Vision Language Models (VLMs)

Xuezhe Ma (Max)

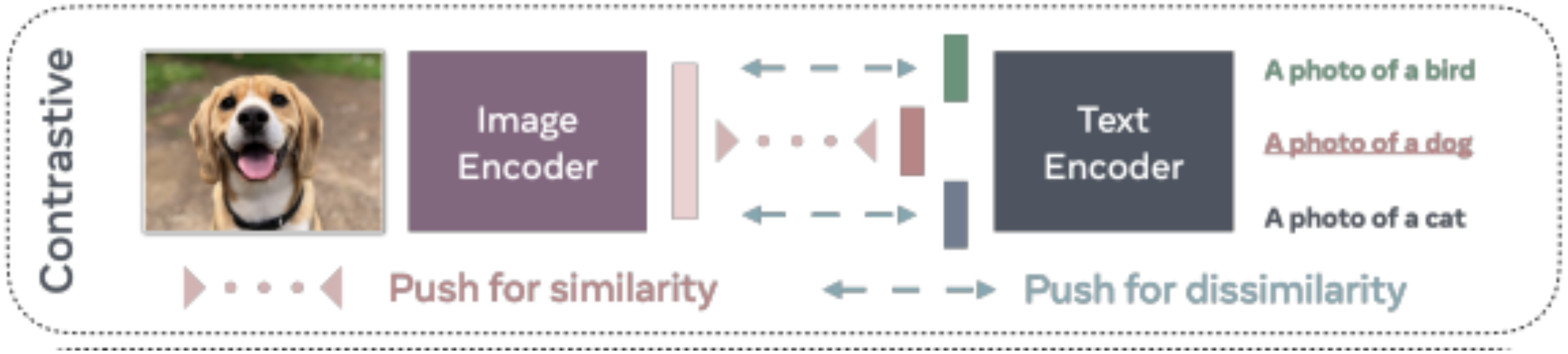
Outline

- **Non-generative VLMs**
  - **Goal:** Text/image understanding
  - Contrastive-based VLMs
  - VLMs from pretrained LLMs
- **Generative VLMs**
  - **Goal:** Text/image understanding & generation
  - Diffusion Models
  - Visual tokenization based models

Contrastive-based VLMs

- Example: CLIP
- Data: pairs of images and their captions
- Networks: one text encoder and one image encoder
- Loss function:

$$\mathcal{L}_{\text{infoNCE}} = -\sum_{(i,j)\in\mathbb{P}} \log\left(\frac{e^{\text{CoSim}(\boldsymbol{z}_i,\boldsymbol{z}_j)/\tau}}{\sum_{k=1}^{N} e^{\text{CoSim}(\boldsymbol{z}_i,\boldsymbol{z}_k)/\tau}}\right)$$

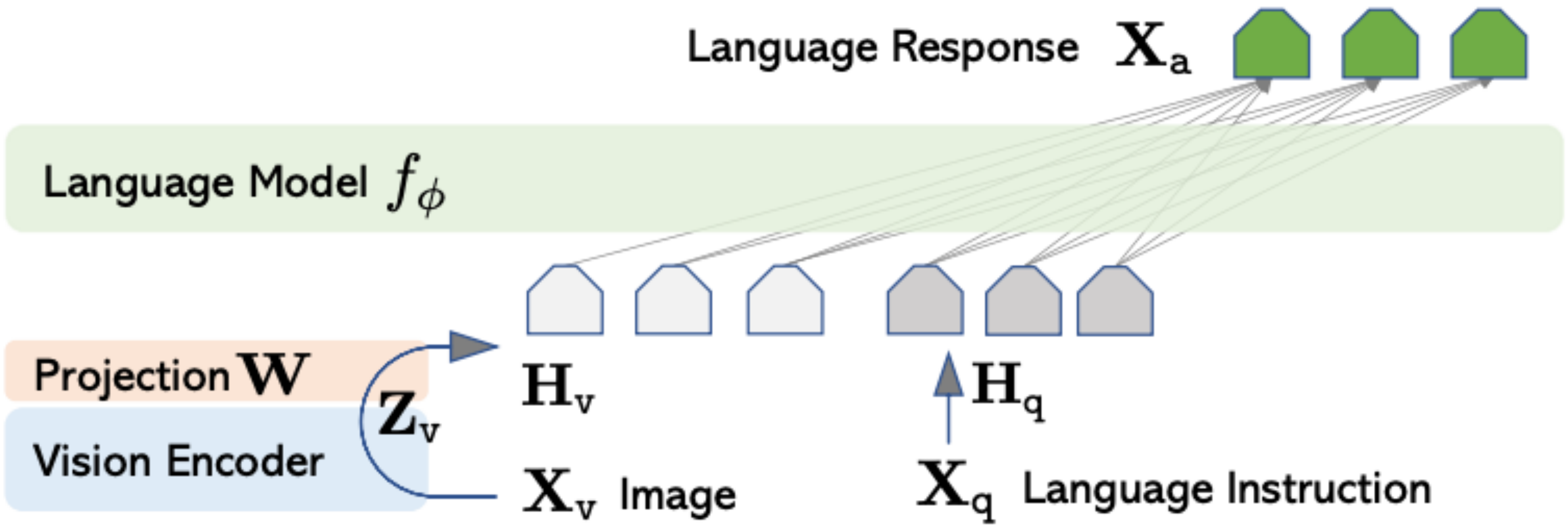


VLMs from Pretrained LLMs

- Example: LLaVA
- Data: pairs of images and their captions
- Network: One vision encoder, one mapping network and one LLM
- Loss function: language modeling

VLMs from Pretrained LLMs

LLaVA



Generative VLMs

Outline

- Non-generative VLMs
  - Goal: Text/image understanding
  - Contrastive-based VLMs
  - VLMs from retrained LLMs
- **Generative VLMs**
  - **Goal:** Text/image understanding & generation
  - Diffusion Models
  - Visual tokenization based models

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

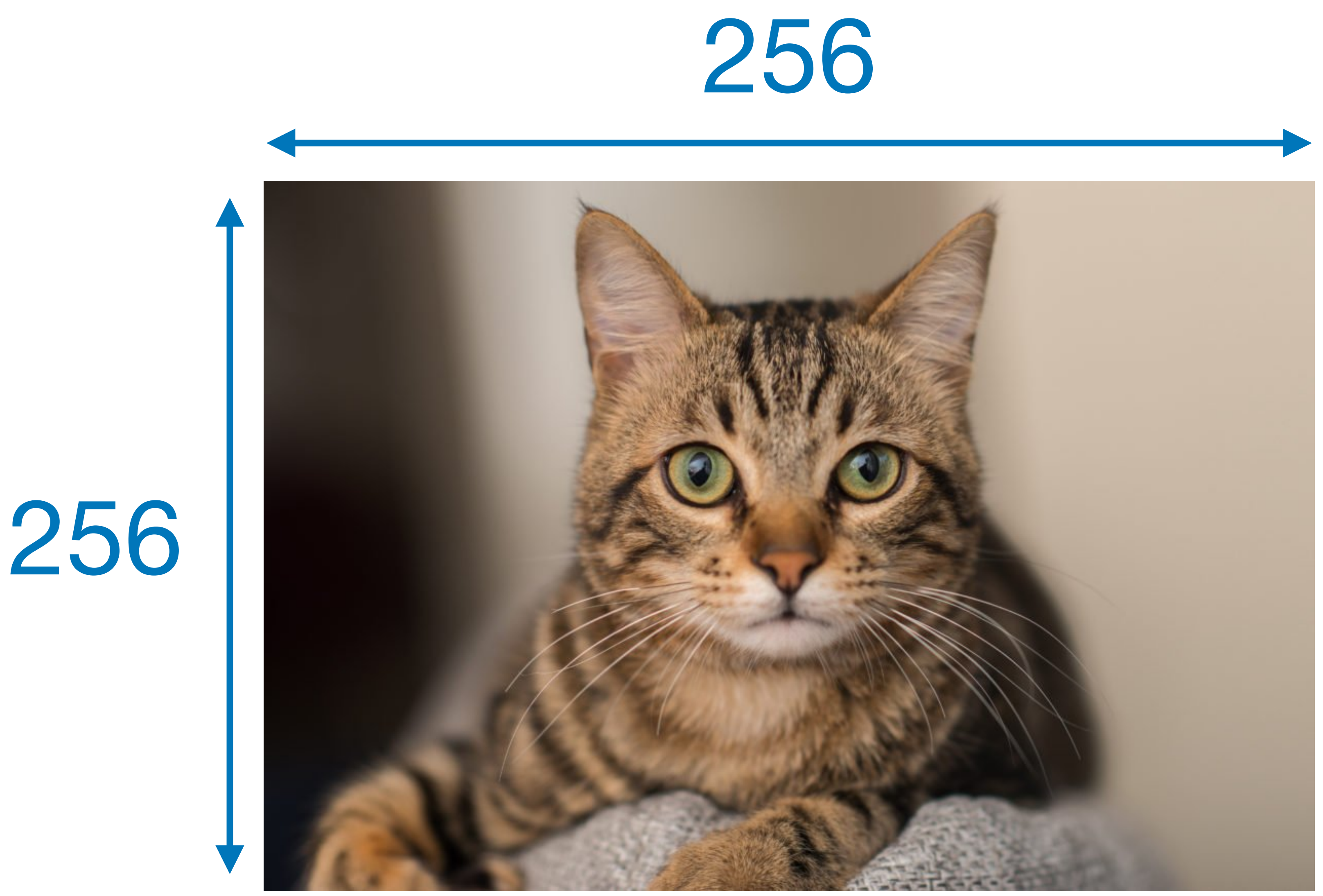
X

Deep Generative Models

$P(X)$

Value only!

Problems on Autoregressive Models for Image



256 x 256 x 3 = 131072 pixels

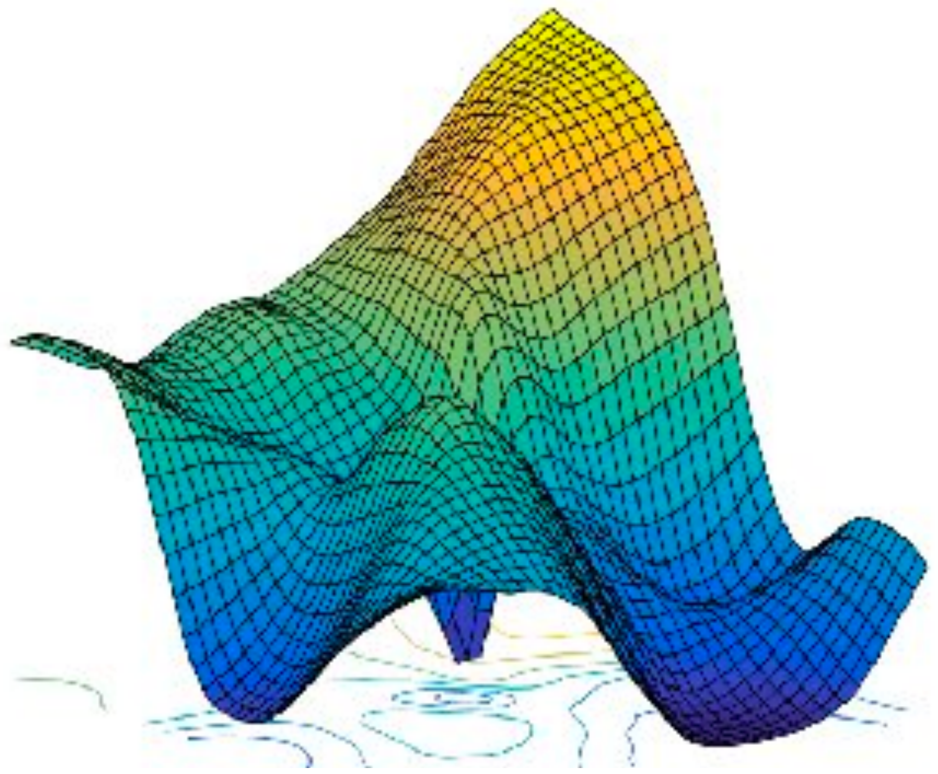
Problems:

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

Generative (Normalizing) Flows

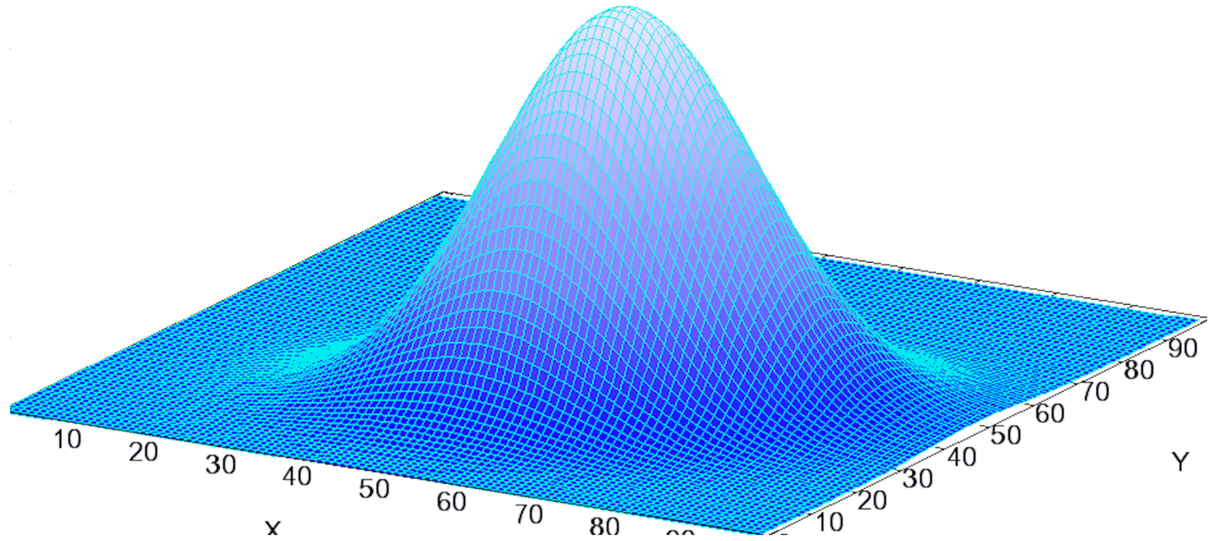
- **Modeling density via invertible mapping**
  - Directly modeling the joint distribution of all variates in X
  - Exact density estimation (no approximation)

Generative (Normalizing) Flows



$$\Gamma = f_\theta(X)$$

$$X = f_\theta^{-1}(\Gamma)$$



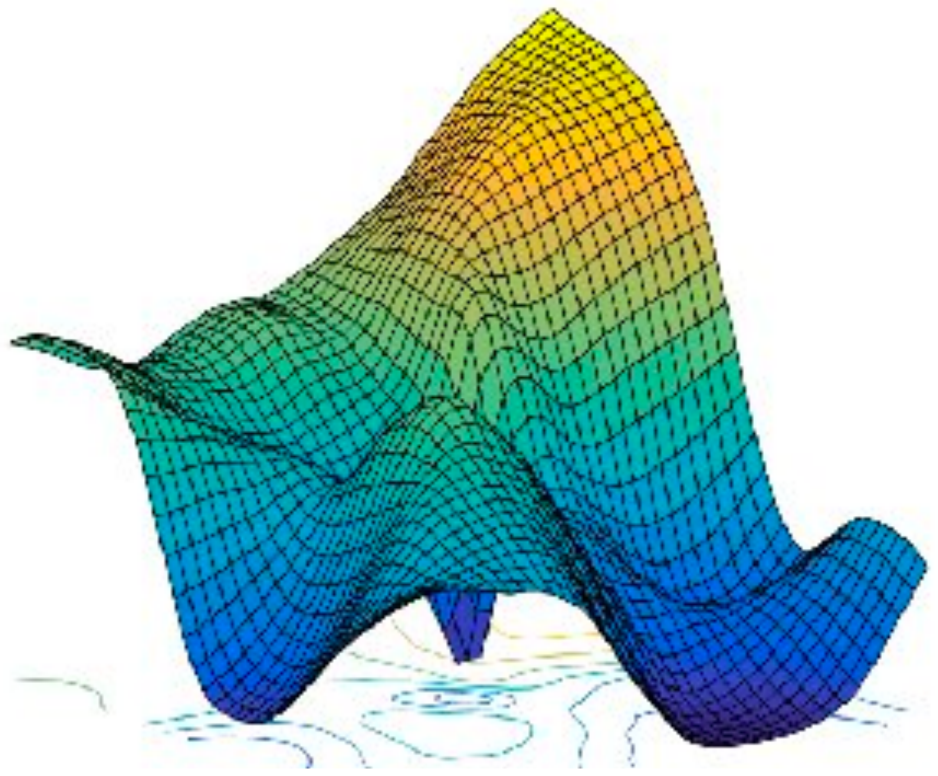
$X \sim p_\theta(X)$

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

Change of Variable formula:

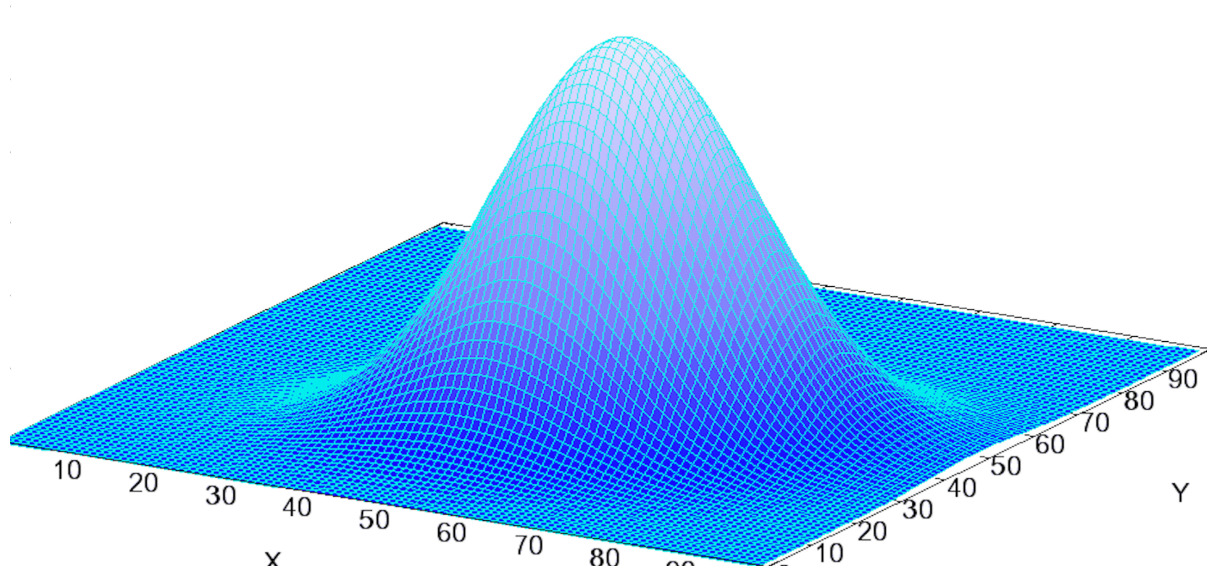
$$p_\theta(x) = \underbrace{p_\Gamma\left(f_\theta(x)\right)}_{\boxed{\textbf{Normal}}} \left| \det \underbrace{\left(\frac{\partial f_\theta(x)}{\partial x} \right)}_{\boxed{\textbf{Jacobian Matrix}}} \right|$$

Generative (Normalizing) Flows



$$\Gamma = f_\theta(X)$$

$$X = f_\theta^{-1}(\Gamma)$$



$X \sim p_\theta(X)$

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

Change of Variable formula:

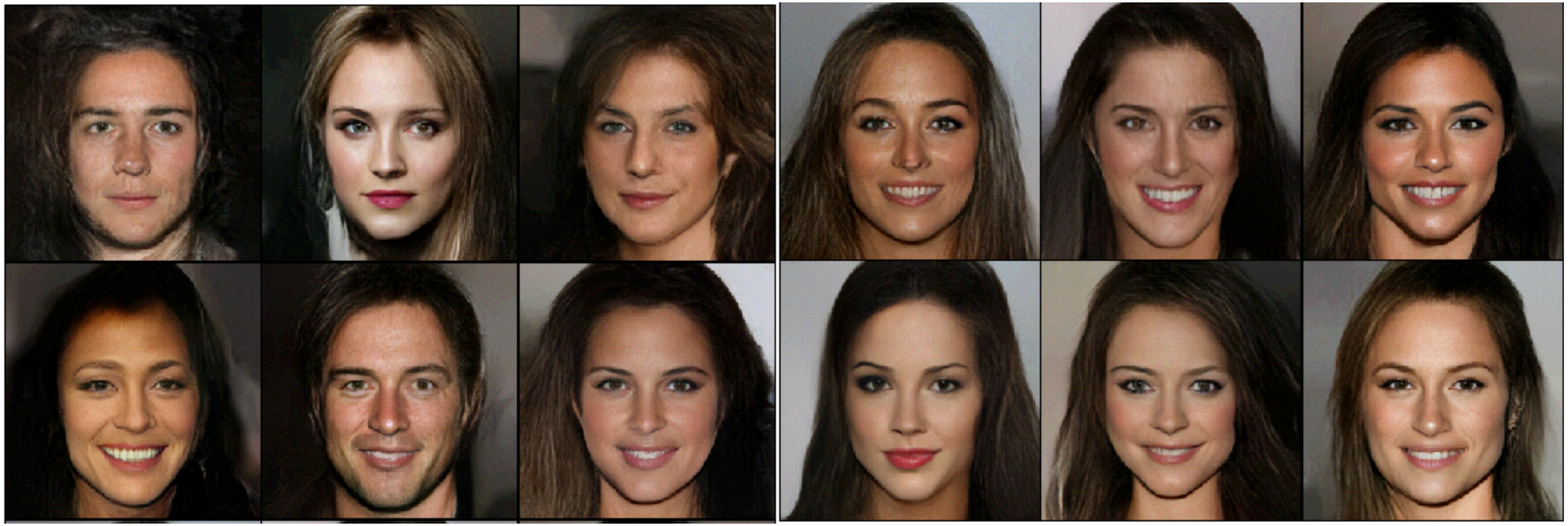
$$p_\theta(x) = p_\Gamma\left(f_\theta(x)\right)\left|\det\left(\frac{\partial f_\theta(x)}{\partial x}\right)\right|$$

Generative Flow: A series of such f

$$X \xleftrightarrow[g_1]{f_1} H_1 \xleftrightarrow[g_2]{f_2} H2 \xleftrightarrow[g_3]{f_3} \cdots \xleftrightarrow[g_K]{f_K} \Gamma$$

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

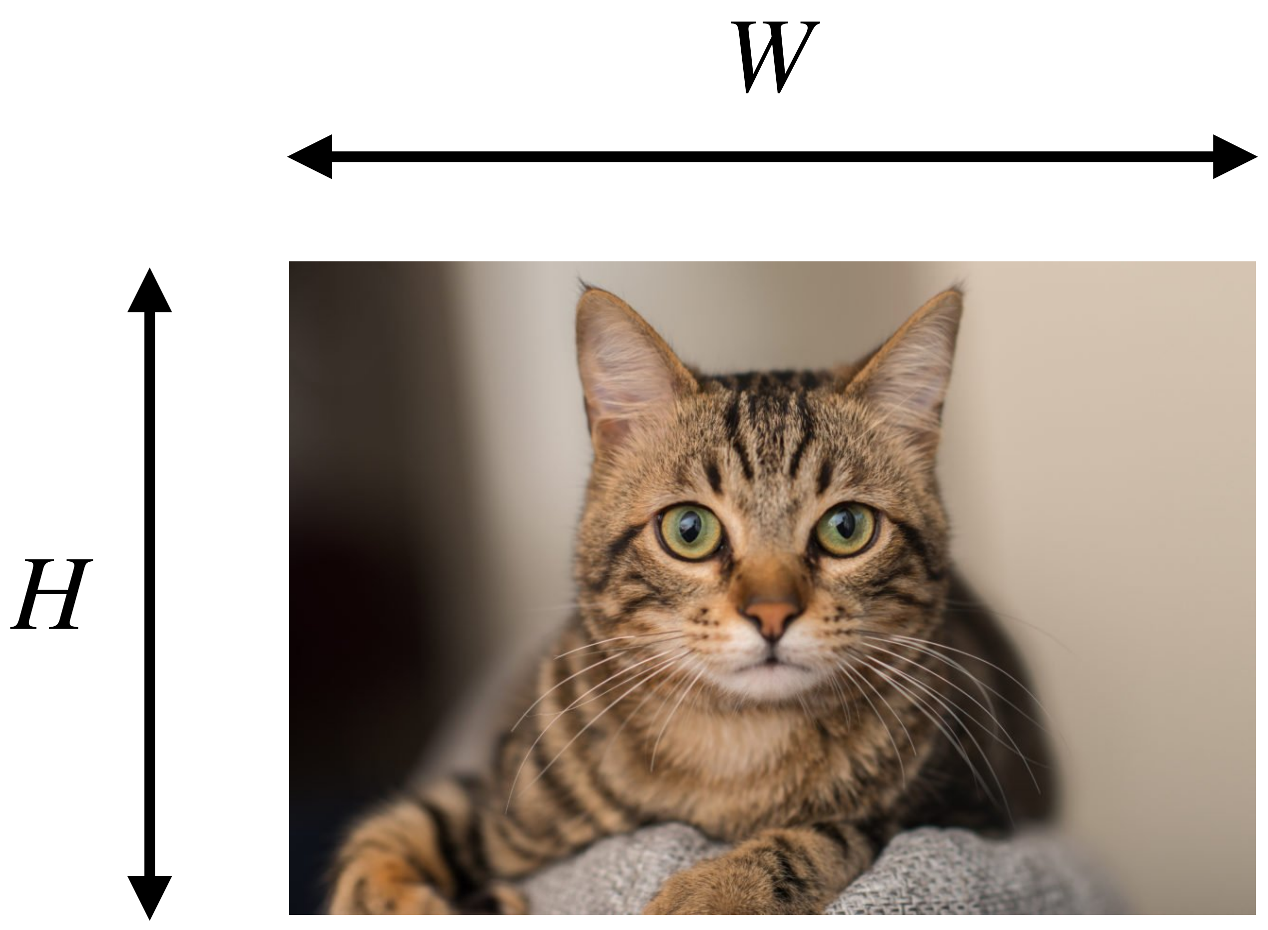
- **Goal: learn to generate new data from samples**
  - How?
  - To model the data distribution $P(X)$
    - Closed-form analytic solution
    - Exact density estimation via “black-box” deep neural networks
    - Density/distribution approximation

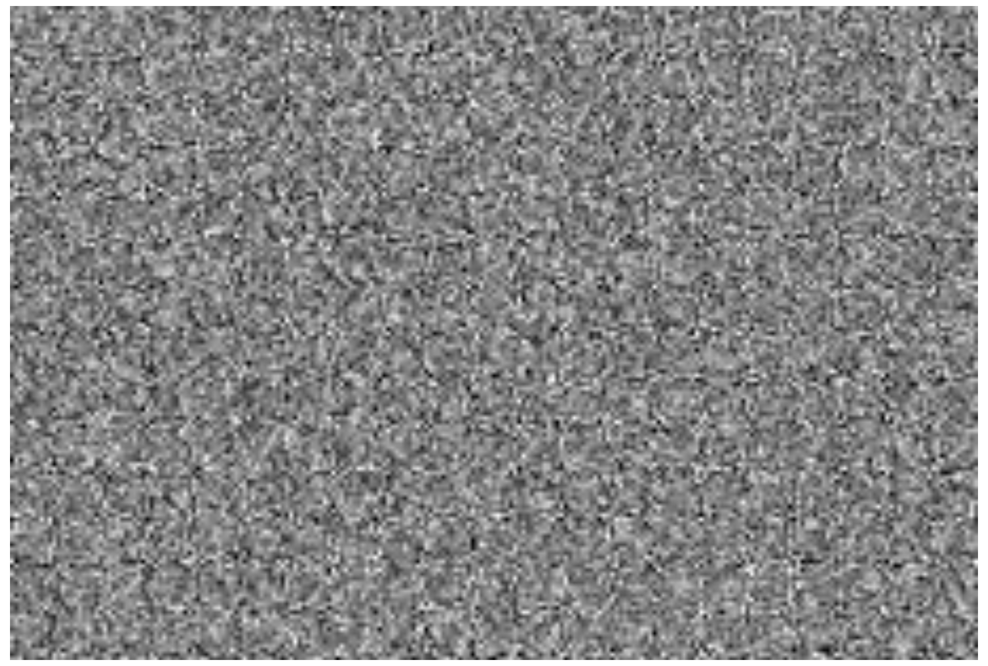
Problems of Exact Density Estimation

- **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**

$q(Z|X)$ posterior

$p(X)$

Latent Variables

Z

$p(Z)$

prior

$p(X|Z)$ generator

Variational Auto-Encoders

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

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

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

Variational Inference

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

Evidence Lower Bound (ELBO)

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

Posterior **Generator** **Prior**

Variational Inference

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

Evidence Lower Bound (ELBO)

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

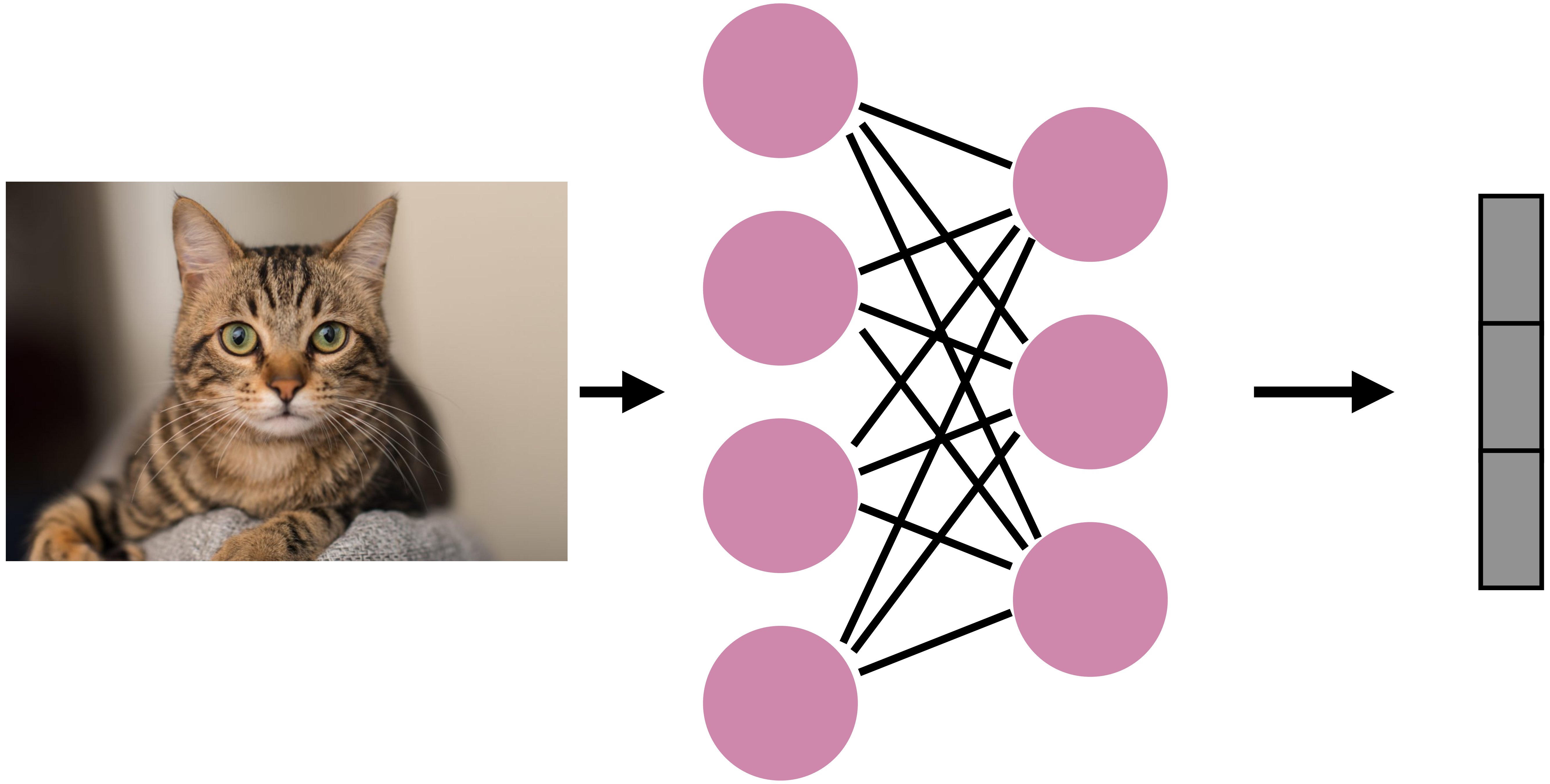
$$= \underbrace{\mathrm{E}_{q(Z|X)}[\log p(X|Z)]}_{\textbf{Reconstruction}} - \underbrace{\mathrm{KL}(q(Z|X)||p(Z))}_{\textbf{KL Regularizer}}$$

Variational Inference

Evidence Lower Bound (ELBO)

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

Posterior



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

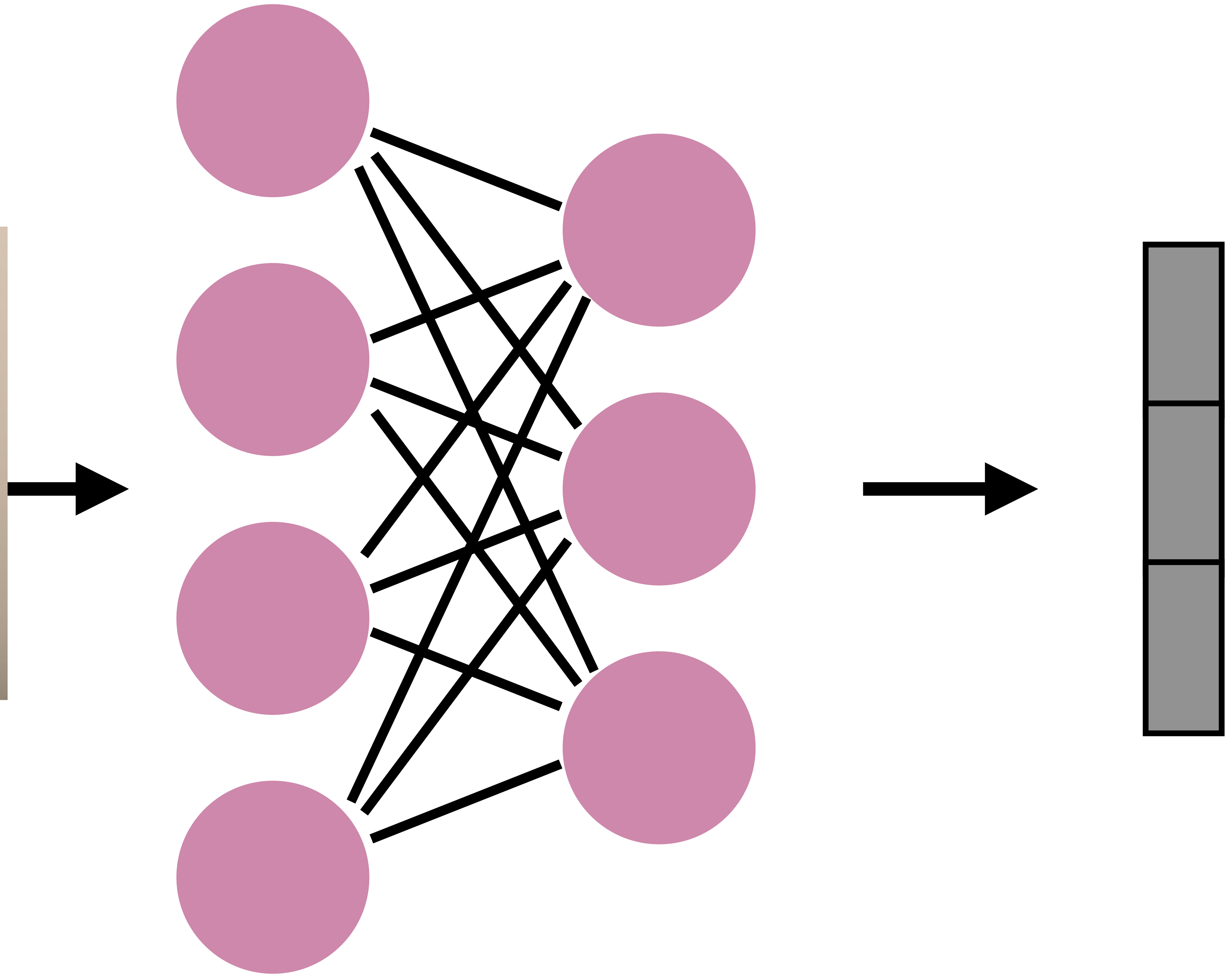
Variational Inference

Evidence Lower Bound (ELBO)

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

Posterior

Prior



X

$q_\phi(Z|X)$

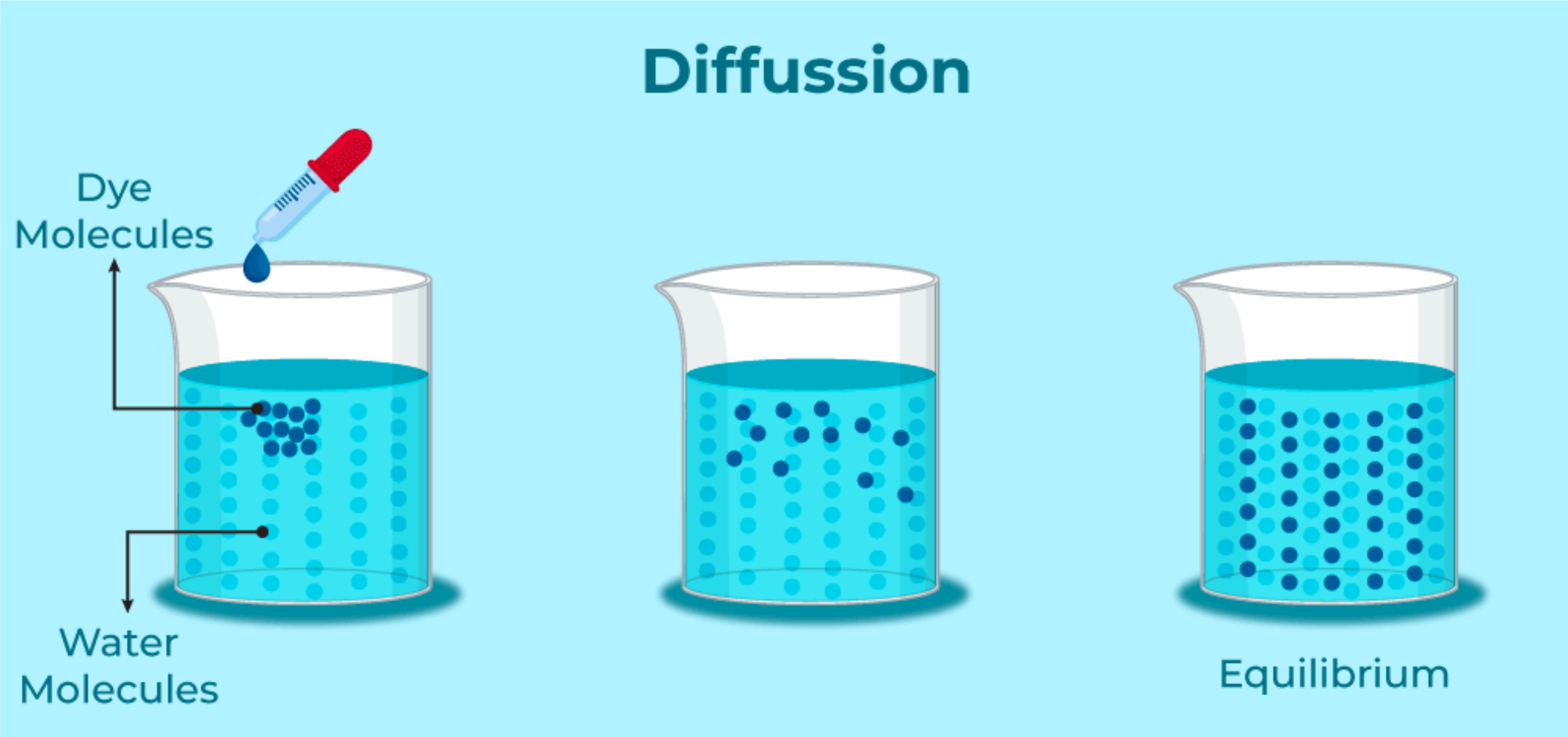
$Z \sim p_\theta(Z)$

Variational Inference

Evidence Lower Bound (ELBO)

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

Diffusion Models



Diffusion Models

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

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

Posterior: $q(Z_1, Z_2, \ldots, Z_T | X) = \prod_{t=1}^{T} q(Z_t | Z_{t-1}), \quad Z_0 := X$

$$q(Z_t | Z_{t-1}) \sim \mathcal{N}(\sqrt{1-\beta_t} \cdot Z_{t-1}, \beta_t I)$$

Forward process

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

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
  - $Z_T \rightarrow Z_{T-1} \rightarrow \ldots \rightarrow Z_1 \rightarrow X$

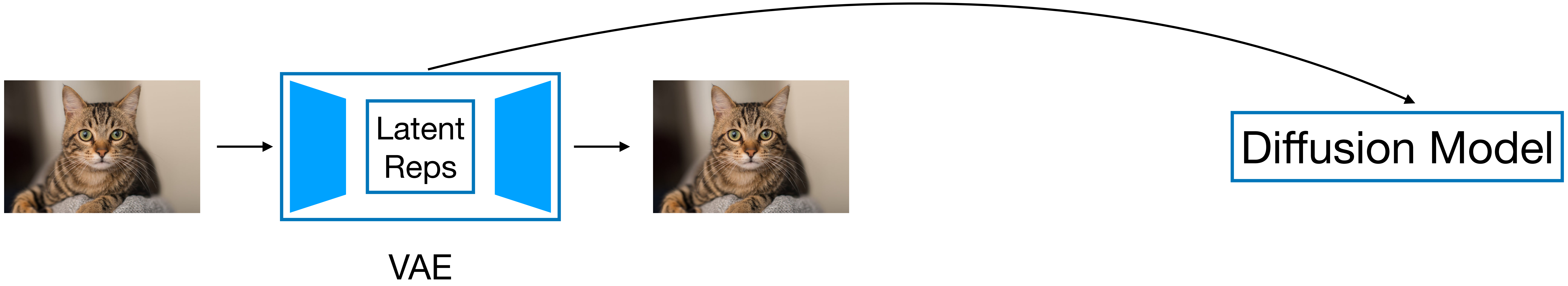
Diffusion Models

- Diffusion models are good at generating high-quality images
- Learning is slow and expensive


Latent Diffusion Models

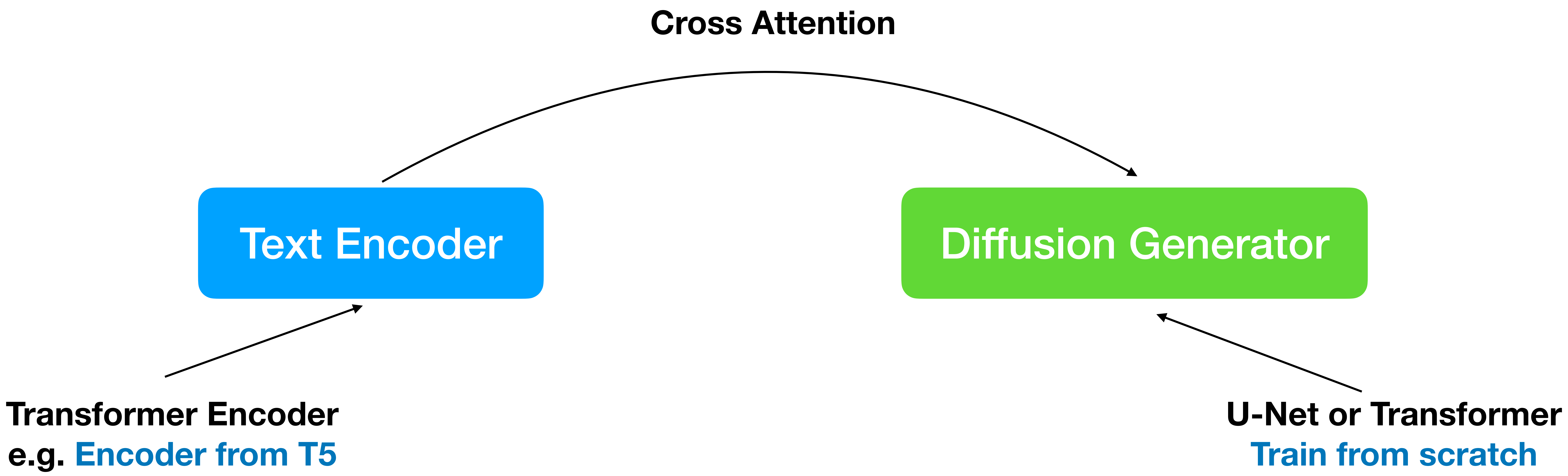
- **Learning from pixels is hard**

- **Combining VAE and Diffusion Models**
  - Stage-I: a latent space VAE
  - Stage-II a diffusion model on top of the latent space



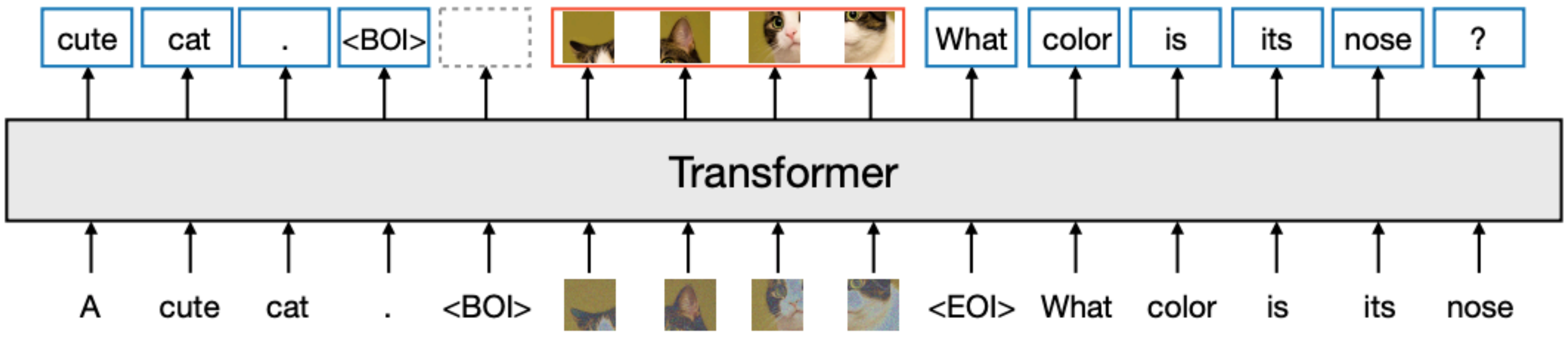
Latent Diffusion Models

Neural Networks in Diffusion Models



Unifying Text and Image in Diffusion Models

Transfusion



Unifying Text and Image in Diffusion Models

the word ‘START’ on a blue t-shirt

A Dutch still life of an arrangement of tulips in a fluted vase. The lighting is subtle, casting gentle highlights on the flowers and emphasizing their delicate details and natural beauty.

A wall in a royal castle. There are two paintings on the wall. The one on the left a detailed oil painting of the royal raccoon king. The one on the right a detailed oil painting of the royal raccoon queen.

Three spheres made of glass falling into ocean. Water is splashing. Sun is setting.

A transparent sculpture of a duck made out of glass.

A chromeplated cat sculpture placed on a Persian rug.

A kangaroo holding a beer, wearing ski goggles and passionately singing silly songs.

an egg and a bird made of wheat bread

Outline

- Non-generative VLMs
  - Goal: Text/image understanding
  - Contrastive-based VLMs
  - VLMs from retrained LLMs
- **Generative VLMs**
  - **Goal:** Text/image understanding & generation
  - Diffusion Models
  - Visual tokenization based models

Visual Tokenization

- Mapping each image patch to a discrete token index
- VQ-VAE

$q(Z|X)$ posterior

Clustering procedure

$p(X)$

Latent Variables

Z

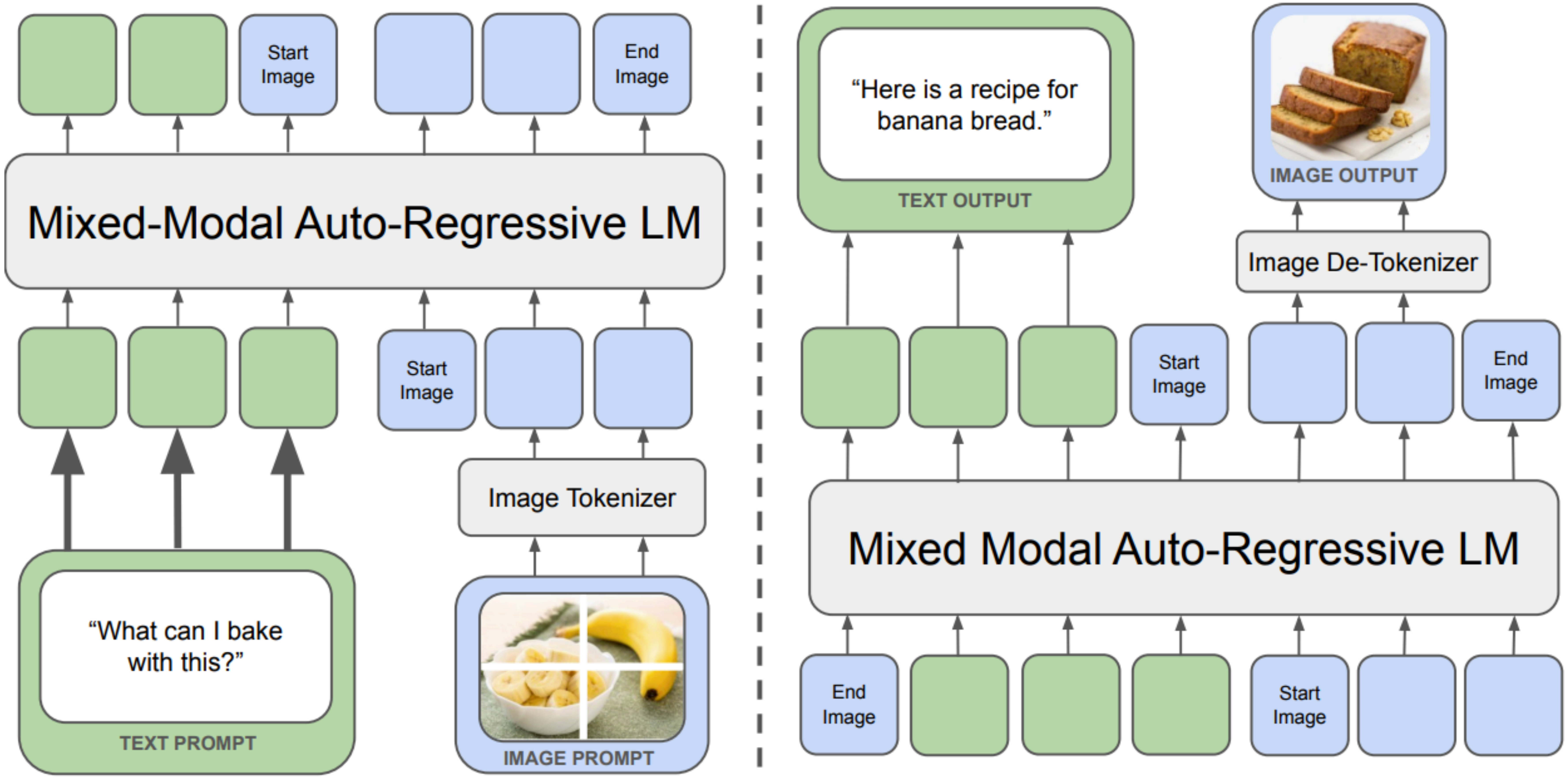
$p(Z)$

prior

$p(X|Z)$ generator

Visual Tokenization

Chameleon

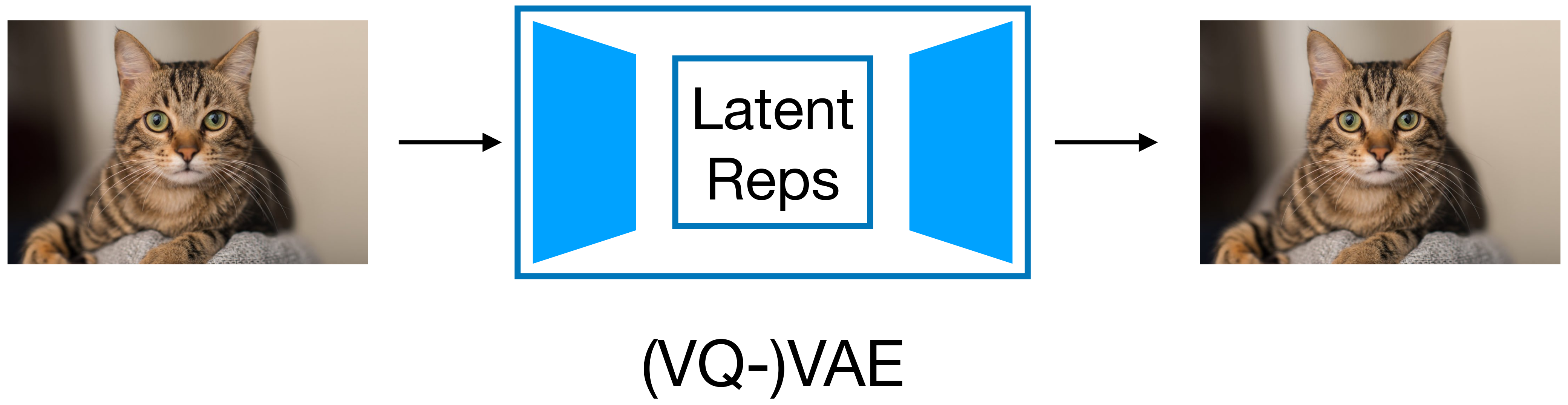


(a) Mixed-Modal Pre-Training

(b) Mixed-Modal Generation

Problems of Two-Stage Models

- Losing image information from latent space
- Falling behind non-generative VLMs on understanding tasks



Thanks!

Q&A