模式识别
Pattern Recognition

李泽桦，复旦大学 生物医学工程与技术创新学院

课件内容参考MIT 6.S978 Deep Generative Models

目录

1 生成模型基础

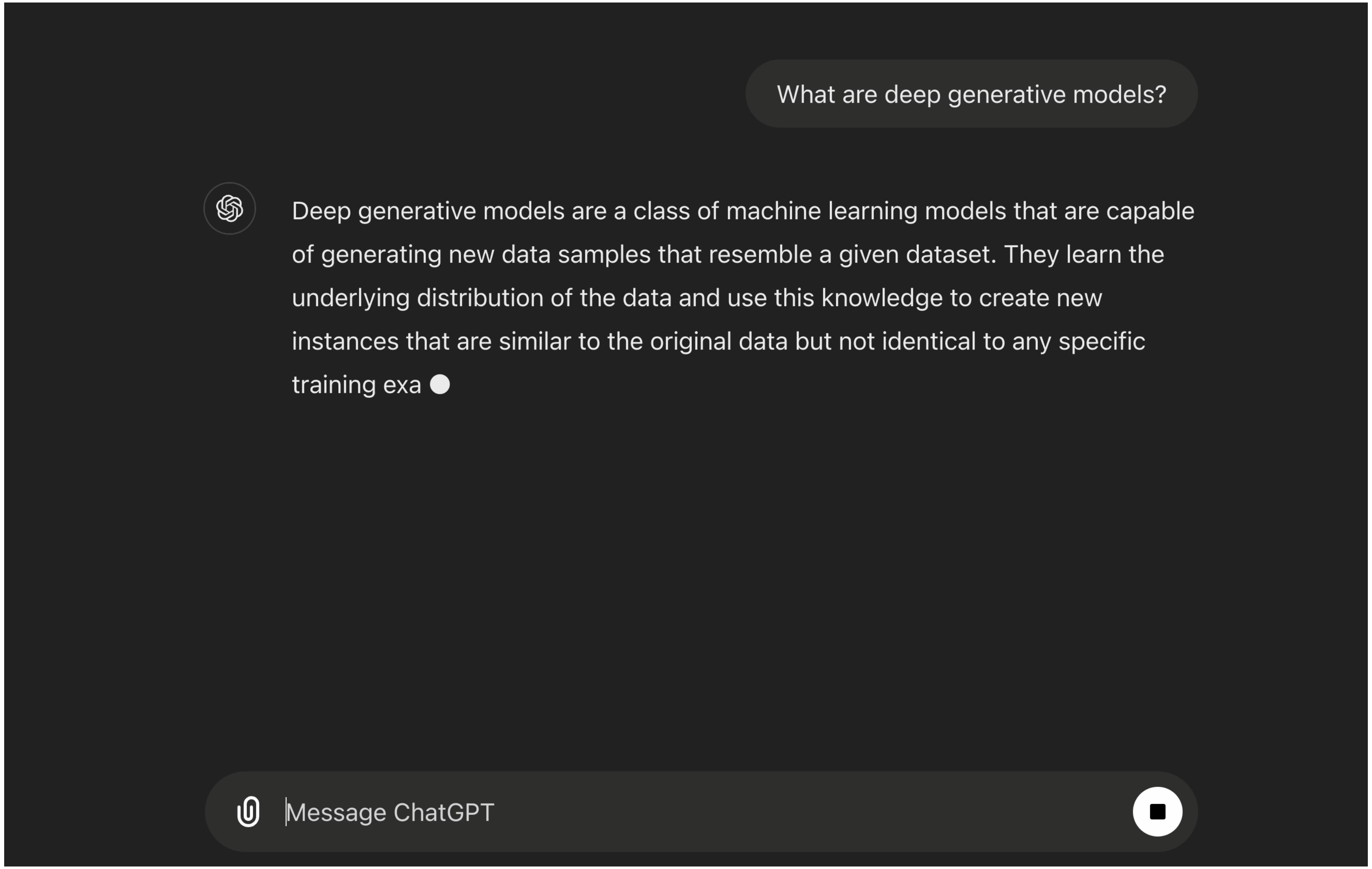
2 独立潜在变量建模

3 条件分布建模

4 现实问题建模

GenAI时代

- 聊天机器人和自然语言对话



GenAI时代

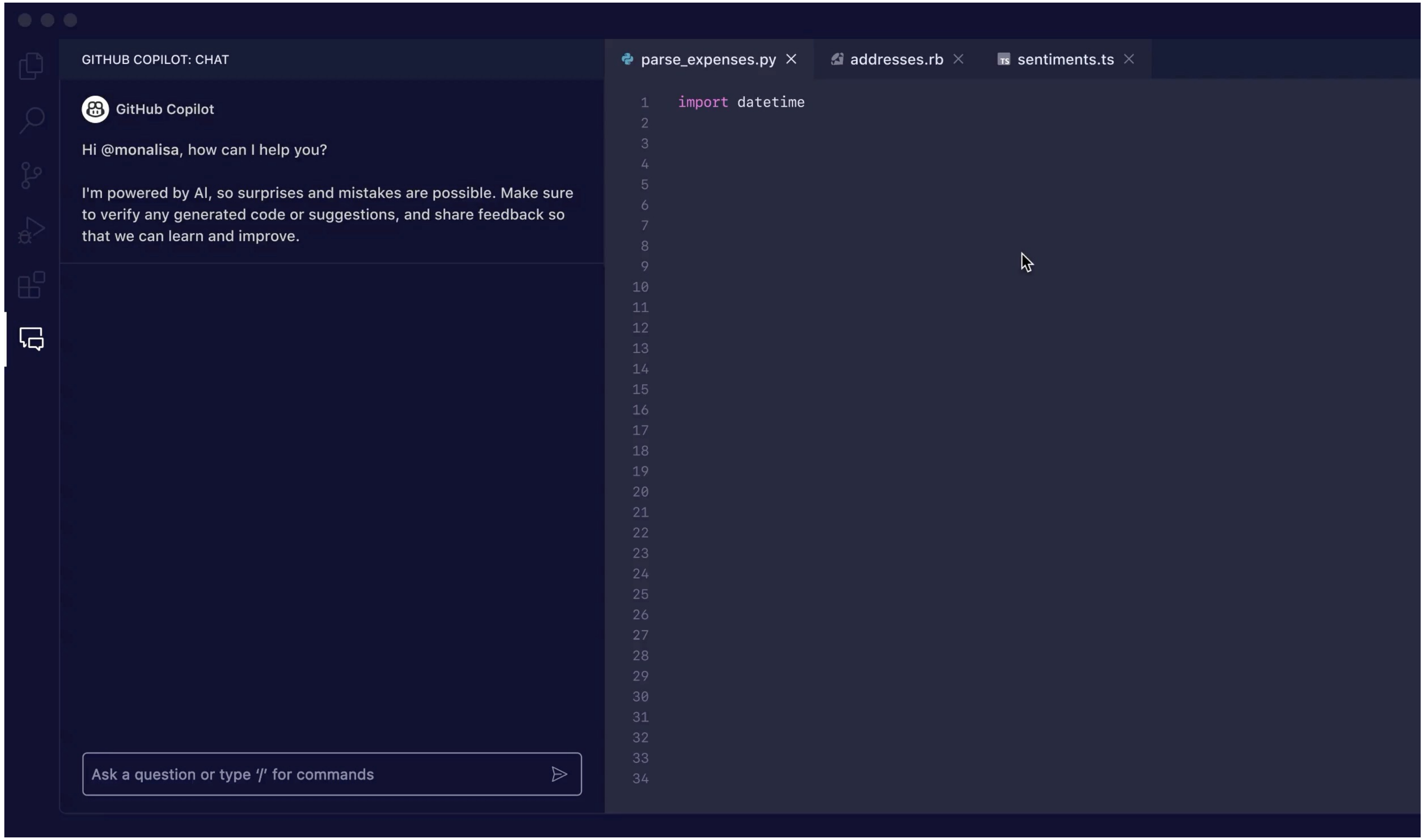
- 文生图

GenAI时代

- 文生视频

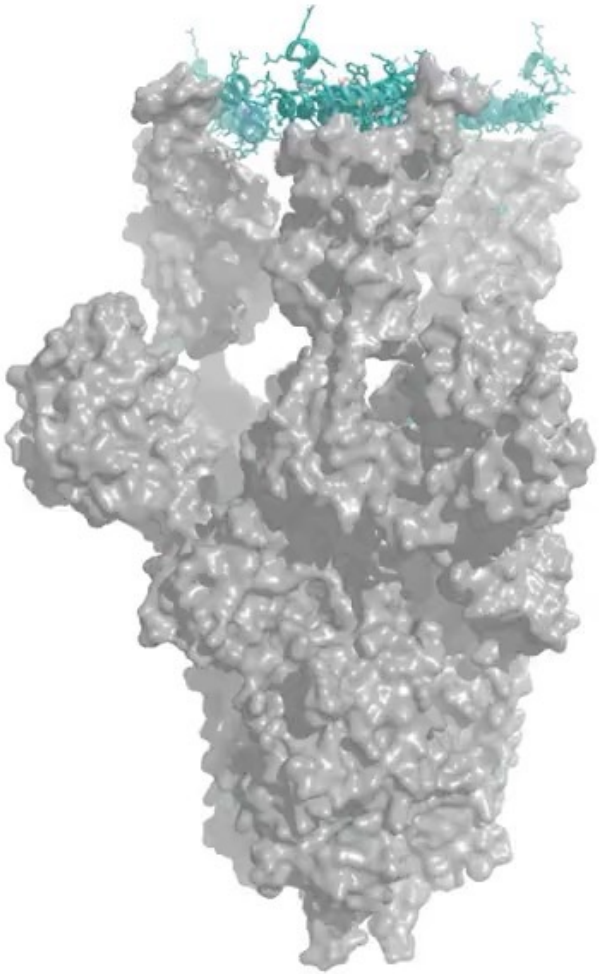
GenAI时代

- 代码生成



GenAI时代

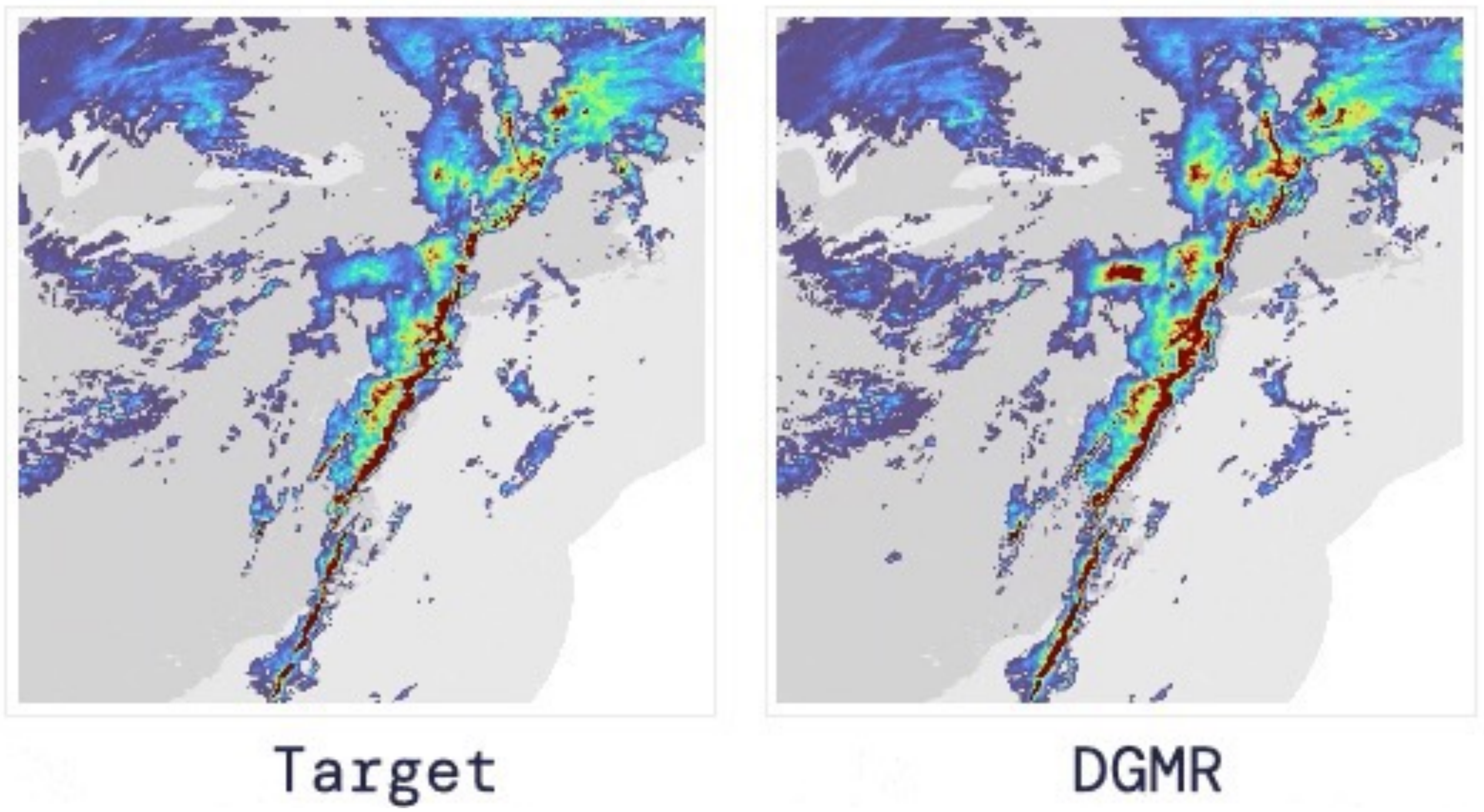
- 蛋白质设计与生成



Watson, et al. De novo design of protein structure and function with RF diffusion, Nature 2023

GenAI时代

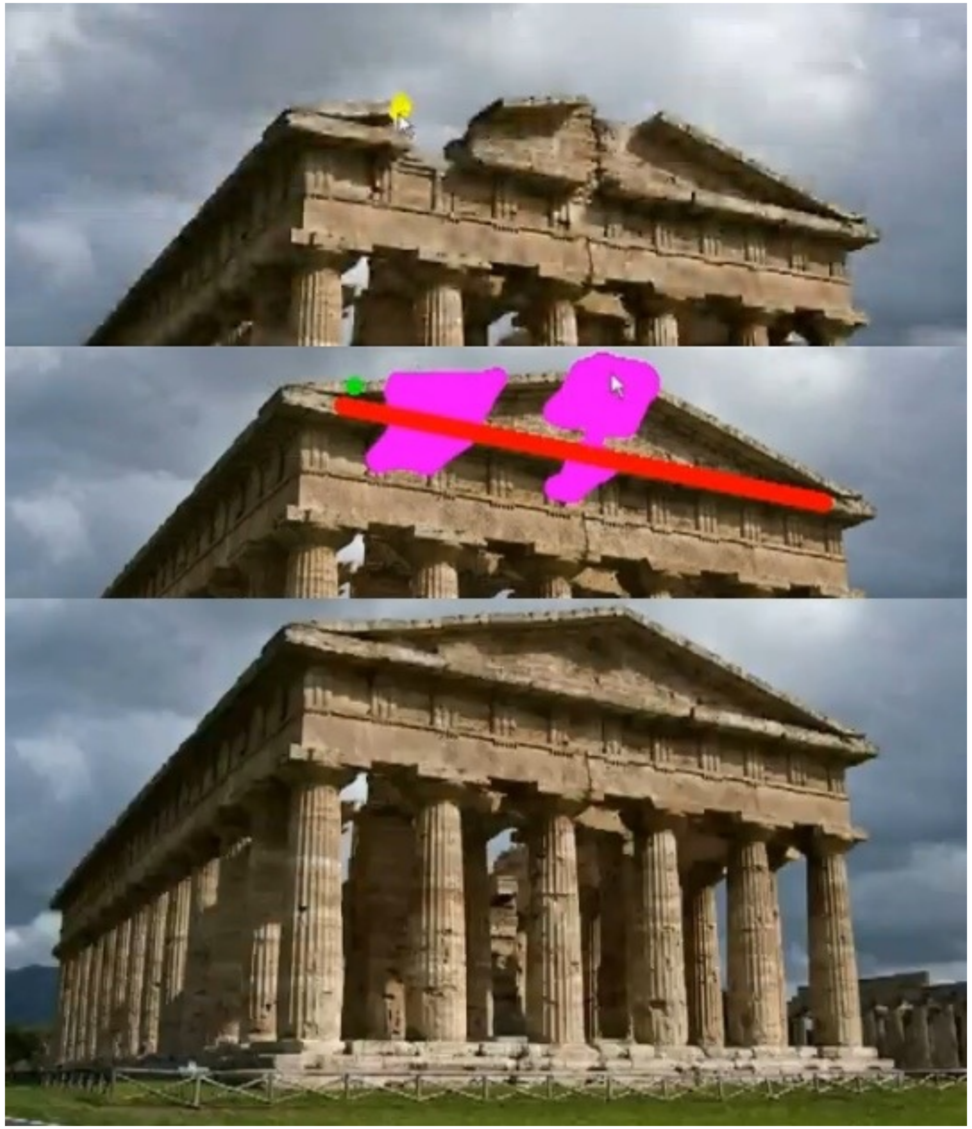
- 天气预报



Skilful precipitation nowcasting using deep generative models of radar, Nature 2021

GenAI时代之前的生成模型

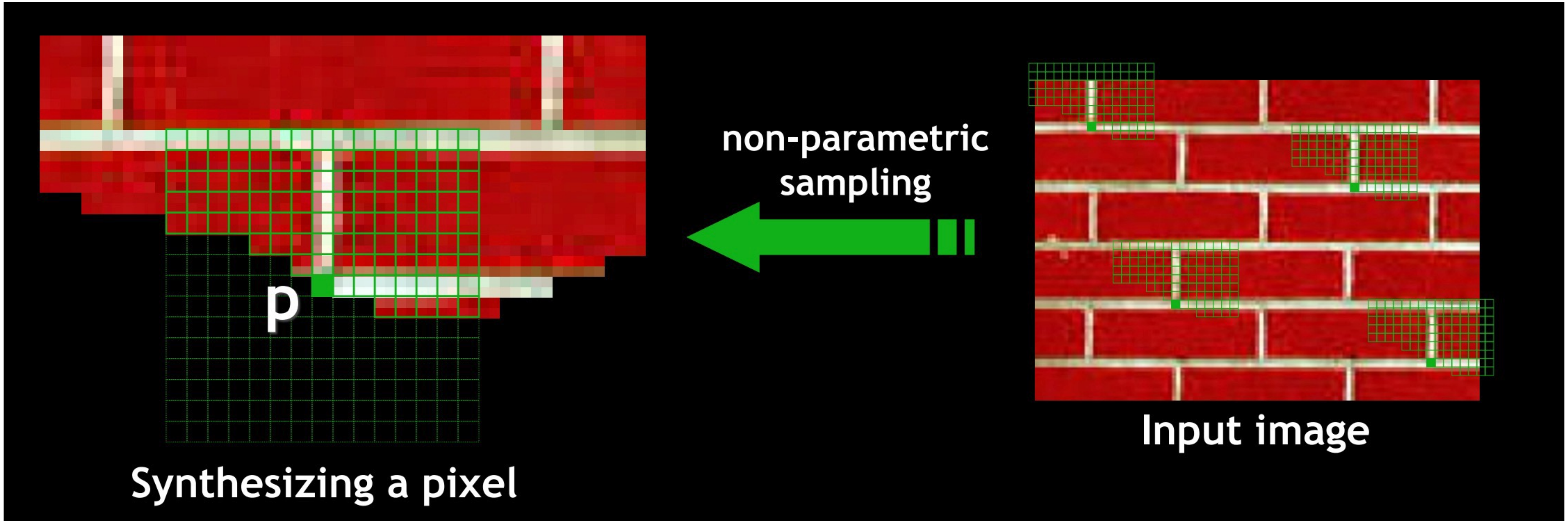
- 2009，PatchMatch
- Photoshop的内容填补功能



PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing, SIGGRAPH 2009

GenAI时代之前的生成模型

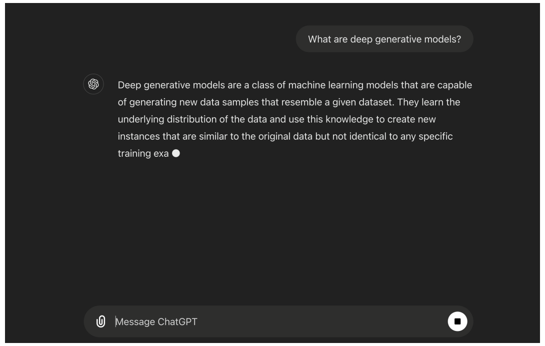
• 1999，纹理生成

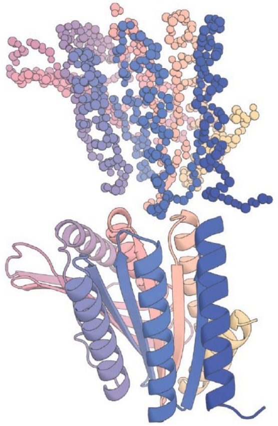


Texture Synthesis by Non-parametric Sampling, ICCV 1999

应用场景中的共性

- 一个输入对应**多个**或无限个预测结果。
- 某些预测比其他预测更“**合理**”。
- 训练数据中可能**不包含精确解。**
- 预测结果可能比输入**更复杂**、信息更丰富、维度更高。



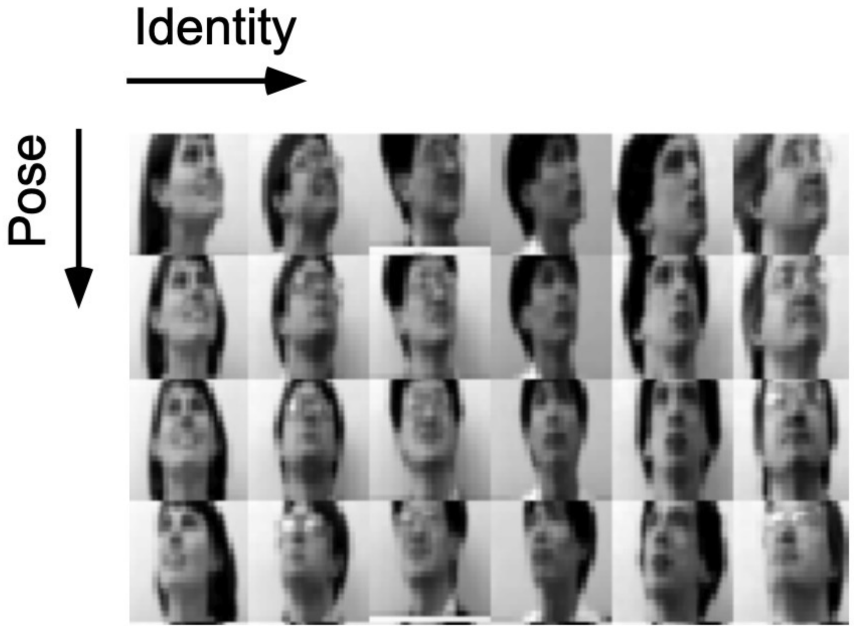


概率建模

- 概率的实际意义可以理解为**数据的生成过程。**

一个关于概率的例子

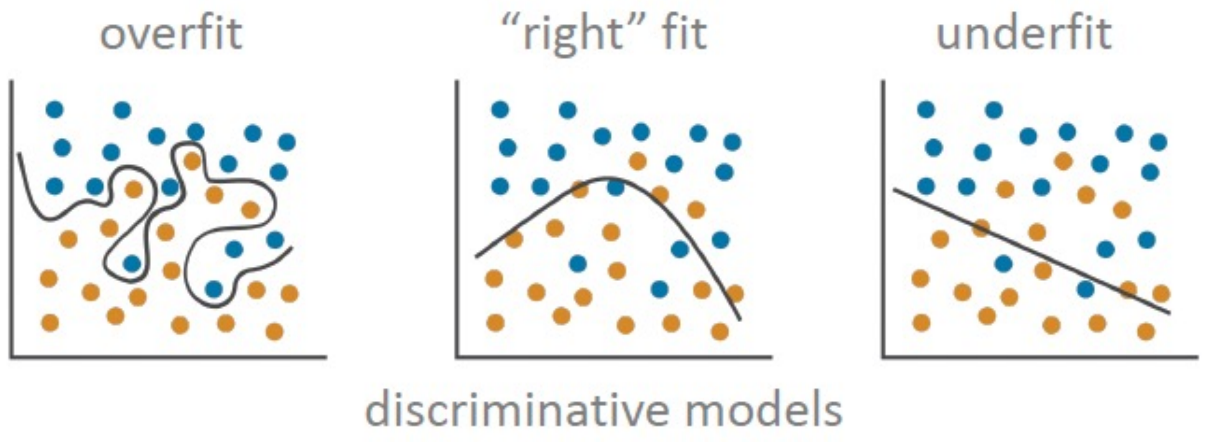
- x为观测到的图像
- 样本取决于隐藏因子 z(pose, Identity)
- x是由“世界模型”基于z来采样得到的



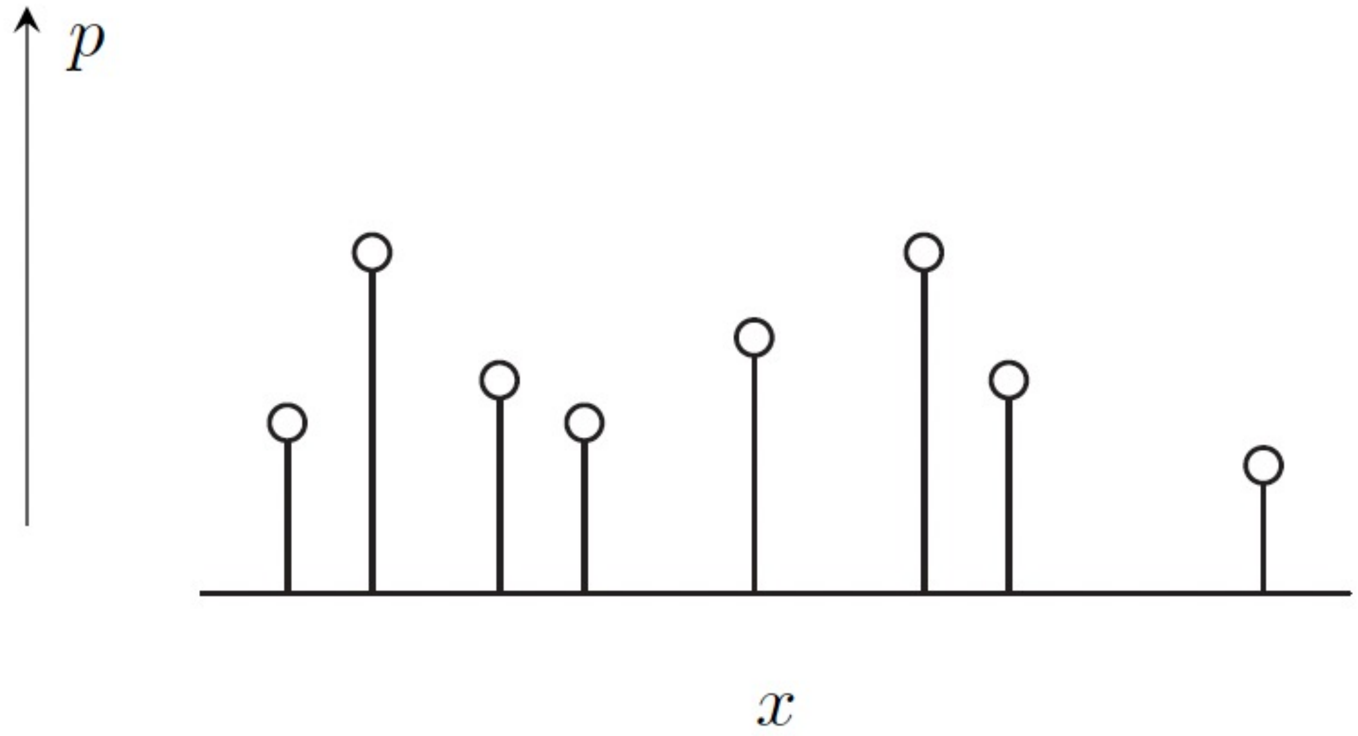
- 概率是建模的一部份。

概率是建模的一部份

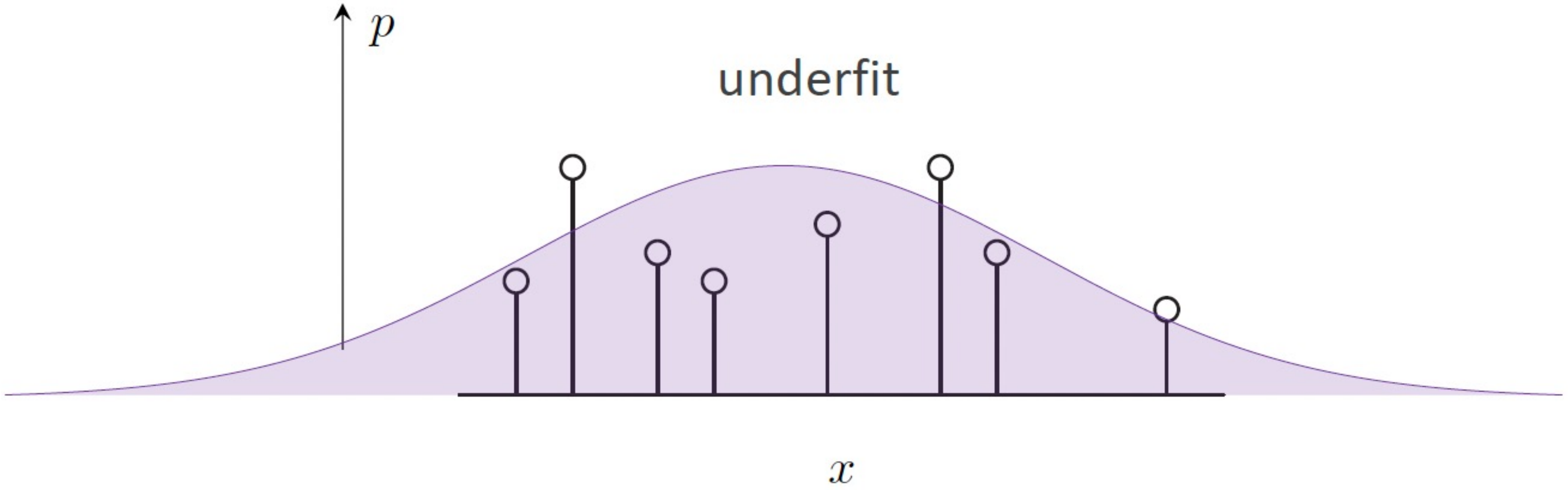
- 我们所能观测到的也只是有限的数据点集。
- 模型通过外推观测数据来建立分布模型。
- 过拟合与欠拟合问题：与判别模型类似。



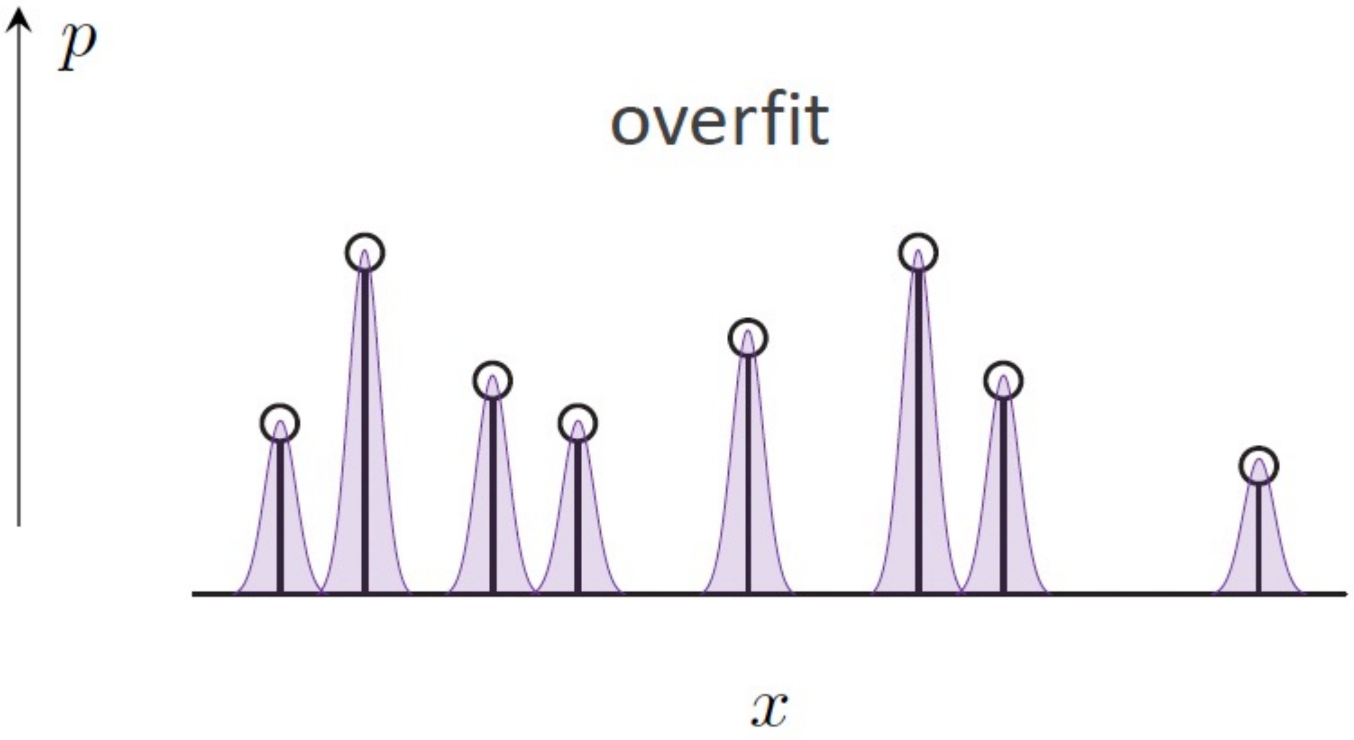
概率是建模的一部份



概率是建模的一部份



概率是建模的一部份



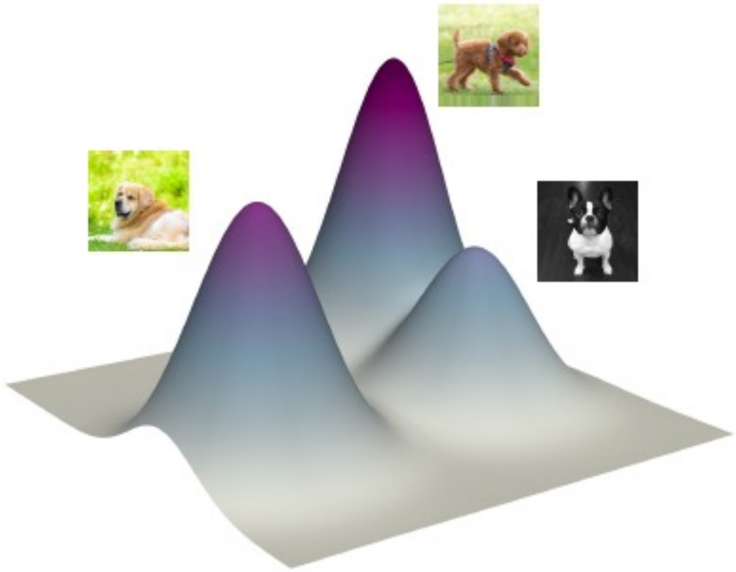
概率建模的生成模型

数据

概率建模的生成模型

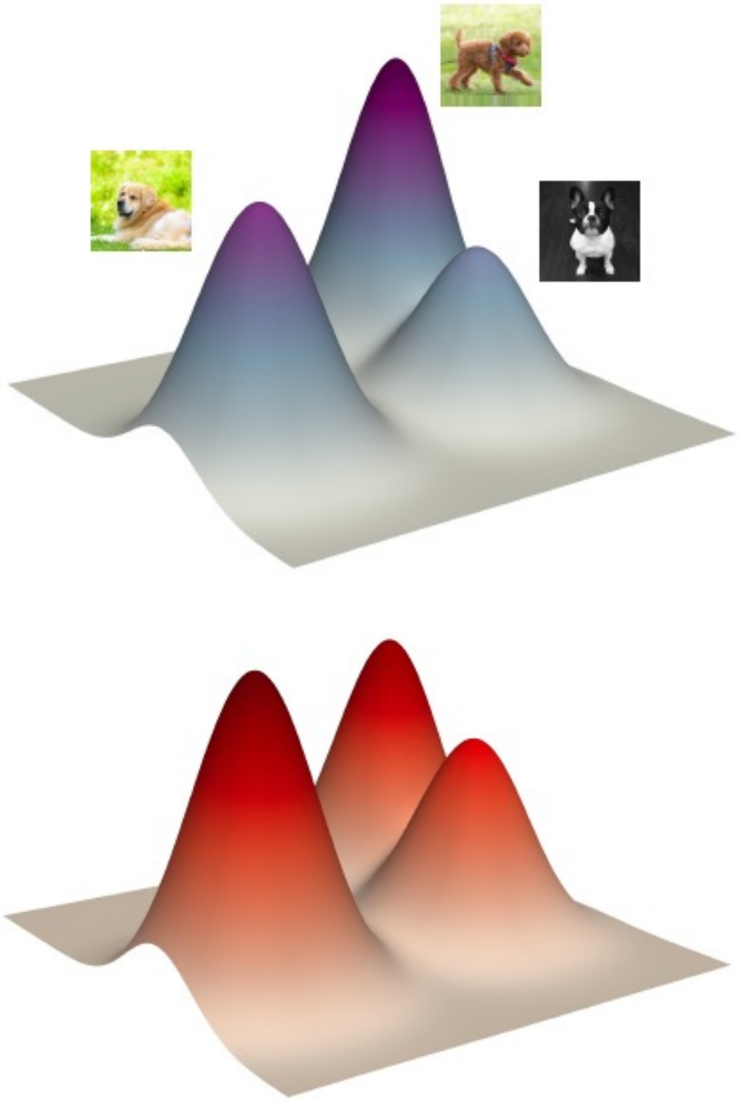
数据

数据分布



概率建模的生成模型

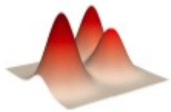
数据



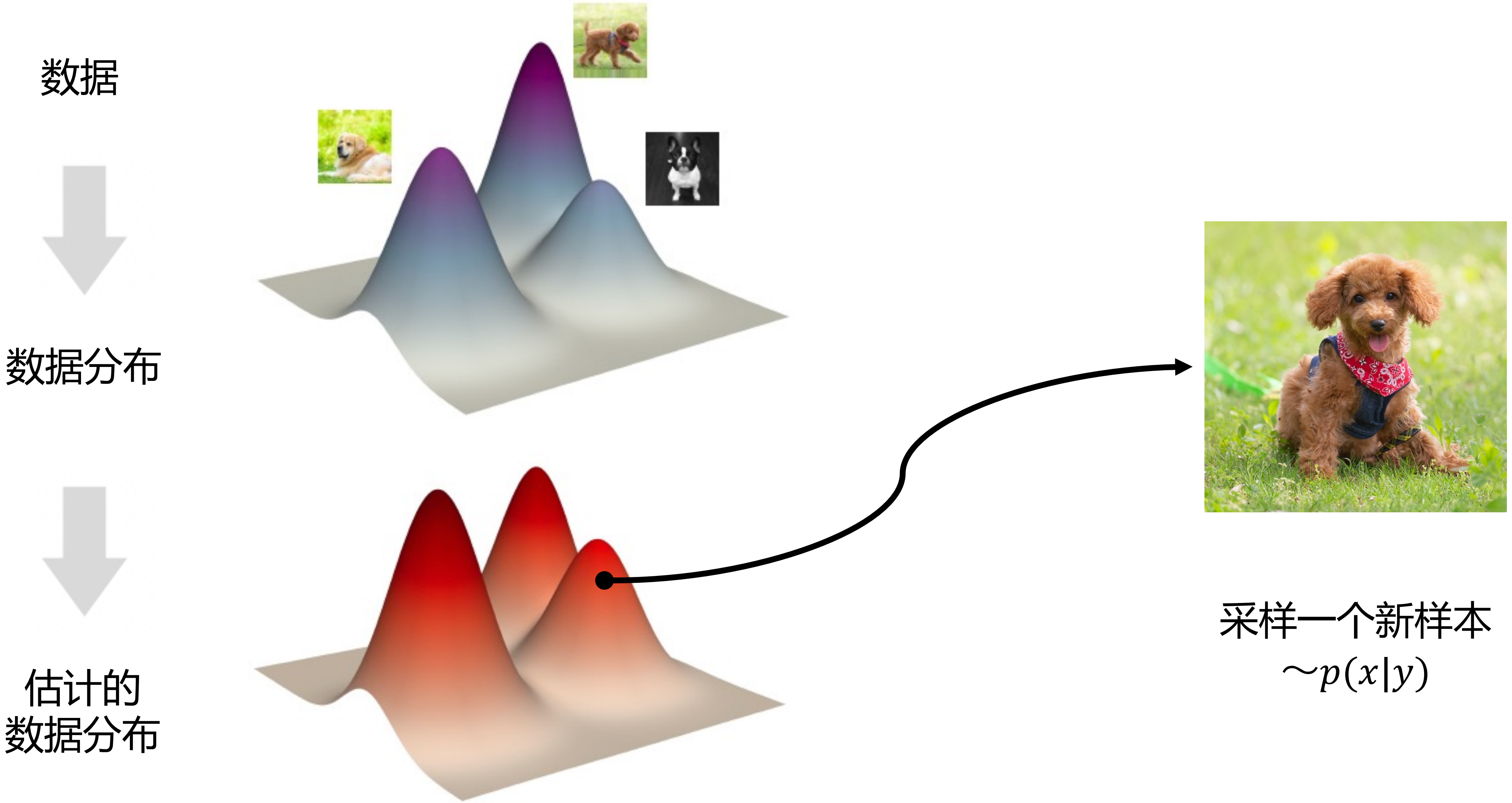
数据分布

估计的
数据分布

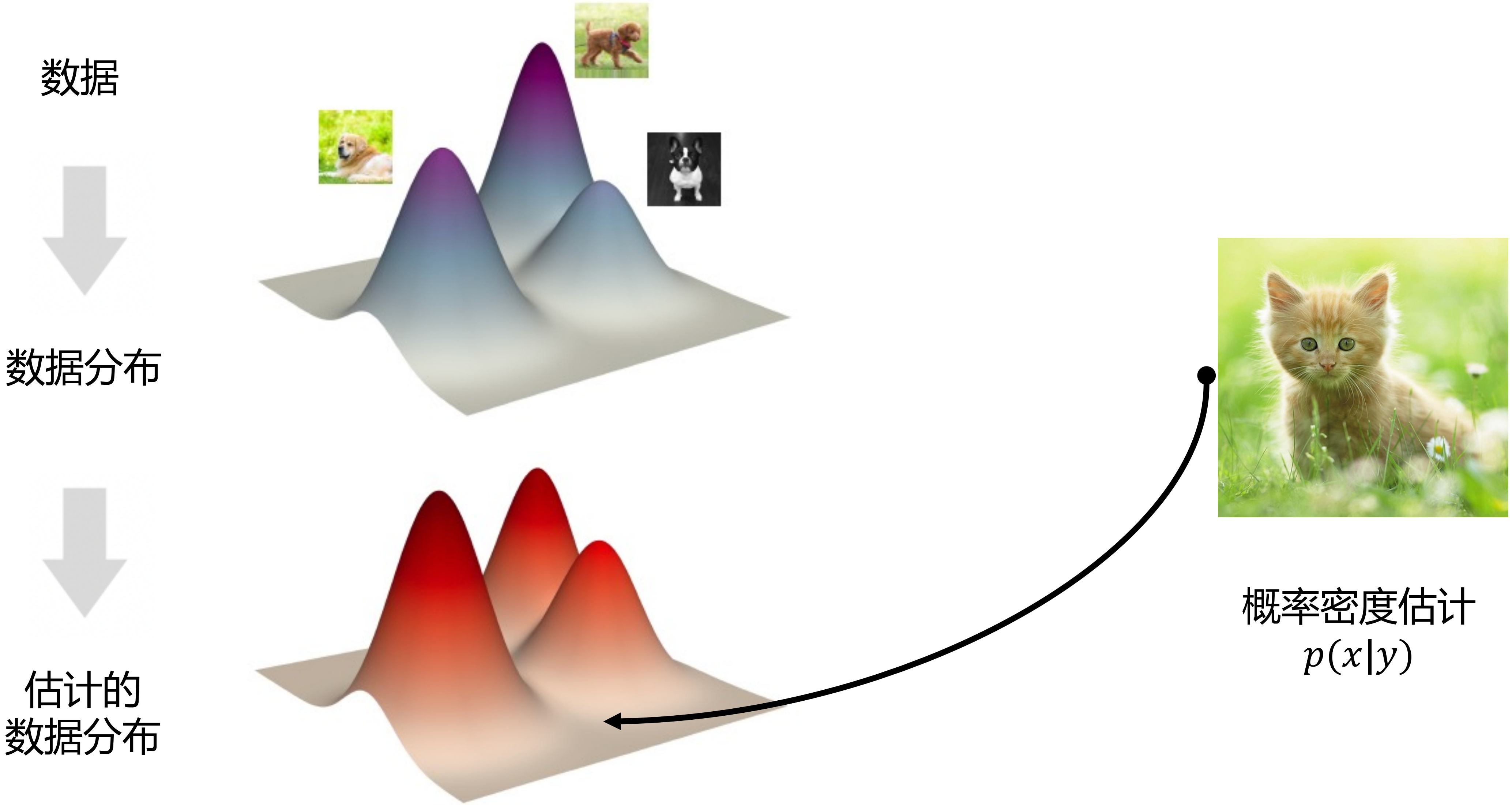
- 优化损失函数

$\mathcal{L}($  ,  $)$

概率建模的生成模型



概率建模的生成模型

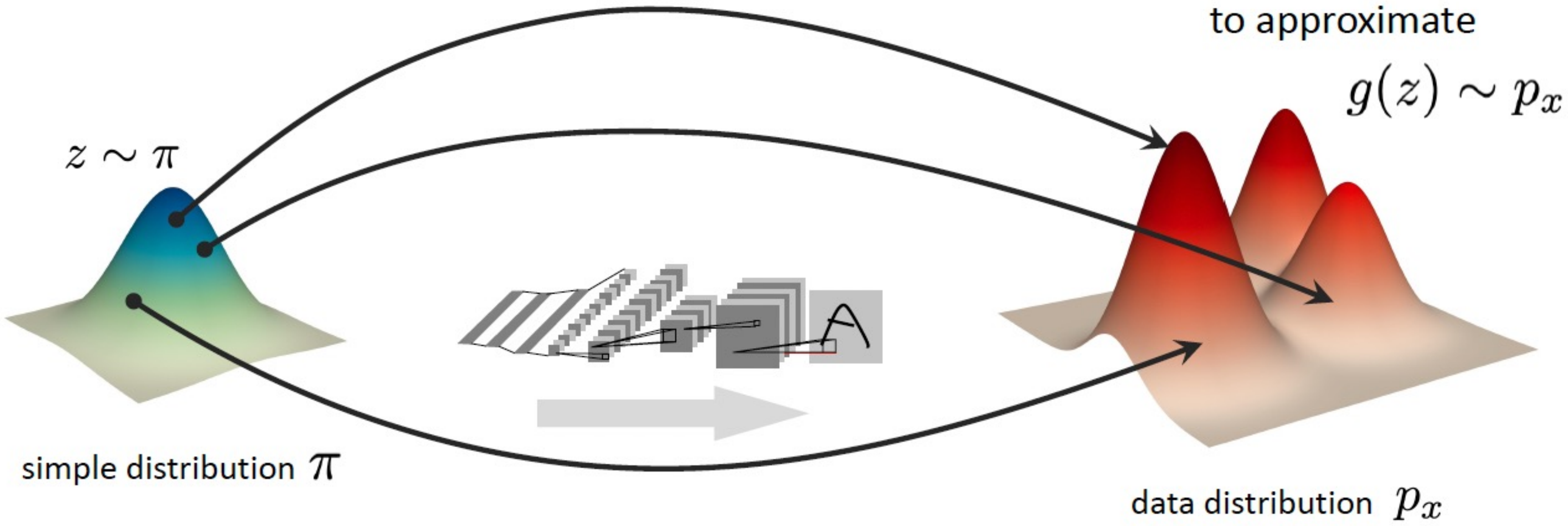


概率建模的生成模型

- 生成式模型通常基于人为设计和推导的统计模型。
- 概率建模并非神经网络专属。
- 概率建模虽是常用方法，但并非唯一途径。

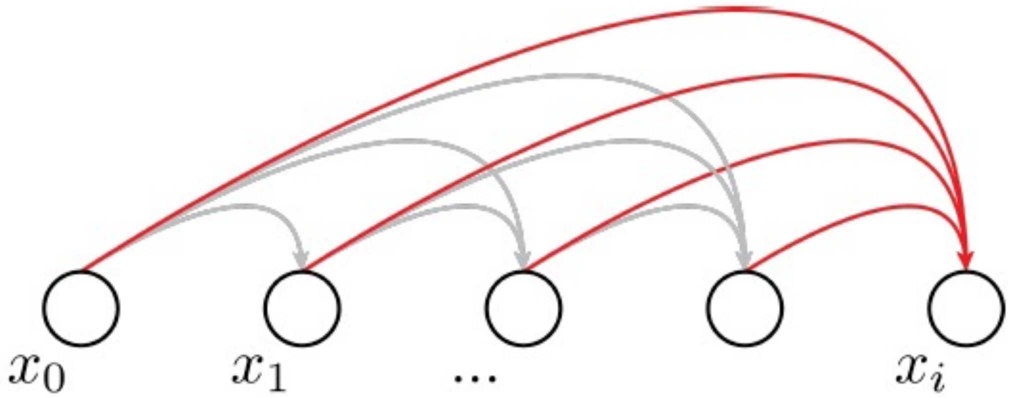
学习概率分布的表征

- 从简单分布（例如高斯分布）到复杂分布



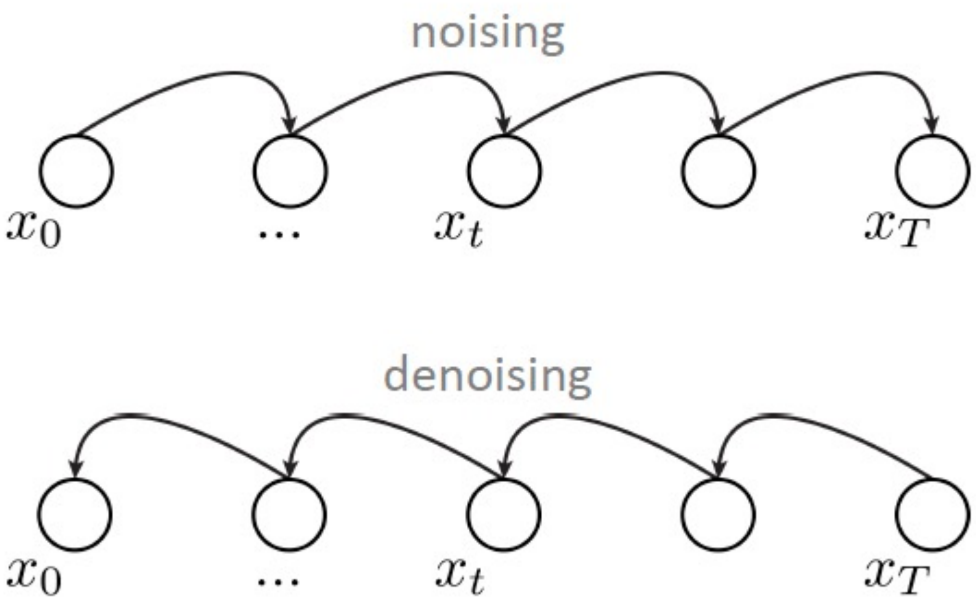
学习概率分布的表征

- 并不是分布建模的所有部分都是学习得到的
- **自回归模型**
  - 映射过程是学习的



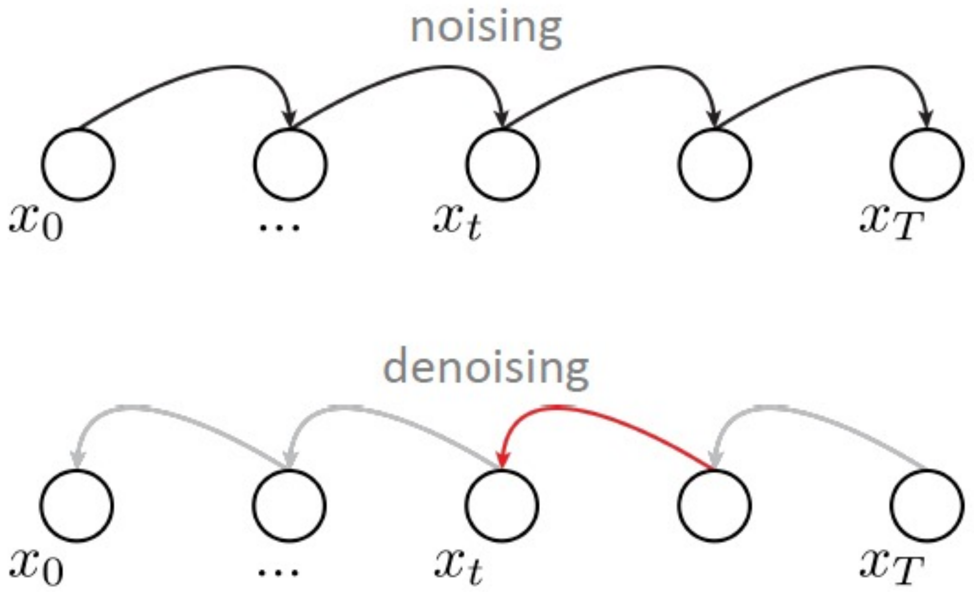
学习概率分布的表征

- 并不是分布建模的所有部分都是学习得到的
- **扩散模型**
  - 依赖关系图是人为定义的，不是学习的



学习概率分布的表征

- 并不是分布建模的所有部分都是学习得到的
- **扩散模型**
  - 映射过程是学习的



深度生成模型的组成部份

- **建模框架**：将问题构建为概率模型
- **表征方法**：使用深度神经网络表征数据及其分布
- **目标函数**：用于衡量预测分布的拟合优度
- **优化过程**：优化神经网络和/或分布分解方式
- **推断机制**：采样得到生成新样本

目录

1 生成模型基础

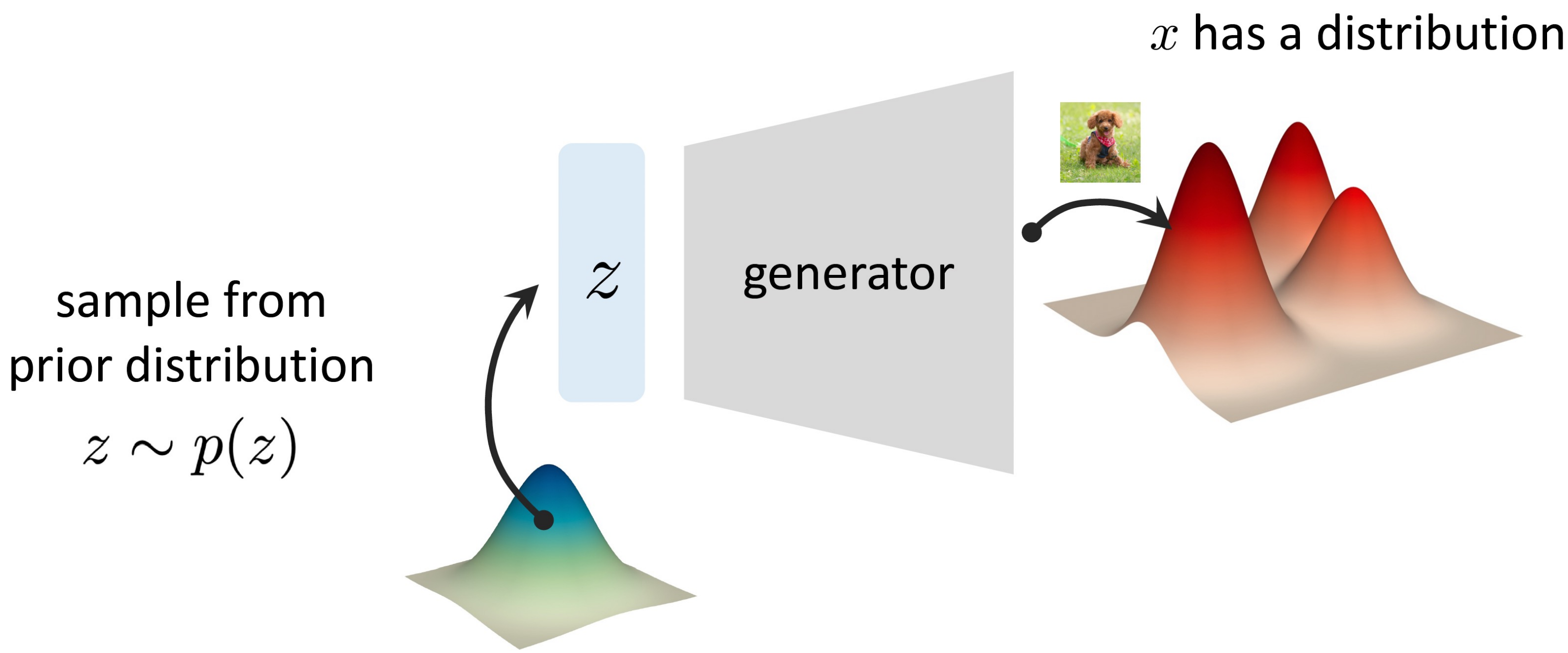
2 独立潜在变量建模

3 条件分布建模

4 现实问题建模

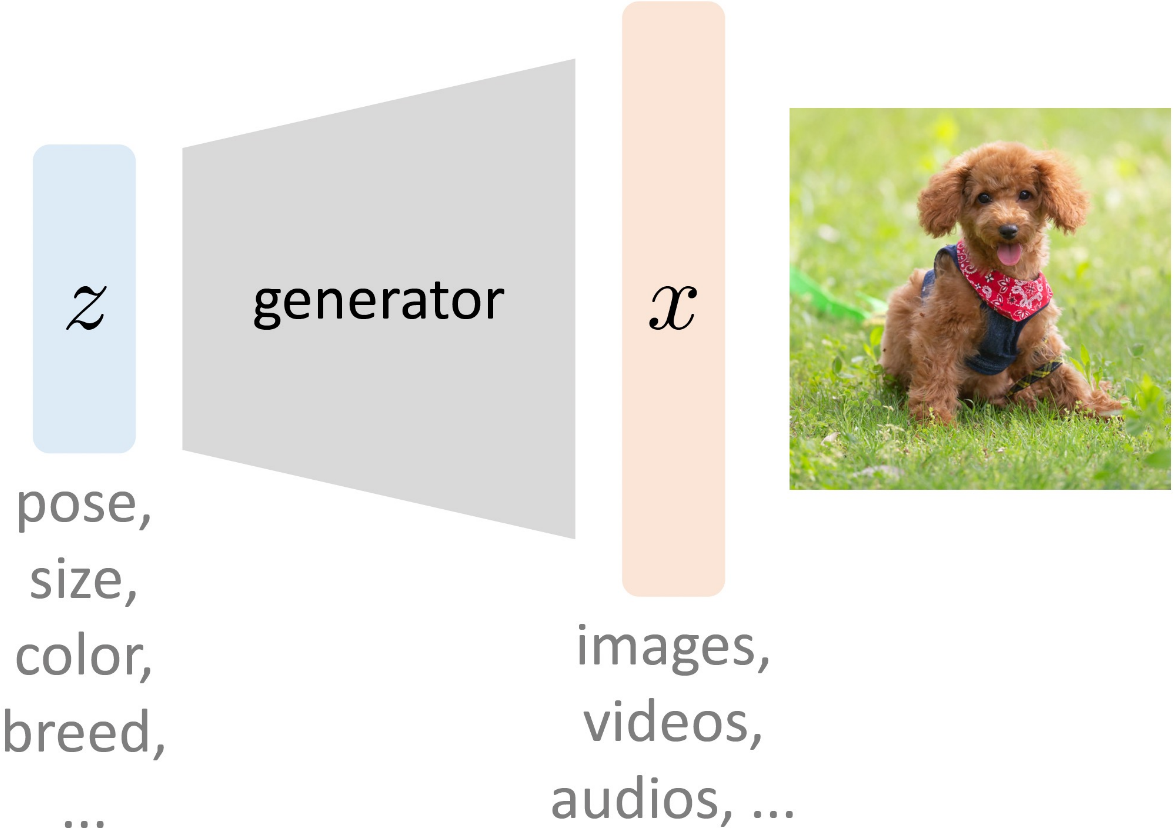
潜在变量模型

- z-潜在变量
- x-观测变量



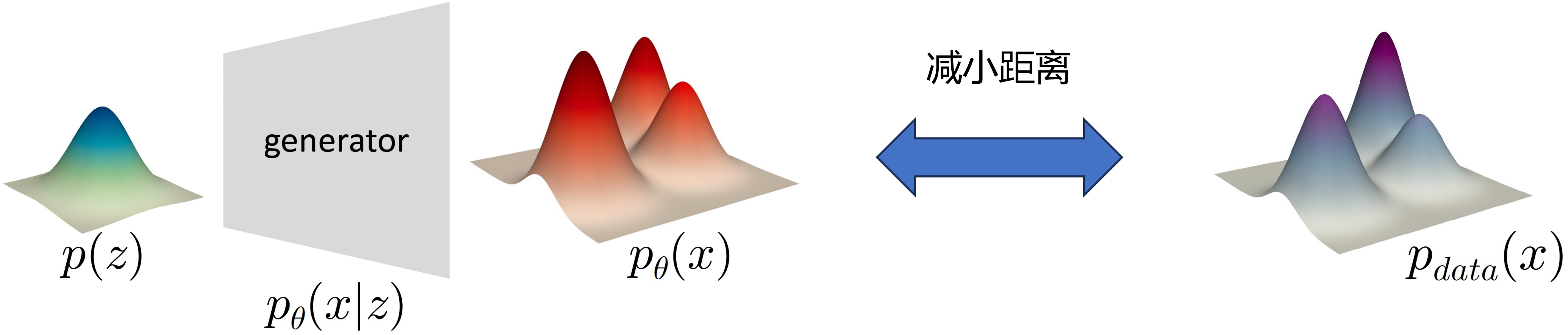
潜在变量模型

- z-潜在变量
- x-观测变量



潜在变量模型

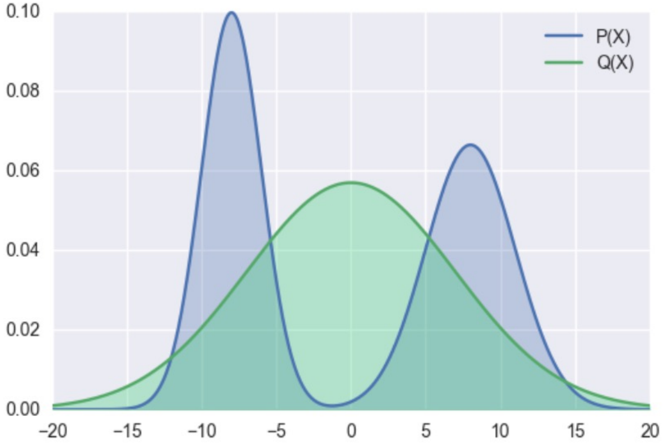
- 利用一个神经网络和参数p_θ来学习映射
- 目标是让学习到的分布和目标分布更接近



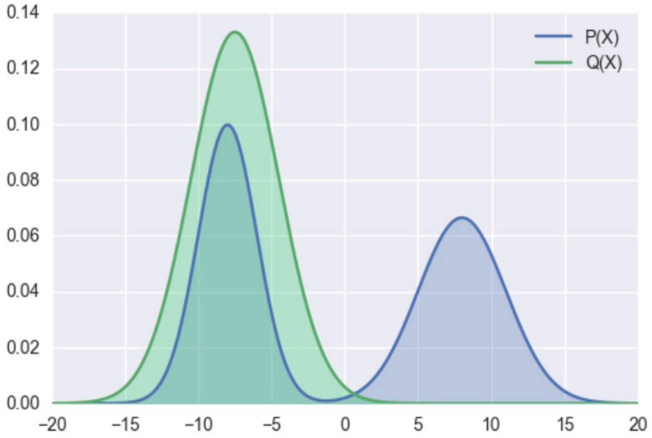
KL散度

- 衡量估计分布Q到**真实分布P**的距离
- KL散度是不对称的，需要对所有**真实分布P**进行积分

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}.$$



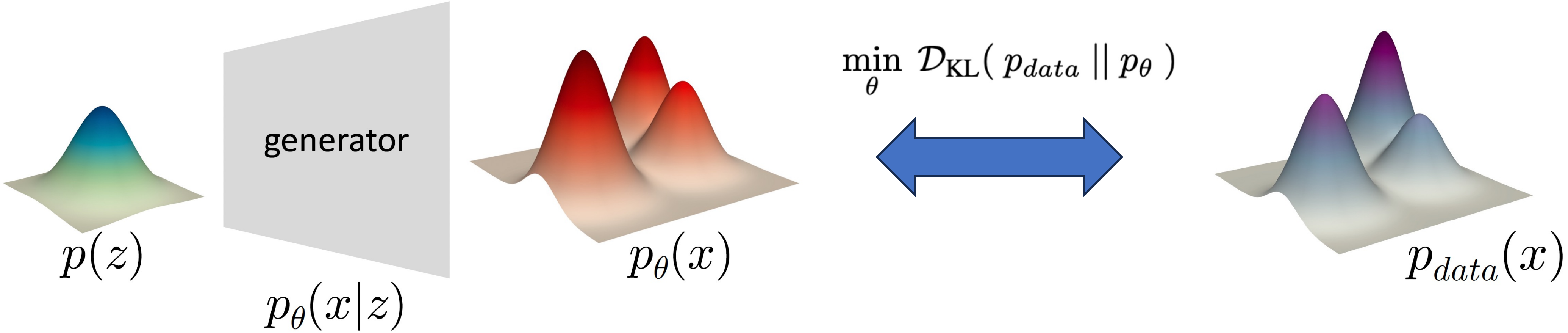
KL散度大



KL散度小

潜在变量模型

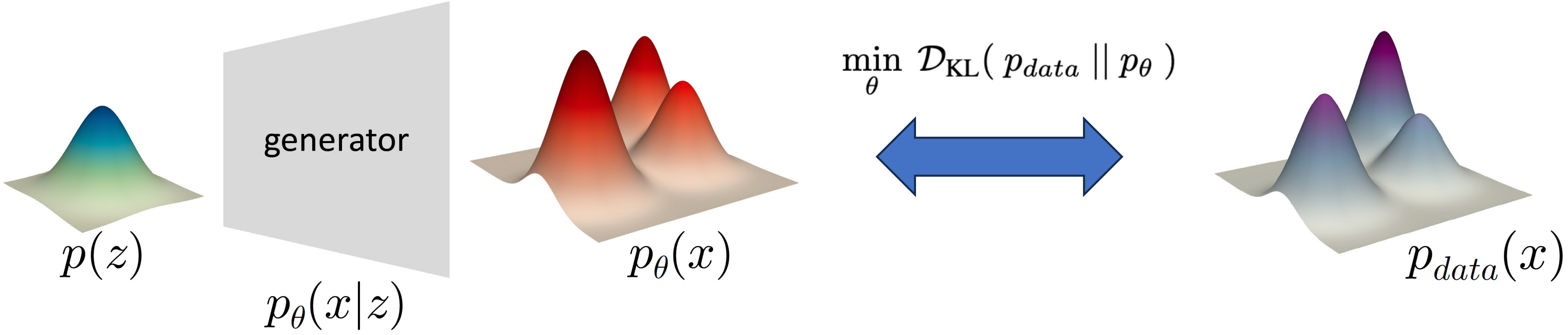
- 利用一个神经网络和参数p_θ来学习映射
- 目标是让学习到的分布和目标分布更接近
- ⇒减小分布间的Kullback-Leibler (KL)散度 $\min_\theta \mathcal{D}_{\mathrm{KL}}(\, p_{data} \,||\, p_\theta \,)$



潜在变量模型

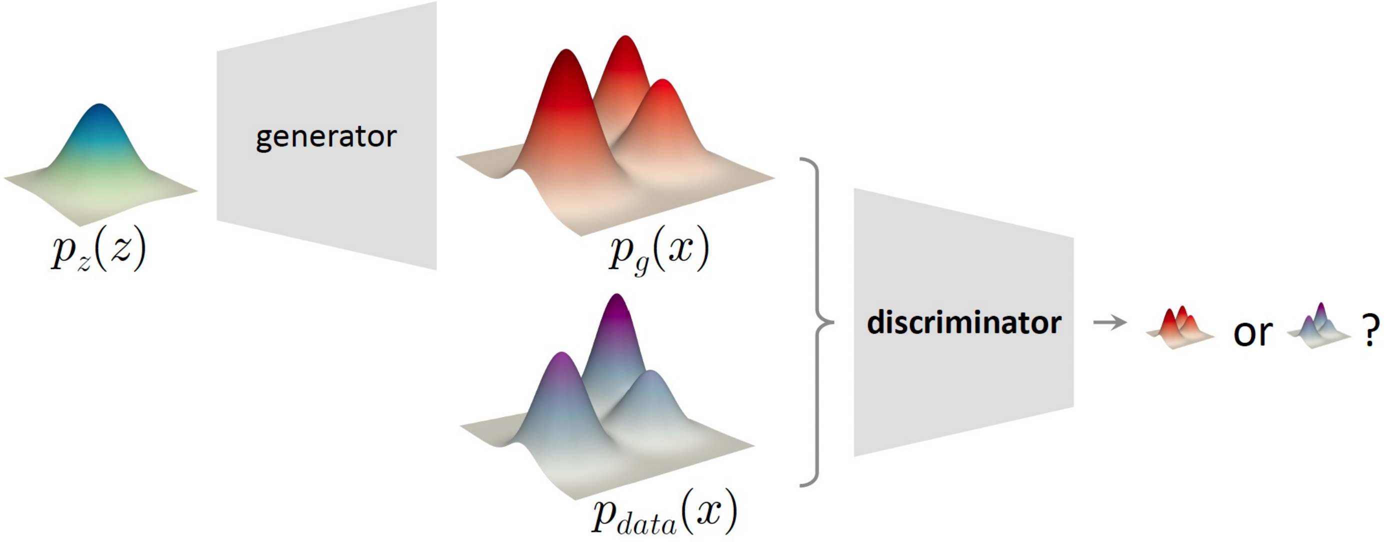
- 利用一个神经网络和参数p_θ来学习映射
- 目标是让学习到的分布和目标分布更接近
- ⇒减小分布间的Kullback-Leibler (KL)散度 $\min_\theta \mathcal{D}_{\mathrm{KL}}(\, p_{data} \,||\, p_\theta \,)$
- ⇒估计分布的最大似然估计 $\max_\theta \mathbb{E}_{x\sim p_{data}} \log p_\theta(x)$

$$
\begin{aligned}
& \arg\min_\theta \ \mathcal{D}_{\mathrm{KL}}(\, p_{data} \,||\, p_\theta \,) \\
= \ & \arg\min_\theta \sum_x p_{data}(x) \log \frac{p_{data}(x)}{p_\theta(x)} \\
= \ & \arg\min_\theta \sum_x -p_{data}(x) \log p_\theta(x) + const \\
= \ & \arg\max_\theta \sum_x p_{data}(x) \log p_\theta(x) \\
= \ & \arg\max_\theta \ \mathbb{E}_{x\sim p_{data}} \log p_\theta(x)
\end{aligned}
$$

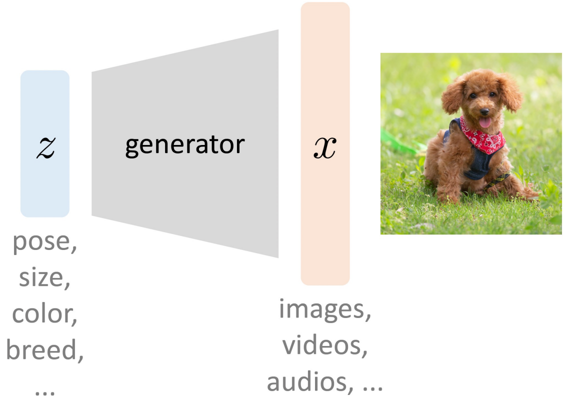


潜在变量模型

- 高维数据的最大似然估计 $\max_{\theta} \mathbb{E}_{x \sim p_{data}} \log p_{\theta}(x)$ 很难
- 可以用神经网络来表征分布差异，e.g. Generative Adversarial Net



- 但没有编码能力，e.g. 隐空间可解释性差

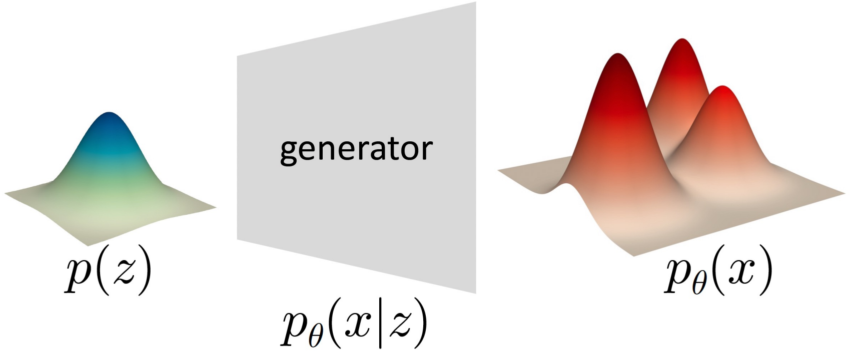


潜在变量模型

- 高维数据的最大似然估计 $\max_{\theta} \mathbb{E}_{x\sim p_{data}} \log p_\theta(x)$ 很难
- 从贝叶斯的角度$p_\theta\ (x)$可以表示为

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

- 但真实概率$p(z)$是难以控制的



- 变分推断（Variational Autoencoder，VAE）的核心思想：化“算不出来” $p(z)$为“找替代品”： $q(z)$

潜在变量模型

$$
\begin{aligned}
&\log p_\theta(x) \\
=\ & \int_z q(z) \log p_\theta(x) dz \\
=\ & \int_z q(z) \log \left(\frac{p_\theta(x|z) p_\theta(z)}{p_\theta(z|x)} \right) dz \\
=\ & \int_z q(z) \log \left(\frac{p_\theta(x|z) p_\theta(z)}{p_\theta(z|x)} \frac{q(z)}{q(z)} \right) dz \\
=\ & \int_z q(z) \left(\log p_\theta(x|z) + \log \frac{p_\theta(z)}{q(z)} + \log \frac{q(z)}{p_\theta(z|x)} \right) dz \\
=\ & \mathbb{E}_{z \sim q(z)} \Big[\log p_\theta(x|z) \Big] - \mathcal{D}_{\mathrm{KL}} \Big(q(z) || p_\theta(z) \Big) + \mathcal{D}_{\mathrm{KL}} \Big(q(z) || p_\theta(z|x) \Big)
\end{aligned}
$$

Rewrite log likelihood by latent z

- for any distribution $q(z)$
- Bayes' rule
- just algebra
- just algebra

潜在变量模型

intractable $\boxed{\log p_\theta(x)}$

Rewrite log likelihood by latent z

$$= \int_z q(z)\log p_\theta(x)dz$$

- for any distribution $q(z)$

$$= \int_z q(z)\log\left(\frac{p_\theta(x|z)p_\theta(z)}{p_\theta(z|x)}\right)dz$$

- Bayes' rule

$$= \int_z q(z)\log\left(\frac{p_\theta(x|z)p_\theta(z)}{p_\theta(z|x)}\frac{q(z)}{q(z)}\right)dz$$

$$= \int_z q(z)\left(\log p_\theta(x|z) + \log\frac{p_\theta(z)}{q(z)} + \log\frac{q(z)}{p_\theta(z|x)}\right)dz$$

$$= \underbrace{\boxed{\mathbb{E}_{z\sim q(z)}\Big[\log p_\theta(x|z)\Big]}}_{\text{tractable}}\ \underbrace{\boxed{-\mathcal{D}_{\mathrm{KL}}\Big(q(z)||p_\theta(z)\Big)}}_{\text{tractable}} + \underbrace{\boxed{\mathcal{D}_{\mathrm{KL}}\Big(q(z)||p_\theta(z|x)\Big)}}_{\text{intractable}}$$

潜在变量模型

- 把tractable的变量放在一起

$$\underbrace{\boxed{\log p_\theta(x)}}_{\text{intractable}} \underbrace{\boxed{-\mathcal{D}_{\mathrm{KL}}\Big(q(z)||p_\theta(z|x)\Big)}}_{\text{intractable}}$$

$$= \underbrace{\underbrace{\boxed{\mathbb{E}_{z\sim q(z)}\Big[\log p_\theta(x|z)\Big]}}_{\text{tractable}} \underbrace{\boxed{-\mathcal{D}_{\mathrm{KL}}\Big(q(z)||p_\theta(z)\Big)}}_{\text{tractable}}}$$

- 这个就是Evidence Low Bound (ELBO)，是可以优化的
- 同时因为KL散度总是正的，ELBO也是$\log p_\theta\ (x)$的下界

潜在变量模型

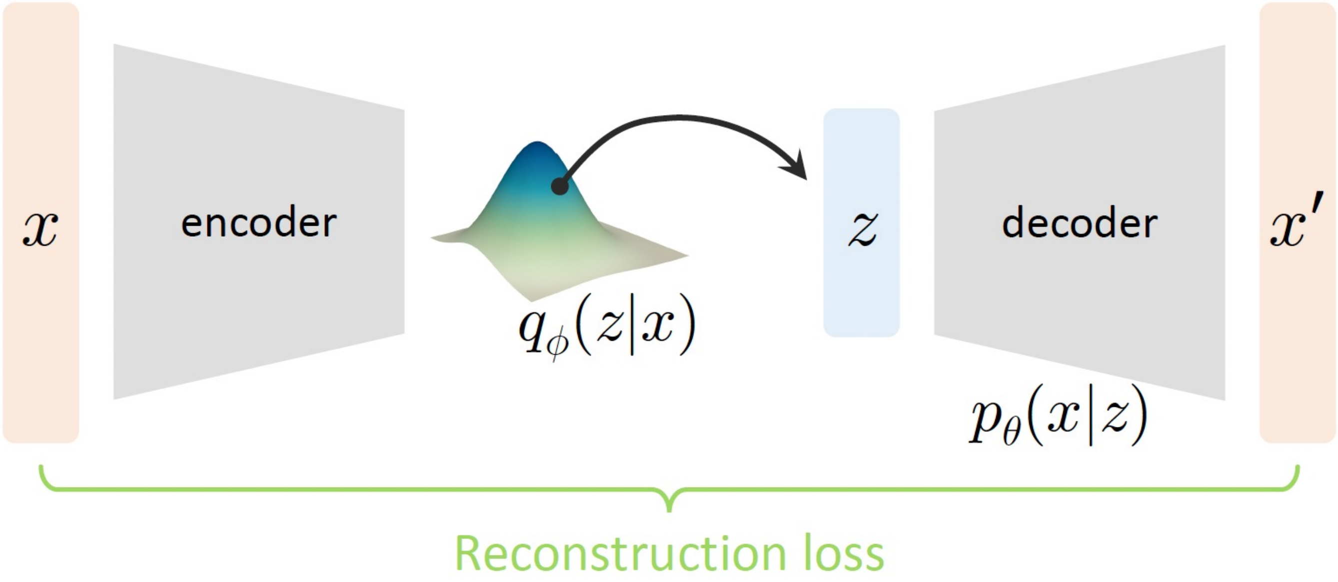
- 最后一步，为了让ELBO能直接优化
- 1、将$q(z)$进一步参数化为$q_\phi(z|x)$
- 2、简化$p_\theta(z)$为一个已知的简单先验$p(z)$

$$\underbrace{\boxed{\mathbb{E}_{z\sim \overset{q_\phi(z|x)}{\cancel{q(z)}}}\left[\log p_\theta(x|z)\right]}}_{\text{tractable}}\ \underbrace{\boxed{- \mathcal{D}_{\mathrm{KL}}\left(\overset{q_\phi(z|x)}{\cancel{q(z)}} \,||\, \overset{p(z)}{\cancel{p_\theta(z)}}\right)}}_{\text{tractable}}$$

VAE

- ELBO作为目标函数：重建损失
- 利用L2损失来减少距离

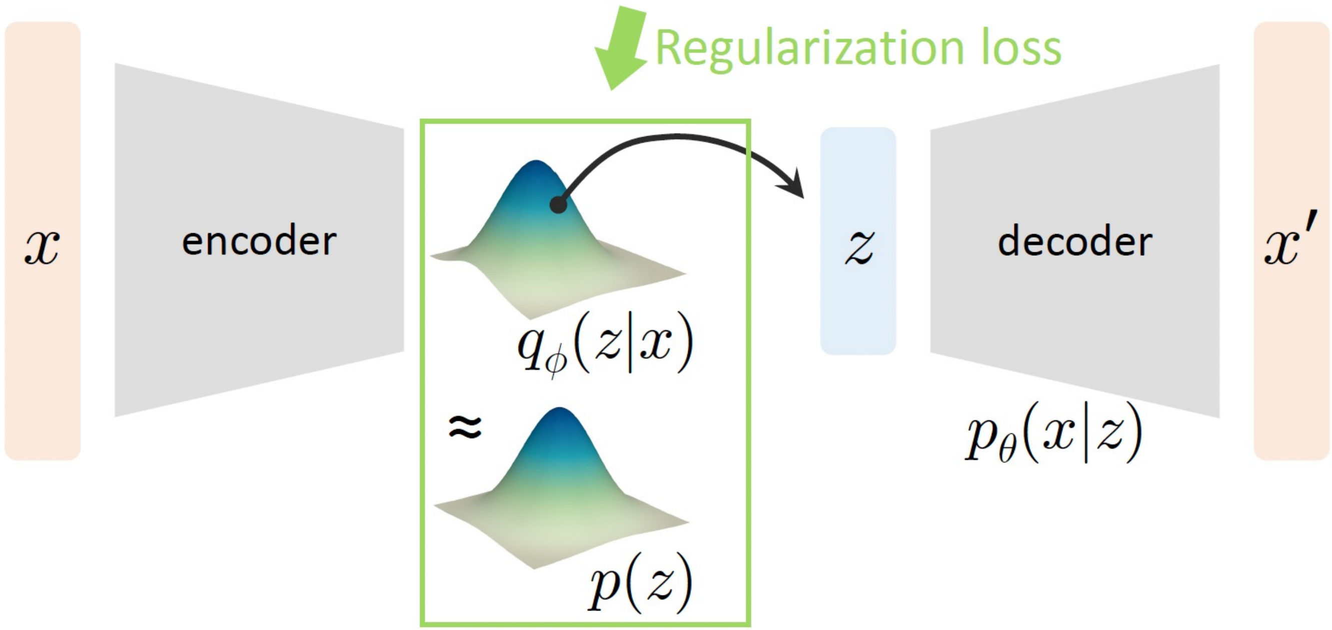
$$\mathcal{L}_{\theta,\phi}(x) = \boxed{-\mathbb{E}_{z\sim q_\phi(z|x)}\left[\log p_\theta(x|z)\right]} + \mathcal{D}_{\mathrm{KL}}\Big(q_\phi(z|x)||p(z)\Big)$$



VAE

- ELBO作为目标函数：正则化损失
- 常用选择为高斯先验，即 $p(z) = \mathcal{N}(z \mid 0, \mathbf{I})$

$$\mathcal{L}_{\theta,\phi}(x) = -\mathbb{E}_{z\sim q_\phi(z|x)}\left[\log p_\theta(x|z)\right] + \boxed{\mathcal{D}_{\mathrm{KL}}\left(q_\phi(z|x)||p(z)\right)}$$



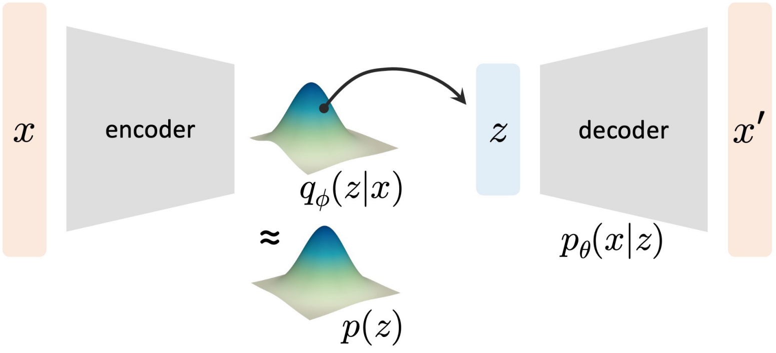
VAE

- ELBO作为目标函数：正则化损失
- 常用选择为高斯先验，即 $p(z) = \mathcal{N}(z \mid 0, \mathbf{I})$
- 将$q_\theta\ (z|x)$建模为高斯 $\mathcal{N}(z \mid \mu, \sigma)$
- 同时建模均值和方差

$$f_\phi(x) \rightarrow \mu, \sigma$$

- 计算与高斯分布的KL散度

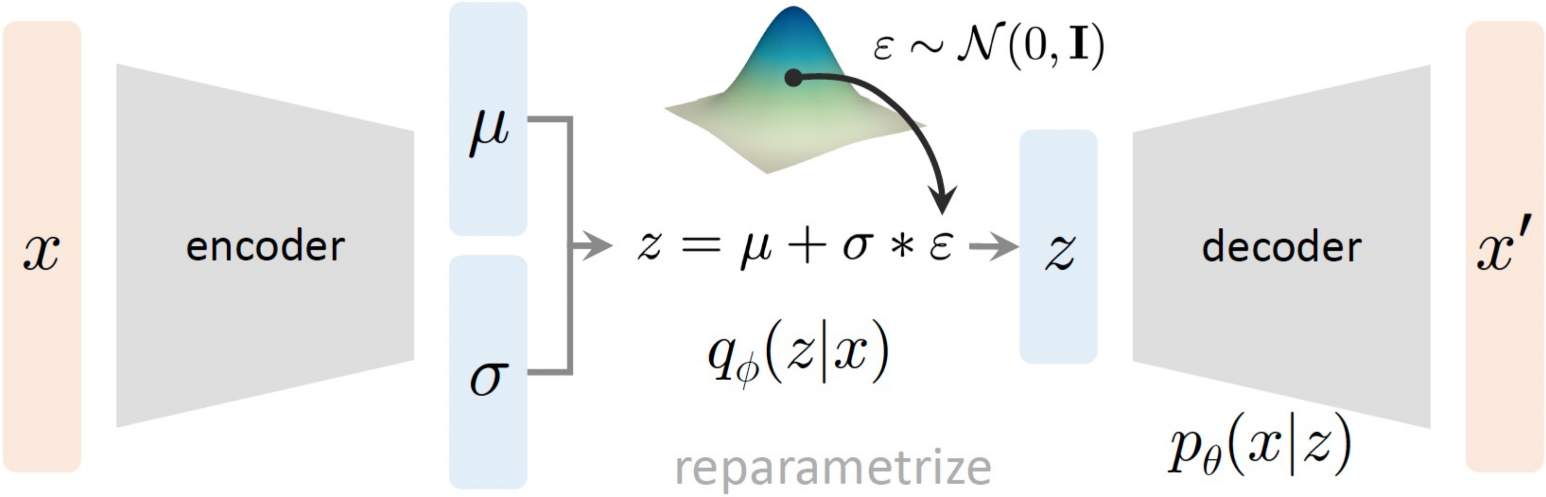
$$\mathcal{D}_{\mathrm{KL}}\Big(\mathcal{N}(z \mid \mu, \sigma) \,||\, \mathcal{N}(z \mid 0, \mathbf{I})\Big)$$



$$\mathcal{D}_{\mathrm{KL}}\Big(q_\phi(z|x)||p(z)\Big)$$

VAE

- 在训练的时候利用重参数化进行采样

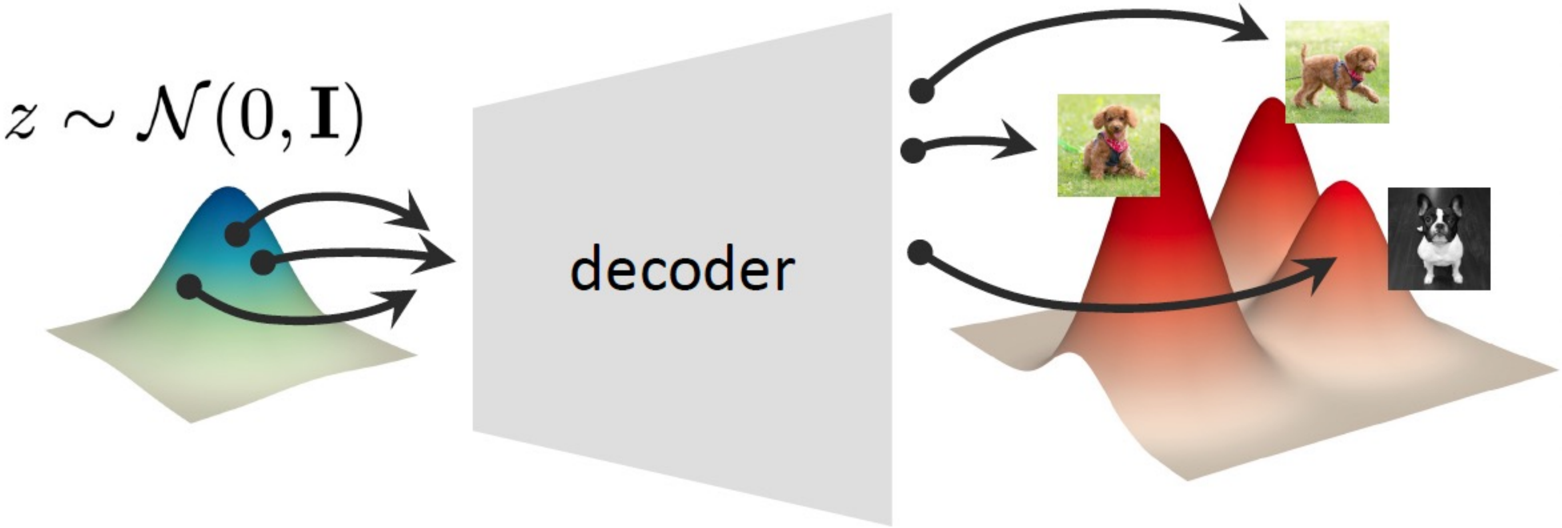


- 就可以得到VAE的训练目标

$$\mathcal{L}_{\theta,\phi} = \mathbb{E}_{x \sim p_{data}(x)} \left[-\mathbb{E}_{z \sim q_\phi(z|x)} \left[\log p_\theta(x|z) \right] + \mathcal{D}_{\text{KL}} \left(q_\phi(z|x) || p(z) \right) \right]$$

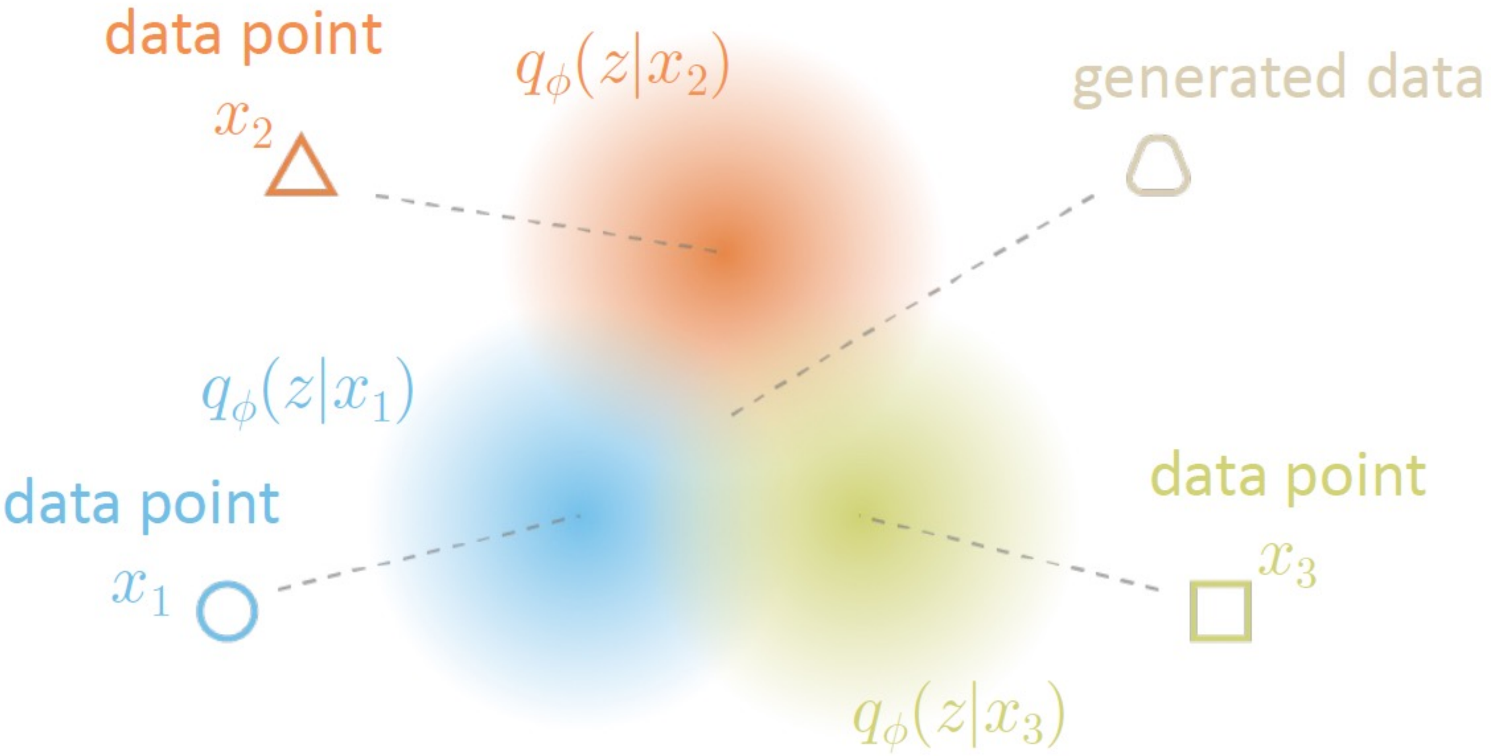
VAE

- 在推理的时候，只需要后半部分，从高斯分布中采样



VAE

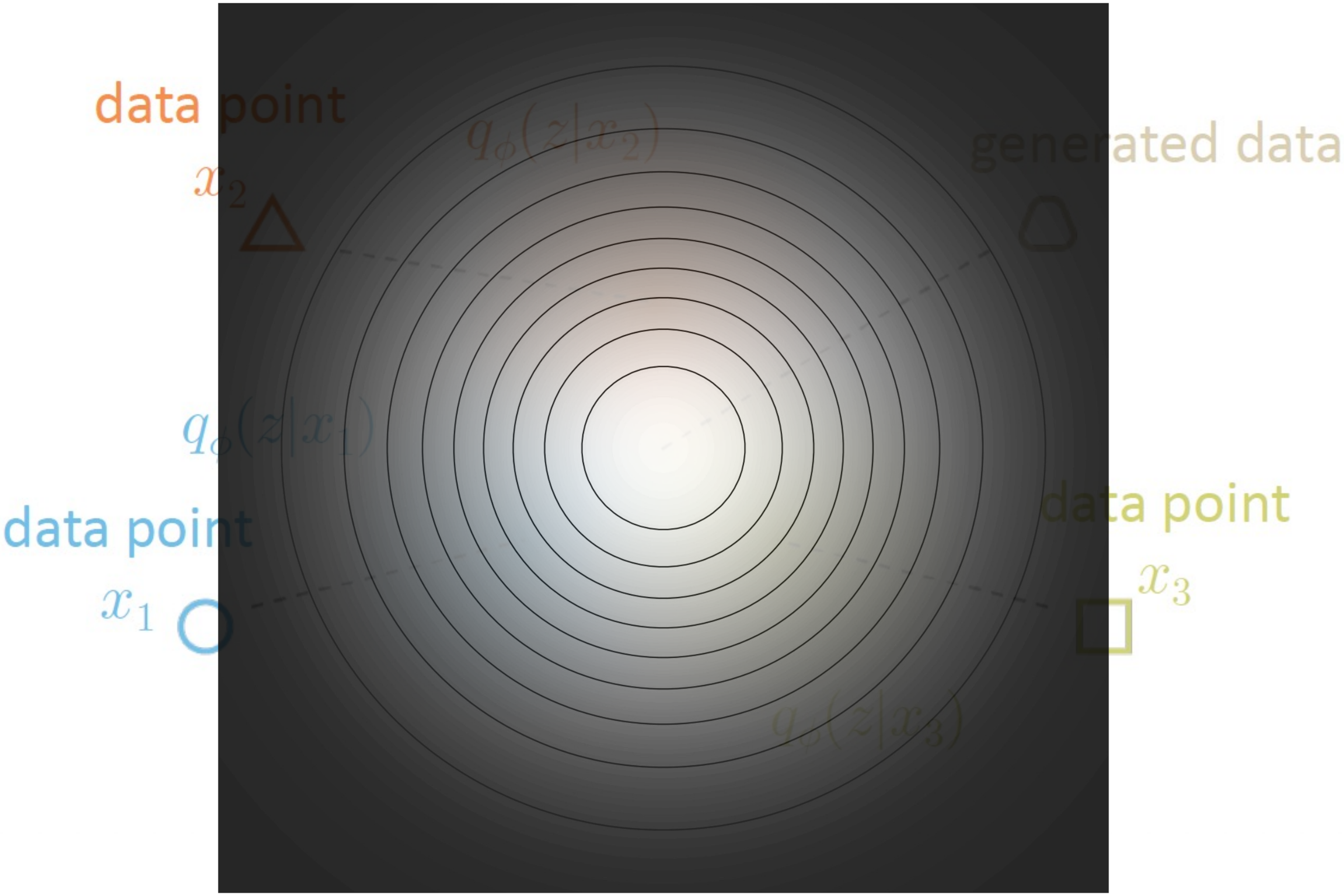
- 建模数据分布

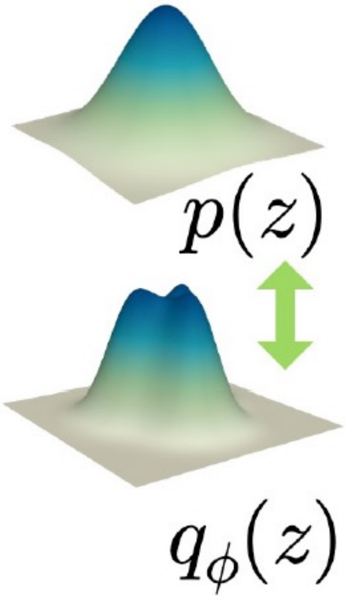


data point
$q_\phi(z|x_2)$
generated data
x_2
$q_\phi(z|x_1)$
data point
x_1
data point
x_3
$q_\phi(z|x_3)$

VAE

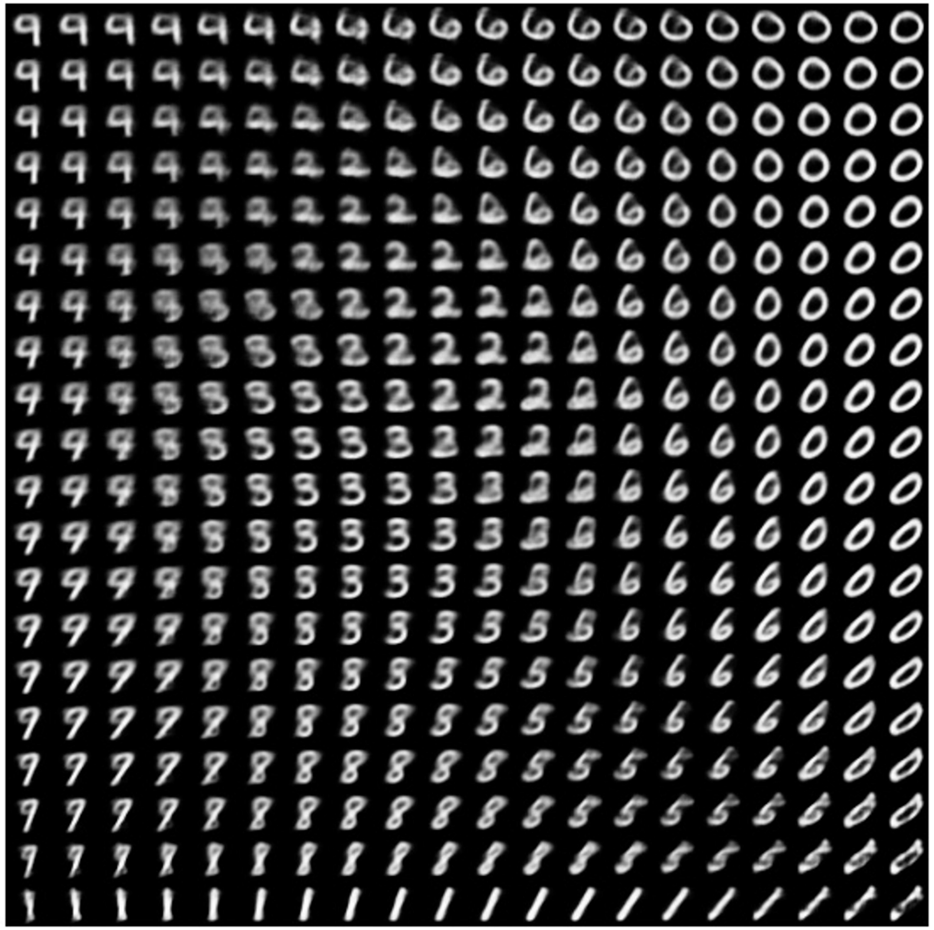
- 简化分布可以让隐空间连续，可差值





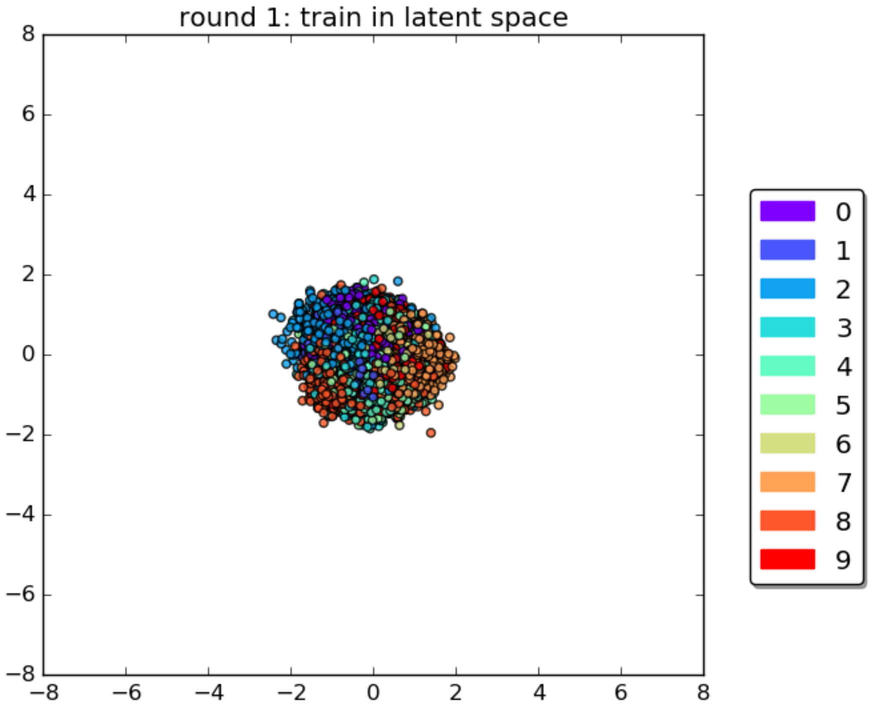
VAE

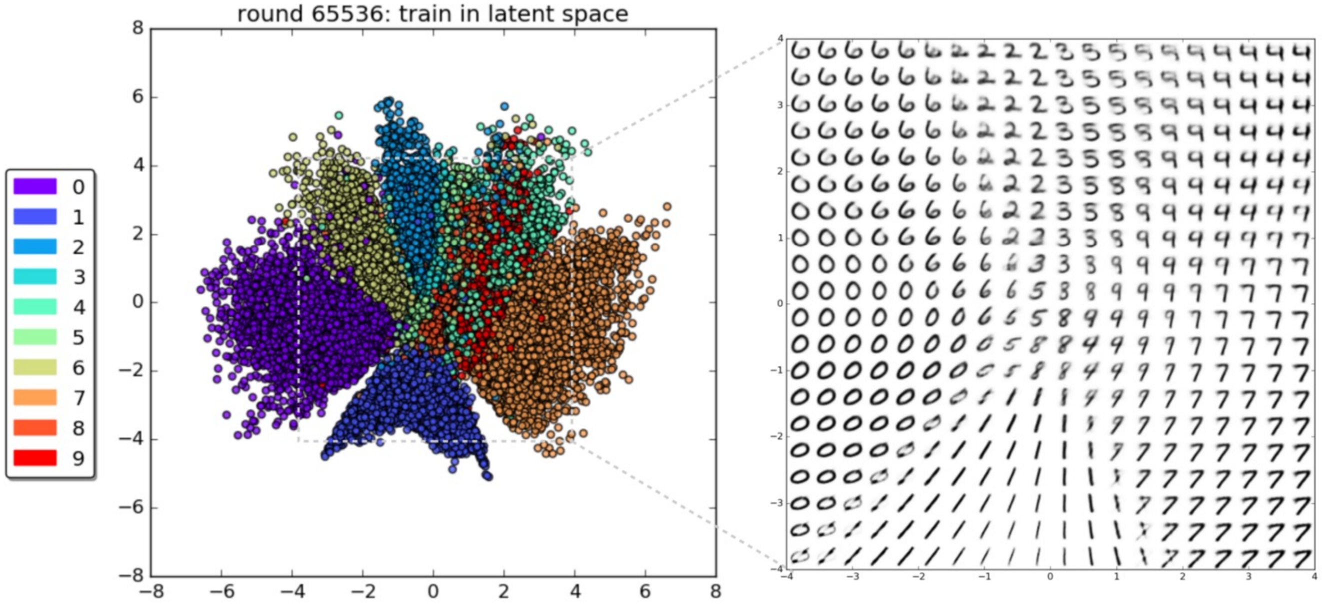
- 手写体的建模，注意每个数字之间的转换



VAE

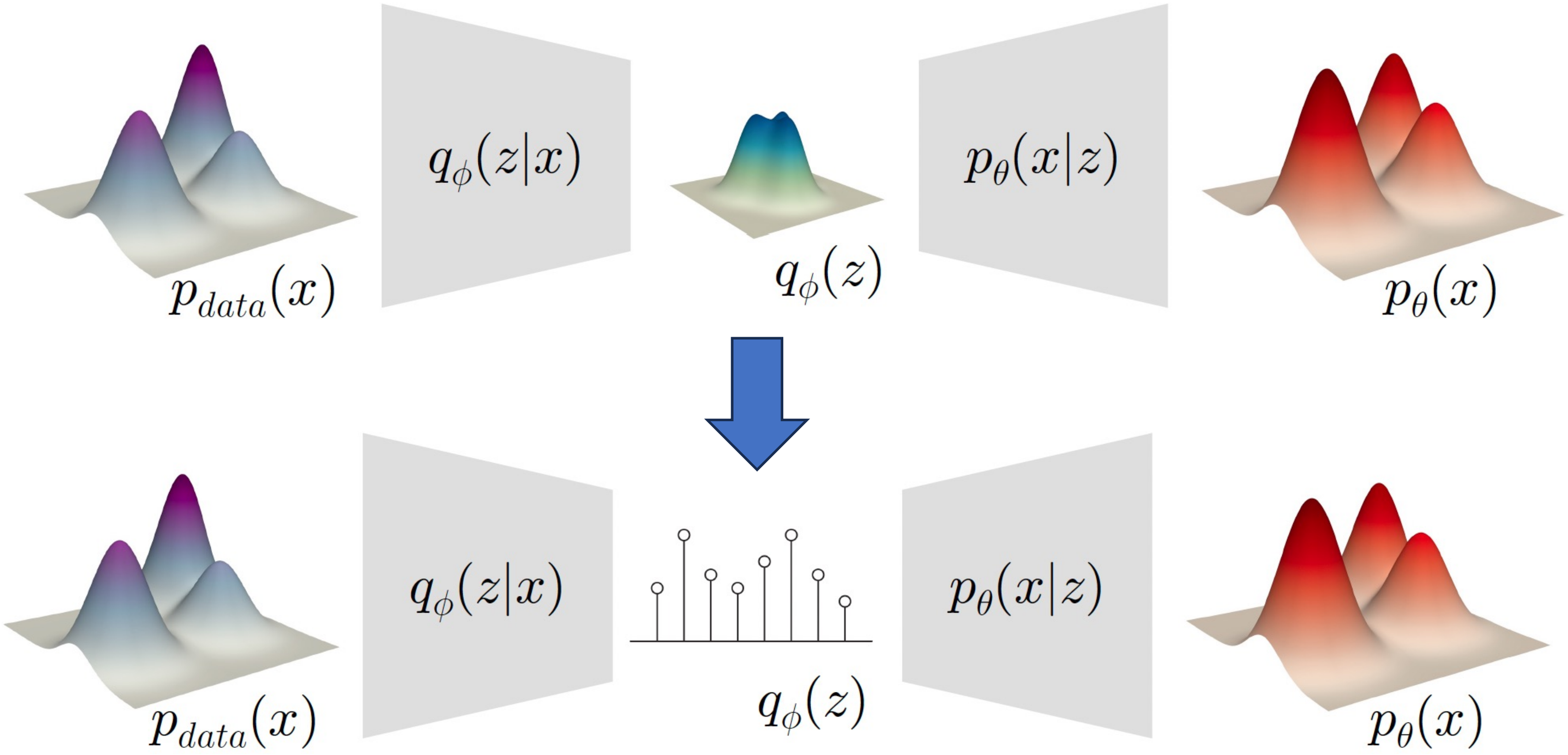
- 可视化VAE的训练过程（仅有2个潜在变量）





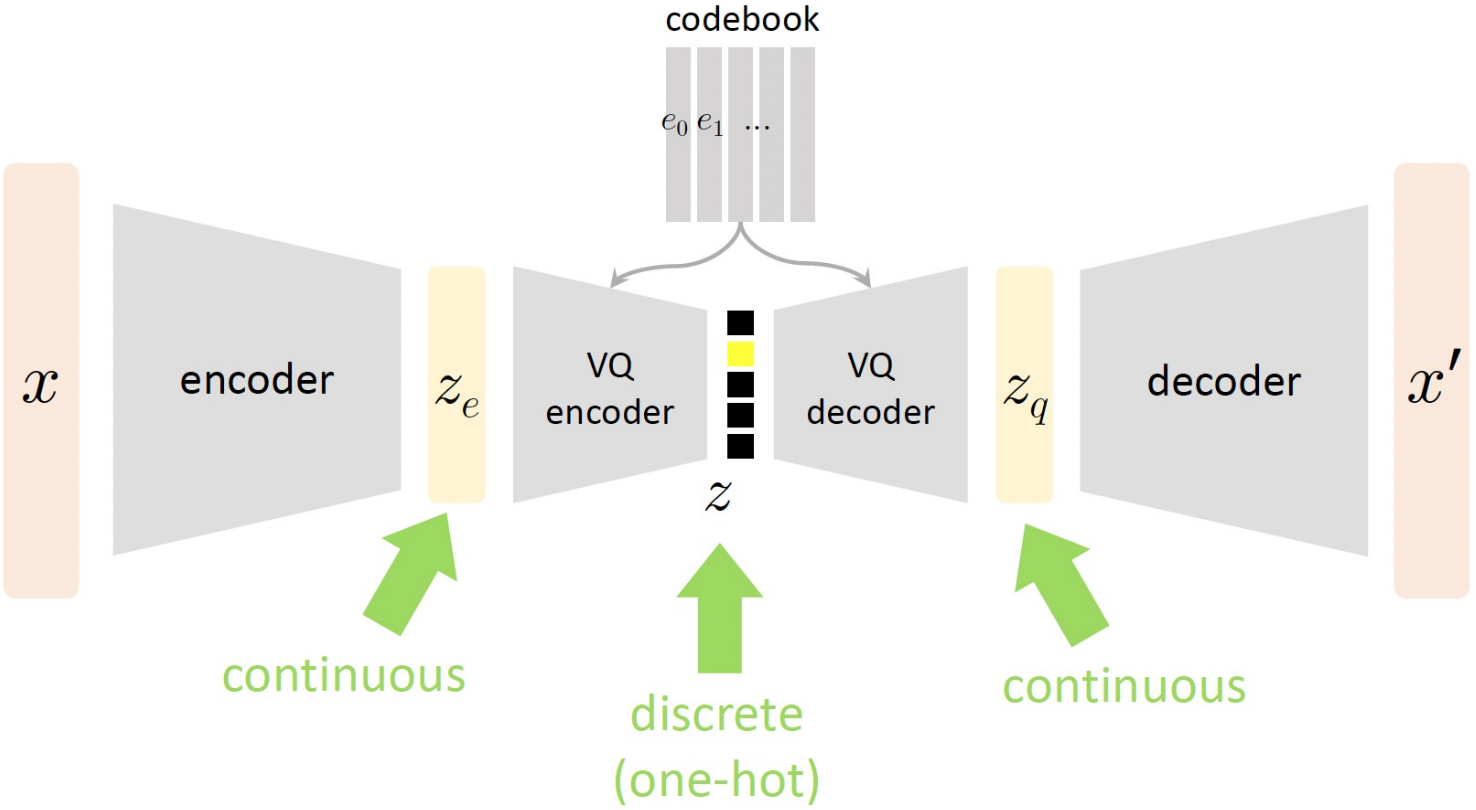
VAE

- 原始VAE的潜在变量是连续的
- 但有时候我们想要一些离散的潜在变量，有一些具体属性含义



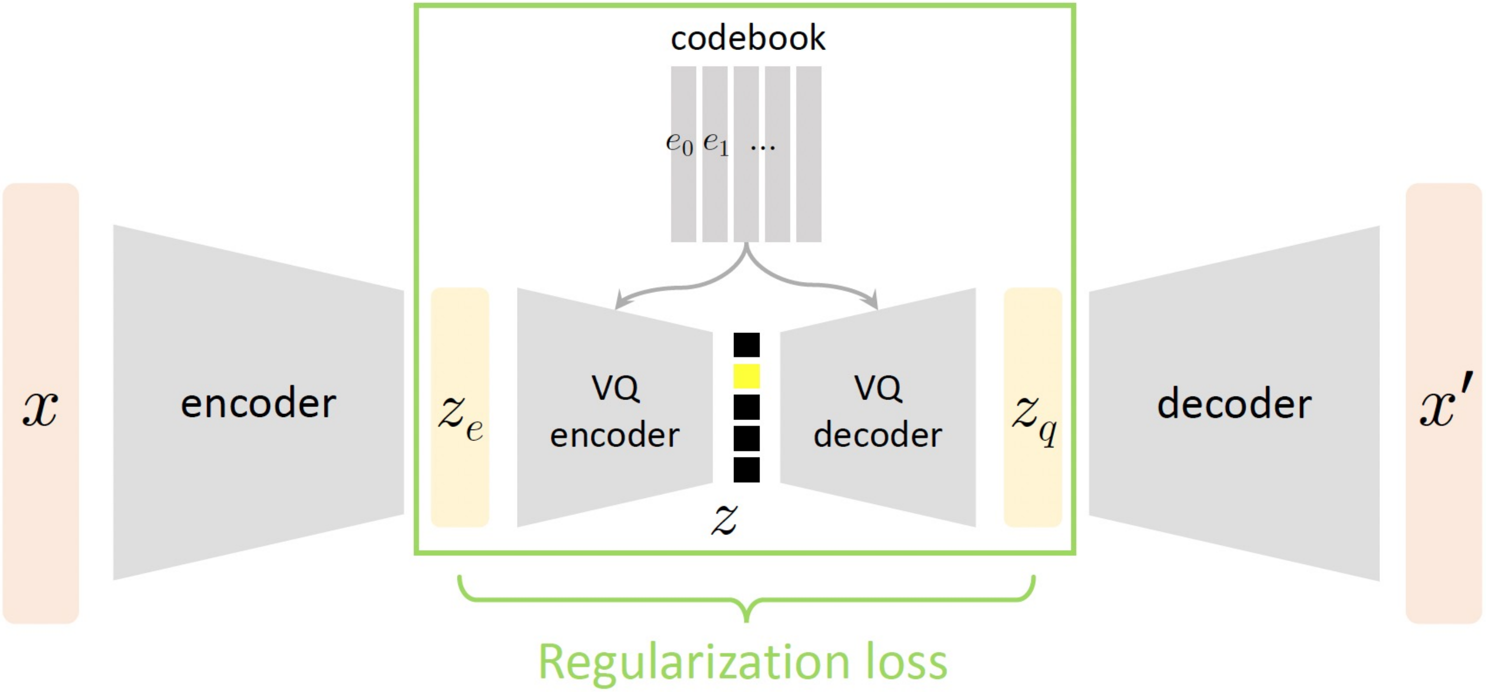
Vector Quantized VAE

- 引入一个codebook
- 利用E-M算法将潜在变量进行聚类，e.g. K-means



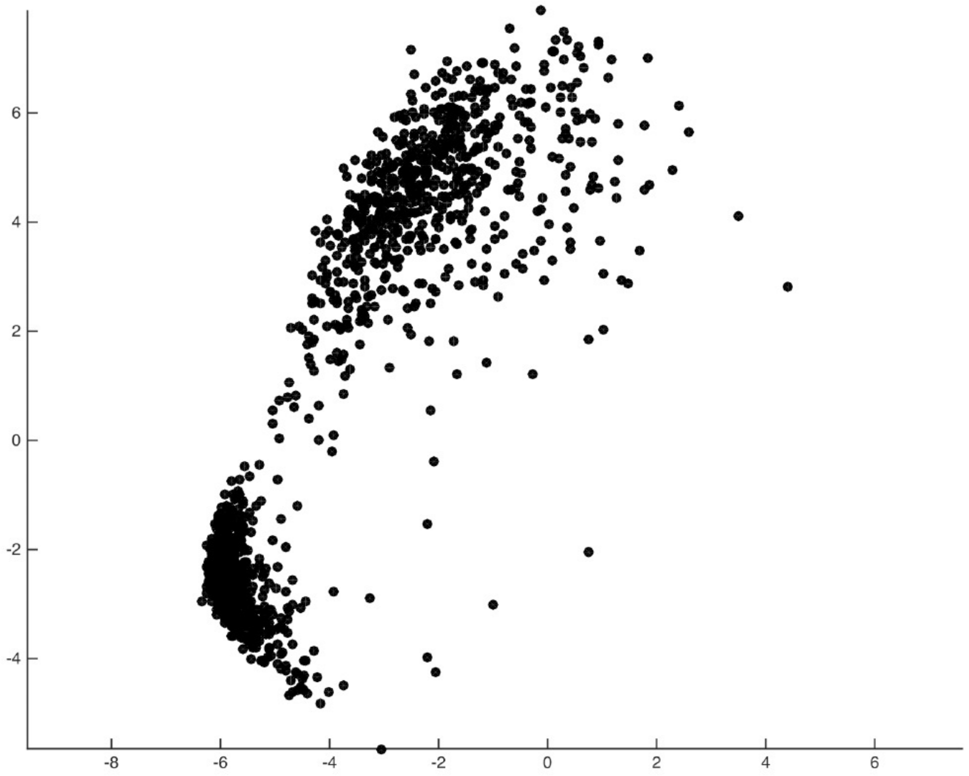
Vector Quantized VAE

- 正则化不需要显示的KL散度计算，聚类会让codebook更均匀



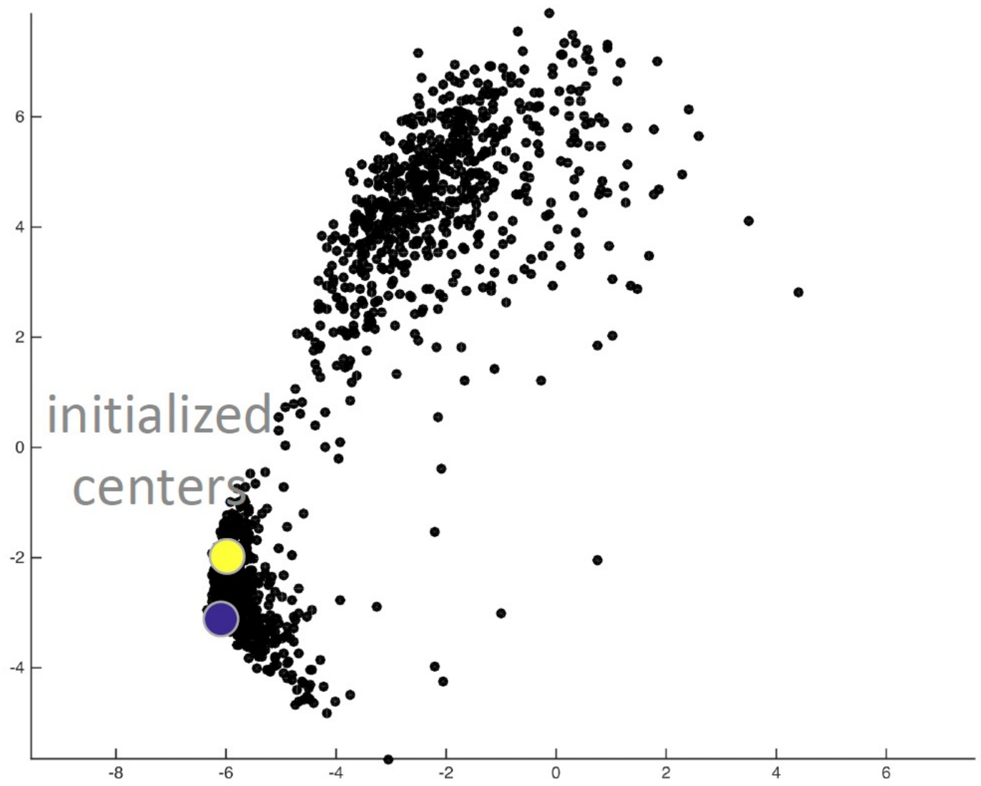
conceptually, this is the K-means reconstruction loss: $\|z_e - z_q\|^2$

Vector Quantized VAE



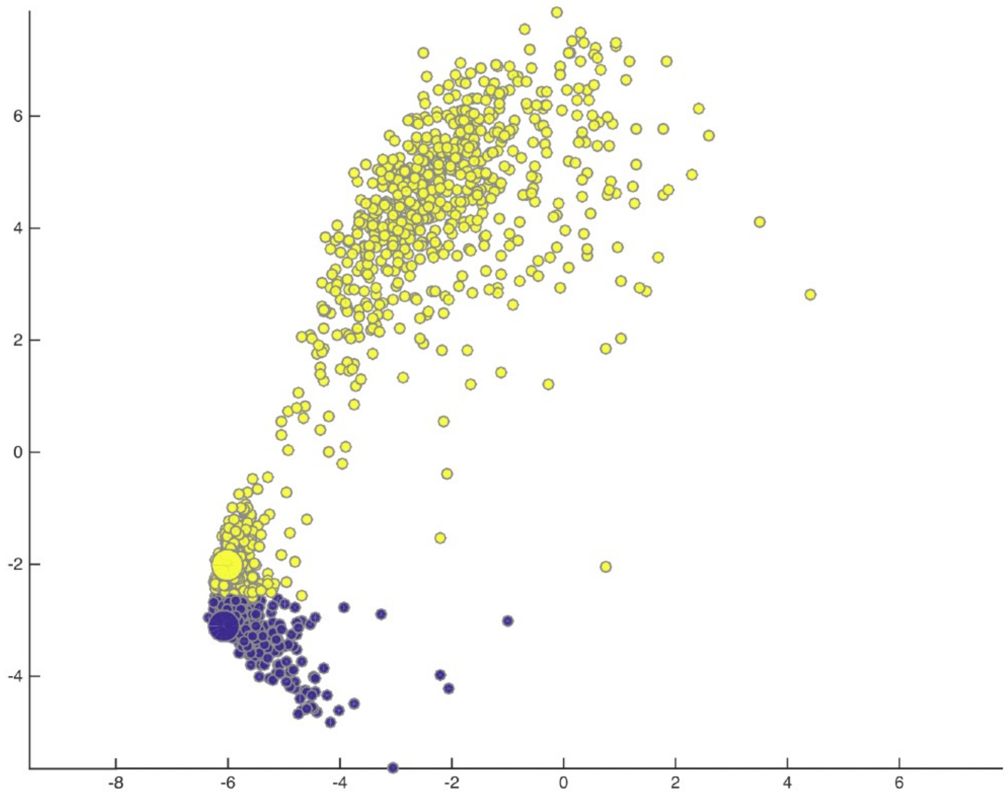
Vector Quantized VAE

- cluster centers: θ



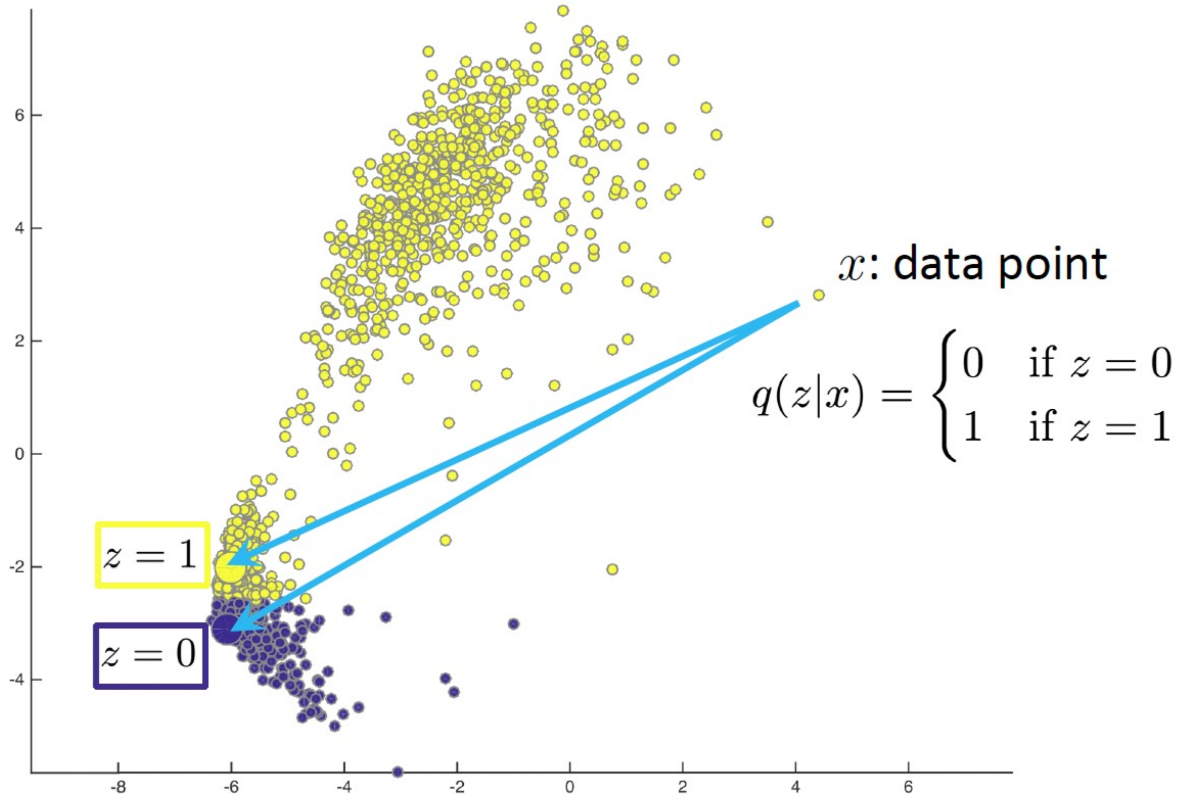
Vector Quantized VAE

- cluster centers: θ
- assignment:



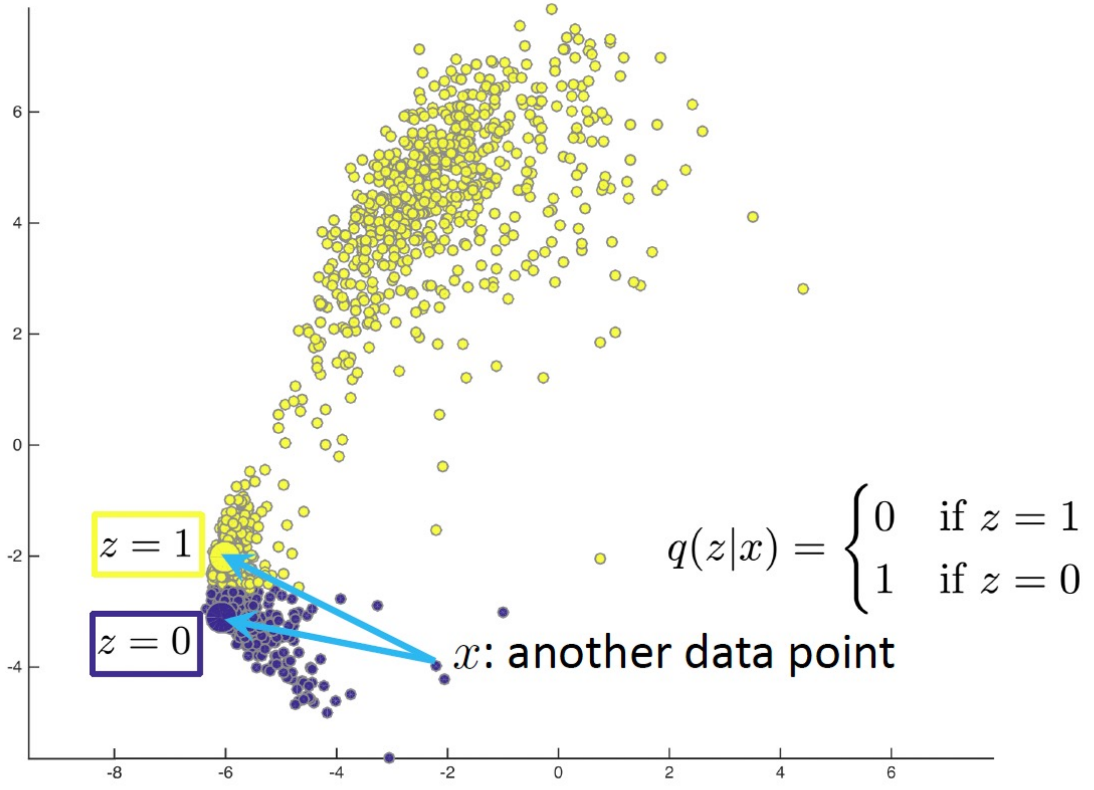
Vector Quantized VAE

- cluster centers: θ
- assignment: E-step



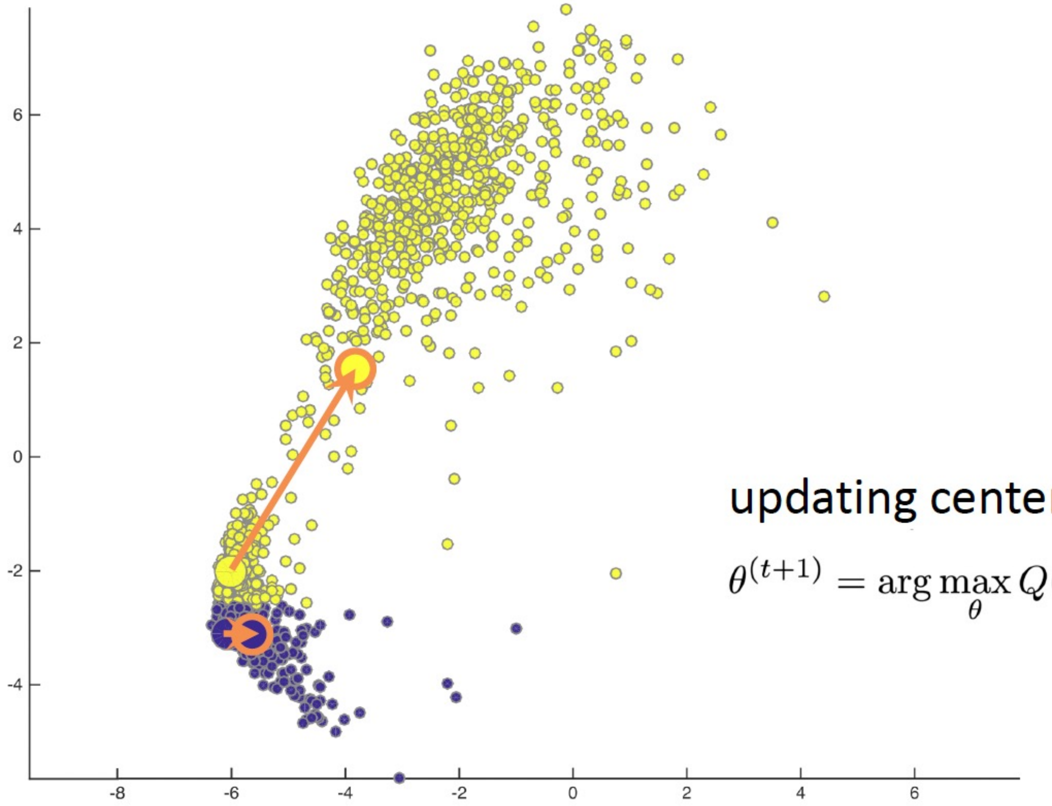
Vector Quantized VAE

- cluster centers: θ
- assignment: E-step



Vector Quantized VAE

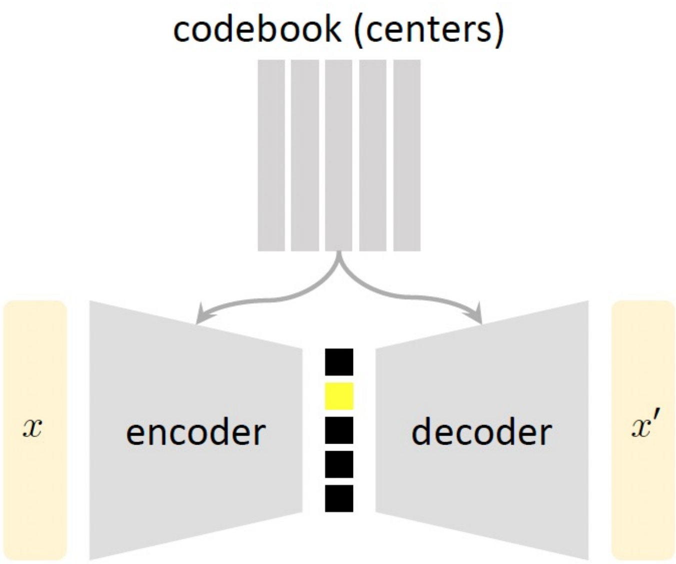
- cluster centers: θ
- assignment: E-step
- update: M-step



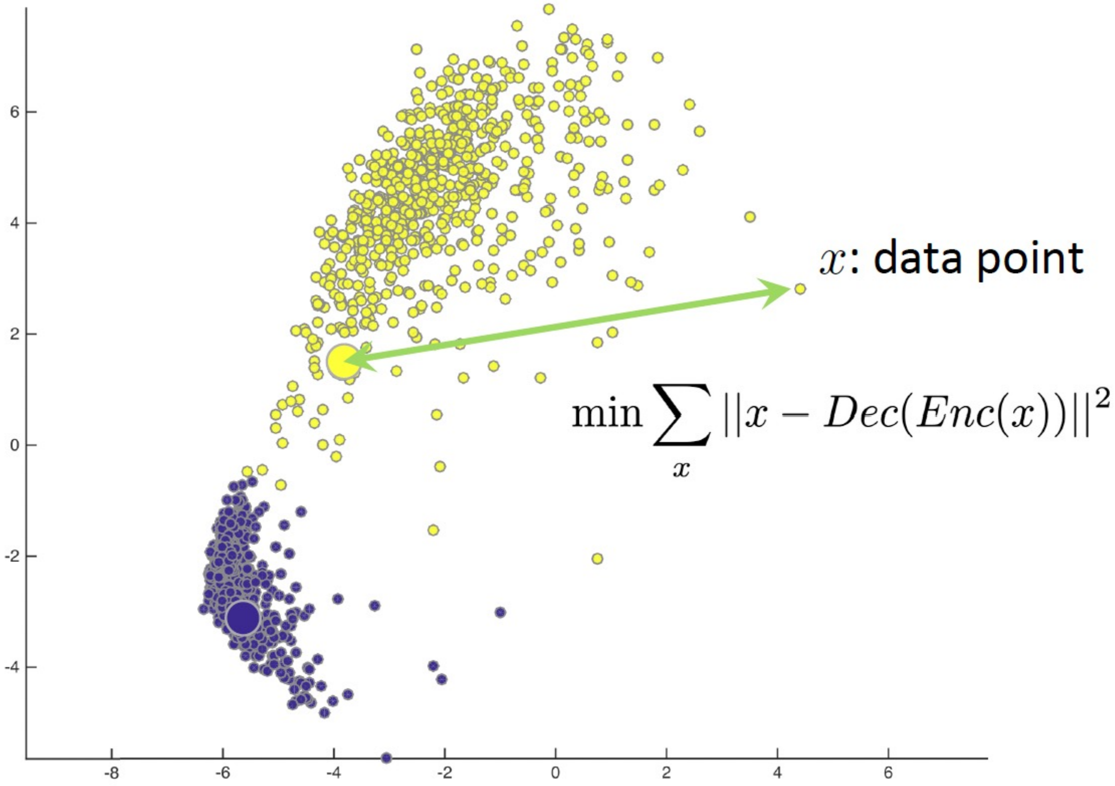
updating centers by:

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta|\theta^{(t)})$$

Vector Quantized VAE

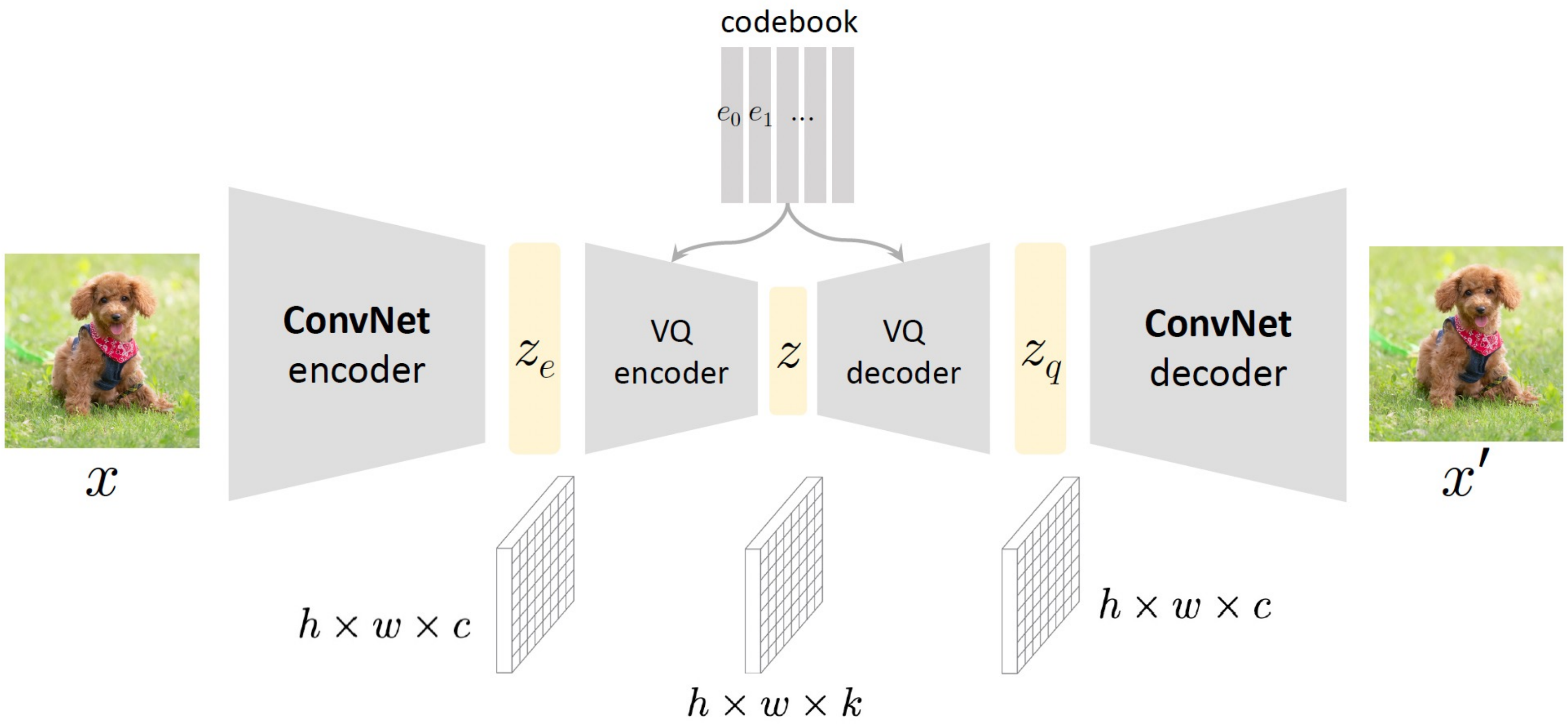


- encode: map x to one-hot
- decode: map one-hot to x'
- x' is a center



Vector Quantized VAE

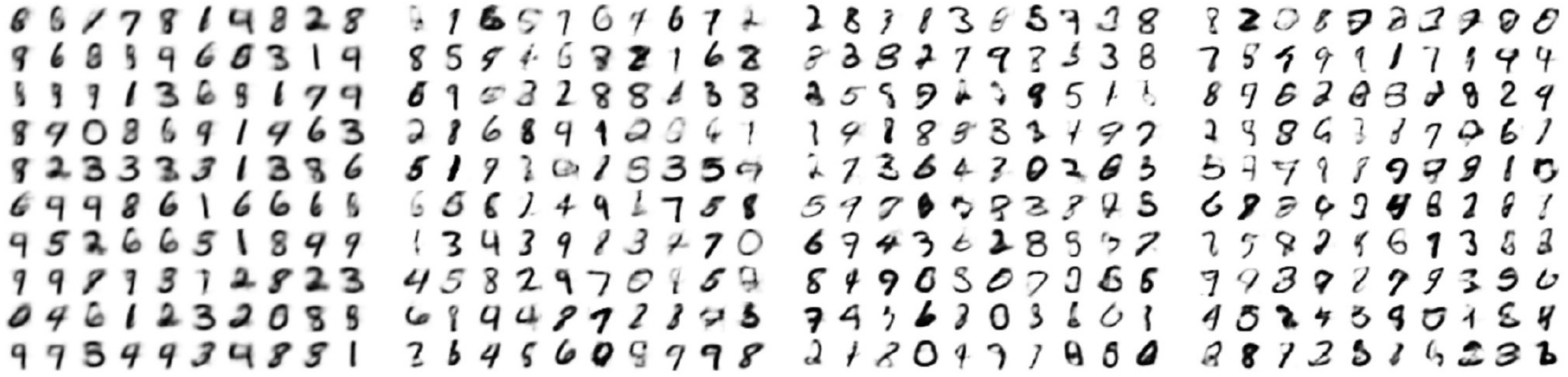
- VQ-VAE是一个很好的tokenizer（压缩为有含义的潜在变量）
- 输出的z为one-hot编码
- 更容易对先验建模，e.g.不需要具体概率，只需要softmax



VAE

- VAE每个潜在变量之间是独立的，但而真实数据都是紧密联系的
- 比如狗的颜色和品种，是强相关的
- 过度简化让高斯采样时生成效果不好，且维度越高越明显

VAE results on 784-d MNIST data



(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

Too strict to model the 784-d (28x28) joint distribution by independent distributions

小结

- VAE利用变分推断的角度构建ELBO
- 因为正则化约束，潜在空间是有含义、可解释的
- VAE和VQ-VAE有完整的编码-解码框架，天生是很好的tokenizer
- 属于独立潜在变量建模，过于建模潜在空间，生成效果不好

目录

1 生成模型基础

2 独立潜在变量建模

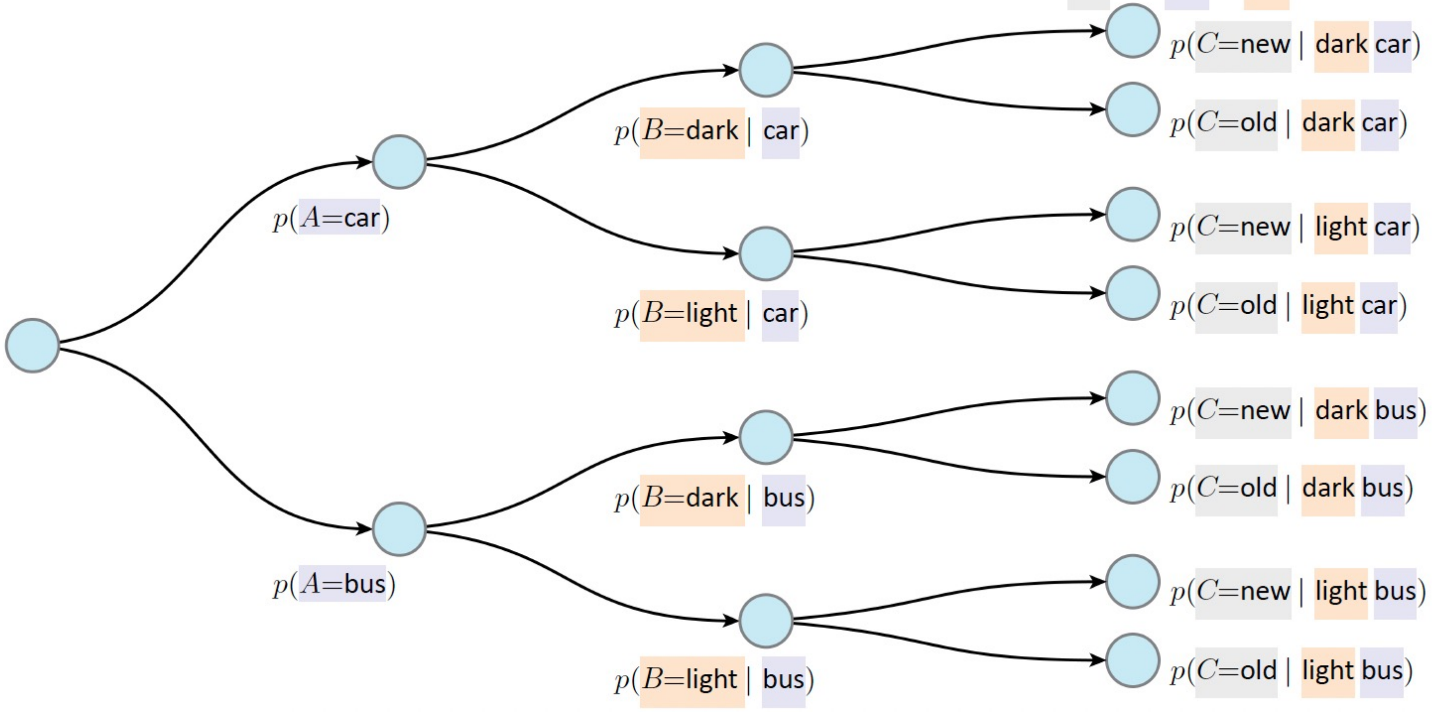
3 条件分布建模

4 现实问题建模

条件分布建模

- 我们希望生成的时候，潜在变量之间是紧密联系的
- 每个潜在变量都在链式法则的链条上

$$p(A,\, B,\, C) = p(A)p(B \mid A)p(C \mid A,\, B)$$



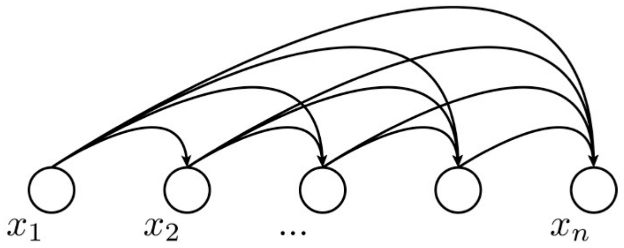
条件分布建模

- 自回归模型

Case 1: Partitioning the input representation space x

Example: Autoregressive Models on text tokens or pixels

$$p(x_1, x_2, ..., x_n) = p(x_1)p(x_2 \mid x_1)...p(x_n \mid x_1, x_2, ..., x_{n-1})$$



条件分布建模

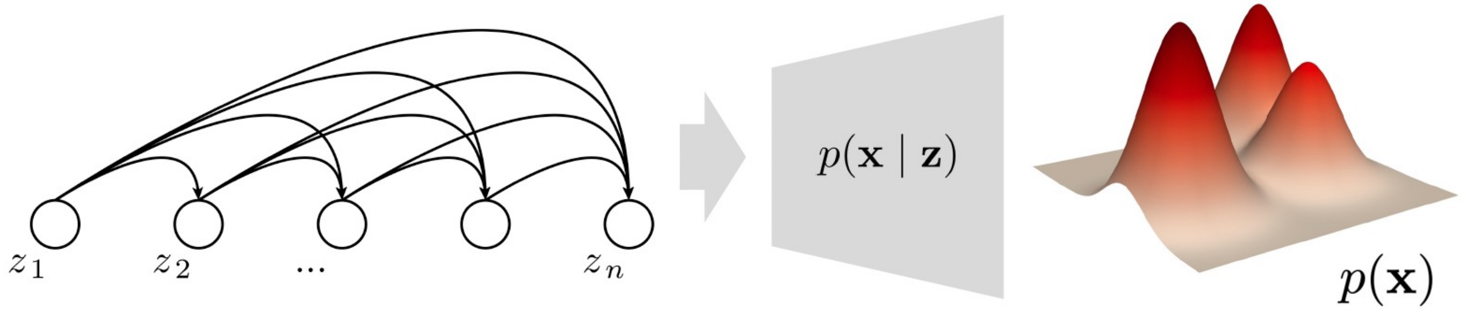
- 潜在空间的自回归模型

Case 2: Partitioning the latent representation space z

Example: Autoregressive Models on VQ-VAE tokens

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x} \mid \mathbf{z})$$

$$\text{with} \quad p(\mathbf{z}) = p(z_1)p(z_2 \mid z_1) \dots p(z_n \mid z_1, z_2, \dots, z_{n-1})$$



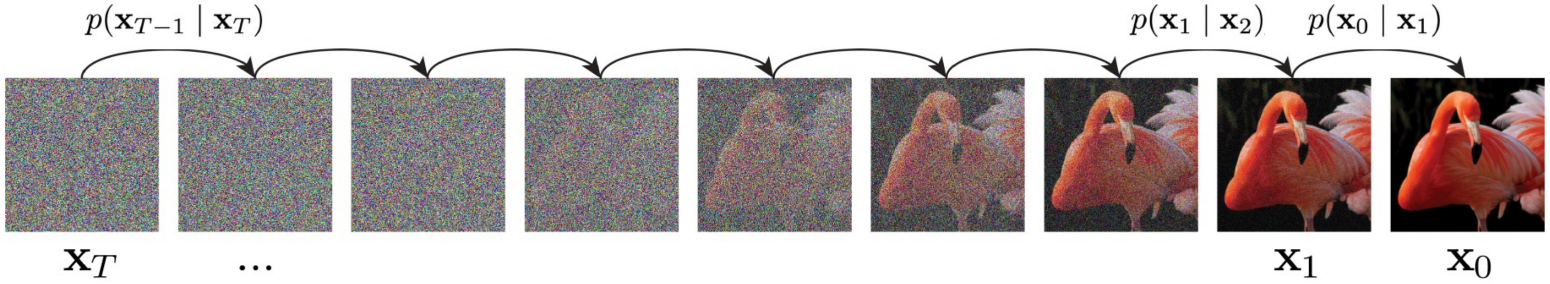
条件分布建模

- 马尔可夫链模型

Case 3: Progressively transforming data distributions

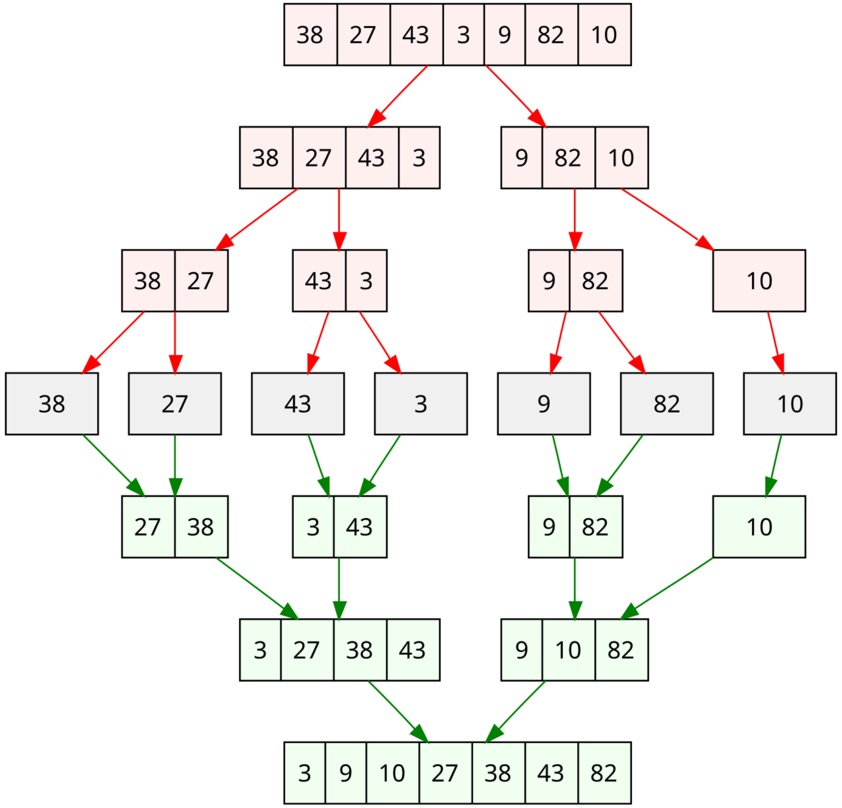
Example: Diffusion Models

$$p(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)p(\mathbf{x}_{T-1} \mid \mathbf{x}_T)...p(\mathbf{x}_1 \mid \mathbf{x}_2)p(\mathbf{x}_0 \mid \mathbf{x}_1)$$



条件分布建模

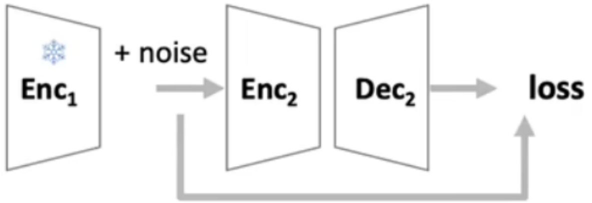
- 条件分布建模的思想与深度学习相同，都是Divide-and-Conquer
- 大本大源的解题方法：将复杂问题分层为简单问题，逐一解决



条件分布建模

- 条件分布建模的思想与深度学习相同，都是Divide-and-Conquer
- 将复杂问题分层为简单问题，逐层解决

逐层训练的深度网络

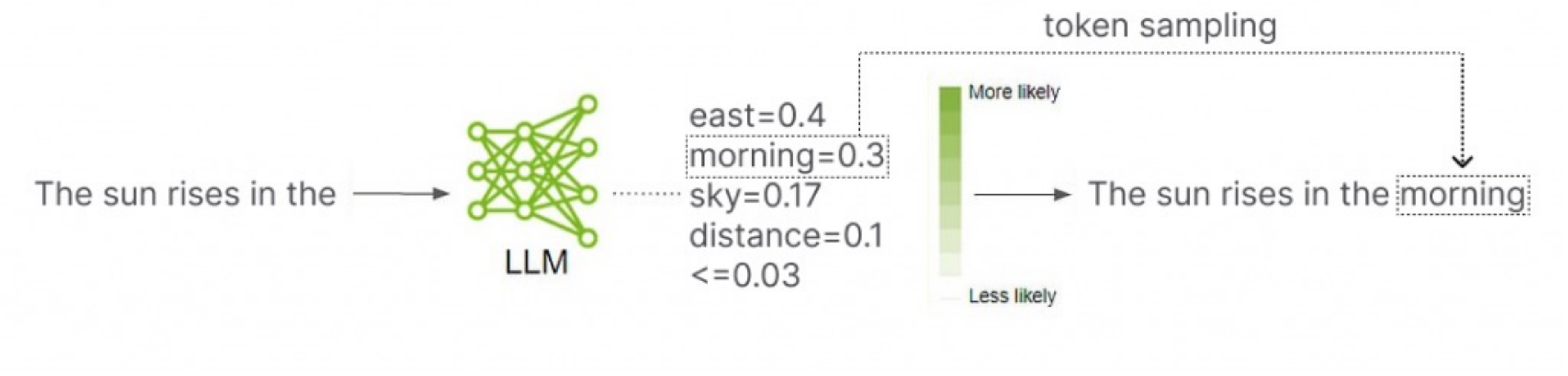


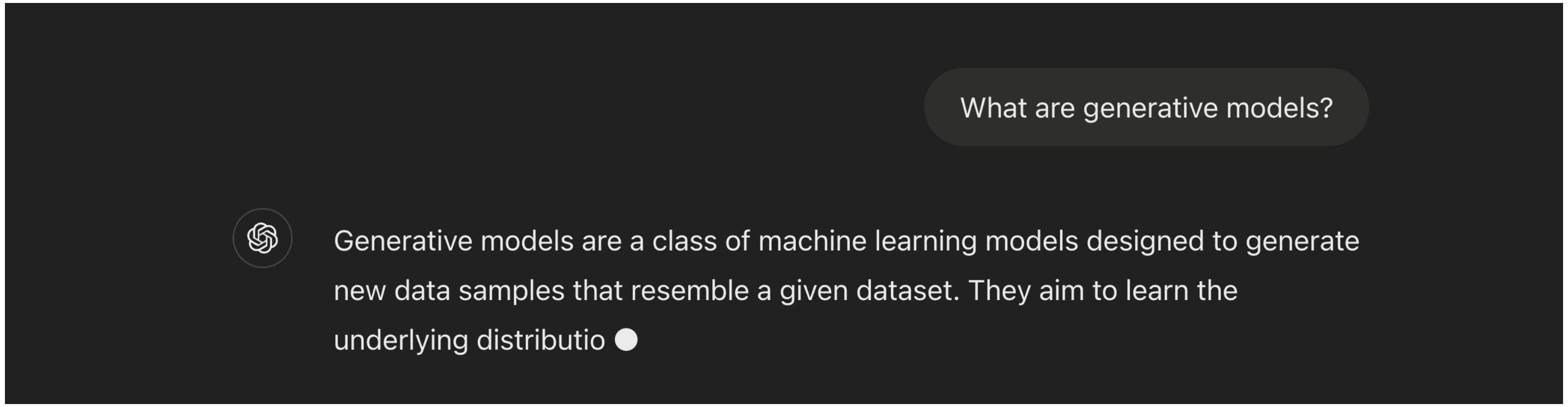
条件分布建模的自回归模型

Autoregressive

自回归模型

- 大语言模型的基础：Next Token Prediction

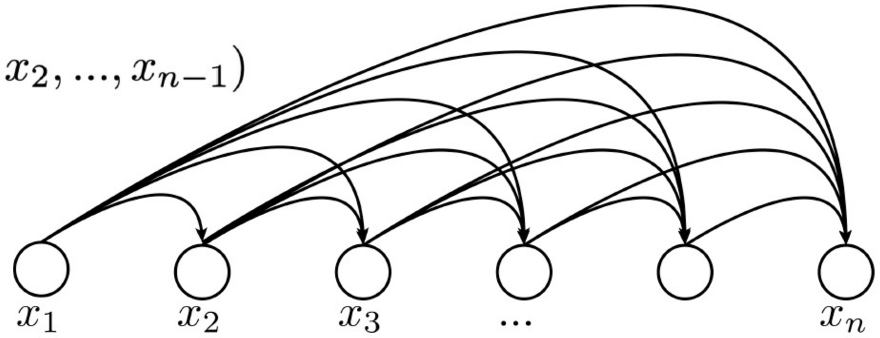




自回归模型

- 自回归模型（Auto Regression, AR）
- Auto：利用自己的输出决定之后的预测
- Rregression：估计不同变量之间的关系

$$
\begin{aligned}
p(x_1, x_2, ..., x_n) &= p(x_1)p(x_2 \mid x_1)...p(x_n \mid x_1, x_2, ..., x_{n-1}) \\
&= \prod_{i=1}^{n} p(x_i \mid x_1, x_2, ..., x_{i-1})
\end{aligned}
$$



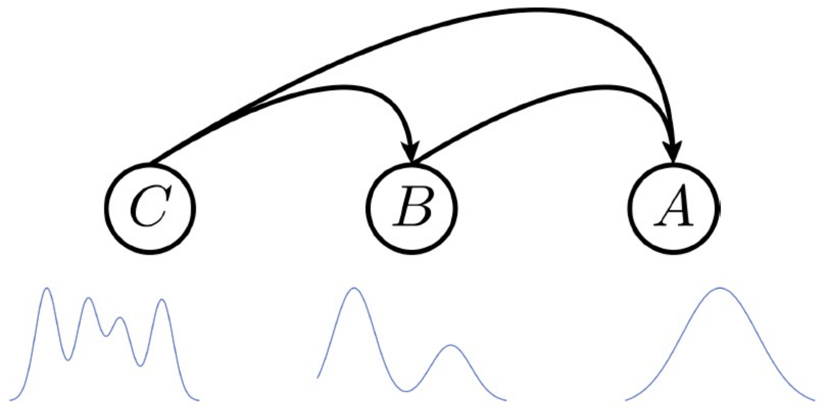
自回归模型

- 为了效率和泛化，每个分布分解之间的关系都用相似的神经网络
- 这样做的另一个好处是网络会带来简化分布的inductive bias

$$p(x_1, x_2, ..., x_n) \quad = \quad p_\theta(x_1)p_\theta(x_2 \mid x_1)...p_\theta(x_n \mid x_1, x_2, ..., x_{n-1})$$

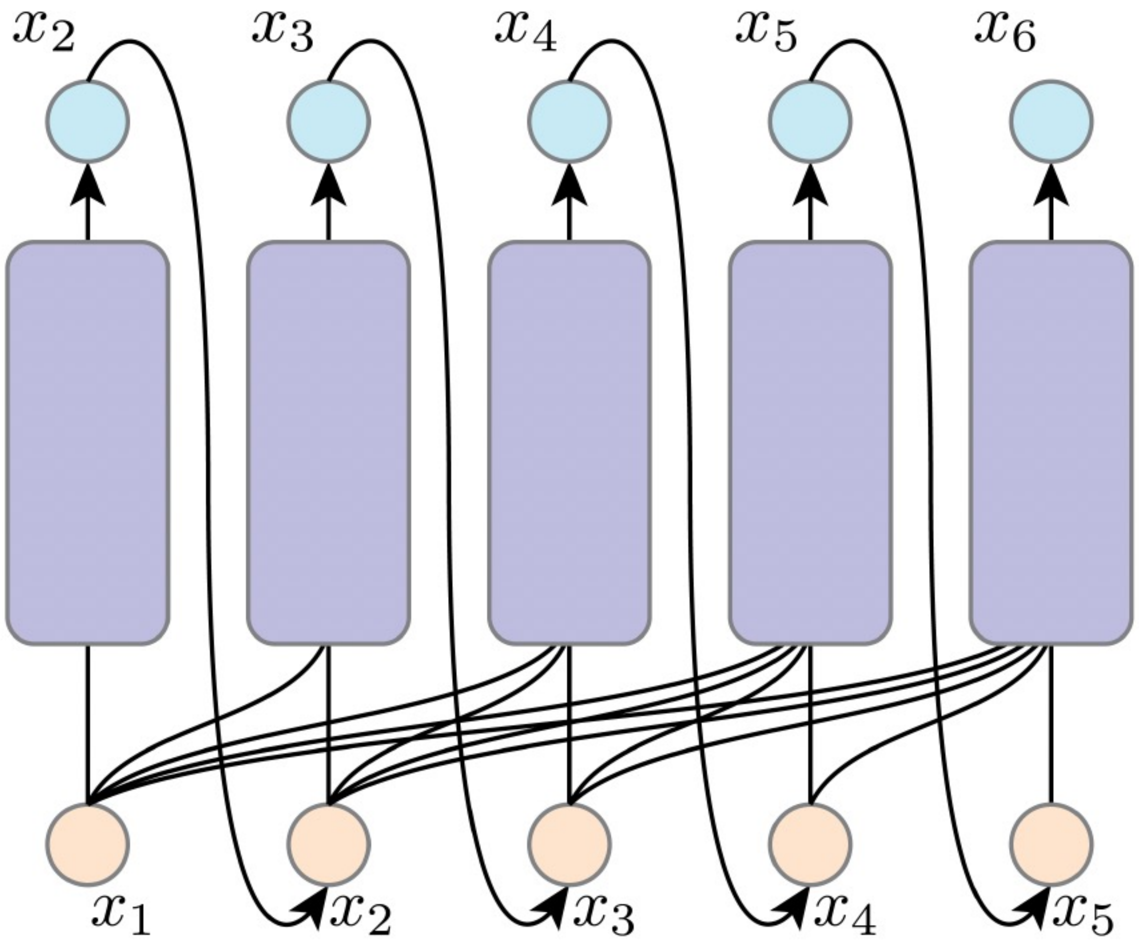
- Conceptually, these are different mappings
- But we model them by **shared architectures** (which can be RNN, CNN, Transformer, ...)
- and by **shared weights** θ

$$p(A,\, B,\, C) = p(C)p(B \mid C)p(A \mid B,\, C)$$



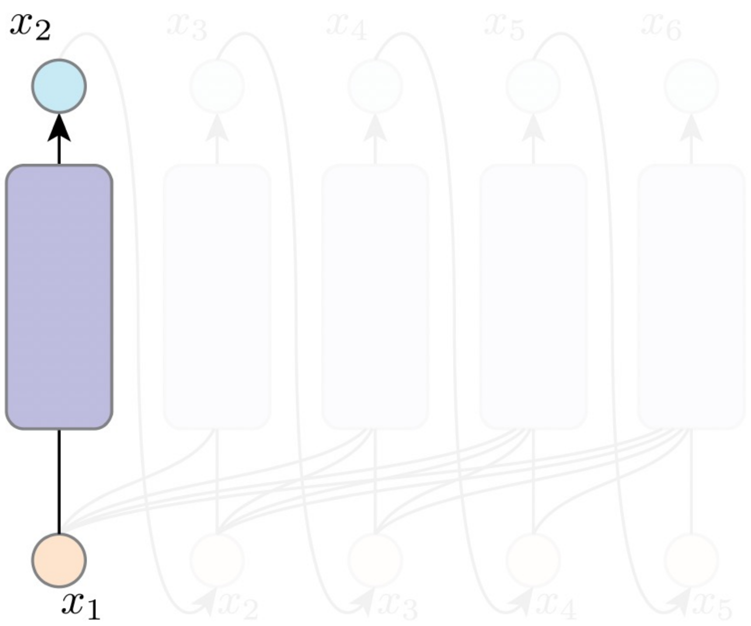
自回归模型

- AR是一个循环过程
- 理论上可以用任何模型



自回归模型

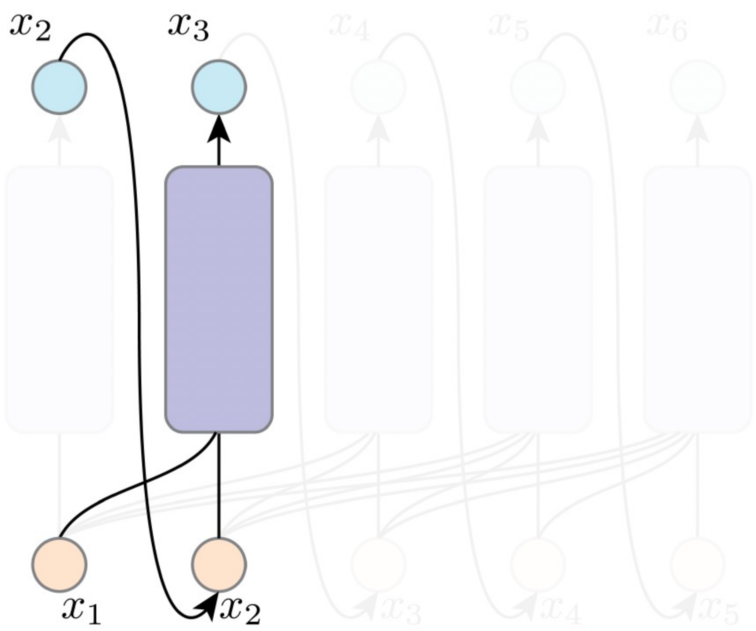
- This net models $p(x_2 \mid x_1)$
- 1 input
- 1 output



自回归模型

- This net models $p(x_3 \mid x_{1,2})$
- **2 inputs**
- 1 output

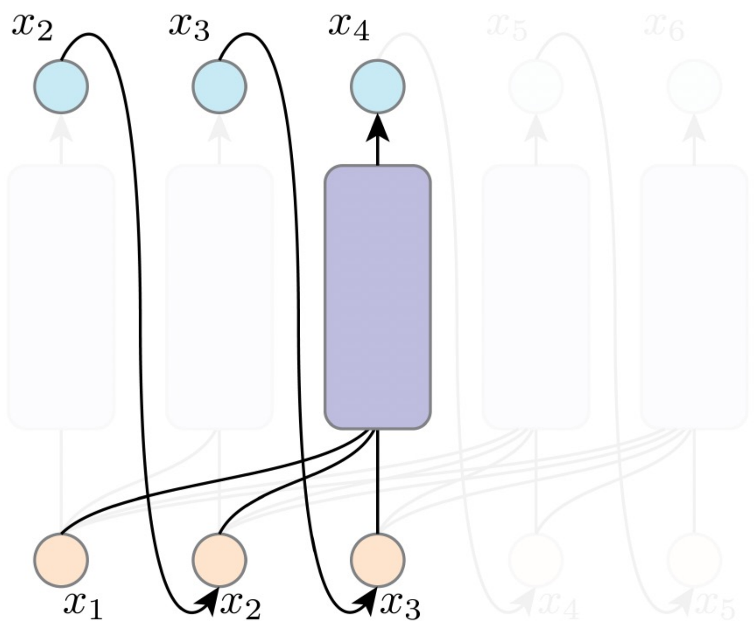
- inputs: outputs from previous steps



自回归模型

- This net models $p(x_4 \mid x_{1,2,3})$
- **3 inputs**
- 1 output

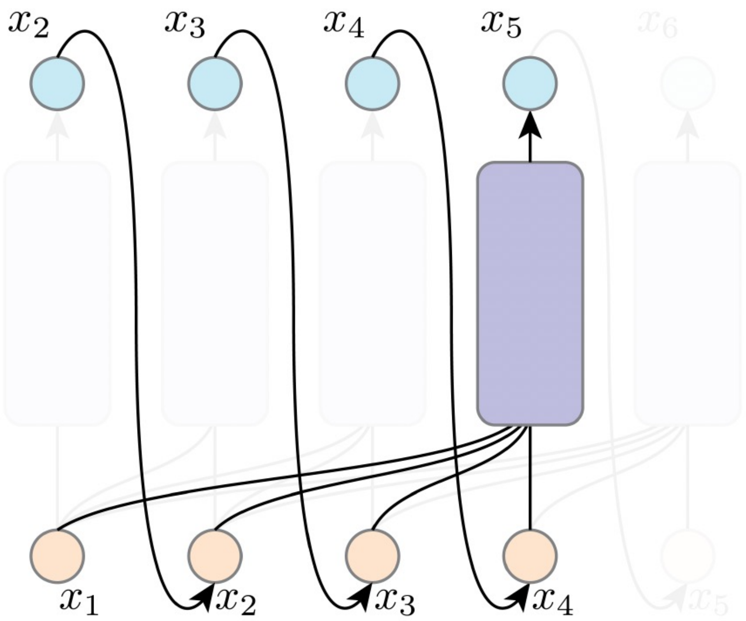
- inputs: outputs from previous steps



自回归模型

- This net models $p(x_5 \mid x_{1,2,3,4})$
- **4 inputs**
- 1 output

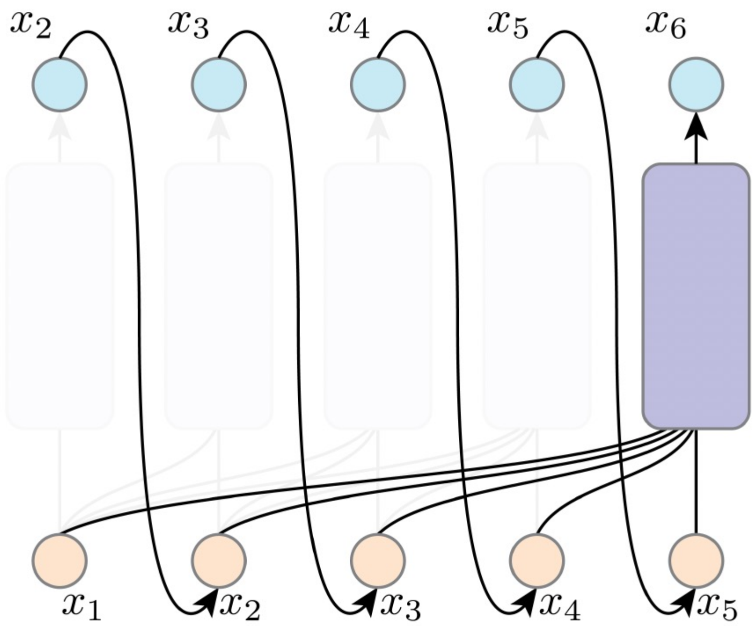
- inputs: outputs from previous steps



自回归模型

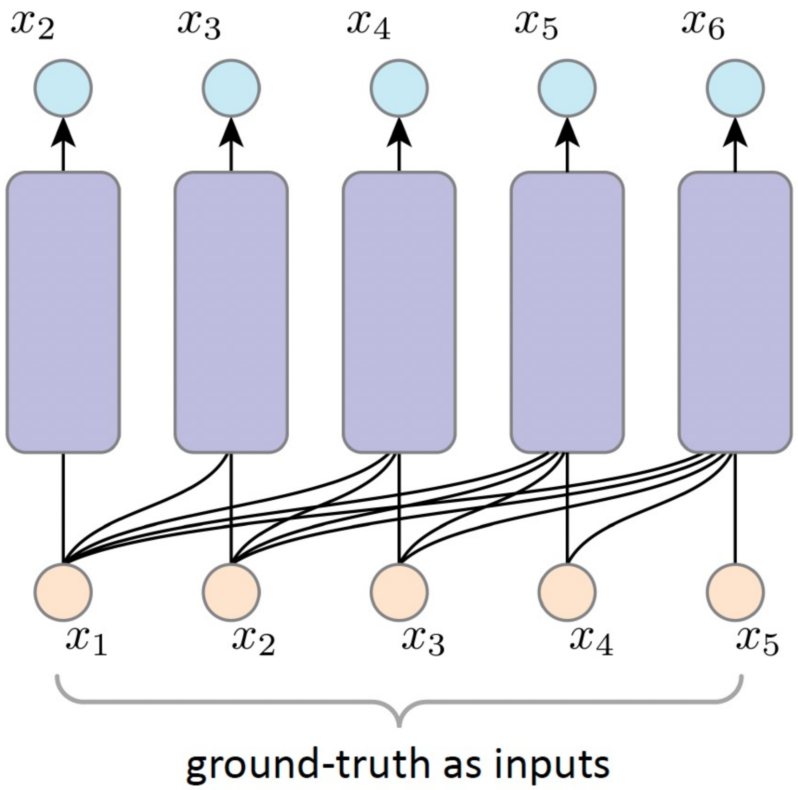
- This net models $p(x_6 \mid x_{1,2,3,4,5})$
- **5 inputs**
- 1 output

- inputs: outputs from previous steps



自回归模型

- 在采样的时候一般基于每个时刻的真实样本
- 训练时的反向传播更加简单



自回归模型

- 图像中的AR：PixelCNN

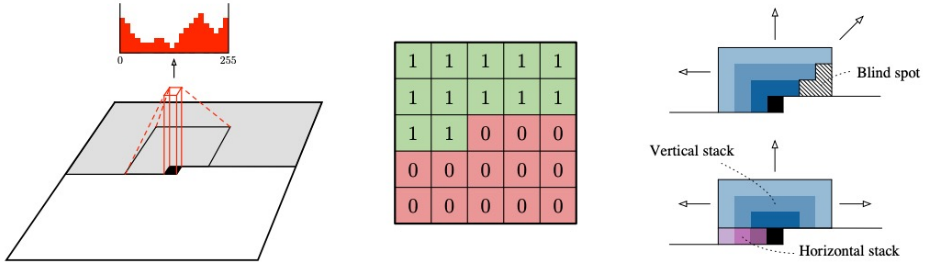
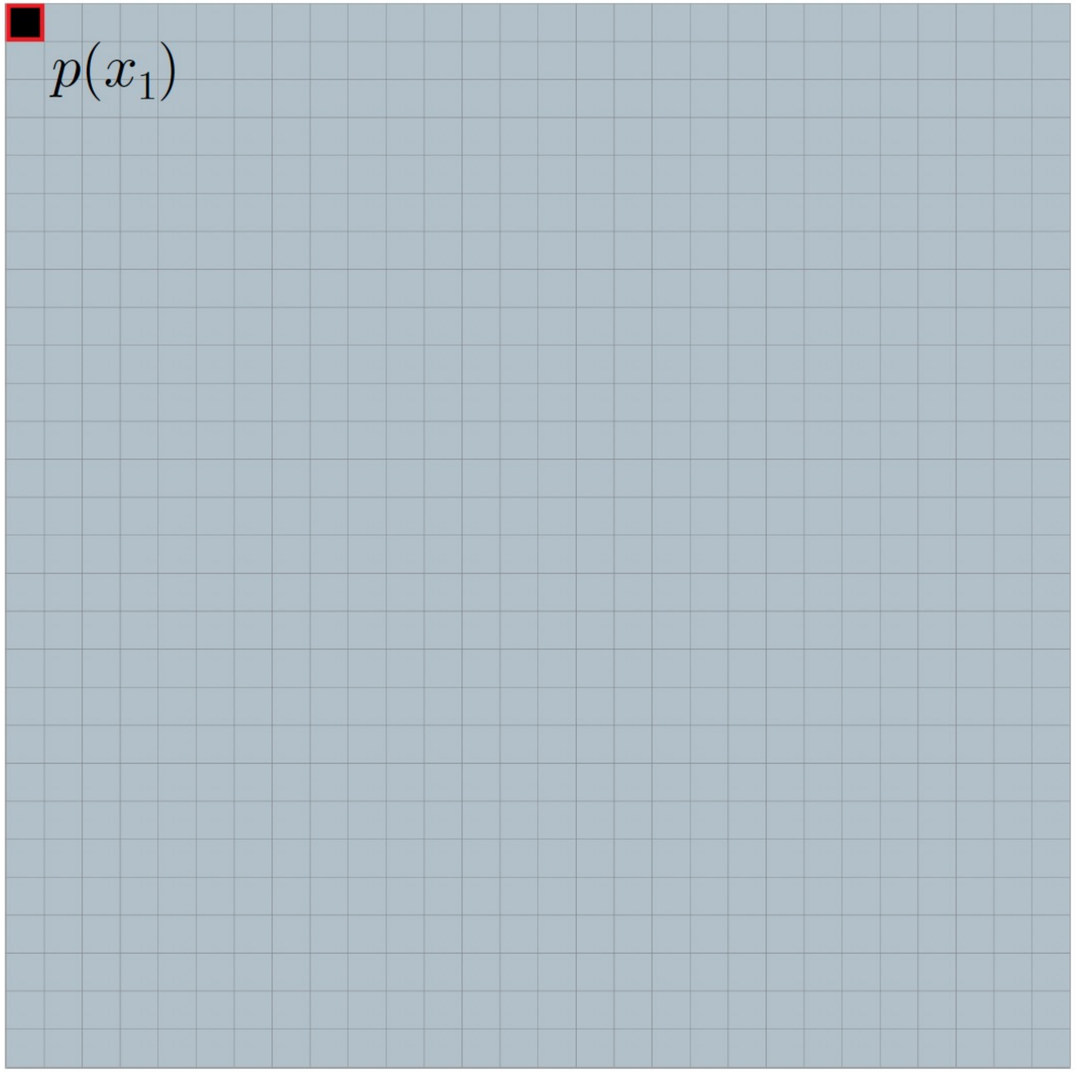


Figure 1: **Left**: A visualization of the PixelCNN that maps a neighborhood of pixels to prediction for the next pixel. To generate pixel x_i the model can only condition on the previously generated pixels $x_1, \ldots x_{i-1}$. **Middle**: an example matrix that is used to mask the 5x5 filters to make sure the model cannot read pixels below (or strictly to the right) of the current pixel to make its predictions. **Right**: Top: PixelCNNs have a *blind spot* in the receptive field that can not be used to make predictions. Bottom: Two convolutional stacks (blue and purple) allow to capture the whole receptive field.



Conditional Image Generation with PixelCNN Decoders

自回归模型

- 图像中的AR：PixelCNN

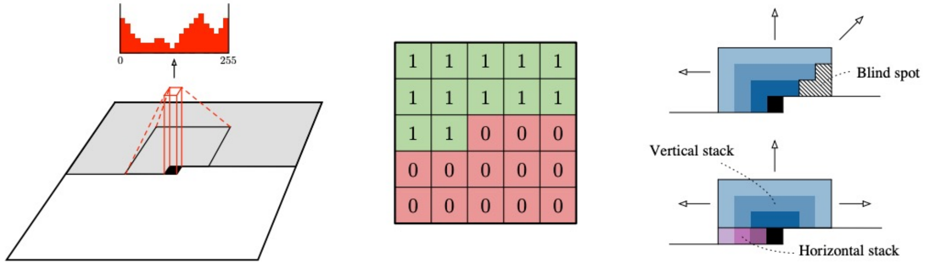
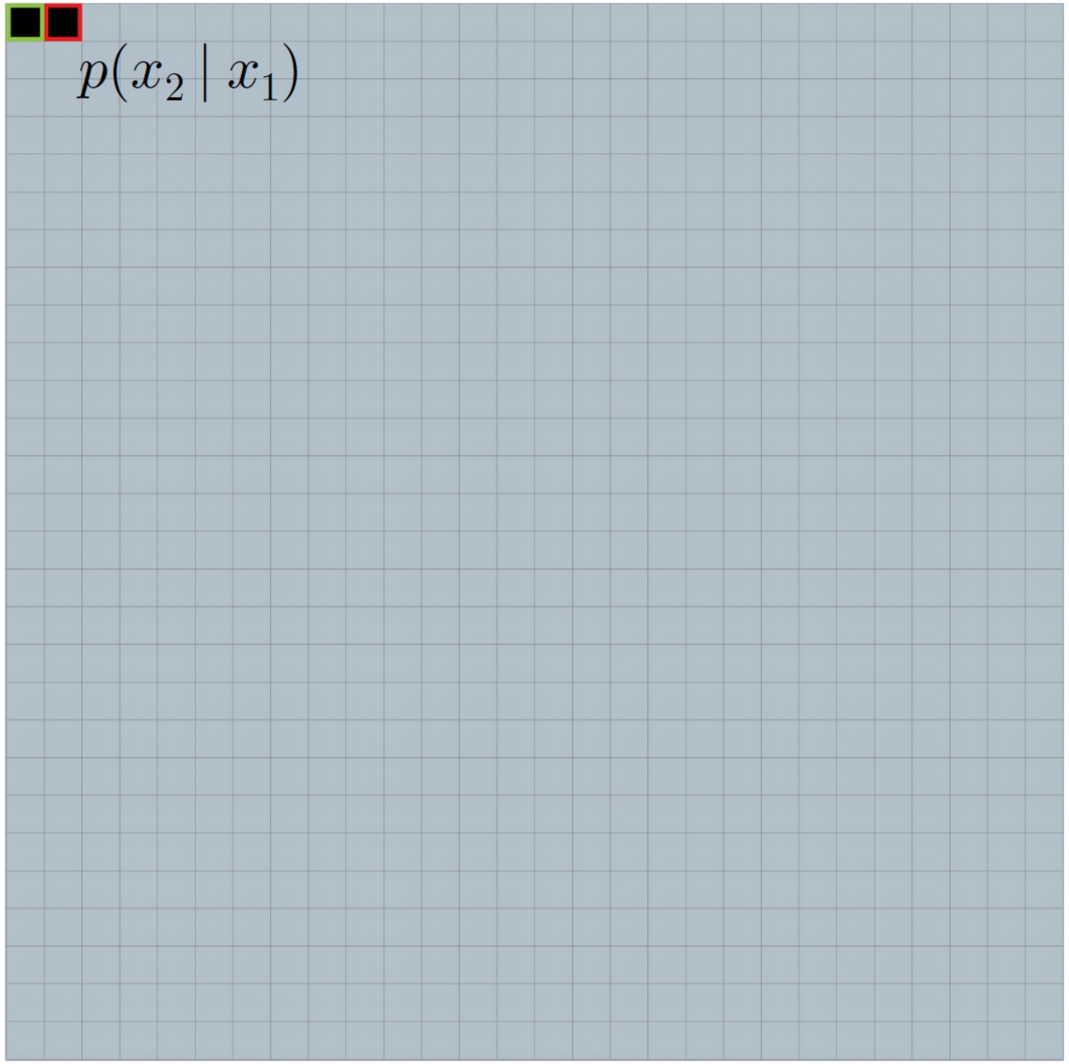


Figure 1: **Left**: A visualization of the PixelCNN that maps a neighborhood of pixels to prediction for the next pixel. To generate pixel x_i the model can only condition on the previously generated pixels $x_1, \ldots x_{i-1}$. **Middle**: an example matrix that is used to mask the 5x5 filters to make sure the model cannot read pixels below (or strictly to the right) of the current pixel to make its predictions. **Right**: Top: PixelCNNs have a *blind spot* in the receptive field that can not be used to make predictions. Bottom: Two convolutional stacks (blue and purple) allow to capture the whole receptive field.



Conditional Image Generation with PixelCNN Decoders

自回归模型

- 图像中的AR：PixelCNN

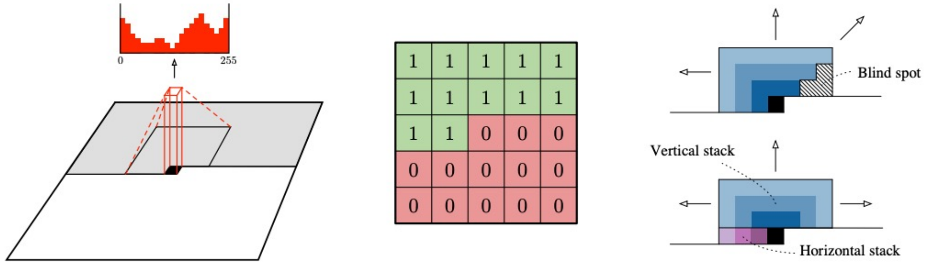
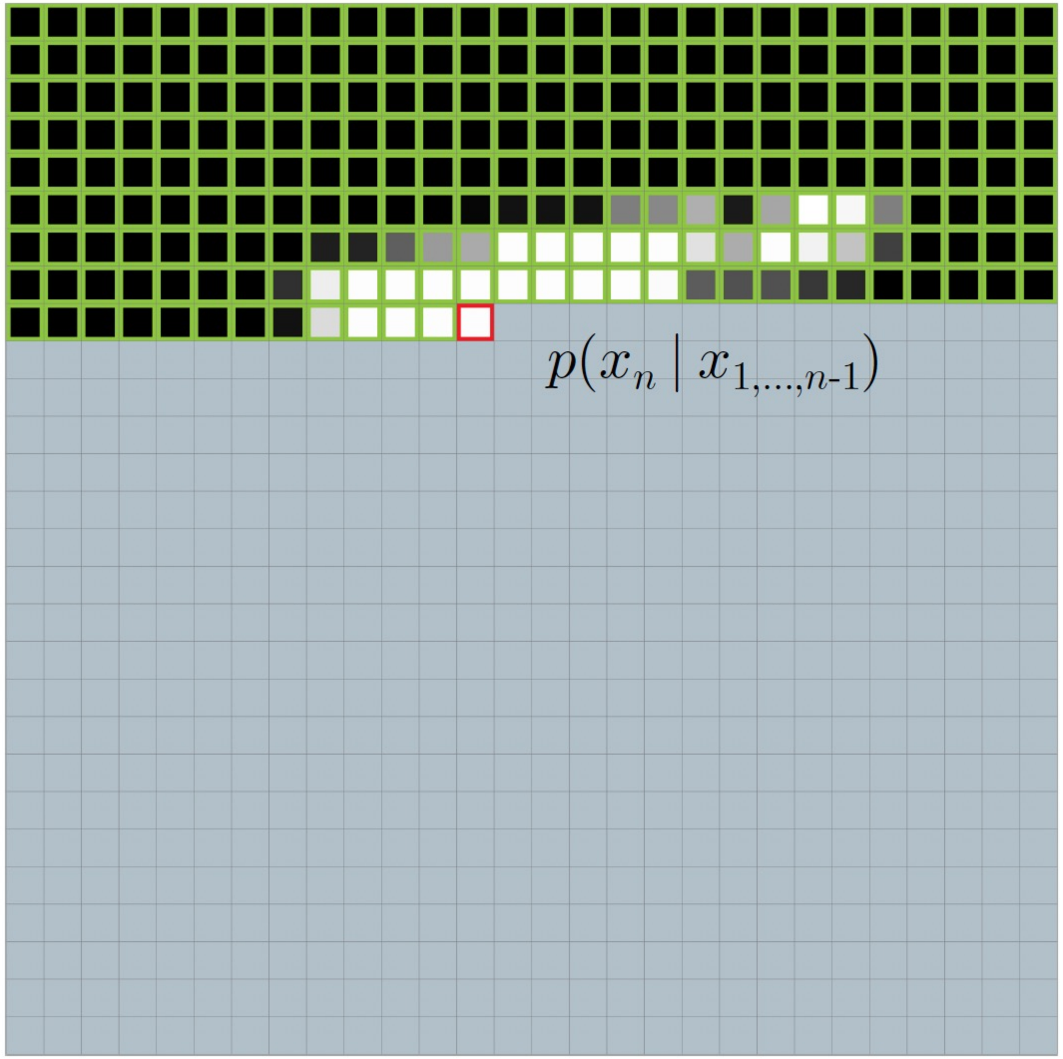


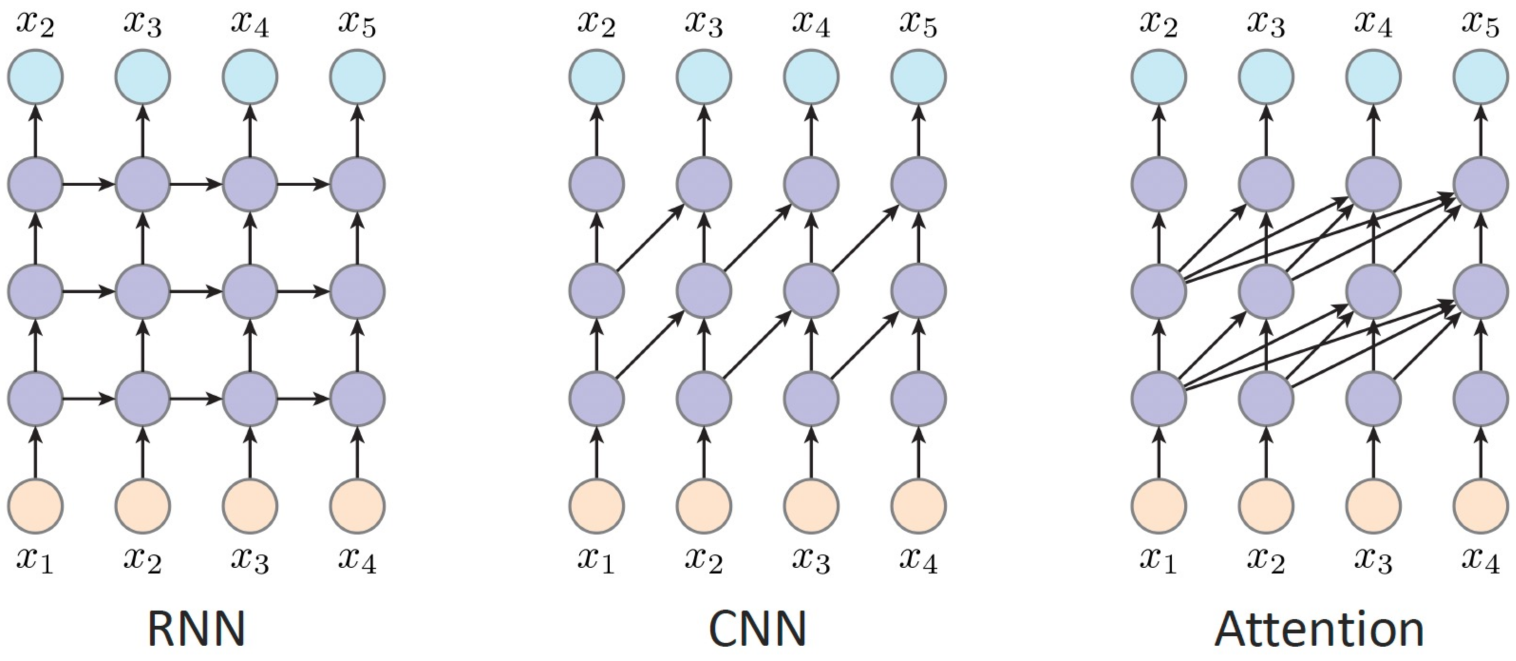
Figure 1: **Left**: A visualization of the PixelCNN that maps a neighborhood of pixels to prediction for the next pixel. To generate pixel x_i the model can only condition on the previously generated pixels $x_1, \dots x_{i-1}$. **Middle**: an example matrix that is used to mask the 5x5 filters to make sure the model cannot read pixels below (or strictly to the right) of the current pixel to make its predictions. **Right**: Top: PixelCNNs have a *blind spot* in the receptive field that can not be used to make predictions. Bottom: Two convolutional stacks (blue and purple) allow to capture the whole receptive field.



Conditional Image Generation with PixelCNN Decoders

自回归模型

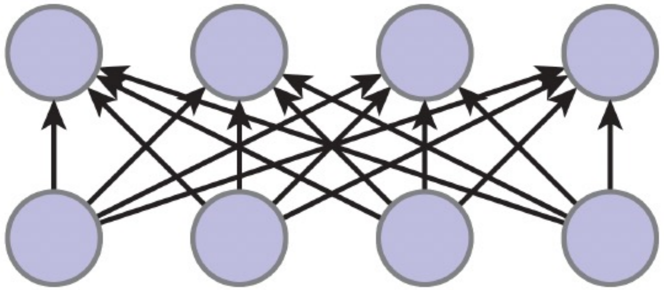
- 自回归网络可以用很多种网络结构实现
- 注意，网络连接都是causal的，即是只有从左到右的变化



自回归模型

full attention

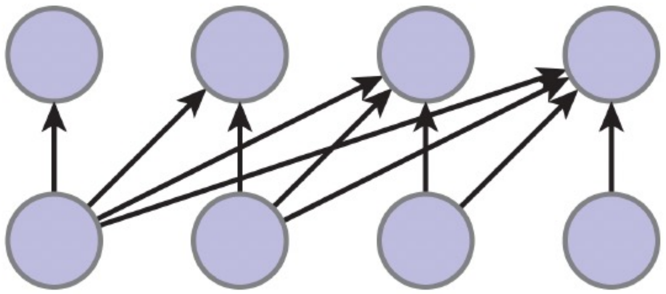
(every step sees all steps)



自回归模型

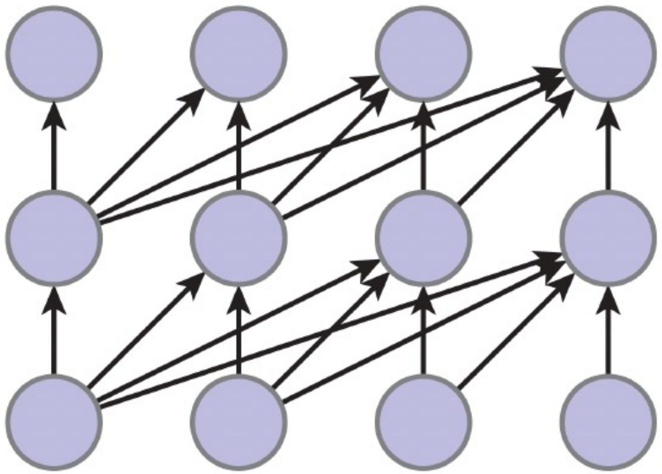
causal attention

(not depend on “future”)



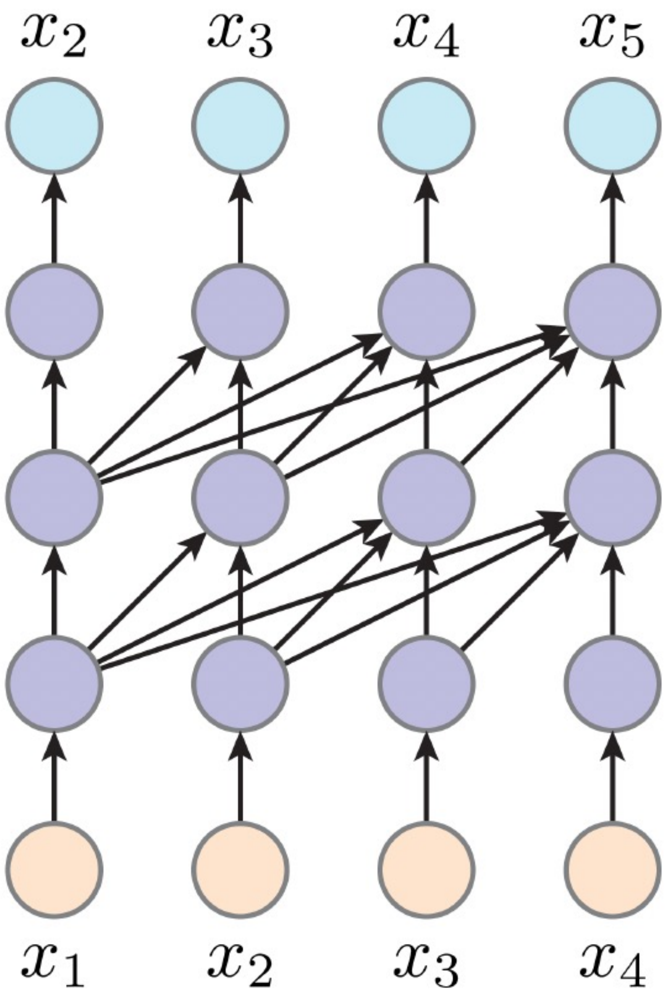
自回归模型

go deep

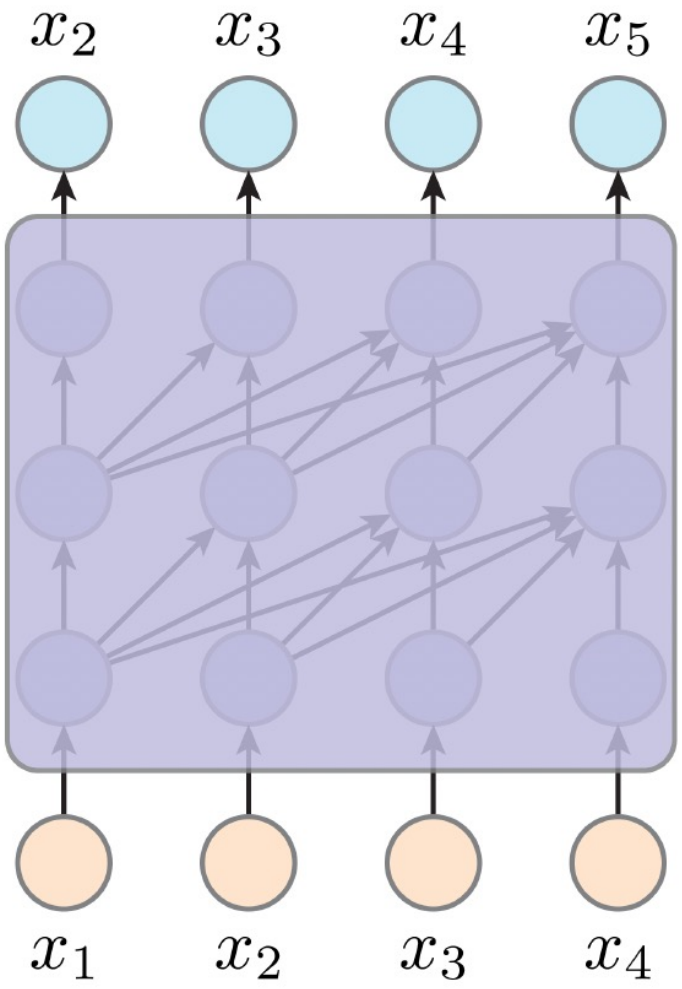


自回归模型

shift target by one step

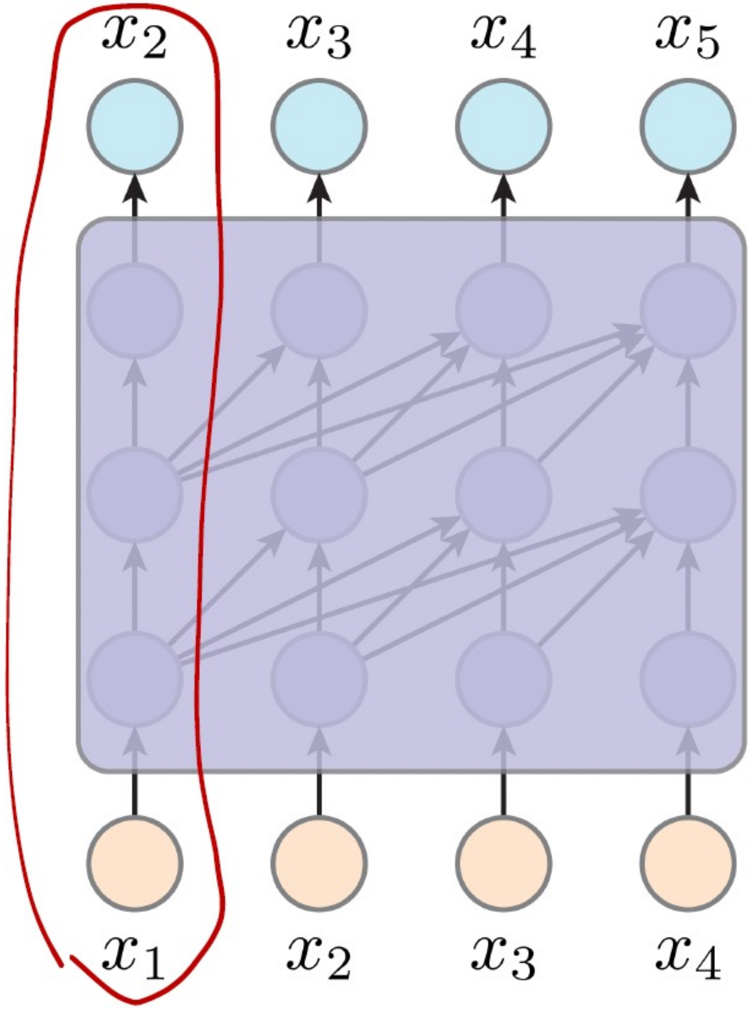


自回归模型



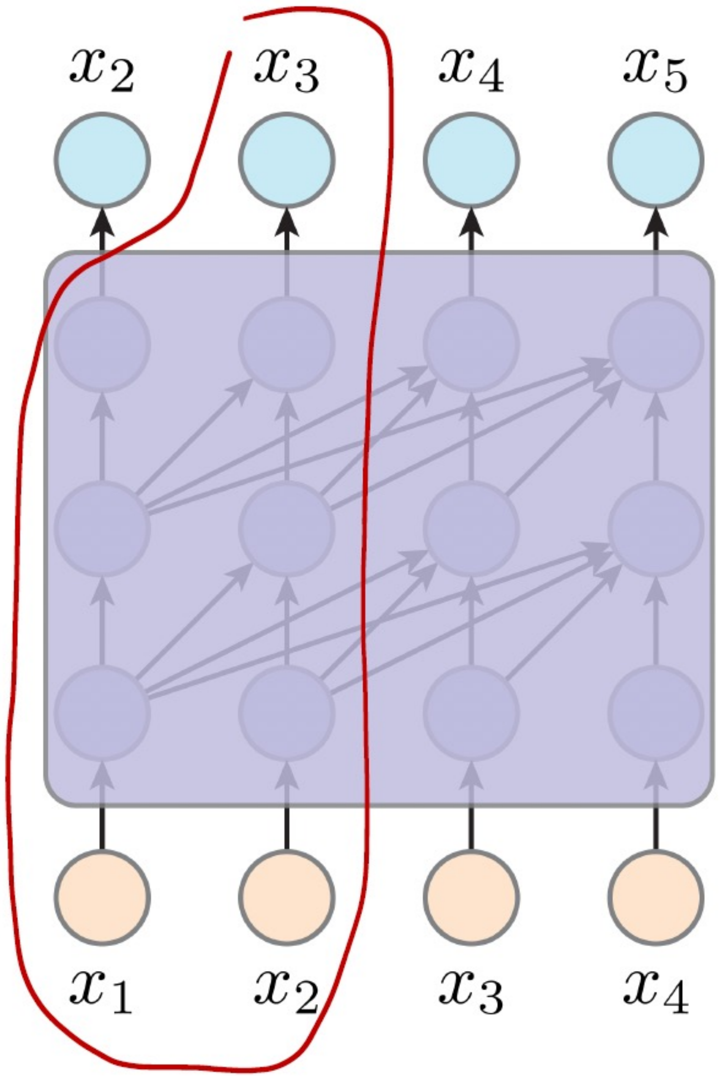
自回归模型

$p(x_2 \mid x_1)$

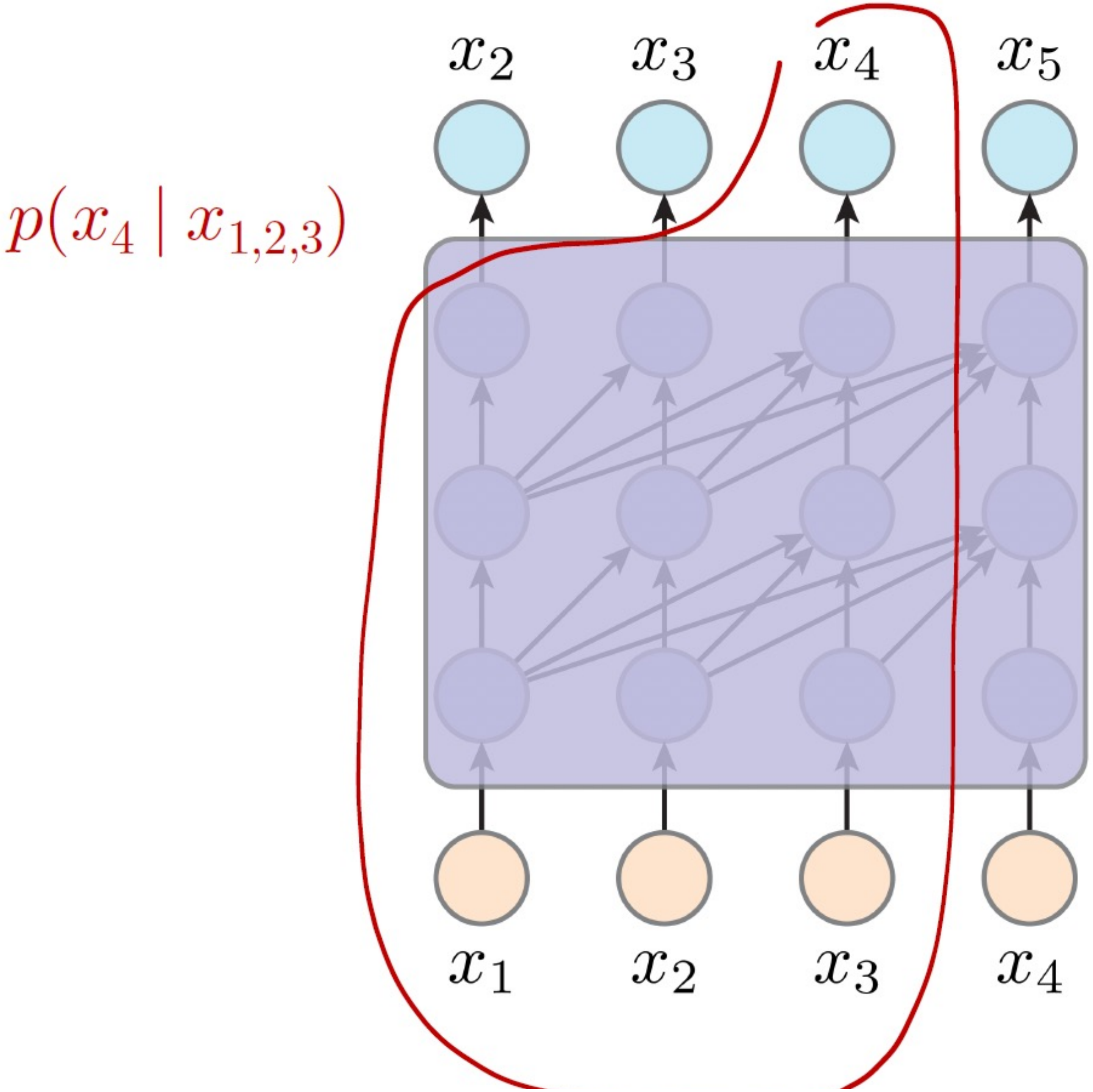


自回归模型

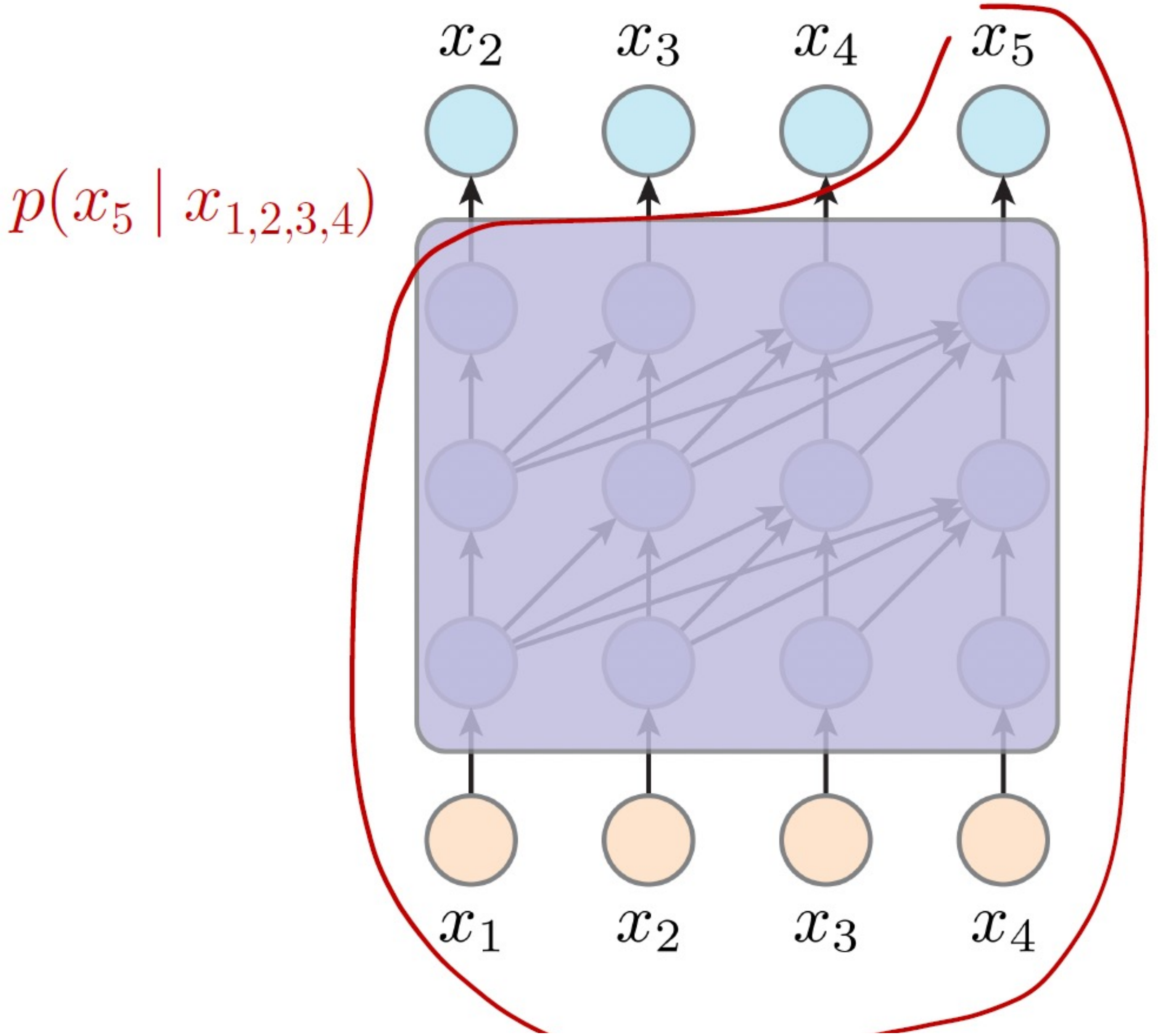
$p(x_3 \mid x_{1,2})$



自回归模型

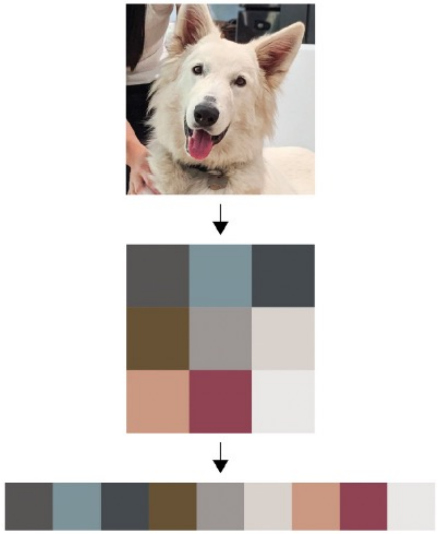


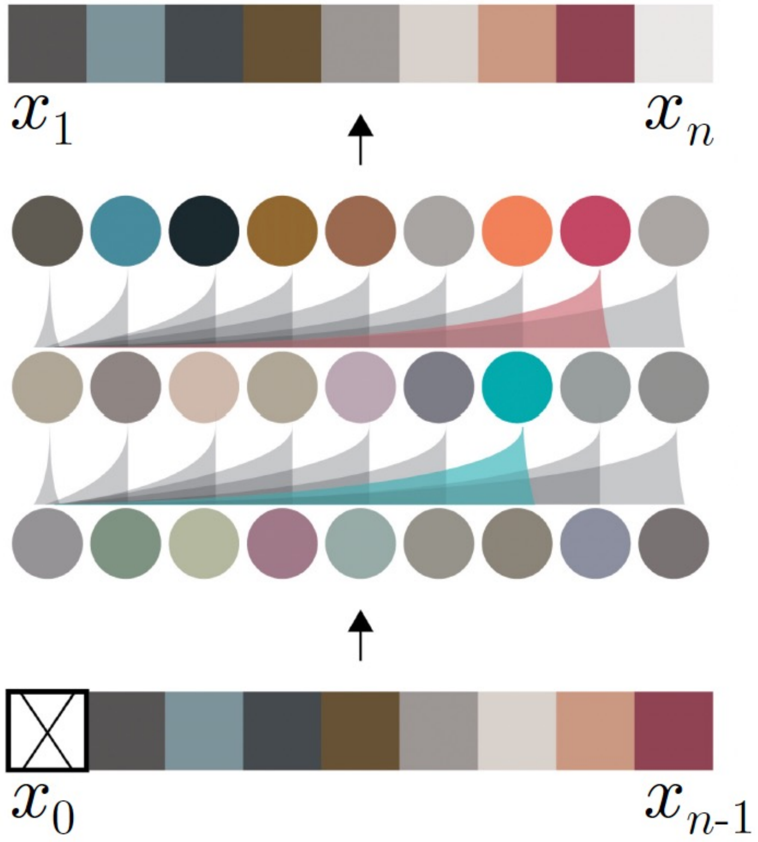
自回归模型



自回归模型

- 例子：image GPT

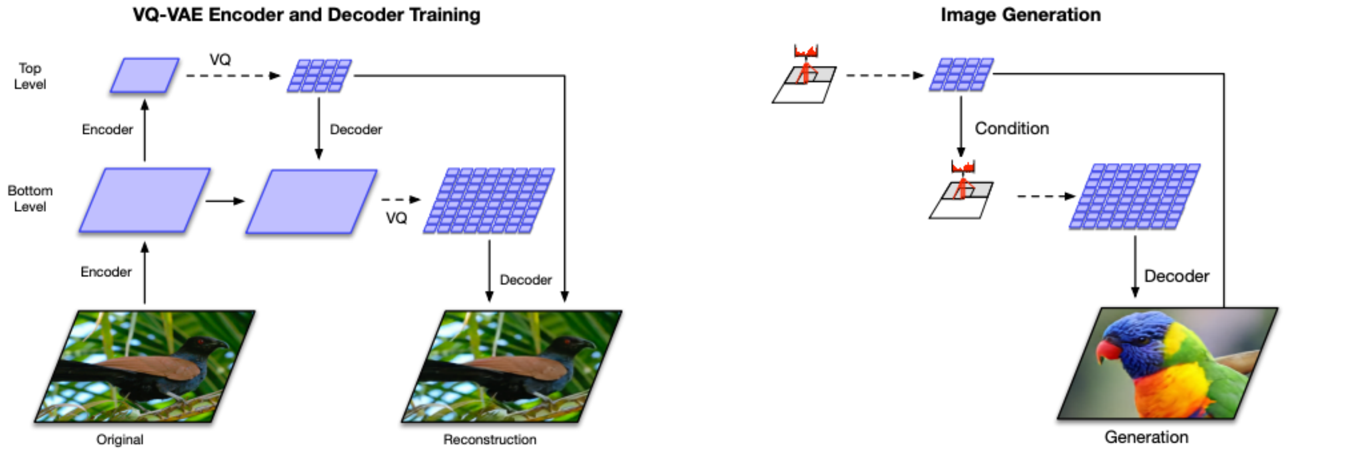




Generative Pretraining from Pixels

VQ VAE 2

- 利用PixelCNN生成VQ VAE的潜在变量
- 建立VQ VAE的潜在变量在连续空间之间联系



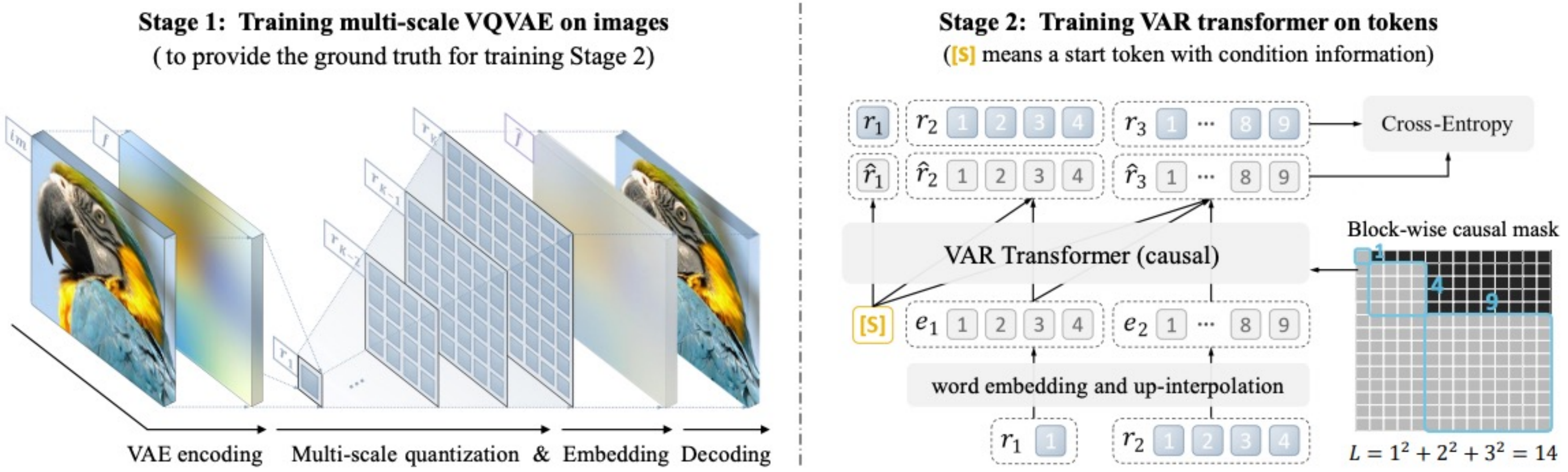
(a) Overview of the architecture of our hierarchical VQ-VAE. The encoders and decoders consist of deep neural networks. The input to the model is a 256×256 image that is compressed to quantized latent maps of size 64×64 and 32×32 for the *bottom* and *top* levels, respectively. The decoder reconstructs the image from the two latent maps.

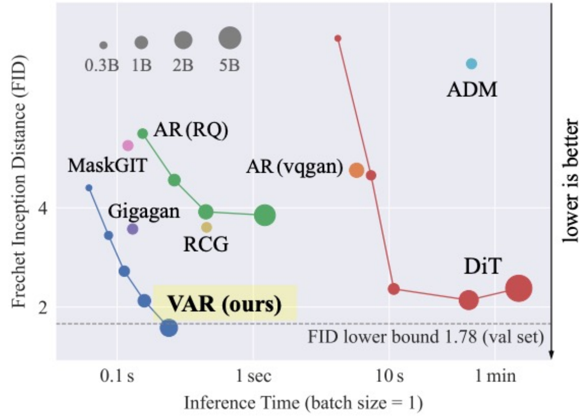
(b) Multi-stage image generation. The top-level PixelCNN prior is conditioned on the class label, the bottom level PixelCNN is conditioned on the class label as well as the first level code. Thanks to the feed-forward decoder, the mapping between latents to pixels is fast. (The example image with a parrot is generated with this model).

Generating Diverse High-Fidelity Images with VQ-VAE-2

Visual autoregressive modeling

- 建立VQ VAE的潜在变量在不同尺度之间联系
- 当下图像生成最先进范式





Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction

目录

1 生成模型基础

2 独立潜在变量建模

3 条件分布建模

4 现实问题建模

现实问题建模

- 生成模型目的是得到 $p(x|y)$

什么可以是y?

- 条件
- 限制
- 类别
- 属性

- 更抽象
- 更少信息量

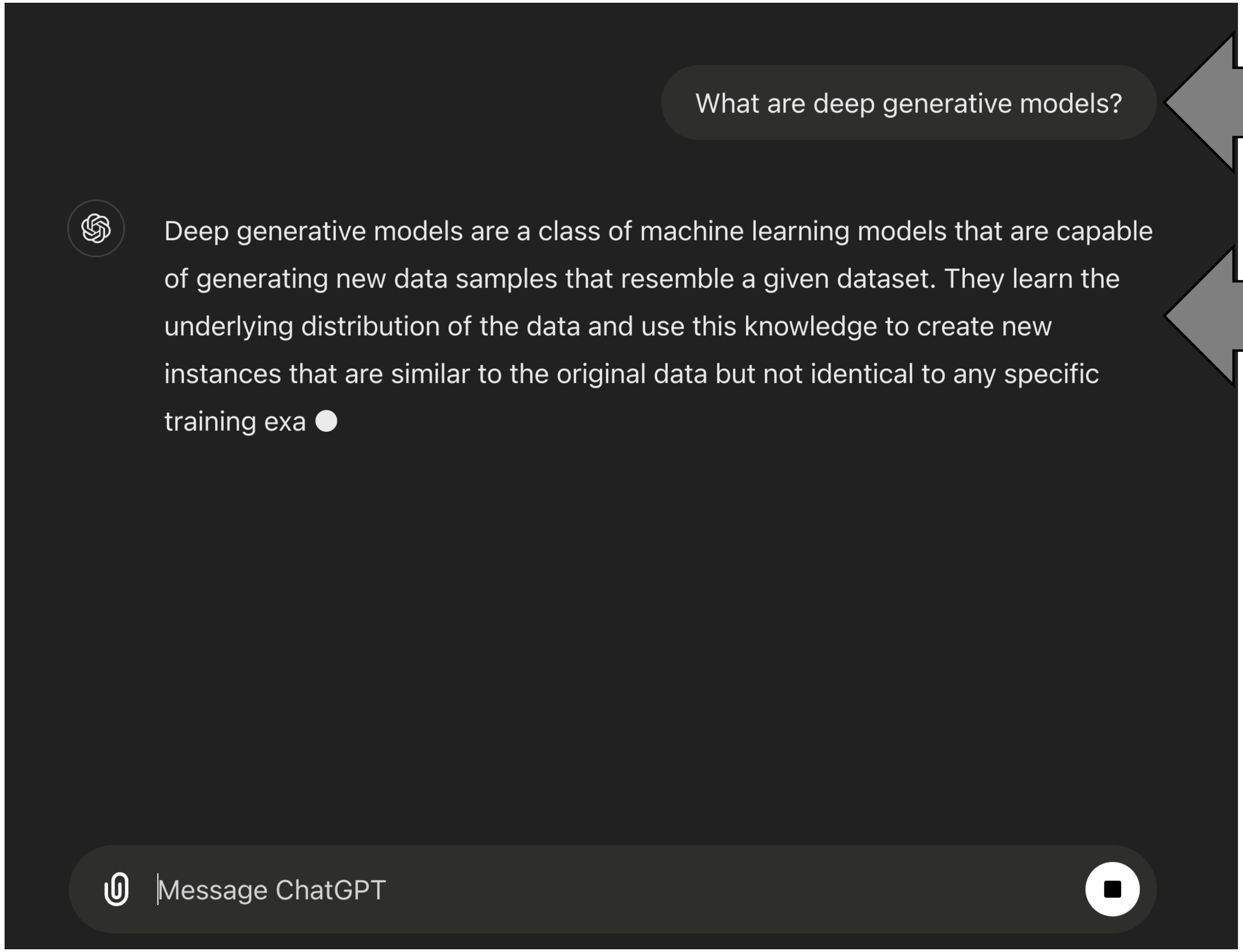
什么可以是x?

- 数据
- 样本
- 观测结果
- 测量结果

- 更具体
- 更多信息量

生成模型中的 $p(x|y)$

- 自然语言对话



y: 文字提示

x: 系统的回答

生成模型中的 $p(x|y)$

- 文生图和文生视频

Prompt: teddy bear teaching a course, with "generative models" written on blackboar

y: 文字提示

x: 生成的图片

生成模型中的 $p(x|y)$

- 文生3D结构

x: 生成的3D结构

y: 文字提示

生成模型中的 $p(x|y)$

- 类别条件的图像生成

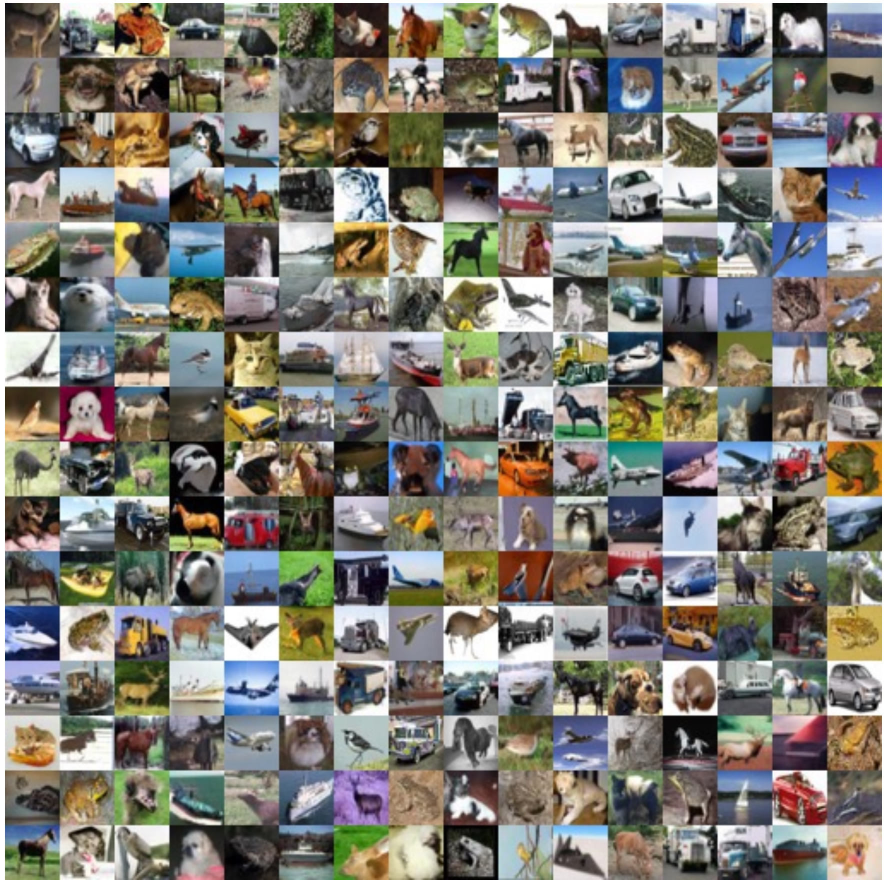
“red fox”

y: 类别提示

x: 生成的图片

生成模型中的 $p(x|y)$

- “无条件的”图像生成



y: 一个隐式条件

“images following CIFAR10 distribution”

x: 生成的类似CIFAR-10图片

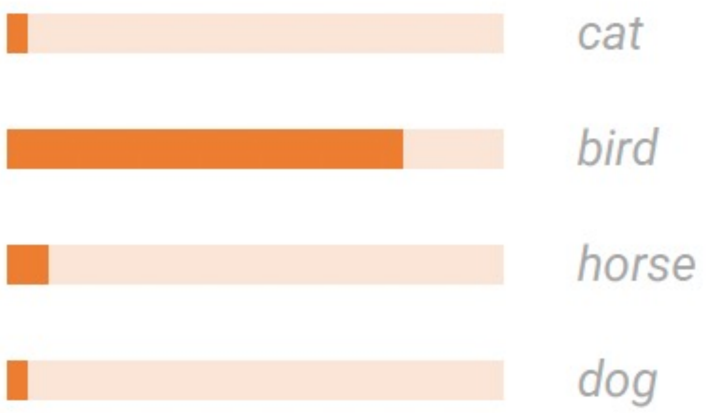
- $p(x|y)$：从CIFAR10中采样的数据
- $p(x)$：所有自然图像

生成模型中的 $p(x|y)$

- 图像分类

y: 输入的图像

x: 图像的类别概率

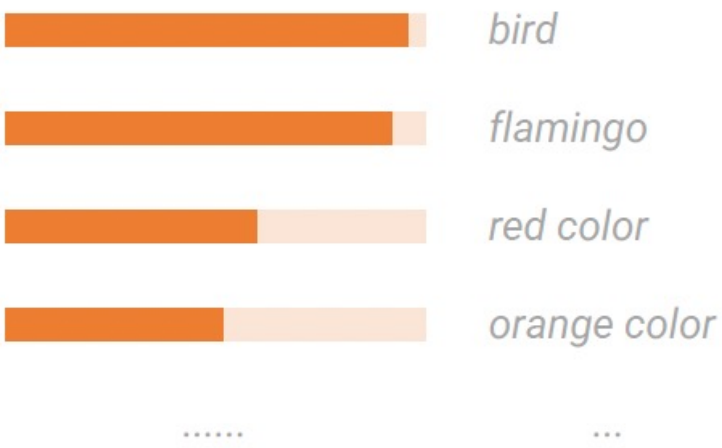


生成模型中的 $p(x|y)$

- 开放词汇识别

y: 输入的图像

x: 图像的可能描述



生成模型中的 $p(x|y)$

- 图像描述

y: 输入的图像

x: 图像的可能描述

a baseball player with a catcher and umpire on top of a baseball field.
a baseball player is sliding into a base.
a baseball player swings at a pitch with the pitcher and umpire behind him.
baseball player with bat in the baseball game.
a batter in the process on the bat in a baseball game.

生成模型中的 $p(x|y)$

- 多模态的自然语言对话

User　What is unusual about this image?

Source: `https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg`

y: 图片文字提示

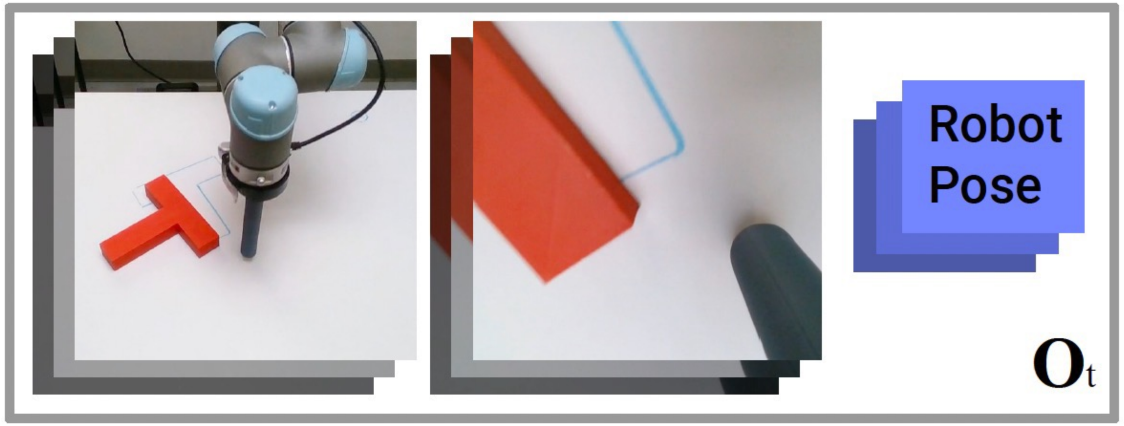
GPT-4　The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.

x: 系统的回答

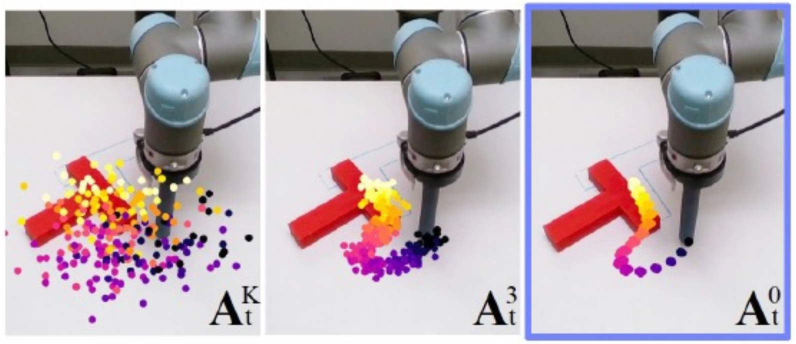
生成模型中的 $p(x|y)$

- 机器人的行为策略

y: 视觉和传感器观测



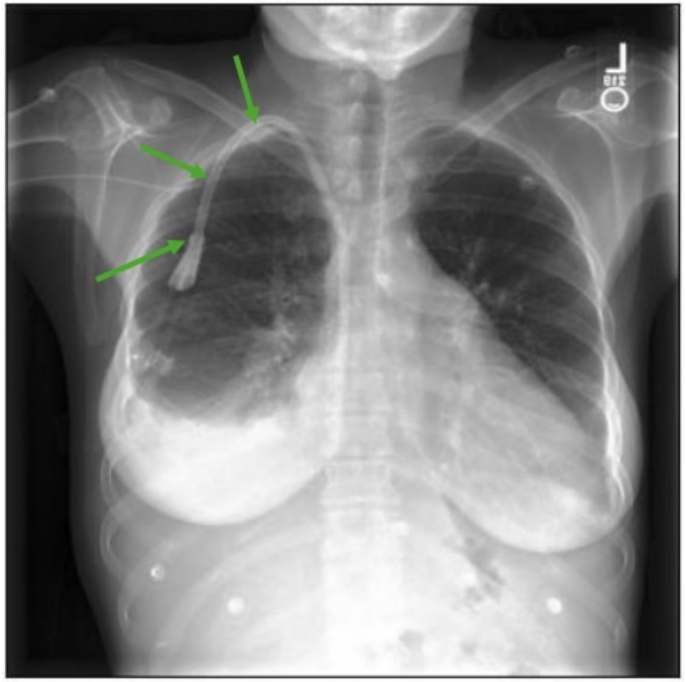
x: 策略（下一步动作）



生成模型中的 $p(x|y)$

- 医学报告生成

y: 医学图像



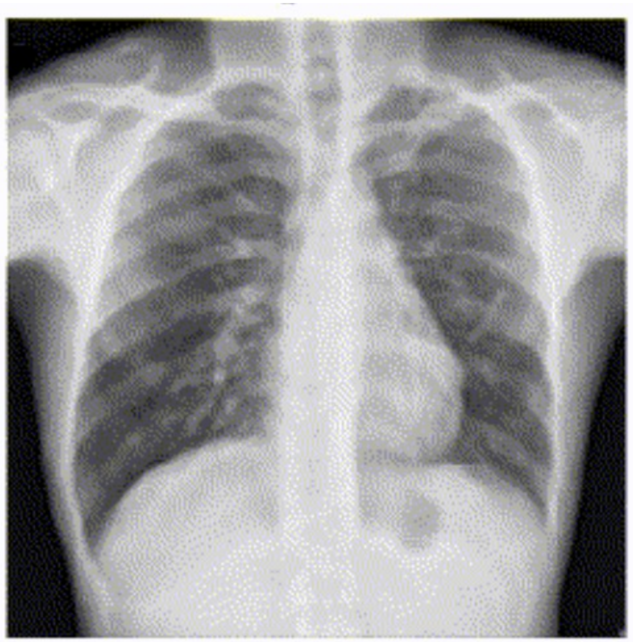
x: 分析报告

The chest radiograph demonstrates bilateral pleural effusions, more pronounced on the right, with associated bibasilar atelectasis. There is evidence of pulmonary vascular congestion and cardiomegaly. There is no pneumothorax. A right-sided dialysis catheter is visualized. The overall appearance is consistent with pulmonary edema. The combinations of findings suggests fluid overload, possibly related to renal dysfunction requiring dialysis.

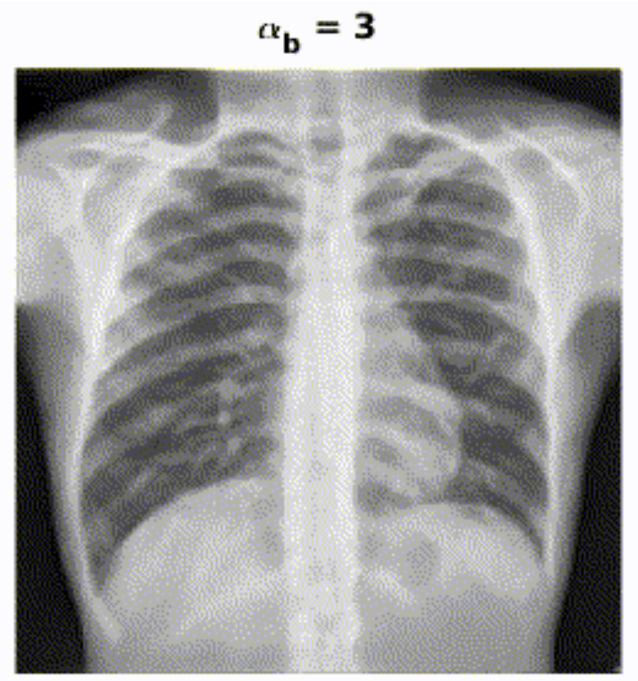
生成模型中的 $p(x|y)$

- 医学图像反事实生成

y: 医学图像和反事实条件



x: 生成的图像



现实问题建模

- 生成模型目的是得到 $p(x|y)$
- 要确定什么是x，什么是y
- 要判断怎么建立x和y的表征，以及他们的关系

小结

- 生成模型利用深度神经网络表征数据及其分布
- VAE模型可以建立独立潜在变量的建模，生成可控但过于简化
- 条件分布建模是高质量生成的关键，AR是一个通用范式
- 生成模型建模比较灵活，理论上所有机器学习问题都可以定义为生成模型