计算机视觉

张健

数字媒体研究中心
信息工程学院
北京大学深圳研究生院

2021.11.10

作业情况

北京大学 PEKING UNIVERSITY

- Github或者主页下载运行一个超分算法，获得结果试着训练一两个Epoch，给出超分结果

生成对抗网络
Generative Adversarial Network

Generative Adversarial Network (GAN)

Goodfellow, Ian, et al. "Generative Adversarial Nets." *Advances in Neural Information Processing Systems*. 2014.

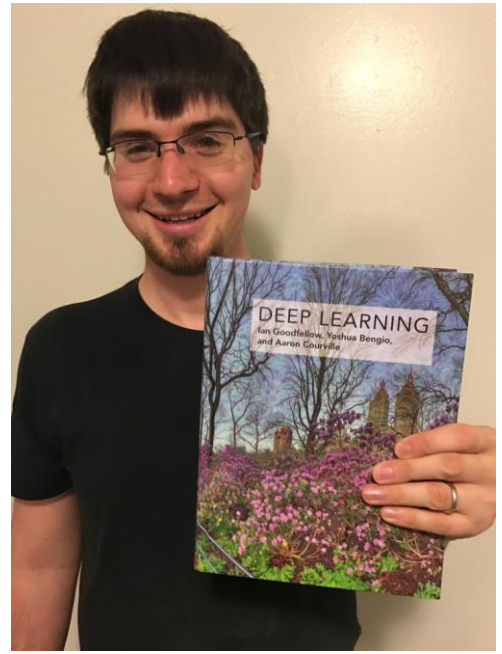
[PDF] Generative Adversarial Nets - NIPS Proceedings

https://papers.nips.cc › paper › 5423-generative-adversarial-nets ▾ 翻译此页

作者: I Goodfellow - 2014 - 被引用次数: 13237 - 相关文章

Generative Adversarial Nets. Ian J. Goodfellow*, Jean Pouget-Abadie†, Mehdi Mirza, Bing Xu, David Warde-Farley,. Sherjil Ozair‡, Aaron Courville, Yoshua ...

NIPS → **NeurIPS**



Yann LeCun’s Comment

What are some recent and potentially upcoming breakthroughs in deep learning?

Yann LeCun, Director of AI Research at Facebook and Professor at NYU

Written Jul 29 · Upvoted by Joaquin Quiñonero Candela, Director Applied Machine Learning at Facebook and Nikhil Garg, I lead a team of Quora engineers working on ML/NLP problems

The most important one, in my opinion, is adversarial training (also called GAN for Generative Adversarial Networks). This is an idea that was originally proposed by Ian Goodfellow when he was a student with Yoshua Bengio at the University of Montreal (he since moved to Google Brain and recently to OpenAI).

This, and the variations that are now being proposed is the most interesting idea in the last 10 years in ML, in my opinion.

https://www.quora.com/What-are-some-recent-and-potentially-upcoming-breakthroughs-in-deep-learning

All Kinds of GAN …

https://github.com/hindupuravinash/the-gan-zoo

北京大学 PEKING UNIVERSITY

GAN

ACGAN

BGAN

CGAN

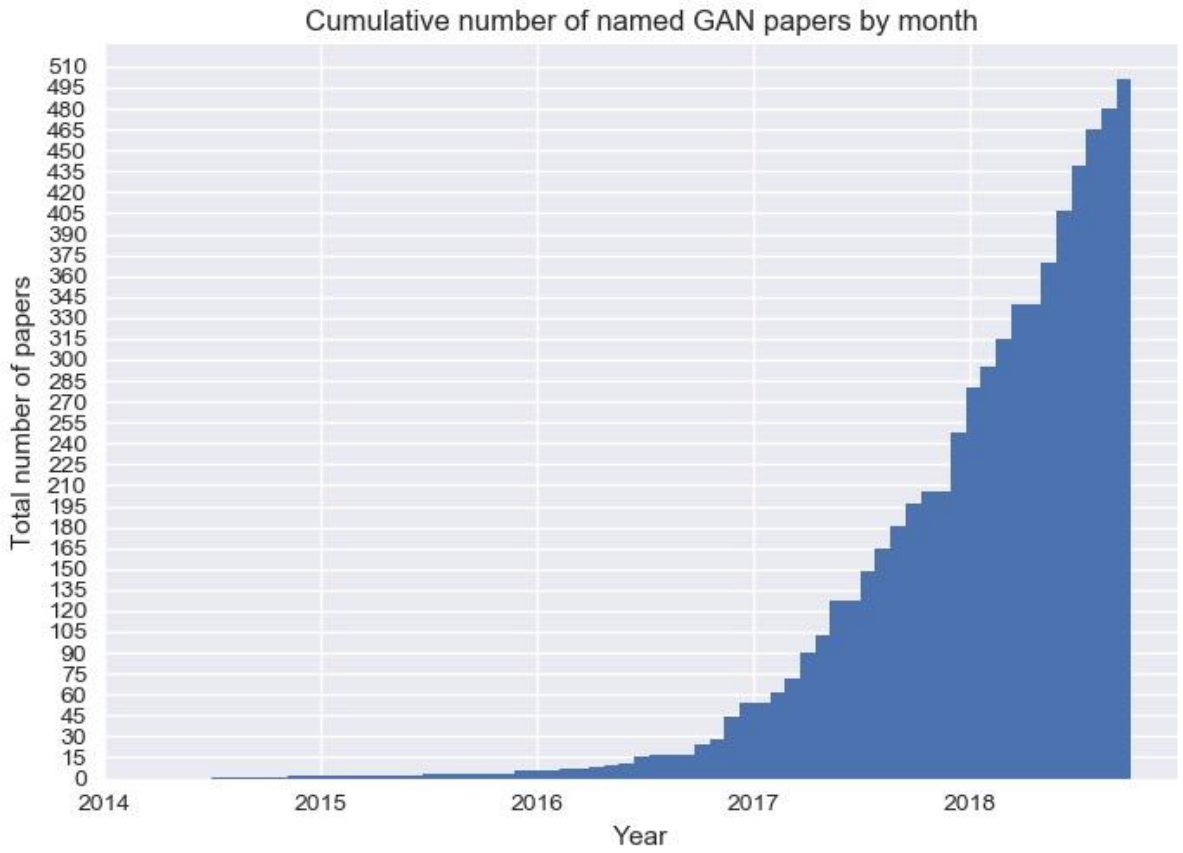
DCGAN

EBGAN

fGAN

GoGAN

⋮ ⋮



Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, Shakir Mohamed, "Variational Approaches for Auto-Encoding Generative Adversarial Networks", arXiv, 2017

[2]We use the Greek α prefix for α-GAN, as AEGAN and most other Latin prefixes seem to have been taken https://deephunt.in/the-gan-zoo-79597dc8c347.

北京大学 PEKING UNIVERSITY

《麻省理工科技评论》评出2018年“全球十大突破性技术”（10 Breakthrough Technologies）

给所有人的人工智能	AI for everyone
对抗性神经网络	Dueling neural networks
人造胚胎	Artificial embryos
基因占卜	Genetic fortune-telling
传感城市	Sensing city
巴别鱼耳塞	Babel-fish earbuds
完美的网络隐私	Perfect online privacy
材料的量子飞跃	Materials' quantum leap
实用型3D金属打印机	3D metal printing
零碳排放天然气发电	Zero-carbon natural gas

生成对抗网络

北京大学 PEKING UNIVERSITY

理解生成（Generation）：

Database

北京大学 PEKING UNIVERSITY

理解生成（Generation）：

Image Generation

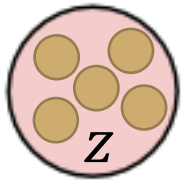
$\begin{bmatrix} 0.3 \\ -0.1 \\ \vdots \\ -0.7 \end{bmatrix}$ $\begin{bmatrix} 0.1 \\ -0.1 \\ \vdots \\ 0.7 \end{bmatrix}$ $\begin{bmatrix} -0.3 \\ 0.1 \\ \vdots \\ 0.9 \end{bmatrix}$

In a specific range

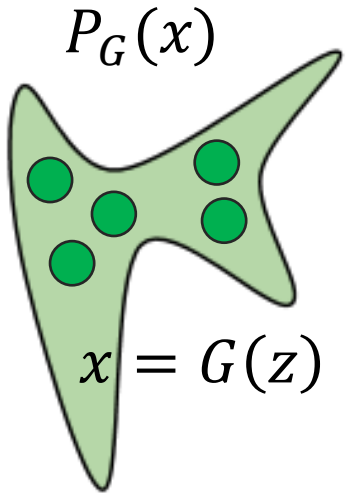
→ NN Generator →

Database

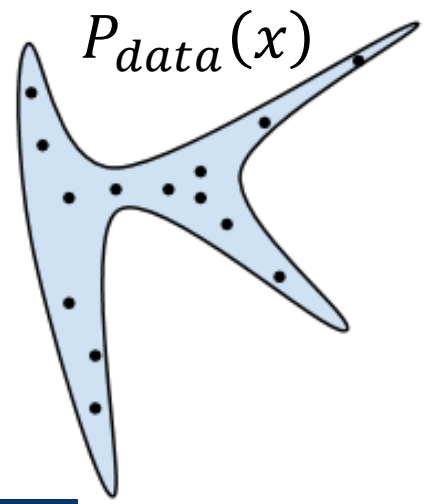
Normal Distribution



→ Generator **G** →



⟷



$$G^* = arg \min_G Div(P_G, P_{data})$$

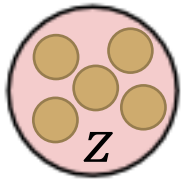
as close as possible

生成对抗网络

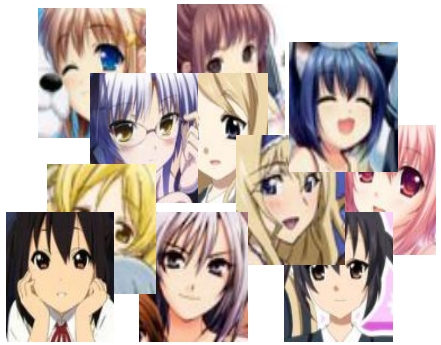
北京大学 PEKING UNIVERSITY

算法目标：

Normal Distribution



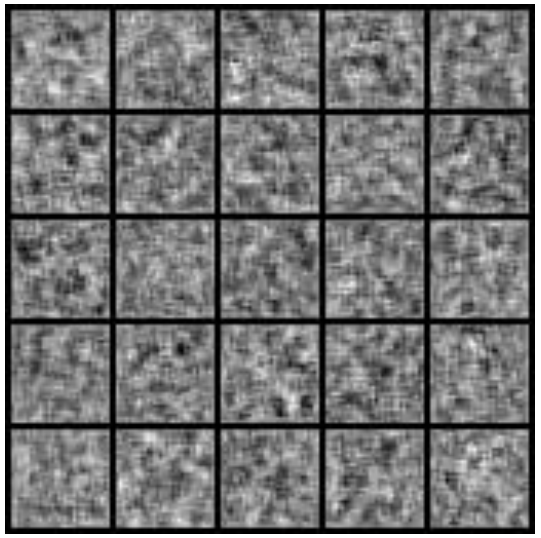
Generator
G

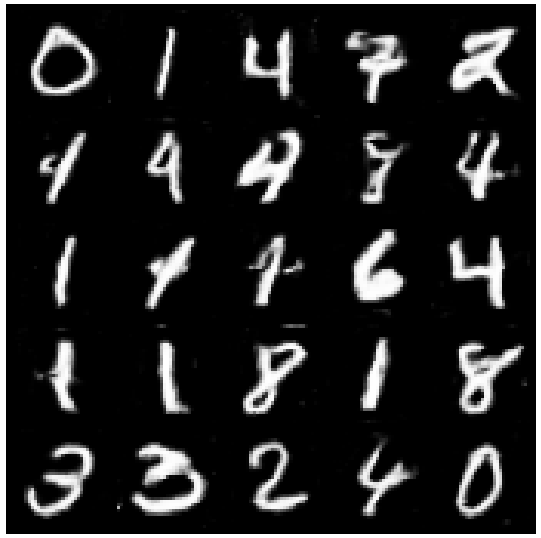


Database

G?

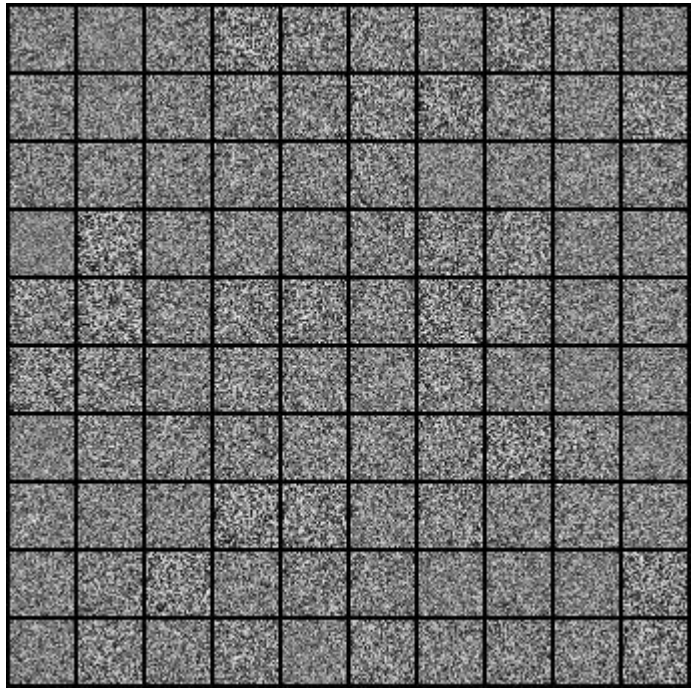
数字生成 DCGAN





https://arxiv.org/abs/1511.06434

数字生成 CGAN

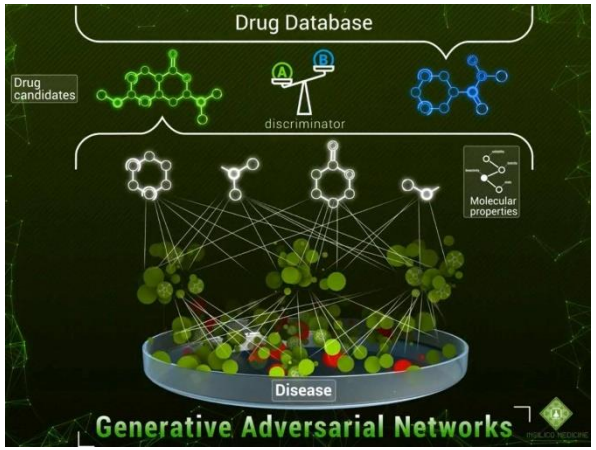


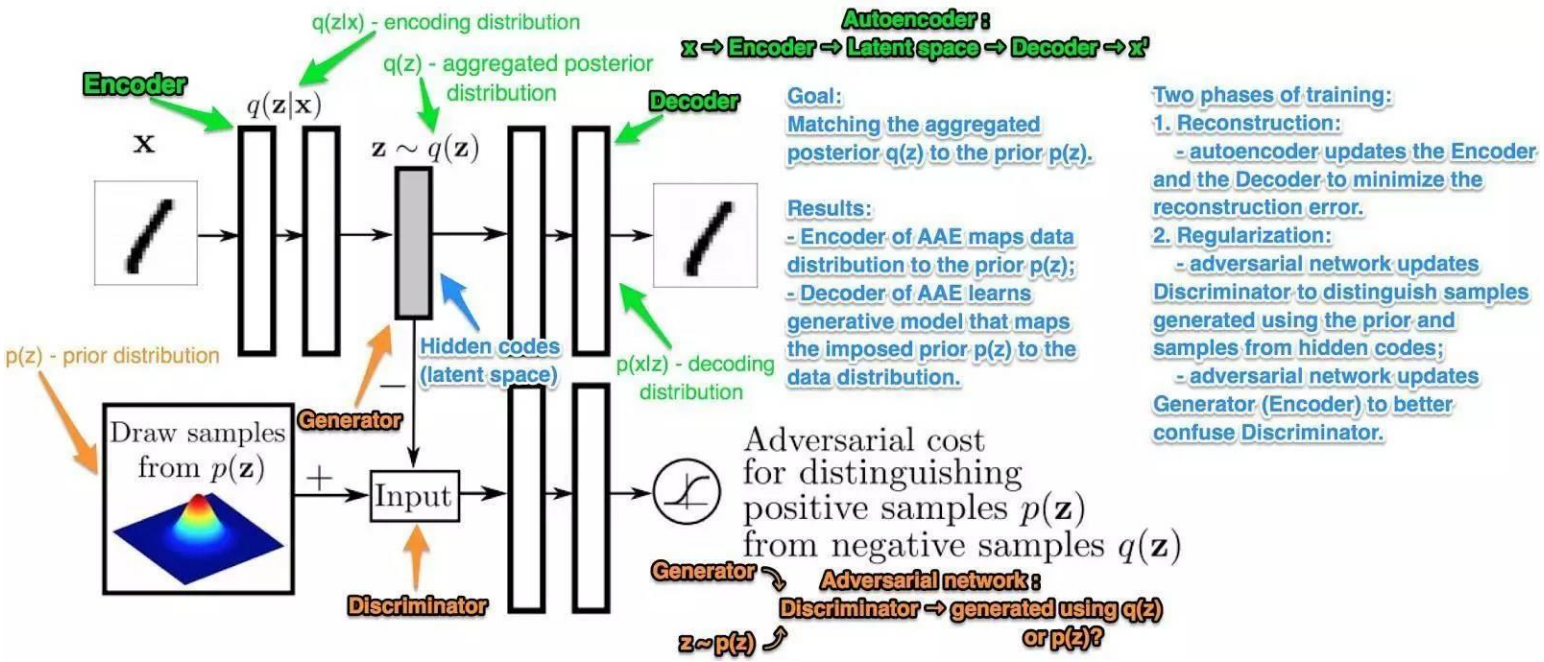
https://arxiv.org/abs/1411.1784

动漫人物生成

https://make.girls.moe/

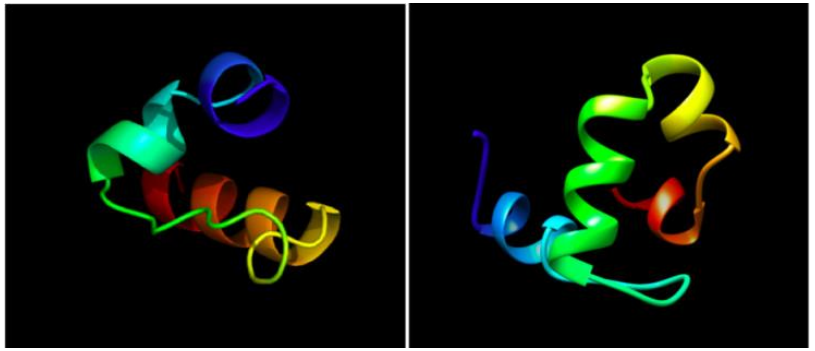
新药物合成

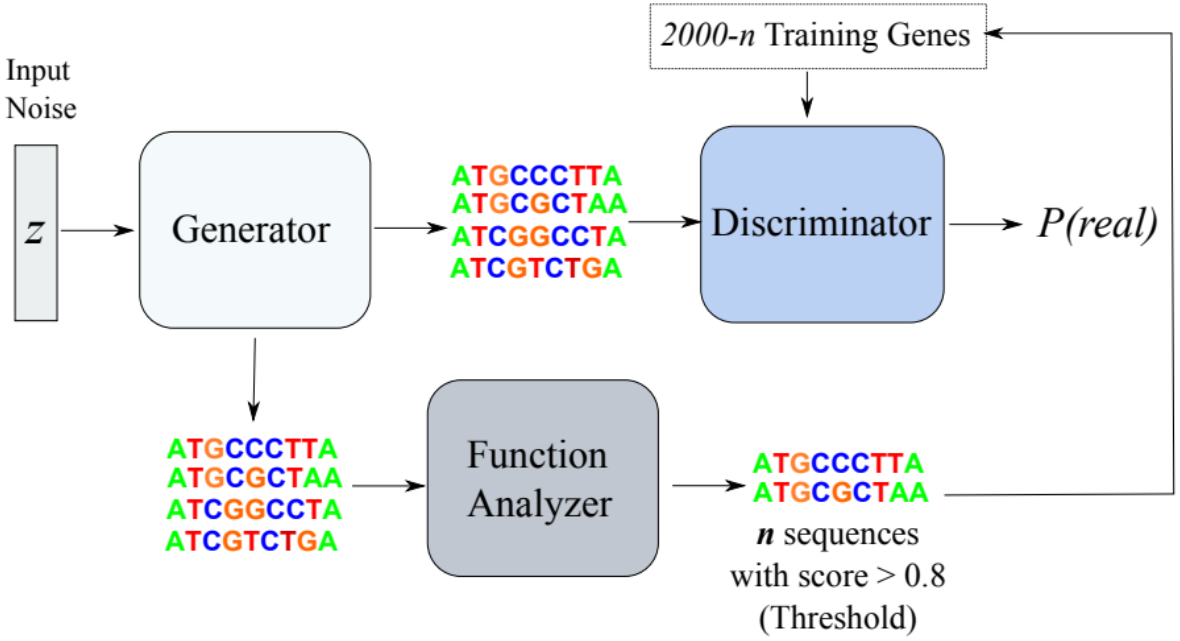




https://www.jianshu.com/p/2b7f8c37def7

基因合成





https://www.leiphone.com/news/201804/f2QBvnIYEQx9X3fh.html

预测年龄增长后的脸

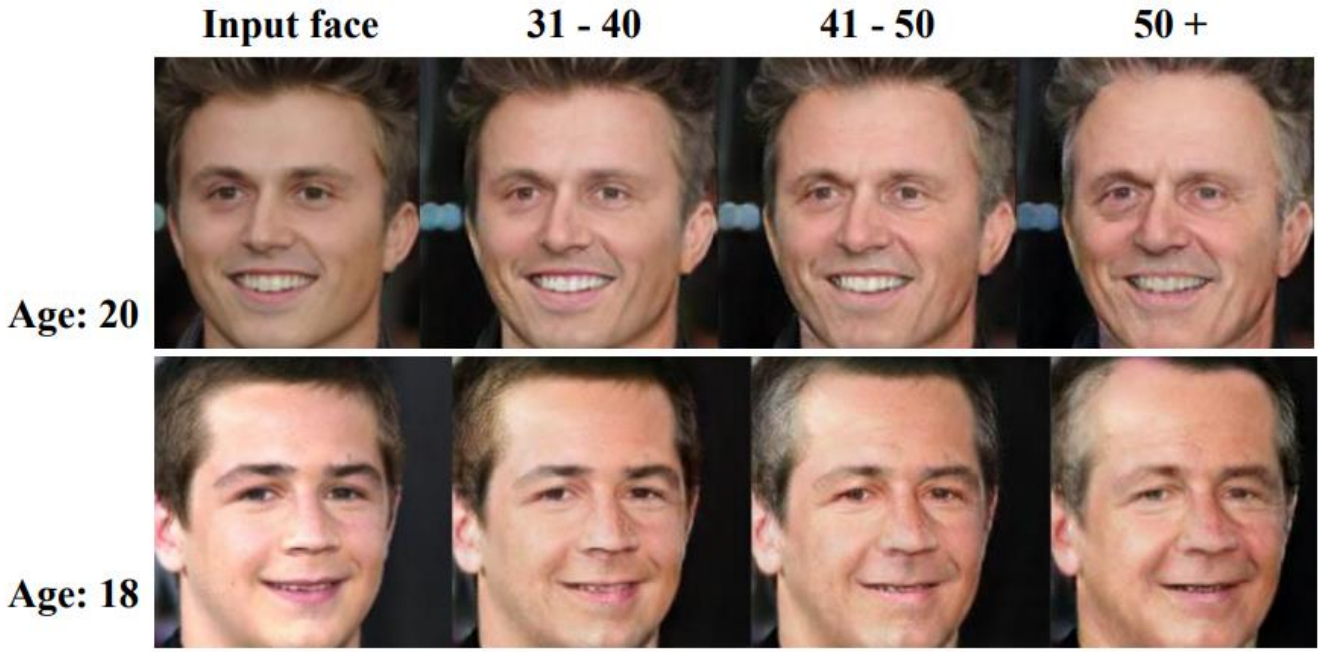
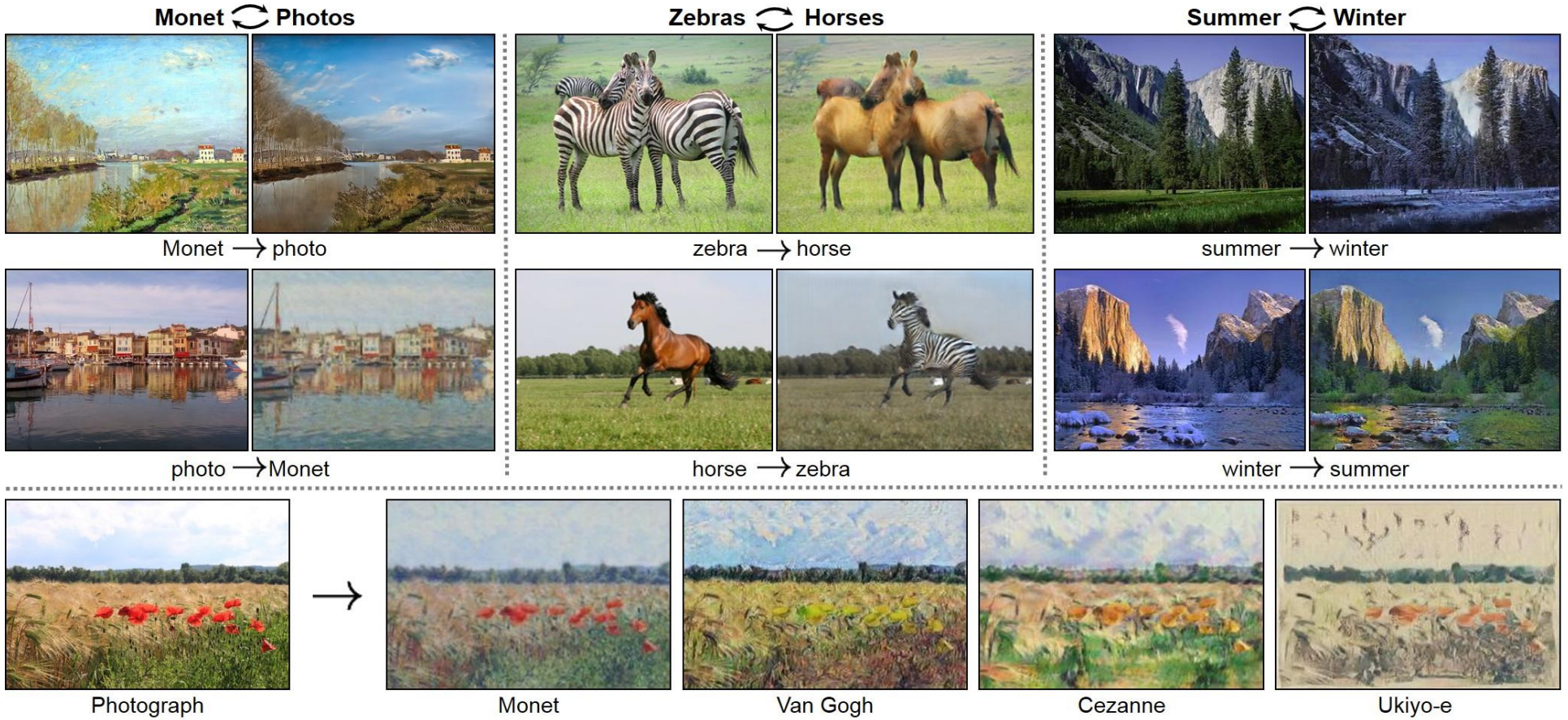


Figure 1. Demonstration of our aging simulation results (images in the first column are input faces of two subjects).

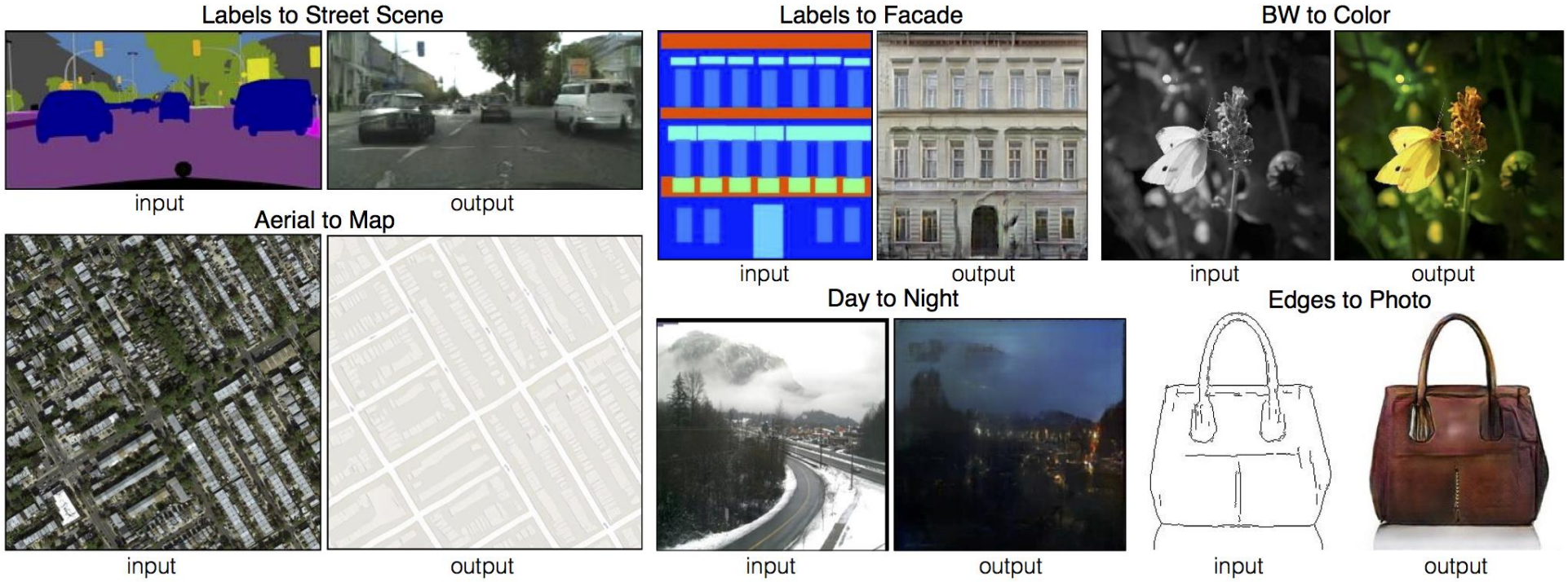
https://arxiv.org/pdf/1711.10352.pdf

图像之间转换



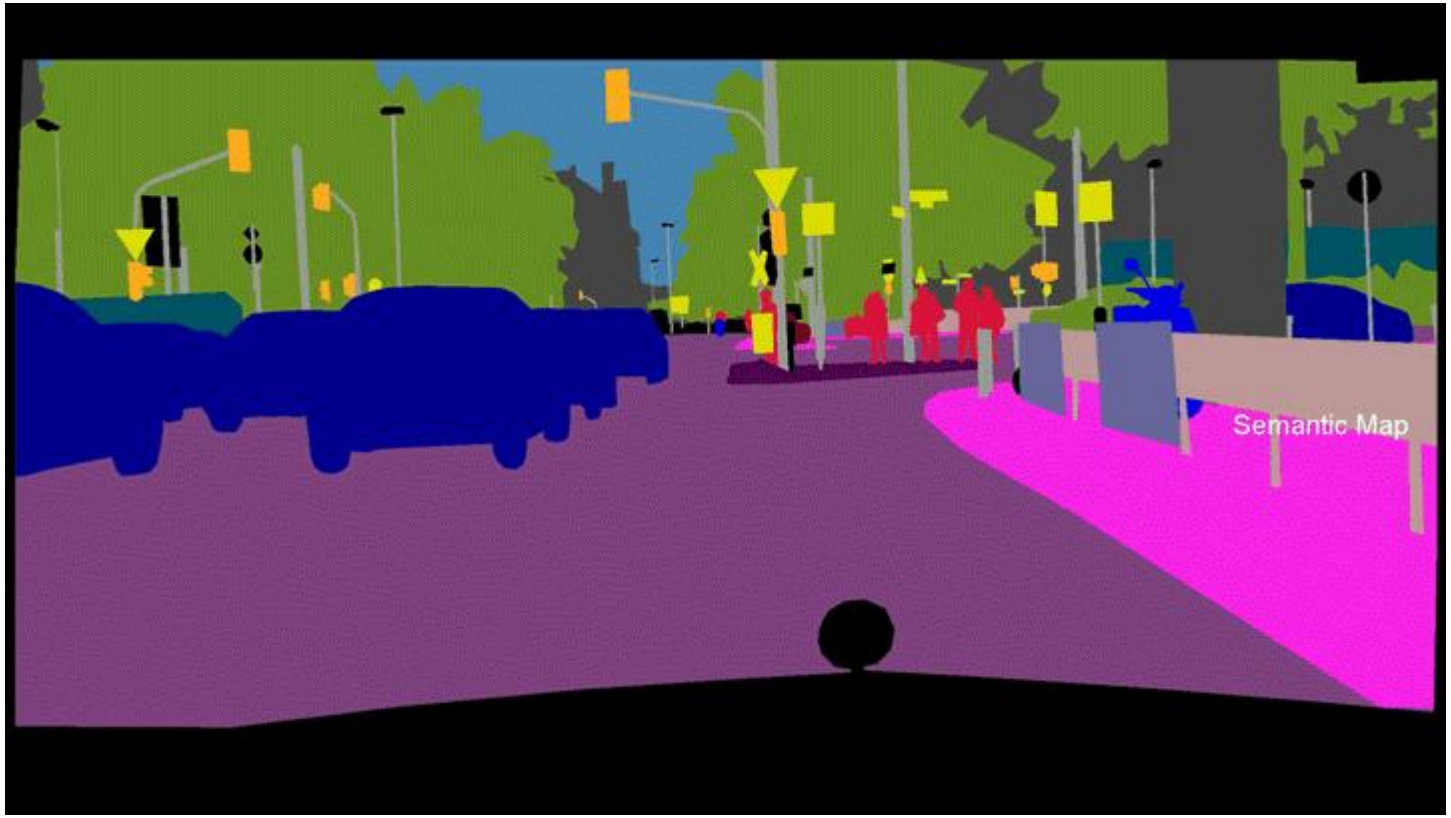
https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

图像之间转换



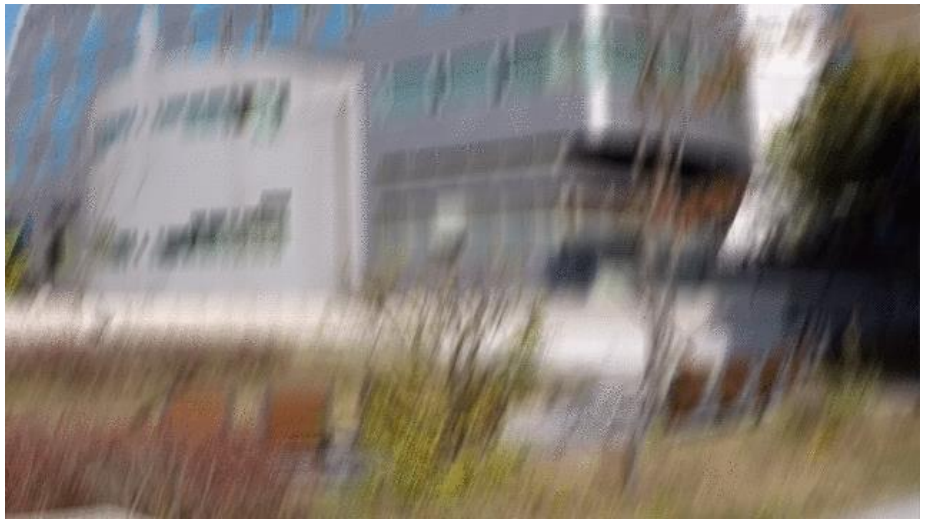
https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

高分辨率图像生成



https://github.com/NVIDIA/pix2pixHD

DeblurGAN



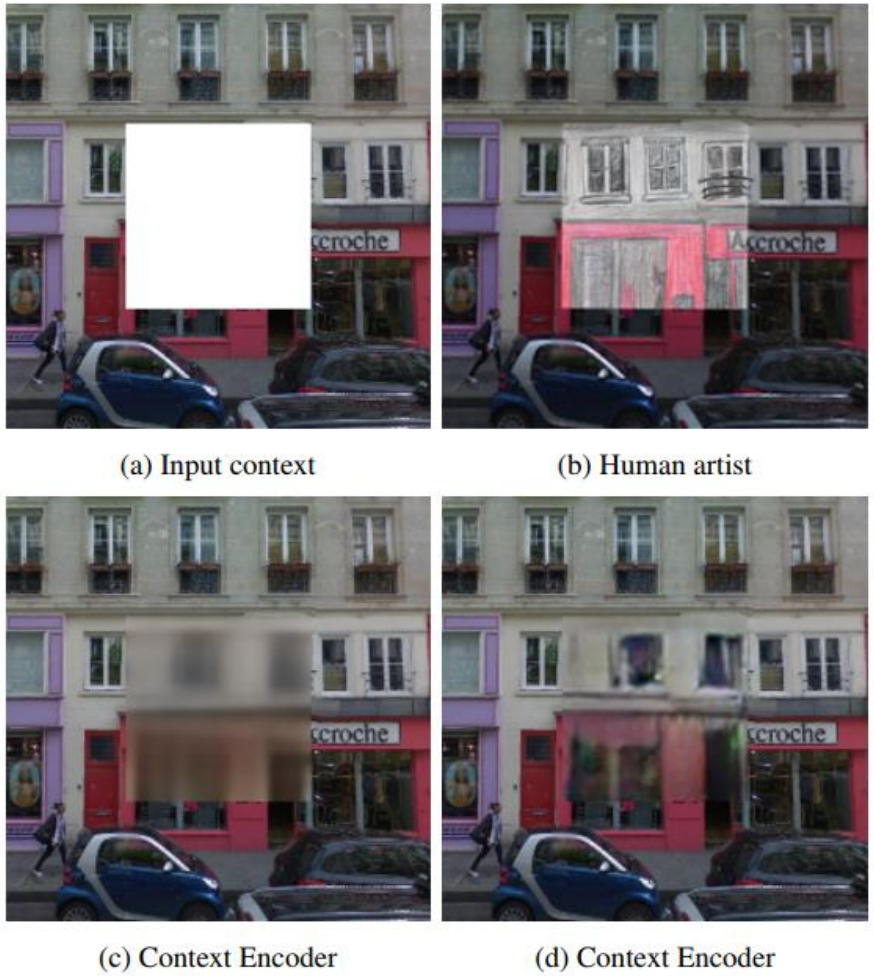
https://github.com/KupynOrest/DeblurGAN

BigGAN

Figure 6: Additional samples generated by our model at 512×512 resolution.

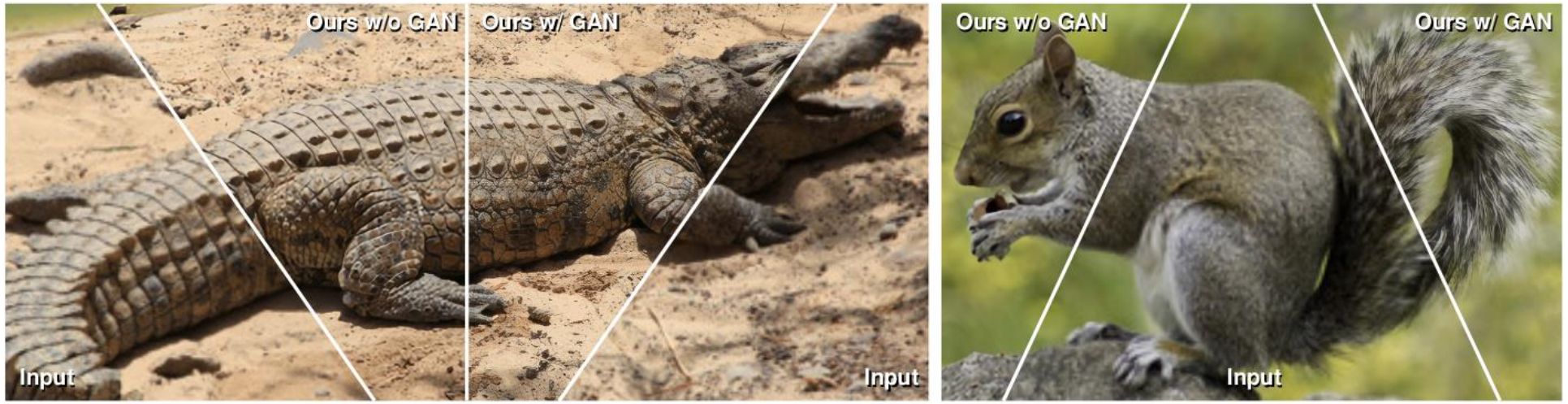
https://arxiv.org/abs/1809.11096

图像修补



Context Encoders: Feature Learning by Inpainting https://arxiv.org/pdf/1604.07379.pdf

图像超分辨率



https://yifita.github.io/publication/prosr/

StyleGAN

https://github.com/NVlabs/stylegan

Source

北京大学 PEKING UNIVERSITY

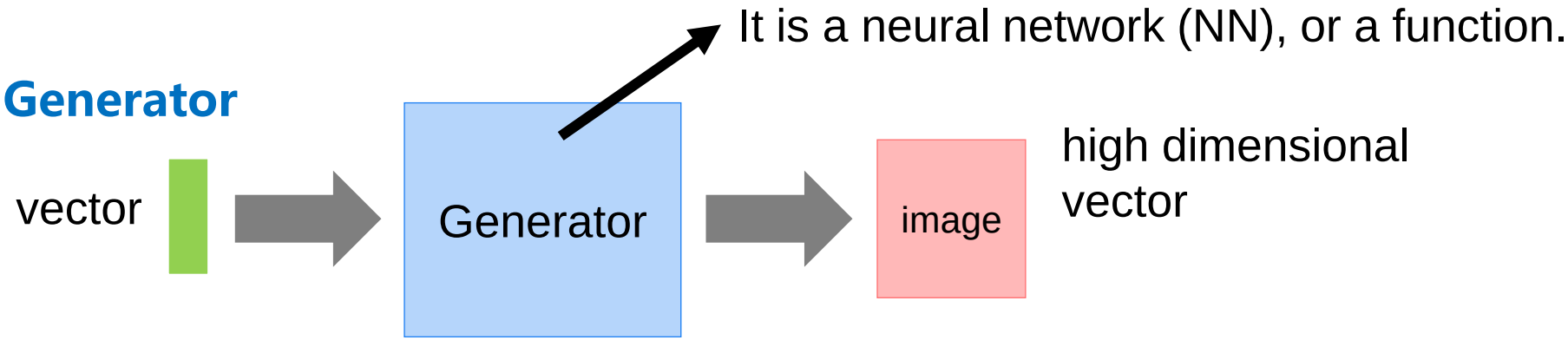
- https://github.com/hindupuravinash/the-gan-zoo
- https://github.com/eriklindernoren/PyTorch-GAN

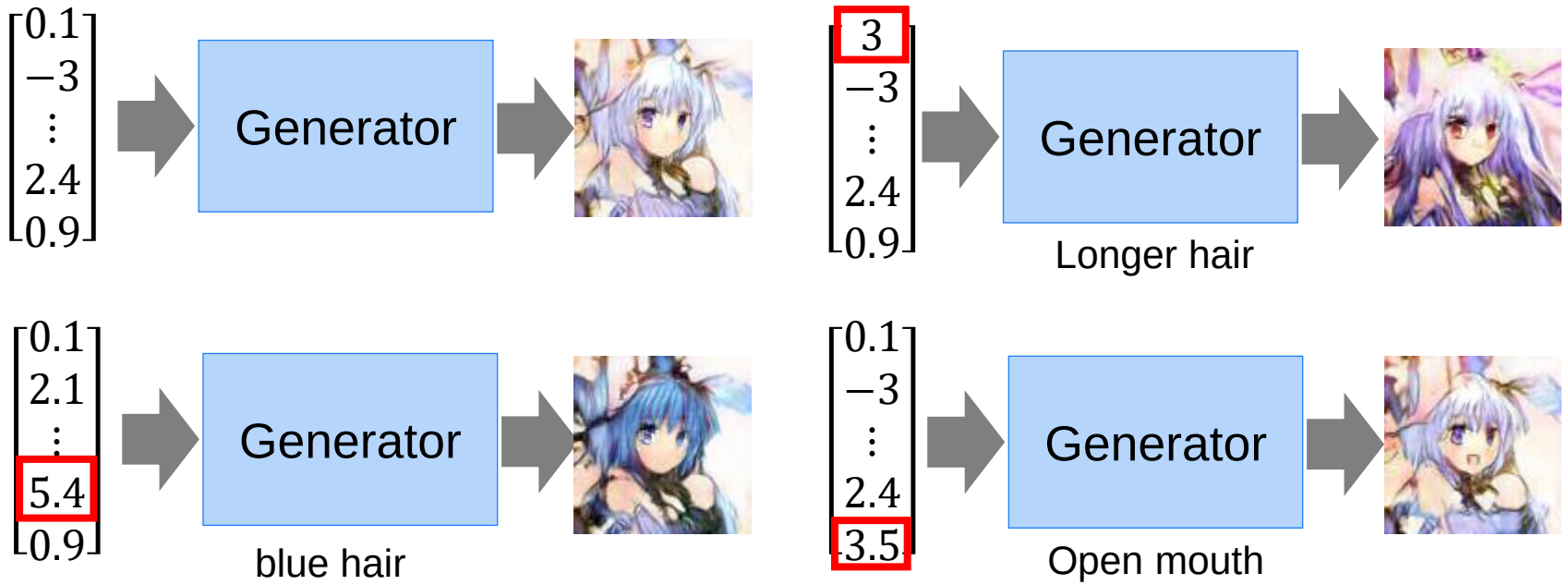
生成对抗网络

北京大学 PEKING UNIVERSITY

算法思想：

首先引入一个生成器 Generator



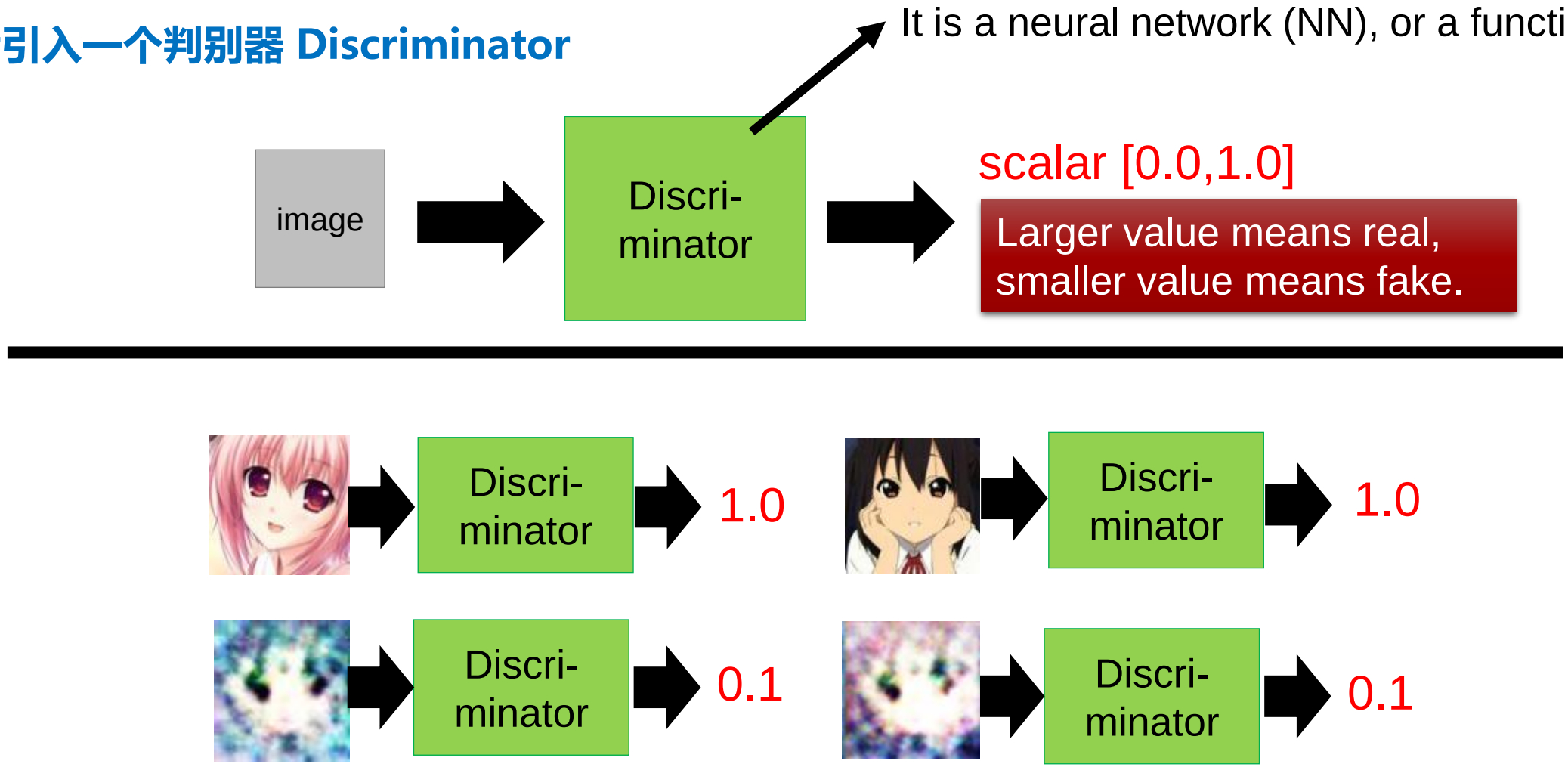


生成对抗网络

北京大学 PEKING UNIVERSITY

算法思想：

然后引入一个判别器 Discriminator

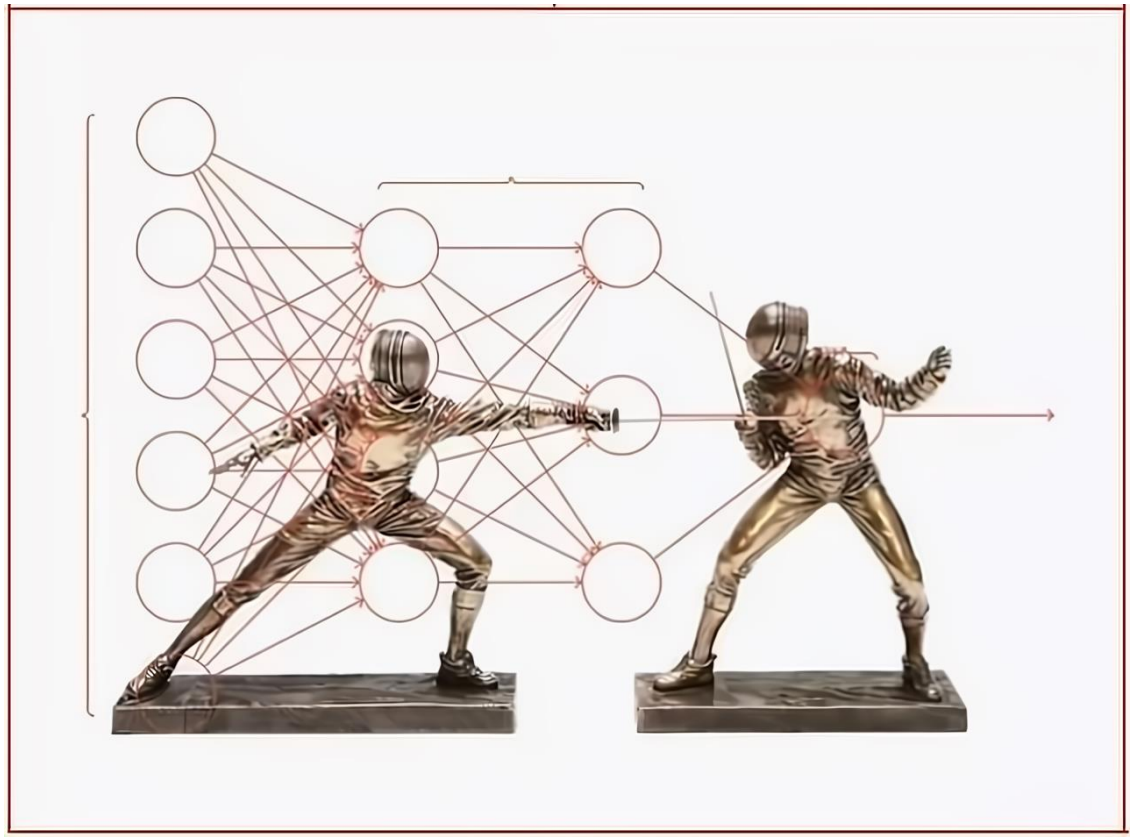


生成对抗网络

北京大学 PEKING UNIVERSITY

算法思想：

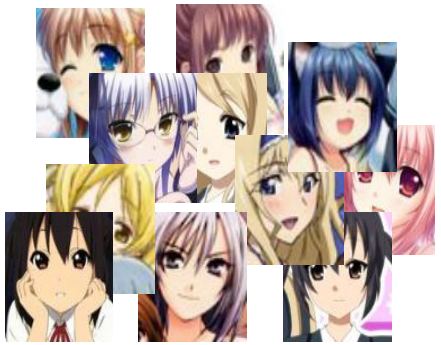
生成器与判别器开始“左右互搏”



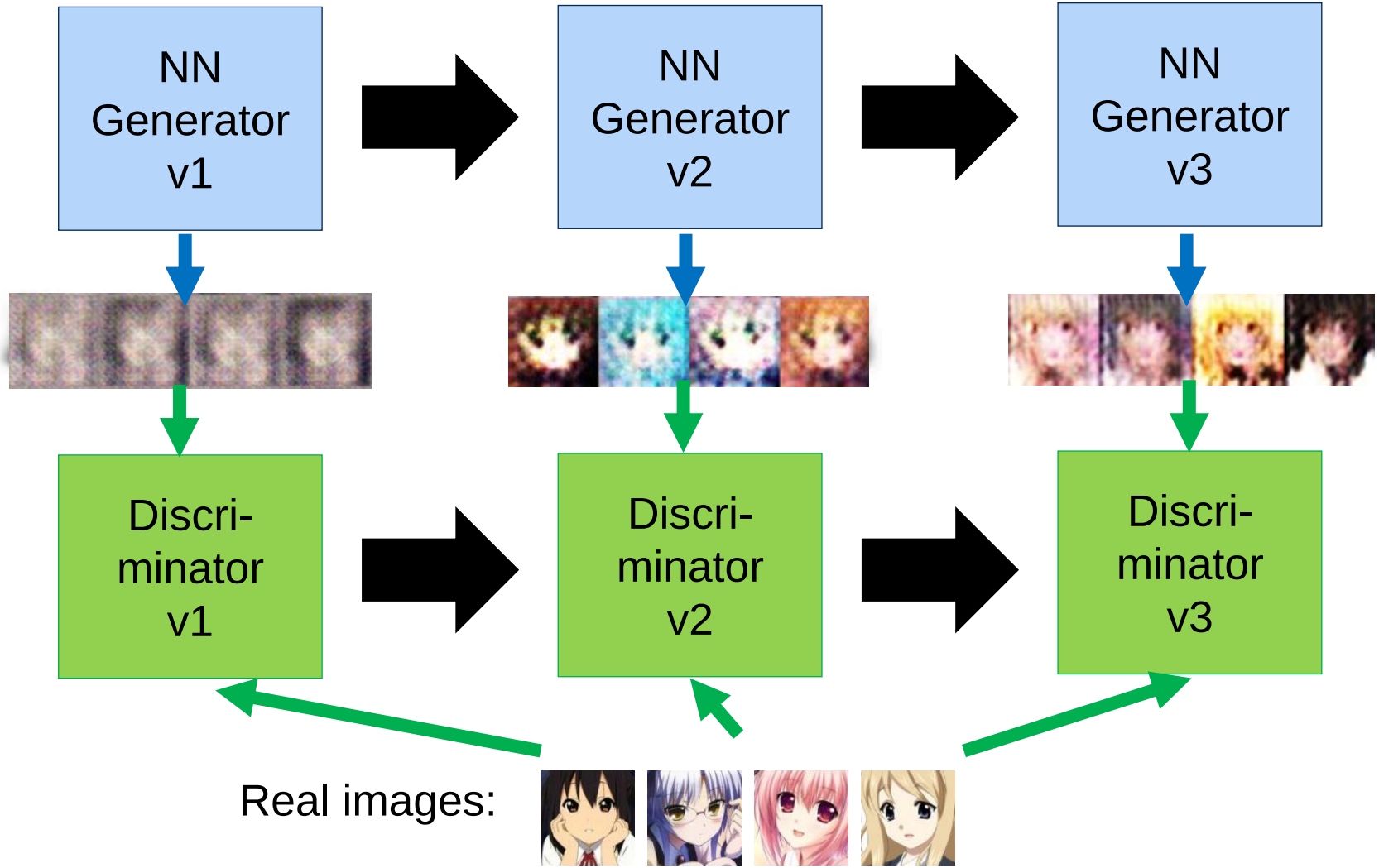
生成对抗网络

北京大学 PEKING UNIVERSITY

算法思想：



Database

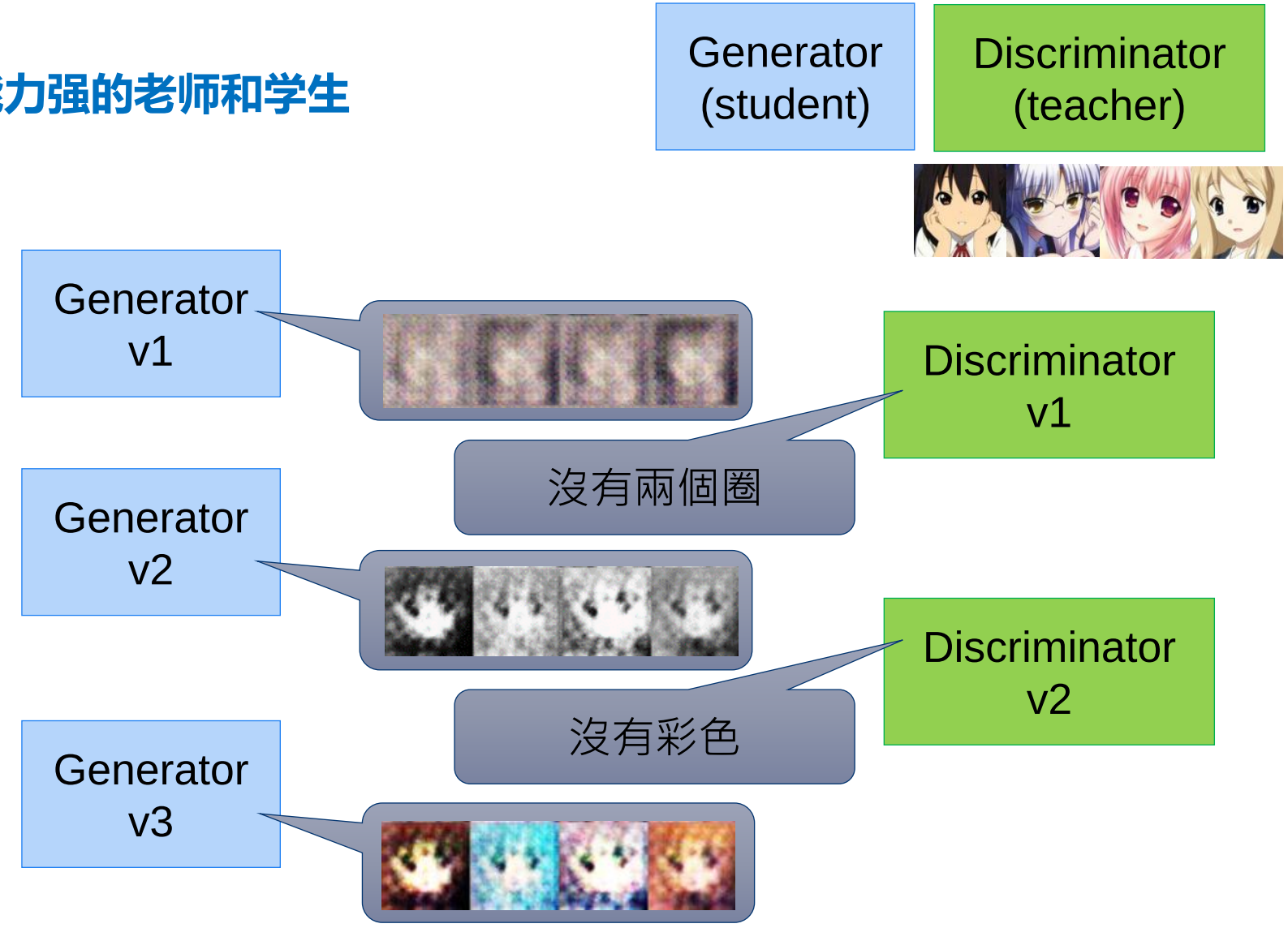


生成对抗网络

北京大学 PEKING UNIVERSITY

算法思想：

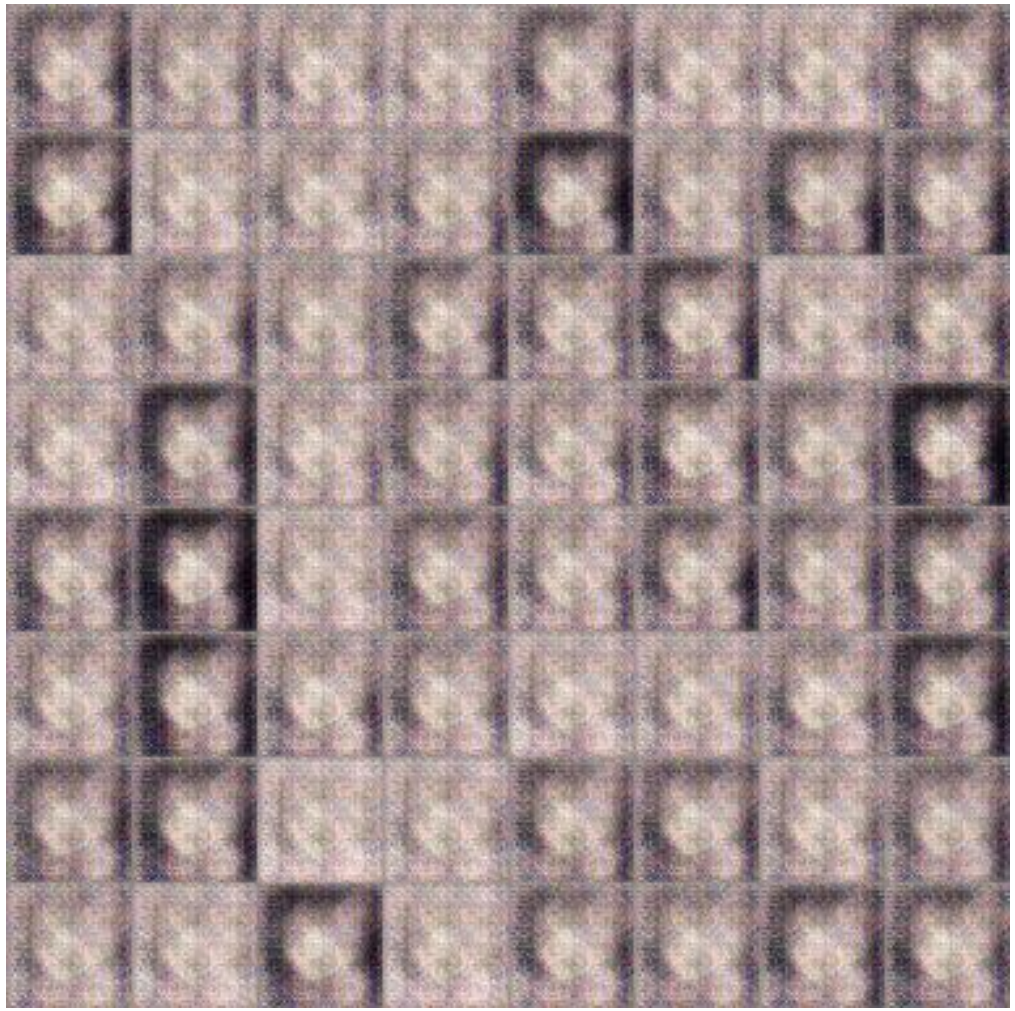
和平比喻：学习能力强的老师和学生



生成对抗网络

北京大学 PEKING UNIVERSITY

算法思想：



100 updates

Database

生成对抗网络

北京大学 PEKING UNIVERSITY

算法思想：

1000 updates

生成对抗网络

北京大学 PEKING UNIVERSITY

算法思想：

2000 updates

生成对抗网络

北京大学 PEKING UNIVERSITY

算法思想：

10,000 updates

生成对抗网络

北京大学 PEKING UNIVERSITY

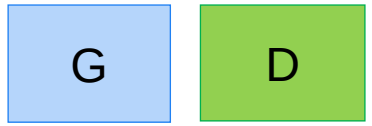
算法思想：

The faces generated by machine.

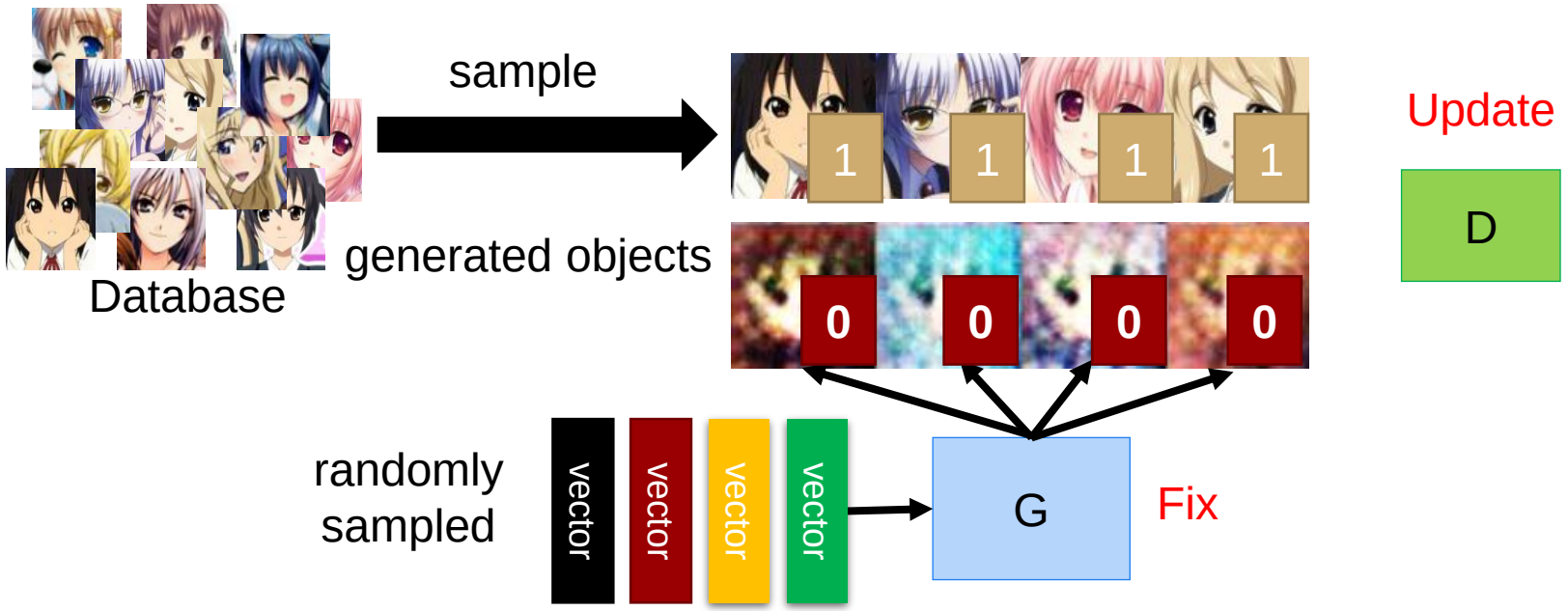
生成对抗网络

算法流程：

- Initialize generator and discriminator
- In each training iteration:



Step 1: Fix generator G, and update discriminator D

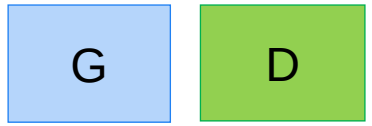


Discriminator learns to assign high scores to real objects and low scores to generated objects.

生成对抗网络

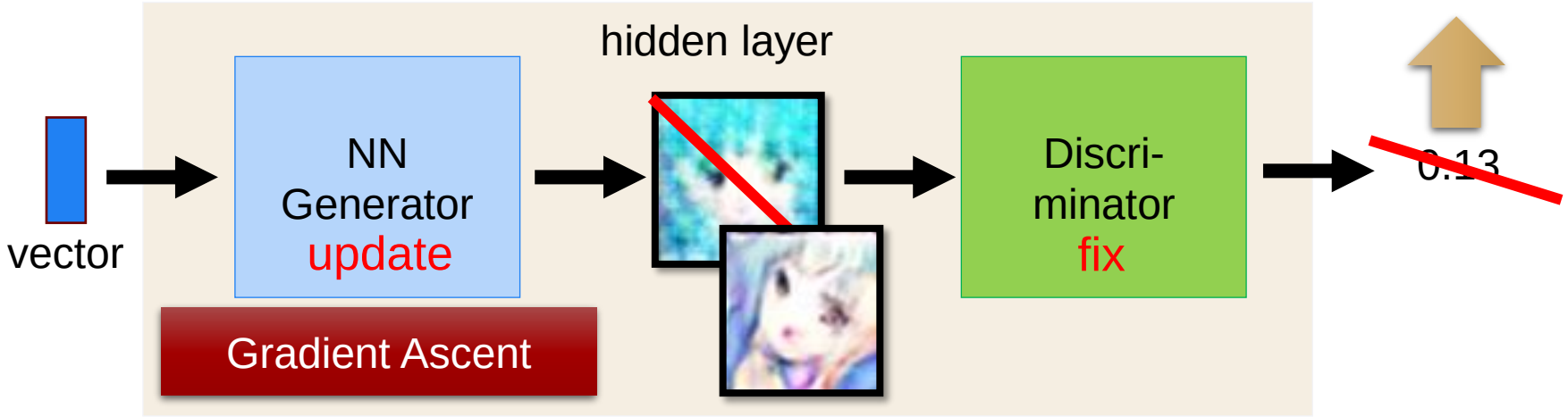
算法流程：

- Initialize generator and discriminator
- In each training iteration:



Step 2: Fix discriminator D, and update generator G

Generator learns to “fool” the discriminator



算法流程：

Algorithm Initialize θ_d for D and θ_g for G

Learning D

- In each training iteration:
  - Sample m examples $\{x^1, x^2, \dots, x^m\}$ from database
  - Sample m noise samples $\{z^1, z^2, \dots, z^m\}$ from a distribution
  - Obtaining generated data $\{\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^m\}$, $\tilde{x}^i = G(z^i)$
  - Update discriminator parameters θ_d to maximize
    - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} logD(x^i) + \frac{1}{m}\sum_{i=1}^{m} log\left(1 - D(\tilde{x}^i)\right)$
    - $\theta_d \leftarrow \theta_d + \eta\nabla\tilde{V}(\theta_d)$

Learning G

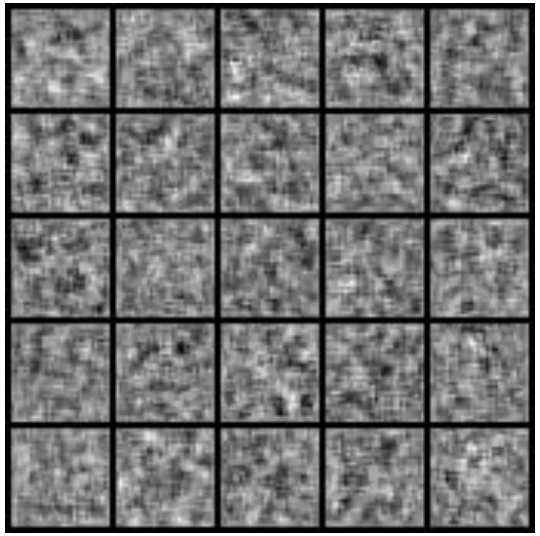
- Sample m noise samples$\{z^1, z^2, \dots, z^m\}$ from a distribution
- Update generator parameters θ_g to maximize
  - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} log\left(D\left(G(z^i)\right)\right)$
  - $\theta_g \leftarrow \theta_g + \eta\nabla\tilde{V}(\theta_g)$

生成对抗网络

北京大学 PEKING UNIVERSITY

代码实现：

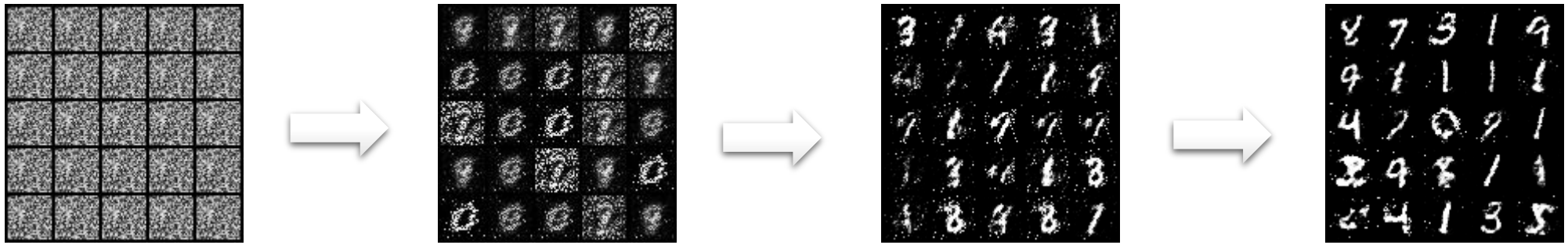
```python
# ----------
#  Training
# ----------

for epoch in range(opt.n_epochs):
    for i, (imgs, _) in enumerate(dataloader):

        # Adversarial ground truths
        valid = Variable(Tensor(imgs.size(0), 1).fill_(1.0), requires_grad=False)
        fake = Variable(Tensor(imgs.size(0), 1).fill_(0.0), requires_grad=False)

        # Configure input
        real_imgs = Variable(imgs.type(Tensor))

        # -----------------
        #  Train Generator
        # -----------------

        optimizer_G.zero_grad()

        # Sample noise as generator input
        z = Variable(Tensor(np.random.normal(0, 1, (imgs.shape[0], opt.latent_dim))))

        # Generate a batch of images
        gen_imgs = generator(z)

        # Loss measures generator's ability to fool the discriminator
        g_loss = adversarial_loss(discriminator(gen_imgs), valid)

        g_loss.backward()
        optimizer_G.step()

        # ---------------------
        #  Train Discriminator
        # ---------------------

        optimizer_D.zero_grad()

        # Measure discriminator's ability to classify real from generated samples
        real_loss = adversarial_loss(discriminator(real_imgs), valid)
        fake_loss = adversarial_loss(discriminator(gen_imgs.detach()), fake)
        d_loss = (real_loss + fake_loss) / 2
```



代码分析

北京大学 PEKING UNIVERSITY

- GAN 代码分析（PyCharm）
- CGAN 代码分析（PyCharm）
- DCGAN 代码分析（PyCharm）
- LSGAN代码分析（PyCharm）



北京大学 PEKING UNIVERSITY

理论分析：

Image Generation

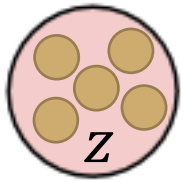
$\begin{bmatrix} 0.3 \\ -0.1 \\ \vdots \\ -0.7 \end{bmatrix}$ $\begin{bmatrix} 0.1 \\ -0.1 \\ \vdots \\ 0.7 \end{bmatrix}$ $\begin{bmatrix} -0.3 \\ 0.1 \\ \vdots \\ 0.9 \end{bmatrix}$

In a specific range

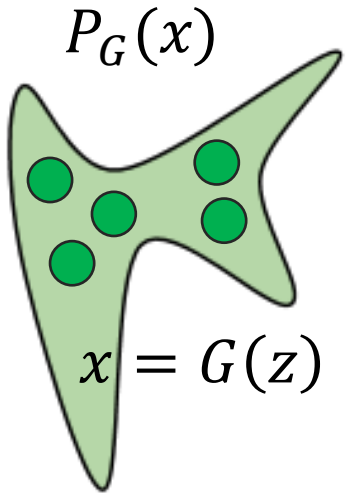
→ NN Generator →

Database

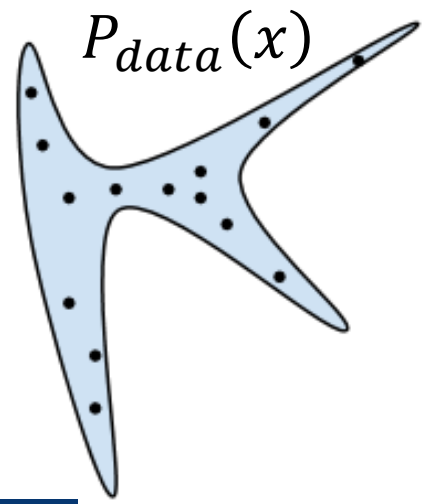
Normal Distribution



→ Generator **G** →



⟷



$$G^* = arg \min_G Div(P_G, P_{data})$$

as close as possible

各种散度对比

Name	$D_f(P\|\|Q)$	Generator $f(u)$
Total variation	$\frac{1}{2}\int \lvert p(x)-q(x)\rvert \,\mathrm{d}x$	$\frac{1}{2}\lvert u-1\rvert$
Kullback-Leibler	$\int p(x)\log\frac{p(x)}{q(x)}\,\mathrm{d}x$	$u\log u$
Reverse Kullback-Leibler	$\int q(x)\log\frac{q(x)}{p(x)}\,\mathrm{d}x$	$-\log u$
Pearson χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)}\,\mathrm{d}x$	$(u-1)^2$
Neyman χ^2	$\int \frac{(p(x)-q(x))^2}{q(x)}\,\mathrm{d}x$	$\frac{(1-u)^2}{u}$
Squared Hellinger	$\int \left(\sqrt{p(x)}-\sqrt{q(x)}\right)^2\,\mathrm{d}x$	$(\sqrt{u}-1)^2$
Jeffrey	$\int (p(x)-q(x))\log\left(\frac{p(x)}{q(x)}\right)\,\mathrm{d}x$	$(u-1)\log u$
Jensen-Shannon	$\frac{1}{2}\int p(x)\log\frac{2p(x)}{p(x)+q(x)}+q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x$	$-(u+1)\log\frac{1+u}{2}+u\log u$
Jensen-Shannon-weighted	$\int p(x)\pi\log\frac{p(x)}{\pi p(x)+(1-\pi)q(x)}+(1-\pi)q(x)\log\frac{q(x)}{\pi p(x)+(1-\pi)q(x)}\,\mathrm{d}x$	$\pi u\log u-(1-\pi+\pi u)\log(1-\pi+\pi u)$
GAN	$\int p(x)\log\frac{2p(x)}{p(x)+q(x)}+q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x-\log(4)$	$u\log u-(u+1)\log(u+1)$

Name	Conjugate $f^*(t)$
Total variation	t
Kullback-Leibler (KL)	$\exp(t-1)$
Reverse KL	$-1-\log(-t)$
Pearson χ^2	$\frac{1}{4}t^2+t$
Neyman χ^2	$2-2\sqrt{1-t}$
Squared Hellinger	$\frac{t}{1-t}$
Jeffrey	$W(e^{1-t})+\frac{1}{W(e^{1-t})}+t-2$
Jensen-Shannon	$-\log(2-\exp(t))$
Jensen-Shannon-weighted	$(1-\pi)\log\frac{1-\pi}{1-\pi e^{t/\pi}}$
GAN	$-\log(1-\exp(t))$

本次作业

北京大学 PEKING UNIVERSITY

- https://github.com/eriklindernoren/PyTorch-GAN上任选一个感兴趣的GAN的程序，下载运行成功。
- 阅读该程序的论文，写出阅读总结，并对应代码标注出论文中的公式以及网络所对应的代码，阐述清楚。
- 不超过两页。

交流&问题?