

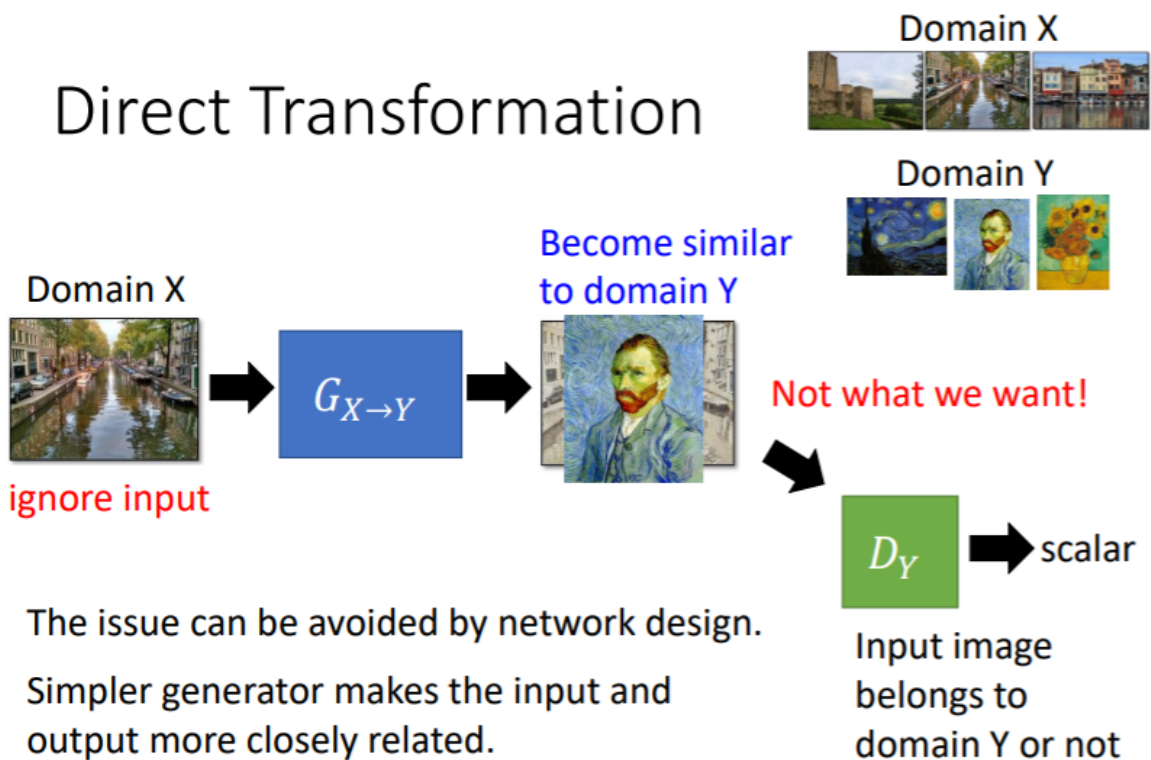
# Unsupervised Conditional Generation

两类做法：1.直接转换，输入和输出不会差别太大（比如颜色、纹理的转换）；2.投影到公共空间，输入和输出的差距可以很大（比如风格转换）

## Direct Transformation

在直接转换方法当中，判别器用来检测输入图片（生成器的输出图片）是否属于Domain Y的图片。但这就存在问题：如何判断生成器的输出与输入是否相关。

解决方法1：无视这个issue,因为生成器更倾向于对input做较少的改动，让输出尽可能接近输入。越简单的生成器越倾向于使输出和输入相关，如果是深层网络，则需要考虑加入其他约束。

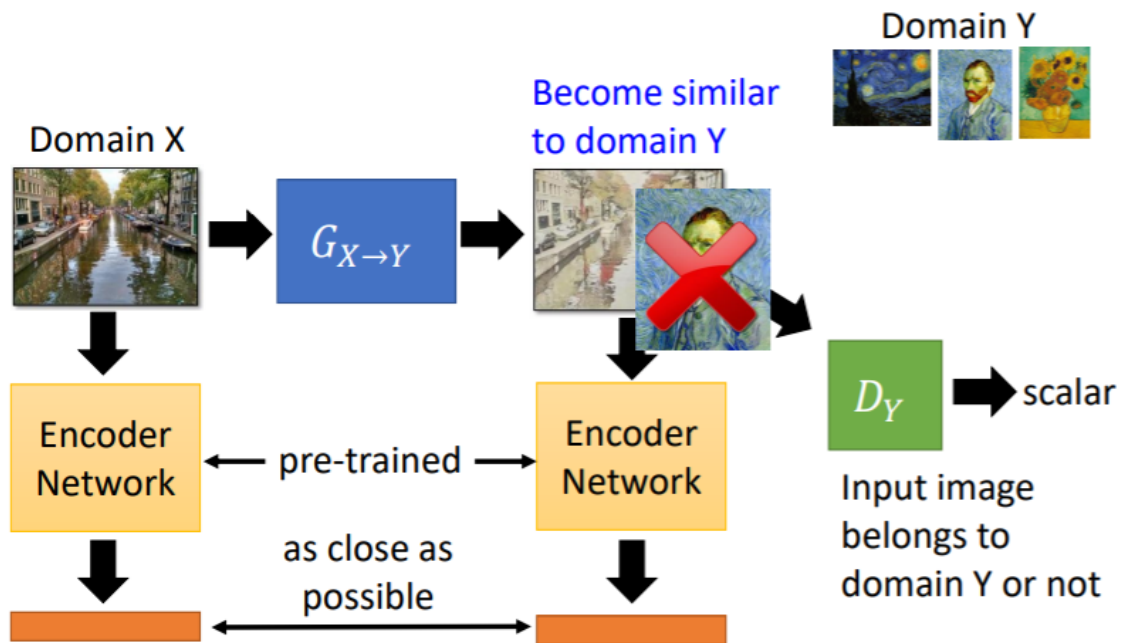


The issue can be avoided by network design.

Simpler generator makes the input and output more closely related.

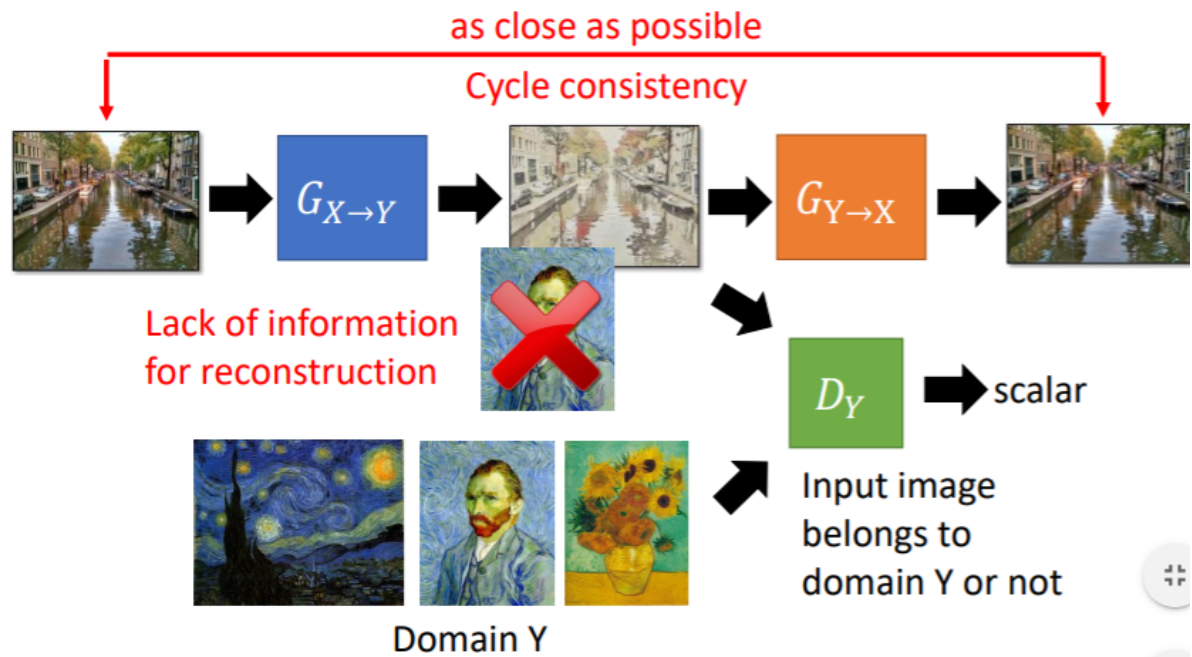
[Tomer Galanti, et al. ICLR, 2018]

解决方法2：将生成器的输入和输出都输入到一个pre-trained网络，比较该网络输出的embedding的差别，并且将其与“骗过”判别器一同作为生成器的训练目标。

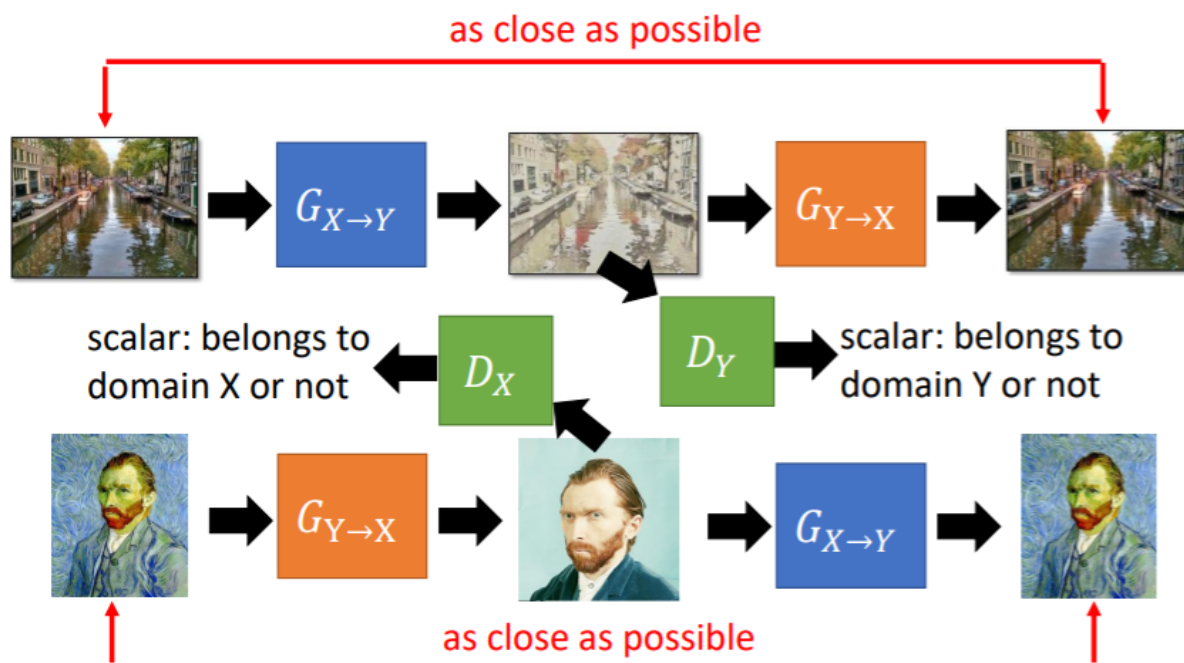


Baseline of DTN [Yaniv Taigman, et al., ICLR, 2017]

解决方法3: Cycle GAN,将生成器的输出输入到另外一个生成器, 将生成的图片与原始输入进行比较。如果中间结果与预期结果差异很大的话, 相关的信息就无法被保留, 那么通过第二个生成器得到的图片就会与原始输入有较大的差别。



Cycle GAN可以双向进行。

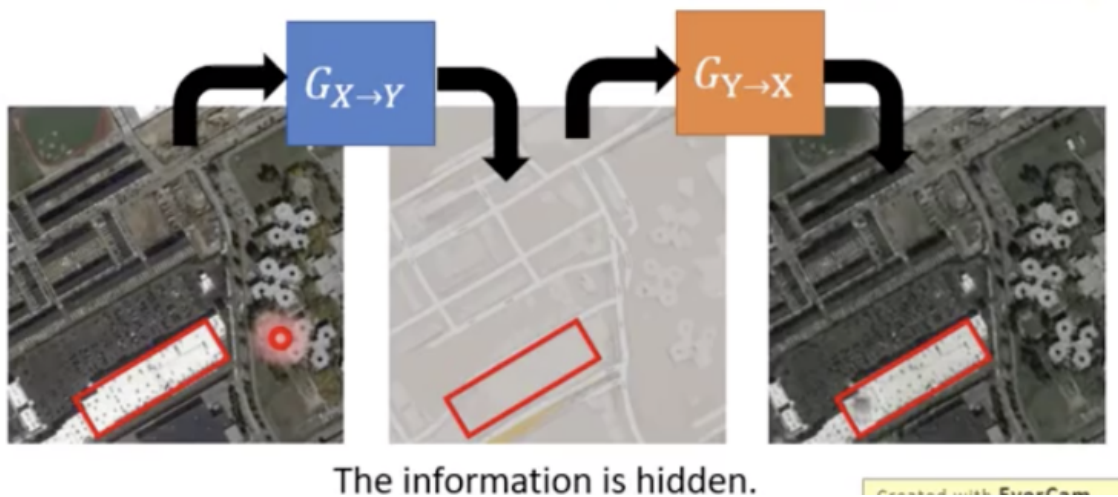


### *Issue of Cycle Consistency*

隐写术：生成器有将原图信息隐藏起来的能力，然后自己恢复。比如下图的红框内，Domain Y的图片是没有黑点的，但是reconstruct之后又恢复了框里的黑点。这样Cycle GAN就失去作用了。

### • CycleGAN: a Master of Steganography (隱寫術)

[Casey Chu, et al., NIPS workshop, 2017]



### *StarGAN*

用于多个Domain之间互相转换。

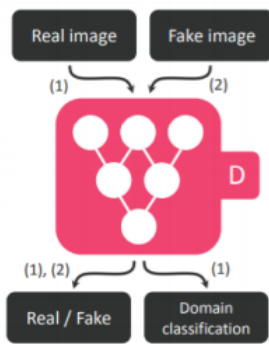
(a) Cross-domain models



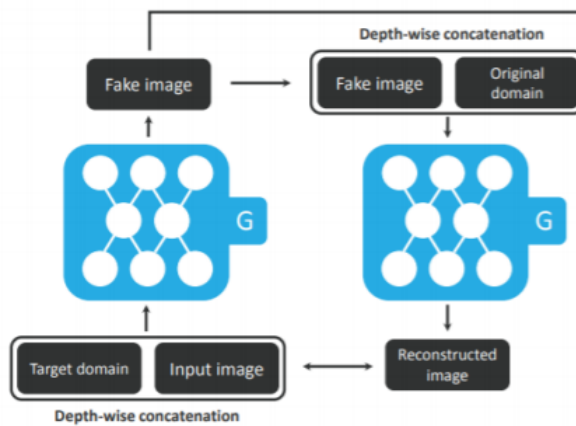
(b) StarGAN



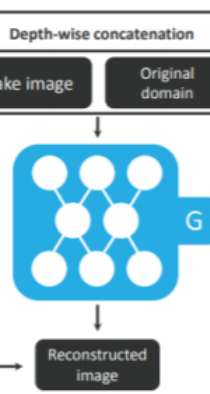
(a) Training the discriminator



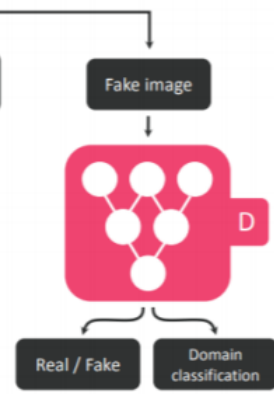
(b) Original-to-target domain



(c) Target-to-original domain



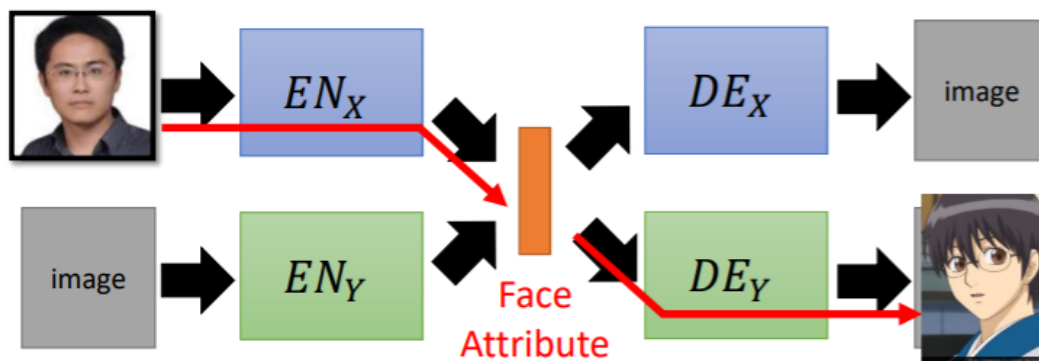
(d) Fooling the discriminator



## Projection to Common Space

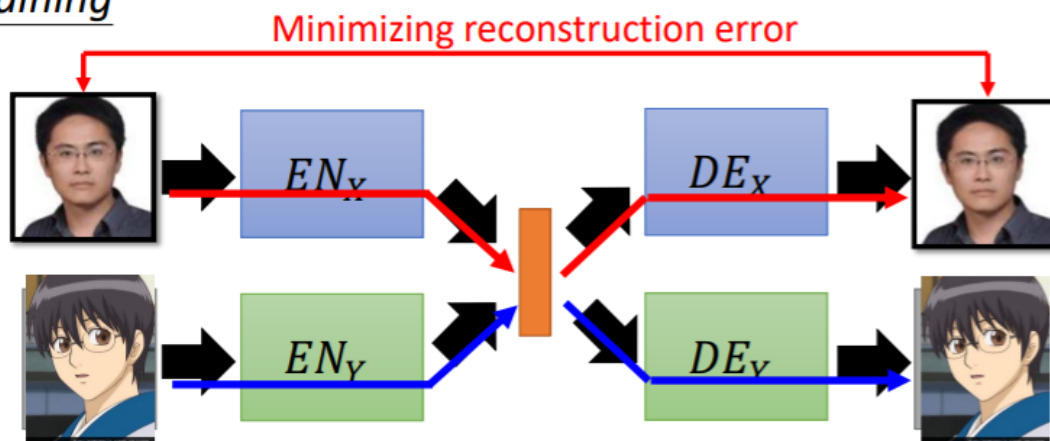
假设有Domain X, Domain Y, 分别存在两个Encoder和两个Decoder, 分别对应不同的Domain。最终的目标是输入真实的图片 (Domain X), 通过ENX得到Representation, 然后输入到DEY, 得到转换后的图片 (Domain Y)。

## Target

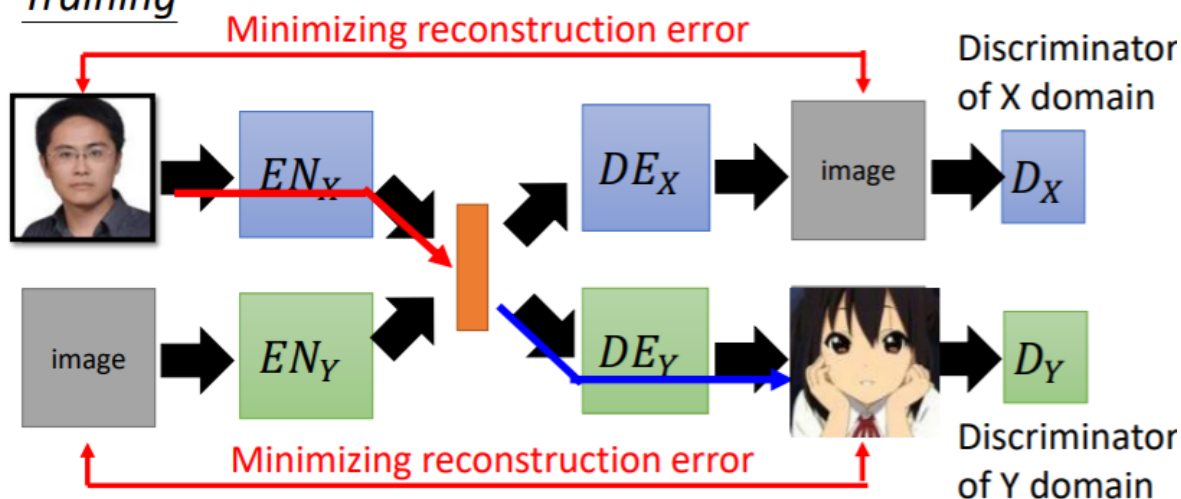


但是无监督学习的条件下，训练这些Encoder和Decoder的方法是将Encoder和Decoder组合成一个Auto-Encoder,然后最小化Reconstruction Error.但是出现问题：这两个VAE-GAN分开训练，之间是没有关联的（得到的Latent Representation不一样）。

## Training



## Training

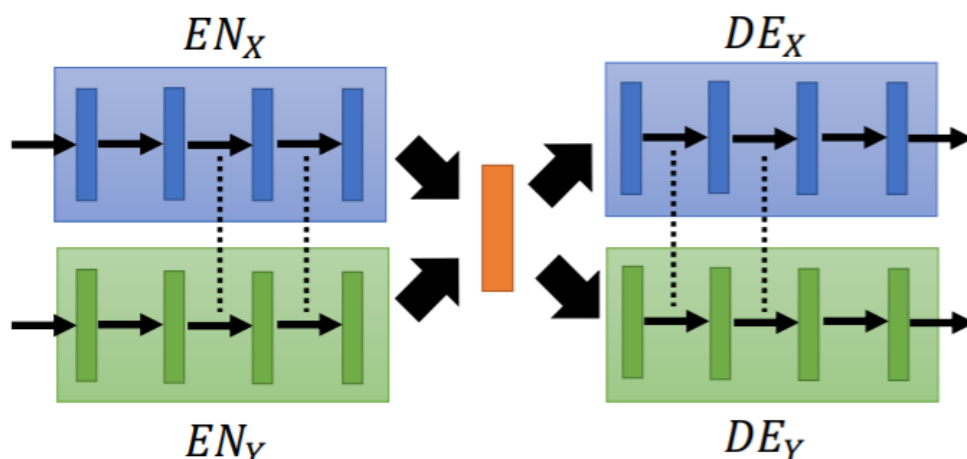


Because we train two auto-encoders separately ...

The images with the same attribute may not project to the same position in the latent space.



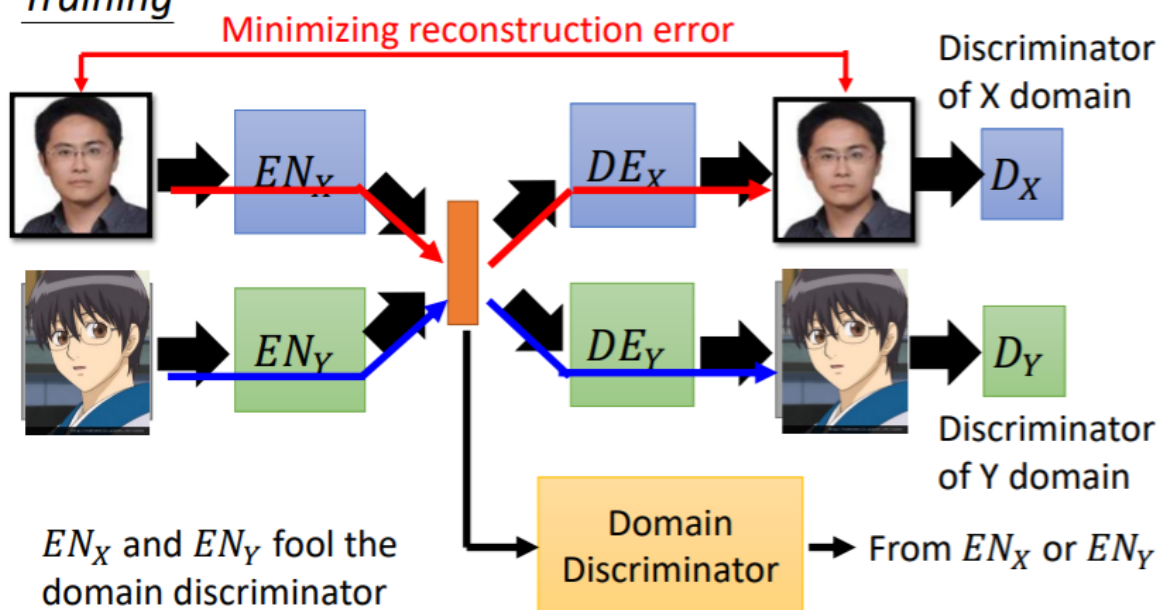
解决方法1：两组encoder和decoder的最后（最前）几层hidden layer的参数是共用的。这样的好处是也许会让两个Encoder的得到的Latent Space是一样的。最极端的情况是Encoder只有一个，输入时附带一个flag来表示是哪个Domain的输入。



## Sharing the parameters of encoders and decoders

解决方法2：引入一个Domain discriminator，强迫两个domain同一维表示同样的东西。如果Domain discriminator无法判断输入是来自于Domain X还是Domain Y,那么就说明他们的Latent space是一样的。

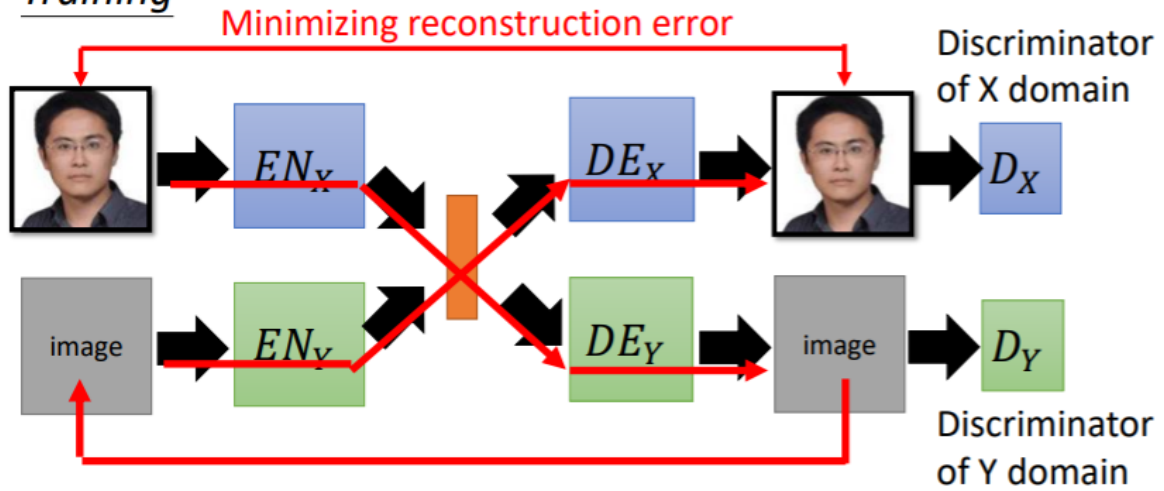
### Training



The domain discriminator forces the output of  $EN_X$  and  $EN_Y$  have the same distribution. [Guillaume Lample, et al., NIPS, 2017]

解决方法3：Cycle Consistency.

### Training



解决方法4: Semantic Consistency

### Training

