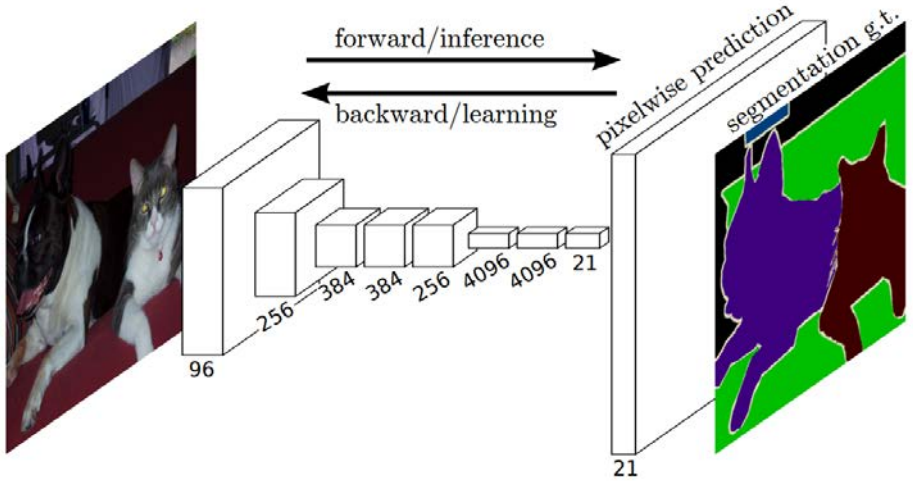
基于深度学习的图像分析技术

- 现代深度学习技术实例
  - 图像识别
  - 目标检测
  - 语义分割
  - 视频理解
  - 目标跟踪
  - 注意力机制及Transformer
  - 生成式模型
  - 自监督学习

语义分割：FCN

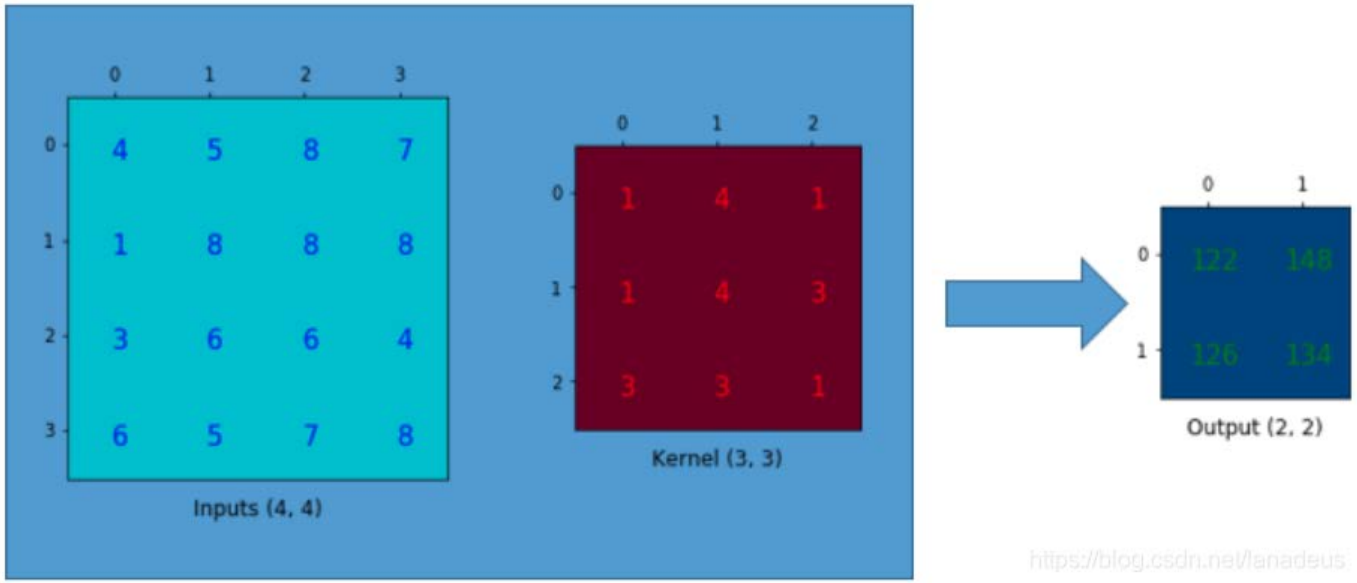
- 主要思想
  - 全卷积 (Fully Convolution)
    - ✓ 将网络中的全连接层变为卷积层
    - ✓ 卷积核的大小即为输入特征图的大小
    - ✓ 卷积核的数量即为全连接层输出神经元的个数
  - 通过FCN将语义分割转化为对图中每个点的分类
  - 采用下采样-上采样结构
  - 改进：将由不同下采样倍数的特征得到的mask结合起来，得到最终分割结果



- J. Long, E. Shelhamer and T. Darrell. “Fully Convolutional Networks for Semantic Segmentation”. In CVPR, 2015.

语义分割：FCN

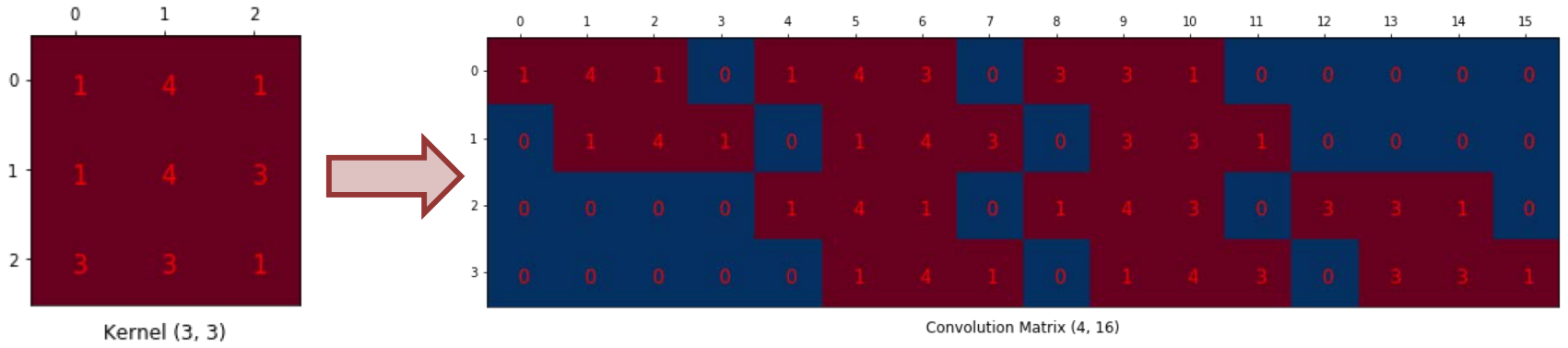
- 传统卷积 (convolution)
  - 卷积运算：矩阵相关



4×4　　3×3　　2×2

语义分割：FCN

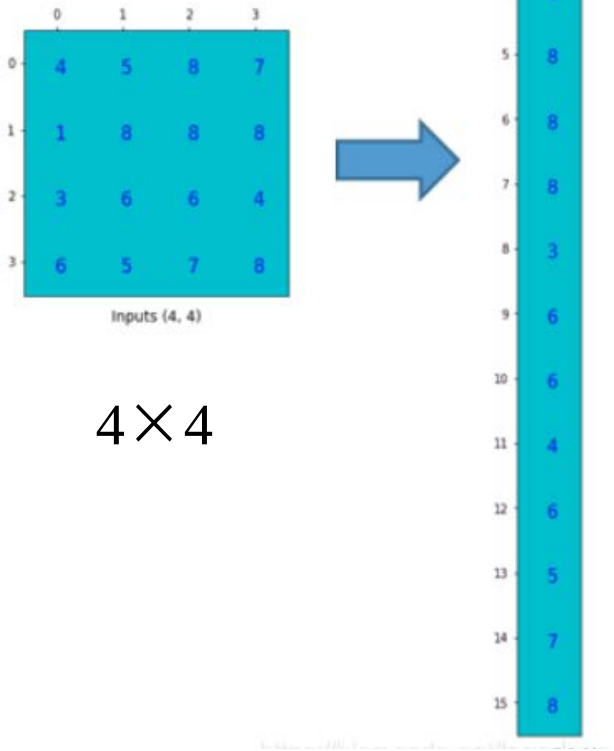
- 传统卷积 (convolution)
  - 具体操作：
    - ① 卷积核展开，空位补零



3×3　　4×16

语义分割：FCN

- 传统卷积 (convolution)
  - 具体操作：
    - ② 输入特征展开，拉伸为列向量

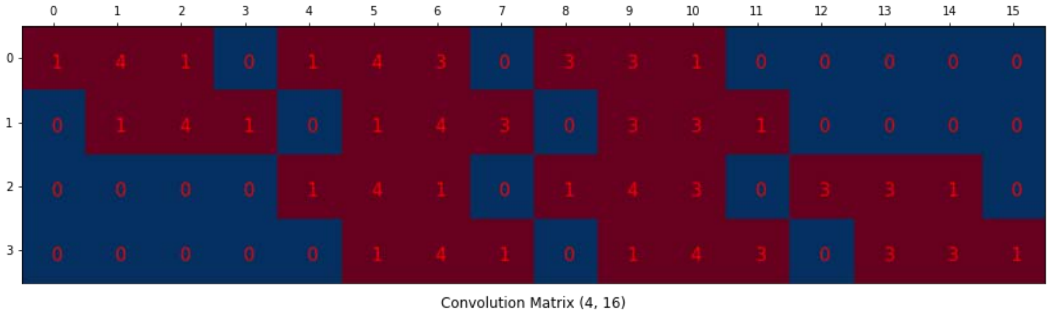


4×4

16×1

语义分割：FCN

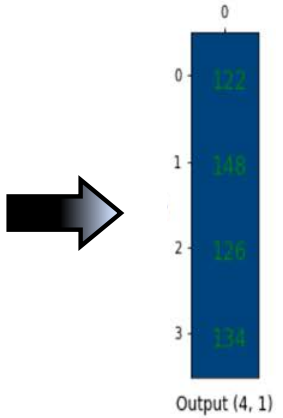
- 传统卷积 (convolution)
  - 具体操作：
    - ③ 卷积核与输入特征进行矩阵相乘



4×16

×

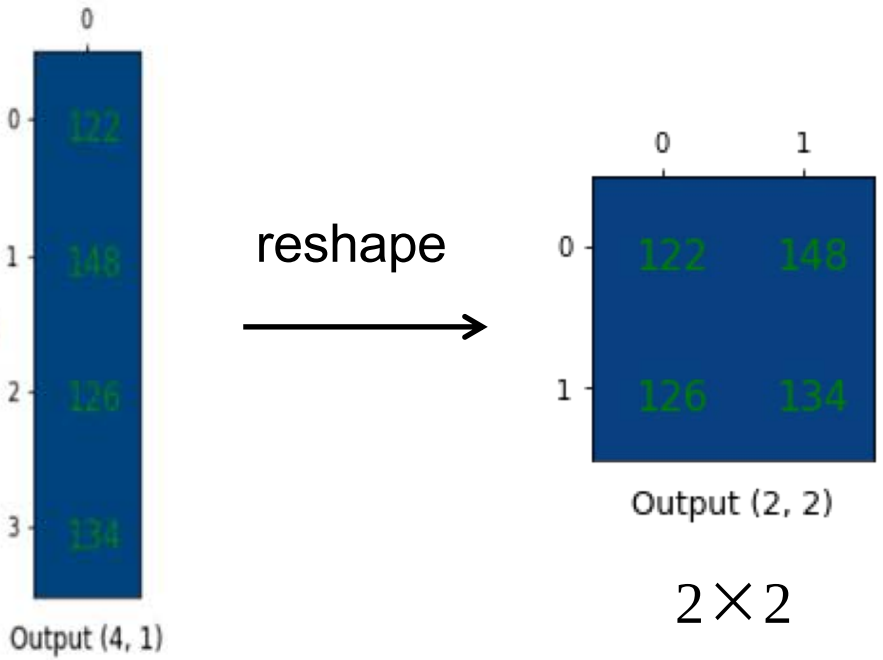
16×1



4×1

语义分割：FCN

- 传统卷积 (convolution)
  - 具体操作：
    - ④ 输出reshape



4×1

2×2

语义分割：FCN

- 转置卷积 (transposed convolution)
  - 将展开后的卷积核转置

$4\times16 \longrightarrow 16\times4$

  - 输入替换为更小尺寸的特征图

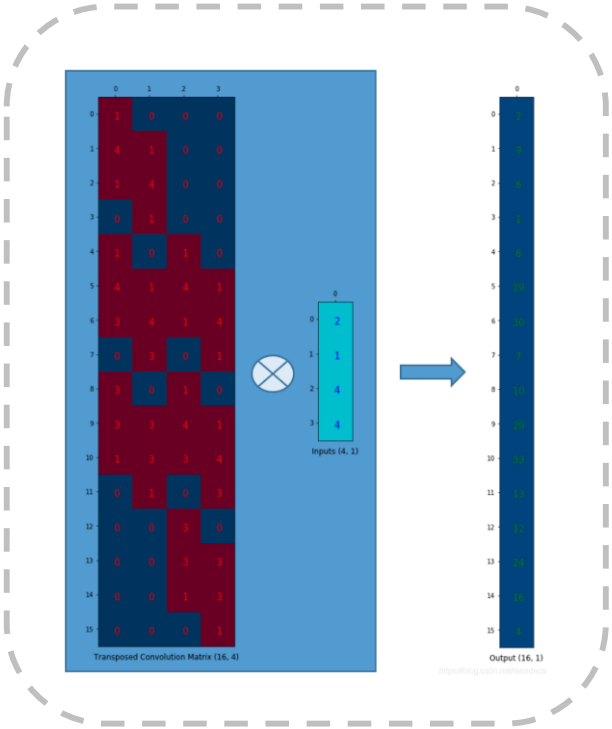
$16\times1 \longrightarrow 4\times1$

  - 矩阵相乘

$16\times4 \times 4\times1 \longrightarrow 16\times1 \longrightarrow 4\times4$

reshape

转置卷积示意图



语义分割：FCN

输入

2	1
4	4

$\otimes$

1	4	1
1	4	3
3	3	1

卷积核

$=$

2	8	2	
2	8	6	
6	6	2	

$+$

	1	4	1
	1	4	3
	3	3	1

$+$

4	16	4	
4	16	12	
12	12	4	

$+$

	4	16	4
	4	16	12
	12	12	4

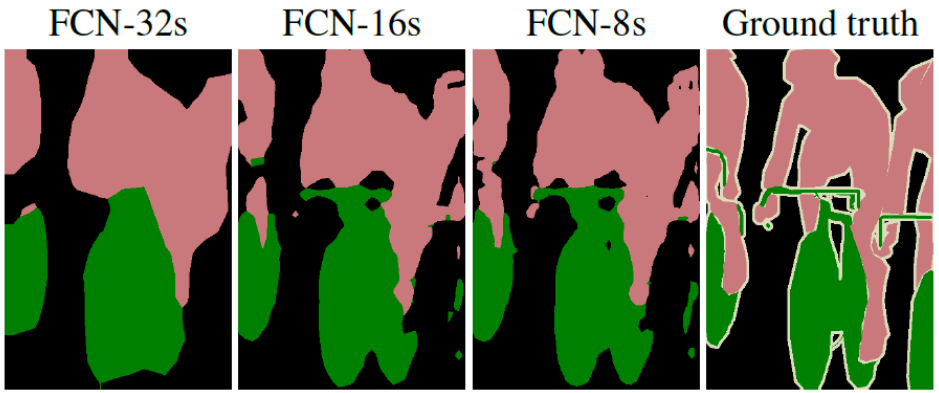
$=$

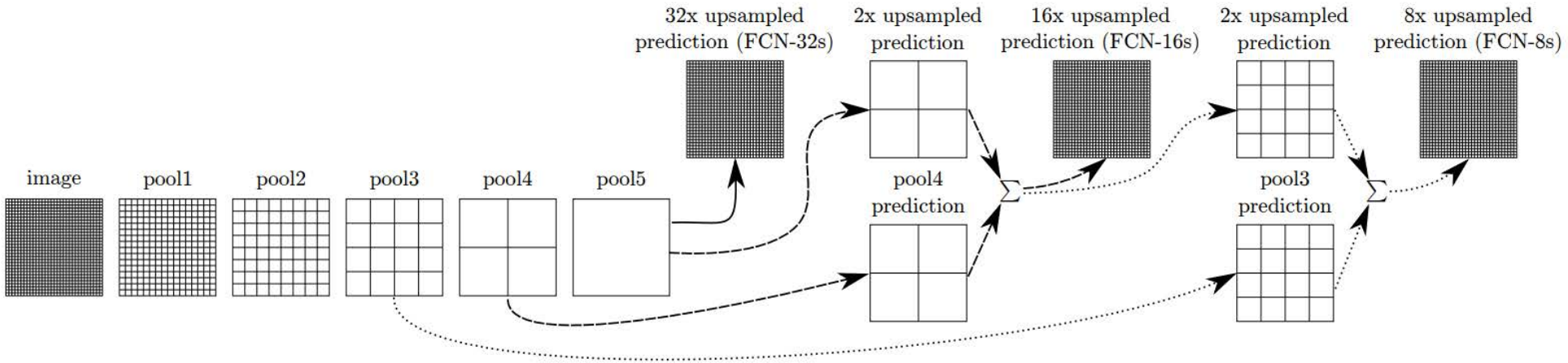
2	9	6	1
6	29	30	7
10	29	33	13
12	24	16	4

输出

语义分割：FCN

- 转置卷积 (transposed convolution)
  - 用于特征图的上采样
  - 可学习的卷积核参数
  - 代替双线性插值
- 特征图的跨层连接 (skip)
  - 上采样 + 逐像素加
  - 补充图像细节

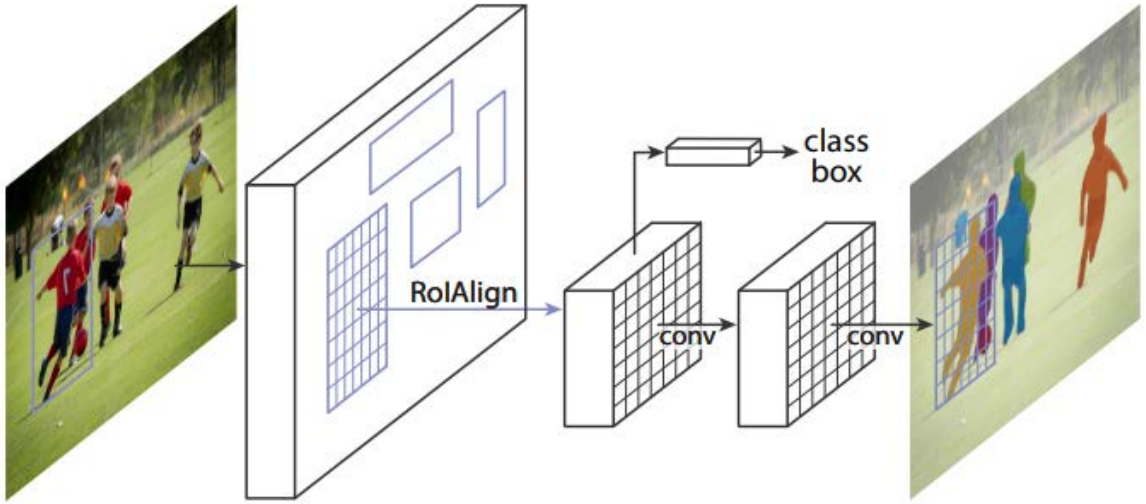




- J. Long, E. Shelhamer and T. Darrell. “Fully Convolutional Networks for Semantic Segmentation”. In CVPR, 2015.

实例分割：Mask R-CNN

- 方法动机
  - 两阶段算法：在实例目标框先验下预测mask
- 解决方案
  - Faster R-CNN基础上增加一个mask head
- 结构框架

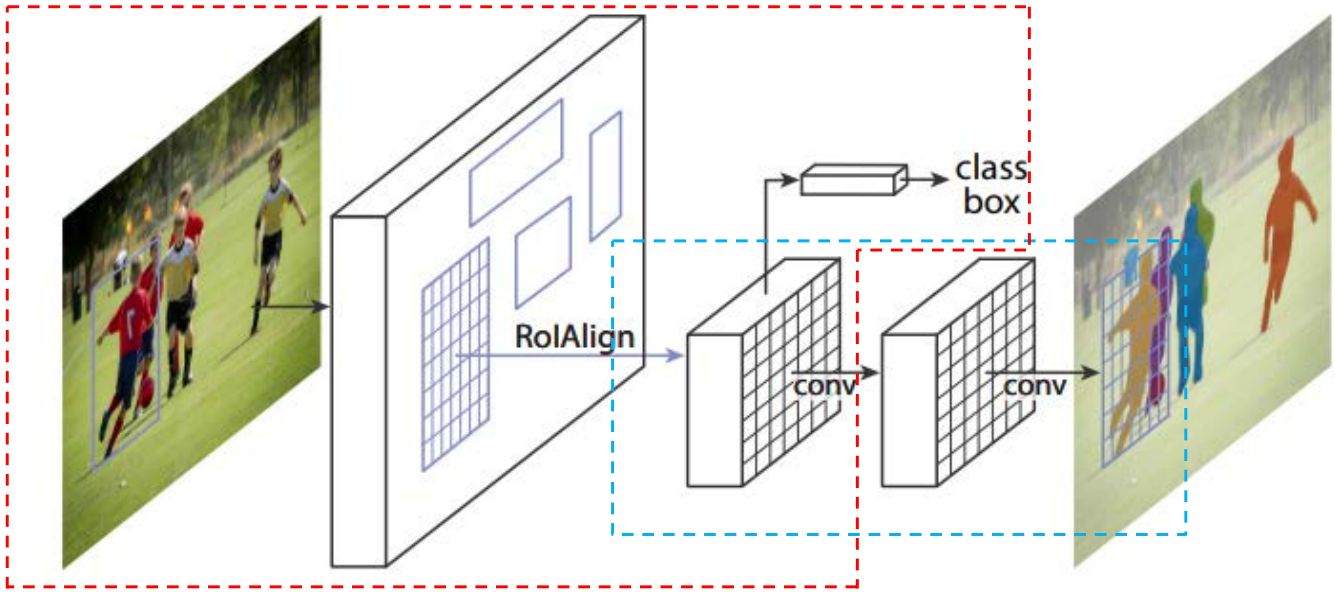


- K. He, et al., “Mask R-CNN”. In ICCV, 2017.

实例分割：Mask R-CNN

□ 结构框架

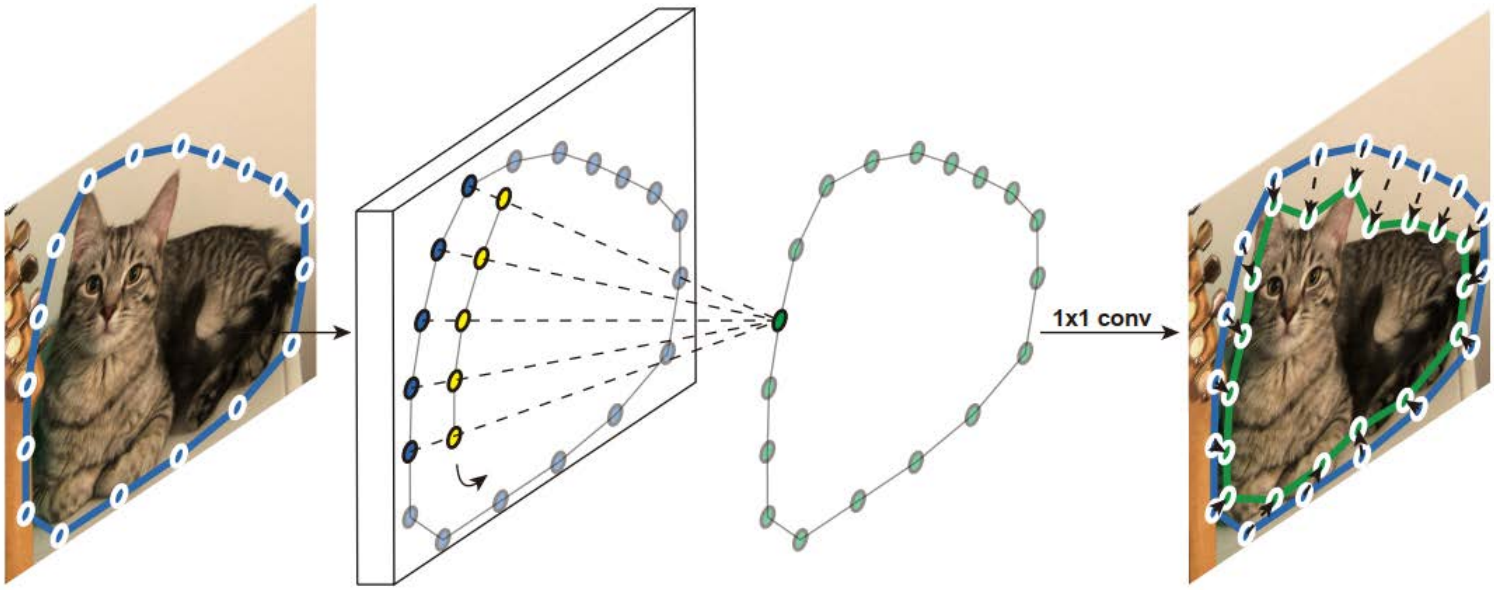
Faster R-CNN: 检测物体，得到目标框



Mask Head: 在目标框内回归前景Mask

实例分割：Deep Snake

- 方法动机
  - 以预测实例轮廓的方式实现实例分割
  - 提取轮廓点特征，预测位移残差，变形得到新轮廓
- 基本方法

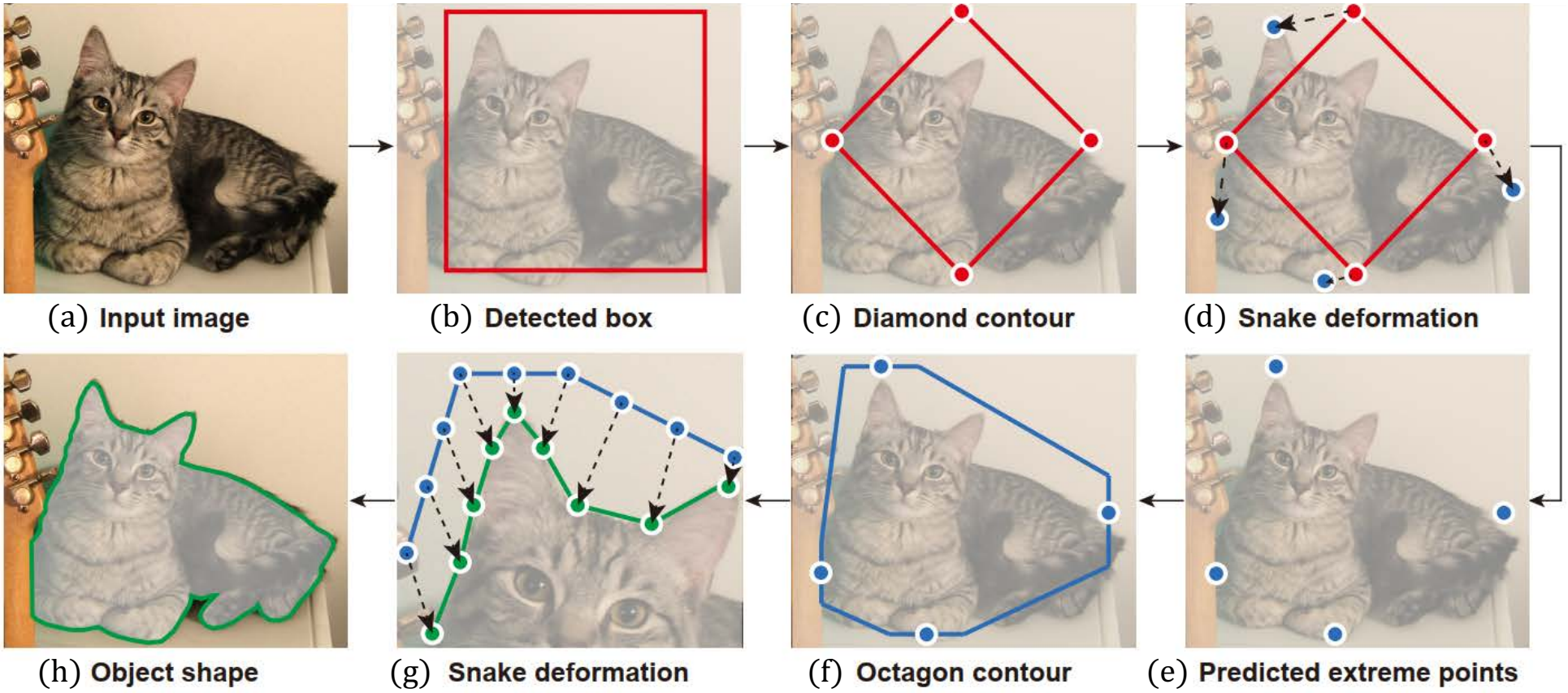


- S. Peng, et al. “Deep Snake for Real-Time Instance Segmentation”. In CVPR, 2020.

实例分割：Deep Snake

方法框架结构

- 检测出输入图像(a)中的目标(b)
- 初始化菱形轮廓(c)，提取特征预测残差(d)，变形得极点(e)
- 初始化八边形轮廓(f)，提取特征预测残差(g)，变形得物体轮廓(h)



基于深度学习的图像分析技术

- 现代深度学习技术实例
  - 图像识别
  - 目标检测
  - 语义分割
  - 视频理解
  - 目标跟踪
  - 注意力机制及Transformer
  - 生成对抗网络
  - 自监督学习

视频理解

□ 动作识别

图像：识别物体 →

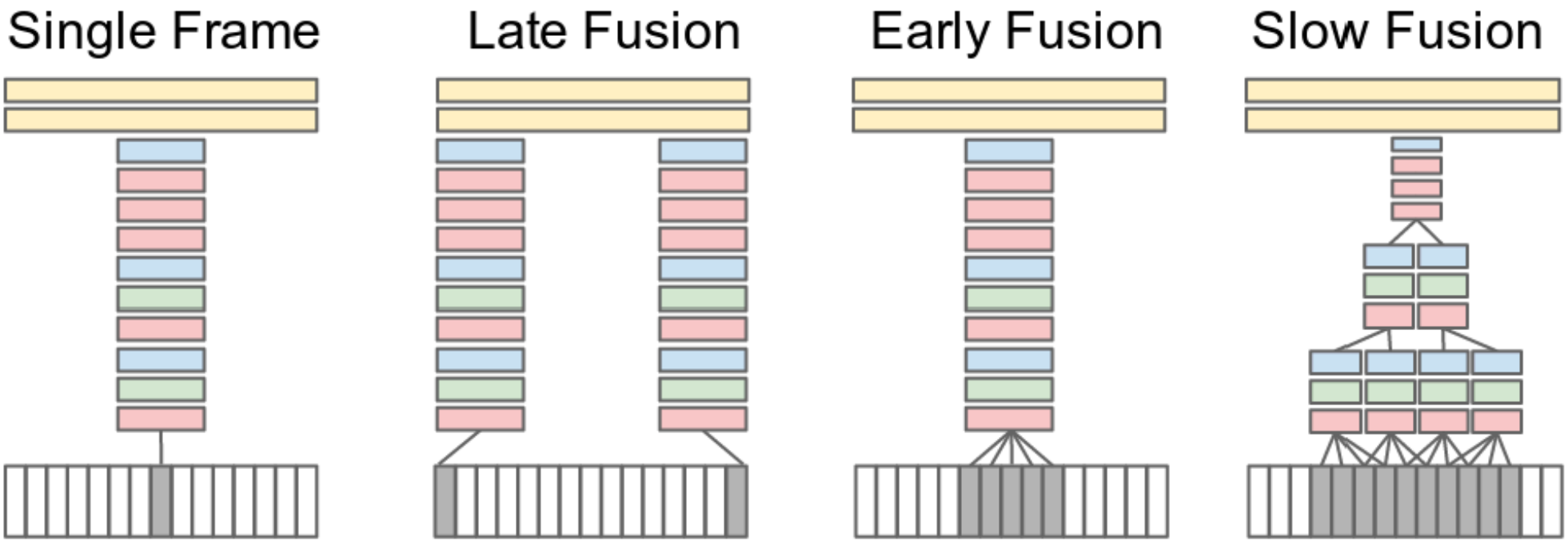
狗
猫
汽车
行人

视频：识别动作 →

游泳
跑步
打网球
吃饭
跳跃

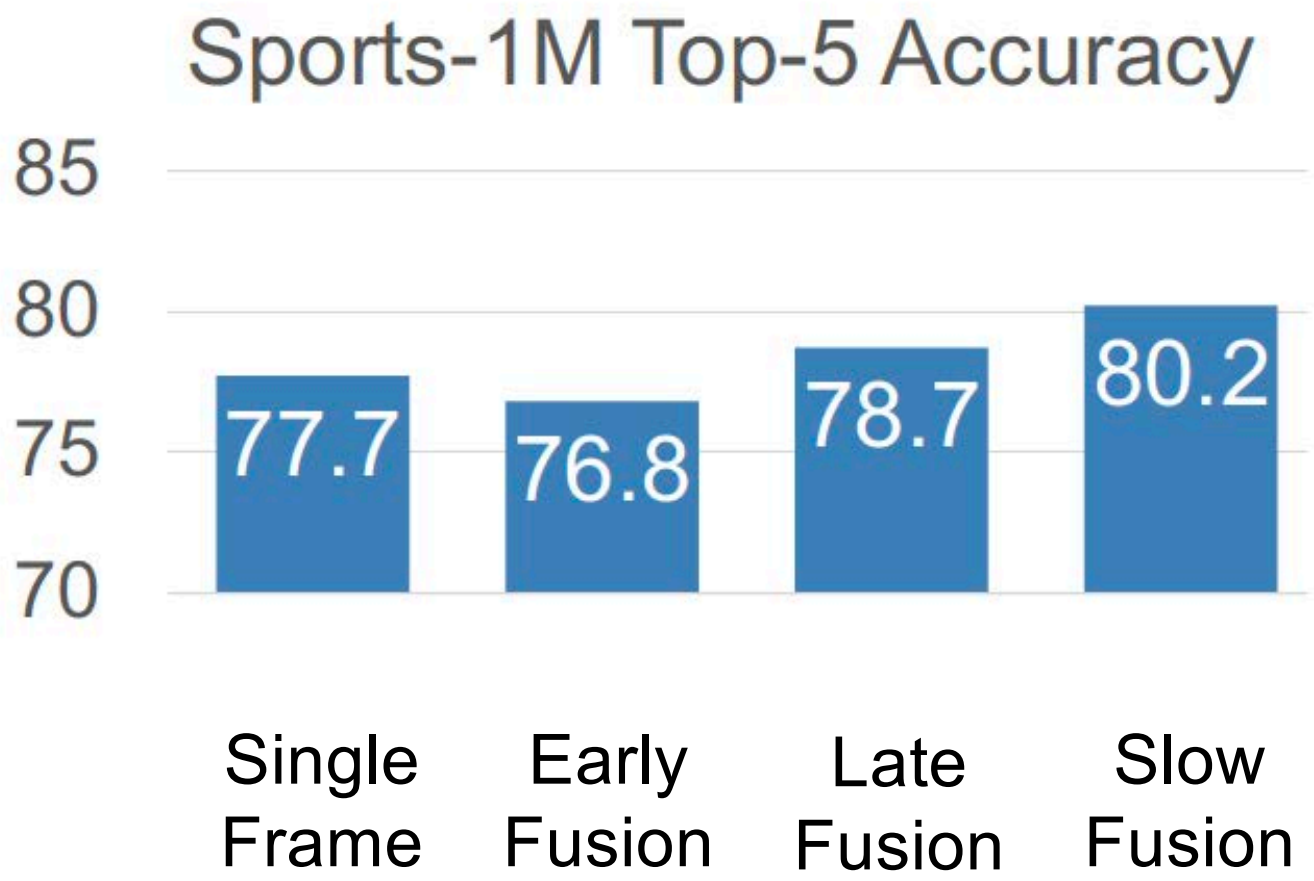
视频分类

- Simple fusion



- A. Karpathy et al. “Large-scale Video Classification with Convolutional Neural Networks”. In CVPR, 2014.

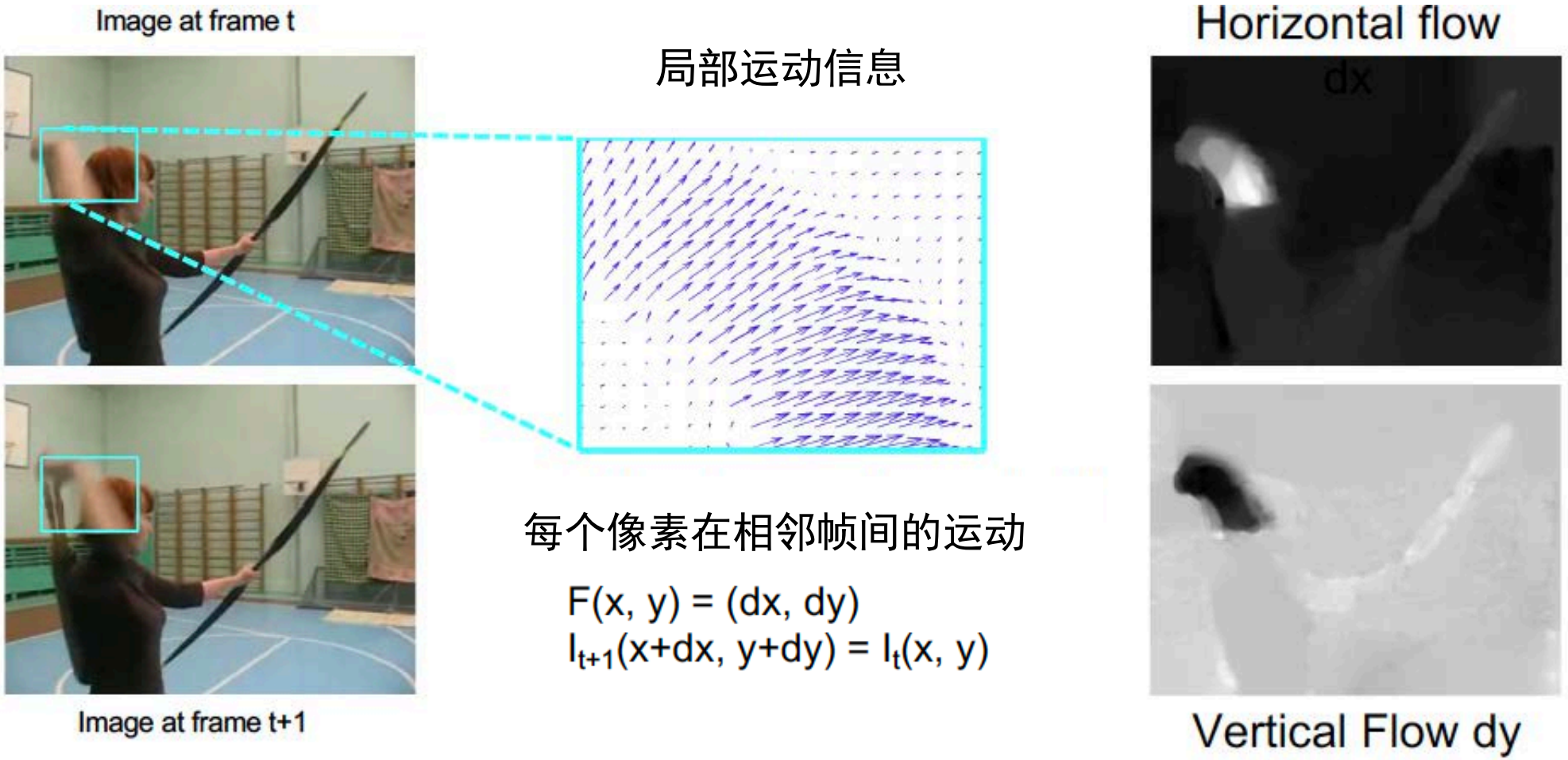
视频分类



- A. Karpathy et al. “Large-scale Video Classification with Convolutional Neural Networks”. In CVPR, 2014.

视频分类

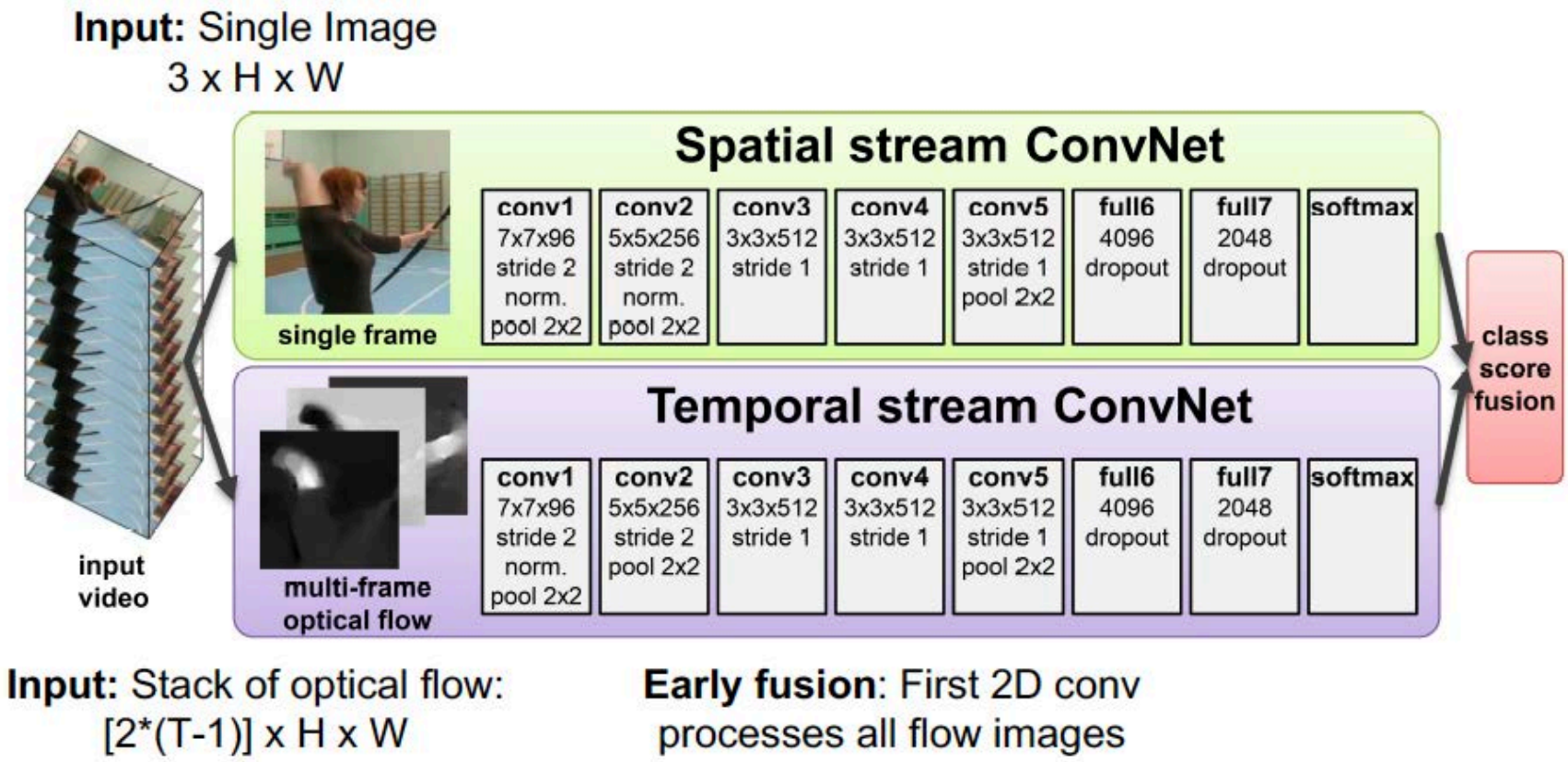
□ 运动信息：光流估计



- K. Simonyan and A. Zisserman. “Two-stream convolutional networks for action recognition in videos”. In NeurIPS, 2014.

视频分类

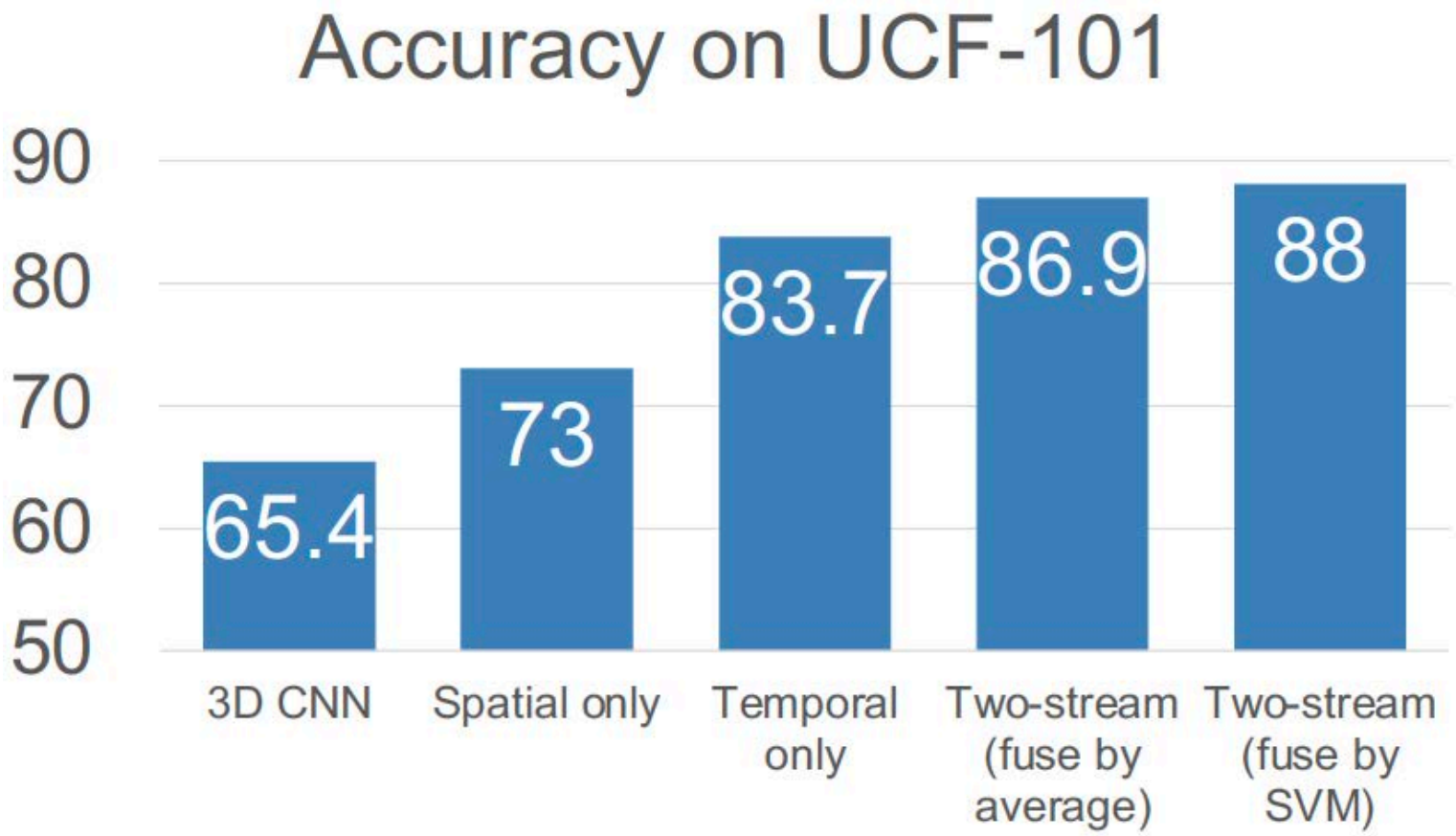
□ 双流网络



- K. Simonyan and A. Zisserman. “Two-stream convolutional networks for action recognition in videos”. In NeurIPS, 2014.

视频分类

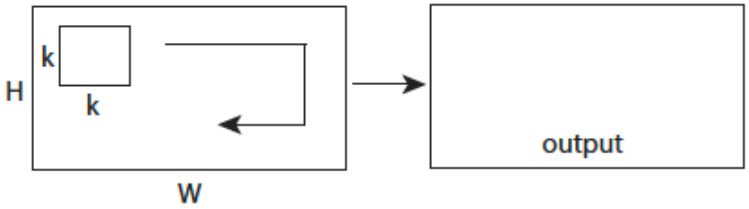
□ 双流网络



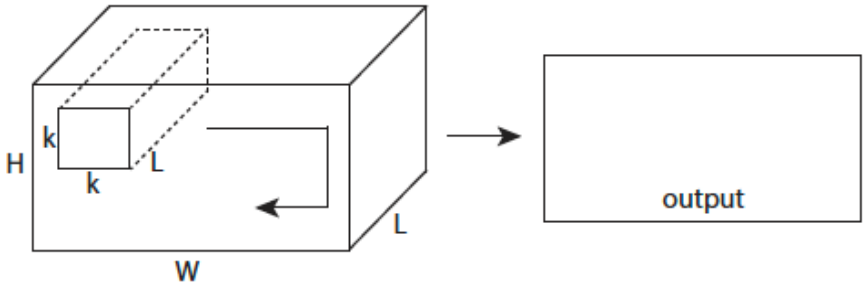
- K. Simonyan and A. Zisserman. “Two-stream convolutional networks for action recognition in videos”. In NeurIPS, 2014.

视频分类

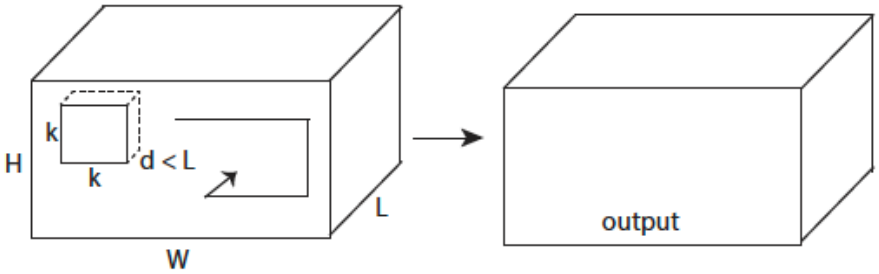
3D CNN



2D Convolution



2D Convolution on multiple frames



3D Convolution

- D. Tran et al. “Learning Spatiotemporal Features with 3D Convolutional Networks”. In ICCV, 2015.

视频分类

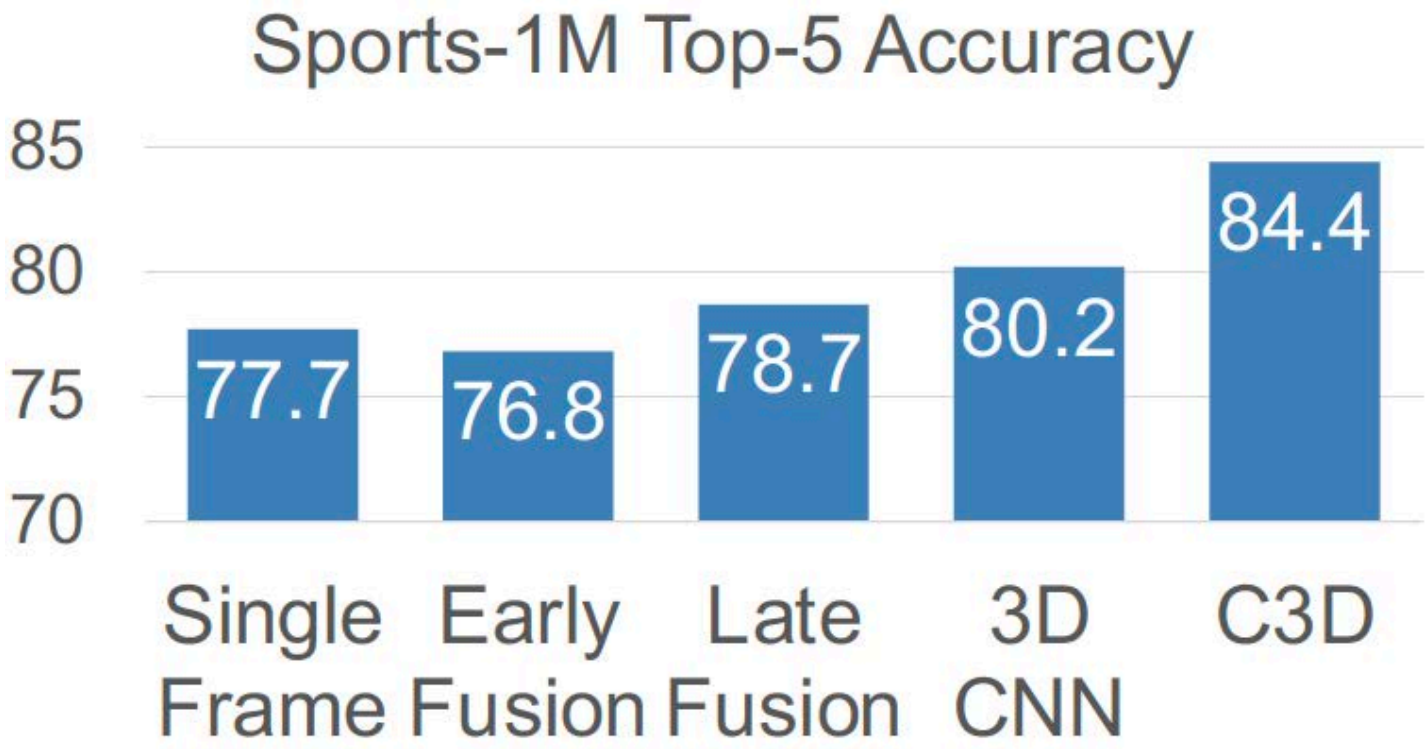
C3D:The VGG of 3D CNNs

- 借鉴VGG，采用3x3x3卷积核和2x2x2池化
- Sports-1M预训练模型，常被用作视频特征提取器
- 问题：运算量巨大
  AlexNet: 0.7 GFLOPS
  VGG-16: 13.6 GFLOPS
  C3D: 39.5 GFLOPS (2.9x VGG)

Layer	Size	MFLOPs
Input	3 x 16 x 112 x 112	
Conv1 (3x3x3)	64 x 16 x 112 x 112	1.04
Pool1 (1x2x2)	64 x 16 x 56 x 56	
Conv2 (3x3x3)	128 x 16 x 56 x 56	11.10
Pool2 (2x2x2)	128 x 8 x 28 x 28	
Conv3a (3x3x3)	256 x 8 x 28 x 28	5.55
Conv3b (3x3x3)	256 x 8 x 28 x 28	11.10
Pool3 (2x2x2)	256 x 4 x 14 x 14	
Conv4a (3x3x3)	512 x 4 x 14 x 14	2.77
Conv4b (3x3x3)	512 x 4 x 14 x 14	5.55
Pool4 (2x2x2)	512 x 2 x 7 x 7	
Conv5a (3x3x3)	512 x 2 x 7 x 7	0.69
Conv5b (3x3x3)	512 x 2 x 7 x 7	0.69
Pool5	512 x 1 x 3 x 3	
FC6	4096	0.51
FC7	4096	0.45
FC8	C	0.05

- D. Tran et al. “Learning Spatiotemporal Features with 3D Convolutional Networks”. In ICCV, 2015.

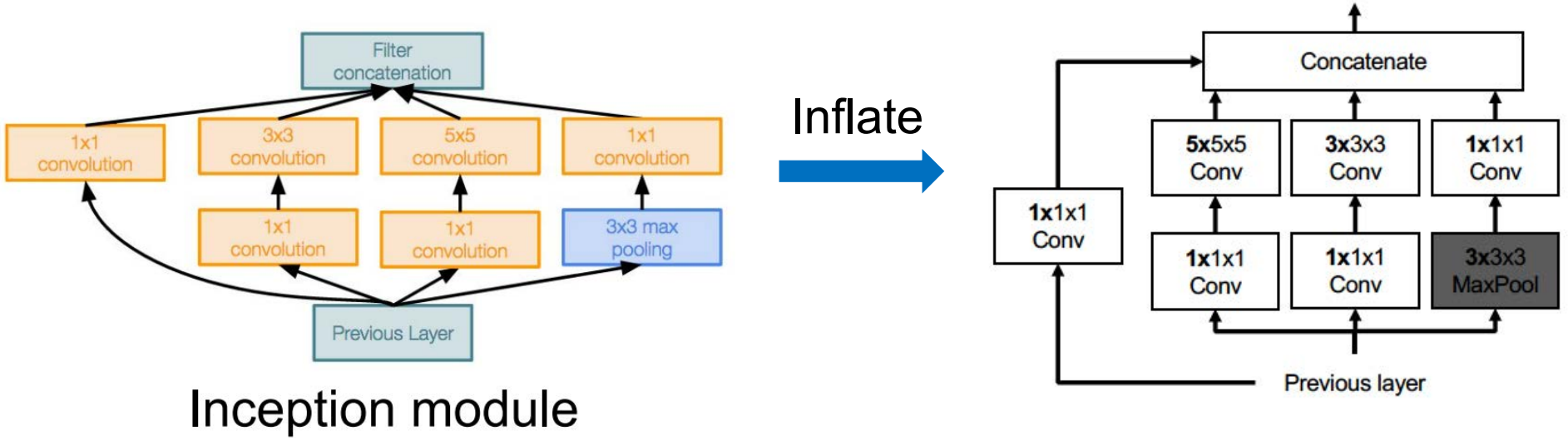
视频分类



- A. Karpathy et al. “Large-scale Video Classification with Convolutional Neural Networks”. In CVPR, 2014.
- D. Tran et al. “Learning Spatiotemporal Features with 3D Convolutional Networks”. In ICCV, 2015.

视频分类

- I3D: Inflating 2D Networks to 3D
  - 选择一个2D CNN网络
  - 将每一个2D K_hxK_w卷积/池化层替换为一个3D K_txK_hxK_w卷积/池化层

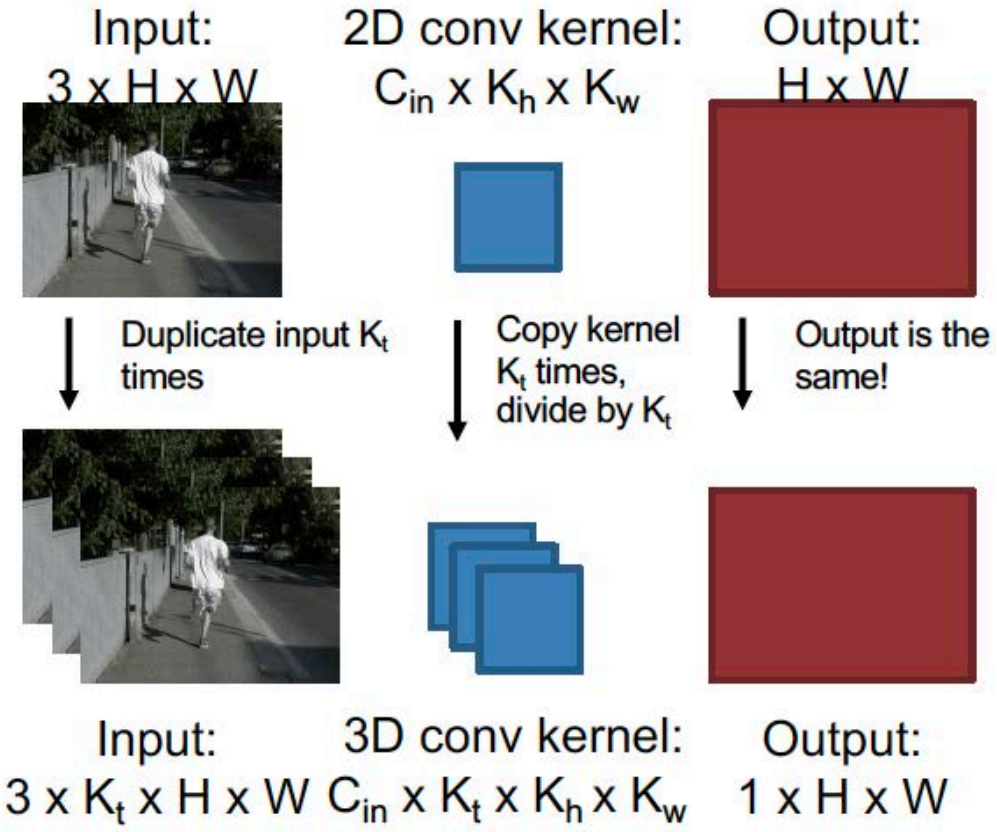


- J. Carreira and A. Zisserman. “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset”. In CVPR, 2017.

视频分类

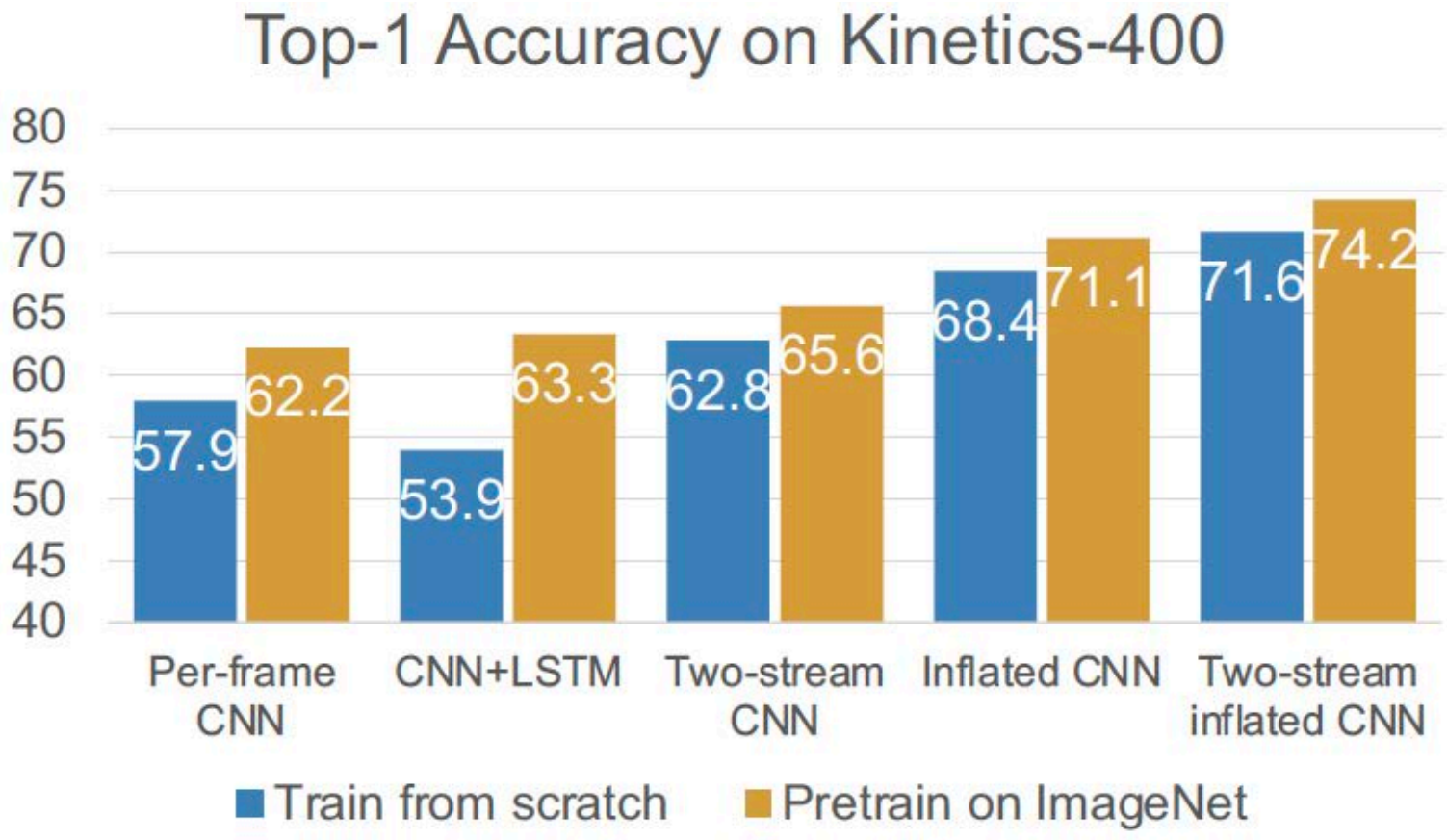
I3D: Inflating 2D Networks to 3D

- 将每一个2D K_hxK_w卷积/池化层替换为一个3D K_txK_hxK_w卷积/池化层
- 用2D卷积参数初始化3D卷积: 参数复制K_t次然后除以K_t
- 对将图像复制K_t次得到的视频，通过3D卷积结果不变



视频分类

- I3D: Inflating 2D Networks to 3D



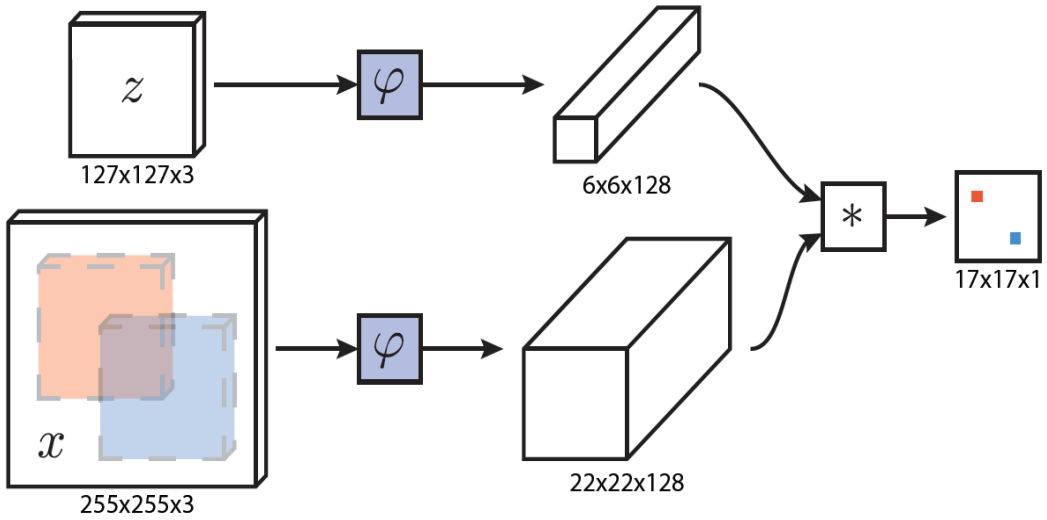
基于深度学习的图像分析技术

- 现代深度学习技术实例
  - 图像识别
  - 目标检测
  - 语义分割
  - 视频理解
  - 目标跟踪
  - 注意力机制及Transformer
  - 生成式模型
  - 自监督学习

目标跟踪：孪生网络

- 方法动机
  - 将目标跟踪任务当做相似性度量
  - 采用全卷积网络
  - 端到端训练，不引入在线更新，效率很高 (~80 FPS GPU)
- 方法框架结构及训练数据

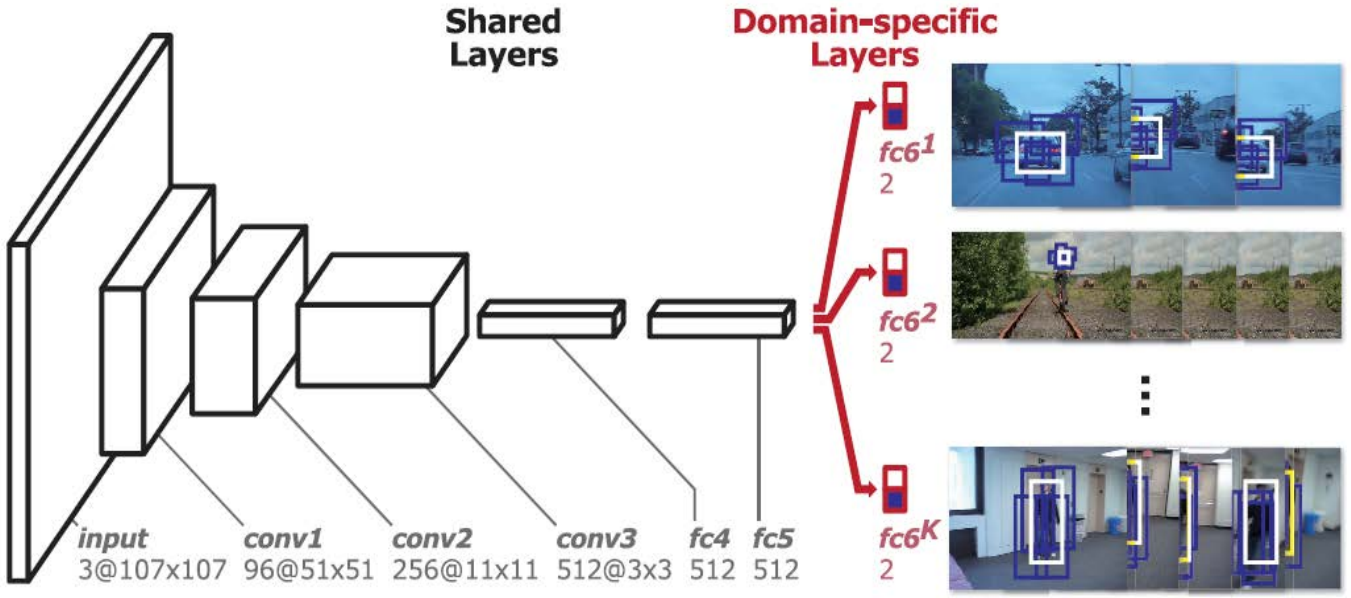
$$y[u] = \begin{cases} +1 & \text{if } k\|u-c\| \le R \\ -1 & \text{otherwise} \end{cases} .$$



- L. Bertinetto , J. Valmadre , J. F. Henriques , A. Vedaldi and P. H. Torr. “Fully-convolutional siamese networks for object tracking”. In ECCV, 2016.

目标跟踪：分类网络

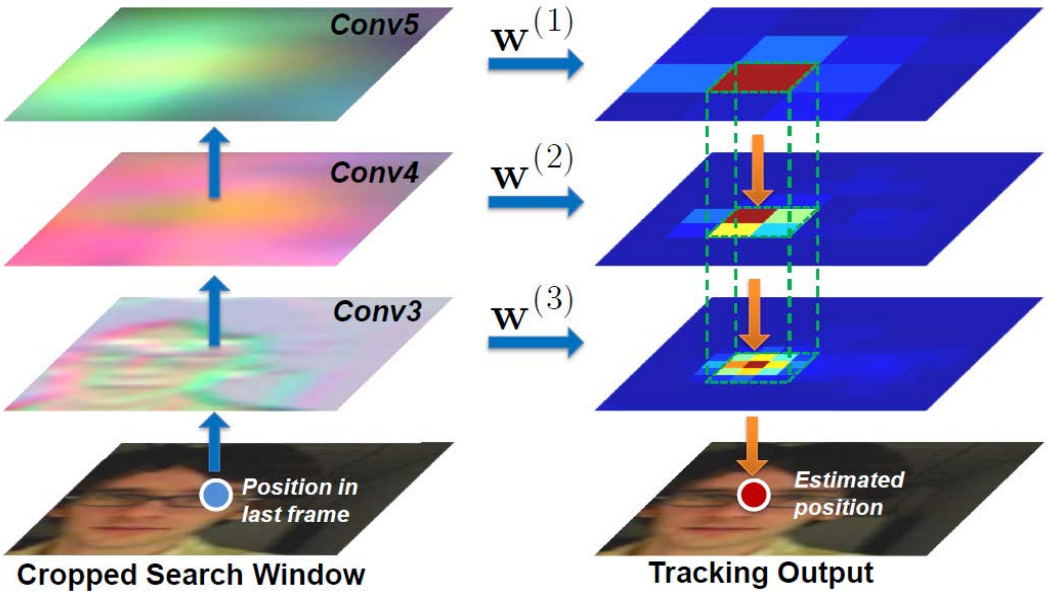
- 方法动机
  - 将跟踪任务视为二分类任务 (目标和背景).
  - 训练: 保持多个分支(multi-domain) 学习共性的特征表达
  - 跟踪: 重新初始化新的分支，并引入在线训练来区分当前视频中的前景和背景。每一帧通过分类大量的候选粒子进行跟踪
- 方法框架结构



- H. Nam and B. Han. “Learning multi-domain convolutional neural networks for visual tracking”. In CVPR, 2016.

目标跟踪：深度特征下的相关滤波器

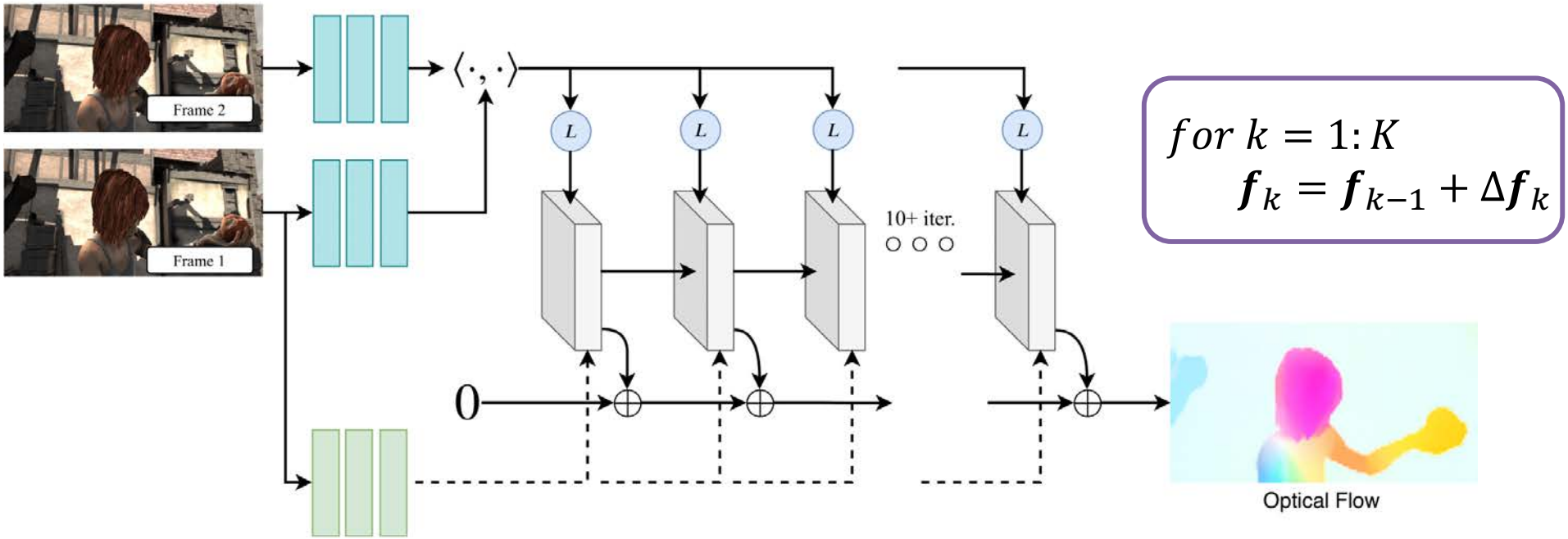
- 方法动机：
  - 采用相关滤波器的框架对目标位置进行回归
  - 同时利用高层语义信息和低层的纹理细节等信息
  - 构造多尺度特征下的相关滤波器并将其结果通过由粗到精进行组合
- 方法框架结构：



- C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. “Hierarchical convolutional features for visual tracking”. In ICCV, 2015.

光流估计：RAFT

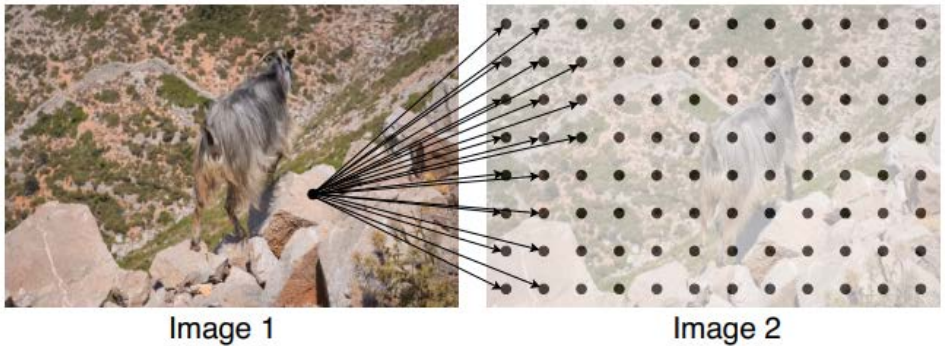
- 方法动机
  - 运用增量学习的思想：引入GRU单元构建更新器，迭代地输出光流残差，渐进地优化输出的光流
- 方法框架结构

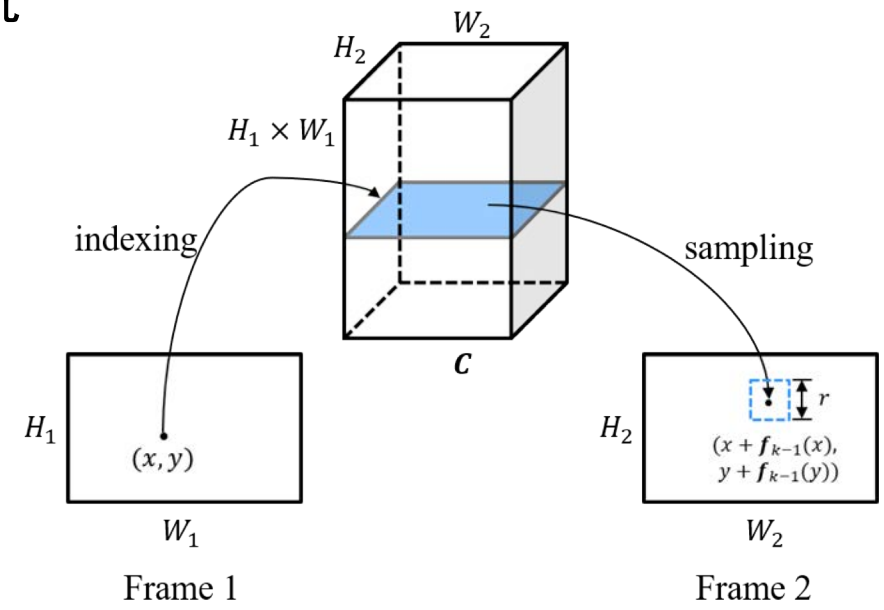


- Zachary Teed and Jia Deng. “RAFT: Recurrent All-Pairs Field Transforms for Optical Flow ”. In ECCV, 2020.

光流估计：RAFT

- 区域相似度特征提取
  - 计算Frame 1和Frame 2的特征相似度矩阵$\boldsymbol{C} \in \mathbb{R}^{H_1 \times W_1 \times H_2 \times W_2}$
  - 第k迭代时：根据第k-1次迭代输出的光流$\boldsymbol{f}_{k-1}$，在$\boldsymbol{C}$上对应位置采样$r \times r$邻域内的特征，输入GRU更新器
  - 该采样特征提供了区域级的相似度信息，以利于更新器捕捉相似区域的位置，优化当前光流



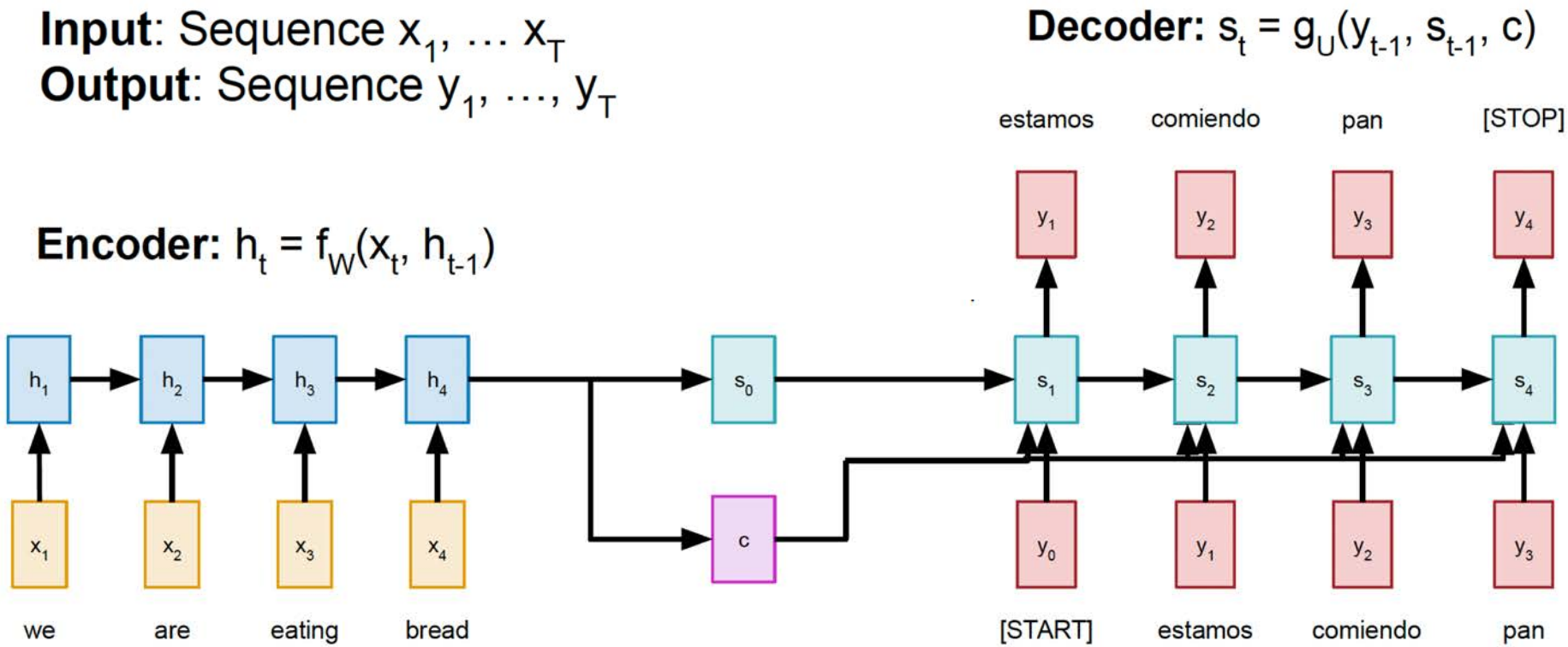


- Zachary Teed and Jia Deng. “RAFT: Recurrent All-Pairs Field Transforms for Optical Flow ”. In ECCV, 2020.

基于深度学习的图像分析技术

- 现代深度学习技术实例
  - 图像识别
  - 目标检测
  - 语义分割
  - 视频理解
  - 目标跟踪
  - 注意力机制及Transformer
  - 生成式模型
  - 自监督学习

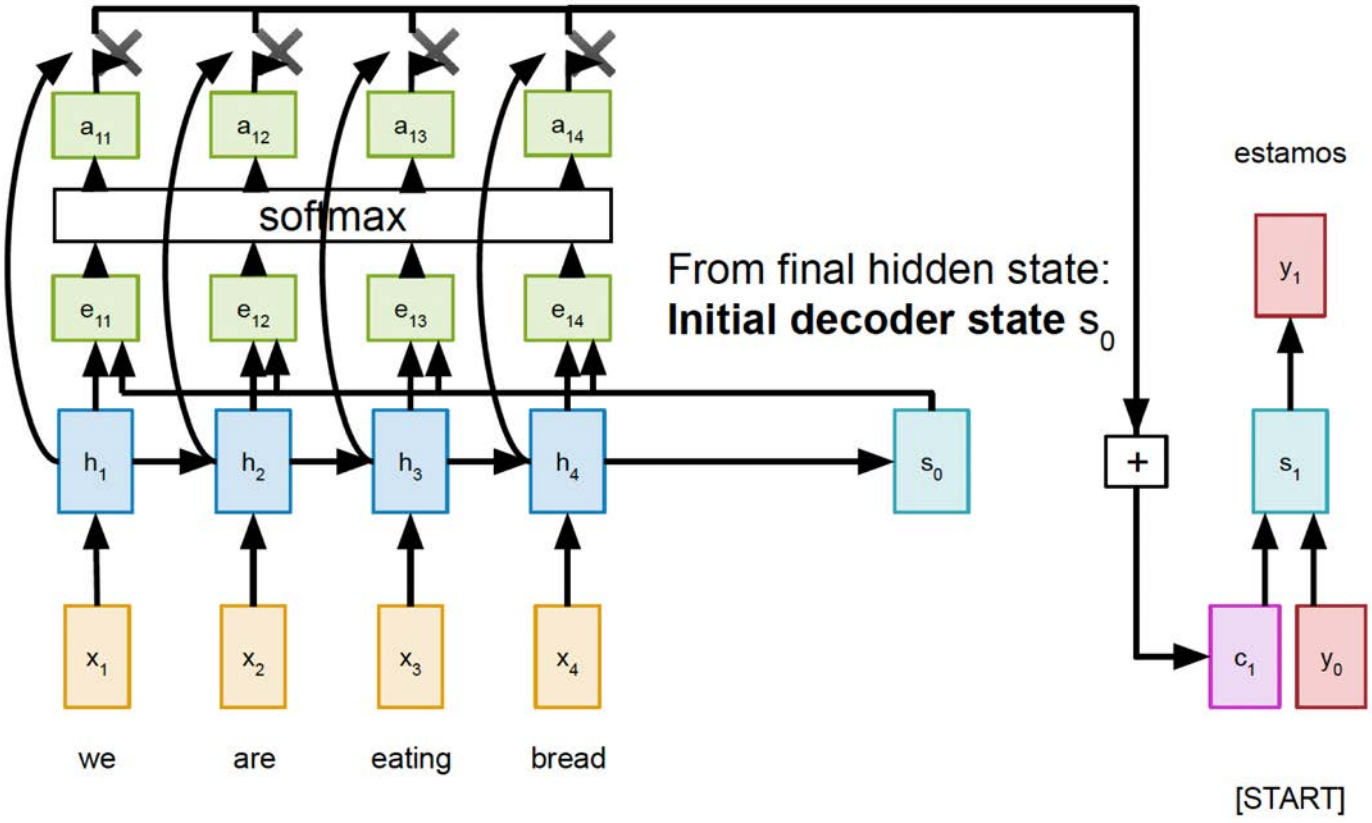
基于RNN的序列到序列模型



问题：输入序列的信息被压缩到单个向量

RNN+注意力

- 引入注意力机制



1. 计算对齐分数 (alignment scores)

$$e_{t,i} = f_{\text{attn}}(s_{i-1}, h_i)$$

2. 归一化得到注意力权值(attention weights)

$$0 < a_{t,i} < 1,$$

$$\sum_i a_{t,i} = 1$$

3. 计算上下文向量

$$c_t = \sum_i a_{t,i} h_i$$

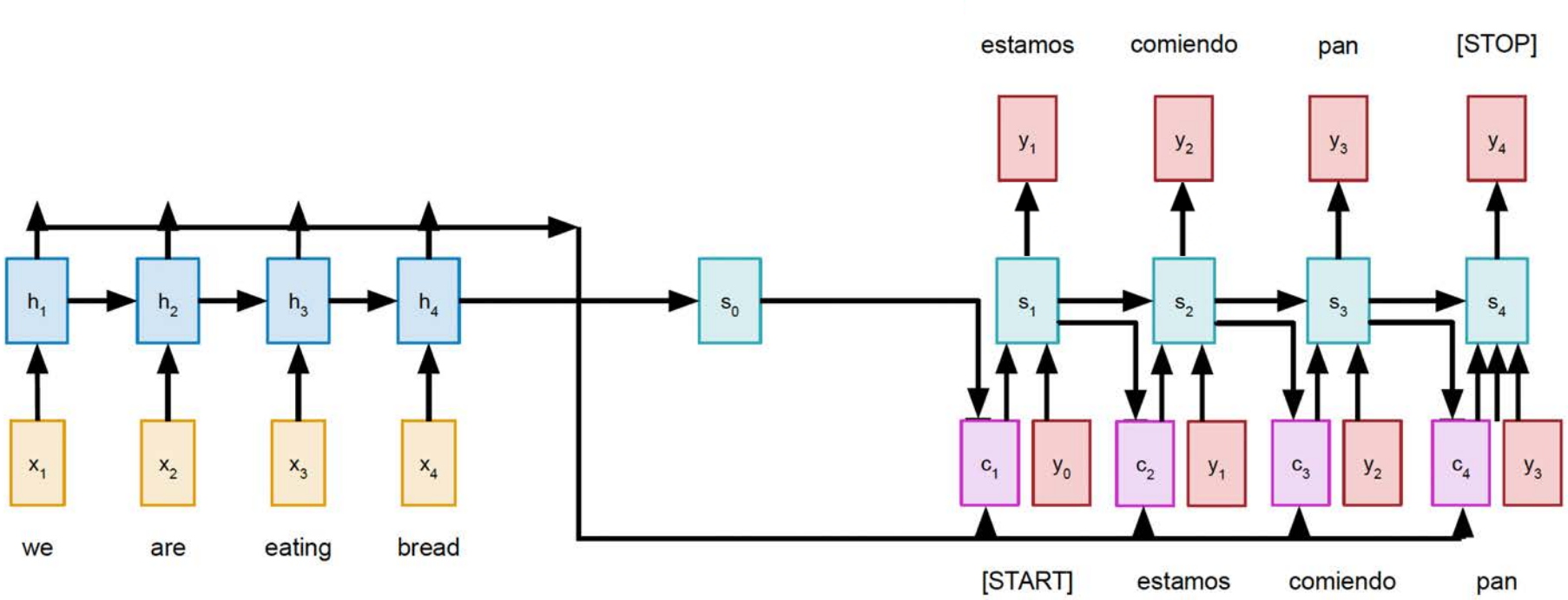
4. 更新s_t

$$s_t = g_U(y_{t-1}, s_{t-1}, c_t)$$

- Dzmitry Bahdanau, et al. “Neural machine translation by jointly learning to align and translate”. In ICLR 2015.

RNN+注意力

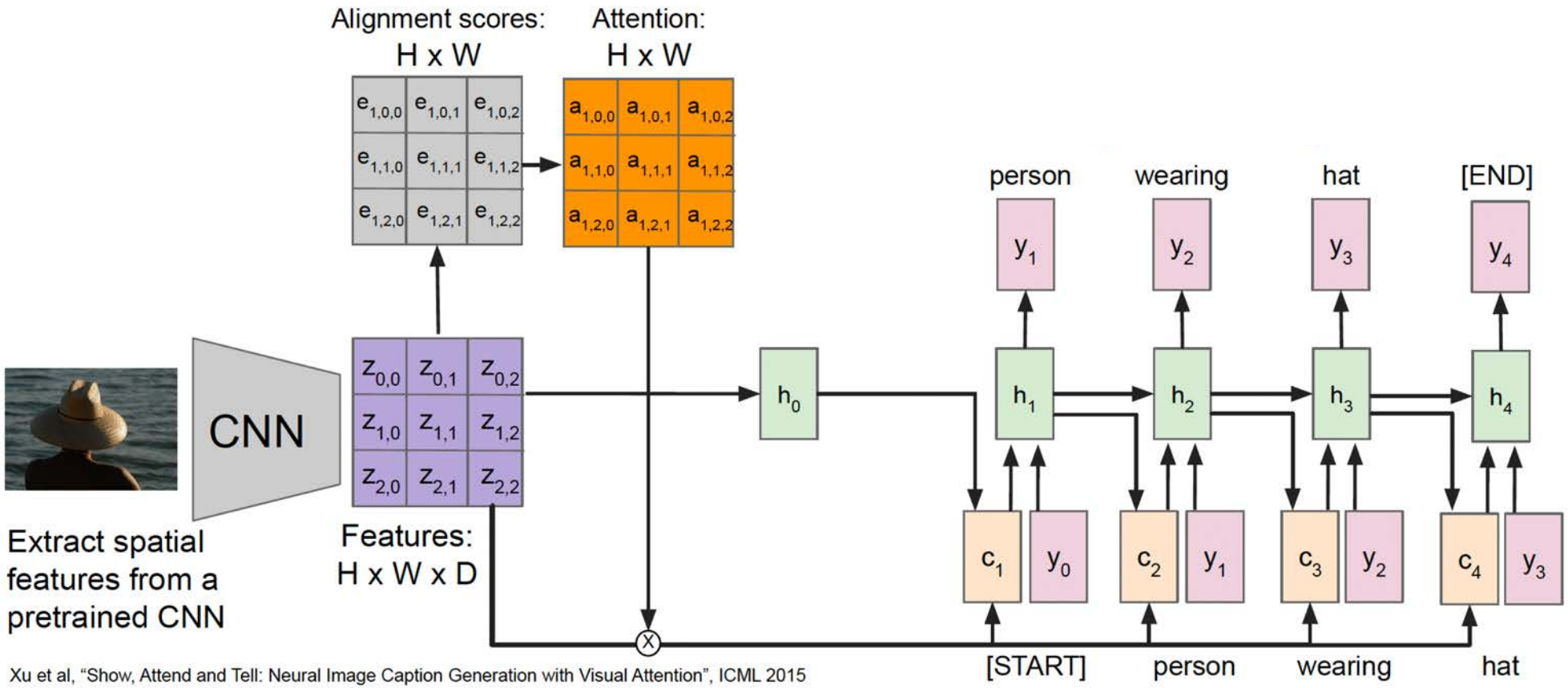
- 引入注意力机制
  - 解码器每一步使用不同的上下文向量
  - 不同时刻关注输入序列的不同部分



- Dzmitry Bahdanau, et al. “Neural machine translation by jointly learning to align and translate”. In ICLR 2015.

RNN+注意力

- Image Captioning



Xu et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In ICML, 2015.

注意力机制

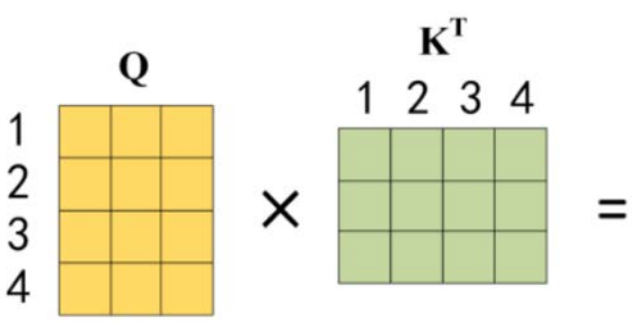
注意力机制

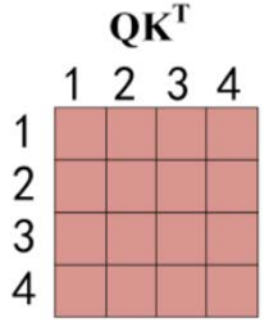
- 注意力模块在计算的时候需要用到矩阵Q(查询), K(键值), V(值)

$$\text{Attention}(Q,K,V)=\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

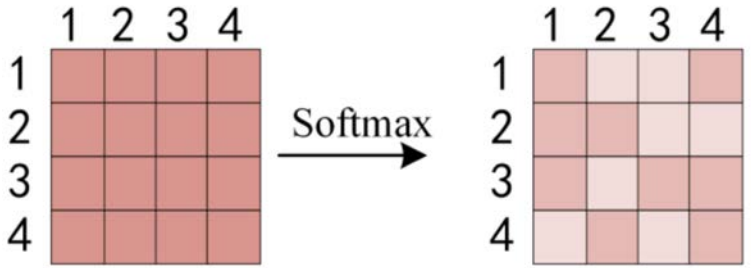
d_k是Q,K矩阵的列数，即向量维度

Q、K 相似性矩阵计算

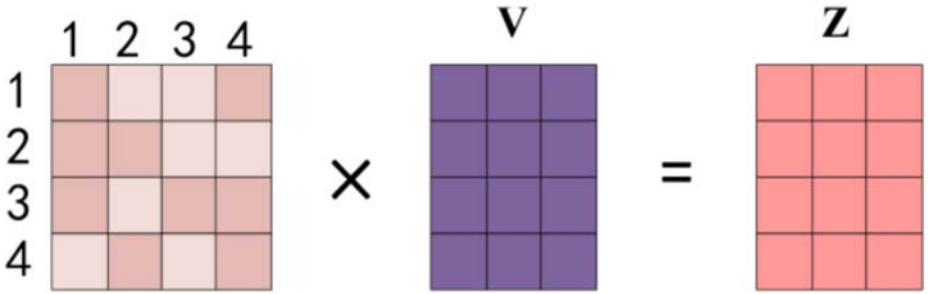




相似性矩阵归一化

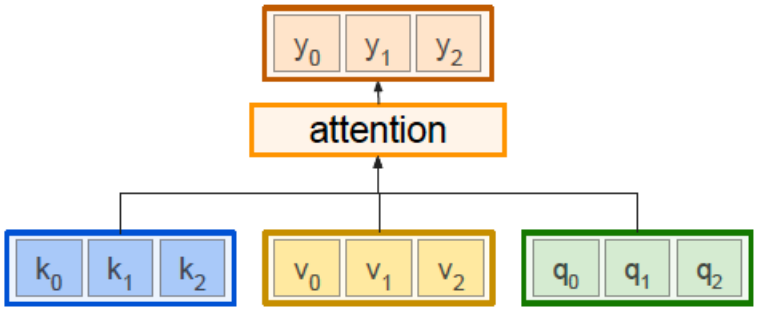


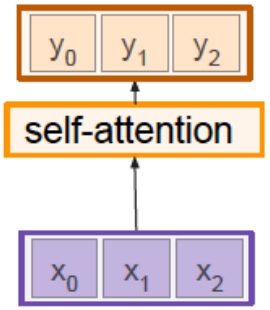
通过相似性矩阵对 V 加权求和



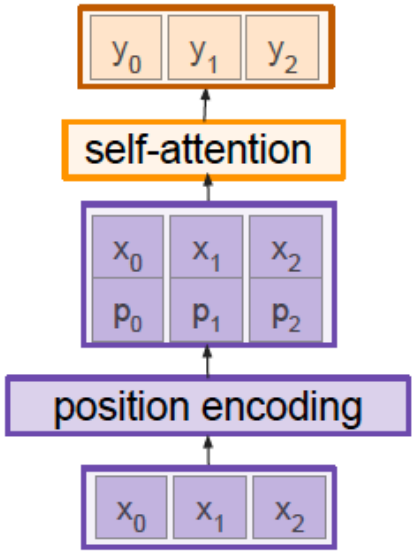
注意力机制

- 自注意力（Self Attention）





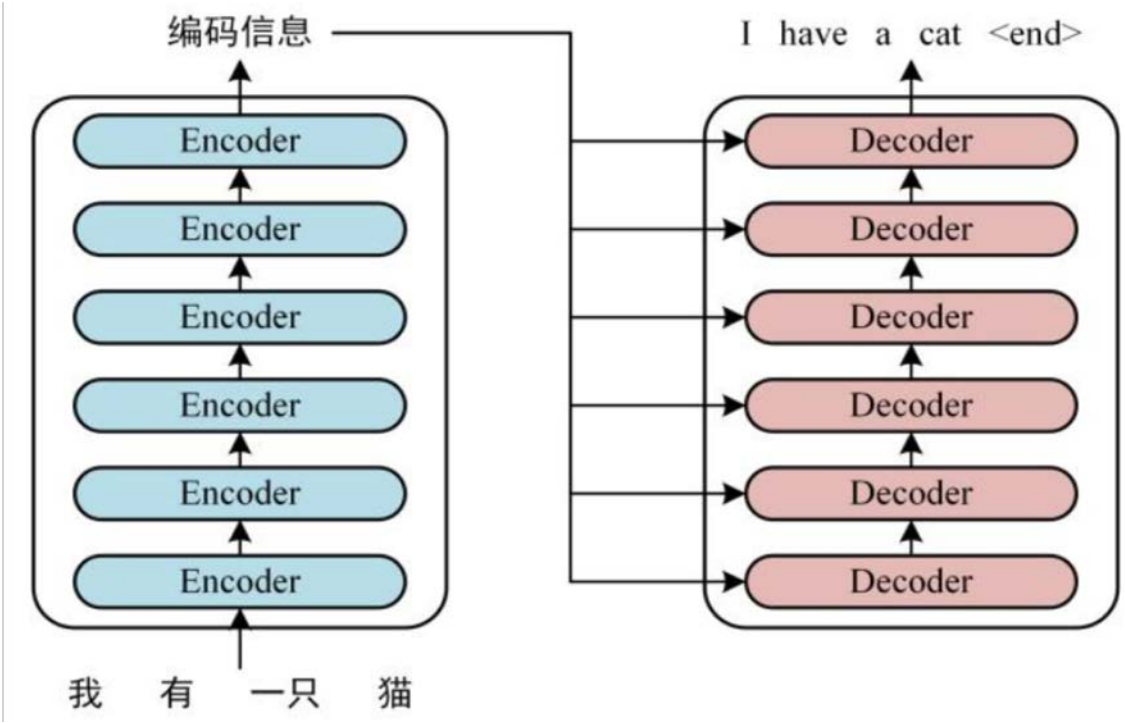
- 位置编码



Transformer

Transformer概况

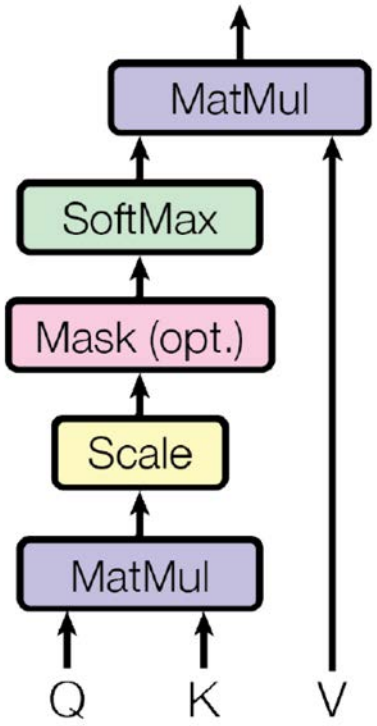
- Transformer是Google团队在2017年提出的一种自然语言处理模型
- Transformer模型使用了注意力机制，使得模型可以并行化训练，而且能够拥有全局信息
- 主要由编码器（Encoder）和解码器（Decoder）组成

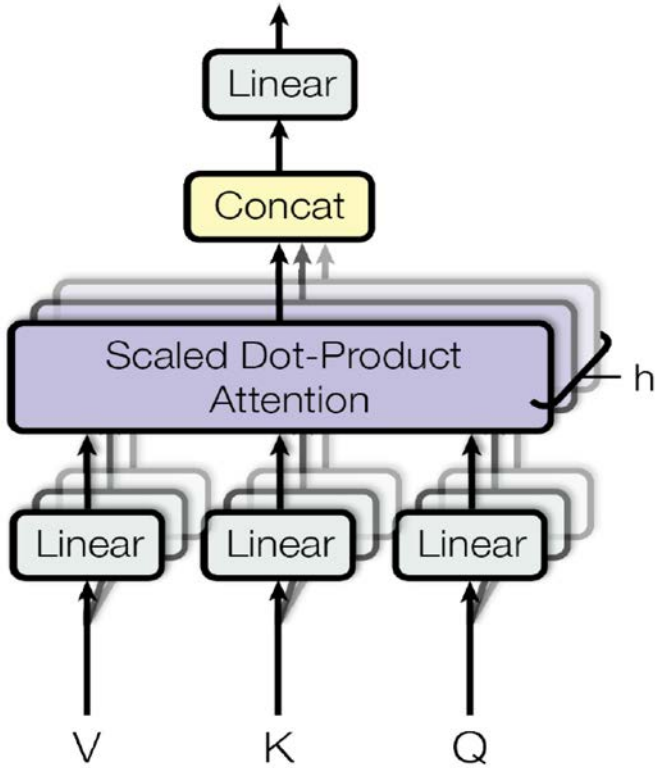


- Ashish Vaswani, et al. “Attention is all you need”. In NeurIPS, 2017.

Transformer

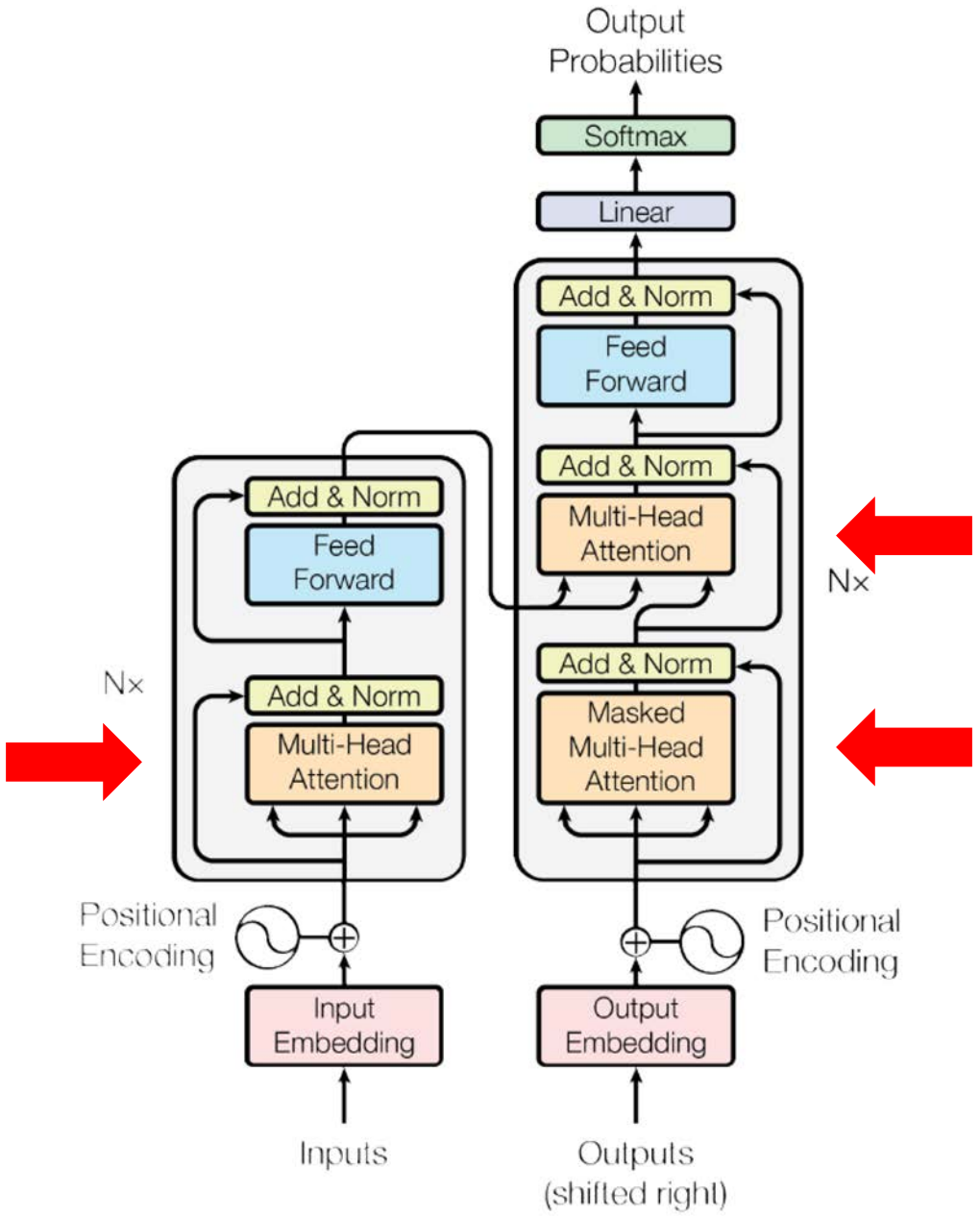
- Transformer概况
  - 注意力模块的基本形式
  - 多头注意力（Multi-head Attention）：Transformer中，通过构造一系列并列的注意力模块，组成多头注意力机制。通过将输入映射到不同的子空间，有助于学习更加丰富的注意力表达。





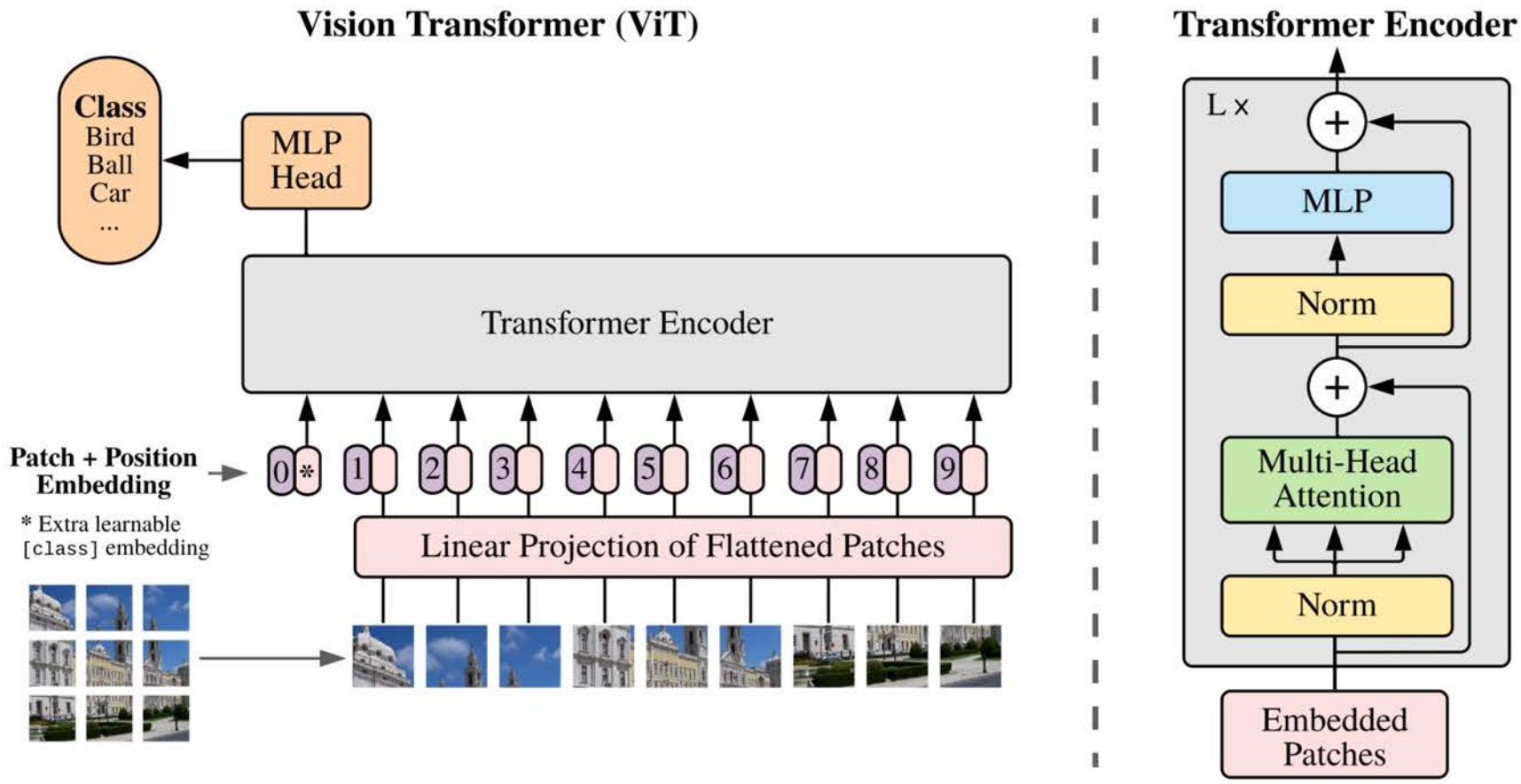
Transformer

- Transformer概况
  - 编码器（encoder）和解码器（decoder）主要由注意力模块组成
  - 编码器和解码器反复堆叠（N=6），以便更好地通过注意力机制获取全局信息



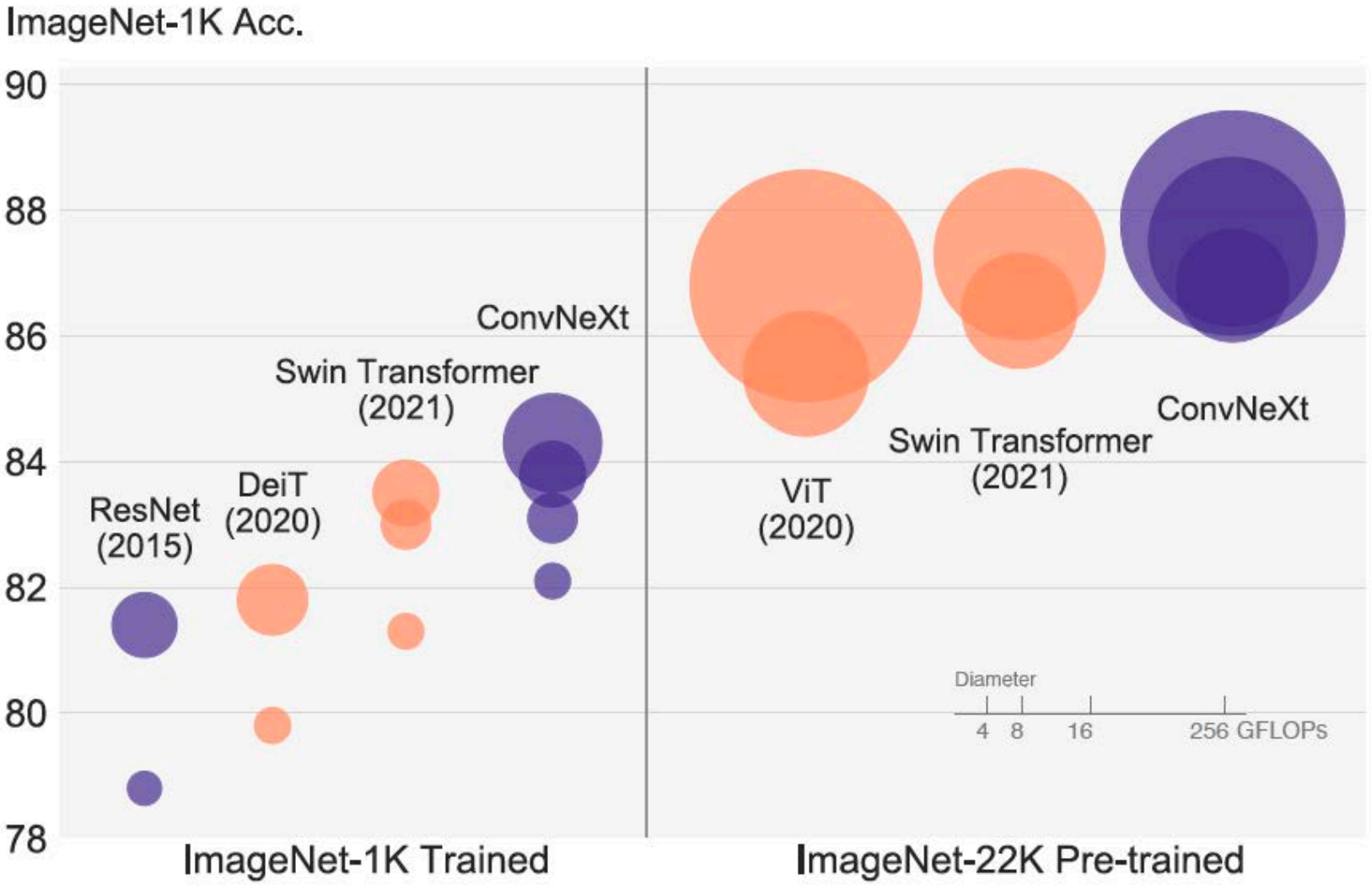
Transformer

视觉Transformer



- Alexey Dosovitskiy, et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In ICLR, 2021.

Transformer VS. CNN



- Z. Liu, et al. “A ConvNet for the 2020s”. In CVPR, 2022.

基于深度学习的图像分析技术

- 现代深度学习技术实例
  - 图像识别
  - 目标检测
  - 语义分割
  - 视频理解
  - 目标跟踪
  - 注意力机制及Transformer
  - 生成式模型
  - 自监督学习

生成式模型（Generative Models）

- 生成式模型
  - 生成式模型概论
  - 自回归生成模型
  - 变分自编码器
  - 扩散模型
  - 生成对抗网络
  - 生成对抗网络的应用

生成式模型的定义

- 通常机器学习的任务就是学习一个模型，应用这一模型，对给定的输入预测相应的输出。对于模型的分类有很多种，其中一种分类是把模型分为：判别模型和生成模型两种。

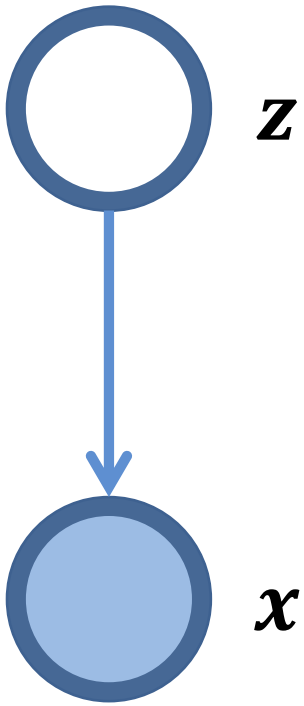
- 判别模型主要是根据输入数据推测数据具备的一些性质，即：已知观察变量x和隐变量z，直接对$p(z|x)$进行建模。它根据输入的观察变量x直接得到隐变量z出现的可能性。
  - 例如：当模型为分类模型时，隐变量z则代表类别变量。

- 生成模型则是对$p(x,z)$进行建模，然后求出条件概率分布$p(z|x)$作为预测隐变量z的模型，即：

$$p(z|x) = \frac{p(x,z)}{p(x)} = \frac{p(x|z)p(z)}{p(x)}$$

同时，我们也可以根据联合概率分布$p(x,z)$采样生成观测变量x。

生成式模型的定义

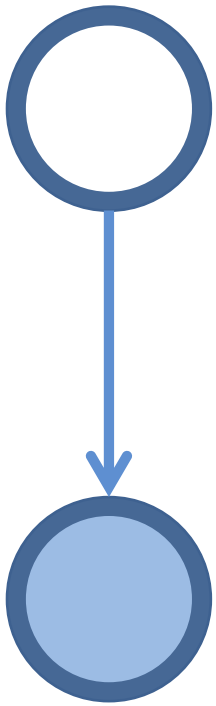


- 通常，我们已知观测变量x服从某固定但未知的分布，与隐变量z构成概率有向图。
- 对于这个概率图，$p(z)$（隐变量的先验）、$p(x|z)$（x相对z的条件概率）及$p(z|x)$（隐变量的后验）三者就可完全描述x和z的关系。两者的联合分布可以表示为：

$$p(x,z) = p(x|z)p(z)$$

- 我们只能观测到x，而z是隐变量，不能被观测。生成模型建模便是通过一个观察集$\mathcal{X}$，估计观测变量x与隐变量z构成的概率图的相关参数。

生成式模型的定义



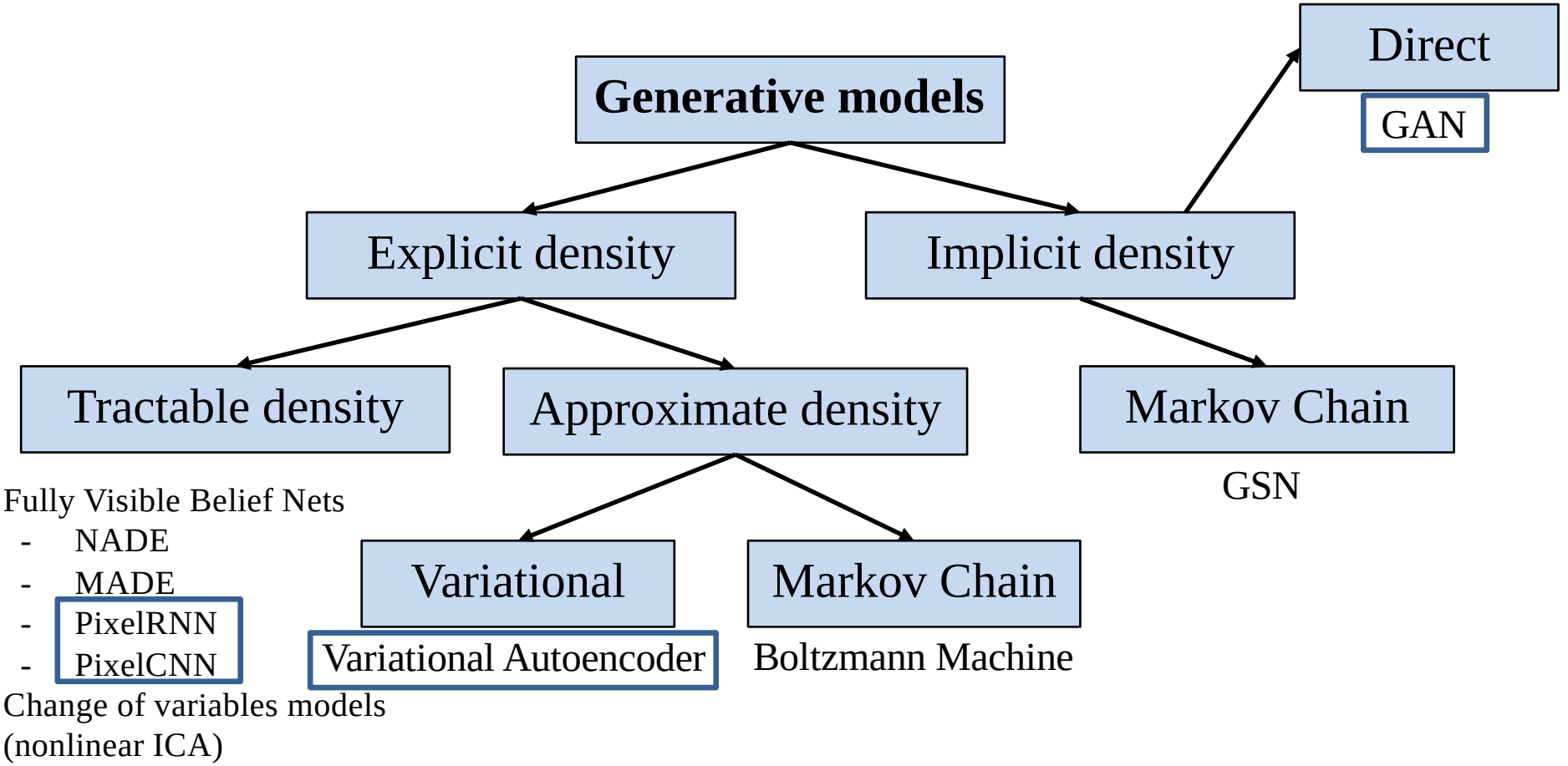
- 对于一个模型，如果它能够建模$p(z)$、$p(x|z)$，我们就称之为生成模型，这有如下两层含义：

I. $p(z)$、$p(x|z)$两者决定了观测变量x与隐变量z的联合分布$p(x,z)$。

II. 利用两者关系可以对观测变量x进行采样。具体做法是：先依隐变量z的先验概率生成样本点$z_i \sim p(z)$，再依观测变量x的条件概率采样$x_i \sim p(x|z)$。

生成式模型分类

- 根据对概率密度函数的表达，生成式模型可以分为两类，显式表达和隐式表达概率密度函数的生成式模型。
- 显式表达概率密度函数的生成式模型又可分为近似概率密度函数的生成式模型和解析概率密度函数的生成式模型。



生成式模型（Generative Models）

- 生成式模型
  - 生成式模型概论
  - 自回归生成模型
  - 变分自编码器
  - 扩散模型
  - 生成对抗网络
  - 生成对抗网络的应用

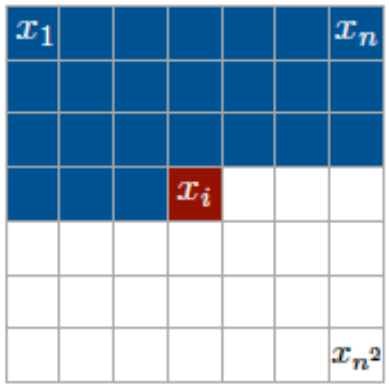
自回归生成模型

- PixelRNN和PixelCNN
  - 建模 $p(\mathrm{x})$

$$p(\mathbf{x}) = \prod_{i=1}^{n^2} p(x_i | x_1, \ldots, x_{i-1})$$

Likelihood of image x

Probability of the i-th pixel value given all previous pixels

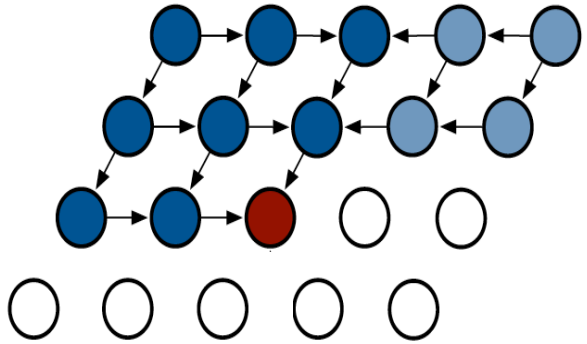


  - 利用训练数据进行最大似然估计

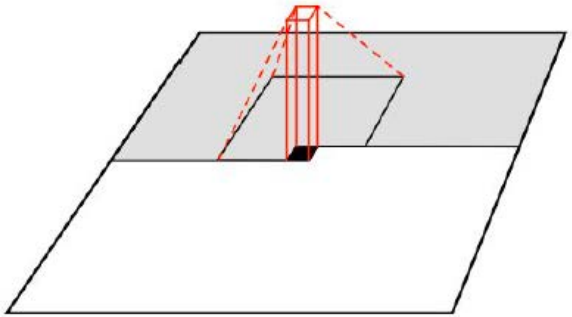
- A. van den Oord, N. Kalchbrenner and K. Kavukcuoglu. “Pixel Recurrent Neural Networks”. In ICML, 2016.

自回归生成模型

- PixelRNN
  - 利用RNN (BiLSTM)建模像素间的依赖关系
  - 缺点：训练、生成速度慢

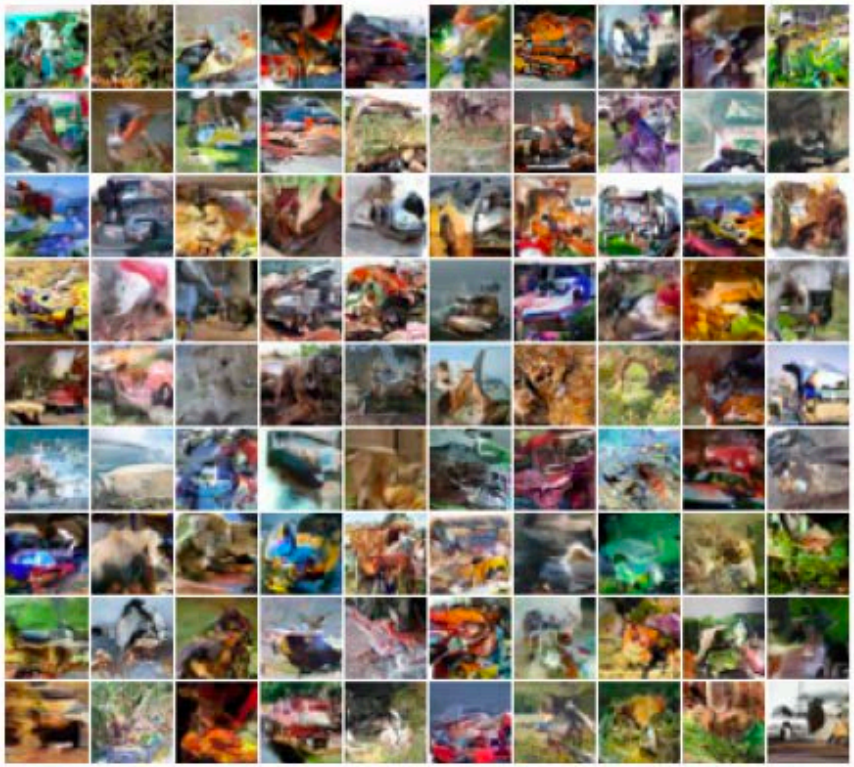


- PixelCNN
  - 利用CNN (masked conv)建模像素间的依赖关系
  - 训练速度比PixelRNN快，生成速度依旧慢

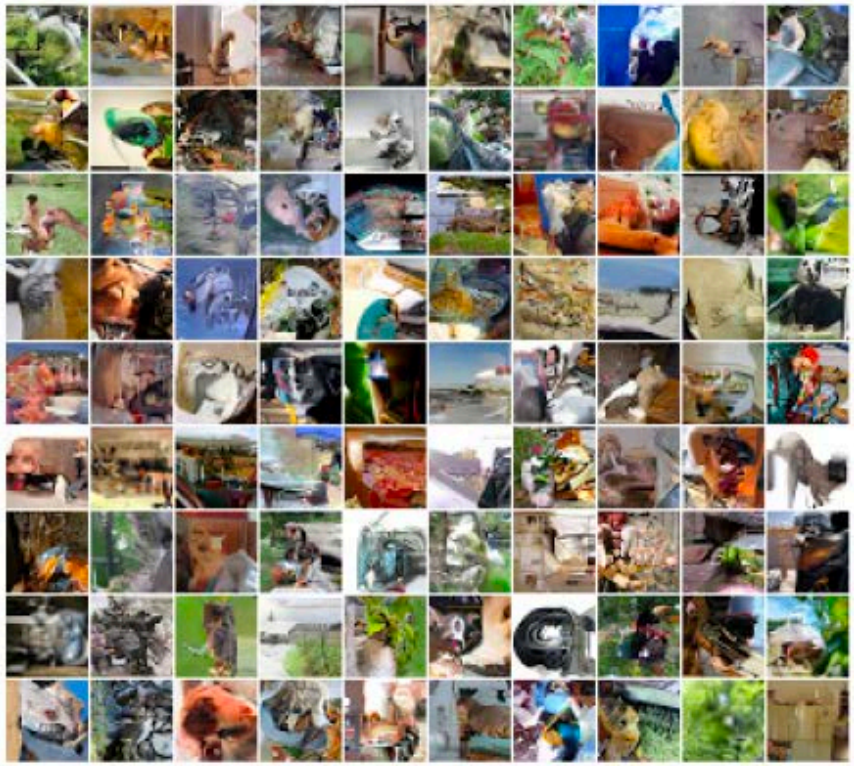


自回归生成模型

- 生成结果



32×32 CIFAR-10



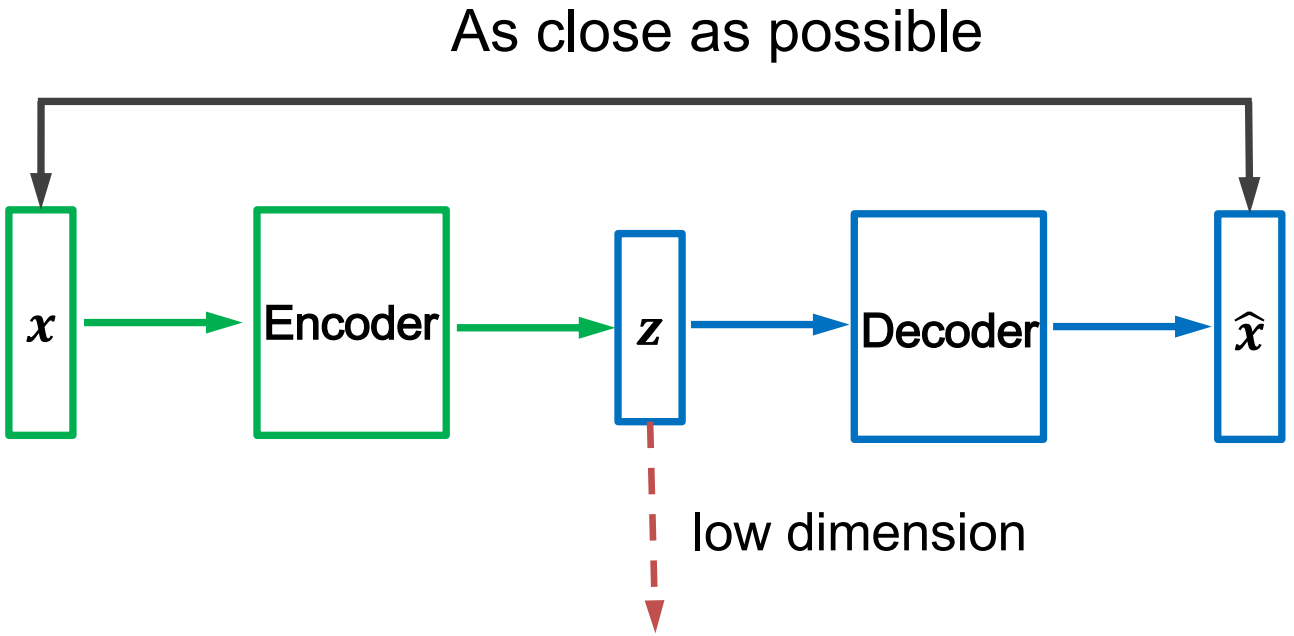
32×32 Imagenet

生成式模型（Generative Models）

- 生成式模型
  - 生成式模型概论
  - 自回归生成模型
  - 变分自编码器
  - 扩散模型
  - 生成对抗网络
  - 生成对抗网络的应用

自编码器(Auto-Encoder)

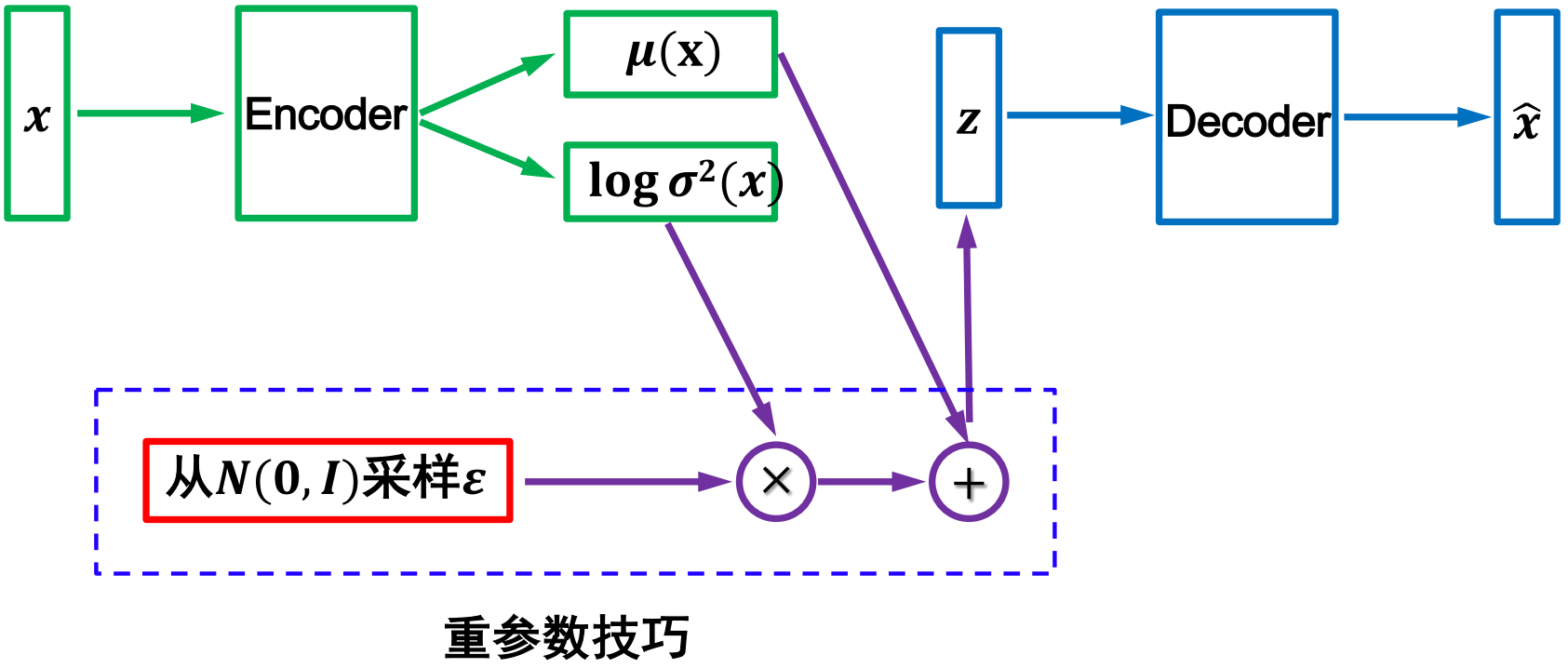
自编码器(Auto-Encoder)



- Embedding, feature, code
- Used as feature for downstream tasks
- Cannot generate new sample

变分自编码器（VAE）

- VAE的整体模型结构
  - 包含:编码器部分、解码器部分以及重参数化采样



- Diederik P. Kingma, and Max Welling. “Auto-Encoding Variational Bayes”. In arXiv:1312.6114, 2013.

变分自编码器（VAE）

- 首先我们有一批观测样本$\{x_1, \dots, x_n\}$，其整体用x来描述。借助隐变量z，观测样本与隐变量的联合分布为：$p(x,z)$，由于我们手头上只有x的样本，因此上式可以改写为：

$$p(x,z) = \tilde{p}(x)p(z|x) \tag{1}$$

  - 注意这里的$\tilde{p}(x)$是根据样本$x_1, x_2, \dots, x_n$确定的关于x的先验分布。尽管我们无法准确写出它的形式，但它是确定的、存在的。

- 接下来，直接对$p(x,z)$进行近似。具体来说，我们设想用一个新的联合概率分布$q(x,z)$来逼近$p(x,z)$，那么我们可以用KL散度来计算它们的距离：

$$KL(p(x,z)||q(x,z)) = \iint p(x,z)\log\frac{p(x,z)}{q(x,z)}\,dzdx \tag{2}$$

- KL散度是我们的最终目标，因为我们希望两个分布越接近越好，所以KL散度越小越好。将(1)式代入(2)式，我们有：

变分自编码器（VAE）

$$KL(p(x,z)||q(x,z)) = \int \tilde{p}(x)[\int p(z|x)\log\frac{\tilde{p}(x)p(z|x)}{q(x,z)}\,dz]\,dx$$

$$= \mathbb{E}_{x\sim\tilde{p}(x)}[\int p(z|x)\log\frac{\tilde{p}(x)p(z|x)}{q(x,z)}\,dz] \qquad (3)$$

(3)式还可以进一步简化：

$$KL(p(x,z)||q(x,z)) = \mathbb{E}_{x\sim\tilde{p}(x)}[\int p(z|x)\log\tilde{p}(x)\,dz] + \mathbb{E}_{x\sim\tilde{p}(x)}[\int p(z|x)\log\frac{p(z|x)}{q(x,z)}\,dz)]$$

$$= \mathbb{E}_{x\sim\tilde{p}(x)}[\log\tilde{p}(x)\underbrace{\int p(z|x)dz}_{1}] + \mathbb{E}_{x\sim\tilde{p}(x)}[\int p(z|x)\log\frac{p(z|x)}{q(x,z)}\,dz)]$$

$$= \underbrace{\mathbb{E}_{x\sim\tilde{p}(x)}[\log\tilde{p}(x)]}_{\text{常量}C} + \mathbb{E}_{x\sim\tilde{p}(x)}[\int p(z|x)\log\frac{p(z|x)}{q(x,z)}\,dz)] \qquad (4)$$

变分自编码器（VAE）

- 通过移项，我们可以令：

$$\mathcal{L} = KL(p(x,z)||q(x,z)) - C = \mathbb{E}_{x\sim\tilde{p}(x)}[\int p(z|x)\log\frac{p(z|x)}{q(x,z)}dz)]$$

- 最小化KL散度也就等价于最小化$\mathcal{L}$。为了得到生成模型，我们把$q(x,z)$写成$q(x|z)q(z)$，于是有：

$$\mathcal{L} = \mathbb{E}_{x\sim\tilde{p}(x)}[\int p(z|x)\log\frac{p(z|x)}{q(x|z)q(z)}dz)]$$

$$= \mathbb{E}_{x\sim\tilde{p}(x)}[-\int p(z|x)\log q(x|z)dz + \int p(z|x)\log\frac{p(z|x)}{q(z)}dz]$$

$$= \mathbb{E}_{x\sim\tilde{p}(x)}[\mathbb{E}_{z\sim p(z|x)}[-\log q(x|z)] + KL(p(z|x)||q(z))] \quad (5)$$

优化目标

变分自编码器（VAE）

$$\mathcal{L} = \mathbb{E}_{x\sim\tilde{p}(x)}[\mathbb{E}_{z\sim p(z|x)}[-\log q(x|z)] + KL(p(z|x)||q(z))]$$

- 得到了优化目标，我们就要想办法找到合适的$q(x|z)$和$q(z)$使得$\mathcal{L}$最小化。首先，为了方便采样，我们假设$z\sim N(0,I)$，即标准的多元正态分布，这就解决了$q(z)$。

- 然后，$p(z|x)$也是(各分量独立的)正态分布，其均值与方差由x来决定，这个“决定”，就是一个神经网络的拟合：

$$p(z|x) = \frac{1}{\prod_{k=1}^{d}\sqrt{2\pi\sigma_{(k)}^2(x)}}\exp\left(-\frac{1}{2}\left\|\frac{z-\mu(x)}{\sigma(x)}\right\|^2\right)$$

- 其中，x是神经网络的输入，输出则是均值$\mu(x)$与方差$\sigma^2(x)$。这里的神经网络就起到了类似Encoder的作用。$\mathcal{L}$中的KL散度这一项可以先算出来：

$$KL(p(z|x)||q(z)) = \frac{1}{2}\sum_{k=1}^{d}(\mu_{(k)}^2(x) + \sigma_{(k)}^2(x) - \log\sigma_{(k)}^2(x) - 1)$$

变分自编码器（VAE）

$$\mathcal{L} = \mathbb{E}_{x\sim\tilde{p}(x)}[\mathbb{E}_{z\sim p(z|x)}[-\log q(x|z)] + KL(p(z|x)||q(z))]$$

- 现在只剩下生成模型部分$q(x|z)$了，对于分布的选择，原论文给出了两种候选方案：伯努利分布或正态分布。
  - 伯努利分布其实就是一个二值分布：

$$p(\varepsilon) = \begin{cases} \rho, & \varepsilon = 1 \\ 1-\rho, & \varepsilon = 0 \end{cases}$$

- 所以伯努利分布只适用于 x 是一个多元的二值向量的情况，比如：MNIST。这种情况下，我们用神经网络$\rho(z)$来算参数ρ，从而得到：

$$q(x|z) = \prod_{k=1}^{D} \left(\rho_{(k)}(z)\right)^{x_{(k)}} \left(1-\rho_{(k)}(z)\right)^{1-x_{(k)}}$$

- 这时可以算出：

$$-\log q(x|z) = \sum_{k=1}^{D} \left[-x_{(k)} \log \rho_{(k)}(z) - \left(1-x_{(k)}\right) \log\left(1-\rho_{(k)}(z)\right)\right]$$ 交叉熵

这里$\rho(z)$就起到了类似Decoder的作用。

变分自编码器（VAE）

$$\mathcal{L} = \mathbb{E}_{x\sim\tilde{p}(x)}[\mathbb{E}_{z\sim p(z|x)}[-\log q(x|z)] + KL(p(z|x)||q(z))]$$

- 然后是正态分布，与$p(z|x)$很像，只是x，z交换了位置：

$$q(x|z) = \frac{1}{\prod_{k=1}^{D}\sqrt{2\pi\sigma_{(k)}^2(z)}}\exp\left(-\frac{1}{2}\left\|\frac{x-\mu(z)}{\sigma(z)}\right\|^2\right)$$

- 这里，神经网络的输入是z，输出是$\mu(z)$与$\sigma^2(z)$。于是：

$$-\log q(x|z) = \frac{1}{2}\left\|\frac{x-\mu(z)}{\sigma(z)}\right\|^2 + \frac{D}{2}\log 2\pi + \frac{1}{2}\sum_{k=1}^{D}\log\sigma_{(k)}^2(z)$$

- 通常情况下，我们会固定方差为一个常数σ^2，这时候有：

$$-\log q(x|z) \sim \frac{1}{2\sigma^2}\|x-\mu(z)\|^2$$

- 于是，这就变成了我们熟悉的MSE损失函数！$\mu(z)$就起到了Decoder的作用。

变分自编码器（VAE）

- 现在，让我们看回VAE的优化目标：

$$\mathcal{L} = \mathbb{E}_{x\sim\tilde{p}(x)}[\mathbb{E}_{z\sim p(z|x)}[-\log q(x|z)] + KL(p(z|x)||q(z))]$$

  - 对于等号右侧的第二项， $p(z|x)$起到Encoder的作用，同时KL散度将Encoder输出的隐向量约束为标准的多元正态分布；

  - 对于等号右侧的第一项， $q(x|z)$起到Decoder的作用。对于二值数据(例如：MNIST)，我们可以对Decoder用sigmoid函数激活，然后用交叉熵作为损失函数，这对应于$q(x|z)$为伯努利分布；而对于一般数据，我们用MSE作为损失函数，这对应于$q(x|z)$为固定方差的多元正态分布；

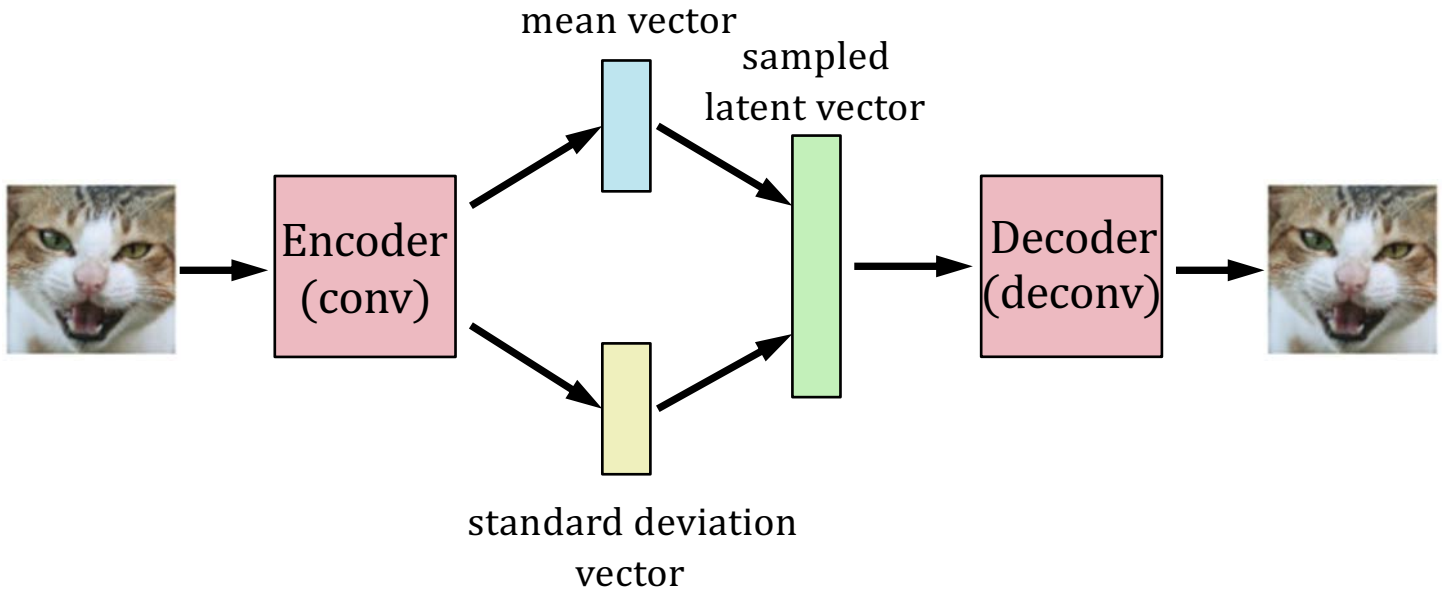
  - 待训练完成后，Decoder就是我们的生成模型。生成过程就是从标准多元正态分布中采样得到隐变量z，再输入Decoder，就可以得到生成样本了。

变分自编码器（VAE）

- VAE的本质是什么？
  - 抛开极大似然估计的思想，只看训练过程。VAE首先通过Encoder将观测样本x编码（映射）为隐空间中的隐变量z，然后再试图通过MSE损失函数（对于二值图像则为交叉熵损失函数）将隐变量z重构回观测样本x。而这样的训练过程与普通的Auto-Encoder是非常相似的，唯一的不同在于VAE多了作用于隐变量的KL散度约束。
  - 再看Auto-Encoder，一个训练好的Auto-Encoder，如果能够在它训练得到的隐空间采样，然后作为Decoder的输入，我们就得到了一个生成模型。但是普通Auto-Encoder的隐空间的分布是复杂且未知的，我们无法进行采样。VAE则对隐空间施加了KL散度约束，使其逼近简单的多元标准正态分布，这就使隐空间的采样变得可能，进而得到我们的生成模型。
  - 所以，从训练过程来看，VAE就是隐空间受约束的Auto-Encoder。

变分自编码器的优缺点

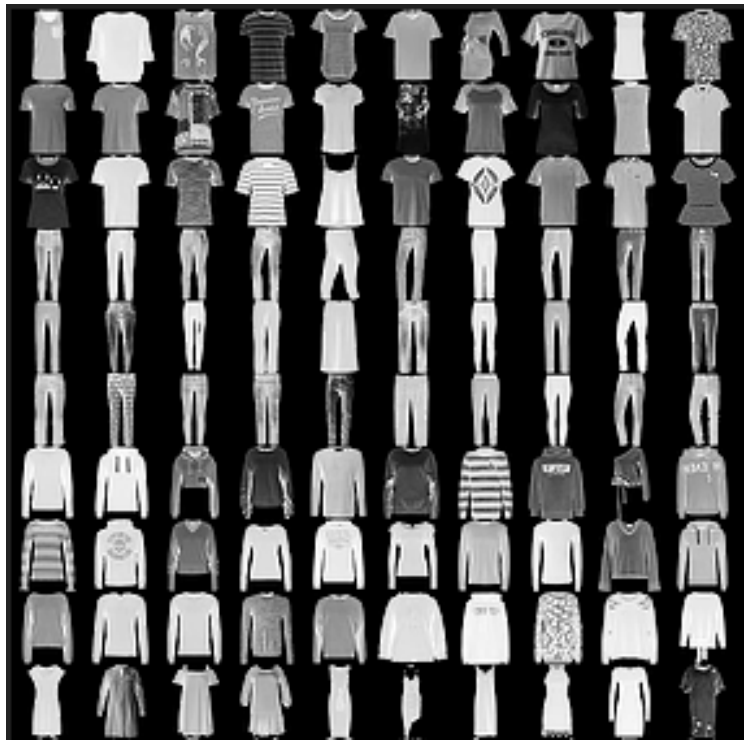
- VAE的优点：
  - VAE的训练过程比较稳定，Loss函数的数值会呈现一个相对稳定的下降趋势；
  - 在训练完成后，VAE的编码器$Encoder$与解码器$Decoder$都可以近似看作理想的高维单映射函数（不同的输入得到不同的输出）。这样，只要隐变量的采样足够多样，就可以保证生成样本的多样性。



变分自编码器的优缺点

- VAE的缺点：

  - VAE所生成的图片会比较模糊。



训练图片

VAE随机生成

生成式模型（Generative Models）

- 生成式模型
  - 生成式模型概论
  - 自回归生成模型
  - 变分自编码器
  - 扩散模型
  - 生成对抗网络
  - 生成对抗网络的应用

扩散模型

Imagen
by Google

Sprouts in the shape of text 'Imagen' coming out of a fairytale book.

Stable diffusion

DALL·E
by OpenAI

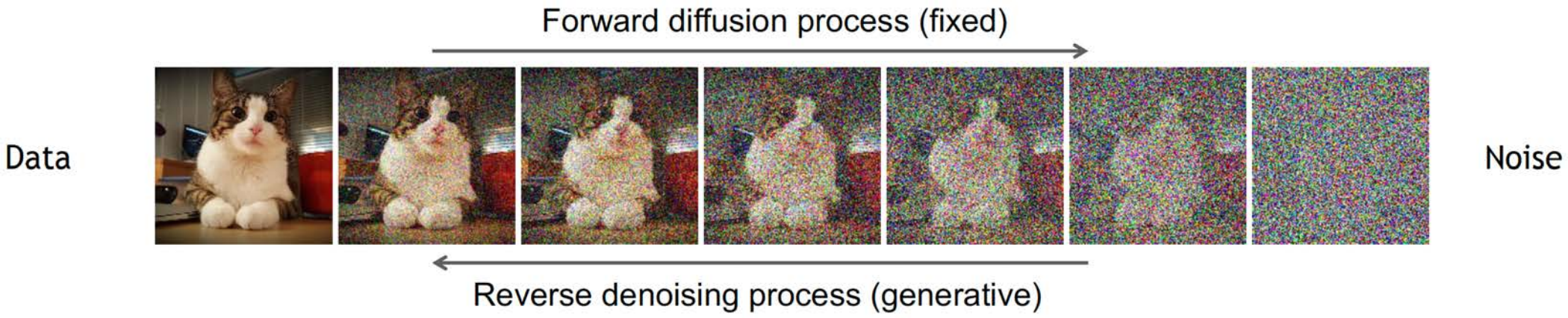
An astronaut riding a horse in photorealistic style.

Midjourney

a cute fluffy bunny grumpily working on her trip itinerary.

扩散模型

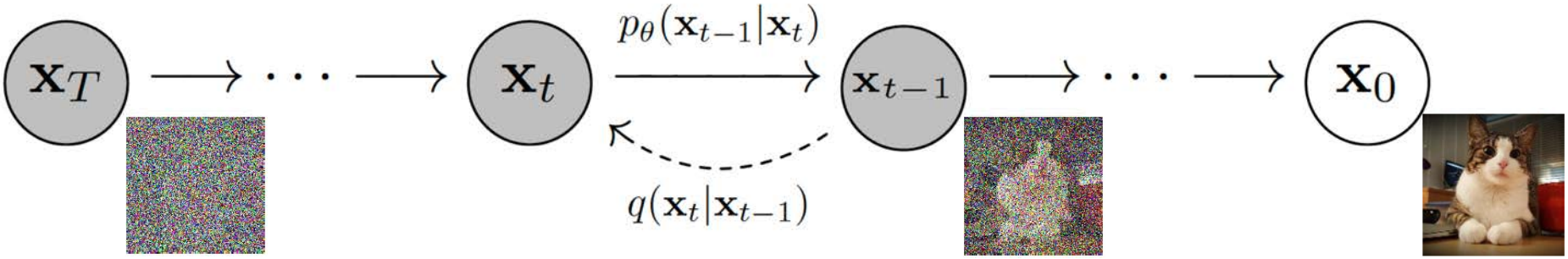
- 去噪扩散概率模型 (Denoising Diffusion Probabilistic Models)
  - 前向扩散过程：逐步添加噪声
  - 逆向去噪过程：通过逐步去噪实现生成



https://cvpr2022-tutorial-diffusion-models.github.io/

扩散模型

□ 去噪扩散概率模型 (DDPM)



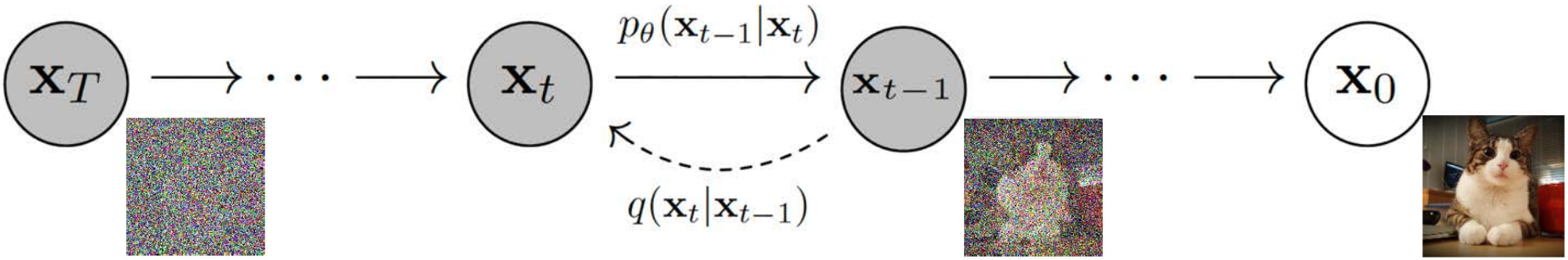
$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

扩散模型

去噪扩散概率模型 (DDPM)

前向扩散过程



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

重参数技巧 $\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\mathbf{z}_{t-1} \quad where\ \mathbf{z}_{t-1} \in \mathcal{N}(0, \mathbf{I})$

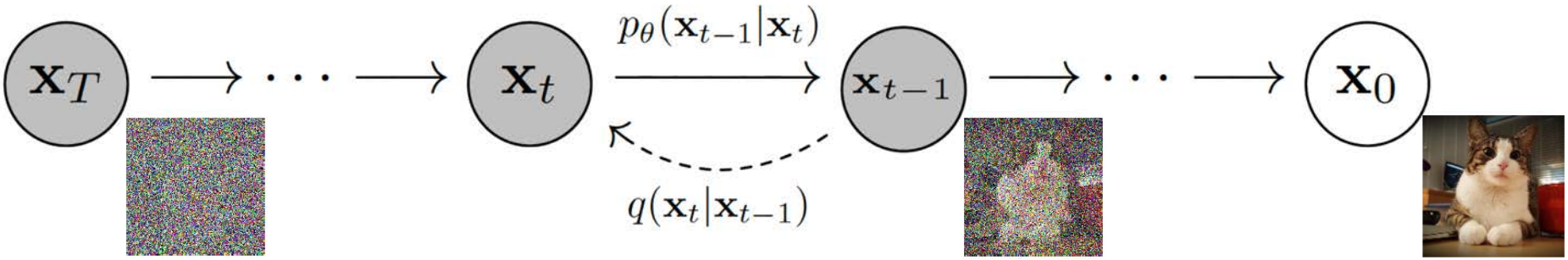
设 $\alpha_t = 1 - \beta_t,\ \bar{\alpha}_t = \prod_{i=1}^{T} \alpha_i,$

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t(1-\alpha_{t-1})}\mathbf{z}_{t-2} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1} \\
&= \dots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z} \qquad \Rightarrow \quad q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})
\end{aligned}$$

扩散模型

去噪扩散概率模型 (DDPM)

逆向去噪过程



$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I})$$

扩散模型

去噪扩散概率模型 (DDPM)

逆向去噪过程

优化目标

$$\mathbb{E}_{q(\mathbf{x}_0)}\left[-\log p_\theta(\mathbf{x}_0)\right] \le \mathbb{E}_{q(\mathbf{x}_0)q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] =: L$$

经推导有

$$L = \mathbb{E}_q\left[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T))}_{L_T} + \sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\right]$$

$$\begin{aligned}
q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) &= q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \\
&= q(\mathbf{x}_t|\mathbf{x}_{t-1})\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \\
&\propto \exp\Big(-\frac{1}{2}\big(\frac{(\mathbf{x}_t-\sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t} + \frac{(\mathbf{x}_{t-1}-\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t-\sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\big)\Big) \\
&= \exp\Big(-\frac{1}{2}\big((\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}})\mathbf{x}_{t-1}^2 - (\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)\mathbf{x}_{t-1} + C(\mathbf{x}_t,\mathbf{x}_0)\big)\Big)
\end{aligned}$$

$$\Rightarrow \quad q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1};\tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0),\tilde{\beta}_t\mathbf{I})$$

$$\tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \qquad \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

扩散模型

去噪扩散概率模型 (DDPM)

逆向去噪过程

$$L = \mathbb{E}_q \left[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \right]$$

$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$ 和 $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ 均为正态分布，其KL散度为

$$L_{t-1} = D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} ||\tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0) - \mu_\theta(\mathbf{x}_t,t)||^2 \right] + C$$

$$\left.\begin{aligned} \tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0) &:= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \\ \mathbf{x}_t &= \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\,\epsilon \qquad \epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I}) \end{aligned}\right\} \Rightarrow \tilde{\mu}_t(\mathbf{x}_t,\mathbf{x}_0) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right)$$

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

则 $L_{t-1} = \mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0),\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\frac{\beta_t^2}{2\sigma_t^2(1-\beta_t)(1-\bar{\alpha}_t)}||\epsilon - \epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon}_{\mathbf{x}_t}, t)||^2\right] + C$

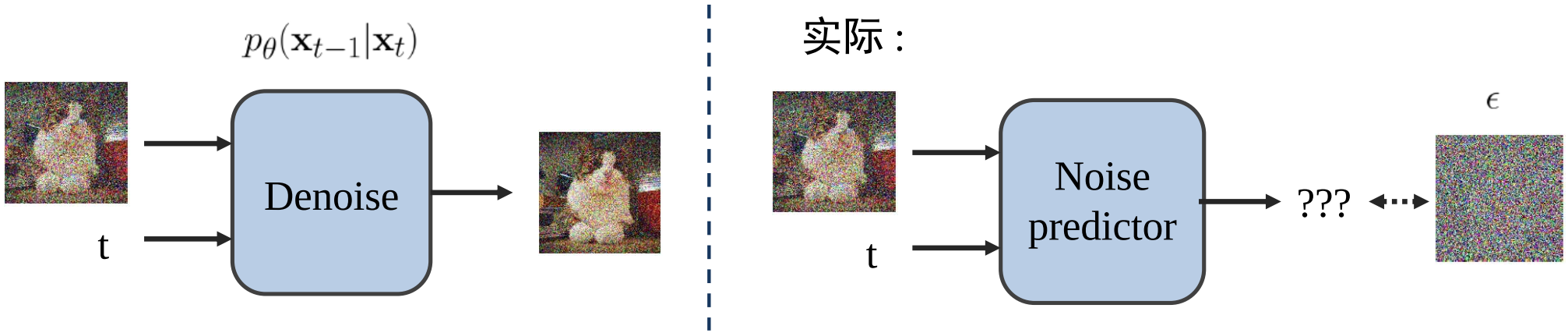
扩散模型

去噪扩散概率模型 (DDPM)

$$L_{\text{simple}} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \sim \mathcal{U}(1,T)} \left[||\epsilon - \epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\, \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon}_{\mathbf{x}_t}, t)||^2 \right]$$

Algorithm 1 Training

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

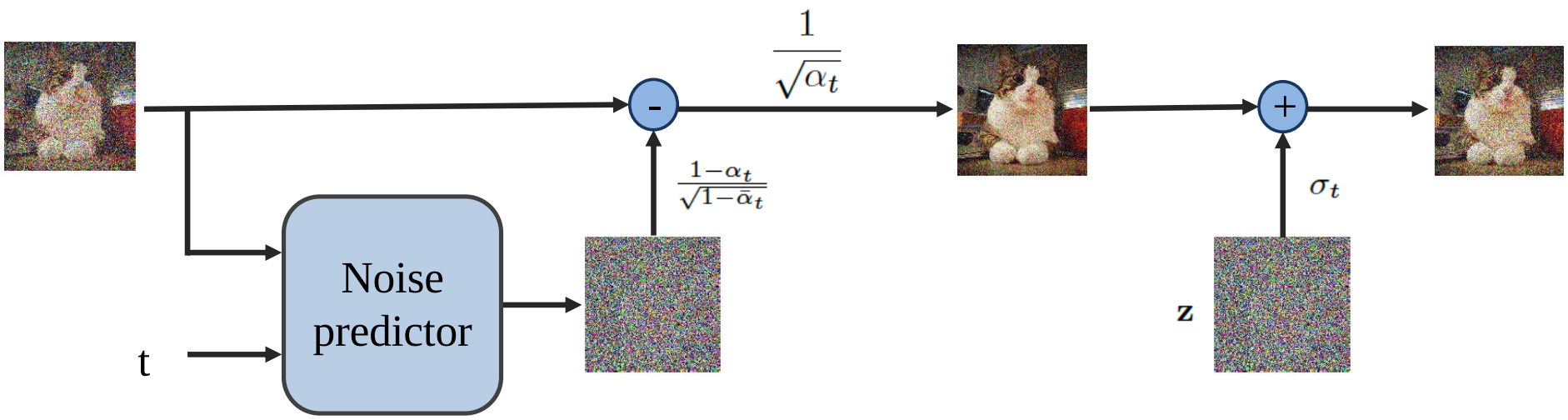


扩散模型

- 去噪扩散概率模型 (DDPM)

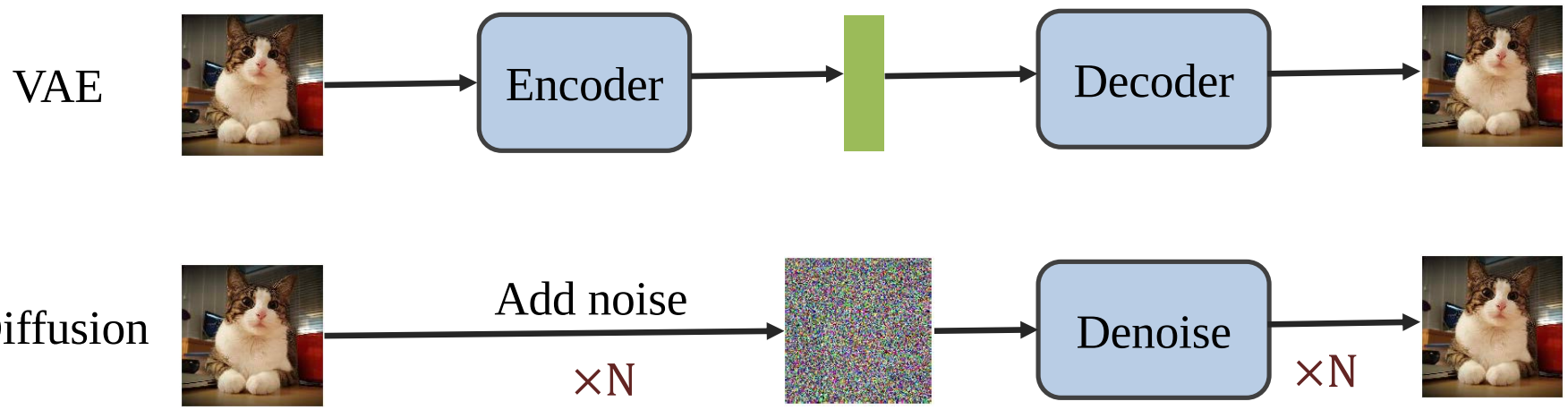
Algorithm 2 Sampling

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: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$



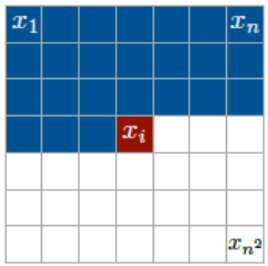
扩散模型

扩散模型 vs. VAE

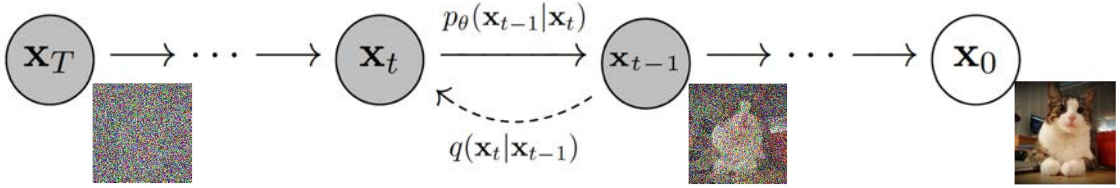


扩散模型 vs. 自回归生成模型

PixelRNN/PixelCNN



Diffusion

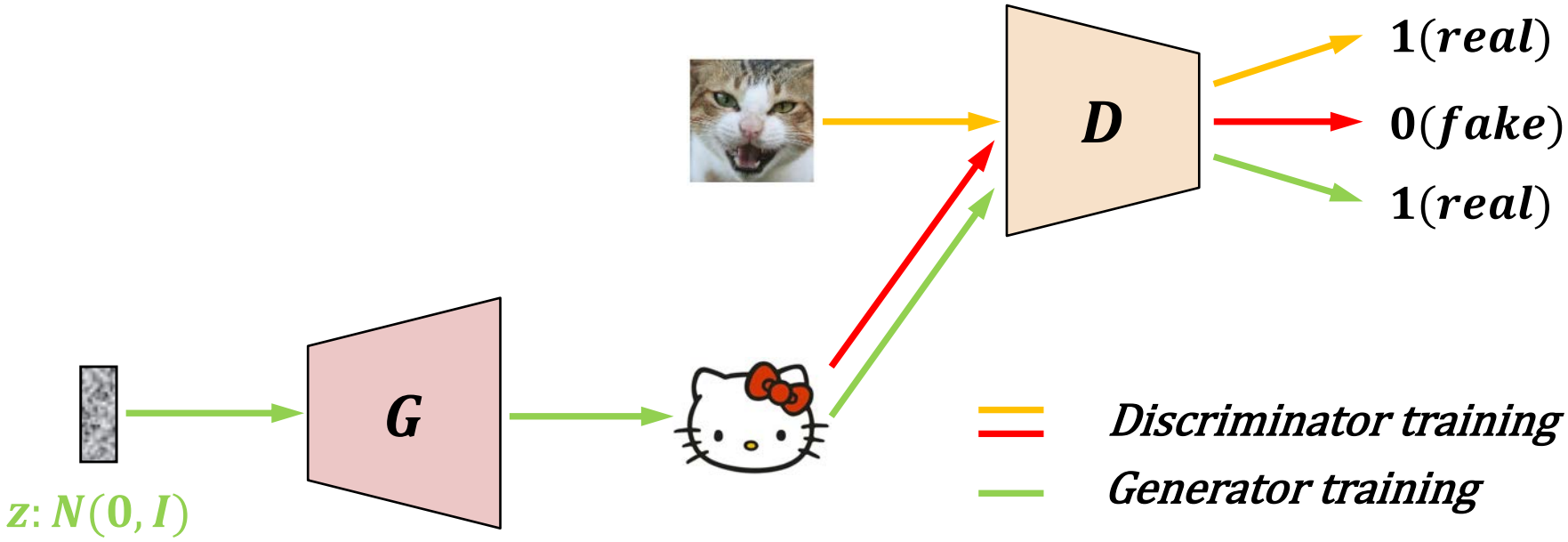


生成式模型（Generative Models）

- 生成式模型
  - 生成式模型概论
  - 自回归生成模型
  - 变分自编码器
  - 扩散模型
  - 生成对抗网络
  - 生成对抗网络的应用

生成对抗网络结构（GAN）

- 生成对抗网络包含两个子网络，生成网络和判别网络，可以分别用G和D表示。
- 生成网络G的输入为均值为0，协方差矩阵为单位矩阵的高斯噪声，输出为生成的“假”数据；判别网络D可以看作一个二分类器，用于分辨输入的数据是“真”数据还是“假”数据。



训练过程

- GAN受博弈论中的零和博弈启发，将生成问题视作判别器D和生成器G这两个网络的对抗和博弈：生成器以随机噪声（一般为均匀分布或者正态分布）为输入，输出生成数据，判别器分辨生成数据和真实数据。前者试图产生更真实的数据，相应地，后者试图更完美地分辨真实数据与生成数据。

- 由此，两个网络在对抗中进步，在进步后继续对抗，由生成式网络得的数据也就越来越完美，越来越逼近真实数据，从而可以生成想要得到的数据。

训练过程

- 设z为随机噪声，x为真实数据，生成网络和判别网络可以分别用G和D表示，其中D可以看作一个二分类器，那么采用交叉熵表示，GAN的优化目标可以写作：

$$\min_G \max_D \mathcal{V}(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z)))]$$

- 其中第一项的$D(x)$表示判别器对真实数据的判断，第二项$D(G(z))$则表示对生成数据的判断。
  - 判别器D的目标是最大化这个公式，也就是甄别出哪些数据是来自真实数据分布的。
  - 生成器G的目标是最小化这个公式，也就是让自己生成的数据被判别器判断为来自真实数据分布。
- 通过这样一个极大极小(Max-Min)博弈，循环交替地分别优化G和D来训练所需要的生成式网络与判别式网络，直到到达Nash均衡点。

损失函数

- 与VAE不同，GAN的优化目标与训练过程来源于博弈的思想，而不需要对隐变量 z 做推断。但实际上继续深入地分析GAN的优化目标就可以发现：它也“暗含着”对分布距离的最小化。

- 固定生成器G的参数，优化更新D的参数时。令$p_r(x)$为真实数据分布，$p_g(x)$为生成数据分布。优化目标可以写为：

$$\begin{aligned}\max_D \mathcal{V}(D,G) &= \mathbb{E}_{x\sim p_r(x)}[\log D(x)] + \mathbb{E}_{x\sim p_g(x)}[\log(1-D(x))] \\ &= \int p_r(x)\log D(x) + p_g(x)\log(1-D(x))\,dx \qquad (1)\end{aligned}$$

- 令(1)式关于$D(x)$的导数为0，可以得到$D(x)$的全局最优解为:

$$D^*(x) = \frac{p_r(x)}{p_r(x)+p_g(x)} \qquad (2)$$

损失函数

$$D^*(x) = \frac{p_r(x)}{p_r(x) + p_g(x)} \tag{2}$$

- (2)式即为最优判别器的表达式，这是一个很直观的结果。当$p_r(x)$与$p_g(x)$完全相同时，则$D^*(x)$恒为0.5，意味着判别器已无法再区分真实数据分布与生成数据分布。

- 固定判别器D的参数，优化更新G的参数时。优化目标可以写为：

$$\min_G \mathcal{V}(D,G) = \mathbb{E}_{x\sim p_r(x)}[\log D(x)] + \mathbb{E}_{x\sim p_g(x)}[\log(1 - D(x))] \tag{3}$$

- 我们将最优判别器的表达式代入(3)式，有：

$$\mathbb{E}_{x\sim p_r(x)} \log \frac{p_r(x)}{p_r(x) + p_g(x)} + \mathbb{E}_{x\sim p_g(x)} \log \frac{p_g(x)}{p_r(x) + p_g(x)} \tag{4}$$

损失函数

$$\mathbb{E}_{x\sim p_r(x)}\log\frac{p_r(x)}{p_r(x)+p_g(x)}+\mathbb{E}_{x\sim p_g(x)}\log\frac{p_g(x)}{p_r(x)+p_g(x)} \tag{4}$$

- 对(4)式变形得到：

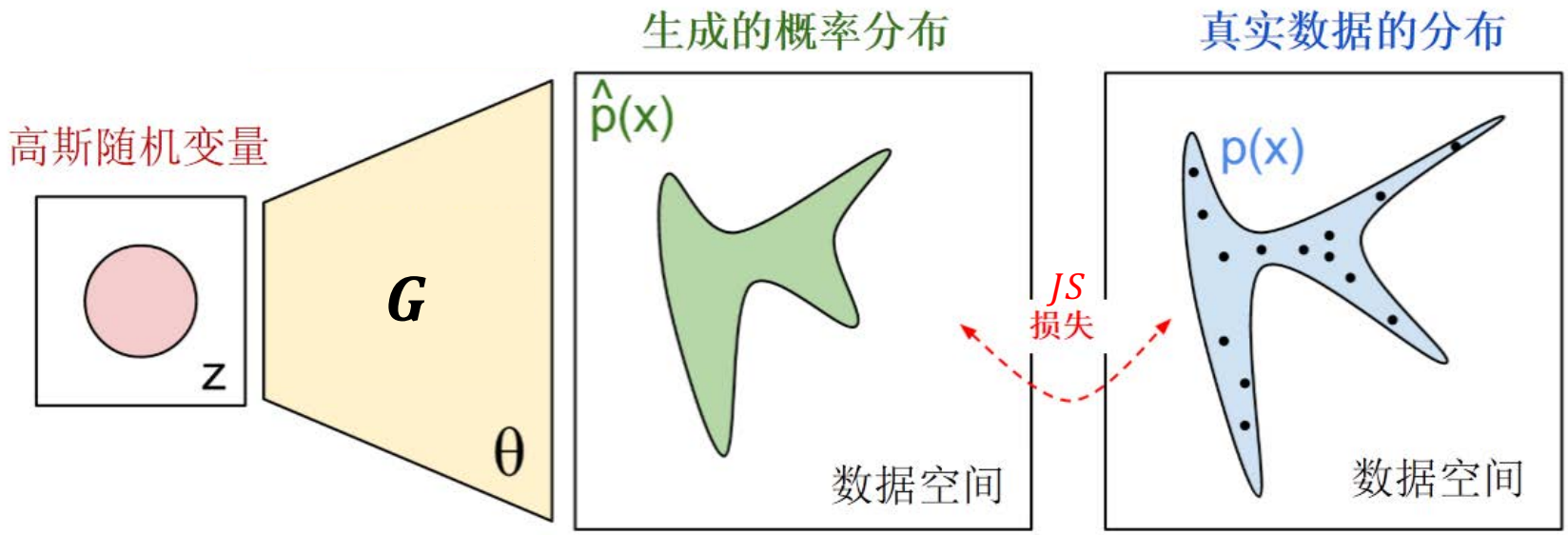
$$\mathbb{E}_{x\sim p_r(x)}\log\frac{0.5\times p_r(x)}{0.5\times(p_r(x)+p_g(x))}+\mathbb{E}_{x\sim p_g(x)}\log\frac{0.5\times p_g(x)}{0.5\times(p_r(x)+p_g(x))}$$

$$=\mathbb{E}_{x\sim p_r(x)}\log\frac{p_r(x)}{0.5\times(p_r(x)+p_g(x))}+\mathbb{E}_{x\sim p_g(x)}\log\frac{p_g(x)}{0.5\times(p_r(x)+p_g(x))}-2\log 2$$

$$=KL(p_r(x)||\frac{p_r(x)+p_g(x)}{2})+KL(p_g(x)||\frac{p_r(x)+p_g(x)}{2})-2\log 2$$

$$=2JS(p_r(x)||p_g(x))-2\log 2 \tag{5}$$

损失函数



$$2JS(p_r(x)||p_g(x)) - 2\log 2 \tag{5}$$

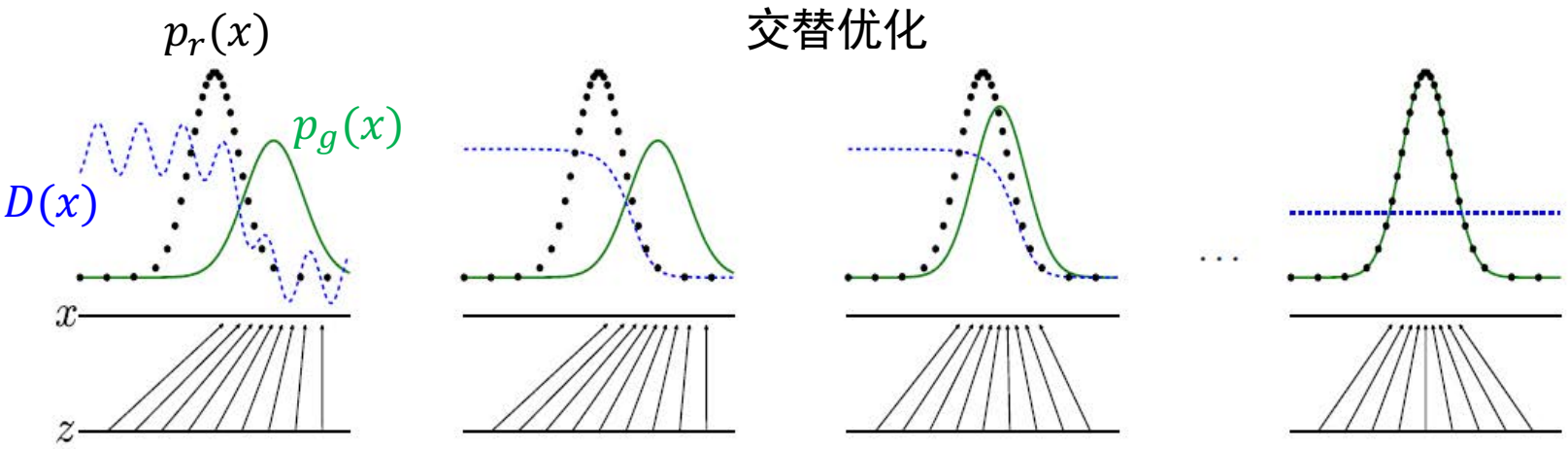
- 可以看到，随着交替优化的进行，判别器D会逐渐接近最优判别器D^*。当这个近似达到一定程度时，生成器G的loss可以近似等价于最小化真实分布与生成器生成分布之间的$JS(Jensen-Shannon)$散度。
- 也就是说：GAN的思想始于博弈论中的零和博弈启发，而等价于数据概率分布的距离优化。

训练过程的理解

- 从训练的角度看，判别器D起到一个二分类器的作用，每一次对D的更新都在增强D区分真实数据与生成数据的能力（即：为两种数据正确分配两种标签），也就是在两种数据间划分正确的决策边界。而G的更新则试图让生成数据也被分类为真实数据，从而使得新的生成数据更加接近决策边界与真实数据。随着交替迭代的不断进行，生成数据会不断接近真实数据，从而最终能够使得D很难再区分，以很高的真实度来拟合真实数据。

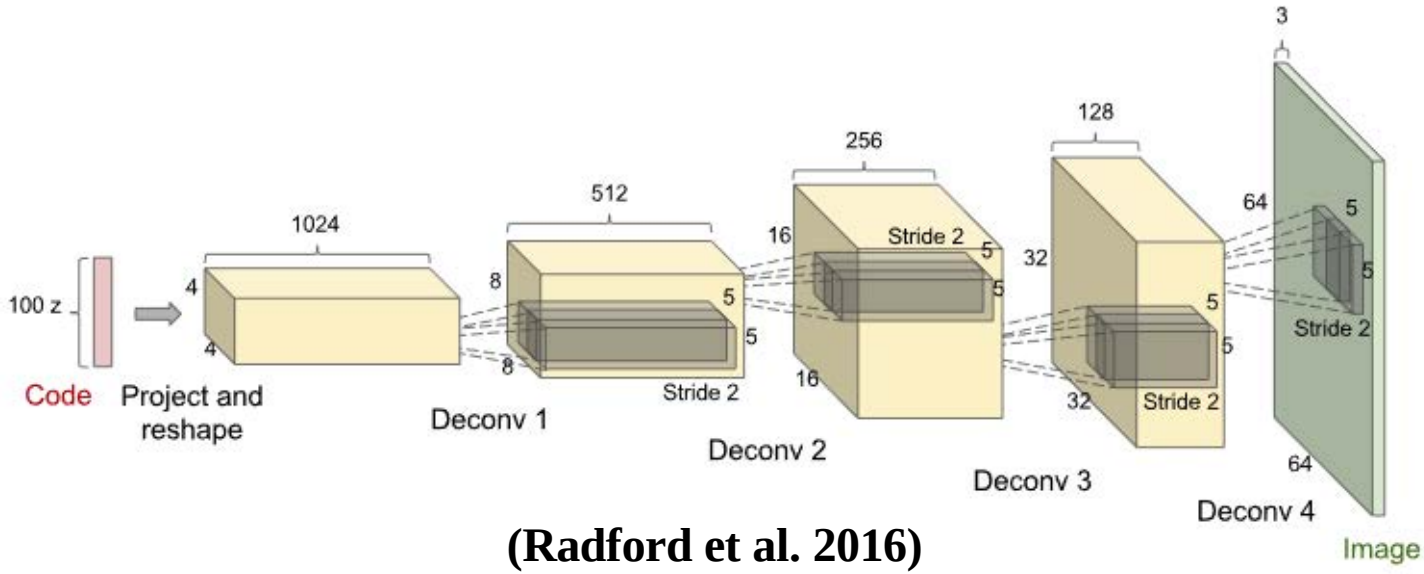
生成器G：将随机向量z映射到数据空间，最大化生成样本被打上真实标签的概率。

判别器D：最大化分配给真实数据与生成数据正确标签的概率。



生成对抗网络结构

- 早期朴素GAN在网络结构上是通过以多层全连接网络为主体的多层感知机(MLP)实现的，然而其调参难度大，训练失败相当常见，生成的图片质量也相当不佳，尤其是对较复杂的数据集而言。



(Radford et al. 2016)

- 因此判别式模型发展的成果被引入到了生成模型中，称作深度卷积对抗神经网络(DCGAN)。DCGAN结构虽然没有带来理论上的解释性，但其强大的图片生成效果使其成为GAN训练中使用非常广泛的网络结构。

- A. Radford, L. Metz, and S. Chintala. “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”. In ICLR, 2016.

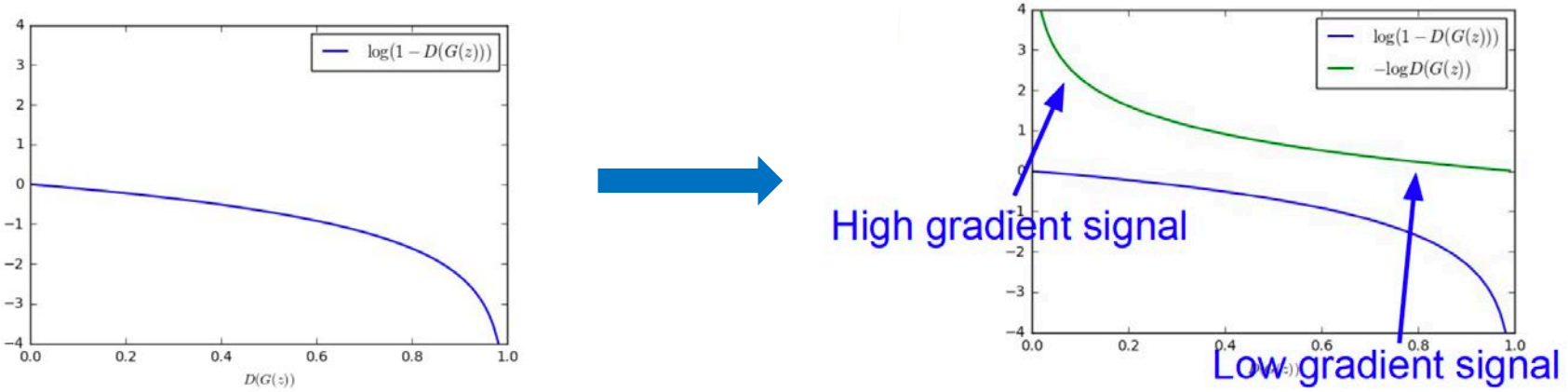
生成对抗网络的优缺点

- GAN的优点：
  - 模型优化只用到了反向传播，而不需要计算马尔可夫链；
  - 模型的训练不需要对隐变量做推断；
  - 基于博弈的思想，判别器作为可自主学习的度量函数，定义十分灵活，使得GAN很容易与其他任务结合起来做对抗训练。

生成对抗网络的优缺点

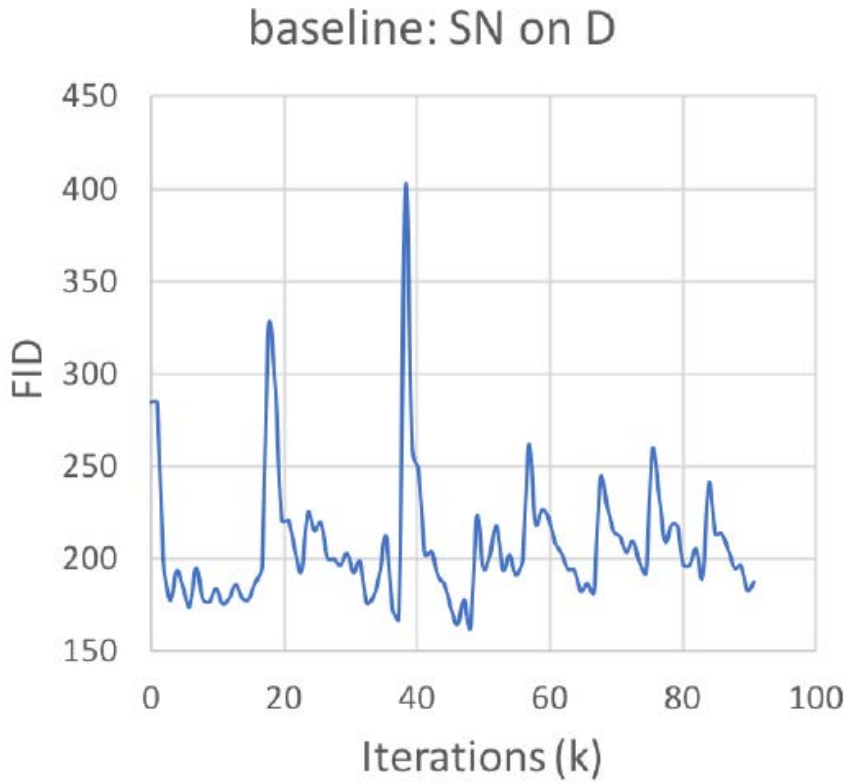
- GAN的缺点：
  - 原始GAN训练的过程中会出现梯度消失的现象，使得生成器G几乎无法获得有效的学习信息，因此很难收敛；

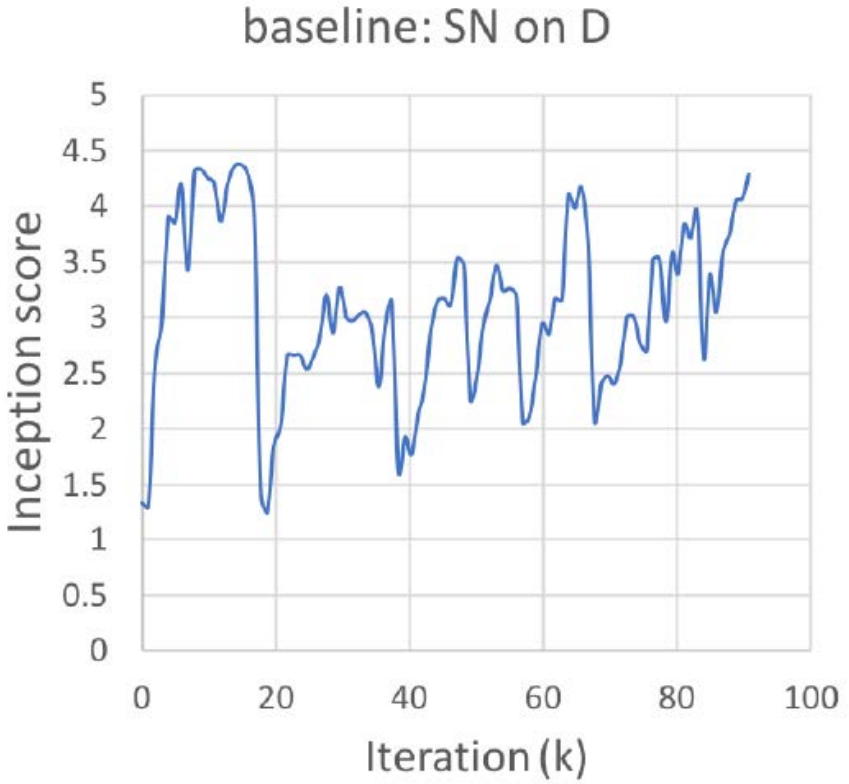
$$\min_G \max_D \mathcal{V}(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p(z)}[\log(1-D(G(z)))]$$



生成对抗网络的优缺点

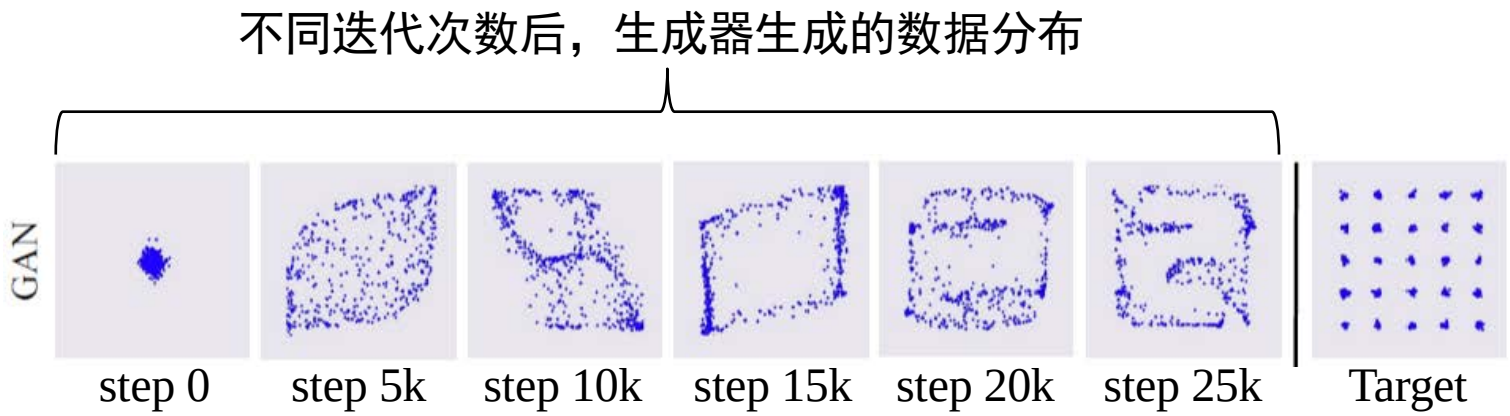
- GAN的缺点：
  - 原始GAN的训练很不稳定，判别器D与生成器G之间需要很好的同步，例如： D每更新k次，G更新一次；
    - ✓ FID: 分布之间的距离
    - ✓ inception score: 生成模式的多样性





生成对抗网络的优缺点

- GAN的缺点：
  - 原始GAN训练完成后，生成的样本会出现模式丢失（mode collapse）的问题。



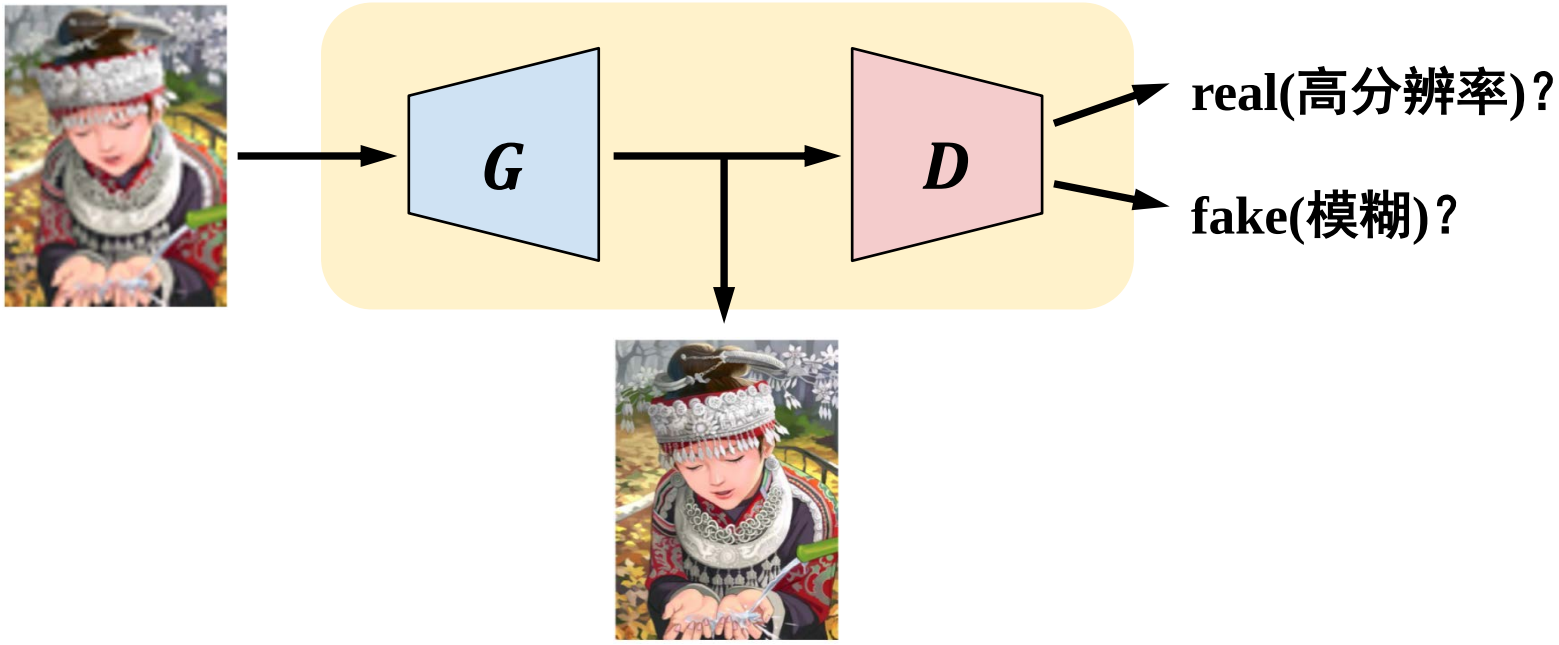
生成图片缺乏多样性

生成式模型（Generative Models）

- 生成式模型
  - 生成式模型概论
  - 自回归生成模型
  - 变分自编码器
  - 扩散模型
  - 生成对抗网络
  - 生成对抗网络的应用
    - ✓ 图像生成
    - ✓ 无监督特征提取
    - ✓ 图像修复
    - ✓ 图像超分辨率
    - ✓ 图像翻译
    - ✓ 基于风格的生成网络

“假”数据定义的拓展

- GAN的判别器可以看作一个二分类器：给真实数据打上“真”标签，给生成数据打上“假”标签。那么，这个所谓的“假”数据的定义真的是一成不变的吗？
- 例如，在超分辨任务中，可以将“假”数据的定义拓展到模糊的图片。通过这样拓展，GAN可以被运用在各种任务之上。



- Ledig, Christian, et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network”. In CVPR, *2017*.

图像生成

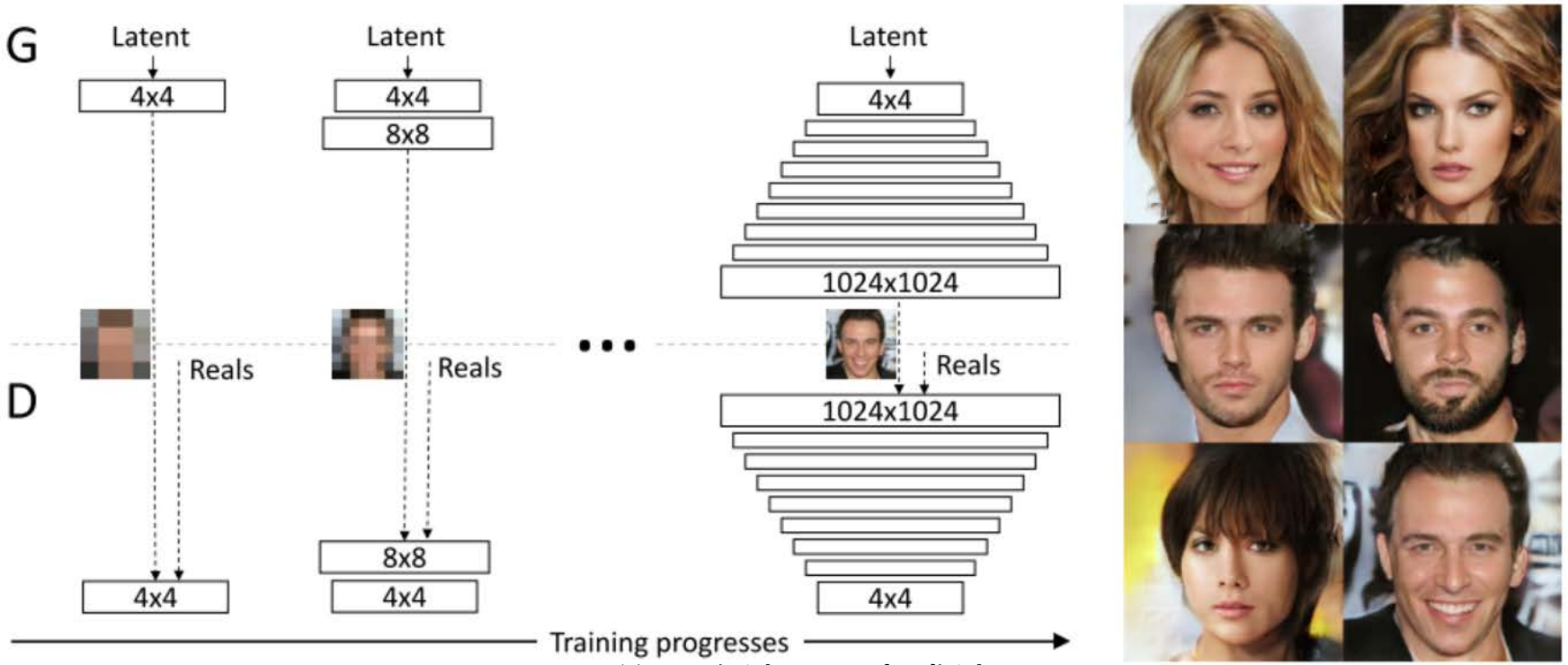
- GAN最初为了图像生成任务被提出，但是朴素的GAN只能生成低分辨率的图片，且质量不高。随着GAN技术的不断进步，近来的BigGAN、PG-GAN等，已经可以生成高分辨率的图像。

DCGAN在 ImageNet 数据集上生成的结果

BigGAN在 ImageNet 数据集上生成的结果

图像生成

- PG-GAN可以在无条件的情况下，生成1024 ×1024大小的高清人脸。通常来说，直接训练生成这样高清的图像是非常困难的。
- PG-GAN采用由粗到细（coarse-to-fine）的逐步优化策略，先生成低分辨率或者低质量的图像，然后不断地增加分辨率或细节。



PG-GAN的网络结果和生成结果

- Karras, Aila, et al. "Progressive growing of GANs for improved quality, stability, and variation". In ICLR, *2018*.

无监督特征提取

- 利用GAN的无监督特征提取主要分为两类方法。第一类方法是利用训练GAN过程中得到的判别器，将判别器卷积层提取的特征作为无监督特征，运用于分类、检测等任务。

Model	Accuracy	Accuracy (400 per class)	max # of features units
1 Layer K-means	80.6%	63.7% (±0.7%)	4800
3 Layer K-means Learned RF	82.0%	70.7% (±0.7%)	3200
View Invariant K-means	81.9%	72.6% (±0.7%)	6400
Exemplar CNN	84.3%	77.4% (±0.2%)	1024
DCGAN (ours) + L2-SVM	82.8%	73.8% (±0.4%)	512

DCGAN判别器在CIFAR10上的分类准确率

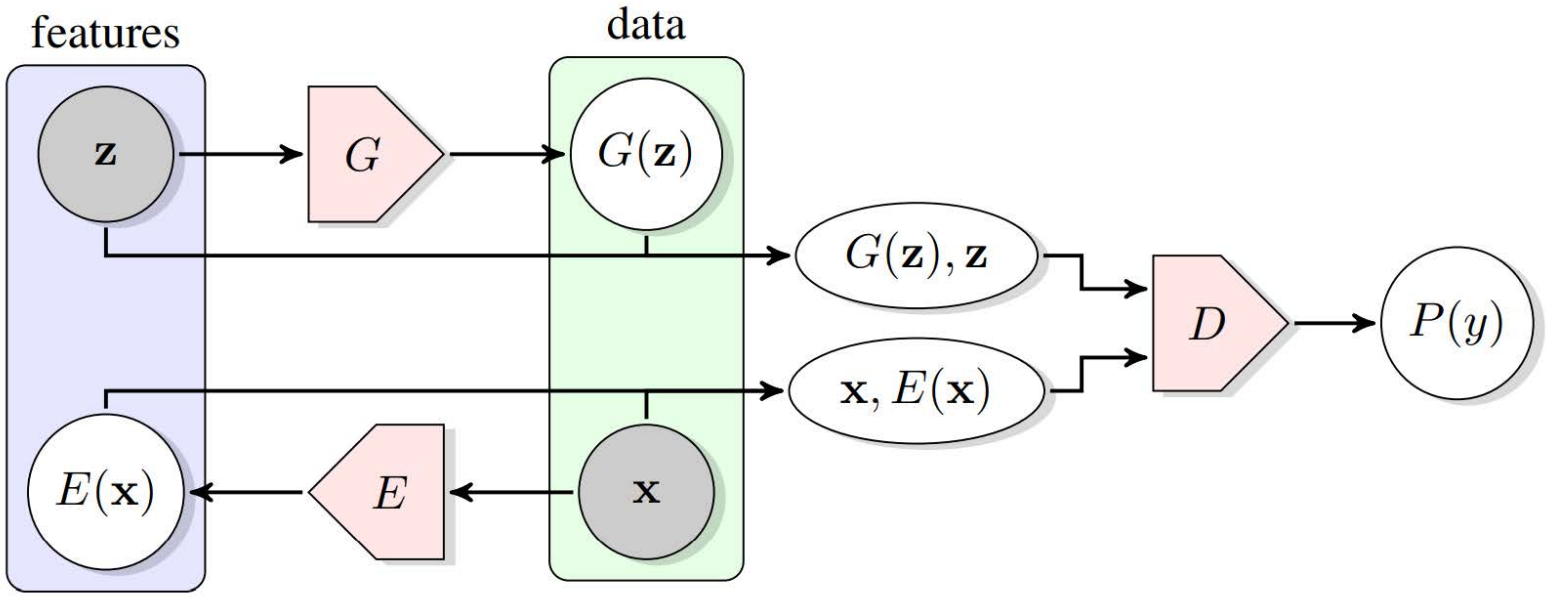
Model	error rate
KNN	77.93%
TSVM	66.55%
M1+KNN	65.63%
M1+TSVM	54.33%
M1+M2	36.02%
SWWAE without dropout	27.83%
SWWAE with dropout	23.56%
DCGAN (ours) + L2-SVM	22.48%
Supervised CNN with the same architecture	28.87% (validation)

DCGAN判别器在SVHN上的分类错误率

- A. Radford, L. Metz, and S. Chintala. “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”. In ICLR, 2016.

无监督特征提取

- 第二类方法是在朴素GAN的基础上，增加一个编码器。编码器与GAN同时训练，编码器用于将“真”样本编码为特征向量。
- 训练结束后，将编码器作为无监督特征提取器使用。



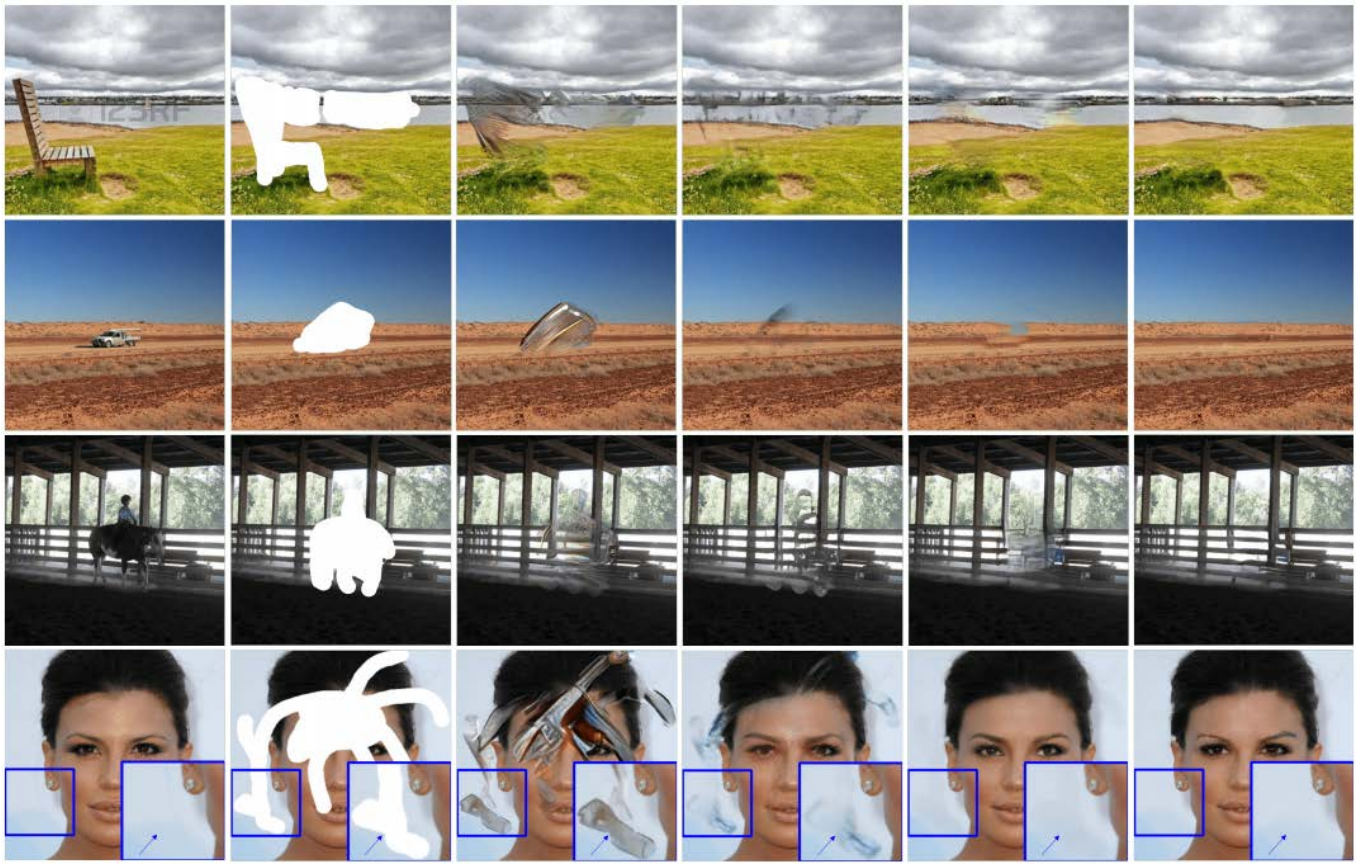
BiGAN 的网络结构

$$V(D,E,G) := \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{X}}}\Big[\underbrace{\mathbb{E}_{\mathbf{z}\sim p_E(\cdot|\mathbf{x})}\left[\log D(\mathbf{x},\mathbf{z})\right]}_{\log D(\mathbf{x},E(\mathbf{x}))}\Big] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{Z}}}\Big[\underbrace{\mathbb{E}_{\mathbf{x}\sim p_G(\cdot|\mathbf{z})}\left[\log\left(1-D(\mathbf{x},\mathbf{z})\right)\right]}_{\log(1-D(G(\mathbf{z}),\mathbf{z}))}\Big].$$

- Jeff Donahue, Philipp Krähenbühl, et al. "Adversarial feature learning”. In ICLR, 2017.

图像修复

□ 图像块修复



Original　Input　Global & Local　Context-Attention　PartialConv　SN-PatchGAN

- Yu, Lin, et al. “Free-Form Image Inpainting with Gated Convolution”. In ICCV, 2019.

图像修复

- 图像去雨滴







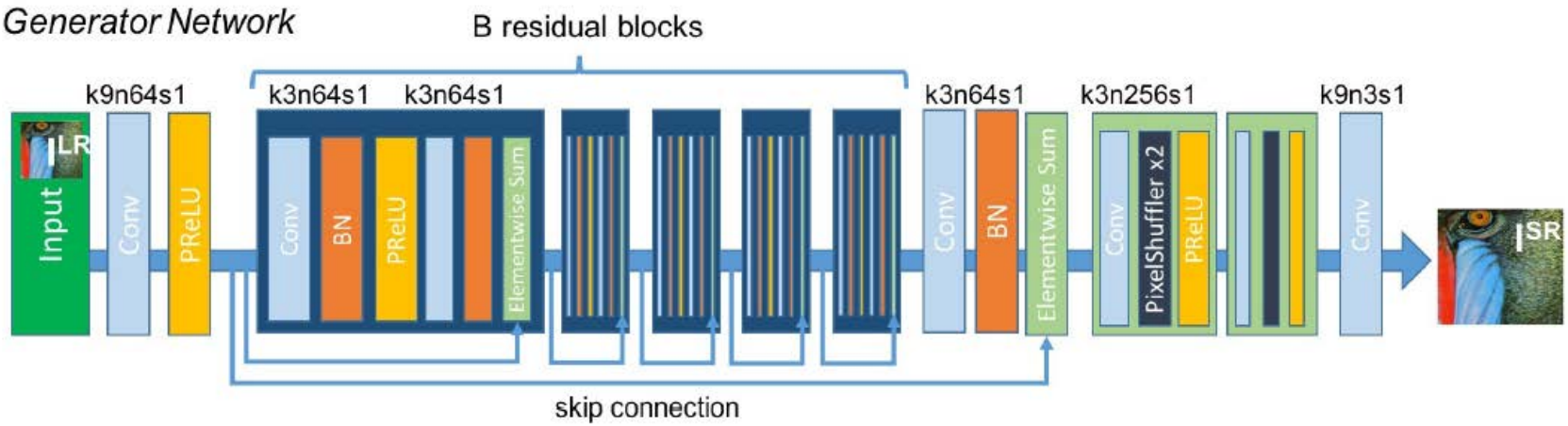

with raindrop　　raindrop removed　　with raindrop　　raindrop removed

- R. Qian, et al. "Attentive generative adversarial network for raindrop removal from a single image”. In CVPR, 2018.

图像超分辨率

- 生成网络：生成低分辨率图片对应的超分辨率图片

低分辨率图片

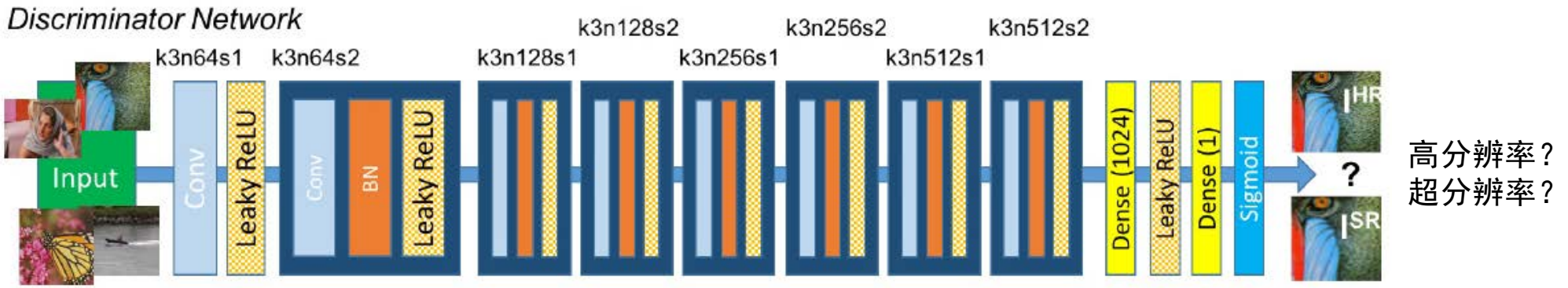


超分辨率图片

- Ledig, Christian, et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network". In CVPR, 2017.

图像超分辨率

- 判别网络：发现生成的与真实的高分辨率图片之间的区别



✓ content loss:

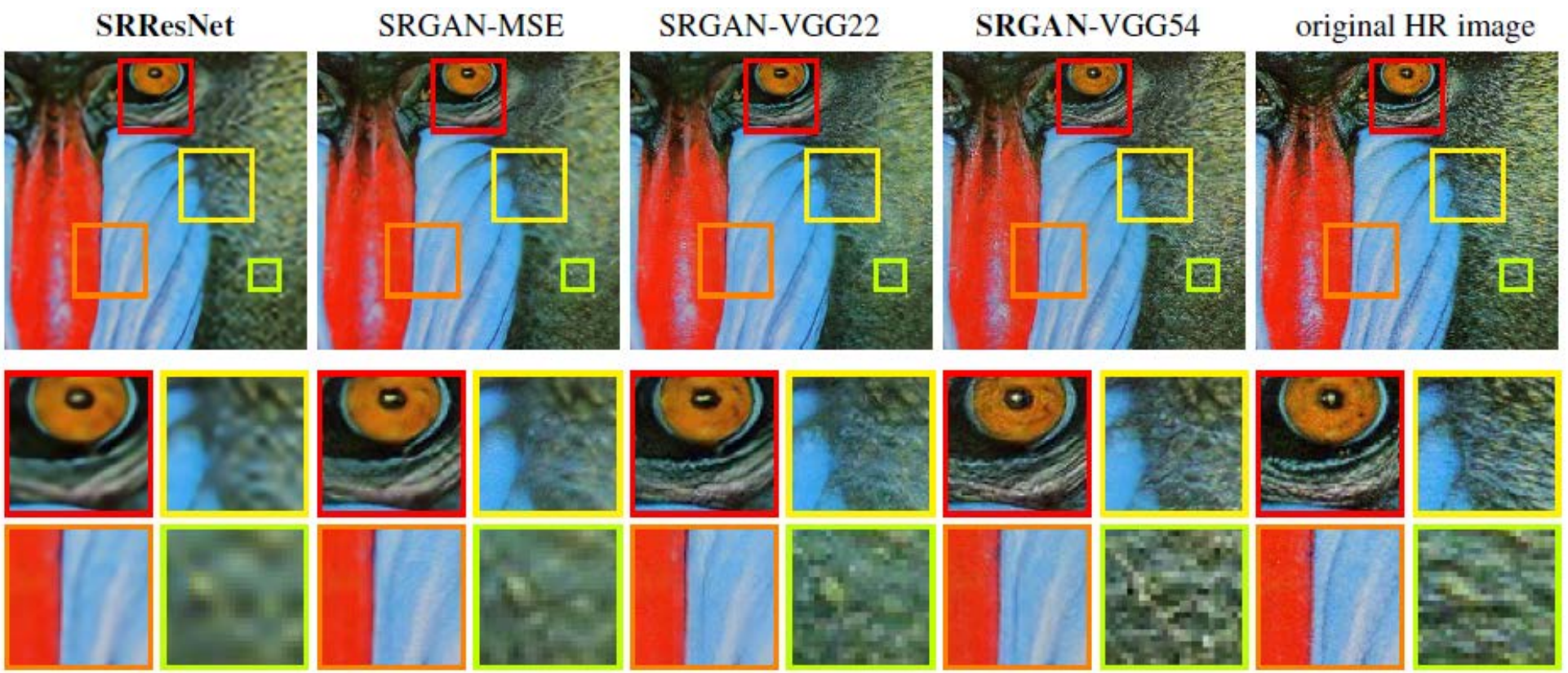
$$l^{SR}_{VGG/i.j} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2$$

✓ adversarial loss:

$$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{I^{HR} \sim p_{\text{train}}(I^{HR})}[\log D_{\theta_D}(I^{HR})] + \mathbb{E}_{I^{LR} \sim p_G(I^{LR})}[\log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR})))]$$

图像超分辨率

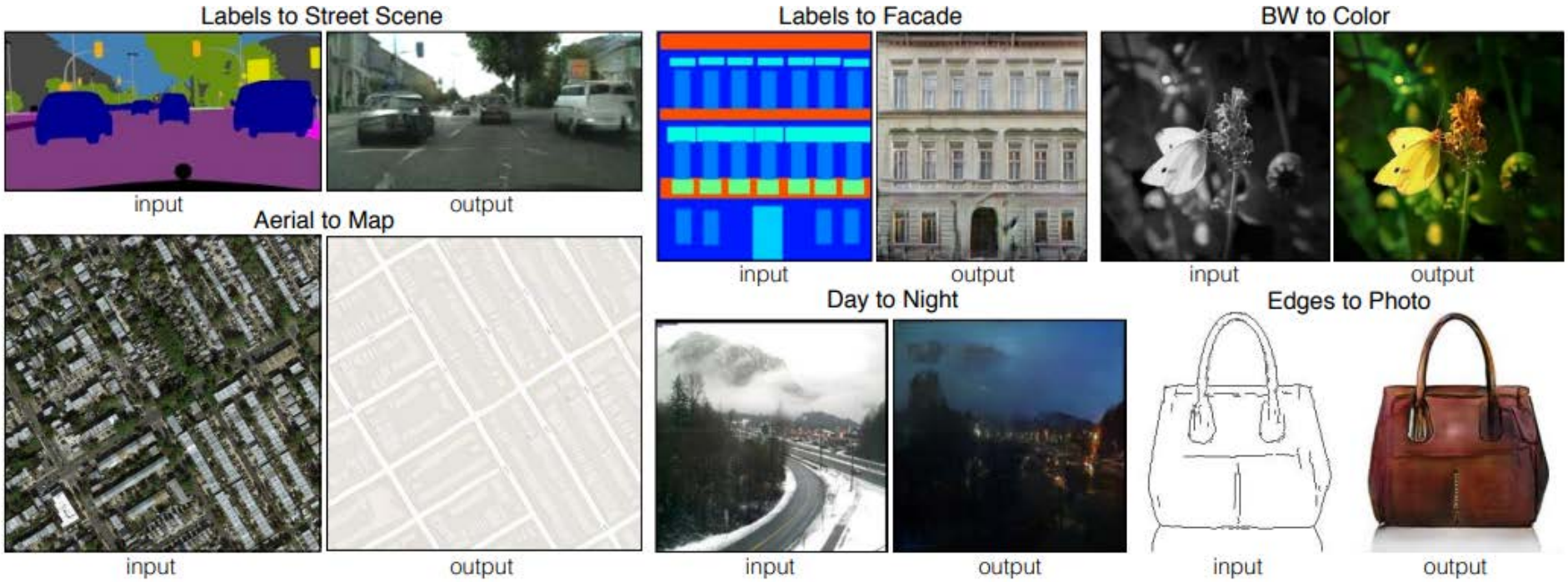
- 实验结果对比



图片超分辨率效果比较

图像翻译

- 研究目的
  - 图像→图像的翻译



- Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks". In CVPR, 2017.

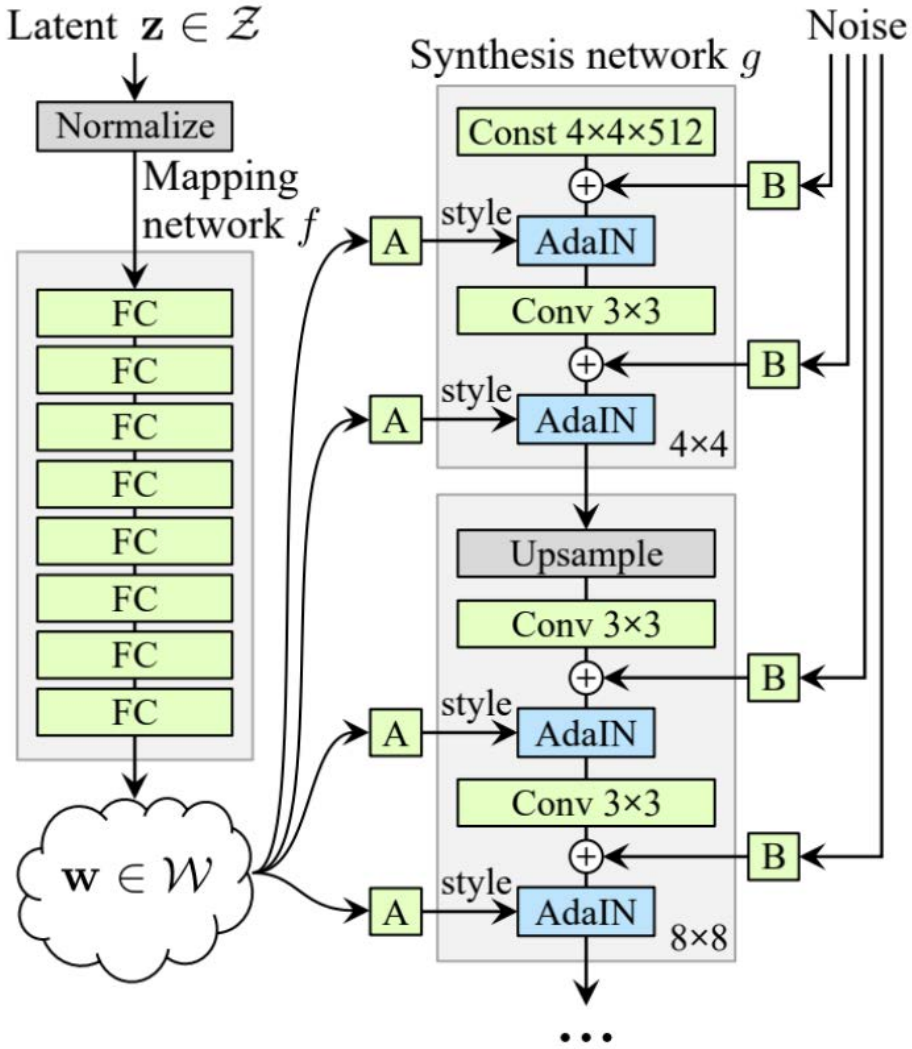
图像翻译

- 在图像翻译任务中，有些时候，获取成对的训练数据是很困难的，有时候甚至根本无法获得。
  - 例如，下图中的自然图片到油画的风格转换，难以获得同一场景下的自然图片与其油画这样的图片对作为训练集。
- 相比使用MSE等方法作为度量，使用GAN作为度量不需要训练数据是成对的。

- J.Y. Zhu, et al. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". In ICCV, 2017.

基于风格的生成网络：StyleGAN

方法框架结构



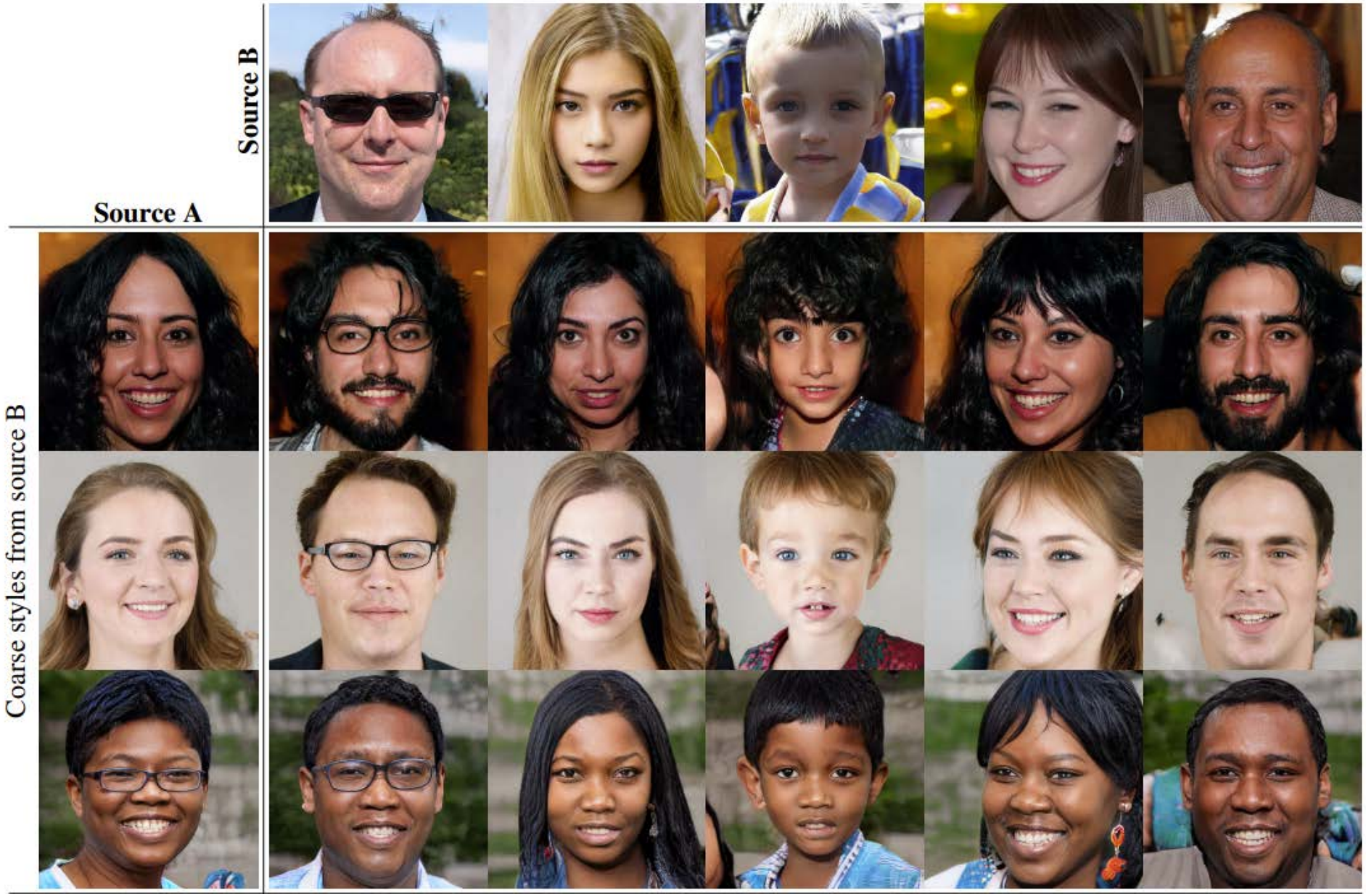
- z 经由多层全连接网络被映射到 $\mathcal{W}$ 空间得到 w，w再经过学习到的仿射变换得到风格变量: $y = (y_s, y_b)$
- y 通过AdaIN (Adaptive Instance Normalization) 结构来控制每一层不同分辨率下的生成

$$\mathrm{AdaIN}\left(x_i, y\right) = y_{s,i}\frac{x_i - \mu\left(x_i\right)}{\sigma\left(x_i\right)} + y_{b,i}$$

- 额外引入的加性噪声则用来为生成图像加入更多的图像细节

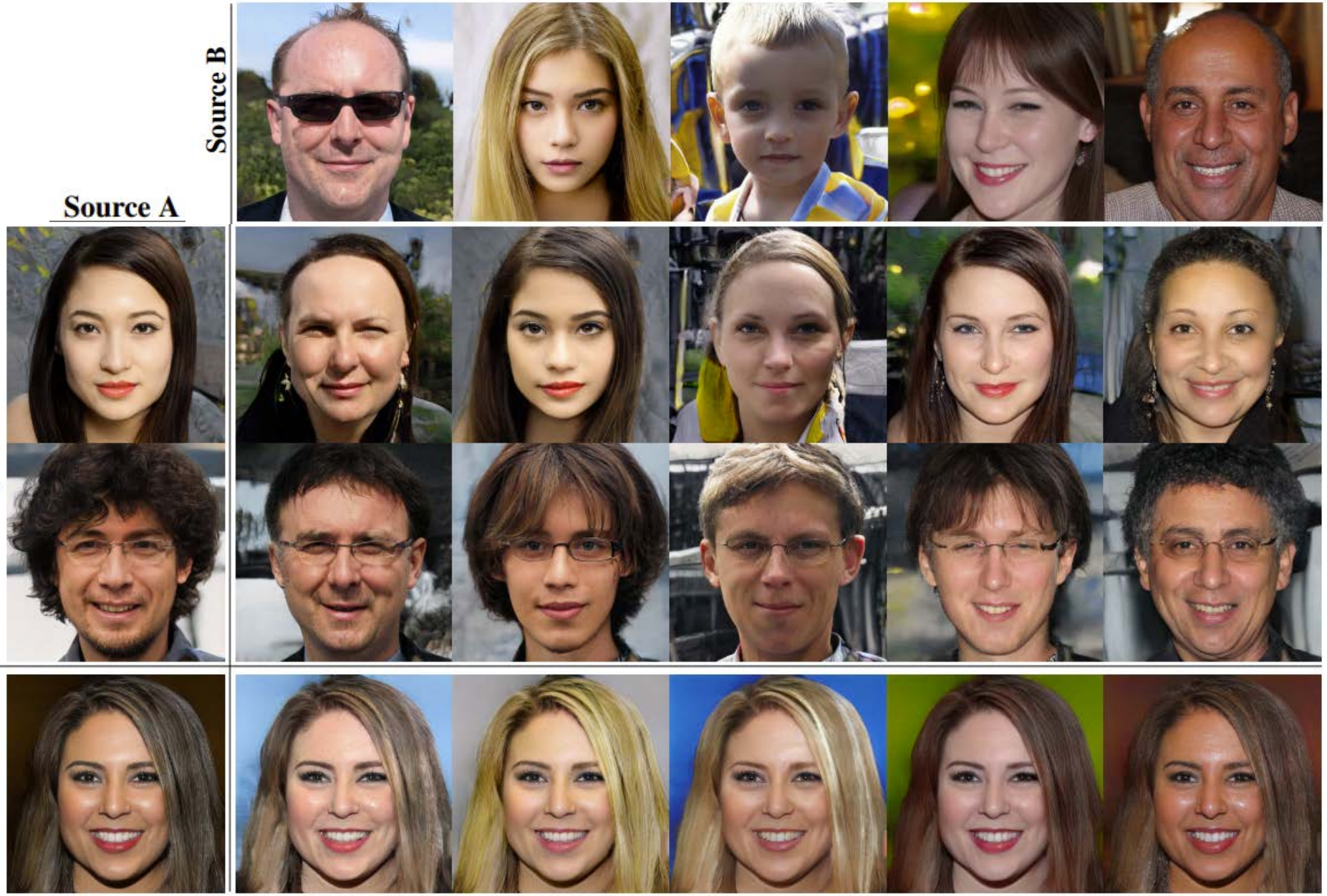
- T. Karras, S. Laine, T. Aila. “A style-based generator architecture for generative adversarial networks”. In CVPR, 2019 .

基于风格的生成网络：StyleGAN



$4^2 \sim 8^2$

基于风格的生成网络：StyleGAN



$16^2 \sim 32^2$

$64^2 \sim 1024^2$

基于风格的生成网络：StyleGAN

基于深度学习的图像分析技术

- 现代深度学习技术实例
  - 图像识别
  - 目标检测
  - 语义分割
  - 视频理解
  - 目标跟踪
  - 注意力机制及Transformer
  - 生成对抗网络
  - 自监督学习

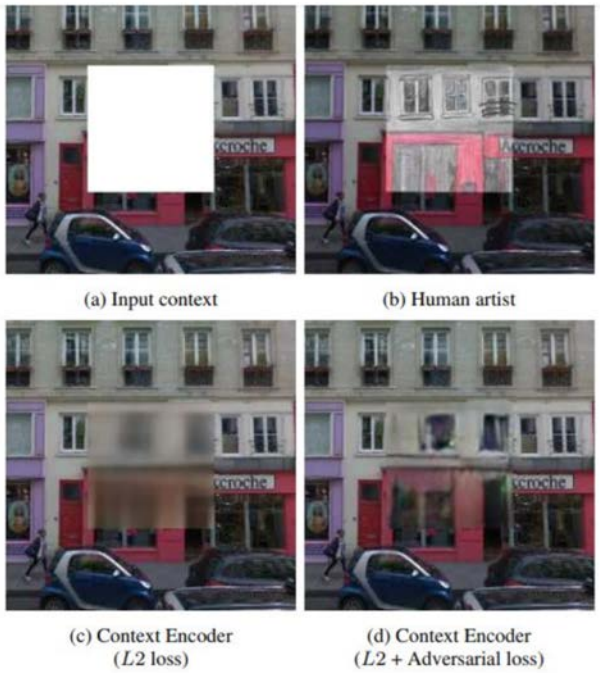
自监督学习

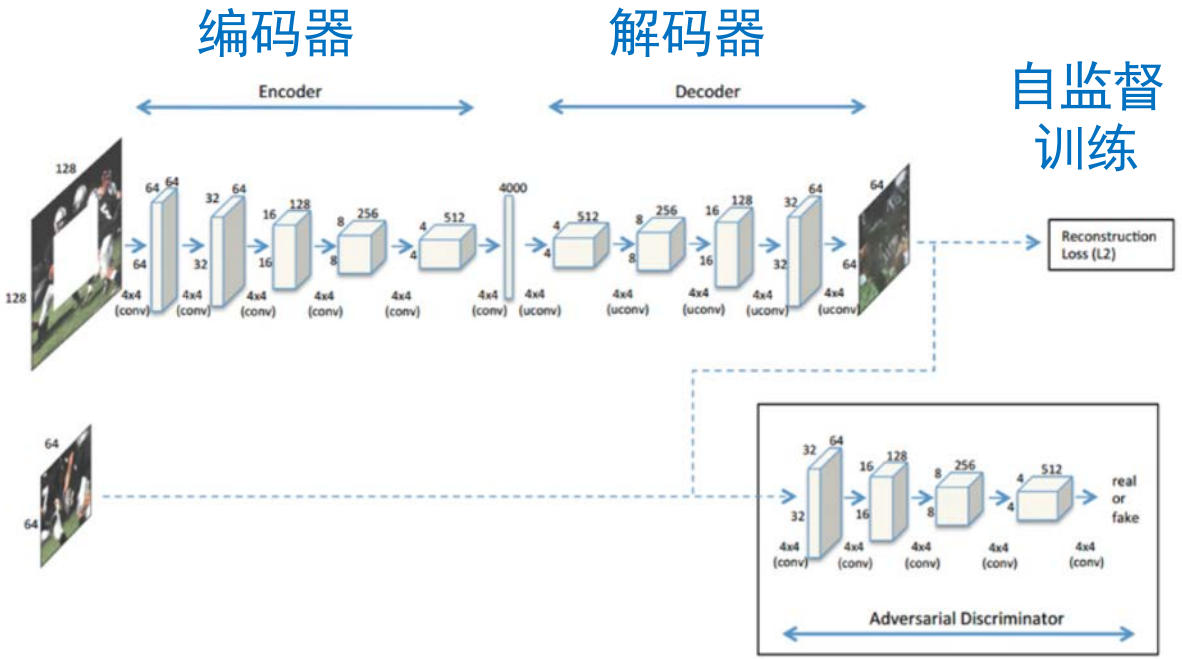
- 基本概念
  - 根据类别未知（没有被标记）的训练样本解决模式识别中的各种问题，称为无监督学习，也常被称为自监督学习
- 自监督学习的由来
  - 缺乏足够的先验知识，因此难以人工标注类别
  - 进行人工标注的成本过高
- 基于深度学习的自监督学习
  - 通常通过挖掘图像、视频中潜在的信息进行模型的自监督训练,训练好的模型可用于特定场景或其他下游任务。
    - ✓ 利用内容信息的自监督学习
    - ✓ 利用空间结构的自监督学习
    - ✓ 利用时序信息的自监督学习
    - ✓ 利用运动信息的自监督学习
    - ✓ 利用样本对比的自监督学习

自监督学习：基于内容信息

方法动机

- 抹除图片局部信息，并使用CNN网络进行恢复
- 如下图(a)，当抹除图片的一部分区域，画家可以手工修复（图(b)），该工作使用基于图像内容的自编码器网络进行图像恢复，如图(c)(d)



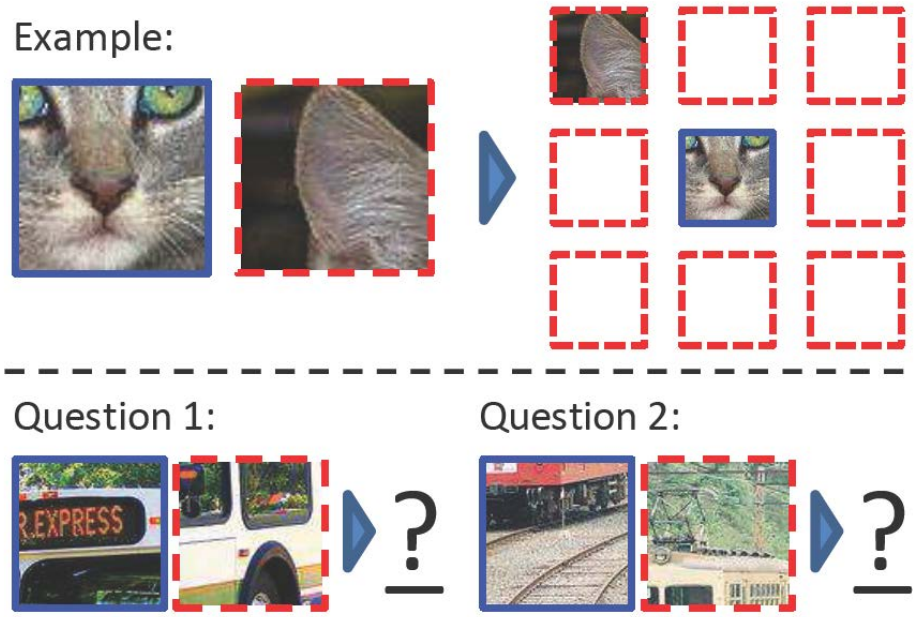


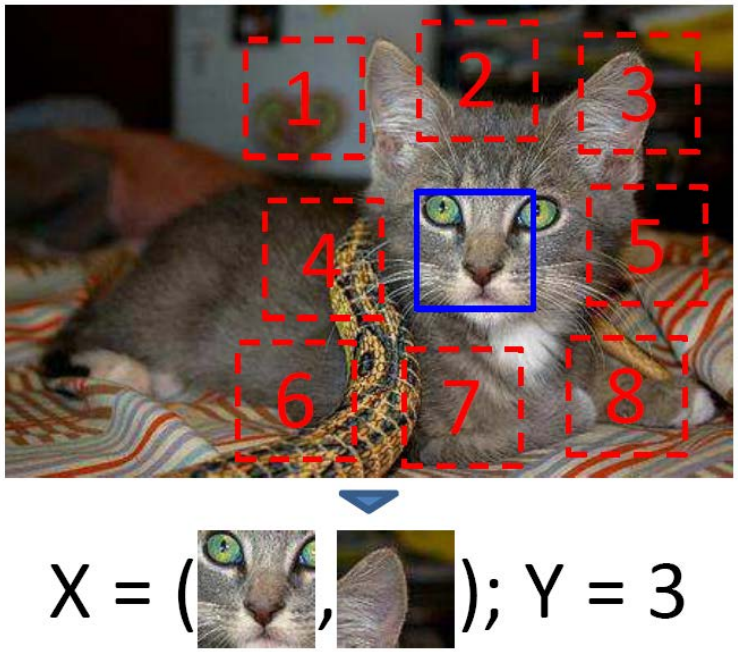
- Deepak Pathak, et al. “Context encoders: feature learning by inpainting”. In CVPR, 2016.

自监督学习：基于空间结构

方法动机

- 图像的局部区域具有空间上的关联，可用于自监督学习
- 左图：给定猫的脸部，我们可以确定另一张图片是猫的左耳
- 右图：将这样一个类似“拼图”的任务以分类的形式进行描述



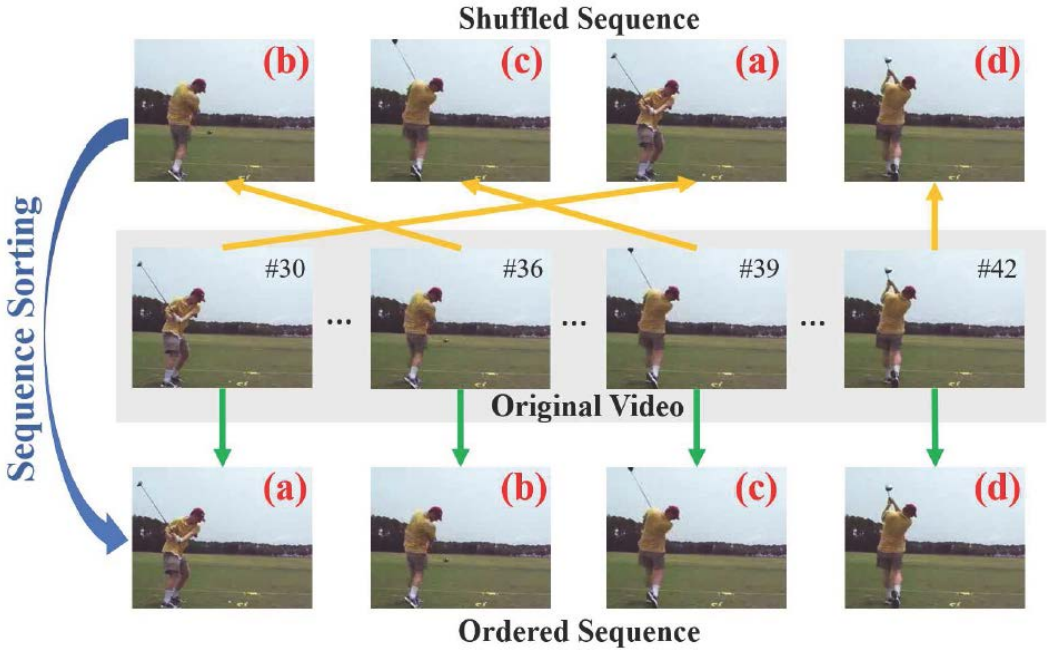


- Carl Doersch, et al. “Unsupervised visual representation learning by context prediction”. In ICCV, 2015.

自监督学习：基于时序信息

方法动机

- 视频帧具有时序上的先后顺序，特别是运动类信息
- 已知视频的帧序，打乱视频帧后，通过网络重新预测视频帧的排序（对不同帧的排列组合进行分类），进行自监督学习

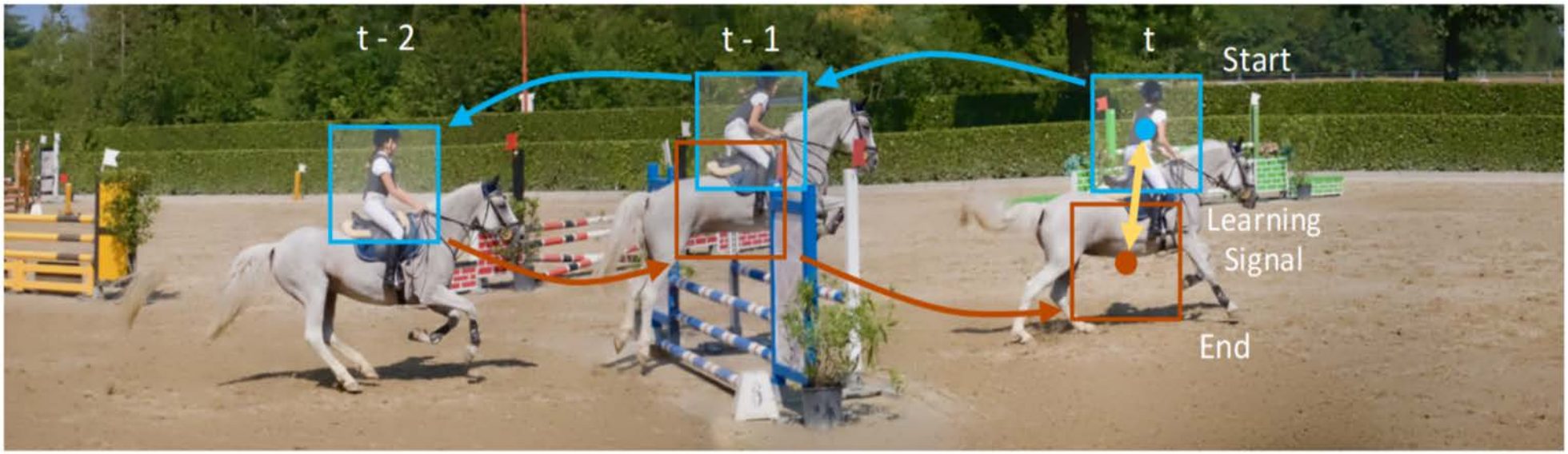


- Hsin-Ying Lee, et al. “Unsupervised representation learning by sorting sequences”. In ICCV, 2017.

自监督学习：基于运动信息

方法动机

- 物体可以双向地进行跟踪，即跟踪器对目标从第一帧跟踪到第N帧，理想地，也可以以第N帧为起始点，反向回到第一帧
- 通过衡量前后跟踪轨迹一致性进行自监督学习

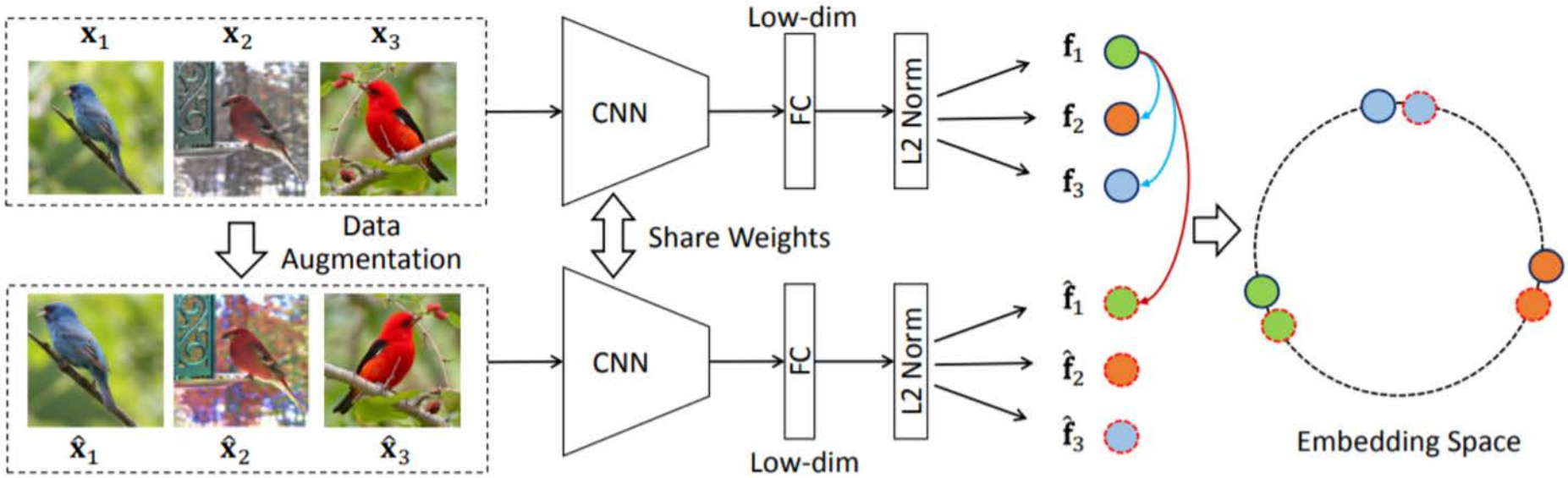


- X.L. Wang, et al. “Learning correspondence from the cycle-consistency of time”. In CVPR, 2019.

自监督学习：基于样本对比

方法动机

- 通过在不同样本间进行比较来学习特征表达。主要通过相似的正样本对，配合不相似的负样本进行对比
- 下图：不同的图片进行数据增广后，变换为三对样本，正样本之间距离拉近，不同样本对相互远离



- Mang Ye, et al. “Unsupervised embedding learning via invariant and spreading instance feature”. In CVPR, 2019.