

# C A G A N 论文复现

The Conditional Analogy GAN: Swapping  
Fashion Articles on People Images, 2017



# 目录

01

CAGAN

02

实现思路

03

代码细节

04

实验

05

提升方向

06

Q&A

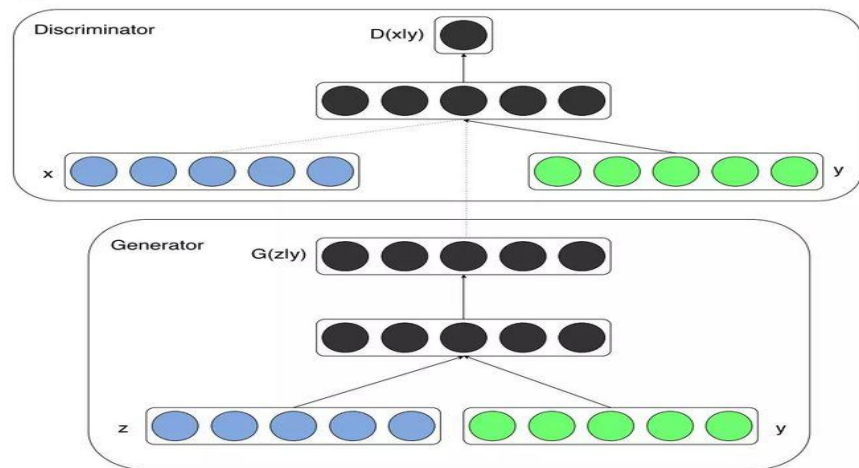
# CAGAN文章的基础

## 1 cGAN (conditional GAN)

将条件信息 $y$ 加入原始GAN的 $x$ 和 $z$ 中。

对应CAGAN文章中，通过在 Discriminator loss 中添加项来学习“某件衣服是否是模特身上穿的”这一隐含关系。

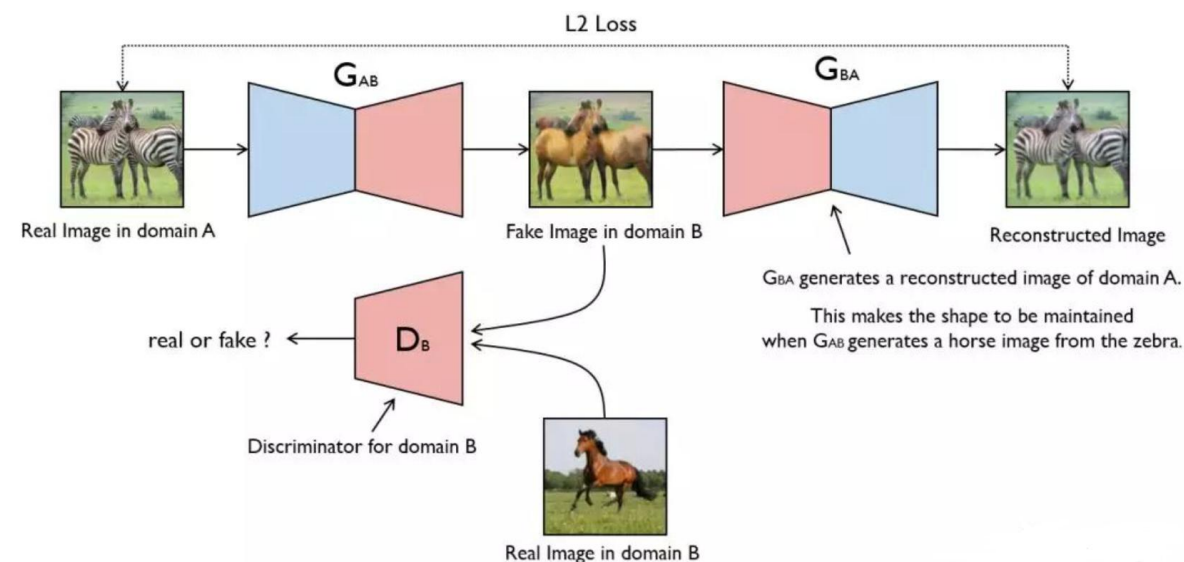
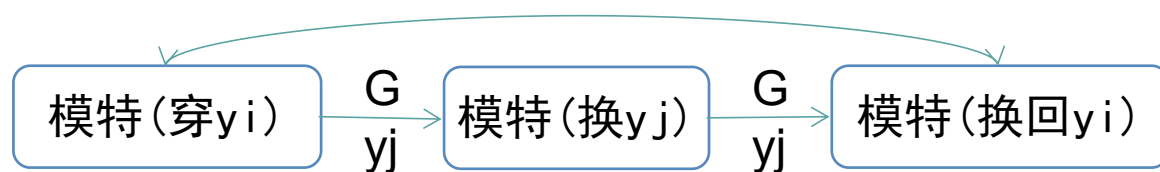
$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x|y)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z|y)))]$$



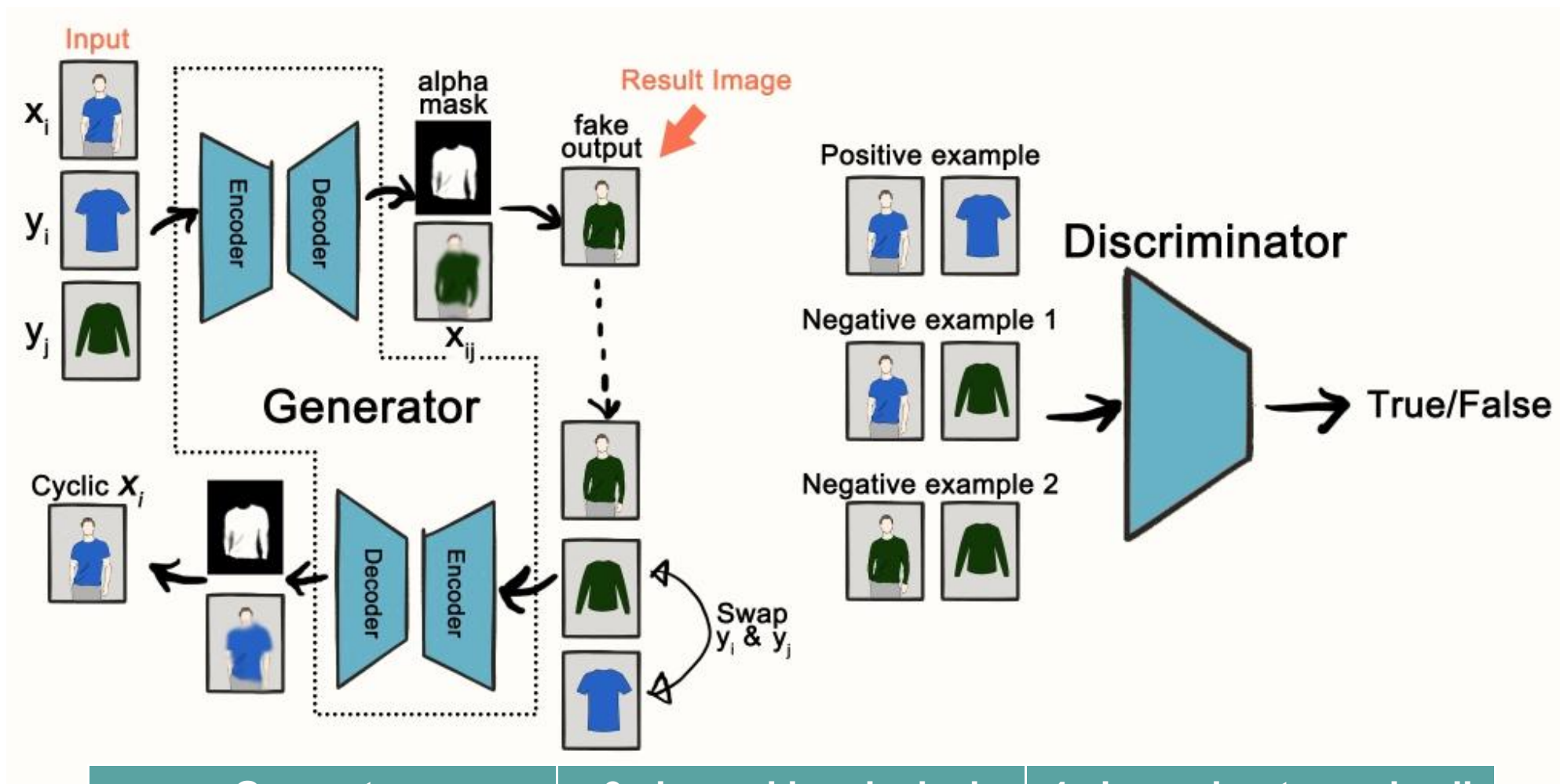
量子位

## 2 CycleGAN

提出cycle consistency。CAGAN中，只训练一个生成器 $G$ ，对应的cycle:



# 实现思路

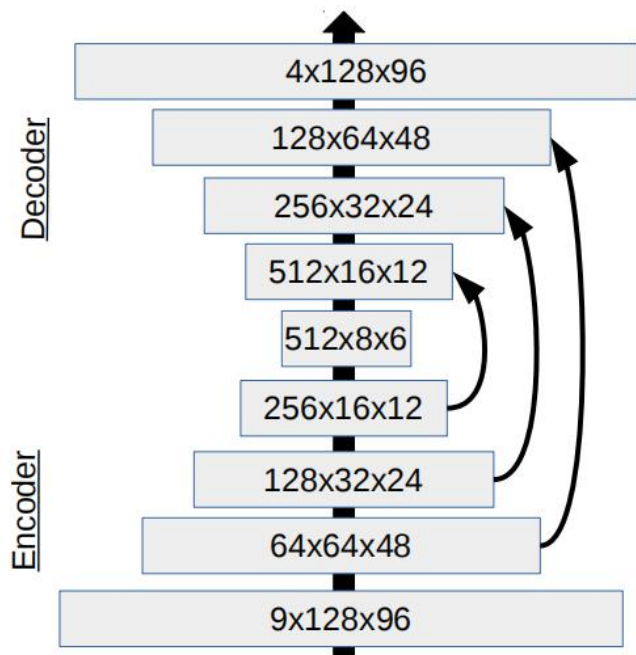


Generator	9 channel in: $x_i, y_i, y_j$	4 channel out: mask, $x_{ij}$
Discriminator	6 channel in: $x, y$	1 channel out: prediction

# 实现思路

- Generator : 修改版U-Net

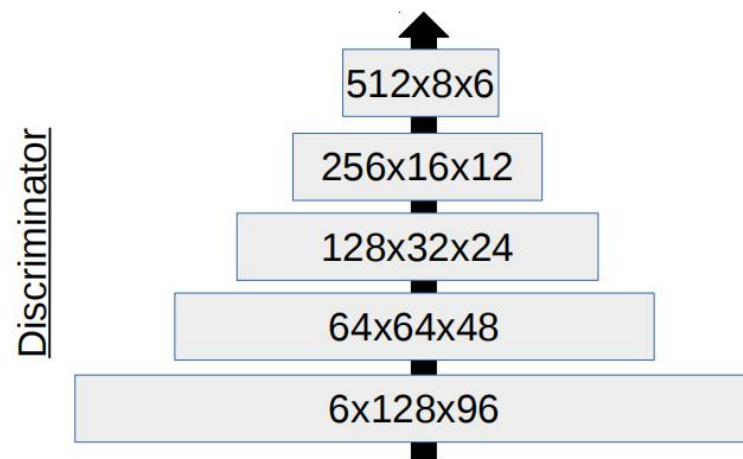
不同点: Encoder部分每一层 feature map 拼接  $x_i, y_i$  (模特及其衣服) 的6通道下采样结果。



- Discriminator:

来自 pix2pix 中提出的 PatchGAN;

输出的实质是  $8 \times 6$  图像, 其中每个像素对应输入图像一个 patch 的可信度。



# 实现思路

- Loss

$$\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \gamma_i \mathcal{L}_{id}(G) + \gamma_c \mathcal{L}_{cyc}(G)$$

$$\begin{aligned} \mathcal{L}_{cGAN}(G, D) = & \mathbb{E}_{x_i, y_i \sim p_{\text{data}}} \sum_{\lambda, \mu} [\log D_{\lambda, \mu}(x_i, y_i)] \\ & + \mathbb{E}_{x_i, y_i, y_j \sim p_{\text{data}}} \sum_{\lambda, \mu} [(1 - \log D_{\lambda, \mu}(G(x_i, y_i, y_j), y_j))] \\ & + \mathbb{E}_{x_i, y_j \neq i \sim p_{\text{data}}} \sum_{\lambda, \mu} [(1 - \log D_{\lambda, \mu}(x_i, y_j))], \end{aligned}$$

$$\mathcal{L}_{id}(G) = \mathbb{E}_{x_i, y_i, y_j \sim p_{\text{data}}} \|\alpha_i^j\|$$

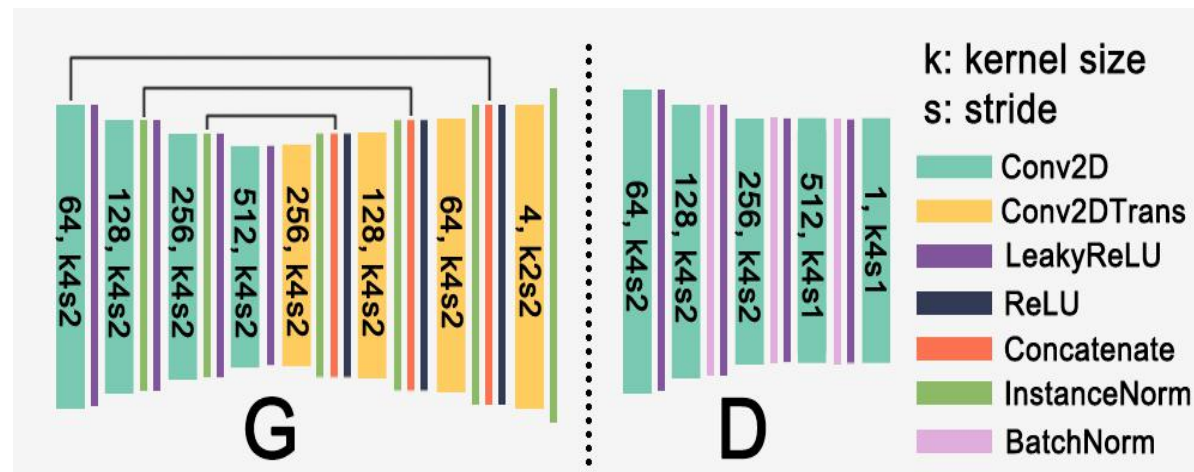
$$\mathcal{L}_{cyc}(G) = \mathbb{E}_{x_i, y_i, y_j \sim p_{\text{data}}} \|x_i - G(G(x_i, y_i, y_j), y_j, y_i)\|$$

- $\mathcal{L}_{cGAN}$ 前两项：传统的GAN Loss, 针对真实的模特-衣服组合和生成的模特-衣服组合。
- 第三项：cGAN项，辨别“衣服不是模特身上穿的” (implicit)
- 要求mask尽量小 (从而只改变模特图的衣服部分)，L1 Loss
- Cycle loss，让生成图的模特穿回原来衣服，求和原始图的L1 loss



# 代码细节

- 总体结构和pipeline, 包括训练/测试/数据读取, 基于开源的CycleGAN
- 复现CAGAN的更改部分:
  - U-Net中, 把原图的池化降采样结果 concatenate在Encoder的各层;
  - D net几乎相同, conv padding 和 normalization 方式区别 (cycleGAN IN; CAGAN BN)
  - Loss 计算
  - Triplet dataloader (模特+衣服1+衣服2)



```
def forward(self, input):  
    xi = input[:,0:3,:,:]  
    yi = input[:,3:6,:,:]  
    xi_yi = torch.cat([xi, yi], dim=1)  
    xi_yi_sz64 = self.pool2(xi_yi)  
    xi_yi_sz32 = self.pool4(xi_yi)  
    xi_yi_sz16 = self.pool8(xi_yi)  
    xi_yi_sz8 = self.pool16(xi_yi)  
  
    # Encoder  
    l1 = self.leaky_relu(self.conv1(input))  
    l1 = torch.cat([l1, xi_yi_sz64], dim=1)  
    l2 = self.norm2(self.conv2(l1))  
    l3 = self.leaky_relu(l2)  
    l3 = torch.cat([l3, xi_yi_sz32], dim=1)  
    l3 = self.norm3(self.conv3(l3))  
    l4 = self.leaky_relu(l3)  
    l4 = torch.cat([l4, xi_yi_sz16], dim=1)  
    l4 = self.leaky_relu(self.norm4(self.conv4(l4)))  
    l4 = torch.cat([l4, xi_yi_sz8], dim=1)  
  
    # Decoder  
    l5 = self.upconv5(l4)  
    l5 = self.norm5(l5)  
    l5 = torch.cat([l5, l3], dim=1)  
    l5 = self.relu(l5) # [8, 512, 16, 12]
```

U-Net 前向部分  
(CAGAN修改版)

上图: CAGAN 的  
G,D 具体实现

左图: Unet中原图  
前6 channel  
avgpooling后, 和  
encoder每层  
leakyReLU输出拼接

# 实验（部分结果分析）

- 原文的效果：



$y_i$

$x_i$

$x_i^j$

$y_j$



# 实验（部分结果分析）

- 复现效果：



# 实验（部分结果分析）

- 复现效果：



# 实验 (difficult cases)

- 衣服款式差别大:



- 长短袖



- “Neck artifacts”

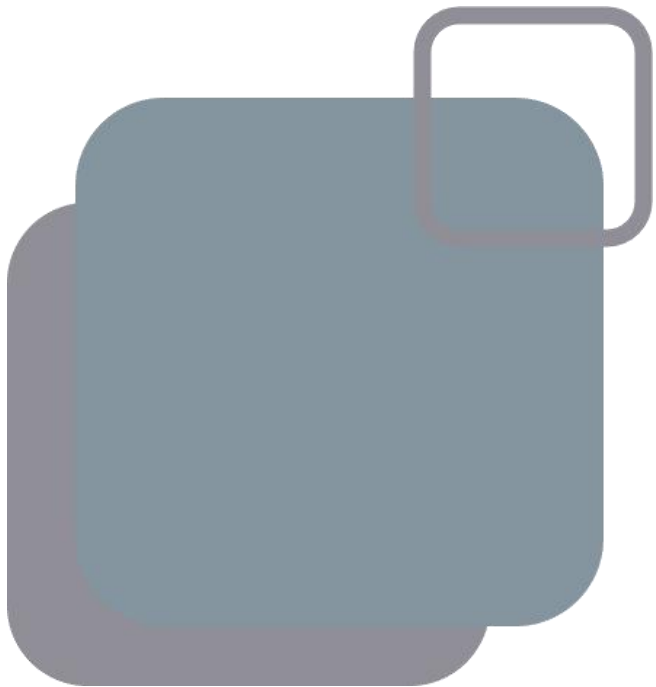


- 衣服的图样细节迁移



# 结果分析及提升思路

- 大致实现了CAGAN原文提出的方法
  - U-Net提取衣服mask的效果不如原文，原因仍然有待考察，或许可以对mask采用更复杂的loss形式；
- 输出的细节比较粗糙（原文的细节也不好）
  - 可以考虑提高生成器复杂度，比如 StackGAN；提高输入分辨率
- 如果作更大的改动，可以考虑将mask的生成作为单独的分割任务，让Generator专注于“穿衣服”。这也是更新的fashion工作中常见的，比如 VITON、SwapGAN等会考虑更精细的语义分割甚至加入pose的提取环节。



# Demo Q & A

我的代码:

<https://www.github.com/pkglimmer/>

