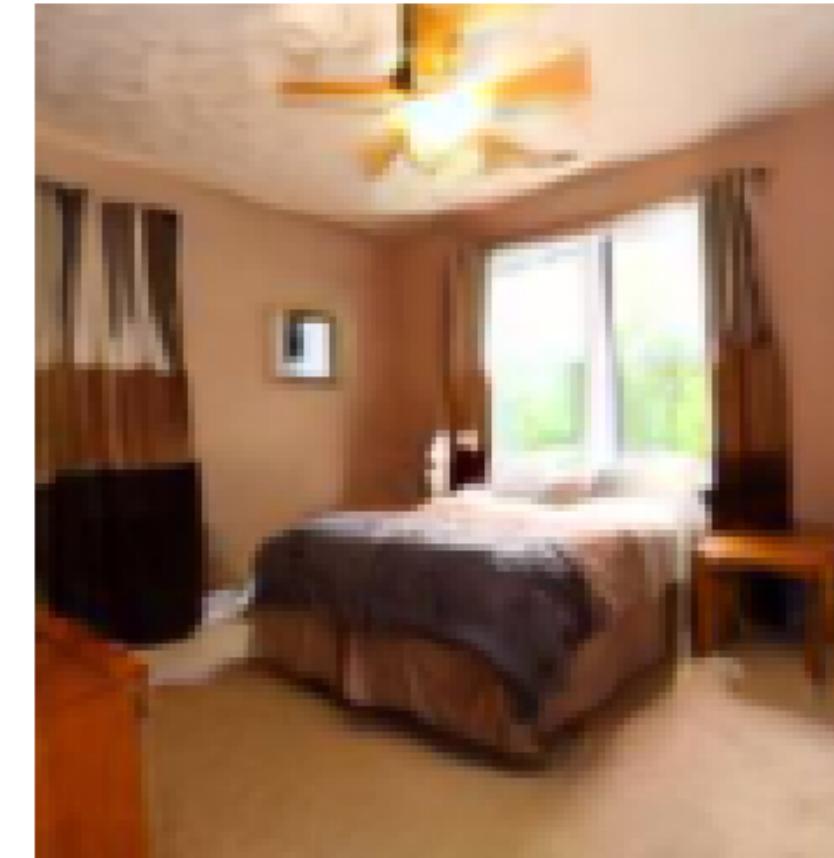
Switch Operation Demo: CIFAR-10



Switch Operation Demo: CIFAR-10



Switch Operation Demo: LSUN-Bedroom



Switch Operation Demo: CelebA-HQ

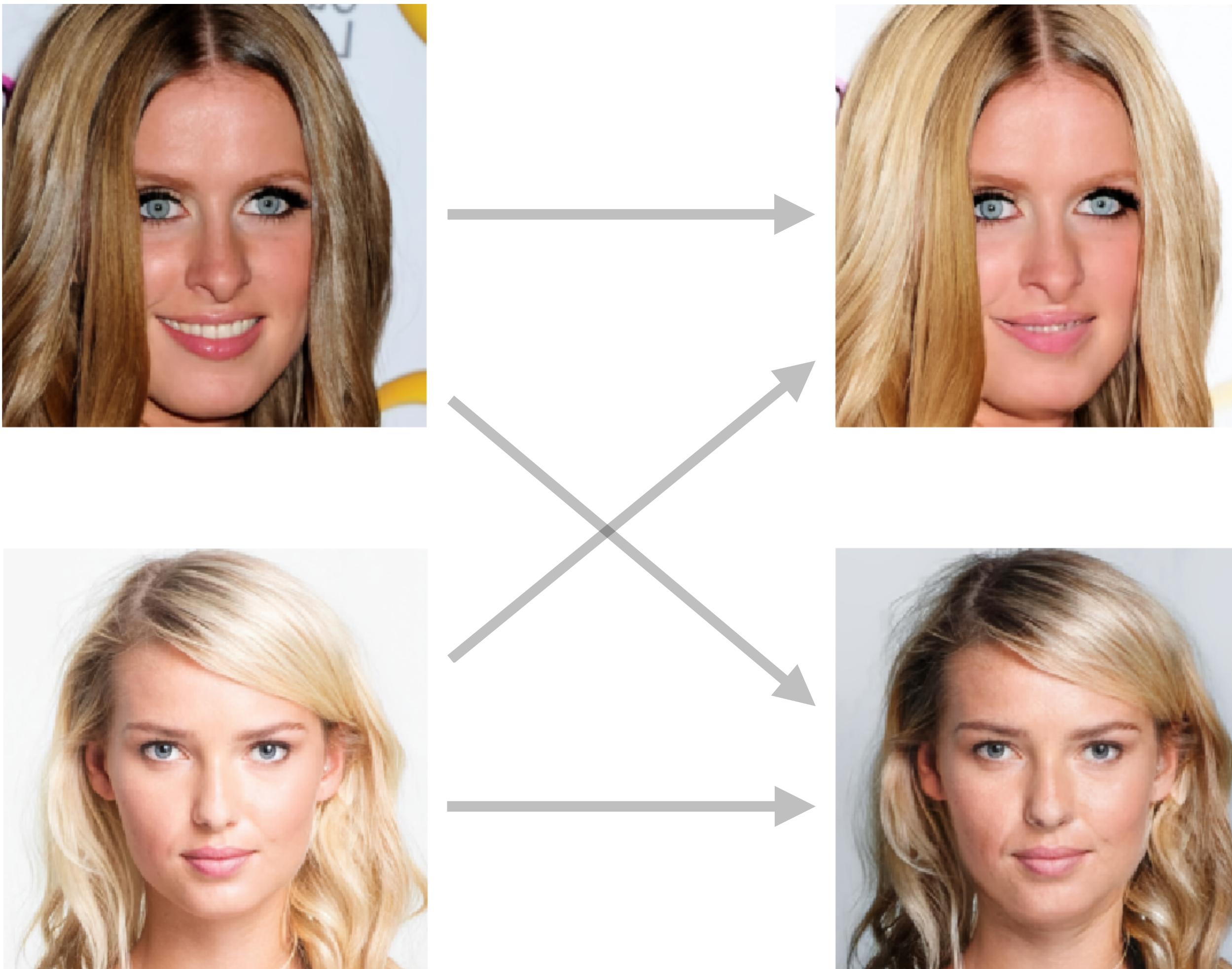
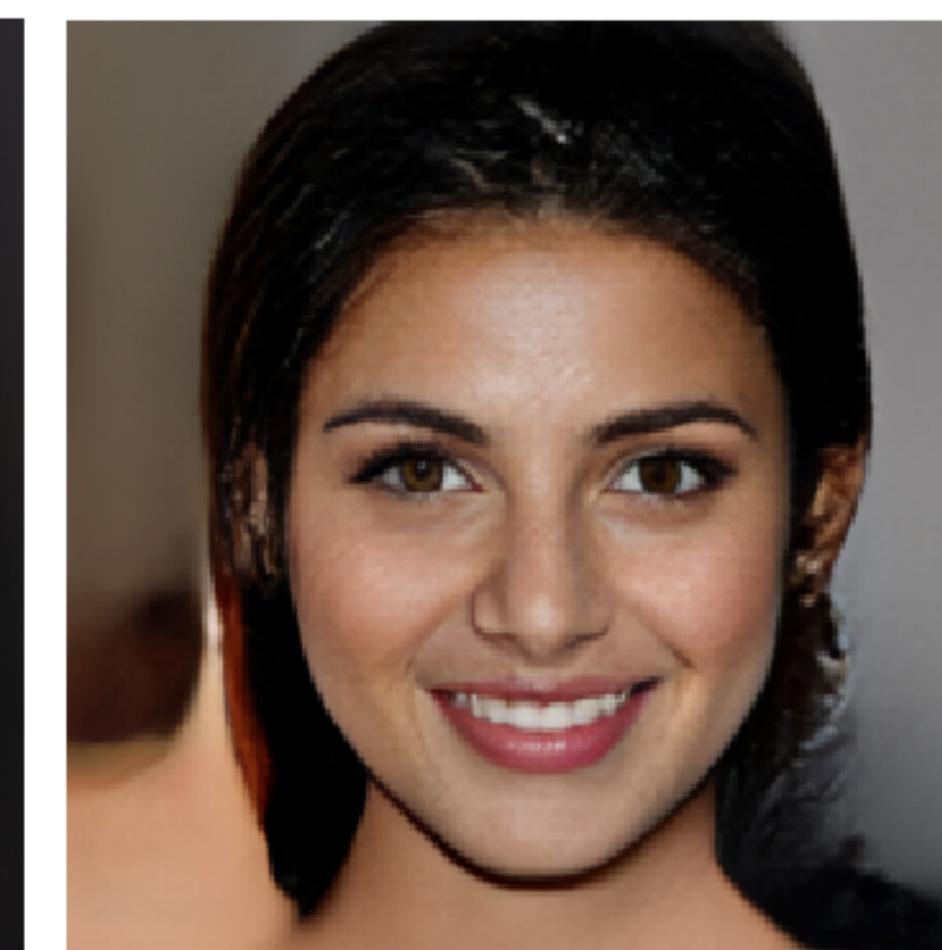
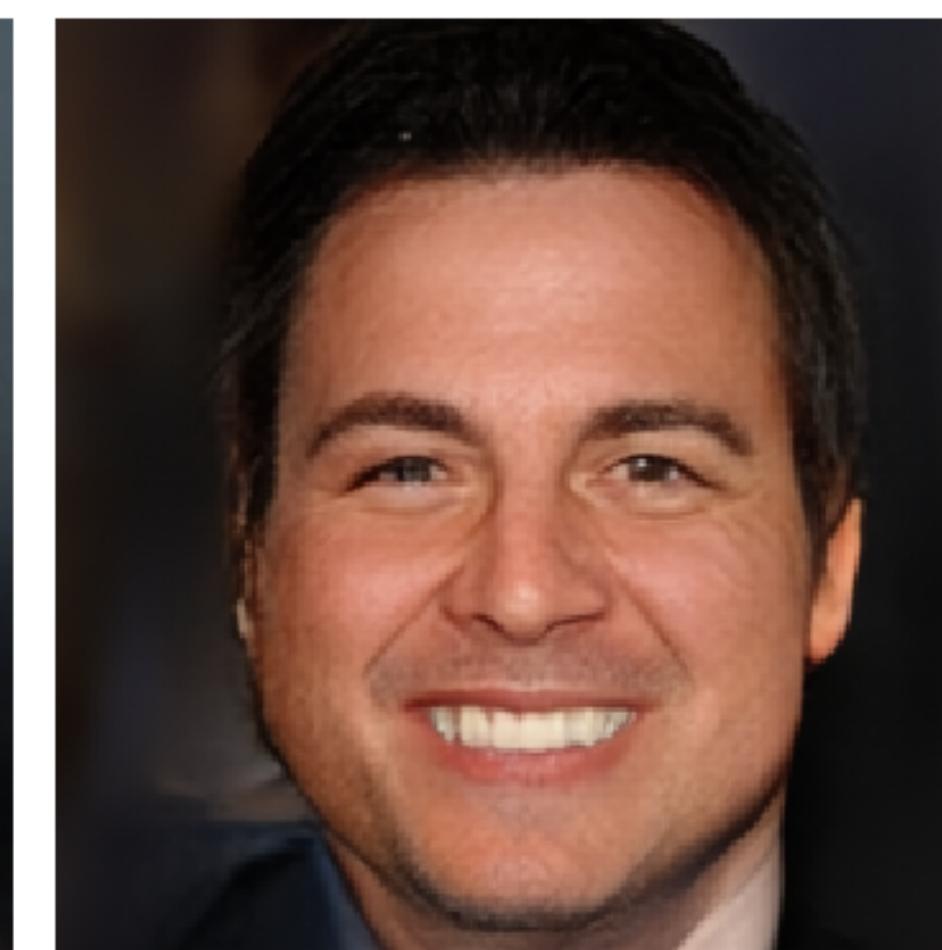
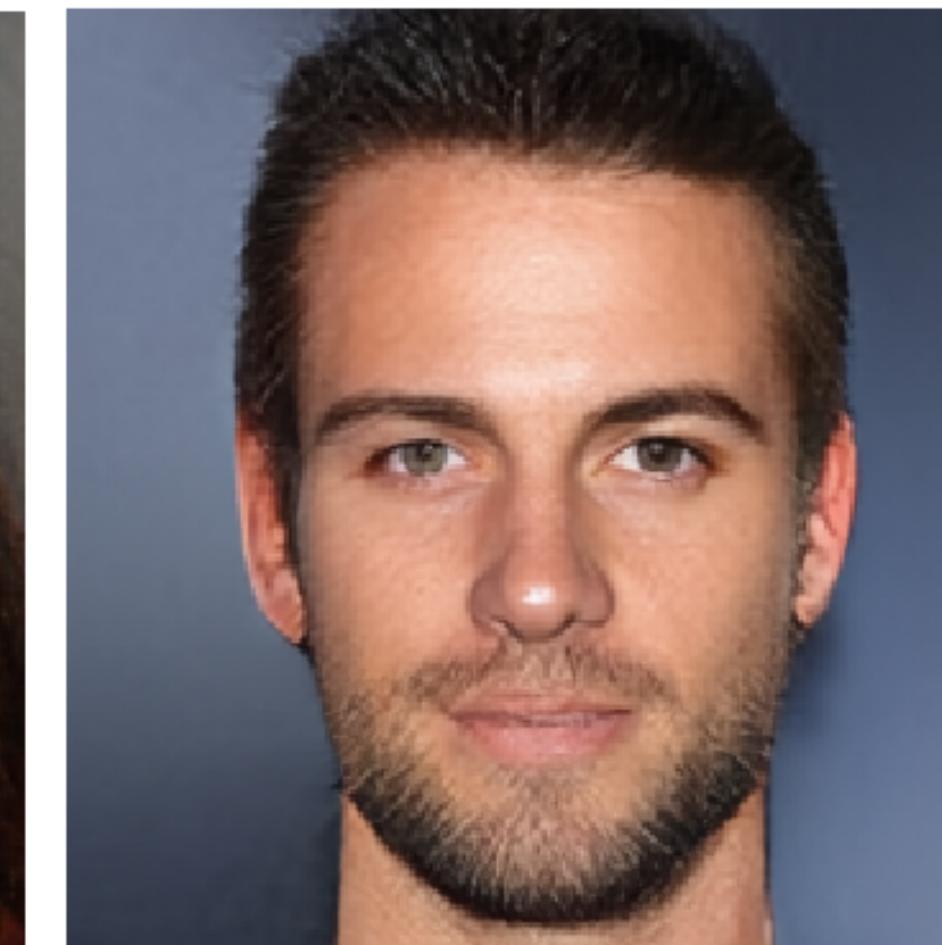
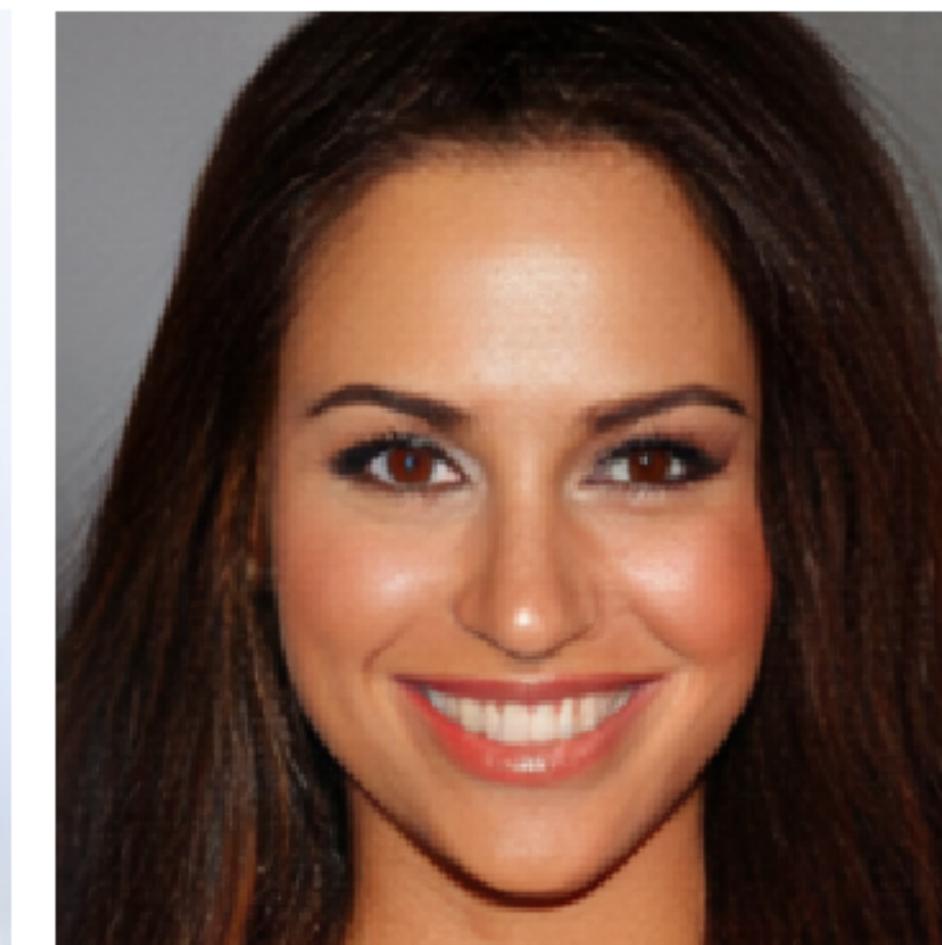


Image Generation

Random Samples



Variational Auto-Encoders

- Prior

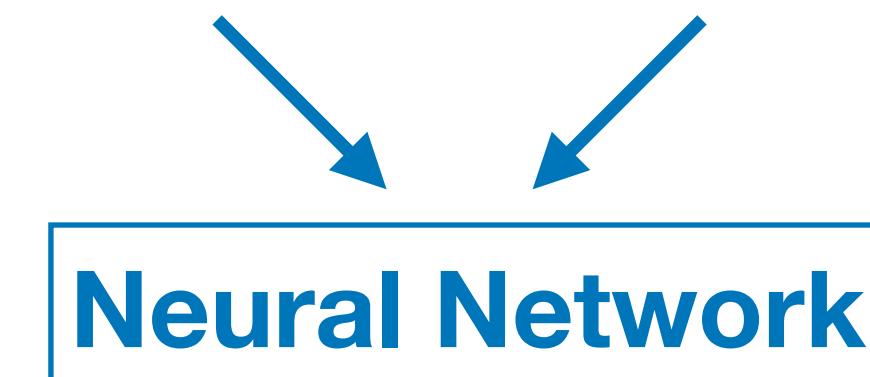
$$P(Z) \sim \mathcal{N}(0, I)$$

Generative Flow

- Posterior

$$q(Z|X) \sim \mathcal{N}(\underline{\mu(X)}, \underline{\sigma^2(X)})$$

High-dimensional Gaussian



- Generator

$$P(X|Z)$$

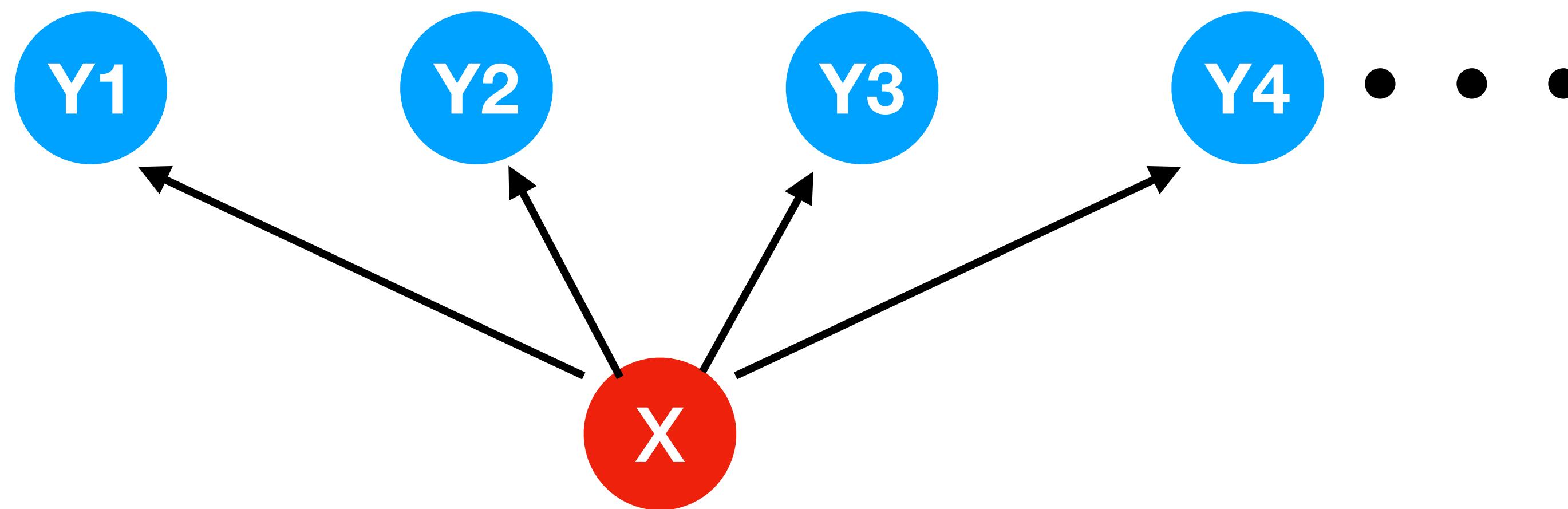
Non-autoregressive model

Non-Autoregressive MT

Non-Autoregressive MT?

- A naïve solution:

$$p_{\theta}(Y|X) = \prod_{t=1}^T p_{\theta}(y_t|X)$$



Too Strong Independent Assumption!

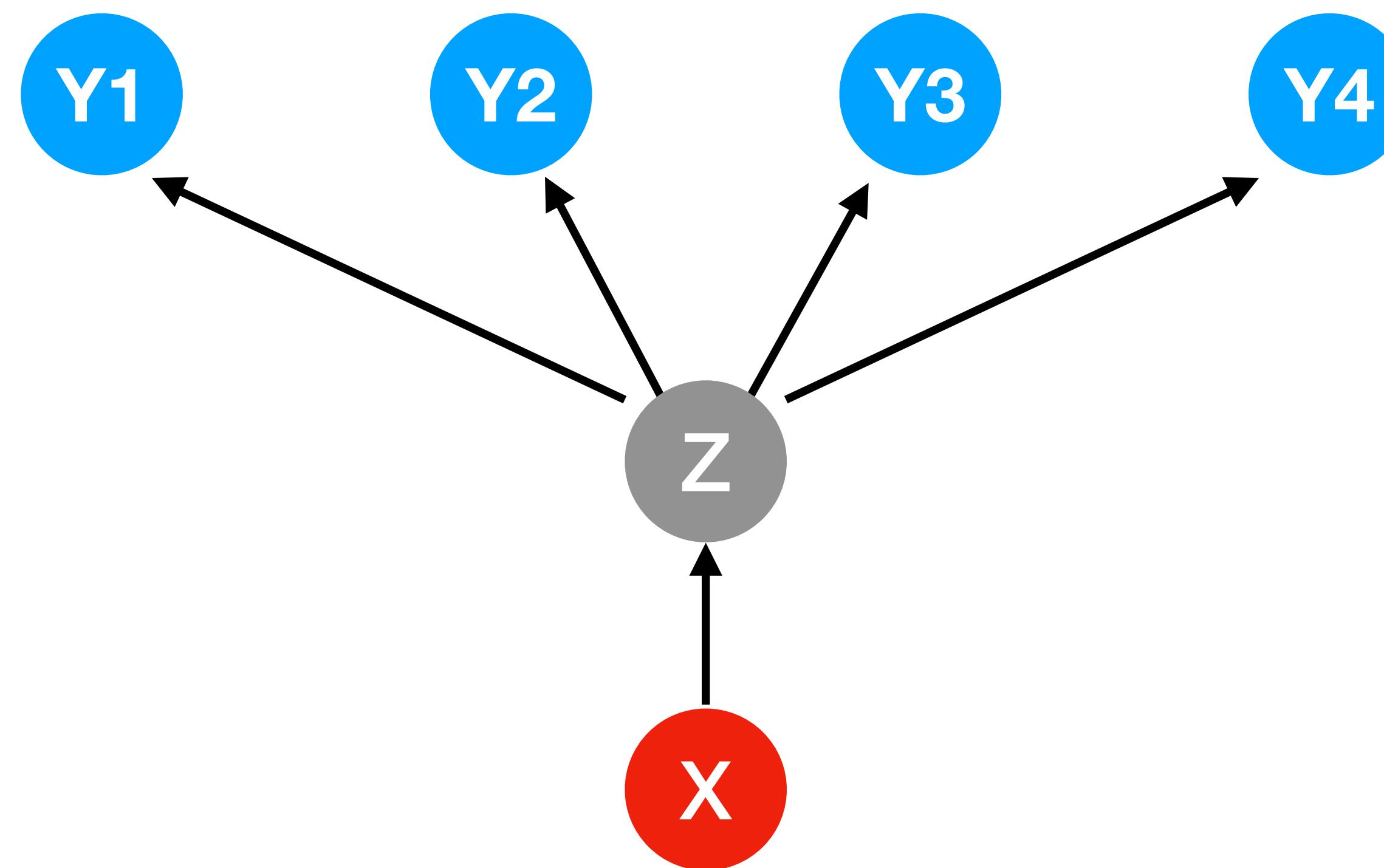
Latent Variable Models

Latent Variable Z

$$p_{\theta}(Y|X) = \int_Z p_{\theta}(Y|Z,X)p_{\theta}(Z|X)dz,$$

Non-Autoregressive

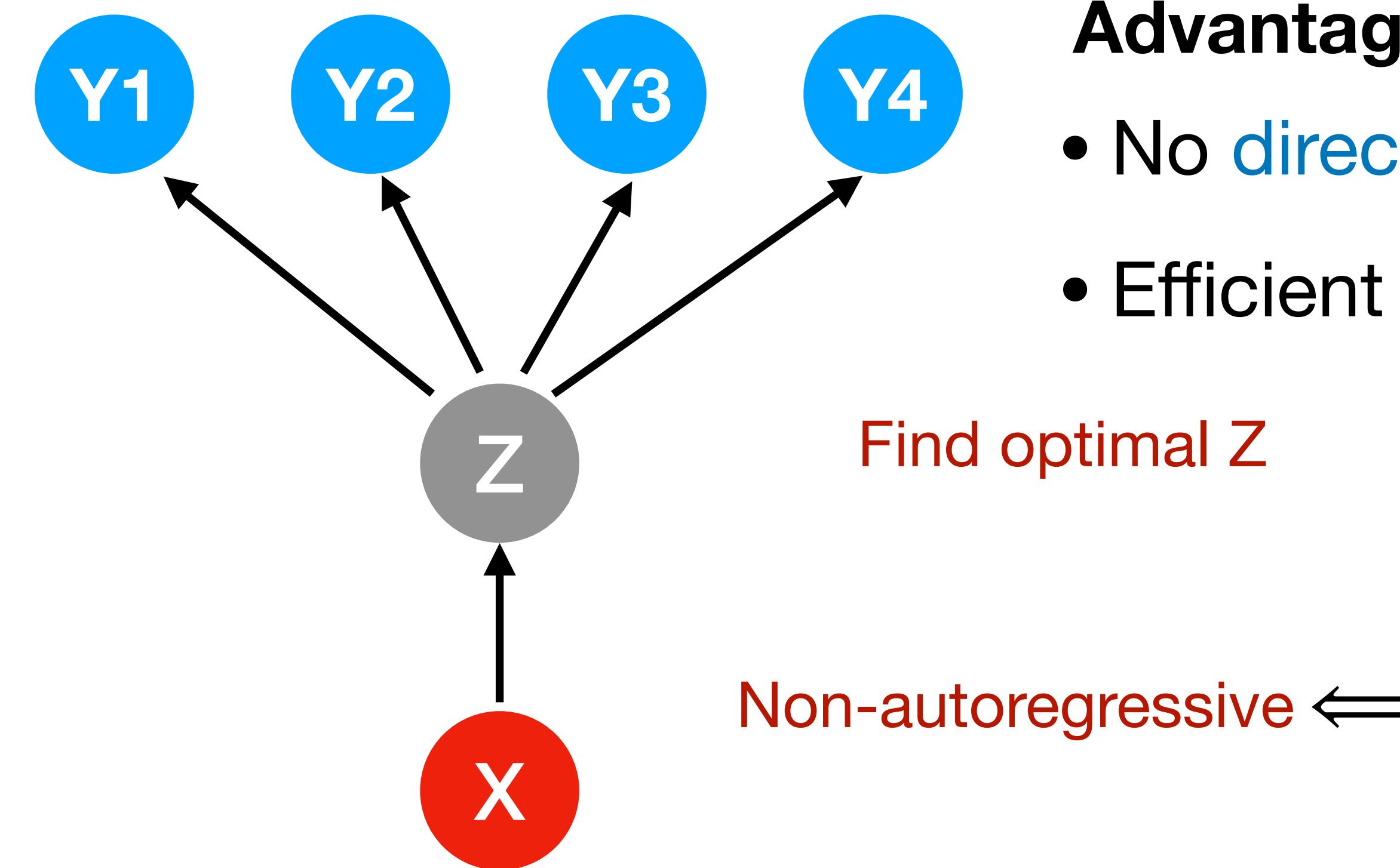
$$p_{\theta}(Y|Z,X) = \prod_{t=1}^T p_{\theta}(y_t|Z,X)$$



Latent Variable Models

Latent Variable Z

Non-Autoregressive



$$p_{\theta}(Y|X) = \int_Z p_{\theta}(Y|Z,X)p_{\theta}(Z|X)dz,$$

$$p_{\theta}(Y|Z,X) = \prod_{t=1}^T p_{\theta}(y_t|Z,X)$$

Advantages:

- No direct independent assumptions between X and Y
- Efficient Decoding:

$$z^* = \operatorname{argmax}_{z \in \mathcal{Z}} p_{\theta}(z|x)$$

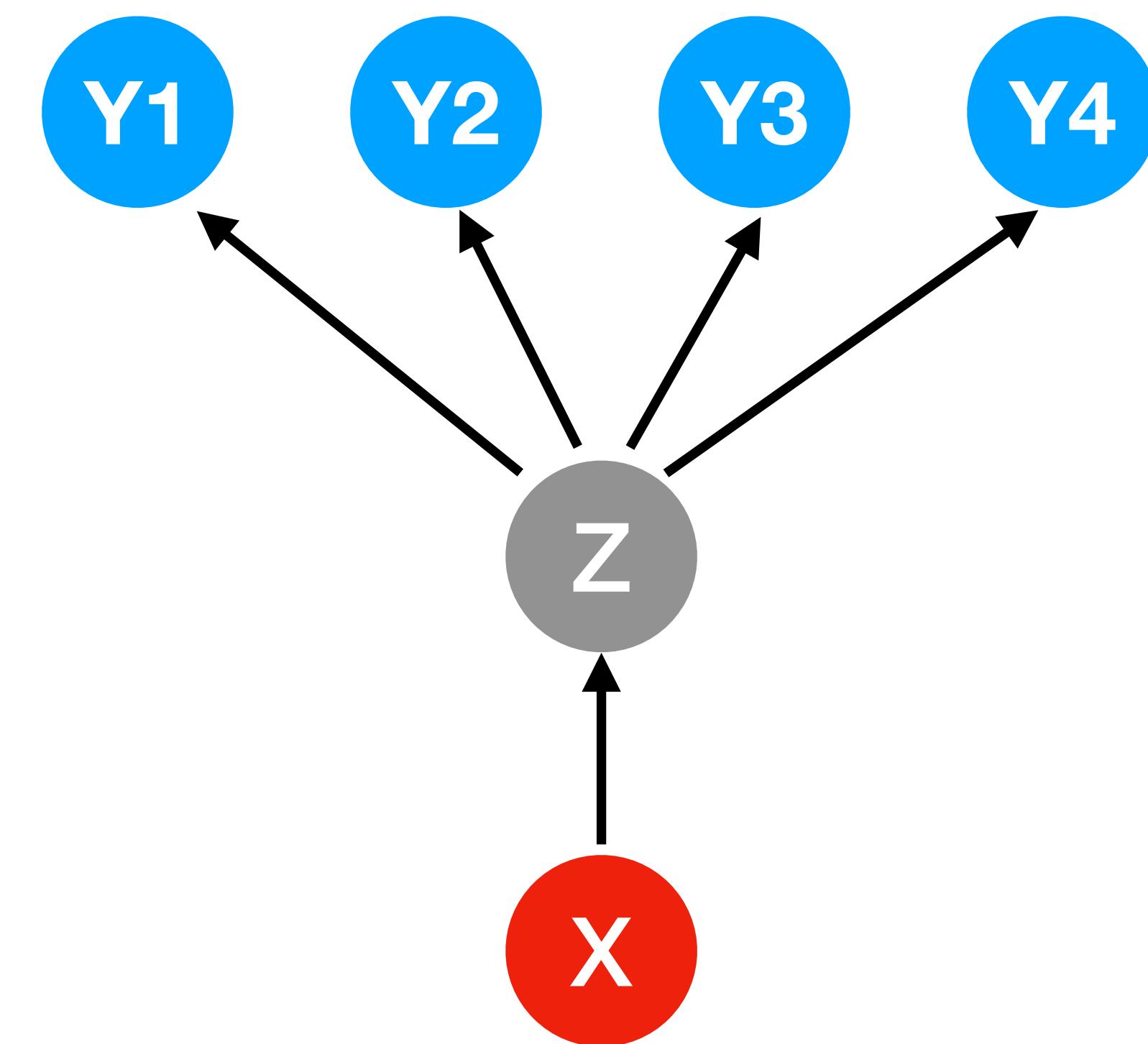
$$y^* = \operatorname{argmax}_{y \in \mathcal{Y}} p_{\theta}(y|z^*, x)$$

$$y_t^* = \operatorname{argmax}_{y_t \in V} p_{\theta}(y_t|z^*, x), \forall t$$

Latent Variable Models

Latent Variable Z

Non-Autoregressive



$$p_{\theta}(Y|X) = \int_Z p_{\theta}(Y|Z,X)p_{\theta}(Z|X)dz,$$

$$p_{\theta}(Y|Z,X) = \prod_{t=1}^T p_{\theta}(y_t|Z,X)$$

Problems:

- How to compute the **integral** of $p_{\theta}(Y|X)$?
 - **Variational Inference**
- Z needs to encode **all the structured dependencies** of Y
 - High-dimensional Z
 - How to model $p_{\theta}(Z|X)$? **Generative Flow**

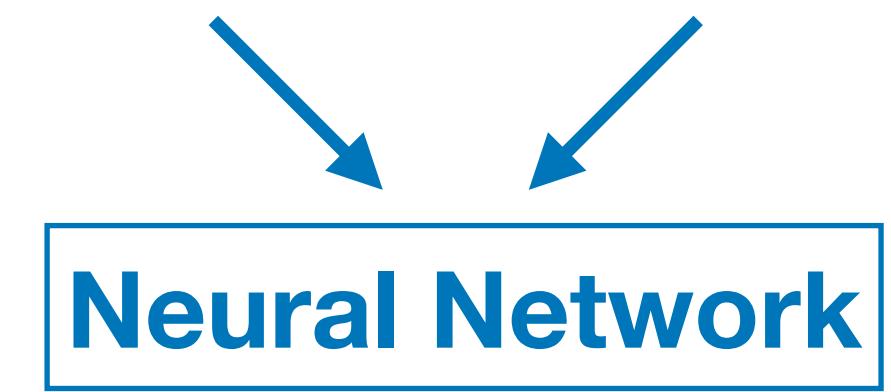
Encoding Structured Dependencies into Z

High-dimensional Latent Variables

$$Z = \{z_1, z_2, \dots, z_T\} \in \mathbf{R}^{d \times T}$$

Posterior: mapping X, Y to Z

$$q_\phi(z_t | Y, X) = \text{Normal}(\underline{\mu_t(Y, X)}, \underline{\sigma_t^2(Y, X)})$$



Encoding Structured Dependencies into Z

High-dimensional Latent Variables

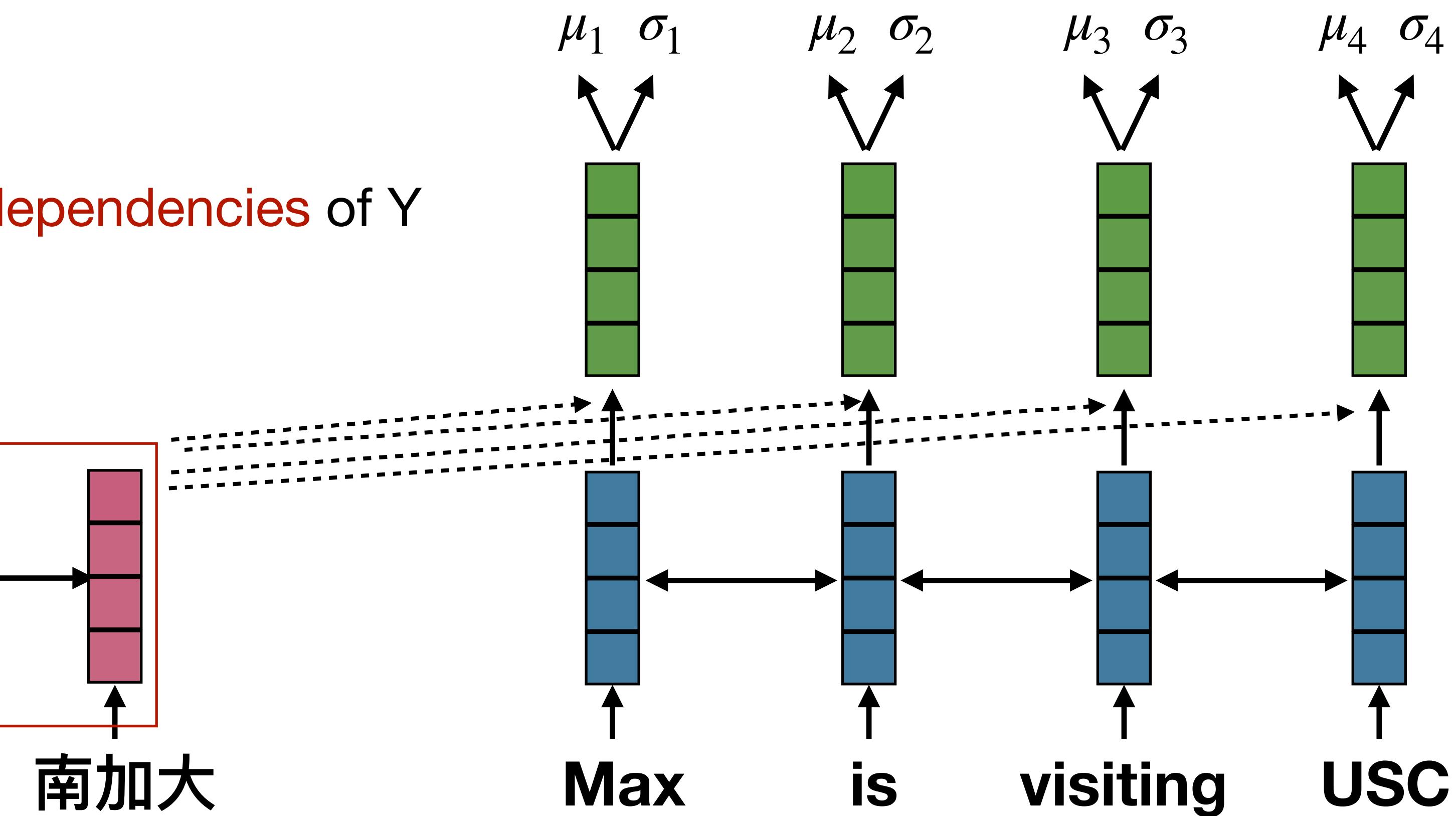
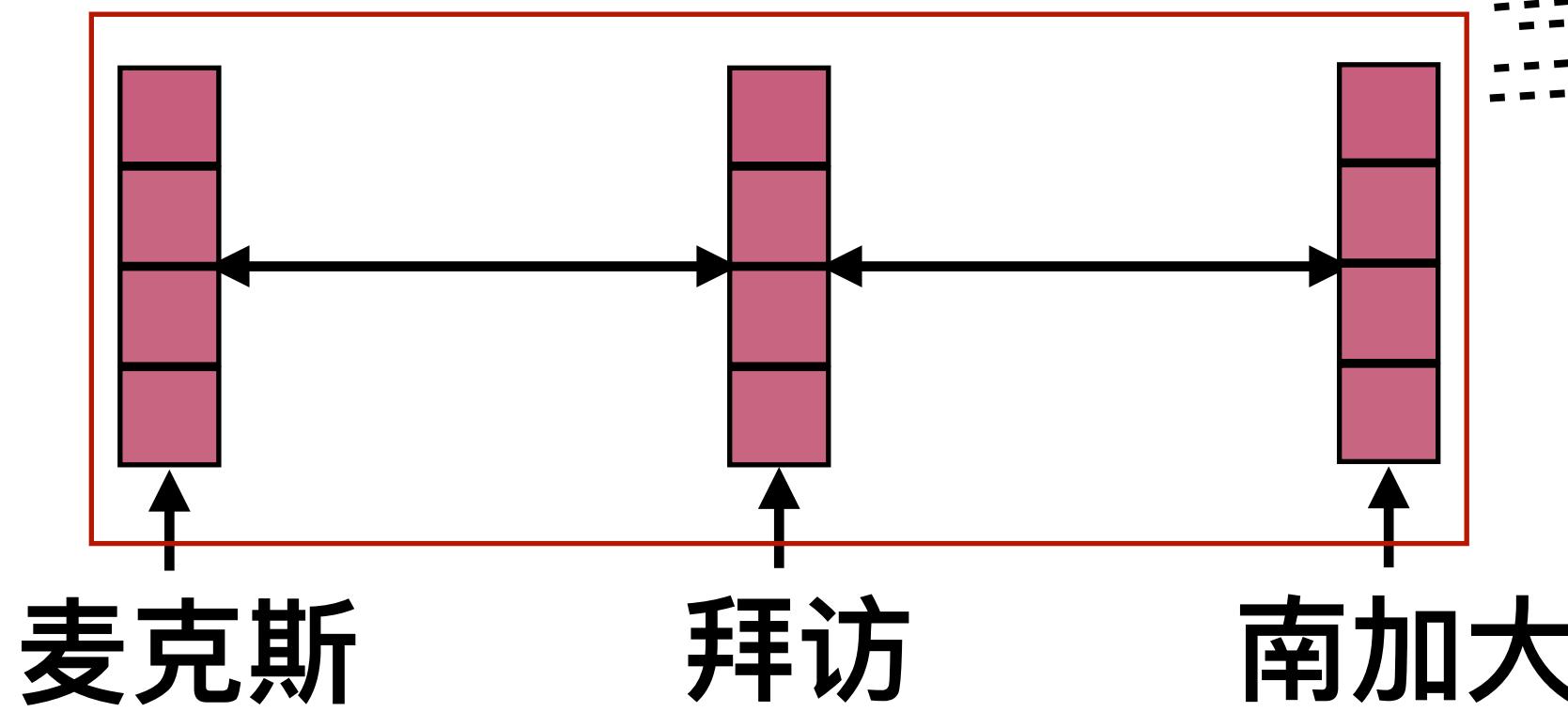
Posterior: mapping X, Y to Z

$$Z = \{z_1, z_2, \dots, z_T\} \in \mathbb{R}^{d \times T}$$

$$q_\phi(z_t | Y, X) = \text{Normal}(\mu_t(Y, X), \sigma_t^2(Y, X))$$

Assumption:

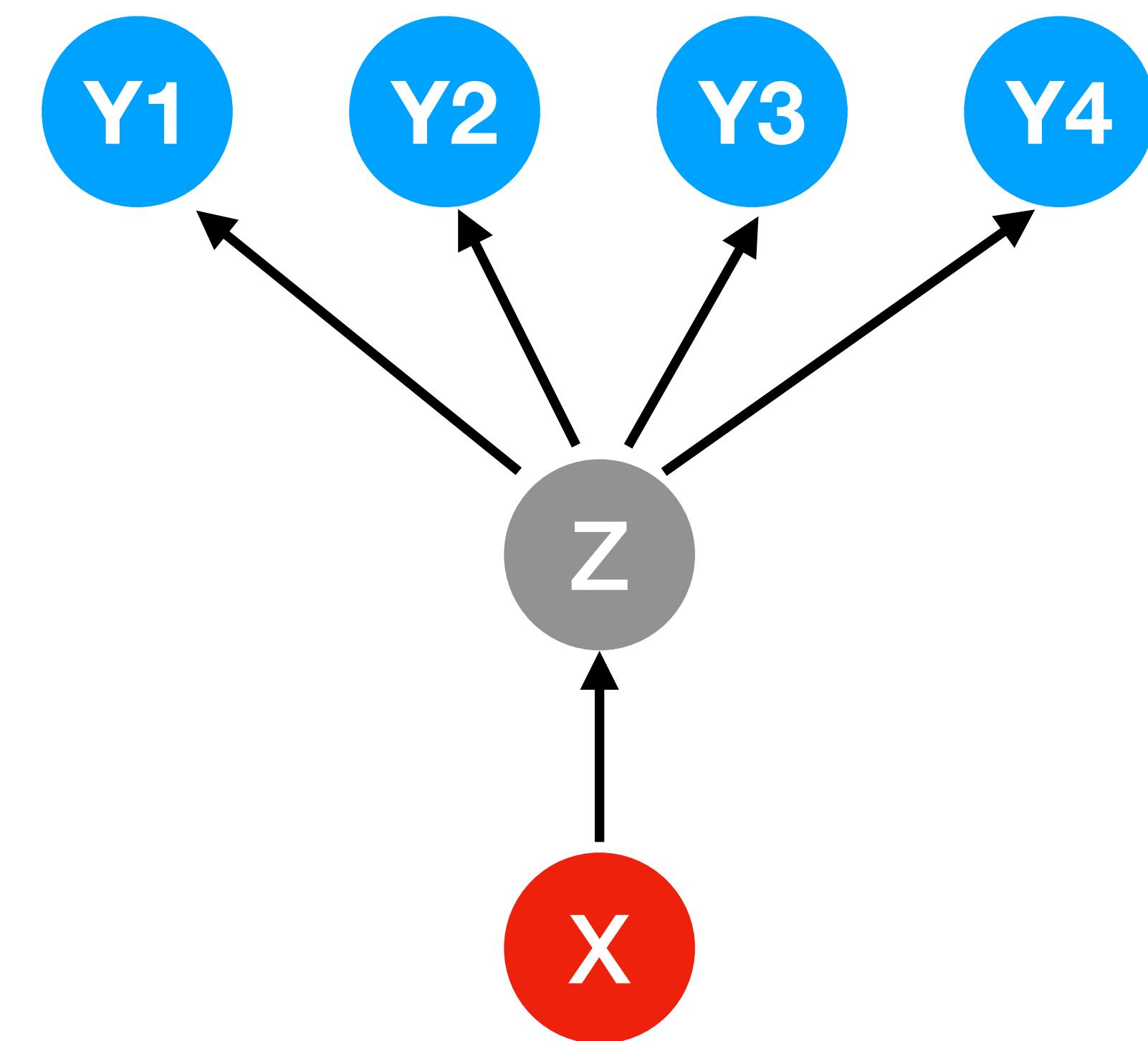
Z encodes all the structured dependencies of Y



Latent Variable Models

Latent Variable Z

Non-Autoregressive



$$p_{\theta}(Y|X) = \int_Z p_{\theta}(Y|Z,X)p_{\theta}(Z|X)dz,$$

$$p_{\theta}(Y|Z,X) = \prod_{t=1}^T p_{\theta}(y_t|Z,X)$$

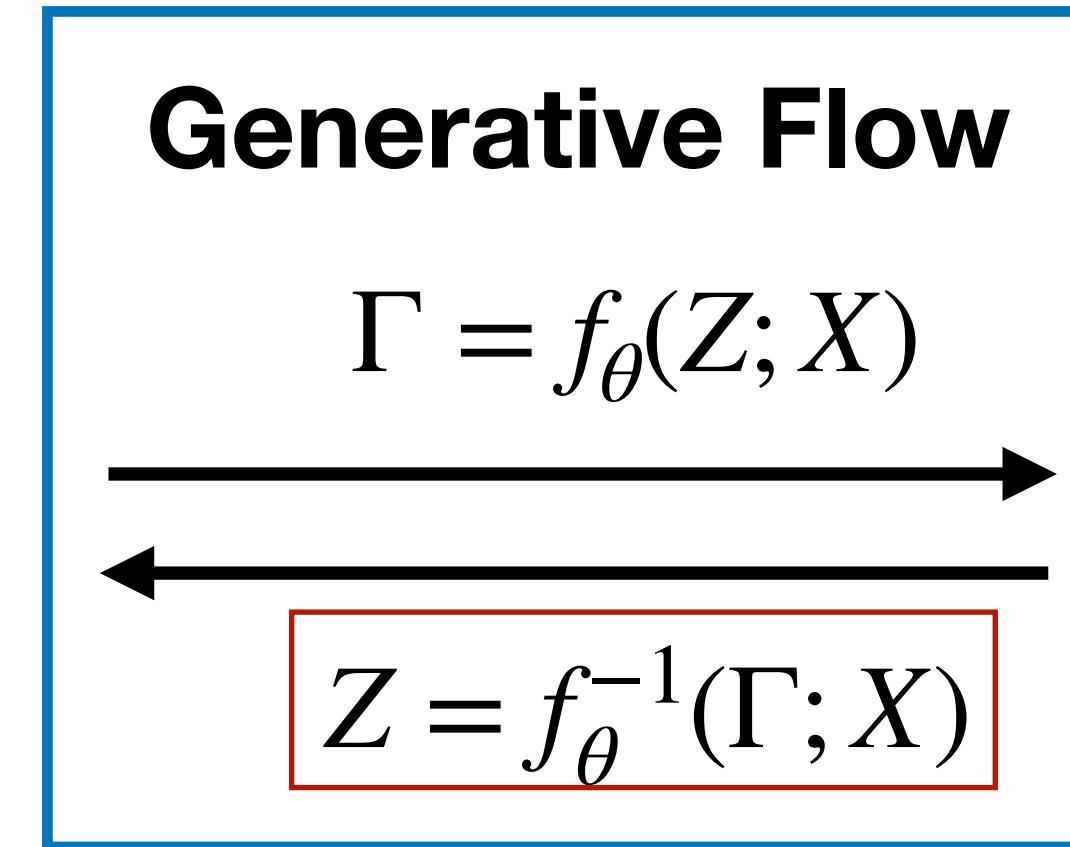
Problems:

- How to compute the **integral** of $p_{\theta}(Y|X)$?
 - **Variational Inference**
- Z needs to encode **all the structured dependencies** of Y
 - High-dimensional Z
 - How to model $p_{\theta}(Z|X)$? **Generative Flow**

Generative Flow for Prior $p_\theta(Z|X)$

- Prior $p_\theta(Z|X)$

$$Z = \{z_1, z_2, \dots, z_T\} \in \mathbb{R}^{d \times T}$$



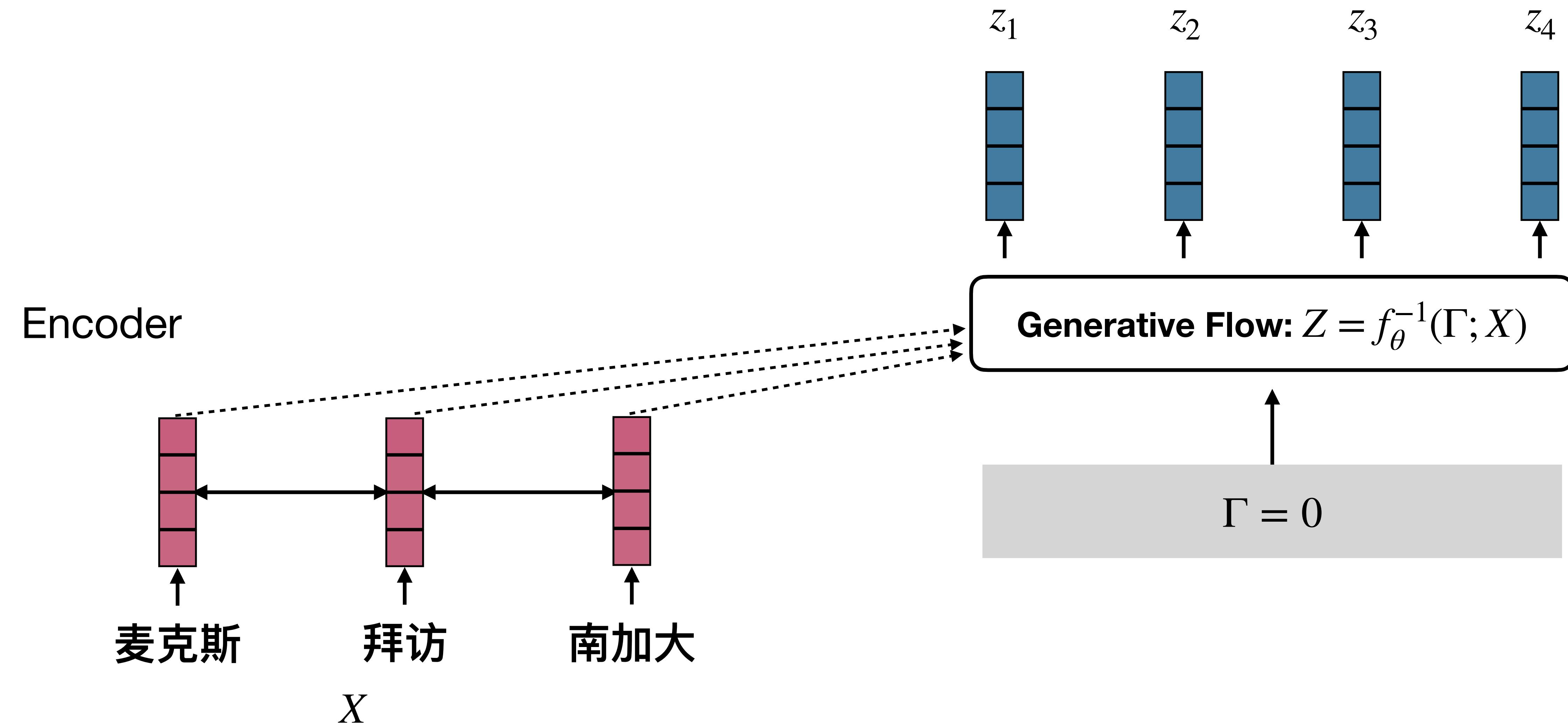
$$\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_T\} \in \mathbb{R}^{d \times T}$$

$$\gamma_t \sim \text{Normal}(0, I), \forall t$$

- Efficient Decoding $z^* = \operatorname{argmax}_{z \in \mathcal{Z}} p_\theta(z|x)$

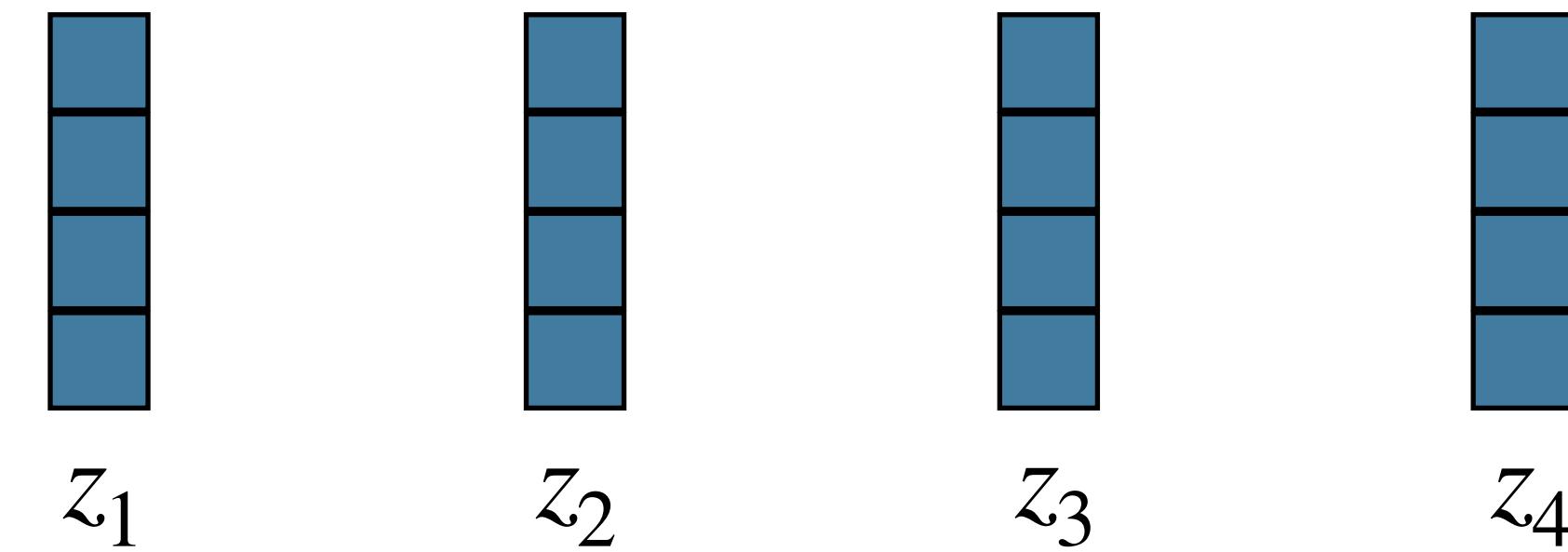
$$z^* \approx f_\theta^{-1}(\Gamma = 0; x)$$

Decoding in FlowSeq

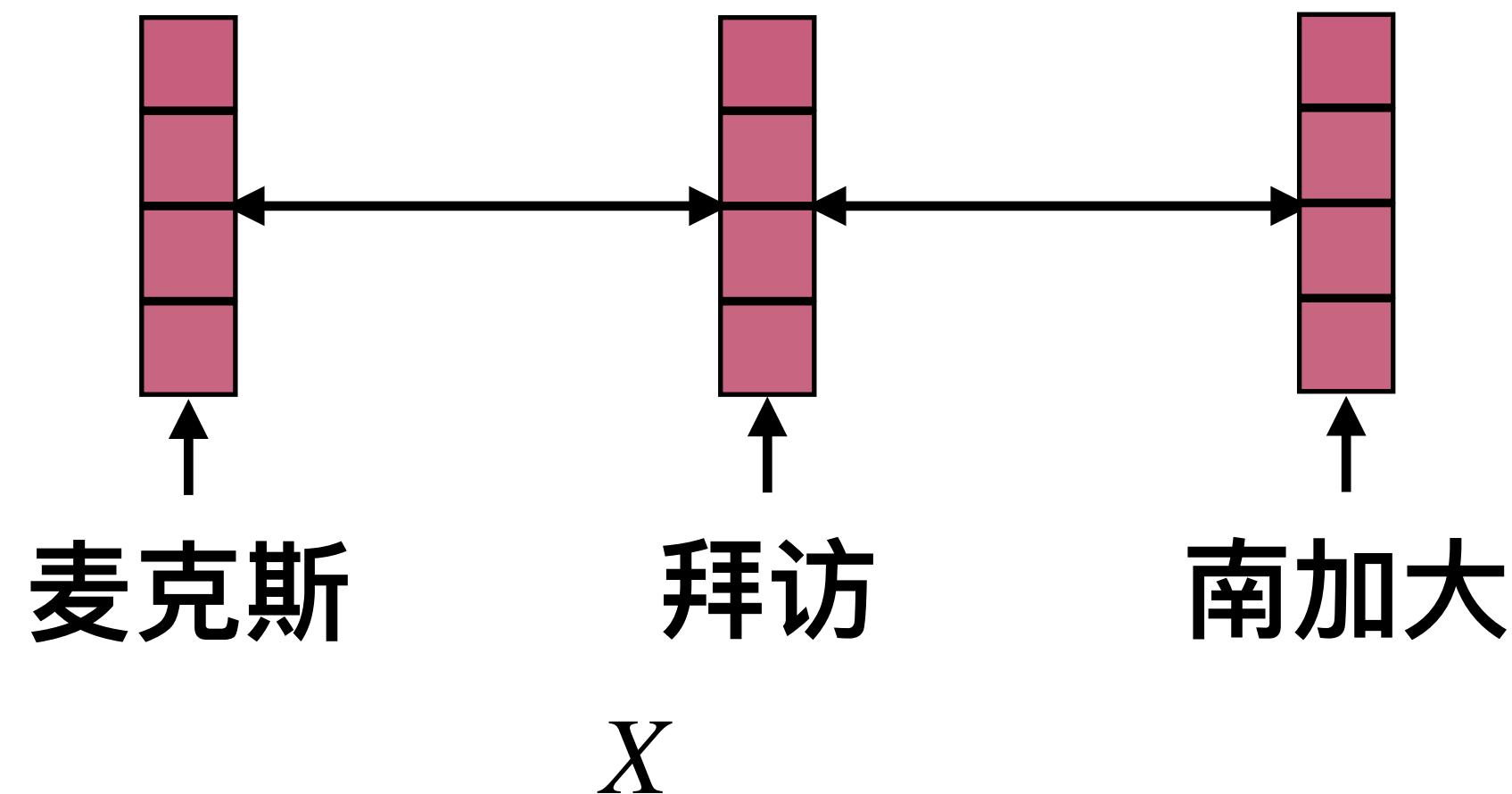


Decoding in FlowSeq

Decoding: $y_t^* = \operatorname{argmax}_{y_t \in V} p_\theta(y_t | z, x), \forall t$



Encoder

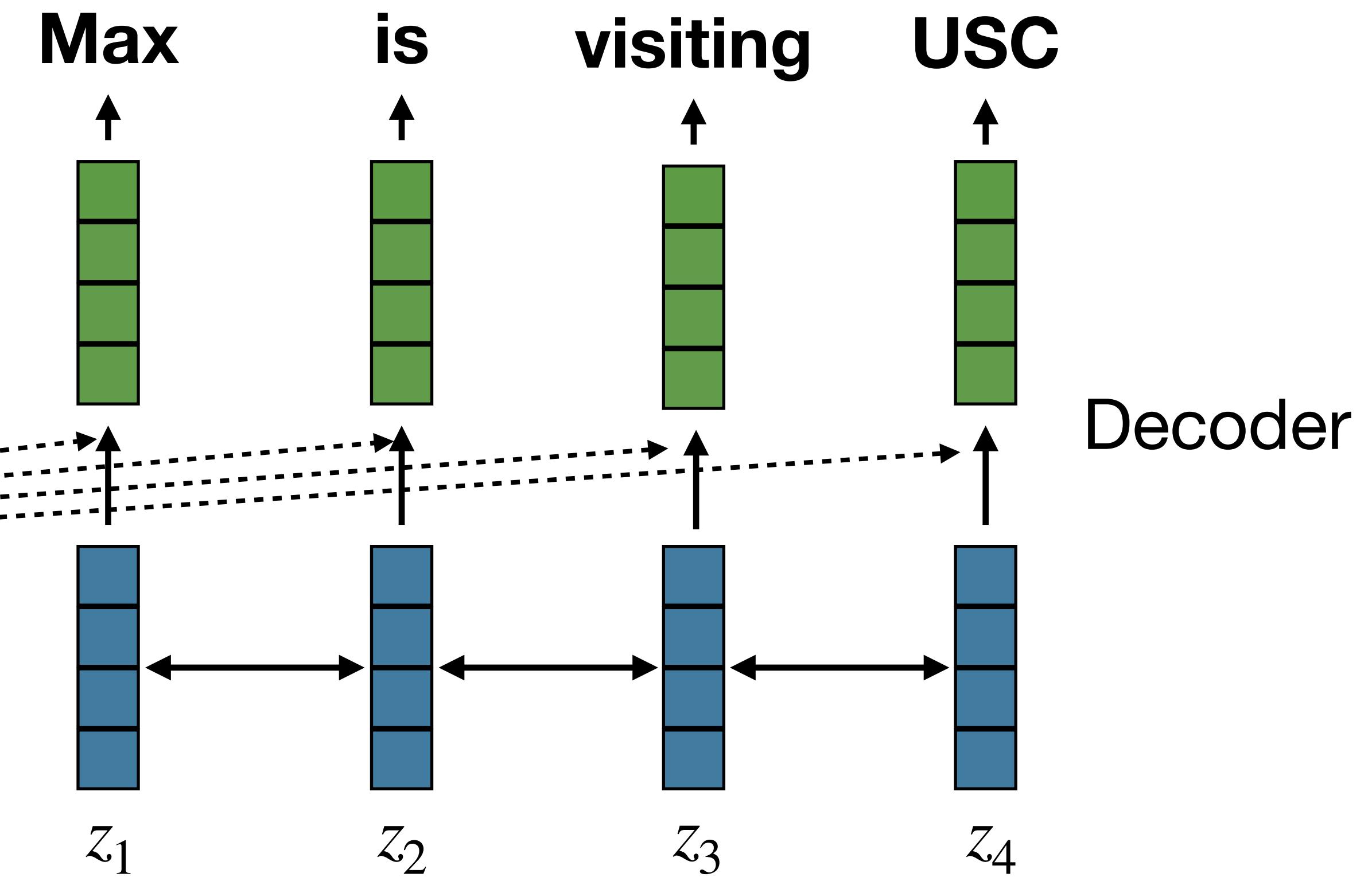
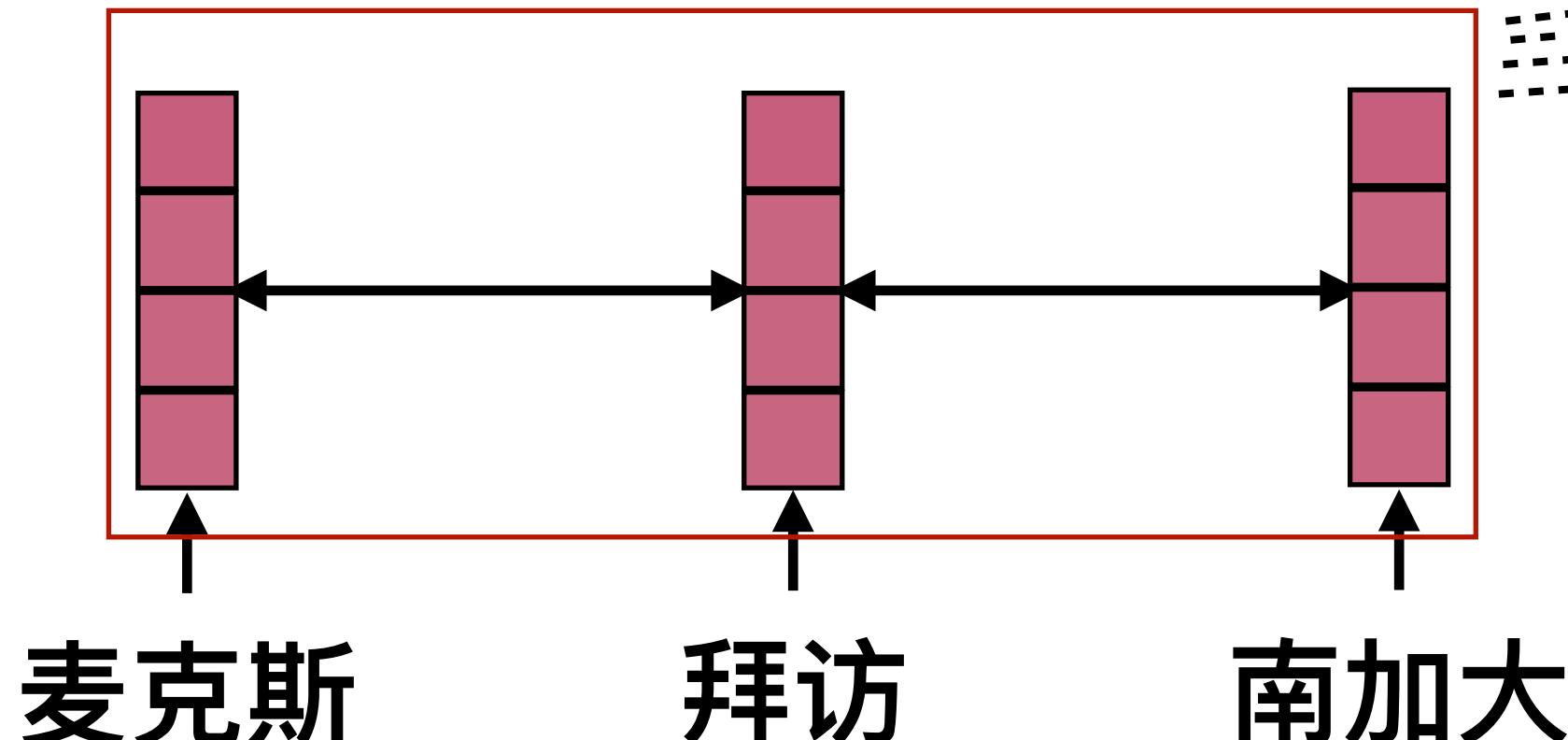


Decoding in FlowSeq

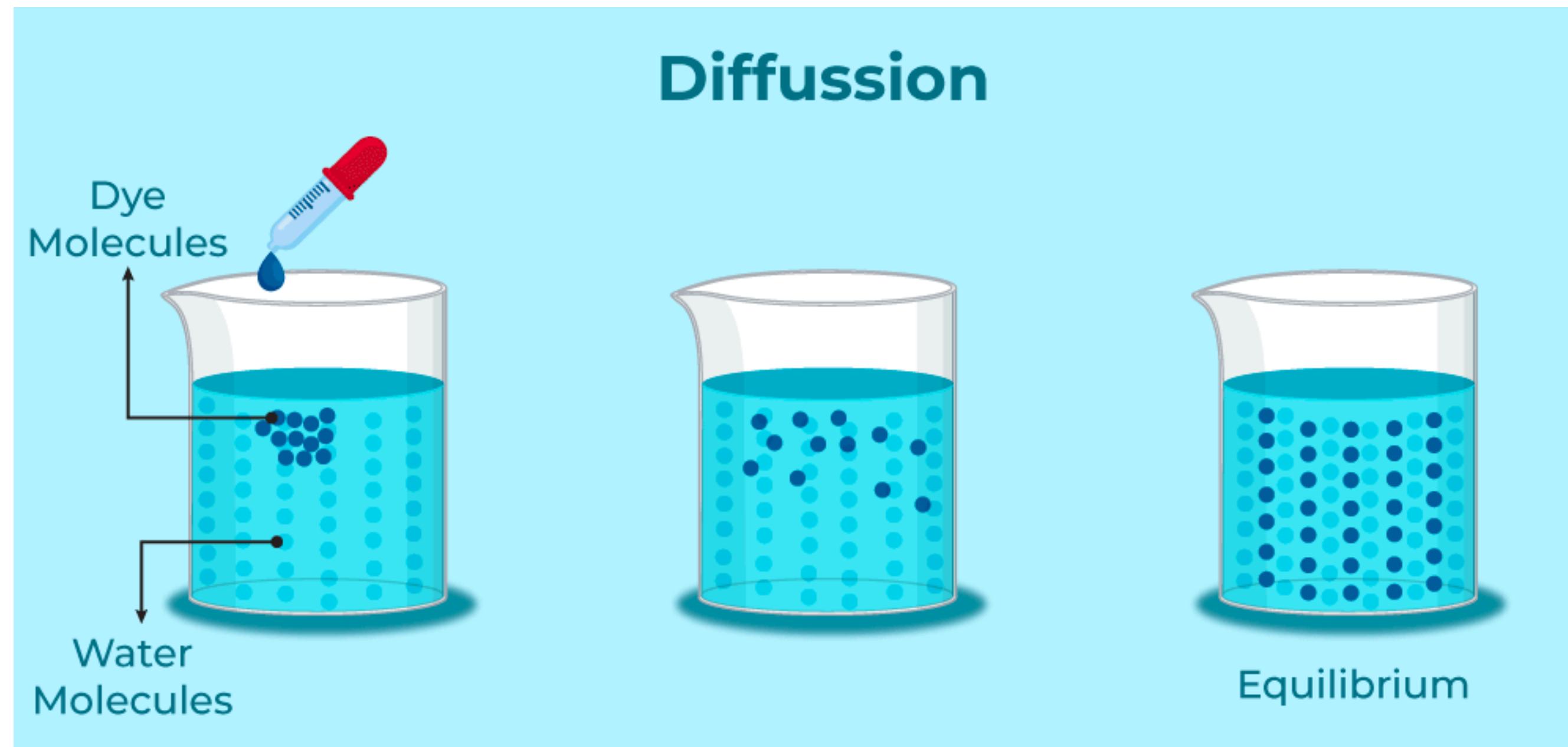
Decoding: $y_t^* = \operatorname{argmax}_{y_t \in V} p_\theta(y_t | z, x), \forall t$

All the decoding steps can be parallel

Encoder

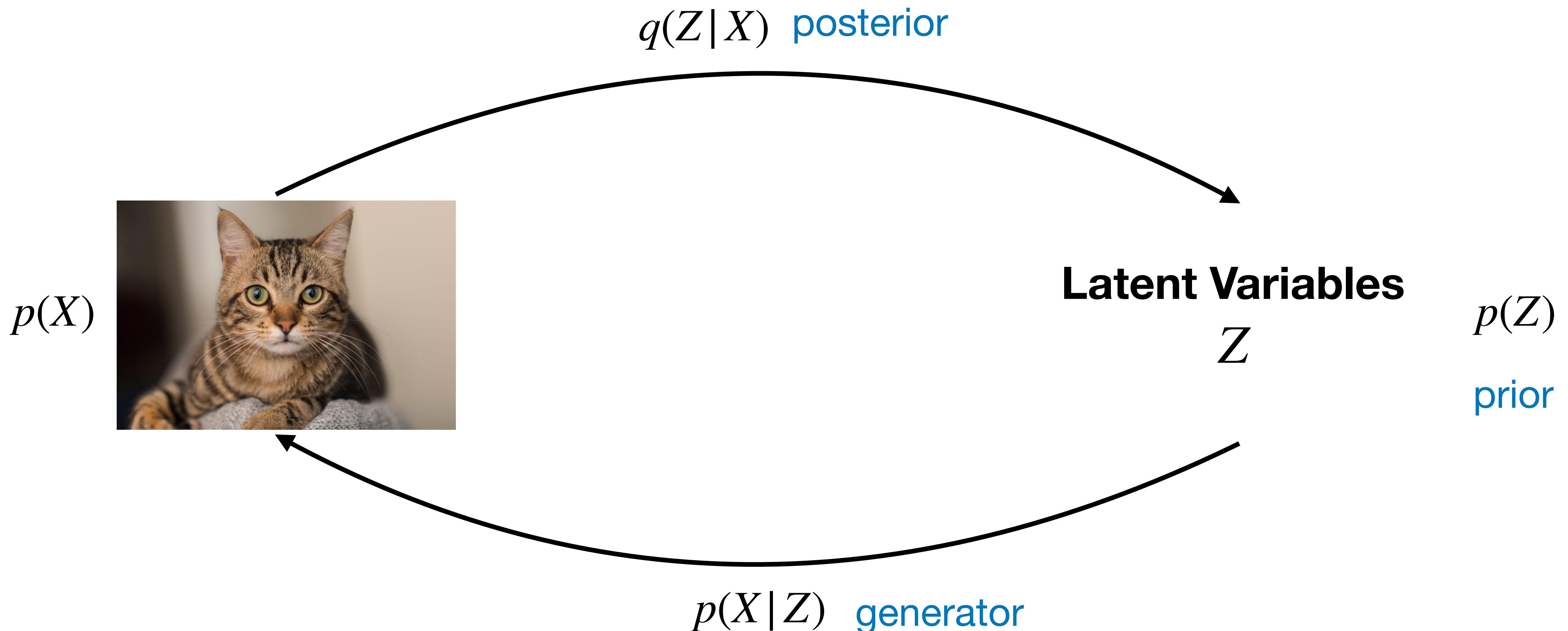


Diffusion Models



Variational Auto-Encoders

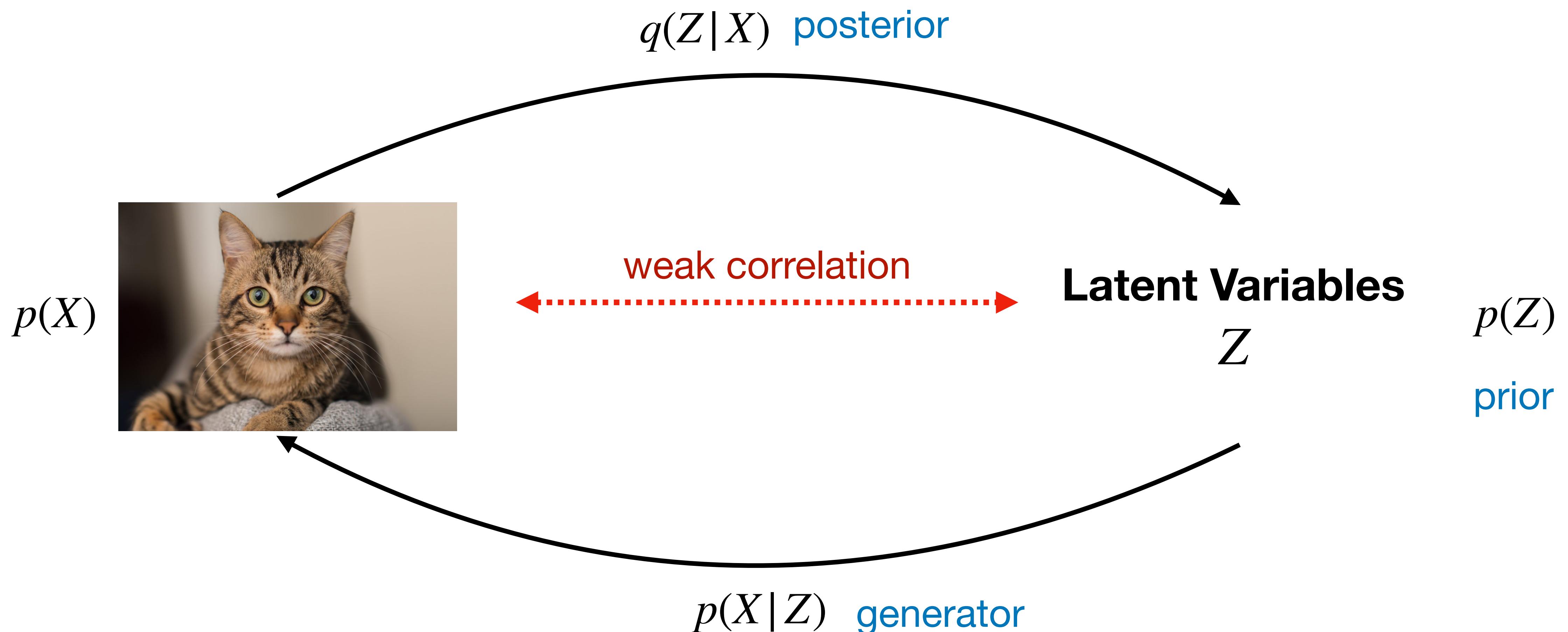
- Why VAEs generate better images than exact density models?
 - Low-dimension Z
 - Simple prior and posterior distributions



Variational Auto-Encoders

- Problems?

- Connection between X and Z is sometimes **weak**



Diffusion Models

- Multi-step hierarchical VAEs
- A chain of latent variables
 - Z_1, Z_2, \dots, Z_T , where each Z_t has the same dimension of X

Prior: $P(Z_T) \sim \mathcal{N}(0, I)$

Posterior:
$$q(Z_1, Z_2, \dots, Z_T | X) = \prod_{t=1}^T q(Z_t | Z_{t-1}), \quad Z_0 := X$$
$$q(Z_t | Z_{t-1}) \sim \mathcal{N}(\sqrt{1 - \beta_t} \cdot Z_{t-1}, \beta_t I)$$

Forward process

Generator:
$$p(X, Z_1, \dots, Z_T) = p(Z_T) \prod_{t=1}^T p(Z_{t-1} | Z_t)$$

Reserve process

$$p(Z_{t-1} | Z_t) \sim \mathcal{N}(\mu(Z_t), \Sigma(Z_t))$$

Diffusion Models

- **Training Objective**
 - ELBO (the same as VAEs)
- **Sampling**
 - Reverse process
 - $Z_T \rightarrow Z_{T-1} \rightarrow \dots \rightarrow Z_1 \rightarrow X$

Diffusion Models

- Diffusion models are good at **generating high-quality images**, but optimized for approximating **density**
- **Why?**



Theoretical Explanation of Diffusion Models

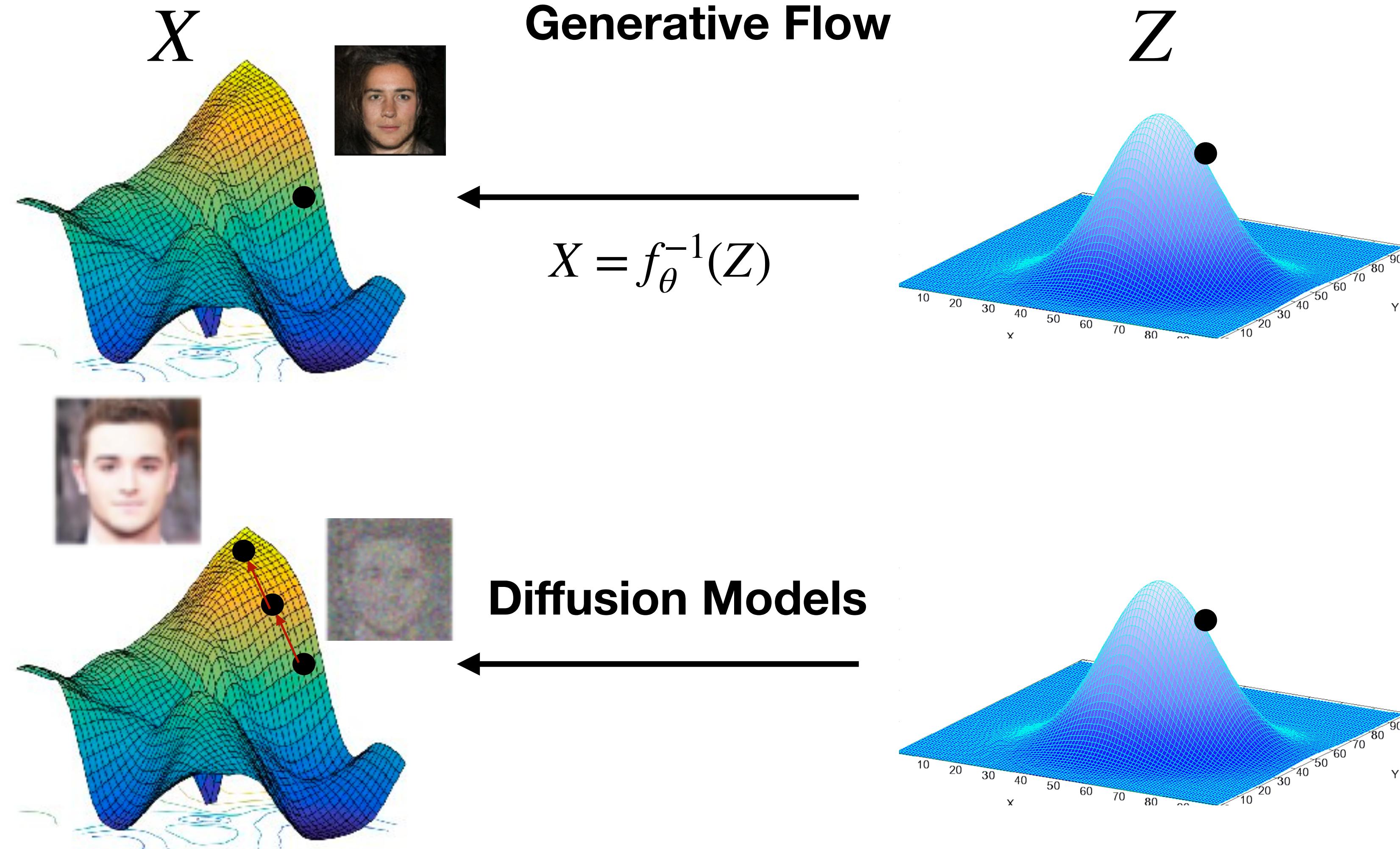
- **Stochastic Gradient Langevin Dynamics**

- $$X_t = X_{t-1} + \frac{\delta}{2} \nabla_X \log q(X_{t-1}) + \sqrt{\delta} \epsilon_t, \text{ where } \epsilon_t \sim \mathcal{N}(0, I)$$

- **Connection with diffusion models**

- The reverse process in diffusion models is equivalent to the Stochastic Gradient Langevin Dynamics

Diffusion Models



Reading Materials

- **Relavant Papers**

- Non-Autoregressive NMT
- Decoupled Representations
- Efficient Attention Mechanisms