GAN与图像生成

介绍生成模型：

- p1定义和分类
- p2举例

介绍GAN

- p3介绍
- p4算法图
- p5理解
- p6问题

介绍GAN的发展

- 损失函数
    - p7 原始GAN
    - p8 fGAN和LSGAN
    - p9 WGAN和WGAN-GP
    - p10 历史
    - p11 效果图
- 模型结构
    - p12分类综述
    - p13直接
    - p14分层
    - p15迭代
    - p16其他
- 介绍GAN在问题上——风格迁移
    - 简述
    - CGAN
    -

介绍优缺点和未来方向

生成模型

定义：

希望用一个概率模型来描述所给的数据分布。不仅具有判定型模型判定的能力（如分类），因为具有数据的分布模型，所以还可以用来生成新的数据。

分类：

生成模型可以分为两大类，**显式密度模型**和**隐式密度模型**，这两个类别又可以分成很多子类。

显式模型的思路是我们首先给数据建一个模型，里面有一个参数，要做的问题就是想办法估计这个参数，其中最常见的方法是极大似然估计。但是由于模型的复杂性，直接优化目标函数计算参数也不是一件简单的事。针对这个问题就产生了许多不同的方法

- 建立容易直接处理的模型：

    - **FVBN**：比如**PixelRNN/CNN**[2016], **WaveNet**[2016]。对图像数据的概率分布p进行显式建模，并利用极大似然估计优化模型。FVBNs的主要缺点是每一次计算只能生成一个条目（比如一个像素点），计算量大，且这些步骤不能并行计算。
    - **流模型**：**NICE**[2014], **RealNVP**[2016], **Glow**[2018]。目前这方面关注比较少。
    - ……
- **对不易处理的密度函数采用近似模型**：比如**VAE**，不直接最优化目标，而是优化一个目标的一个上界。代价是不够精准，生成的图像比较模糊。

隐式密度估计模型，不需要定义显式的密度函数。最经典的例子就是**GAN**，它用G和C的对抗避开这个问题，直接训练模型。GAN类方法是目前图像生成领域的主流方法。

GAN介绍

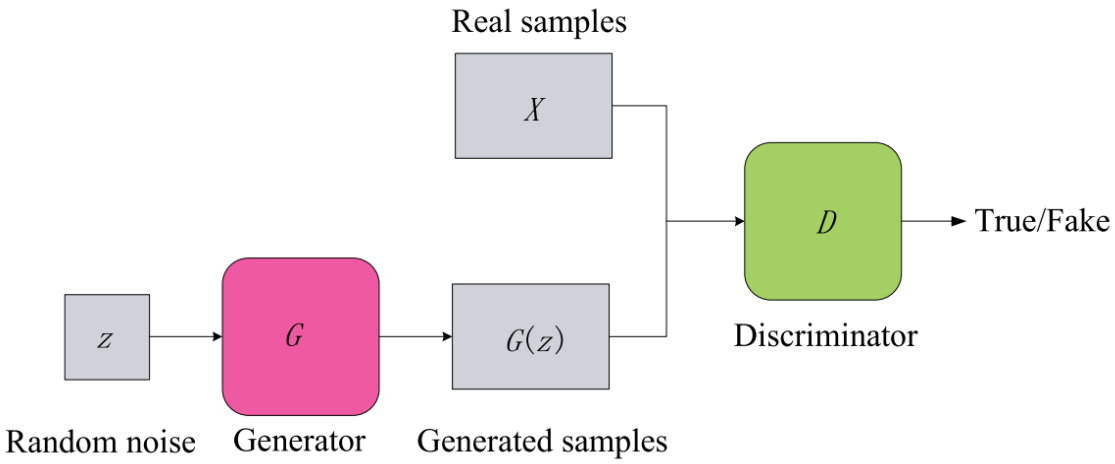


FIGURE 1. The general structure of a generative adversarial network.

GAN的核心部件有两个：**生成器G**和**判别器D**。生成器G用于从随机噪音z（后面也有其他变体）生成一张图像G(z)，它的目的是让生成的图像G(z)尽可能接近真实数据X，骗过判别器D的眼睛；判别器D的目的就是辨别收到的图像是来自生成器的假图像G(z)还是来自真实数据X。生成器G和判别器D互相竞争，循环交替训练，直至最终收敛到一个纳什均衡。这时候我们得到的生成器G就可以用于完成图像生成任务。

在数学上，这相当于求解一个minimax问题，公式如下：

$$\min_G \max_D V(D,G) = \mathrm{E}_{x\sim Pdata(x)}[\log D(x)] + \mathrm{E}_{z\sim Pz(z)}[\log(1-D(G(z)))] \quad (1)$$

注意到V(D,G)就是二分类问题的交叉熵损失函数。因此这个损失函数V是很符合直觉的想法。

Algorithm 1 GAN Algorithm

Input: 随机噪声$\{\mathbf{z}_1,\ldots,\mathbf{z}_m\}$ in $\mathbb{R}^d$; 真实样本$\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}\subset\mathcal{X}$.

Output:生成样本X_{fake}.

1: **for** $t=0$ to $T-1$ **do**

2: 从高斯噪声分布γ中随机采样出m个样本，即$\{\mathbf{z}_1,\ldots,\mathbf{z}_m\}$ in $\mathbb{R}^d$;

3: 从真实数据分布$\mathcal{X}$中随机采样出m个样本，即$\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}\subset\mathcal{X}$;

4: 通过小批量随机梯度下降法来更新判别器D_ω的参数，具体公式为：

$$\nabla_\omega\frac{1}{m}\sum_{i=1}^{m}\left[\log D_\omega\left(\mathbf{x}_i\right)+\log\left(1-D_\omega\left(G_\theta\left(\mathbf{z}_i\right)\right)\right)\right]$$

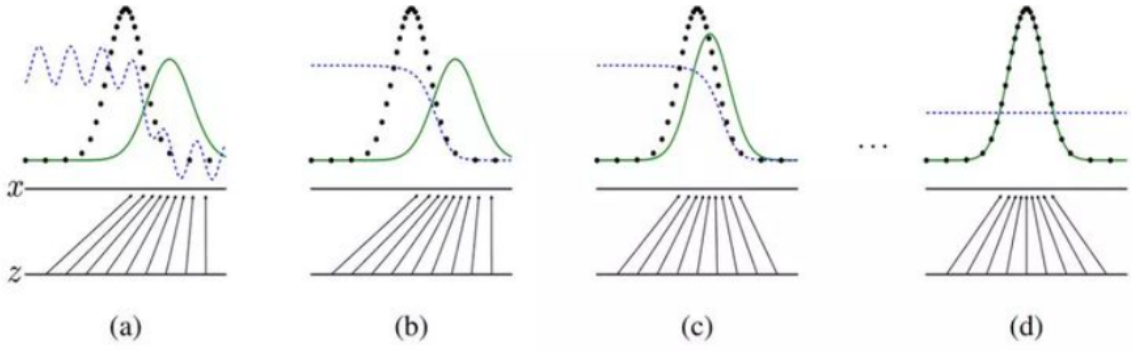
5: 再从高斯噪声分布γ中随机采样出另外的m个样本，即$\{\mathbf{z}_1,\ldots,\mathbf{z}_m\}$ in $\mathbb{R}^d$;

6: 通过小批量随机梯度下降法来更新判别器G_θ的参数，具体公式为：

$$\nabla_\theta\frac{1}{m}\sum_{i=1}^{m}\log\left(1-D_\omega\left(G_\theta\left(\mathbf{z}_i\right)\right)\right)$$

7: **return** X_{fake}

GAN的训练过程从数学角度理解。样本数据集有一个分布（对每一个像素建模），生成器生成的数据也会有一个分布（噪声随机变量与函数G的复合），判别器区别是否是真样本，就是要学习找出两个分布的差异点；而生成器的训练过程就是不断去减小分布的差异，直到最后生成与真实数据一样的分布，判别器再也无法区分为止。



GAN的问题

GAN在生成图像的质量上比起VAE更好，但是原始GAN模型本身也存在一些问题，主要的问题有两个：

（1）训练不稳定：虽然 GAN 在图像生成方面非常有效，但它的训练过程非常不稳定，需要很多技巧才能获得良好的结果。比如梯度消失问题，判别器越好，生成器的梯度消失越严重，这样会导致在网络训练上很多时候生成器的参数基本上不会发生改变，无法训练。

（2）模式崩溃（collapse mode）：这是指生成器开始反复产生相同的输出（或一小组输出）的现象。因为判别器不需要考虑生成样品的种类，而只关注于确定每个样品是否真实，这使得生成器只需要生成少数高质量的图像就足以愚弄判别者。

例如在 MNIST 数据集包含从 0 到 9 的数字图像，但在极端情况下，生成器只需要学会完美地生成十个数字中的一个以完全欺骗判别器，然后生成器停止尝试生成其他九位数，缺少其他九位数是**类间模式崩溃**的一个例子。**类内模式崩溃**的一个例子是，每个数字有很多写作风格，但是生成器只学习为每个数字生成一个完美的样本，以成功地欺骗鉴别器。

GAN的发展

对于损失函数：

- **原始GAN[2014]**

  在信息论中，用KL 散度（Kullback-Leibler divergence）来衡量两个分布的差异

$$KL(p_1\|p_2) = \mathbb{E}_{\mathbf{x}\sim p_1}\log\frac{p_1}{p_2},$$

在离散型变量的情况下，KL散度衡量的是，当我们使用一种被设计成能够使得概率分布 Q 产生的消息的长度最小的编码，发送包含由概率分布 P 产生的符号的消息时，所需要的额外信息量。

因为 KL 散度是非负的并且衡量的是两个分布之间的差异，所以它经常被用作衡量分布之间的某种距离。然而因为它不是对称的，所以它并不是数学意义上的距离。为了解决这一个问题，让p1p2的选择不成为一个问题，就有了JS散度

$$JS(p_1\|p_2) = \frac{1}{2}KL(p_1\|\frac{p_1+p_2}{2}) + \frac{1}{2}KL(p_2\|\frac{p_1+p_2}{2}).$$

原始的GAN的损失函数，在判别器D达到最优的时候，对生成器G就相当于优化JS散度。也就是说原始的GAN是以JS散度为衡量标准去让生成分布逼近样本分布的。

- **fGAN**

  f散度：

$$D_f(p\|q) := \mathbb{E}_{\mathbf{x}\sim q}\left[f\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right)\right] = \int_{\mathbb{R}^n} f\left(\frac{p(x)}{q(x)}\right)q(x)dx$$

  这个散度推广了KL散度和JS散度，通过对f取不同的函数，我们得到了更多可行的损失函数。（常见的f与loss列表见论文）并且作者还给出了利用共轭函数将f散度转变成可以计算的形式的方法。

- **LSGAN**

LSGAN使用的loss是

$$\min_D J(D) = \min_D[\frac{1}{2}\mathbf{E}_{x\sim p_{data}(x)}[D(x)-a]^2 + \frac{1}{2}\mathbf{E}_{z\sim p_z(z)}[D(G(z))-b]^2]$$

$$\text{原作中取 } a=c=1, b=0 \text{ 。}$$

它也属于一种f散度。LSGAN使用最小二乘损失函数代替原始GAN的损失函数的优点是：

1. 训练更稳定：**对离群样本惩罚更大**，比原始GAN训练更加充分、稳定
2. 改善生成质量：对离群样本惩罚更大，使得生成图片更接近真实数据、更清晰

缺点是：

1. 对离离群点的过度惩罚, 可能导致样本生成的”多样性”降低
2. 仍然是基于“散度”的loss，无法根本上解决梯度消失问题

- **WGAN**

  前面几种基本上都可以说是小修小补，WGAN则是一个非常重要的理论突破。

  Wasserstein距离/EM距离：

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\left[\|x - y\| \right]$$

W距离又称之为推土机距离，它的起源是optimal transport problem。直观理解我们可以把概率分布想象成一堆土，真实图像的分布和生成图像的分布就是两堆形状不同的土。W距离就相当于把第二堆土移动变成第一堆土的最小消耗。

梯度消失问题的本质原因，是因为对于两个分布来说，如果完全不重合（互相没有任何信息），那么它们之间的散度就是固定的（=log2）。所以使用原始的基于KL散度的各种loss函数时，梯度就会为0。且判别器训练得越好，那么就越可能会梯度消失，生成器就无法学习。非常矛盾。

Wasserstein距离完全解决了梯度消失问题。WGAN与原始GAN相比，参数更加不敏感，训练过程更加平滑。虽然它在理论上避免了模式崩溃，但模型收敛的时间比以前的 GAN 要长。

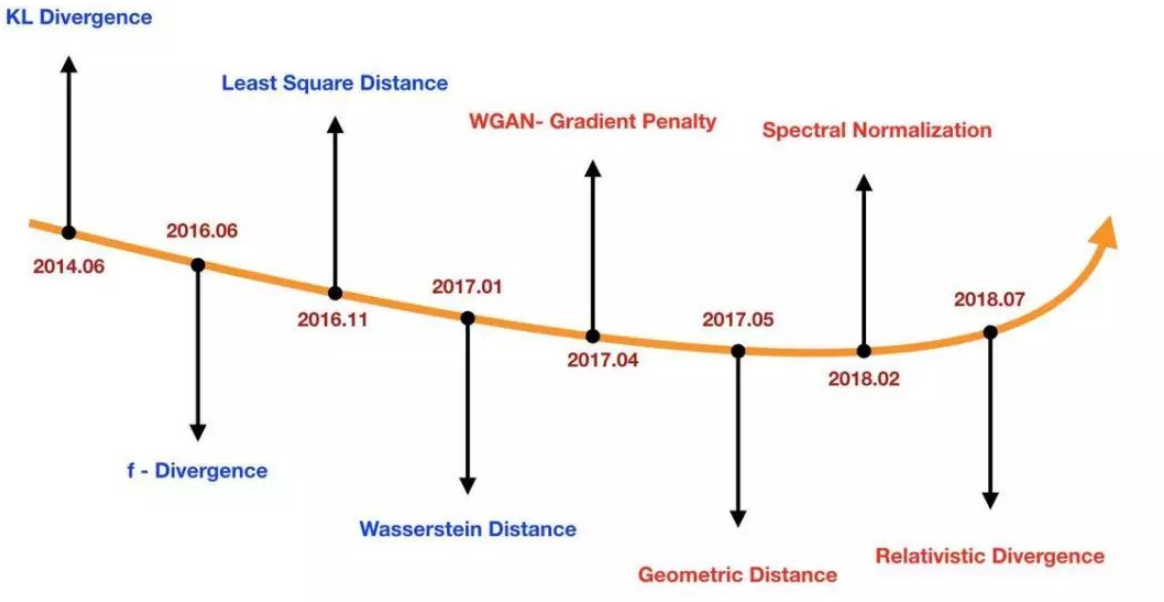
- **WGAN-GP**

  对WGAN的一些小改进。

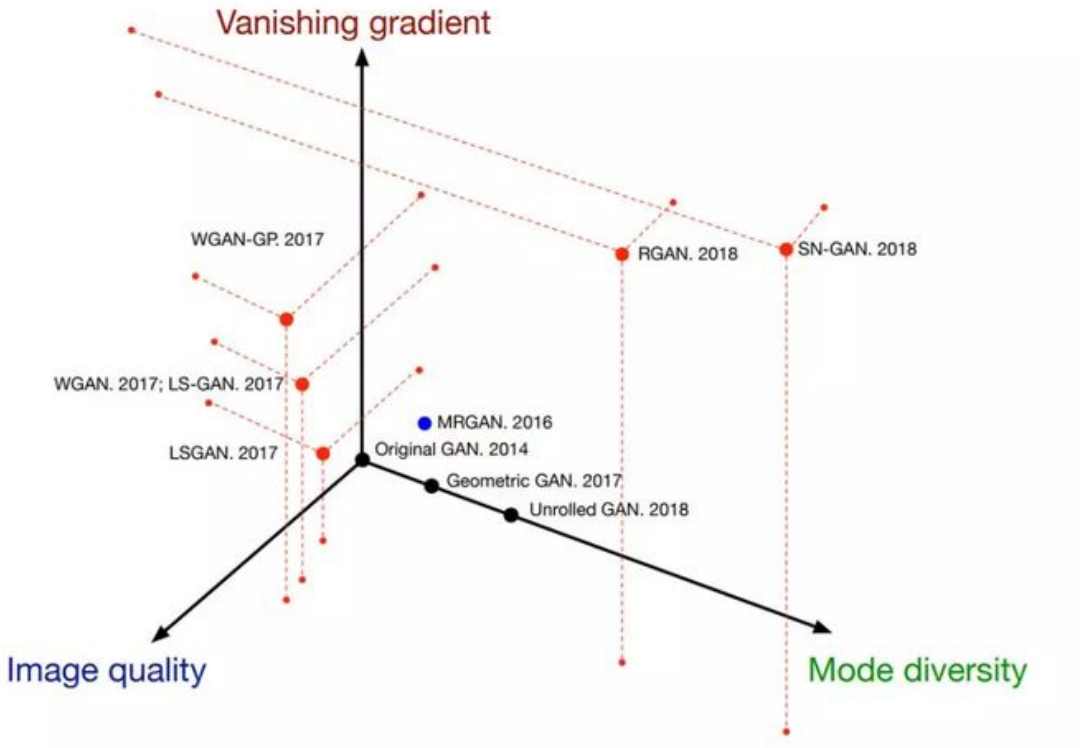
  将WGAN中的梯度截断改为梯度惩罚，使得GAN训练更加稳定，收敛更快，同时能够生成更高质量的样本。

  WGAN-GP 通常可以产生良好的图像并极大地避免模式崩溃，并且很容易将此培训框架应用于其他 GAN 模型。

这是GAN在损失函数上的发展轨迹：



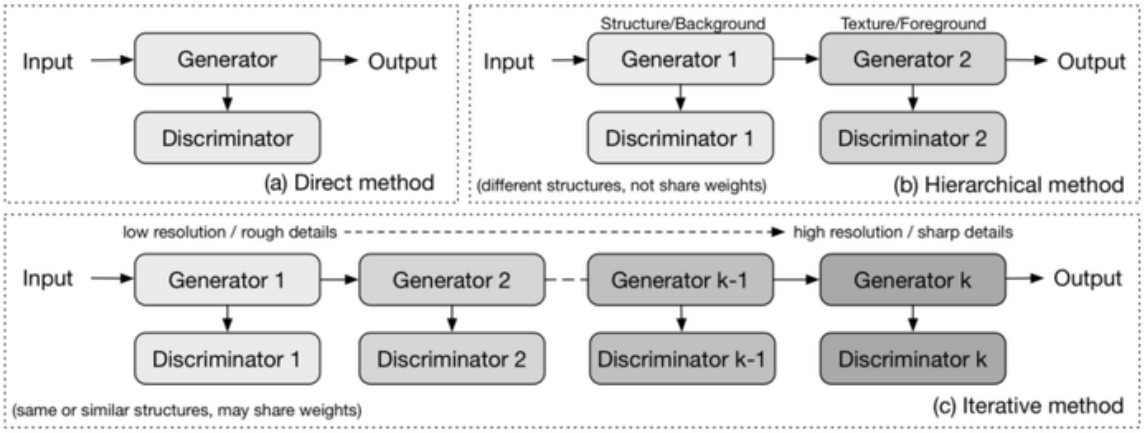
不同的损失函数主要从**梯度消失**问题、**图像质量**和**样本多样性**三个方面提升 GAN 的效果。不同的度量方法也会有其独特的角度，例如 WGAN-GP 就特别关注梯度消失问题与图像质量，Spectral normalization GAN 对于处理梯度消失与样本多样性是最高效的。下图是不同的损失函数在这3个维度方面的表现：



模型架构

GAN的基本模型里没有限定生成器和判别器的数量和具体使用什么模型，而生成器与判别器的模型能力对GAN最后的效果至关重要。因此模型选择也经历了许多变化。

根据不同的GAN所拥有的生成器和判别器的数量，可以将GAN图像生成的方法概括为三类：**直接方法**，**迭代方法**和**分层方法**[17]。



- **（1）直接法**

  早期的GANs都遵循在其模型中使用一个生成器和一个判别器的原理，并且生成器和判别器的结构是直接的，没有分支。之前提到的原始GAN[1]、f-GAN [19]都属于这类方法。这类方法在设计和实现上比较容易，通常也能得到良好的效果。其中经典的方法DCGAN [9]

  - **DCGAN[2015]**

    在 14 年原版 GAN 中，Goodfellow 采用最简单的浅层全连接网络作为判别器与生成器的架构，它可以在 MNIST 等低分辨率的数据集上获得很好的效果。

既然全连接可行，那么很自然的想法是利用深度卷积网络加强它的能力，DCGAN[9]将CNN引入生成器和判别器，借助CNN更强的拟合与表达能力，使得参数量和训练效果更好，大大提高了生成图像的能力。

DCGAN为实际训练GANs中提供了很多的指导方向，具体有如下五点：

- 用跨步卷积替换生成器和判别器中的任何池化层。
- 在生成器和鉴别器中使用批处理规范化。
- 移除完全连接的隐藏层以获得更深层的架构。
- 除了使用Tanh的输出之外，在生成器中对所有层使用ReLU激活，。
- 在所有层的判别器中使用LeakyReLU激活。

需要注意的是判别器是从高维向量到低维向量是卷积过程，生成器是低维向量到高维向量是一个**反卷积**过程。（反卷积可以理解为卷积的逆运算，从被卷积后的结果还原卷积之前。但毕竟卷积不是个可逆映射，所以这个并不是严格意义的逆运算，也做不到100%的还原，但大概思想是这样）

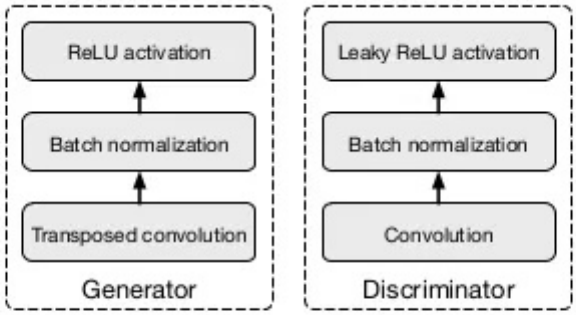
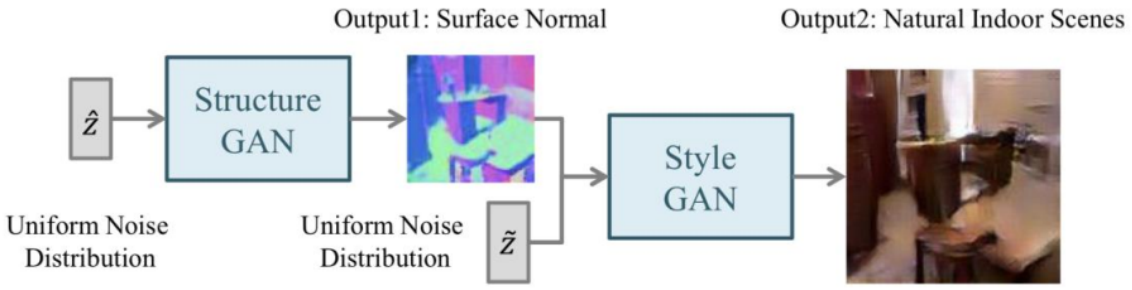


Fig. 6. Building blocks of DCGAN, where the generator uses transposed convolution, batch-normalization and ReLU activation, while the discriminator uses convolution, batch-normalization and LeakyReLU activation

（2）分层法

分层法的主要思想是将图像分成两部分，如“样式和结构”和“前景和背景”，然后在其模型中使用两个生成器和两个鉴别器，其中不同的生成器生成图像的不同部分，然后再结合起来。两个生成器之间的关系可以是并联的或串联的。

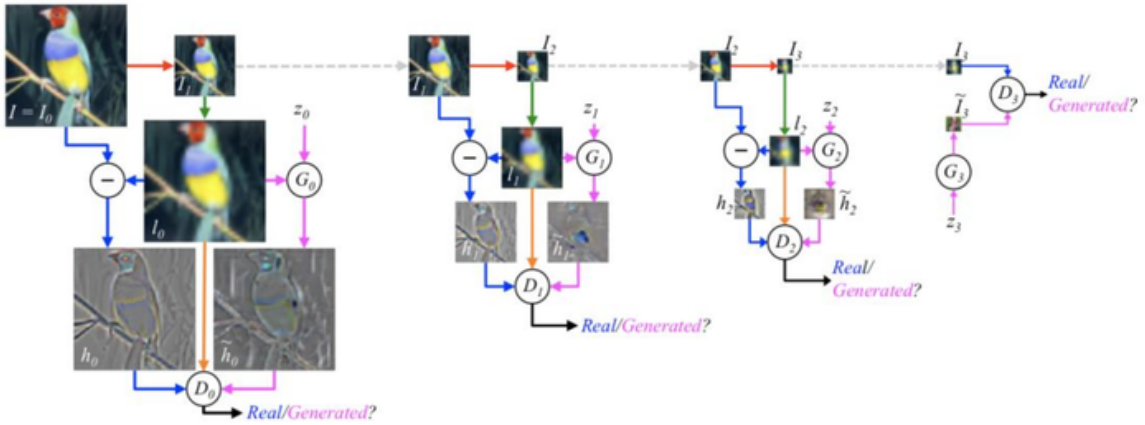
以SS-GAN [2018]为例，其使用两个GAN，一个Structure-GAN用于生成表面结构，然后再由Style-GAN补充图片细节，最后生成图片，整体结构如下所示：



（3）迭代法

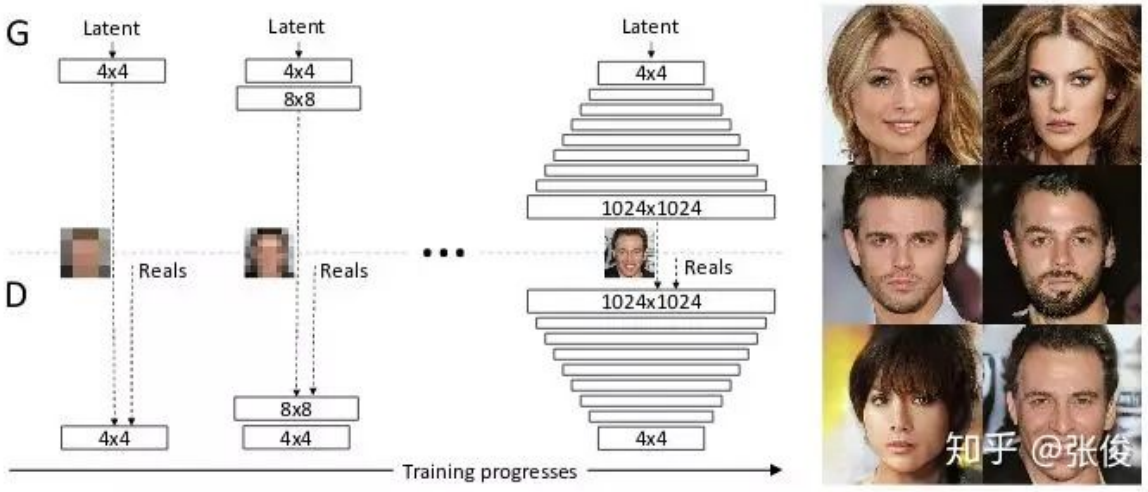
迭代法使用具有相似或甚至相同结构的多个生成器，经过迭代生成从粗到细的图像。

以LAPGAN [2015]为例： LAPGAN 利用卷积网络由低像素向高像素层级地生成图像。LAPGAN中的多个生成器执行相同的任务：最低级别的生成器仅将噪声向量作为输入并输出图像，而其他生成器都从前一个生成器获取图像并将噪声矢量作为输入，这些生成器结构的唯一区别在于输入/输出尺寸的大小，每一次迭代后的图像都拥有更多清晰的细节。

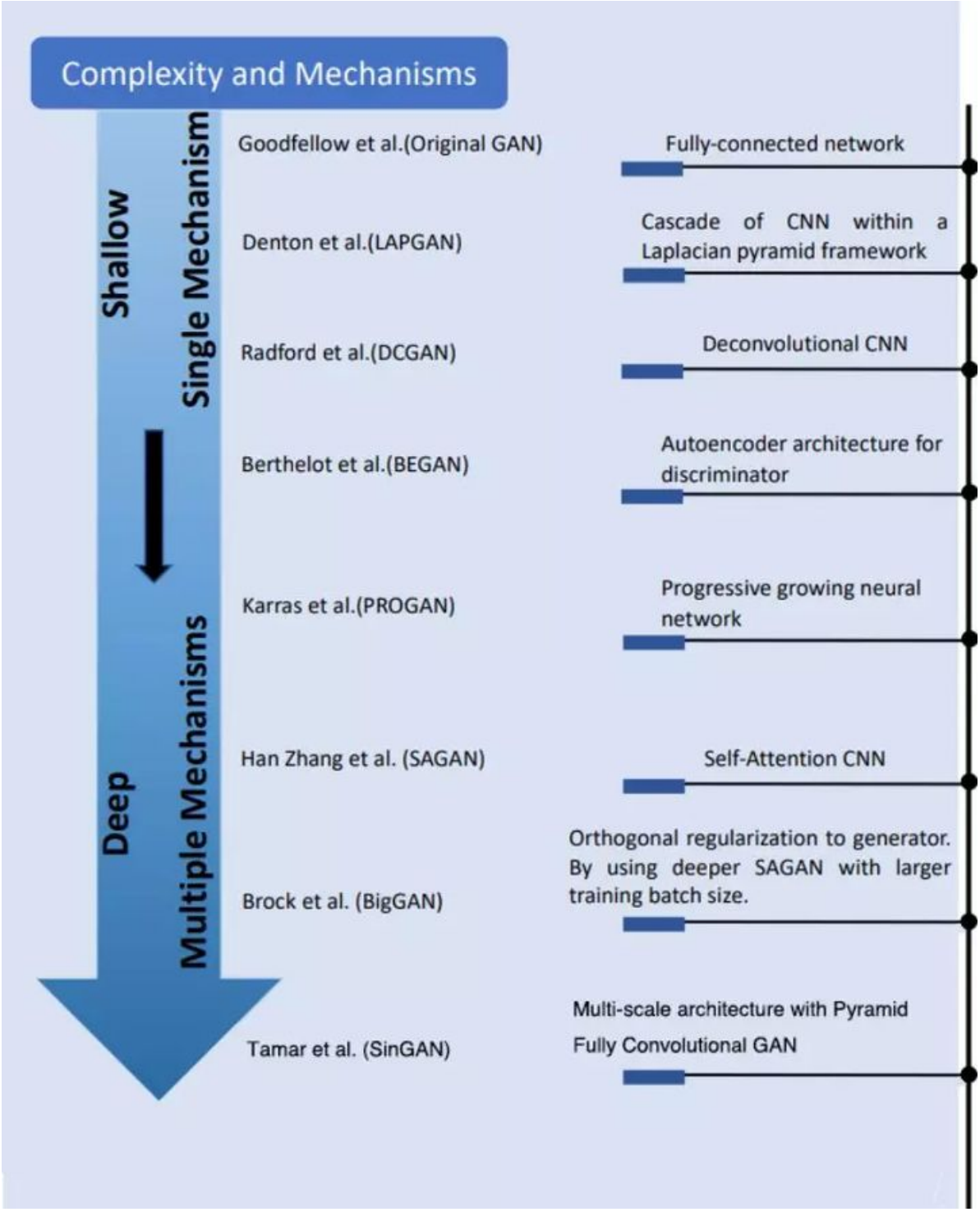


其它方法

为了生成更好的更高分辨率的图像，***ProgressiveGAN*** [2017] 建议首先训练 4×4 像素的生成器和判别器，然后逐渐给生成器和判别器增加额外的层，再次训练，一步步使输出分辨率加倍至 1024×1024。这种方法允许模型首先学习粗糙结构，然后专注于稍后重新定义细节，而不是必须同时处理不同规模的所有细节。



- 上面是按照类别进行的发展流程，如下图所示是根据时间和结构的复杂性的 GAN 的代表性架构演进：



GAN在风格迁移上

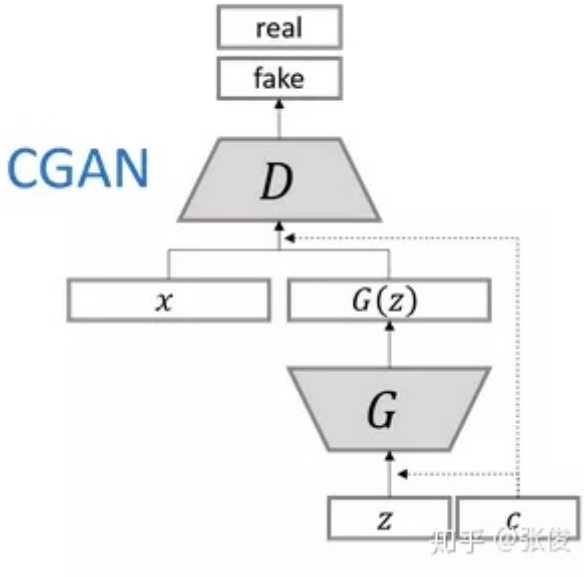
- ## GAN在图像到图像的应用

  图像到图像的转换被定义为将一个场景的可能表示转换成另一个场景的问题，例如图像结构图映射到 RGB 图像，或者反过来。该问题与风格迁移有关，其采用内容图像和样式图像并输出具有内容图像的内容和样式图像的样式的图像。

  图像到图像转换可以被视为风格迁移的概括，因为它不仅限于转移图像的风格，还可以操纵对象的属性（如在面部编辑的应用中）。

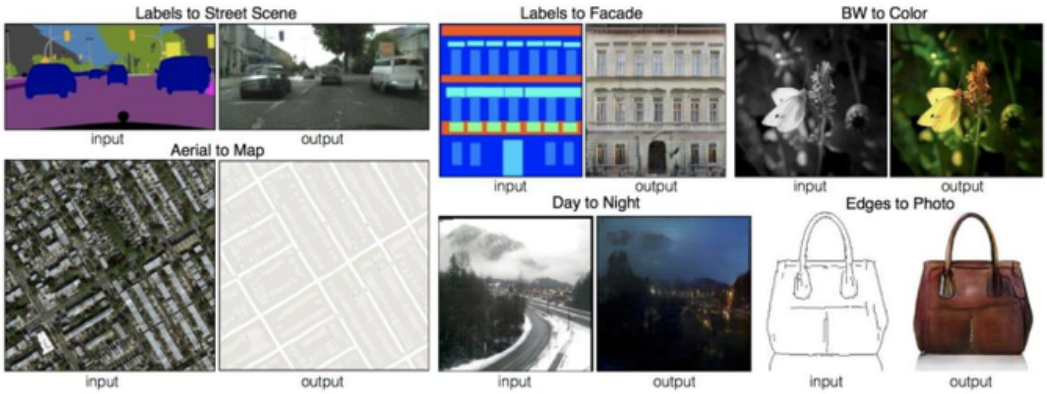
- ## CGAN[2014] / 条件GAN

  在原始 GAN 中，无法控制要生成的内容，因为输出仅依赖于随机噪声。我们可以将条件输入 c 添加到随机噪声 Z，以便生成的图像由 G(c,z) 定义。这就是 ***CGAN*** [6]，通常条件输入矢量 c 与噪声矢量 z 直接连接即可，并且将得到的矢量原样作为发生器的输入，就像它在原始 GAN 中一样。条件 c 可以是图像的类，对象的属性或嵌入想要生成的图像的文本描述，甚至是图片。



- ## Pix2Pix[2016]

  Pix2Pix [29] 在 CGAN 的损失函数上增加了 L1 正则化项，使得生成器不仅被训练以欺骗判别器而且还生成尽可能接近真实标注的图像，使用 L1 而不是 L2 的原因是 L1 产生较少的模糊图像。

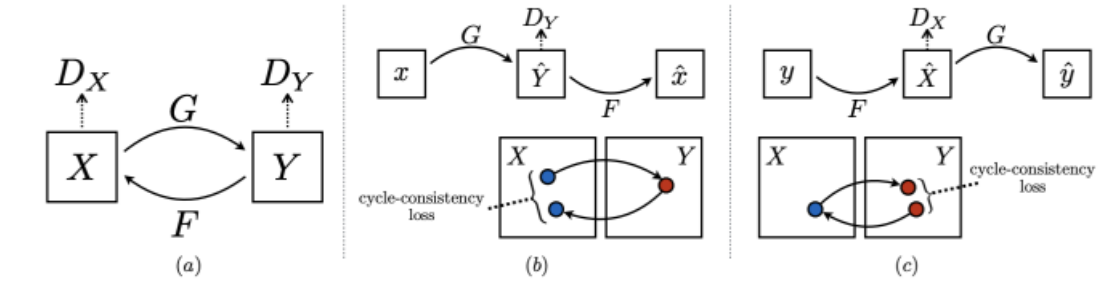


- CycleGAN [2017]、DualGAN [2017]和DiscoGAN [2017]

  以上两种方法都是有监督的。虽然有监督下图像转换可以得到很好的效果，但需要的条件信息以及paired image成为其很大的限制。但如果用无监督学习，学习到的网络可能会把相同的输入映射成不同的输出，这就意味着，我们输入任意 x_i 并不能得到想要的输出 y_i 。

  CycleGAN [24]、DualGAN [25] 和DiscoGAN [26]突破了这个限制，这几项工作都提出了一致/重构损失（consistent loss），采取了一个直观的思想：即生成的图像再用逆映射生成回去应该与输入的图像尽可能接近。在转换中使用两个生成器和两个判别器，两个生成器 G_{XY} 和 G_{YX} 进行相反的转换，试图在转换周期后保留输入图像。

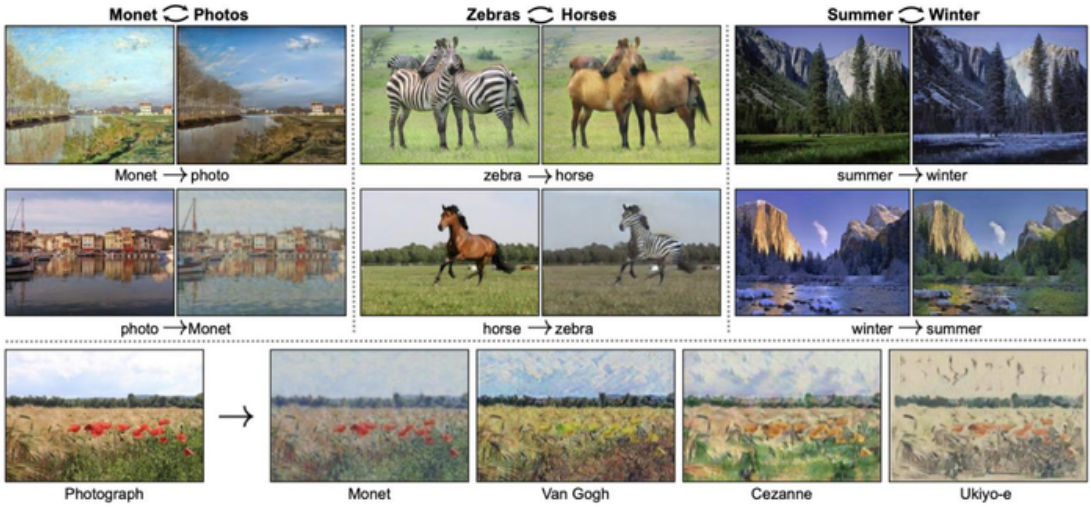
  以CycleGAN为例，在CycleGAN中，有两个生成器， G_{XY} 用于将图像从域X传输到Y，G_{XY} 用于执行相反的转换。此外，还有两个判别器D_X 和 D_Y 判断图像是否属于该域。



图九：cycleGAN结构

其Consistent loss由L1进行描述：

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}\left[\|F(G(x)) - x\|_1\right] + \mathbb{E}_{y \sim p_{\text{data}}(y)}\left[\|G(F(y)) - y\|_1\right]$$



图十：CycleGAN的生成效果

- CartoonGAN [2018]、AnimeGAN [2019]

  尽管基于学习的风格迁移已经获得了很大成功，但最先进的方法仍无法生产具有可接受质量的漫画风格图像。生成漫画风图像的难点有两个：

  （1）动画风格具有独特的特点，具有高度的简化和抽象性。

  （2）卡通图像往往具有清晰的边缘、平滑的着色阴影和相对简单的纹理，这对现有方 法中基于纹理描述符的损失函数提出了重大的挑战。

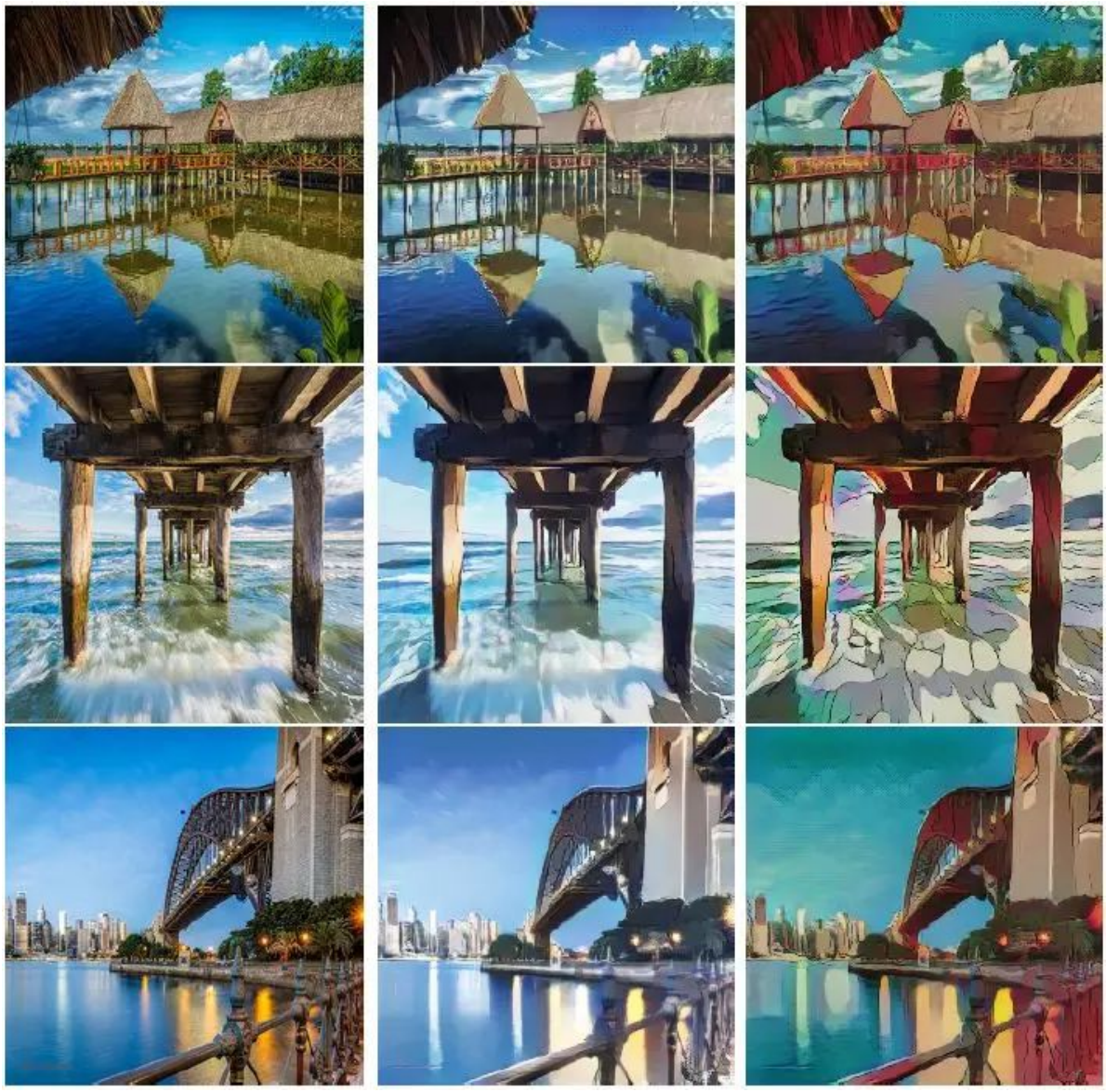
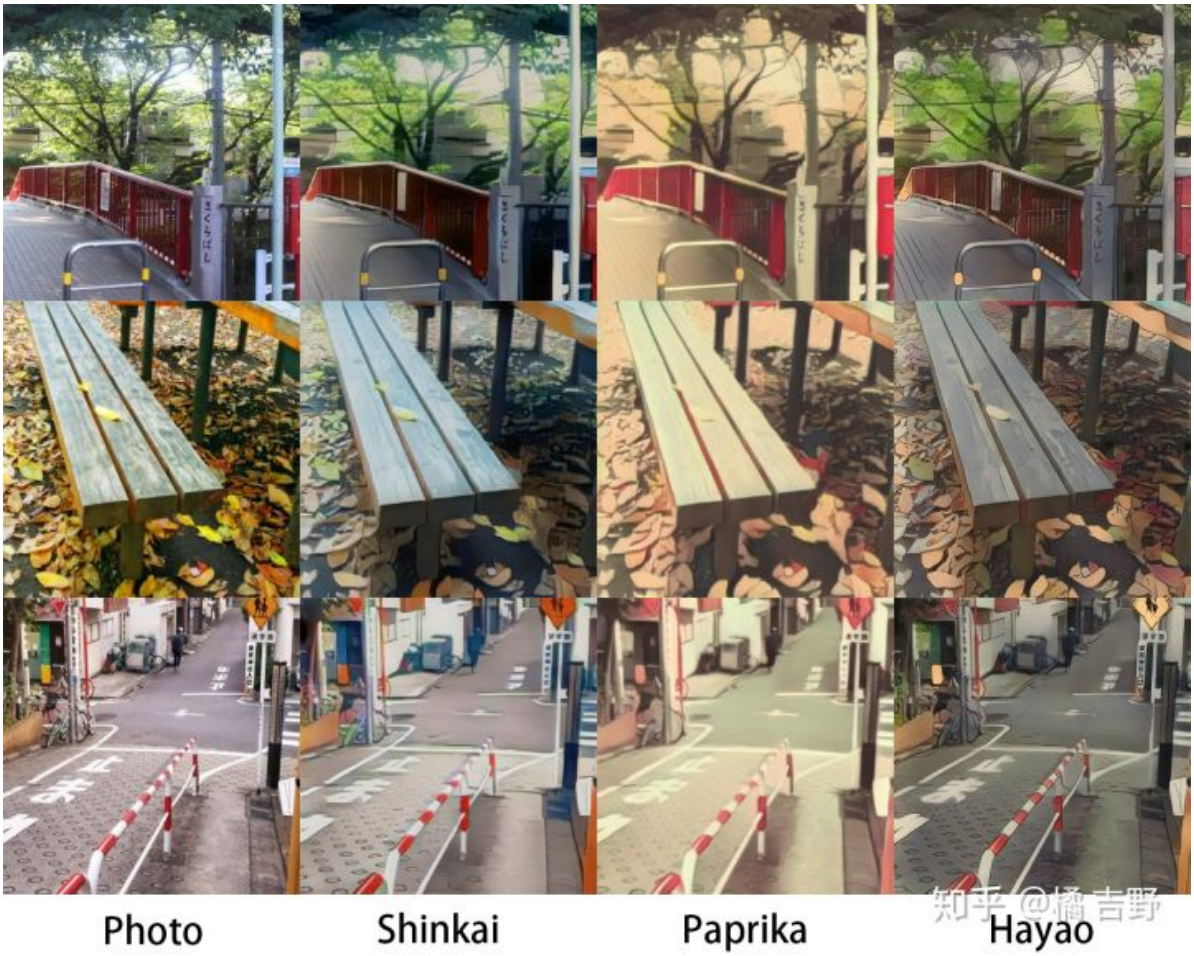


图 5. CartoonGAN 生成的不同艺术家漫画风格：（a）为输入照片。（b）为新海诚风格。（c）为宫崎骏风格。


Photo
CartoonGAN
ComixGAN
AnimeGAN



上图给出的是AnimeGAN在三个不同动漫风格中所得到的结果。这三个风格分别是：新海诚的《君の名は。》，金敏的《パプリカ》，宫崎骏的《風立ちぬ》。

事实上已经有游戏利用AI对现实图片进行风格迁移的方式制作立绘和CG，比如游戏《青箱》（使用了CartoonGAN）、《距离男主自杀还剩七天》（使用了VGG）等。

GAN的优缺点

总而言之， GAN的设计相对于很多其他显式生成模型有很多优点：

- 它可以并行产生样本， 相比于PixelCNN，PixelRNN这些模型，GAN生成样本非常快。
- 生成函数设计限制较少。 这一点是相对流模型等显式算法的优势。
- 不需要马尔可夫链。 这一点比玻尔兹曼机和生成随机网络有优势。
- GAN不需要通过引入下界来近似似然函数。这一点是针对VAE的优势。
- GAN通常被认为比其他方法可以产生更好的样本。

缺点：

- 训练不稳定，容易崩溃。这个问题有学者提出了许多解决方案，比如WGAN，LSGAN等
- 模式崩溃。尽管有很多相关的研究，但这个问题依然还没完全解决。

OPEN QUESTIONS与未来的研究方向

理论方面：

- GAN的训练崩溃，模式崩溃问题等依然有待研究改进。
- GAN对超参数很敏感，调参困难的问题有待改进。
- 降低GAN需要的数据量。
- 如何评估生成图像的质量。

应用方面：

1. GAN用于视频处理
2. GAN用于3D模型的生成/合成问题，如 3D colorization [124], 3D face reconstruction [125], [126], 3D character animation [127], and 3D textured object generation
3. data augmentation数据增强, due to its ability to synthesize high-quality images, especially in areas with data paucity, such as medical image analysis生成训练集，特别是对于数据珍贵难以获得的如医学领域
4. 模块化、游戏领域