Computer Vision

第十三周 生成对抗网络

庞彦
yanpang@gzhu.edu.cn

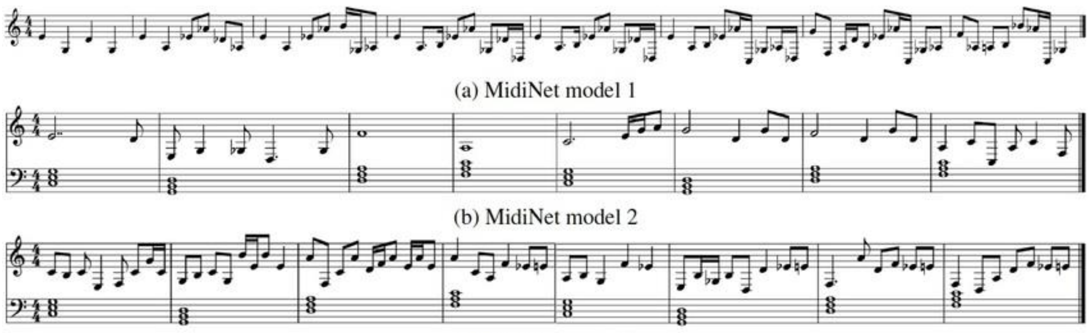
01 Generative Adversarial Networks（GANs）

生成对抗网络

Generative Adversarial Networks （GANs）

Game development and animation production are expensive and hire many production artists for relatively routine tasks.

GAN can auto-generate and colorize Anime characters.



(a) MidiNet model 1

(b) MidiNet model 2

(c) MidiNet model 3

Pose Guided Person Image Generation

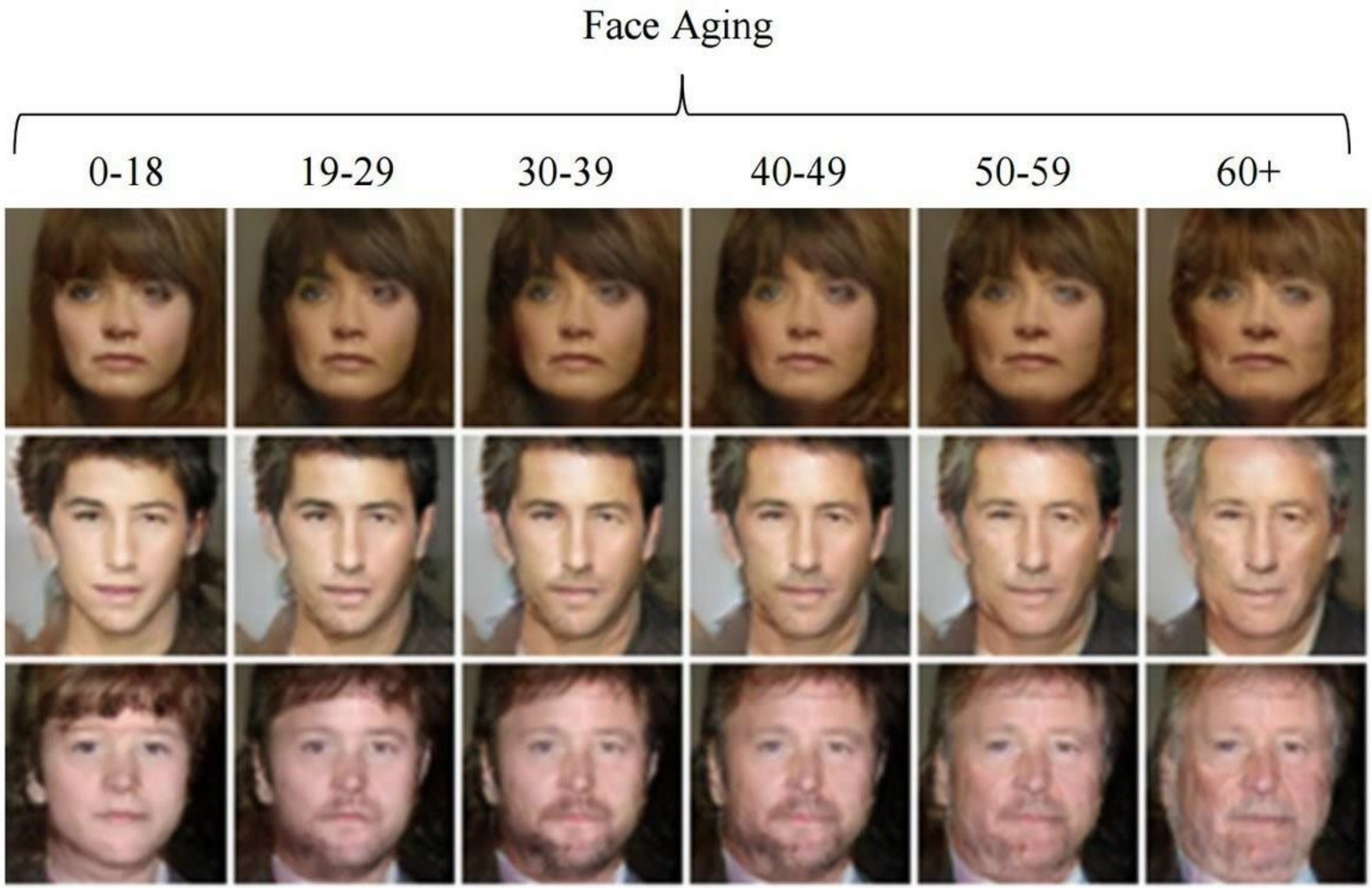
With an additional input of the pose, we can transform an image into different poses.

(c) Generating from a sequence of poses

CycleGAN

Cross-domain transfer GANs will be likely the first batch of commercial applications. These GANs transform images from one domain (say real scenery) to another domain (Monet paintings or Van Gogh).

Face Aging



PixelDTGAN

Suggesting merchandise based on celebrity pictures has been popular for fashion blogger and e-commerce.

PixelDTGAN creates clothing images and styles from an image.

A source image. Possible target images.

StackGAN

Text to image is one of the earlier application of domain-transfer GAN. We input a sentence and generate multiple images fitting the description.

DiscoGAN

DiscoGAN provides matching style: many potential applications.

DiscoGAN learns cross domain relationship without labels or pairing. For example, it successfully transfers style (or patterns) from one domain (handbag) to another (shoe).

(b) Handbag images (input) & **Generated** shoe images (output)

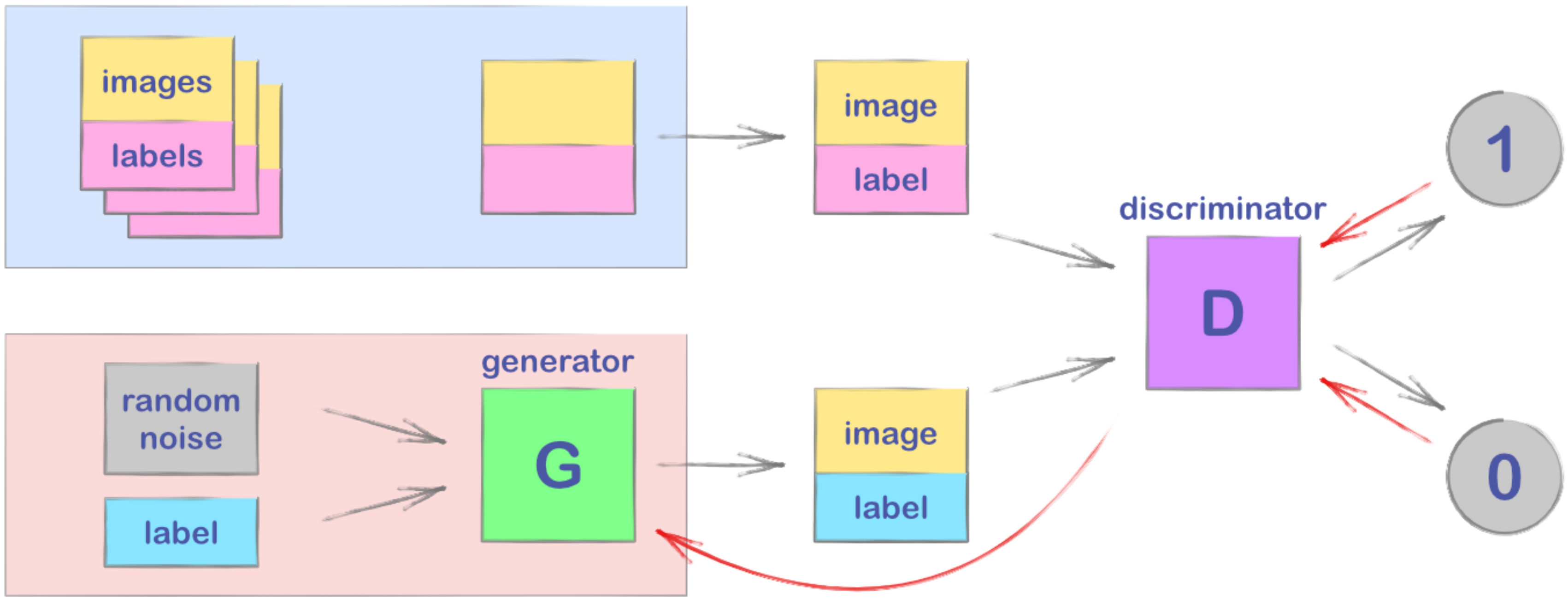
Generative Adversarial Networks （GANs）

GANs have various applications on commercial market.

Welcome to GANs’ world.

Figure 1: Class-conditional samples generated by our model.

Basic Ideas of GAN



Generator

Image Generation

We will control what to generate latter. → Conditional Generation

$$\begin{bmatrix}0.3\\-0.1\\\vdots\\-0.7\end{bmatrix}\begin{bmatrix}0.1\\-0.1\\\vdots\\0.7\end{bmatrix}\begin{bmatrix}-0.3\\0.1\\\vdots\\0.9\end{bmatrix} \rightarrow \boxed{\text{NN Generator}} \rightarrow$$

In a specific range

Sentence Generation

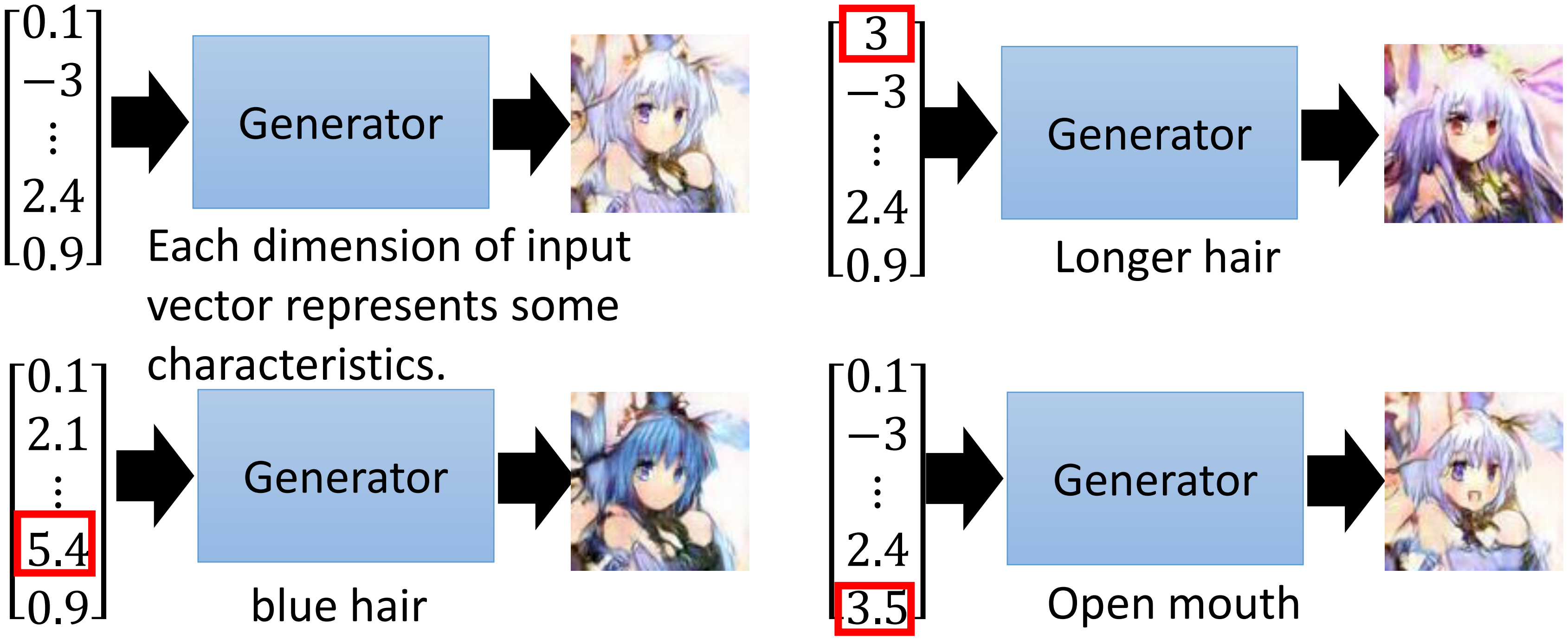
$$\begin{bmatrix}0.3\\0.5\\\vdots\\-0.6\end{bmatrix}\begin{bmatrix}0.2\\-0.3\\\vdots\\0.2\end{bmatrix}\begin{bmatrix}-0.3\\0.2\\\vdots\\0.3\end{bmatrix} \rightarrow \boxed{\text{NN Generator}} \rightarrow$$

What's up?

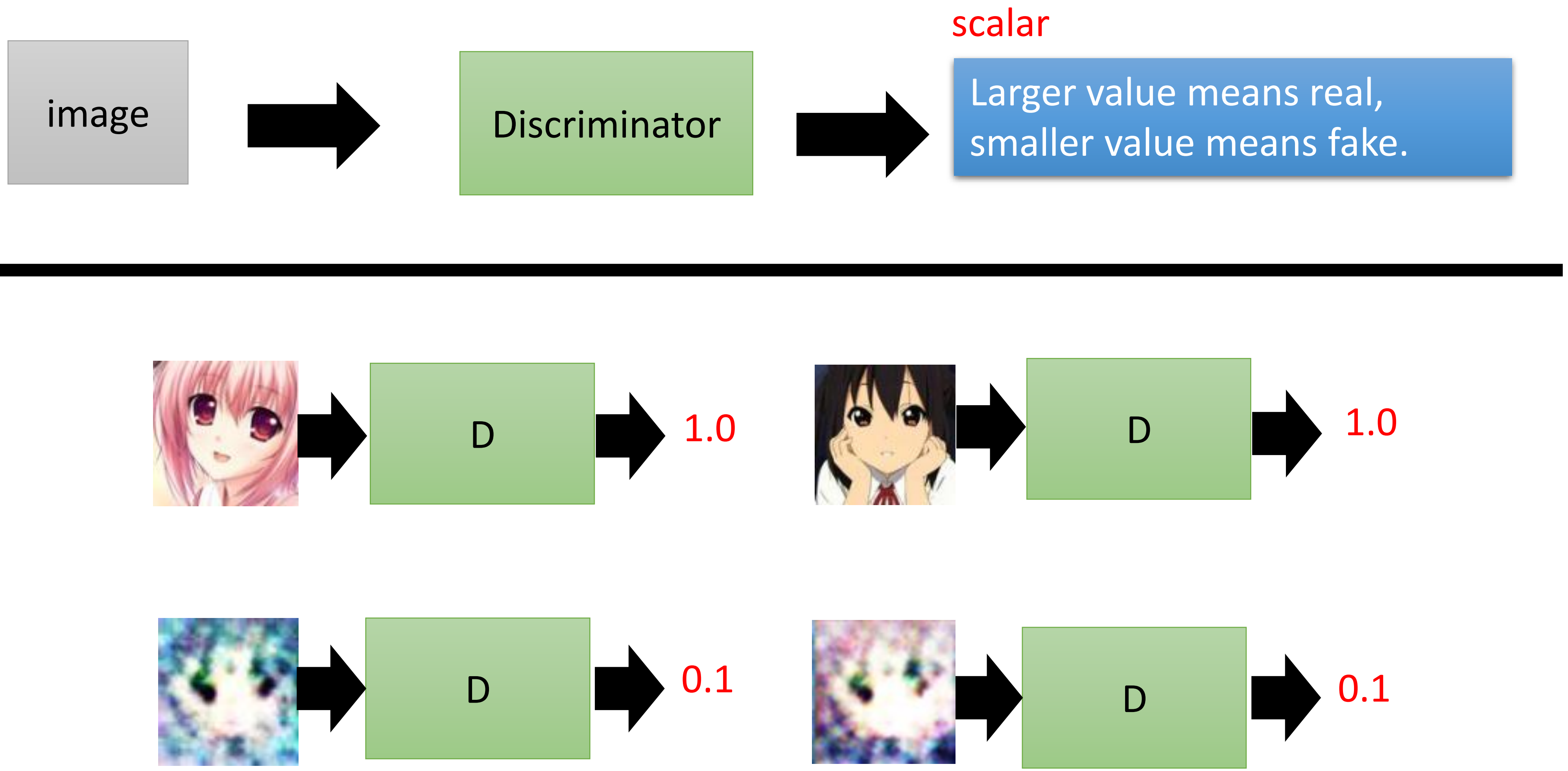
How's going?

Any question?

Genera
Generator

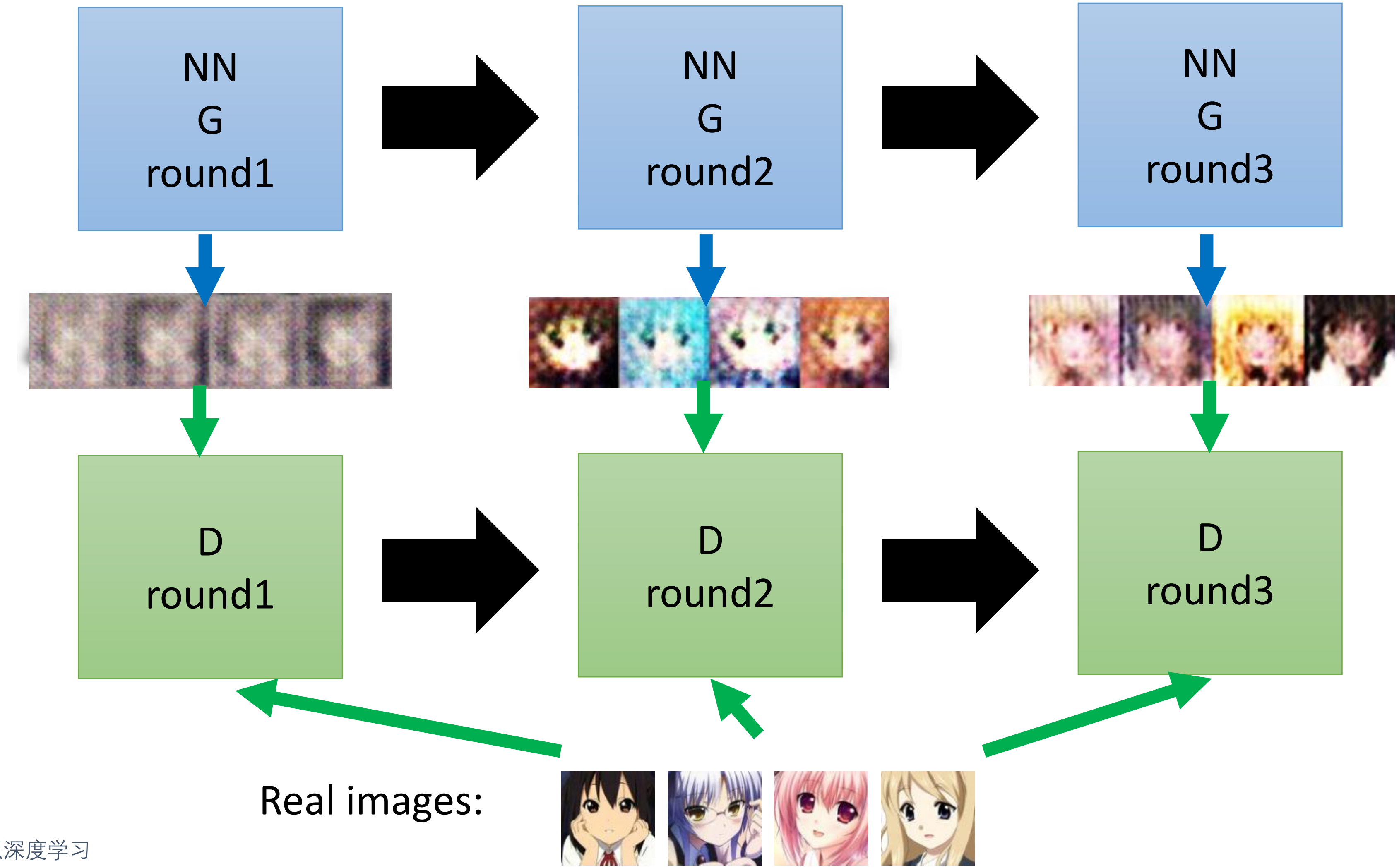


Discriminator



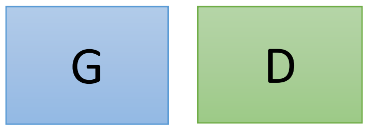
Generator vs Discriminator

Generator vs Discriminator

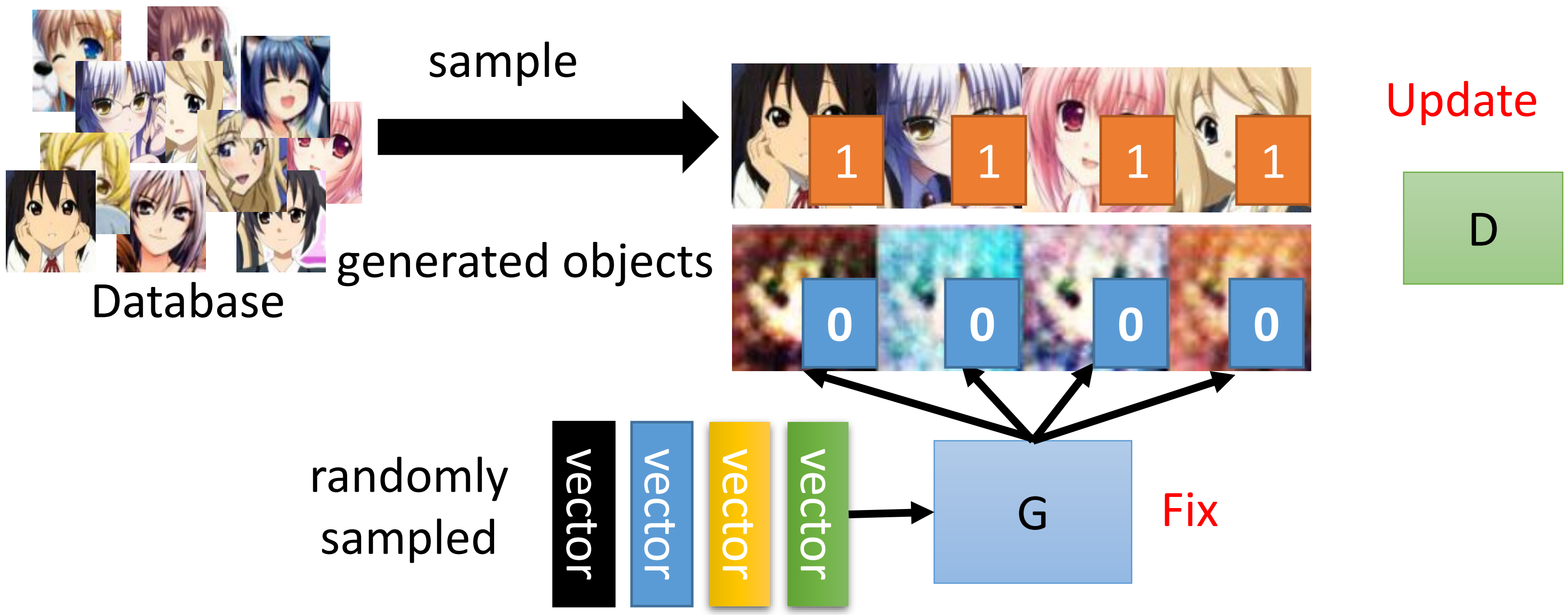


Algorithm

- Initialize generator and discriminator
- In each training iteration:



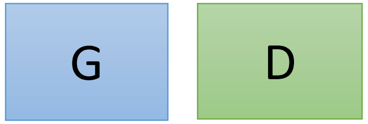
<u>Step 1</u>: Fix generator G, and update discriminator D



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

Algorithm

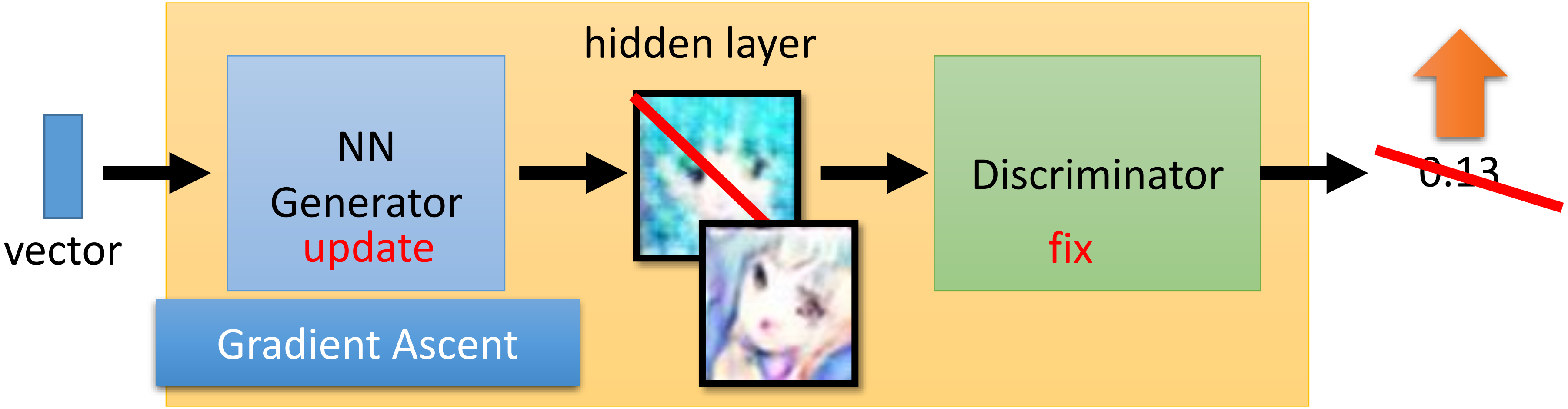
- Initialize generator and discriminator



- In each training iteration:

Step 2: Fix discriminator D, and update generator G

Generator learns to “fool” the discriminator



large network

Algorithm

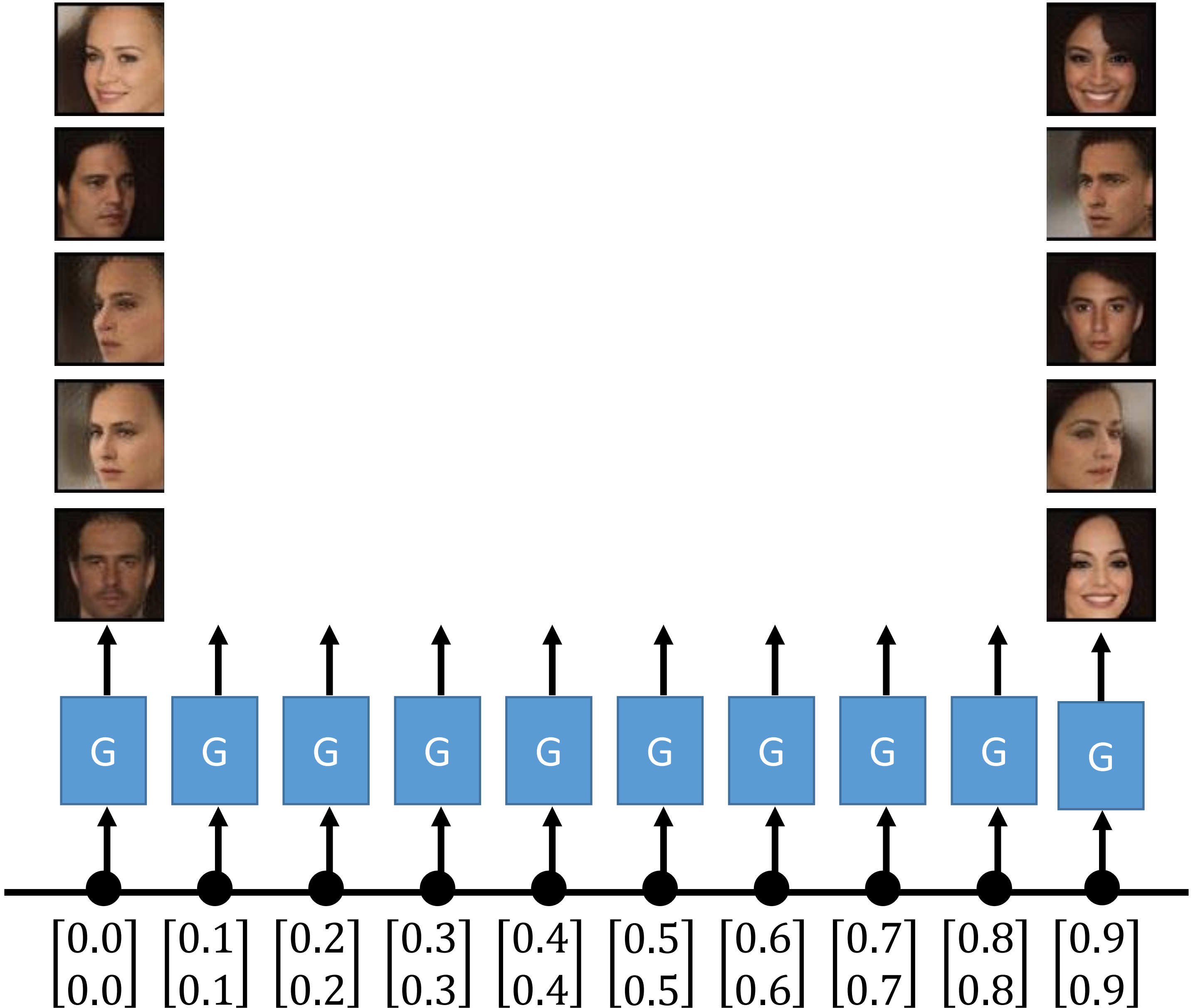
Initialize θ_d for D and θ_g for G

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

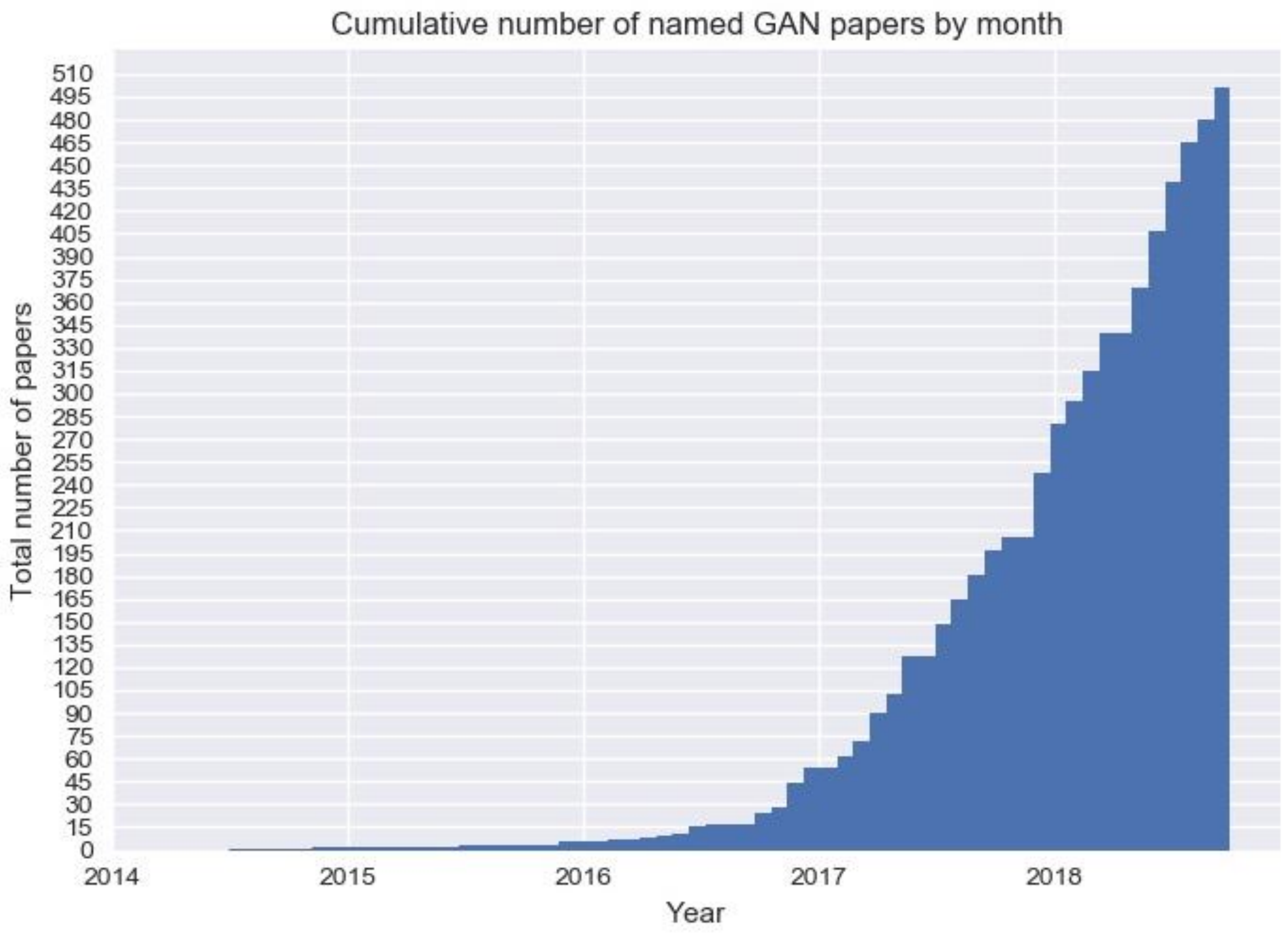
Learning D

Learning G

Generative Adversarial Networks （GANs）



Generative Adversarial Networks （GANs）



GAN ZOO

Q&A