

# Introduction of Generative Adversarial Network (GAN)

Shangsong Liang

Sun Yat-sen University

Produced by Hung-yi Lee

# Generative Adversarial Network (GAN)

- How to pronounce “GAN”?



# Yann LeCun's comment

## What are some recent and potentially upcoming breakthroughs in unsupervised 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 Huang Xiao



Adversarial training is the coolest thing since sliced bread.

I've listed a bunch of relevant papers in a previous answer.

Expect more impressive results with this technique in the coming years.

What's missing at the moment is a good understanding of it so we can make it work reliably. It's very finicky. Sort of like ConvNet were in the 1990s, when I had the reputation of being the only person who could make them work (which wasn't true).

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

# 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 Quiñero 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).

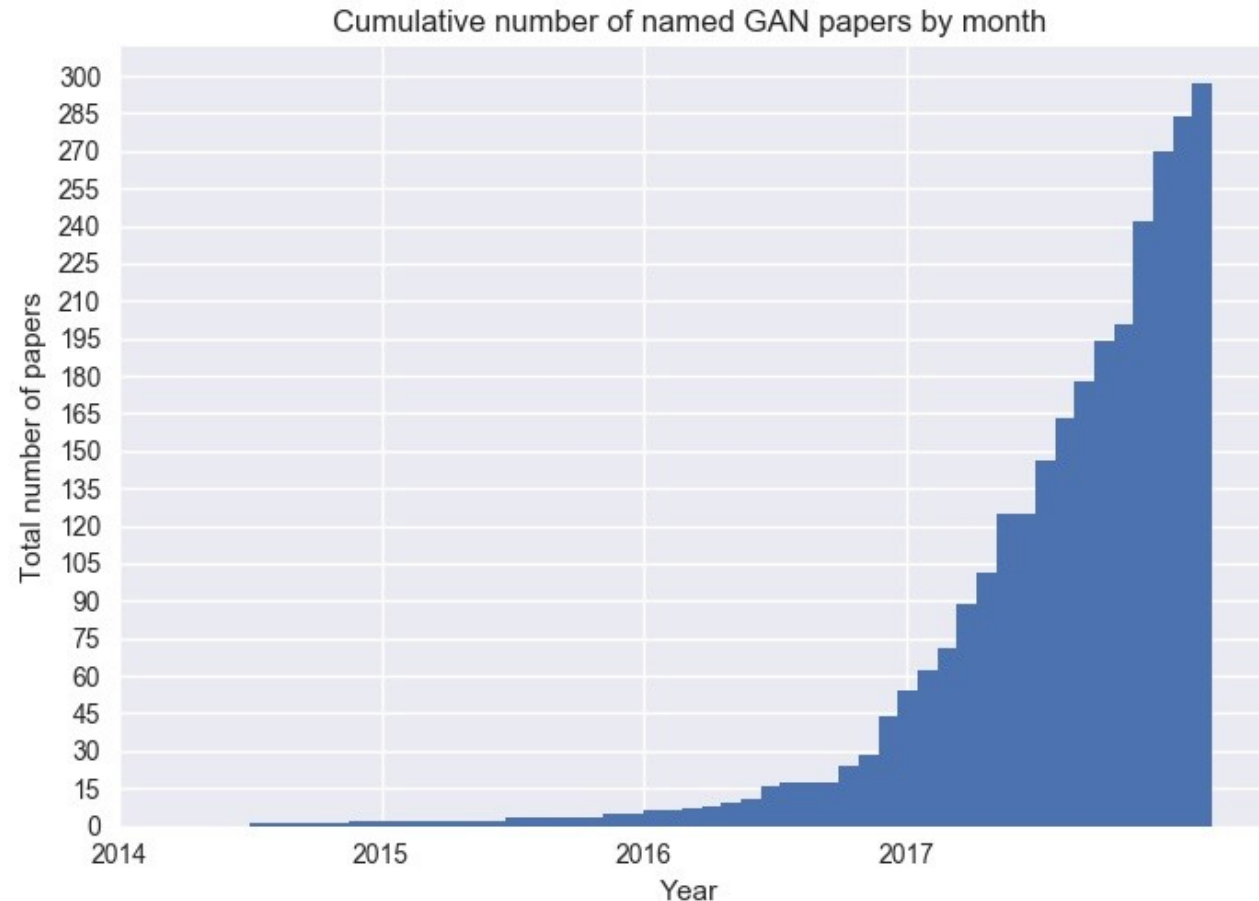
This, and the variations that are now being proposed is the most interesting idea in the last 10 years in ML, in my opinion.

<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>

GAN  
ACGAN  
BGAN  
CGAN  
DCGAN  
EBGAN  
fGAN  
GoGAN  
⋮  
⋮

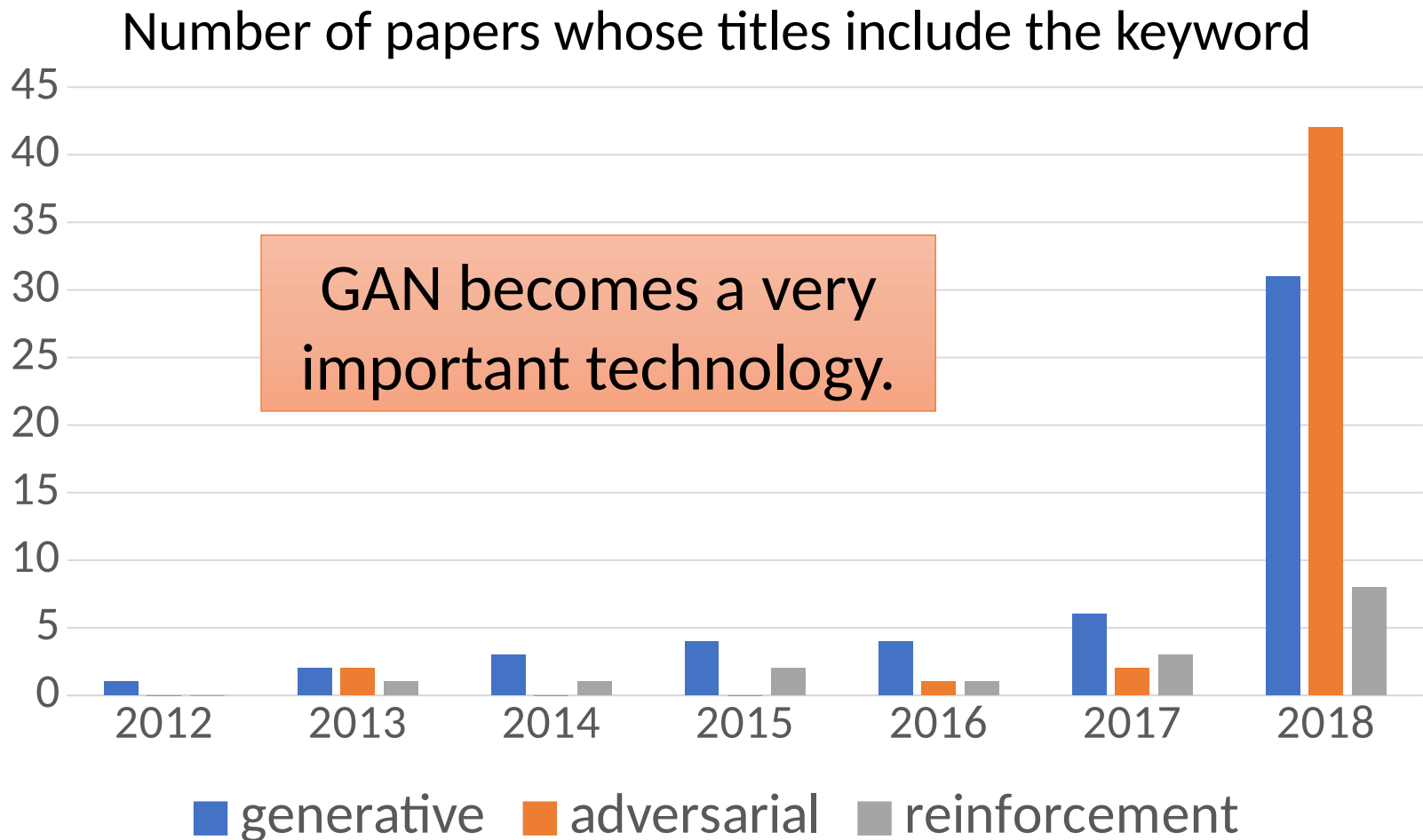


Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, Shakir Mohamed, “Variational Approaches for Auto-Encoding Generative Adversarial Networks”, arXiv, 2017

<sup>2</sup>We use the Greek  $\alpha$  prefix for  $\alpha$ -GAN, as AEGAN and most other Latin prefixes seem to have been taken  
<https://deephunt.in/the-gan-zoo-79597dc8c347>.

# ICASSP

Keyword search on session index page, so session names are included.



# Outline

Basic Idea of GAN

GAN as structured learning

Can Generator learn by itself?

Can Discriminator generate?

A little bit theory

# Generation

We will control what to generate  
later. Conditional 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



## Sentence Generation

$$\begin{bmatrix} 0.3 \\ -0.1 \\ \vdots \\ -0.7 \end{bmatrix} \begin{bmatrix} 0.1 \\ -0.1 \\ \vdots \\ 0.2 \end{bmatrix} \begin{bmatrix} -0.3 \\ 0.1 \\ \vdots \\ 0.5 \end{bmatrix}$$



NN  
Generator

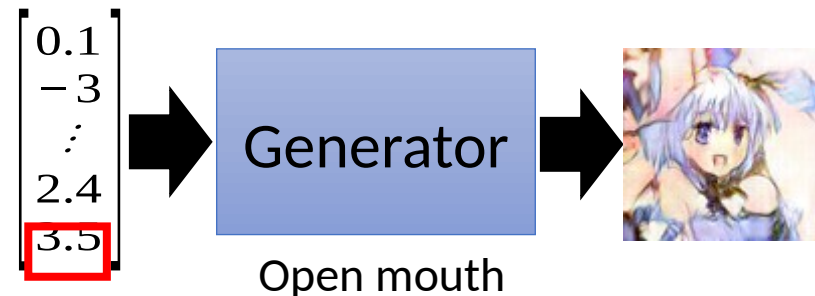
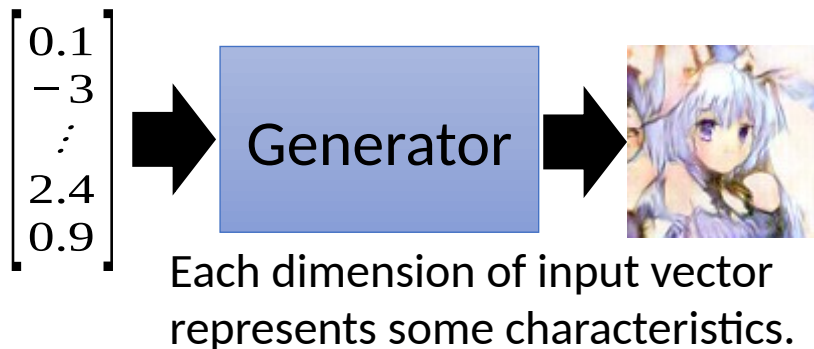
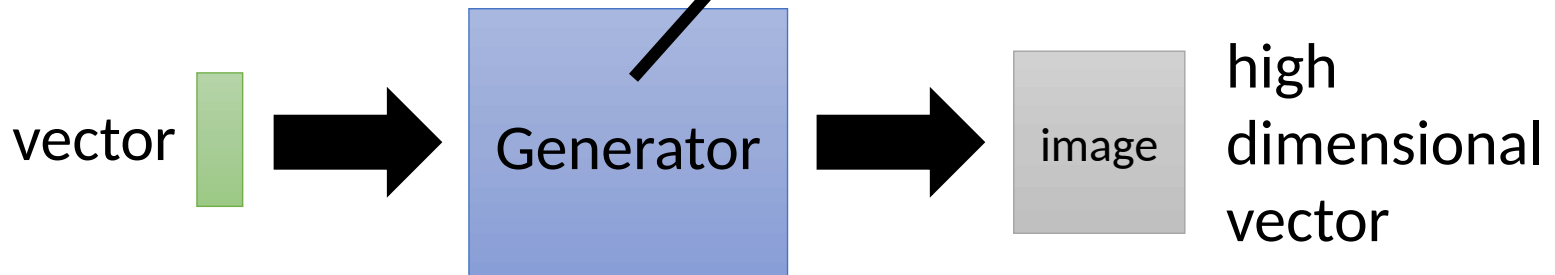


How are you?  
Good morning.  
Good afternoon.



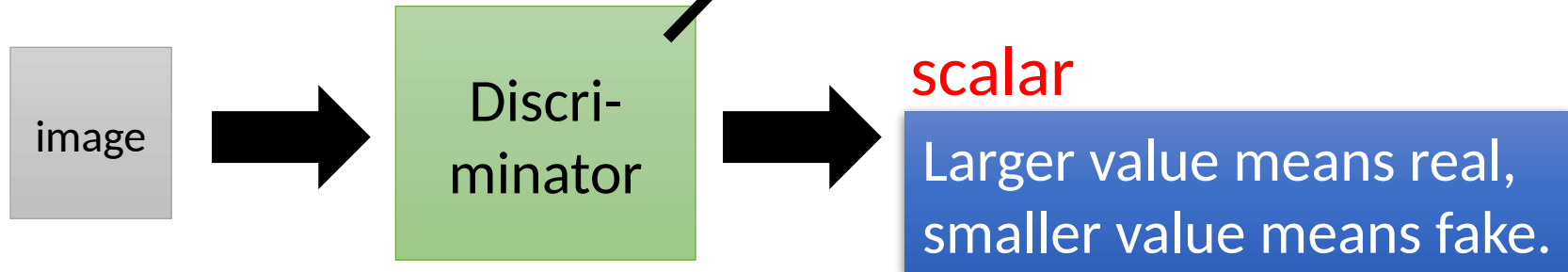
# Basic Idea of GAN

It is a neural network (NN), or a function.

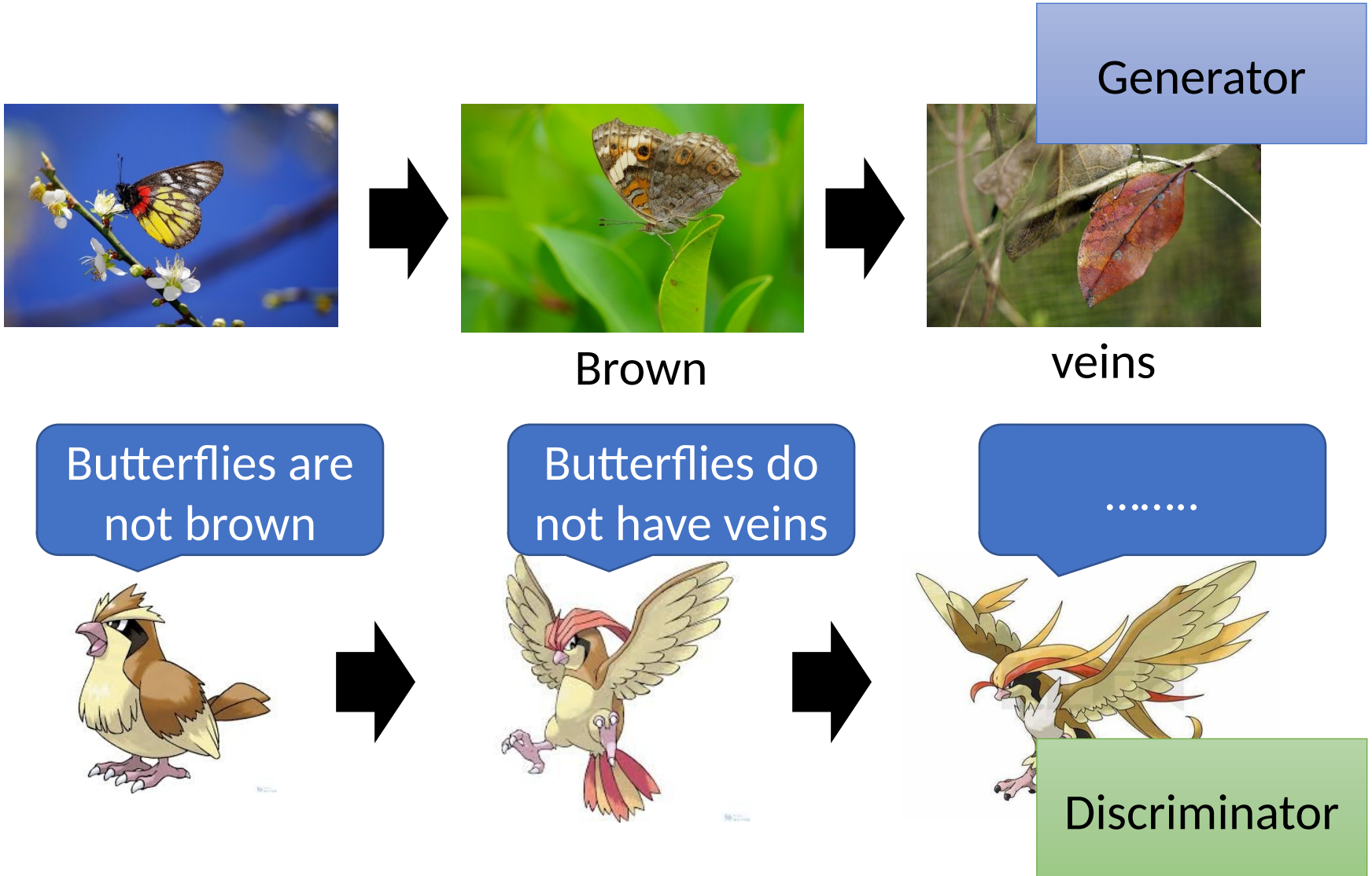


# Basic Idea of GAN

It is a neural network (NN), or a function.

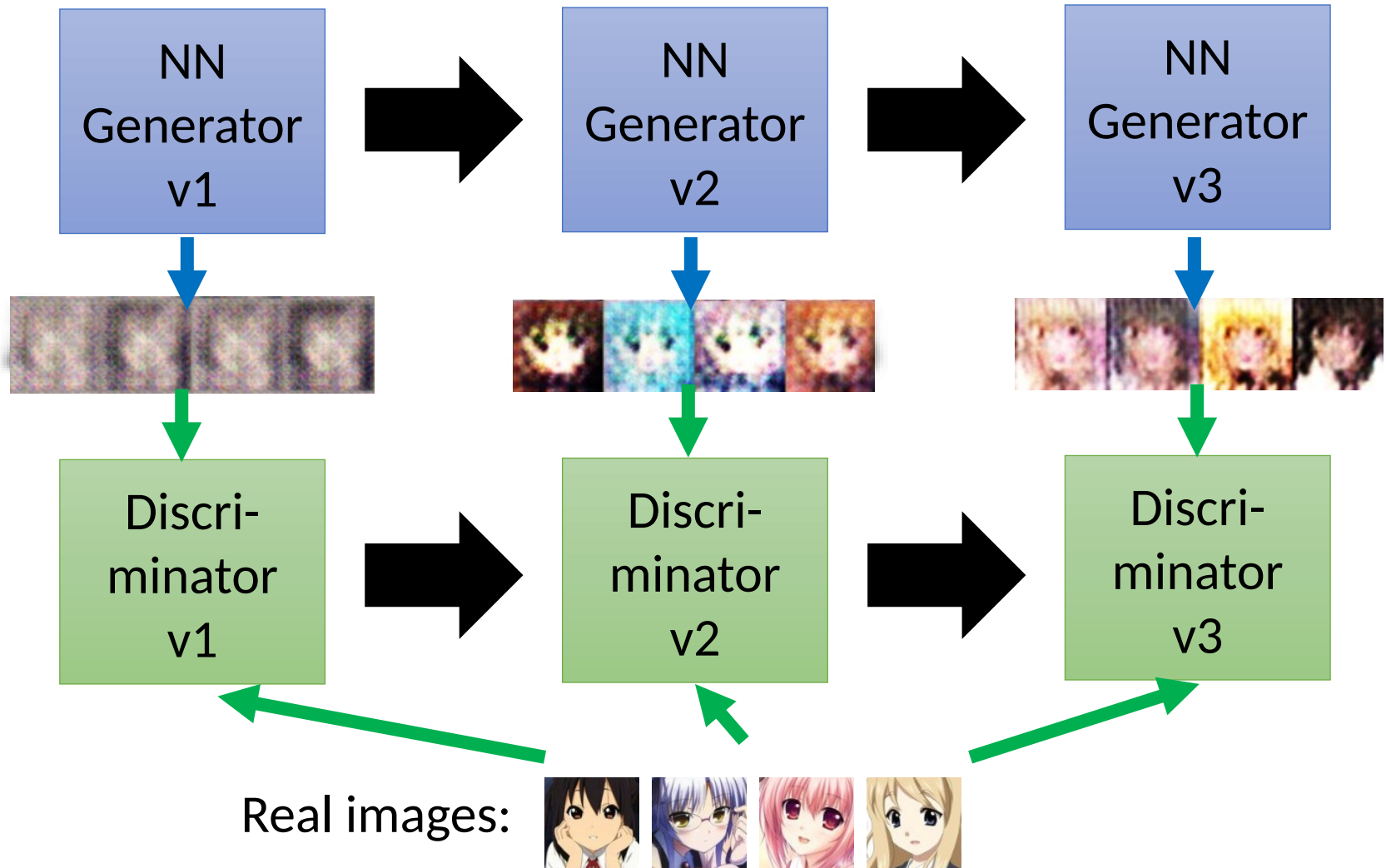


# Basic Idea of GAN



# Basic Idea of GAN

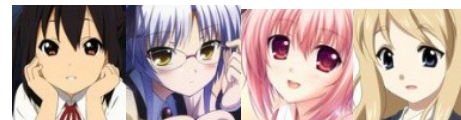
This is where the term  
“*adversarial*” comes  
from. You can explain the process  
in different ways.....



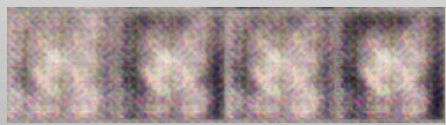
# Basic Idea of GAN (和平的比喻)

Generator  
(student)

Discriminator  
(teacher)



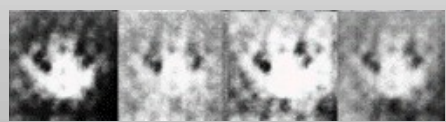
Generator  
v1



Discriminator  
v1

沒有兩個圈

Generator  
v2



Discriminator  
v2

沒有彩色

Generator  
v3



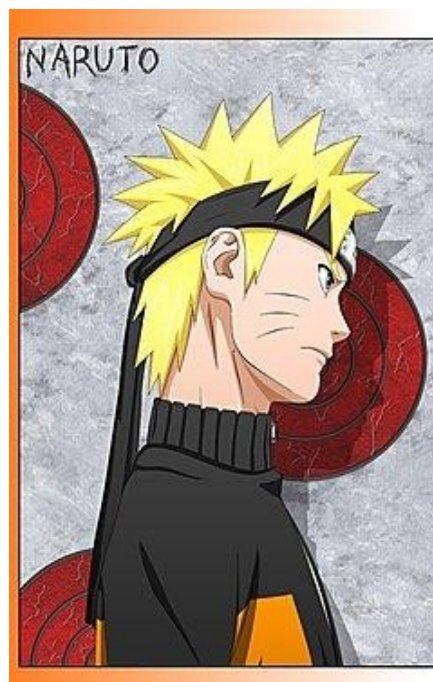
為什麼不自己學？

為什麼不自己做？



# Generator v.s. Discriminator

- 寫作敵人，唸做朋友

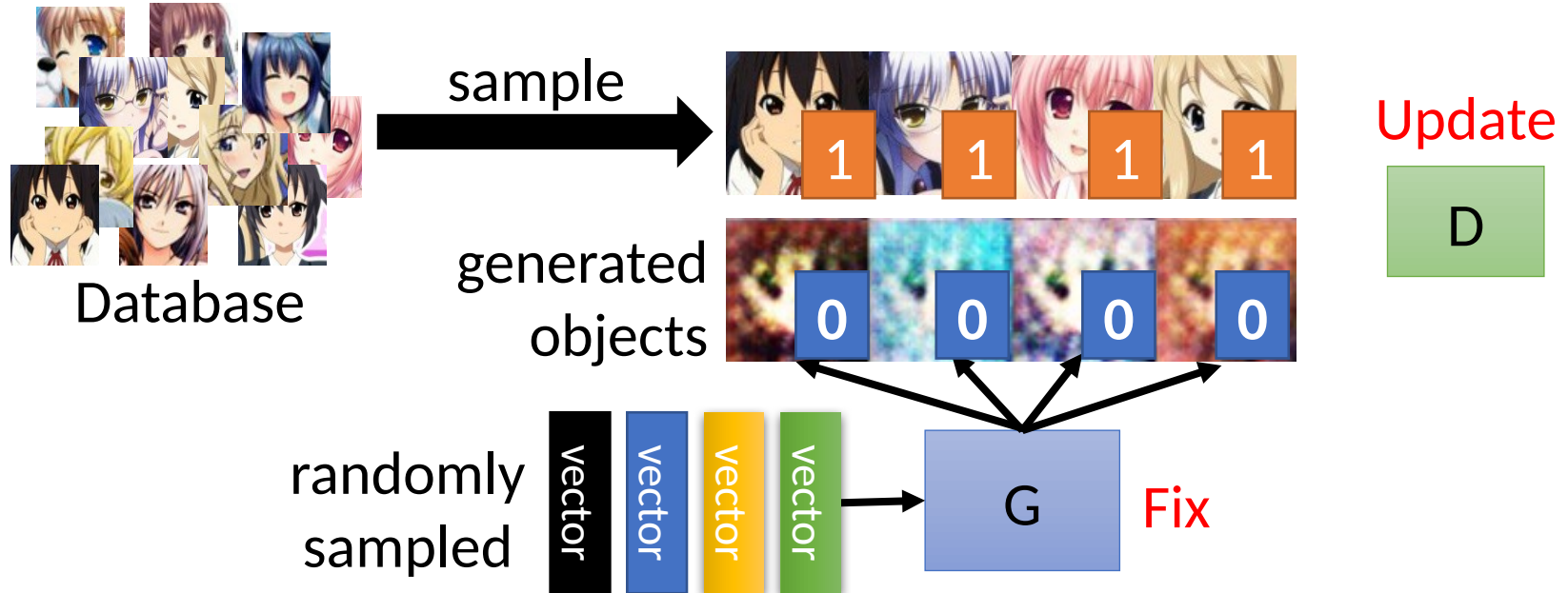


# Algorithm

- 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.

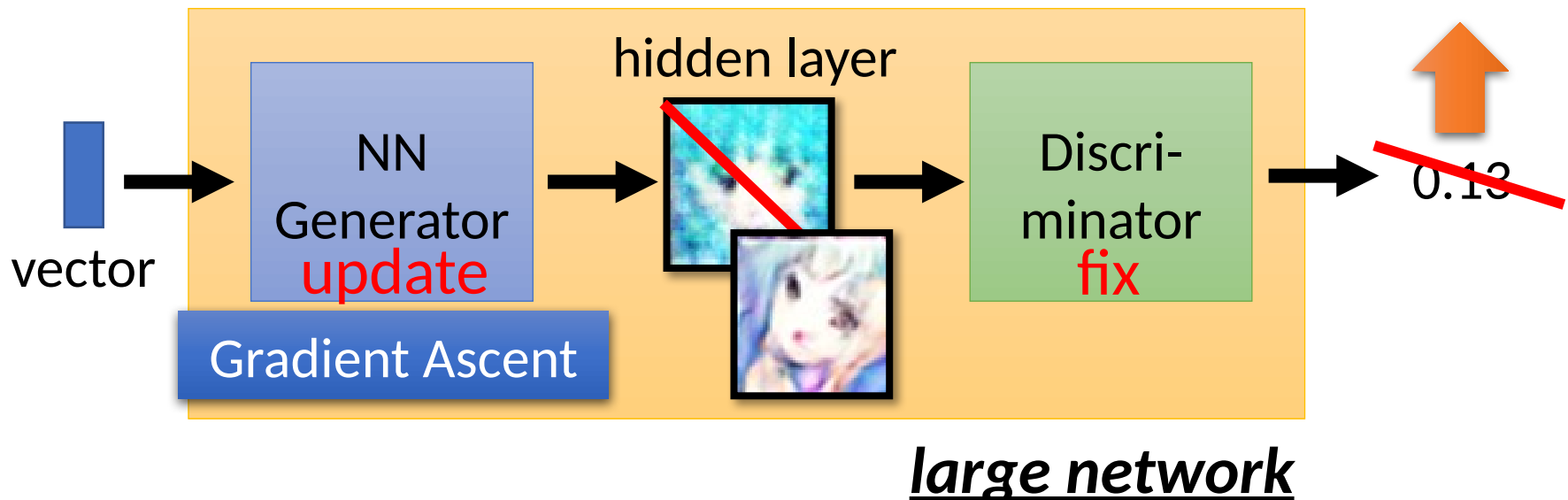
# Algorithm

- Initialize generator and discriminator
- In each training iteration:



**Step 2**: Fix discriminator D, and update generator G

Generator learns to “fool” the discriminator





# Algorithm Initialize for D and for G

- In each training iteration:

- Sample  $m$  examples from database
- Sample  $m$  noise samples from a distribution
- Obtaining generated data ,
- Update discriminator parameters to maximize

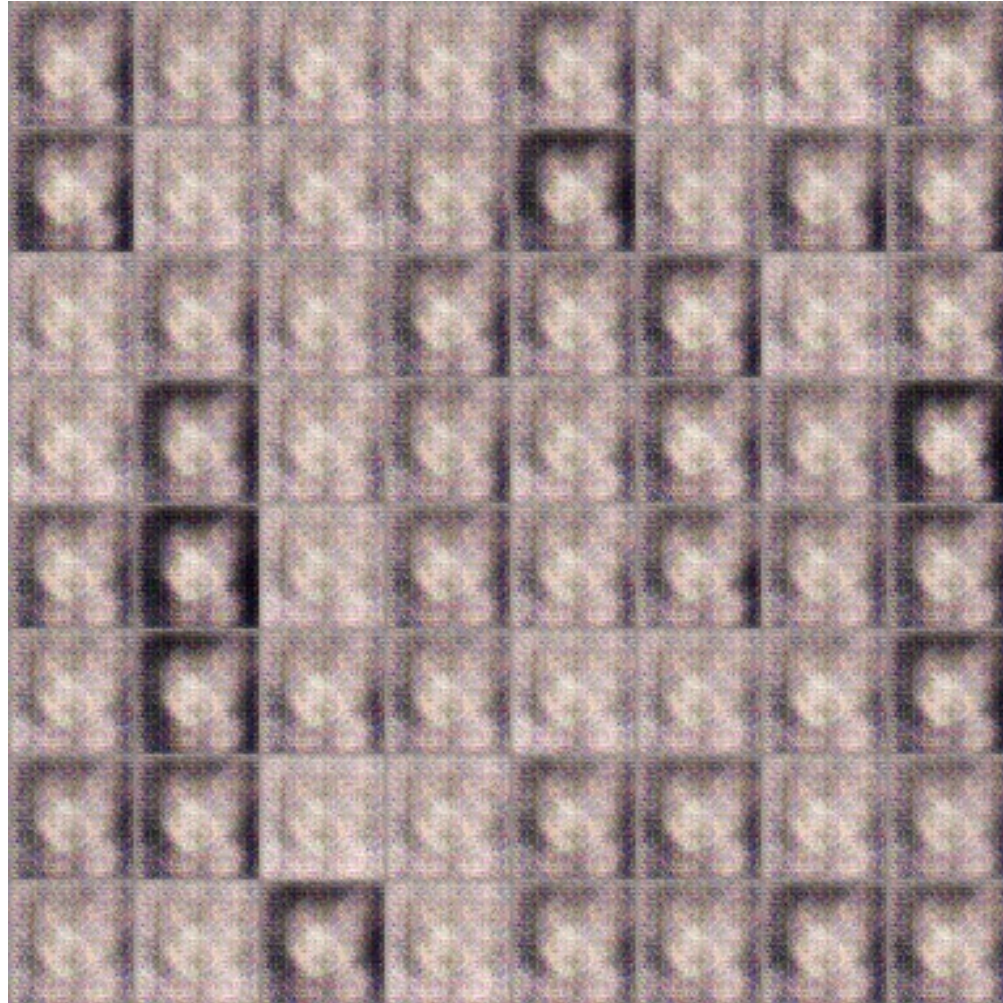
- Sample  $m$  noise samples from a distribution
- Update generator parameters to minimize

Learning  
D

Learning  
G

# Anime Face Generation

100 updates



Source of training data: <https://zhuanlan.zhihu.com/p/24767059>

# Anime Face Generation



1000 updates



# Anime Face Generation



2000 updates

# Anime Face Generation



5000 updates



# Anime Face Generation



10,000 updates

# Anime Face Generation



20,000 updates



# Anime Face Generation



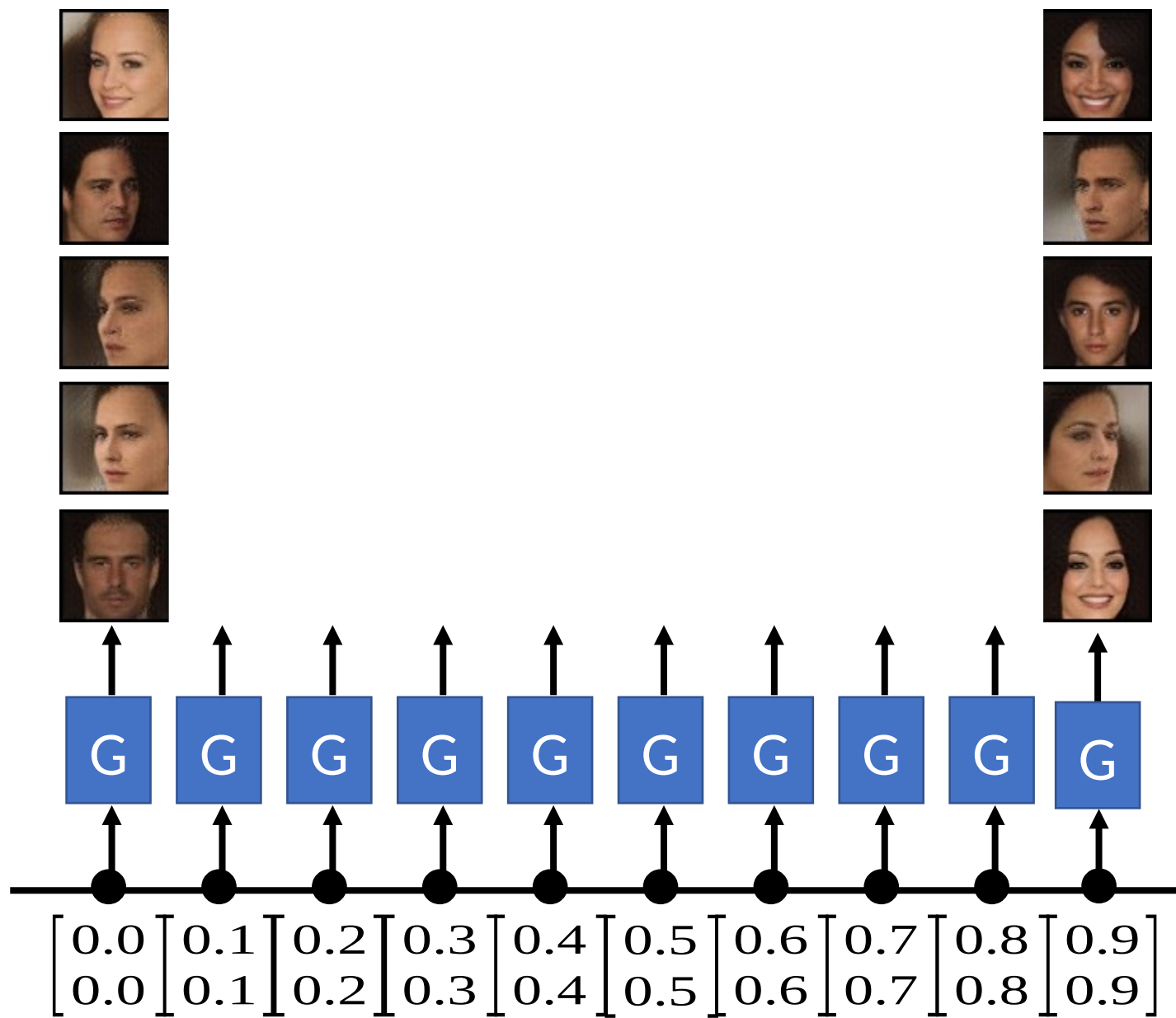
50,000 updates





The faces  
generated by  
machine.

圖片生成：  
吳宗翰、謝濬丞、  
陳延昊、錢柏均



感謝陳柏文同學提供實驗結果

# Outline

Basic Idea of GAN

GAN as structured learning

Can Generator learn by itself?

Can Discriminator generate?

A little bit theory

# Structured Learning

Machine learning is to find a function  $f$

$$f : X \rightarrow Y$$

**Regression:** output a scalar

**Classification:** output a “class” (one-hot vector)

1	0	0
---	---	---

Class 1

0	1	0
---	---	---

Class 2

0	0	1
---	---	---

Class 3

**Structured Learning/Prediction:** output a sequence, a matrix, a graph, a tree .....

Output is composed of components with dependency

# Output Sequence

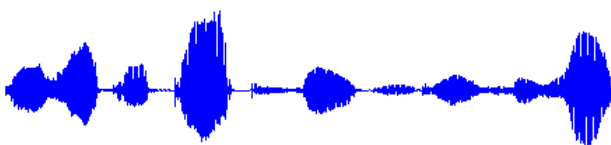
$$f : X \rightarrow Y$$

## Machine Translation

$X$  : “機器學習及其深層與結構化”  
(sentence of language 1)

$Y$  : “Machine learning and having it deep and structured”  
(sentence of language 2)

## Speech Recognition

$X$  :   
(speech)

$Y$  : 感謝大家來上課”  
(transcription  
)

## Chat-bot

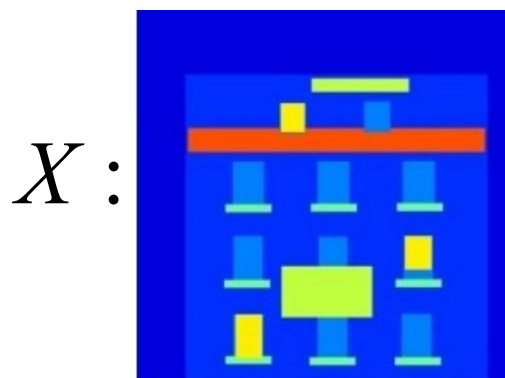
$X$  : “How are you?”  
(what a user says)

$Y$  : “I’m fine.”  
(response of machine)

# Output Matrix

$$f : X \rightarrow Y$$

## Image to Image



## Colorization:



Ref: <https://arxiv.org/pdf/1611.07004v1.pdf>

## Text to Image

$X :$  “this white and yellow flower have thin white petals and a round yellow stamen”

$Y :$



ref: <https://arxiv.org/pdf/1605.05396.pdf>

# Why Structured Learning Challenging?

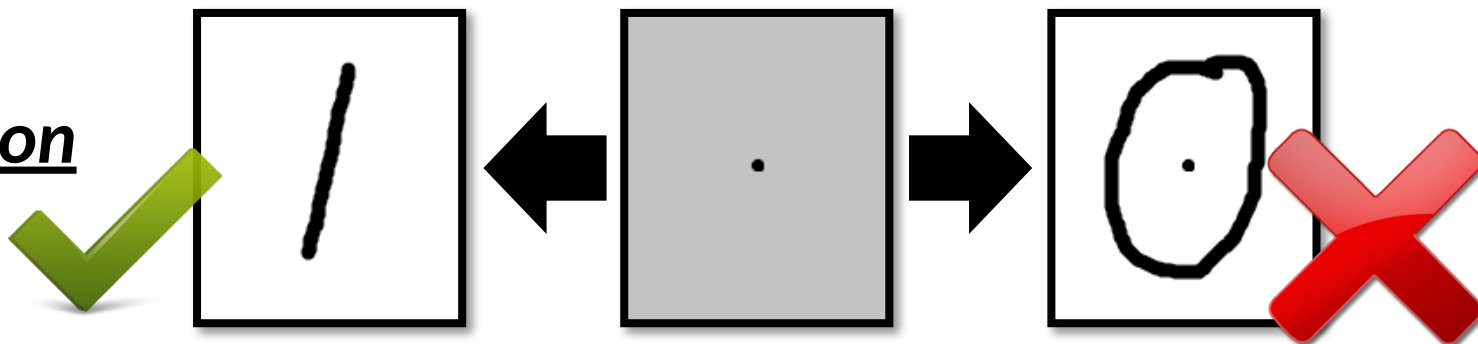
- **One-shot/Zero-shot Learning:**
  - In classification, each class has some examples.
  - In structured learning,
    - If you consider each possible output as a “class” .....
    - Since the output space is huge, most “classes” do not have any training data.
    - Machine has to create new stuff during testing.
    - Need more intelligence




# Why Structured Learning Challenging?

- Machine has to learn to do **planning**
  - Machine generates objects component-by-component, but it should have a big picture in its mind.
  - Because the output components have dependency, they should be considered globally.

Image  
Generation



Sentence  
Generation

這個婆娘不是人 

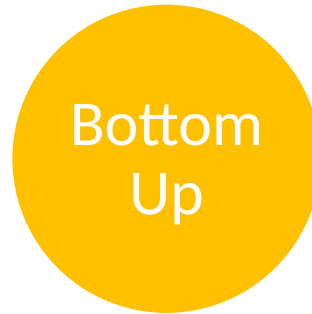
九天玄女下凡塵 



# Structured Learning Approach

## **Generator**

Learn to generate the object at the component level



## **Discriminator**

Evaluating the whole object, and find the best one



# Outline

Basic Idea of GAN

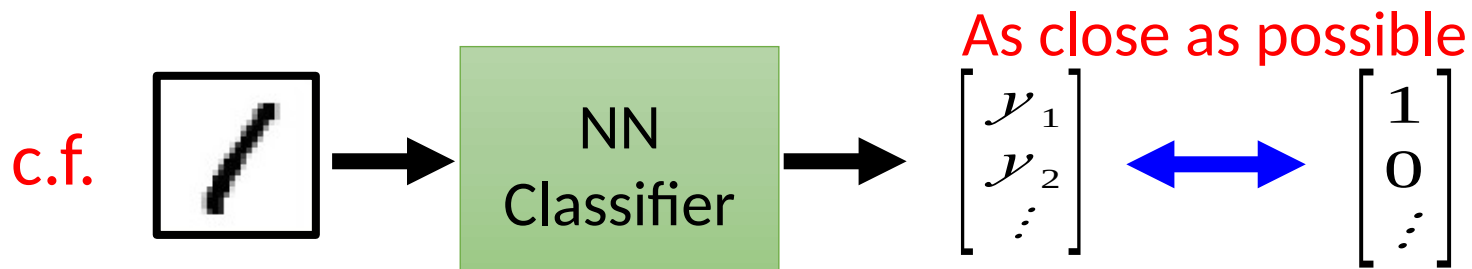
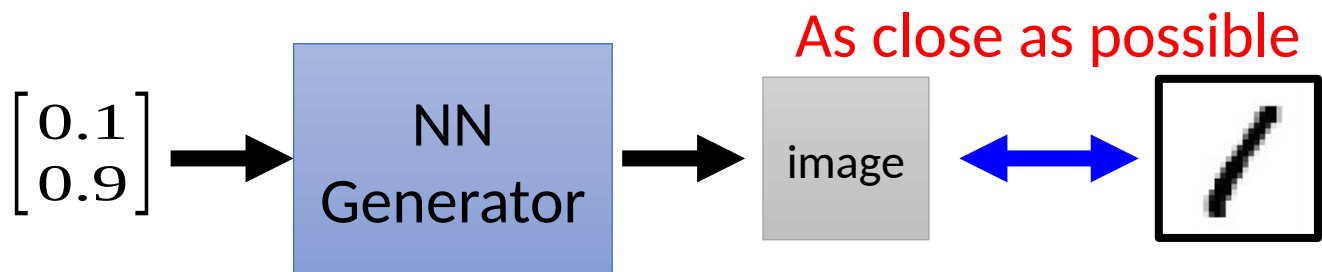
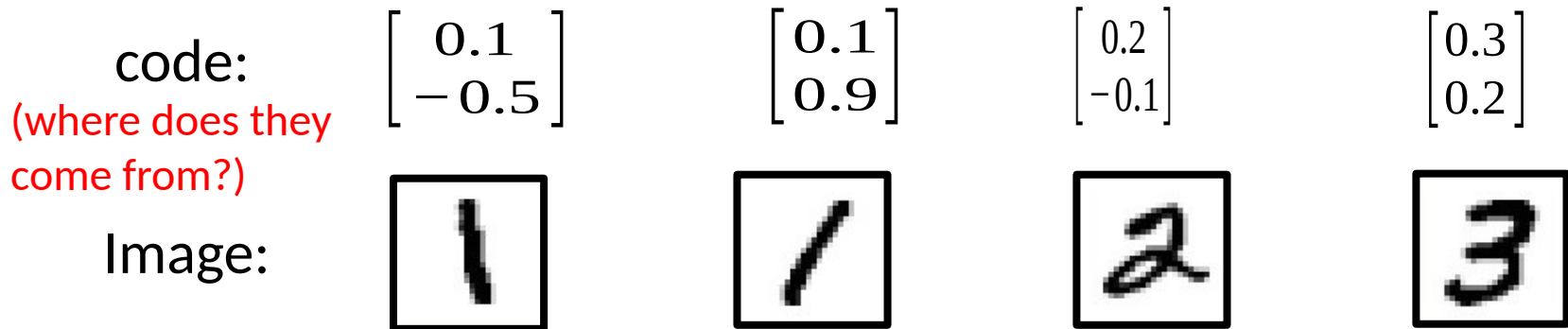
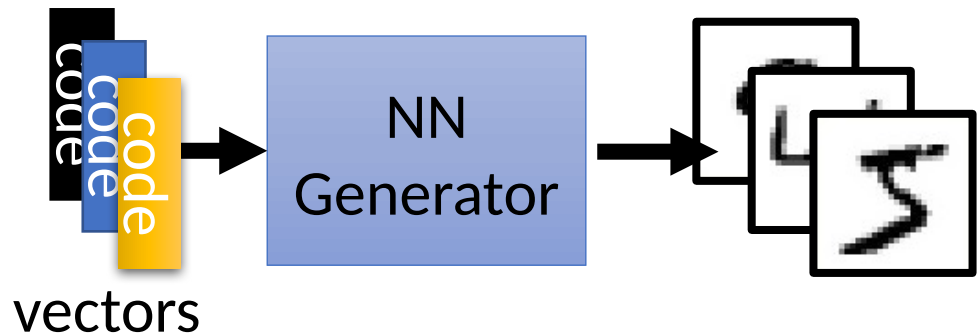
GAN as structured learning

Can Generator learn by itself?

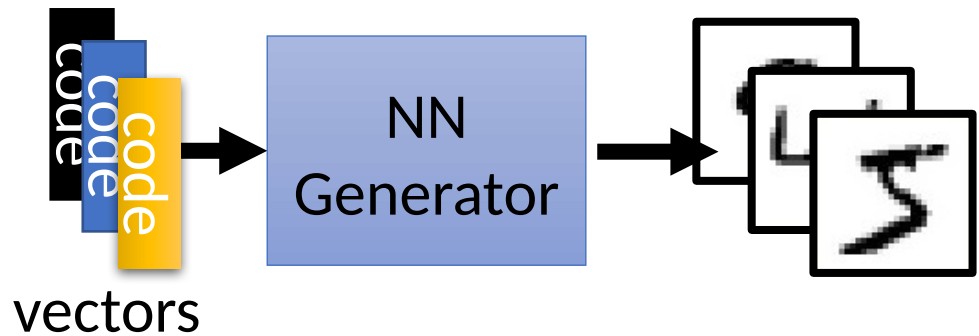
Can Discriminator generate?

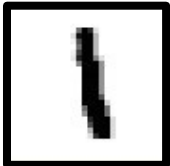



A little bit theory

# Generator

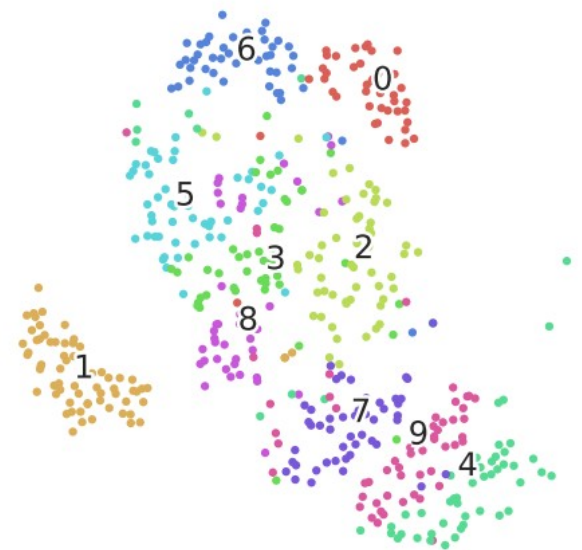
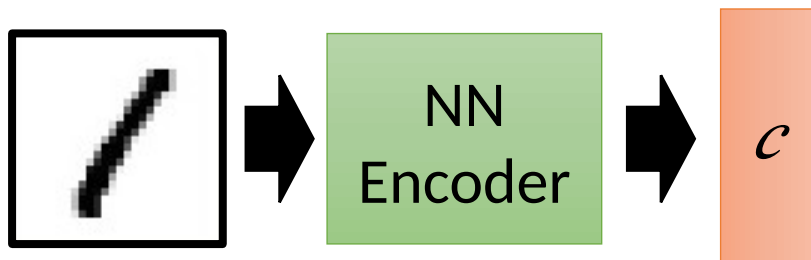


# Generator

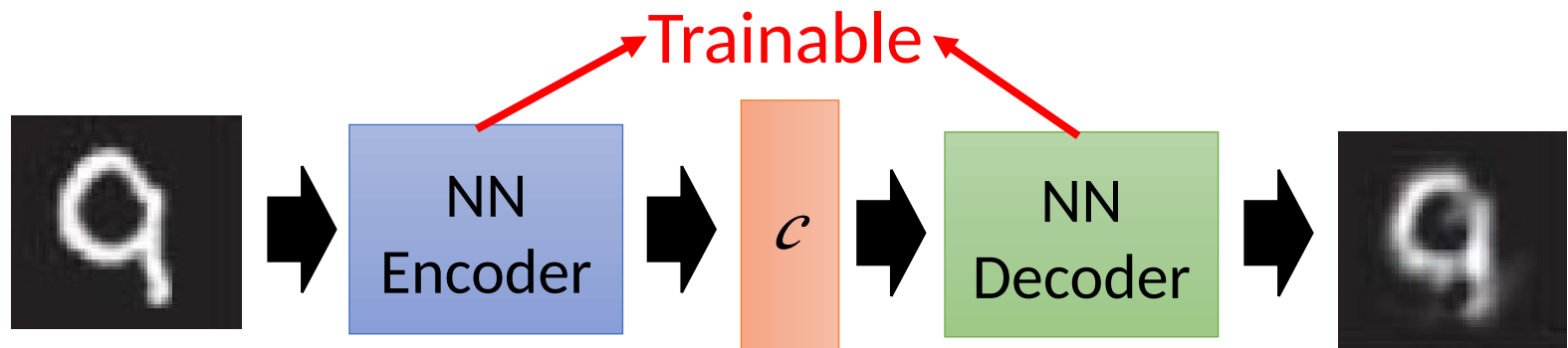
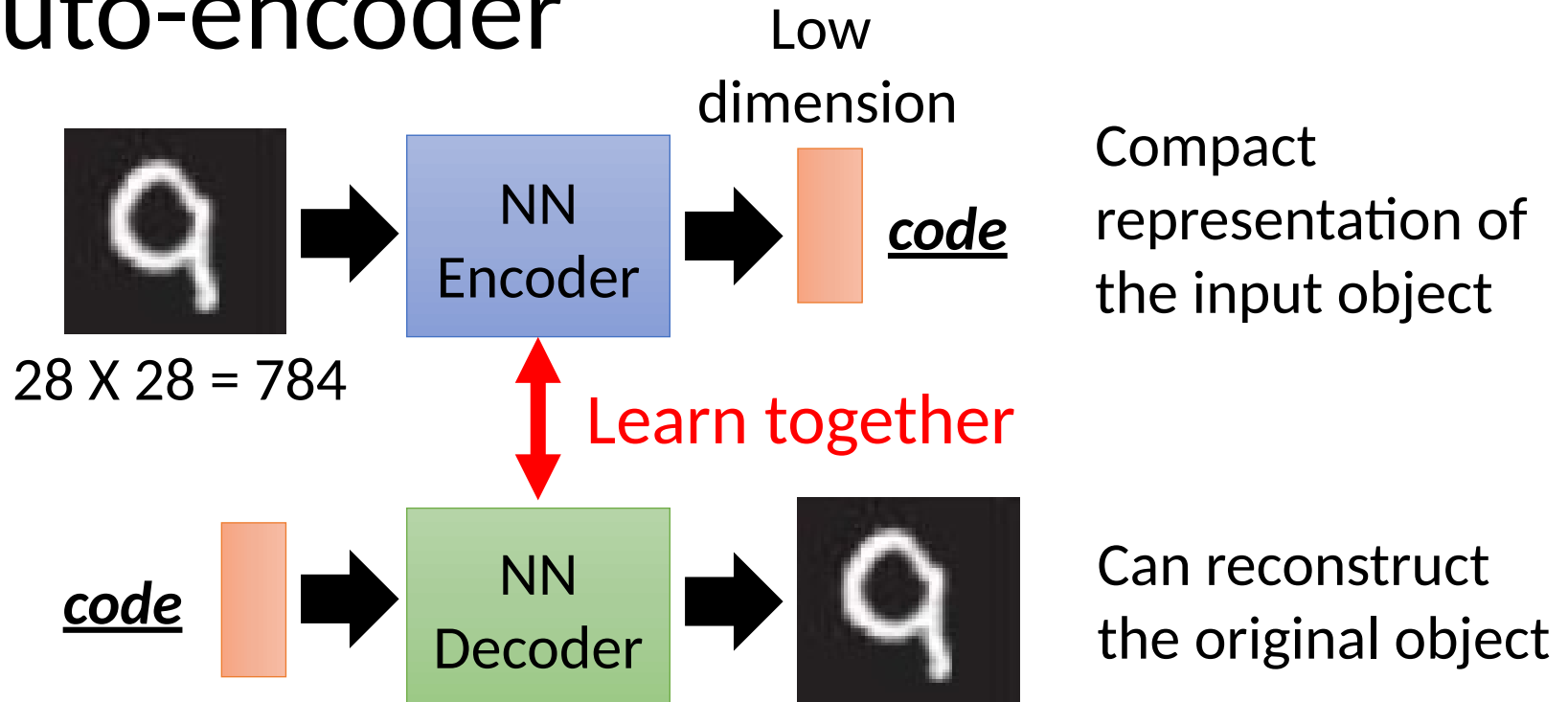


code: (where does they come from?)	$\begin{bmatrix} 0.1 \\ -0.5 \end{bmatrix}$	$\begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}$	$\begin{bmatrix} 0.2 \\ -0.1 \end{bmatrix}$	$\begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}$
Image:				

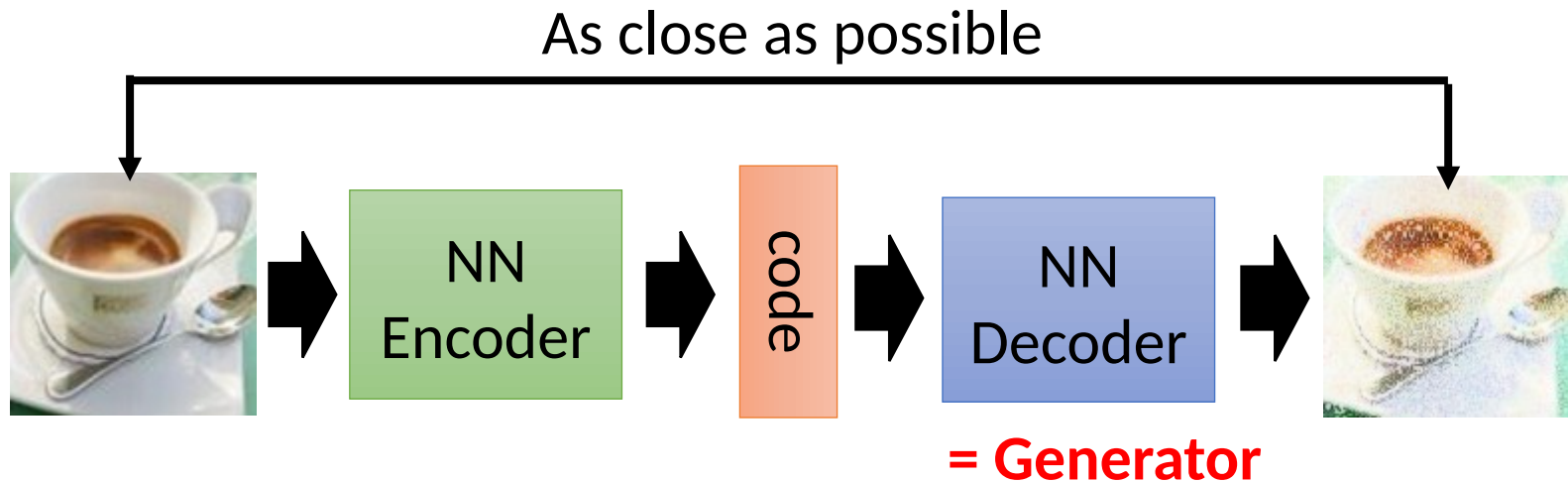
Encoder in auto-encoder  
provides the code 😊



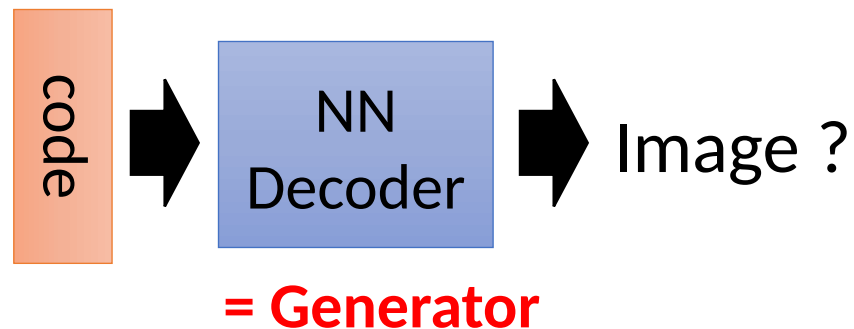
# Auto-encoder



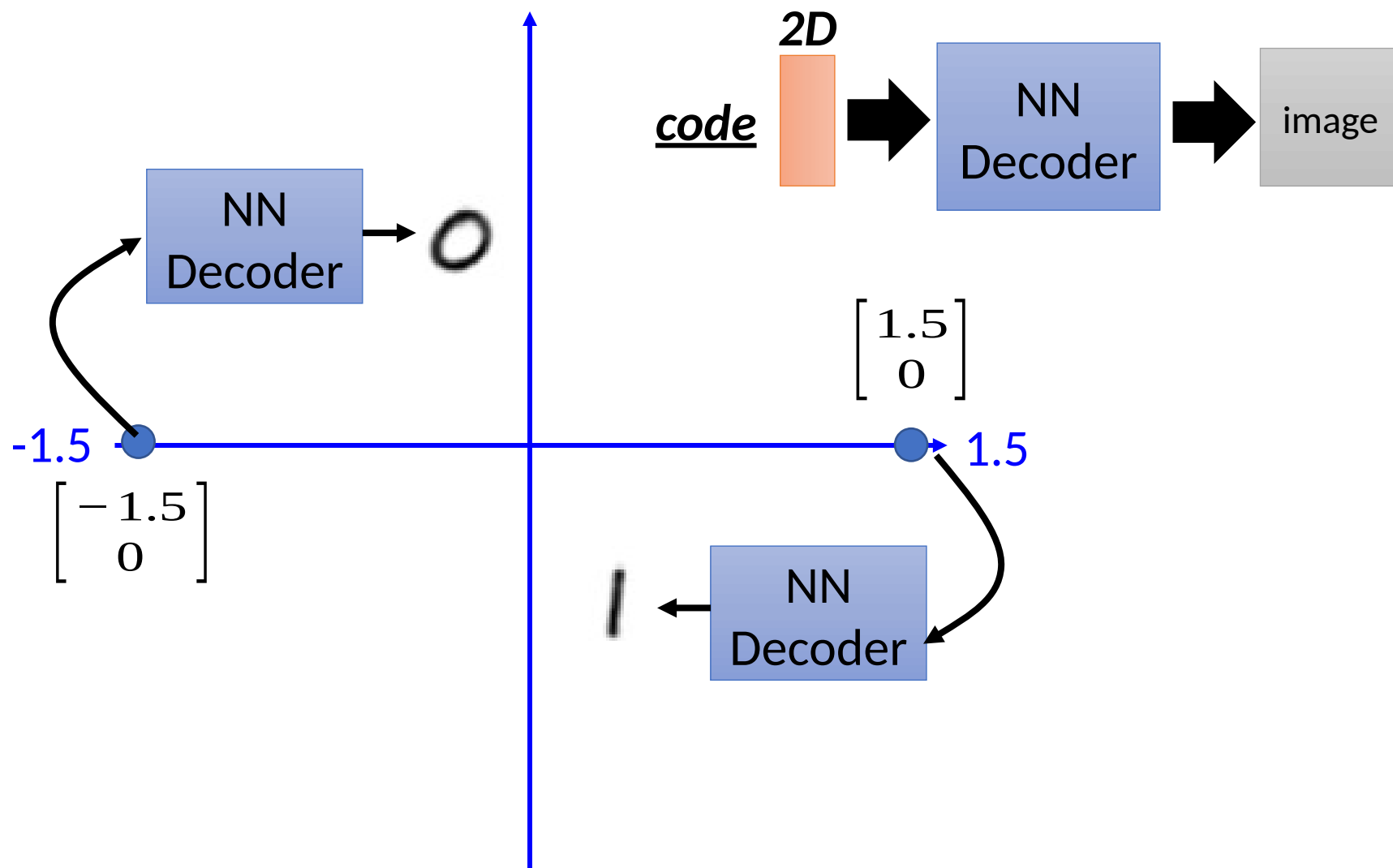
# Auto-encoder



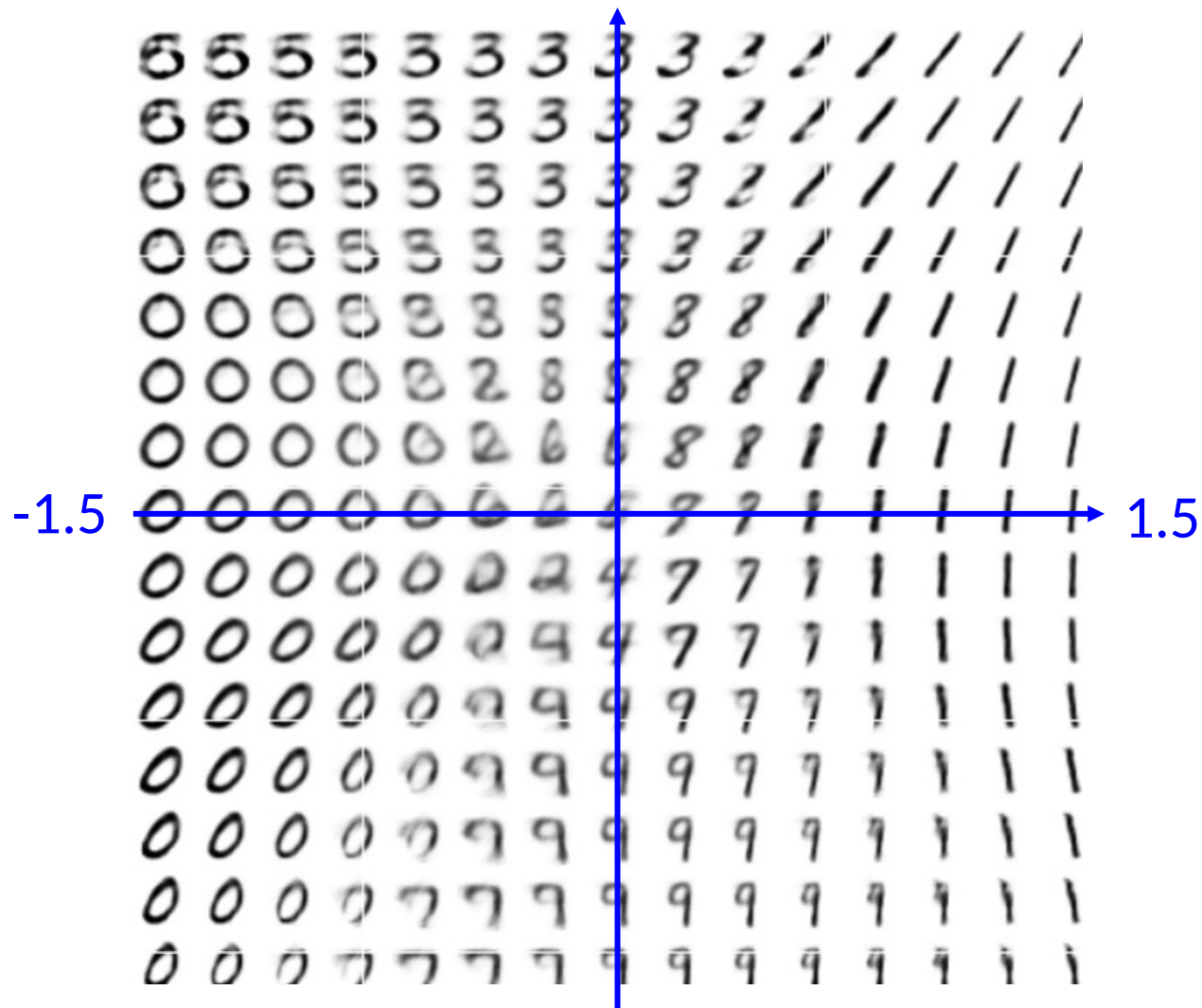
Randomly generate  
a vector as code



# Auto-encoder

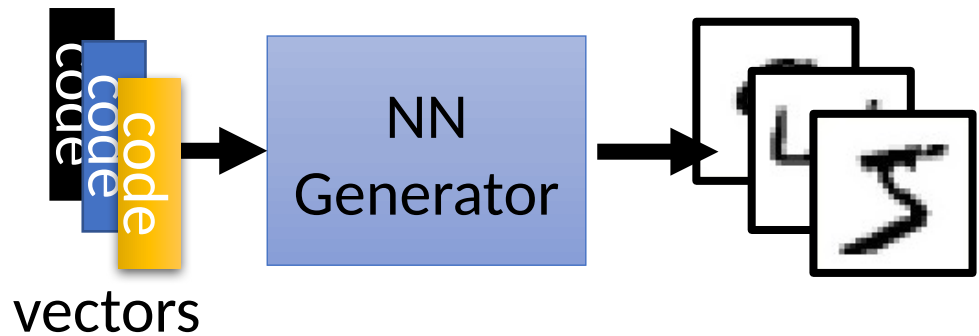


# Auto-encoder





# Auto-encoder



code: (where does them come from?)

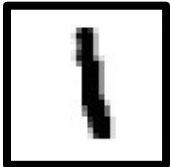



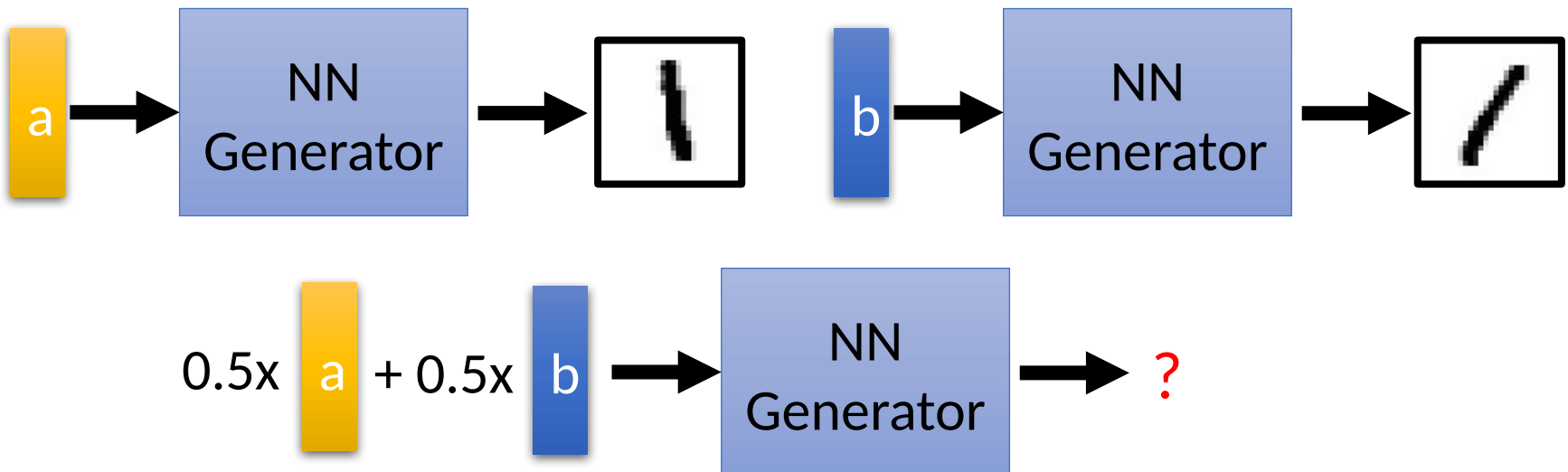
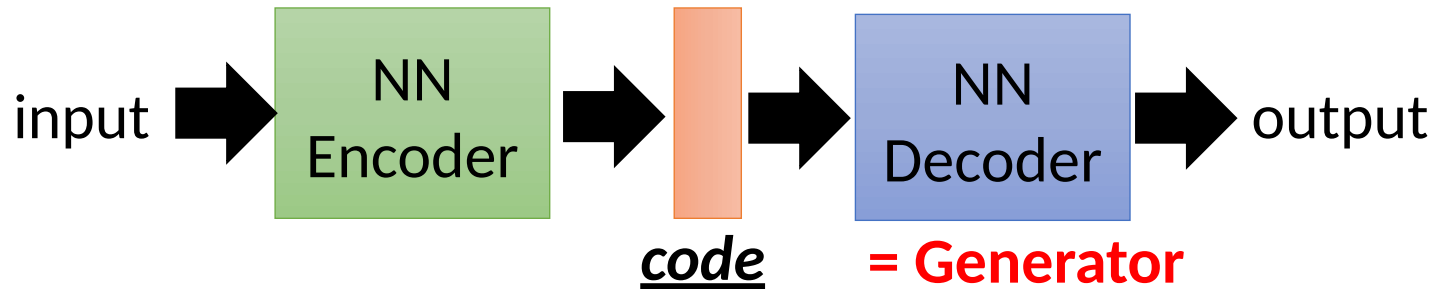
$\begin{bmatrix} 0.1 \\ -0.5 \end{bmatrix}$	$\begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}$	$\begin{bmatrix} 0.2 \\ -0.1 \end{bmatrix}$	$\begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}$
			

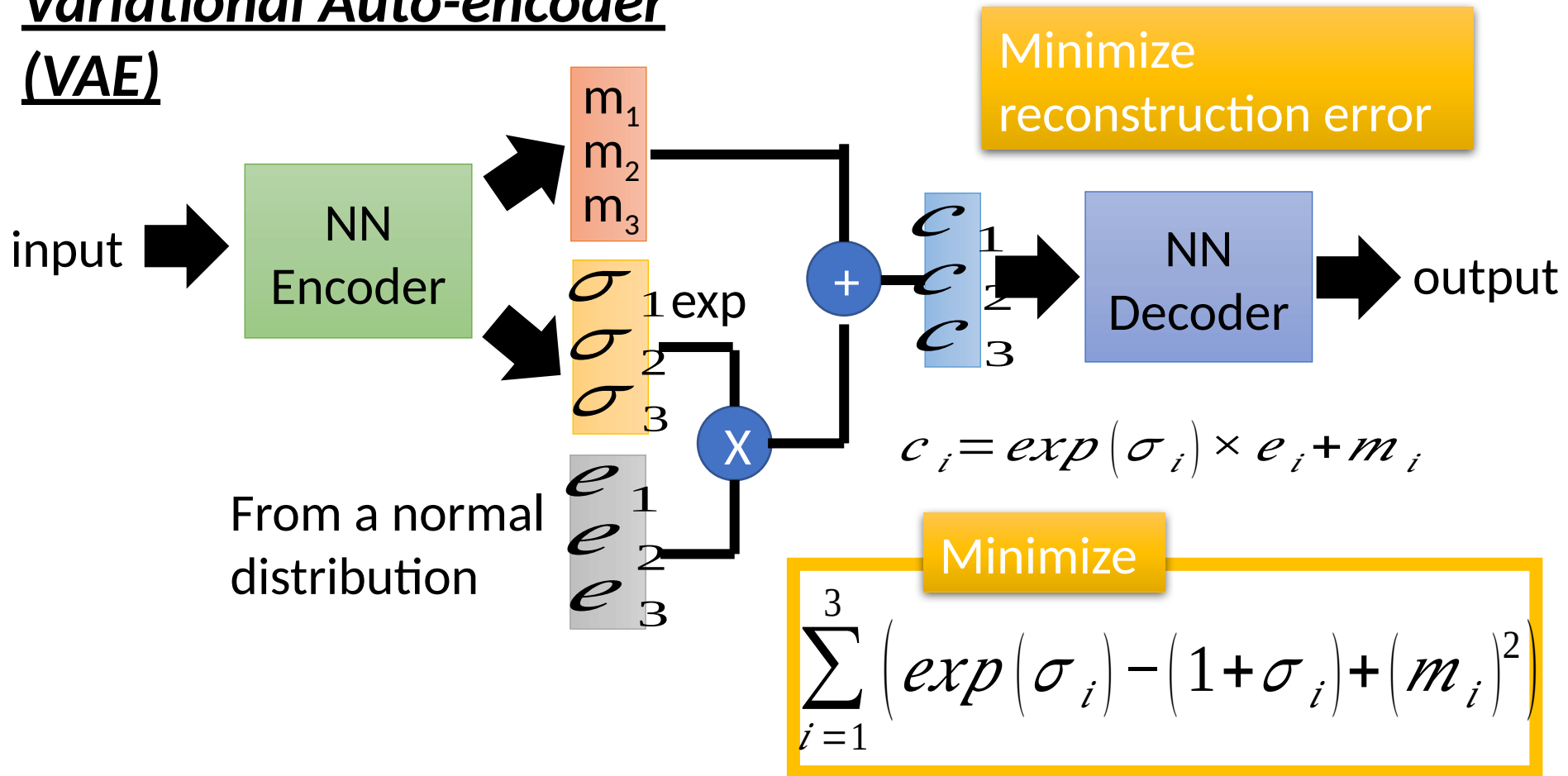
Image:



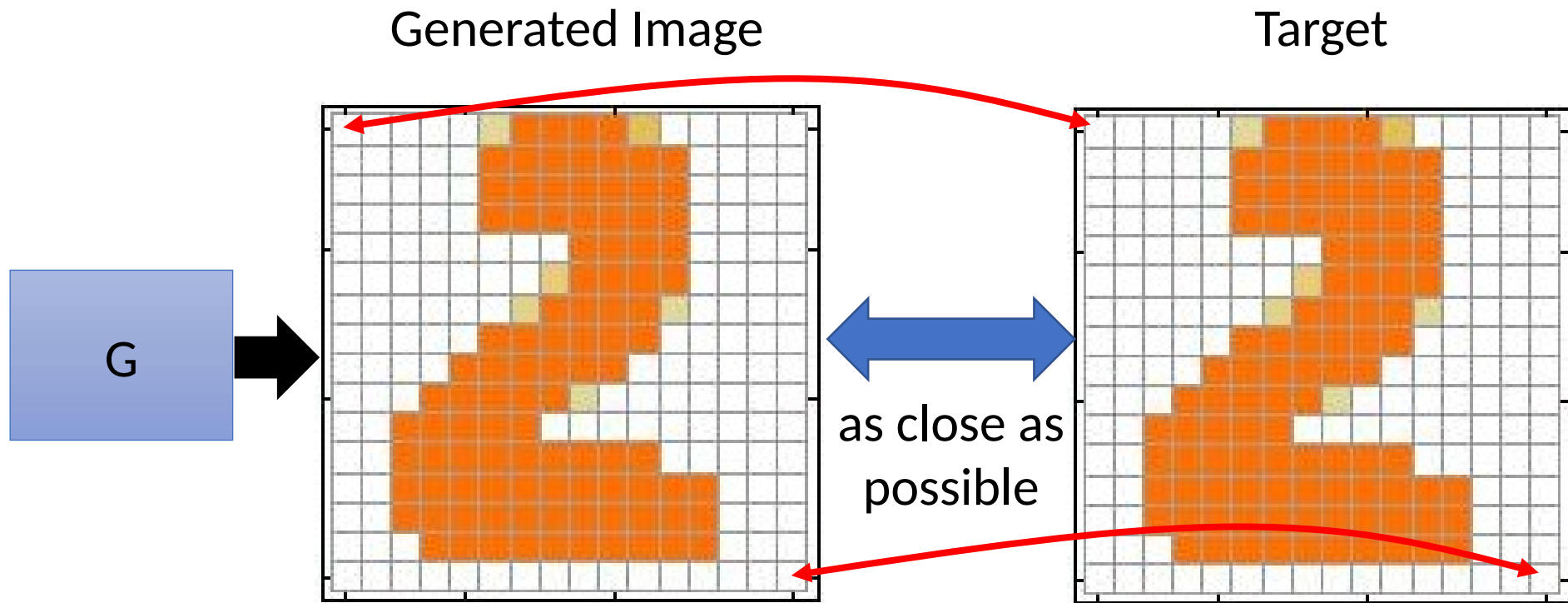
# Auto-encoder



# Variational Auto-encoder (VAE)



# What do we miss?

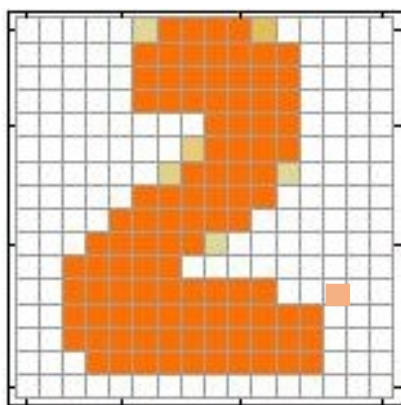
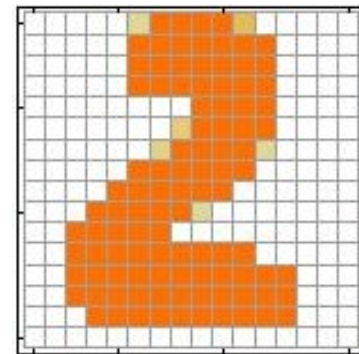


It will be fine if the generator can truly copy the target image.  
What if the generator makes some mistakes .....

Some mistakes are serious, while some are fine.

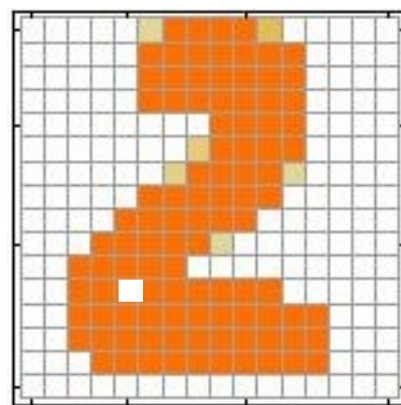
# What do we miss?

Target



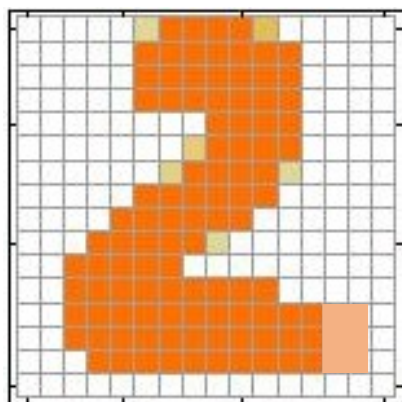
1 pixel error

我覺得不行



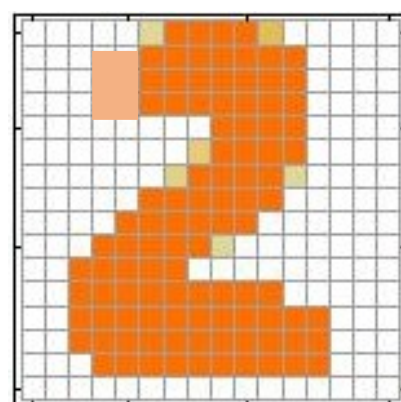
1 pixel error

我覺得不行



6 pixel errors

我覺得其實  
可以

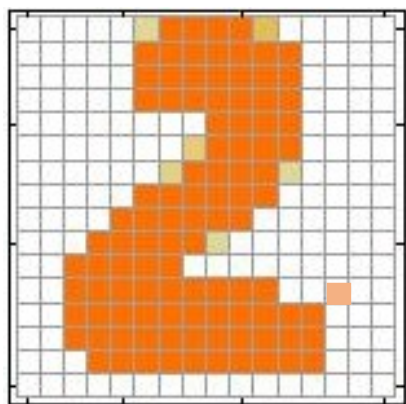


6 pixel errors

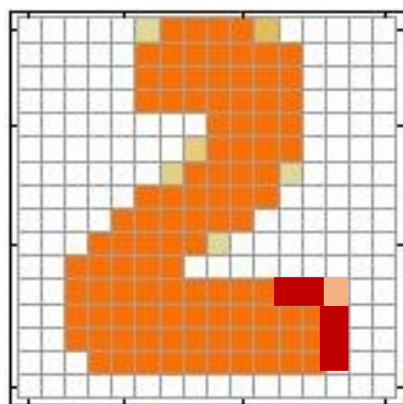
我覺得其實  
可以

# What do we miss?

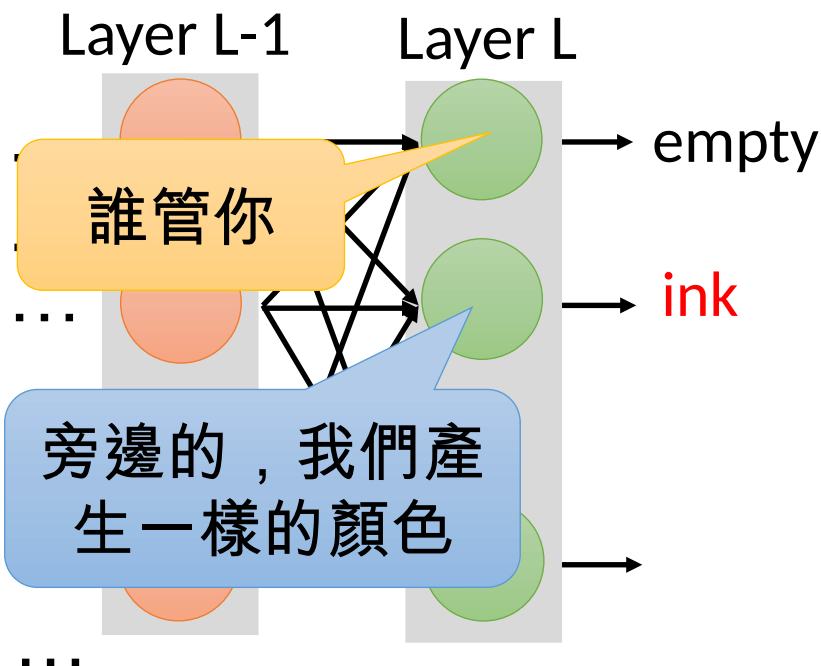
Each neural in output layer corresponds to a pixel.



我覺得不行



我覺得其實可以

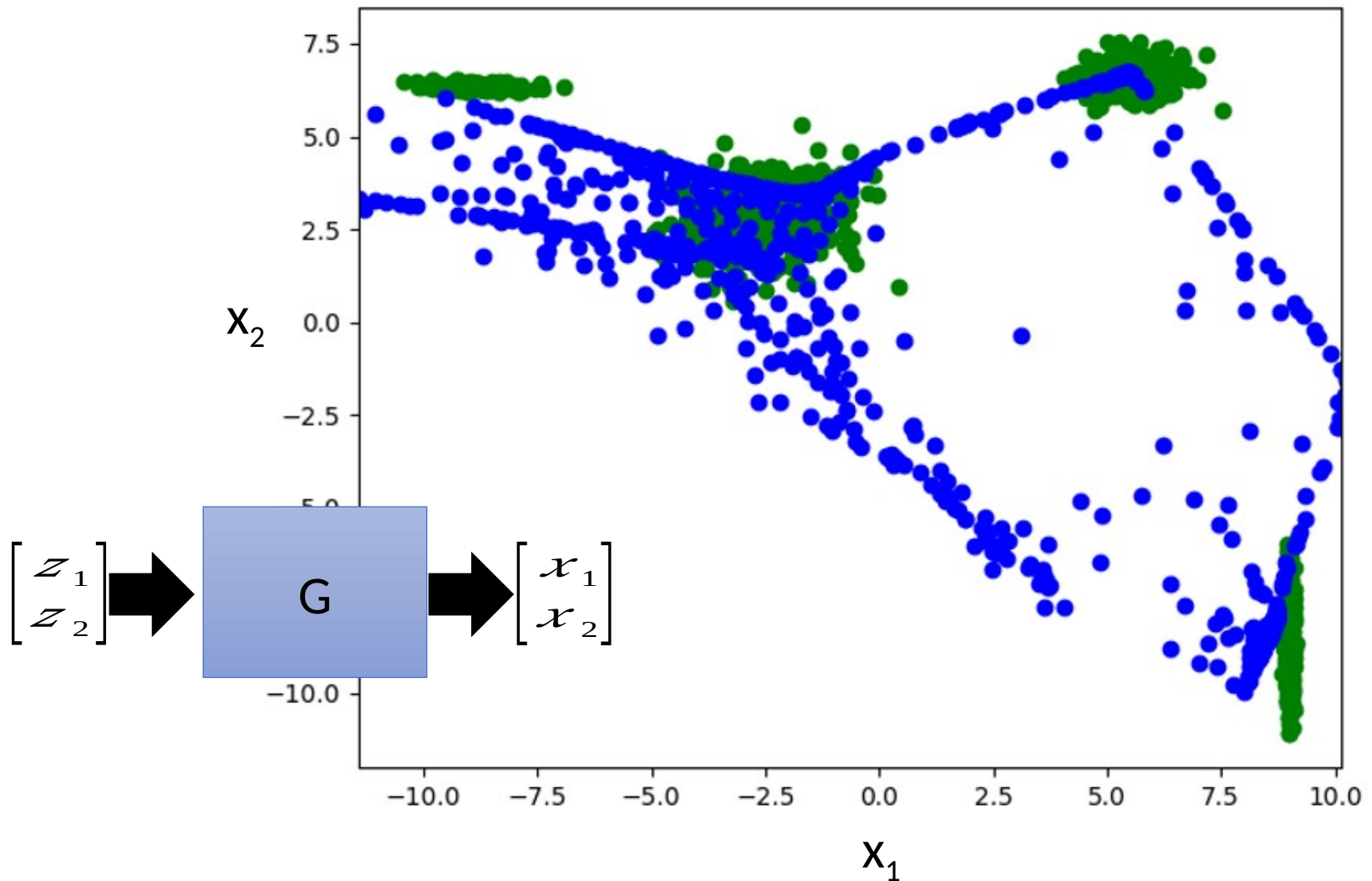


The relation between the components are critical.

Although highly correlated, they cannot influence each other.

Need deep structure to catch the relation between components.

# (Variational) Auto-encoder



# Outline

Basic Idea of GAN

GAN as structured learning

Can Generator learn by itself?

Can Discriminator generate?

A little bit theory

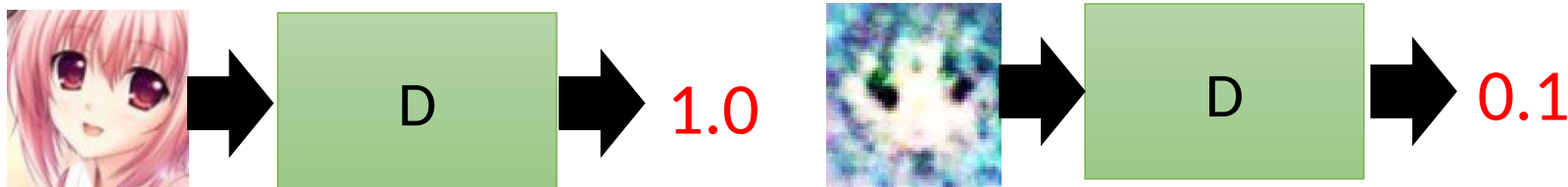
# Discriminator

Evaluation function, Potential Function, Energy Function ...

- Discriminator is a function  $D$  (network, can deep)

$$D : X \rightarrow \mathbb{R}$$

- Input  $x$ : an object  $x$  (e.g. an image)
- Output  $D(x)$ : scalar which represents how “good” an object  $x$  is



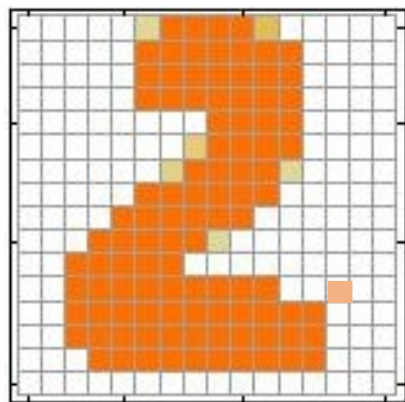
Can we use the discriminator to generate objects?

Yes.

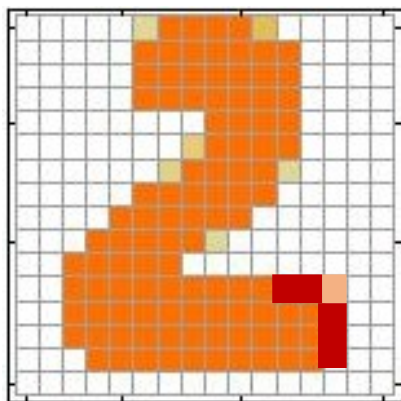


# Discriminator

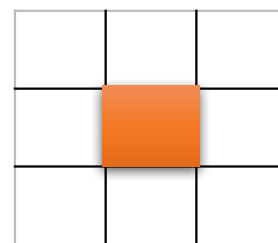
- It is easier to catch the relation between the components by top-down evaluation.



我覺得不行



我覺得其實 OK



This CNN filter is good enough.

# Discriminator

- Suppose we already have a good discriminator  $D(x)$   
...

## Inference

- Generate object  $\tilde{x}$  that

$$\tilde{x} = \arg \max_{x \in X} D(x)$$

Enumerate all possible

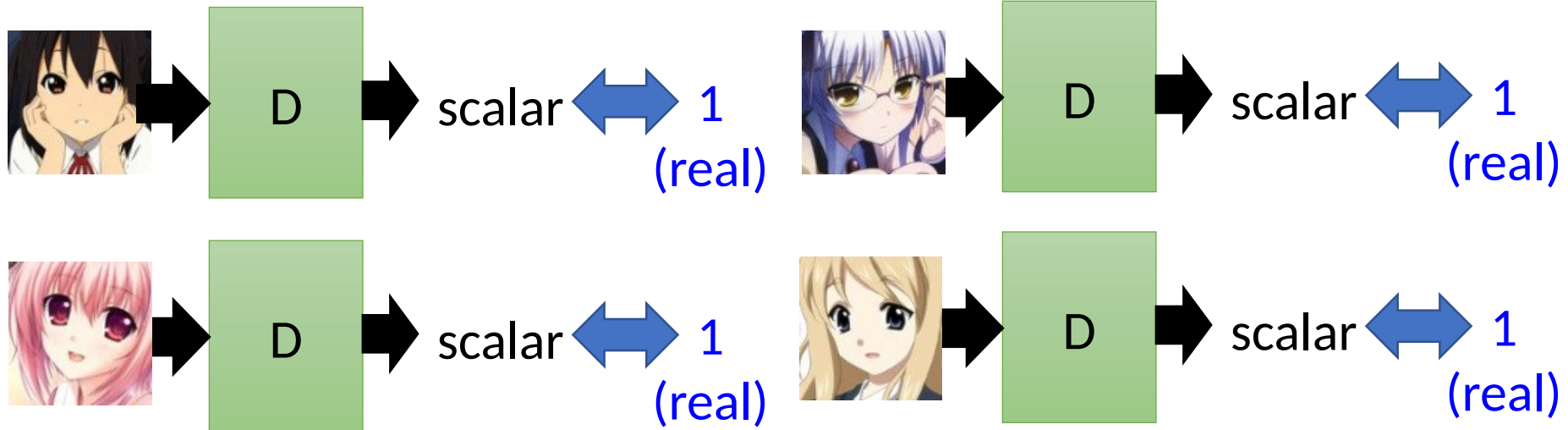
$x$  !!!

It is feasible ???

How to learn the discriminator?

# Discriminator - Training

- I have some real images

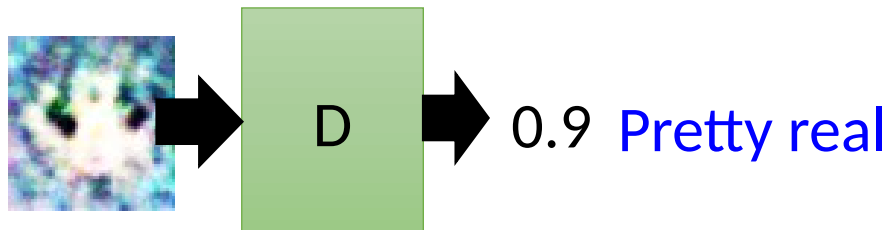
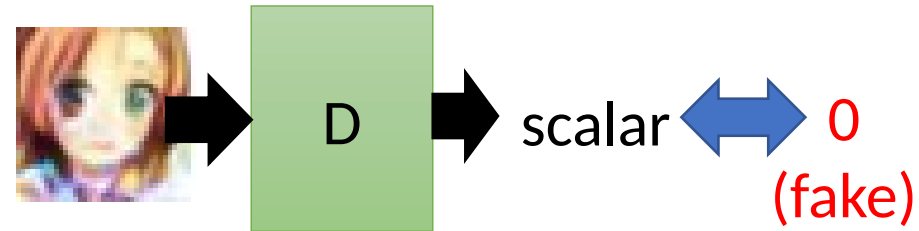
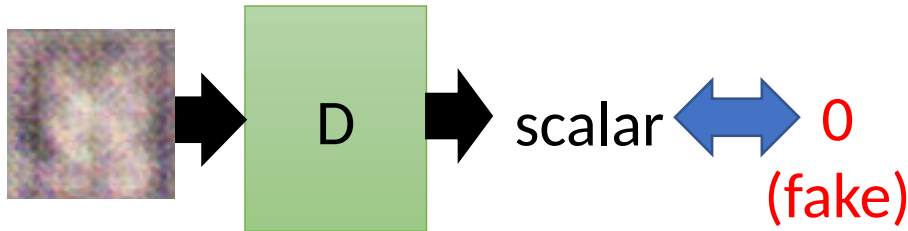
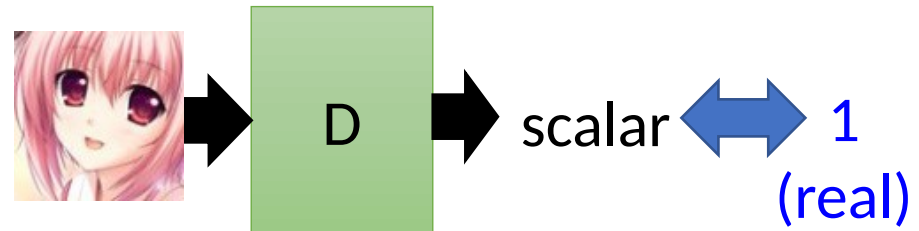
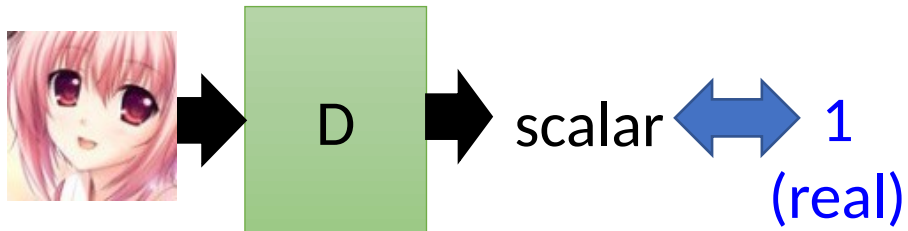


Discriminator only learns to output “1” (real).

Discriminator training needs some negative examples.

# Discriminator - Training

- Negative examples are critical.



How to generate realistic negative examples?

# Discriminator - Training

- General Algorithm

- Given a set of **positive examples**, randomly generate a set of **negative examples**.



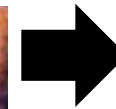
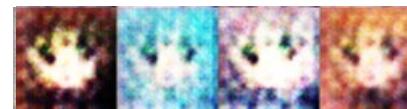
- **In each iteration**



- Learn a discriminator D that can discriminate positive and negative examples.



v.s.



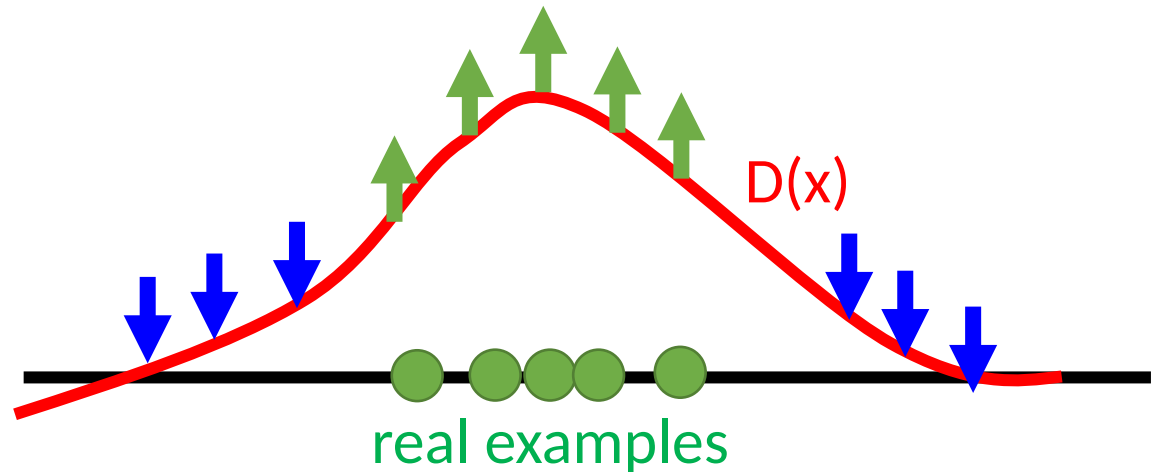
D

- Generate negative examples by discriminator D



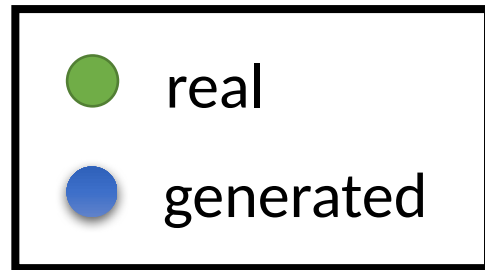
$$\tilde{x} = \arg \max_{x \in X} D(x)$$

# Discriminator - Training

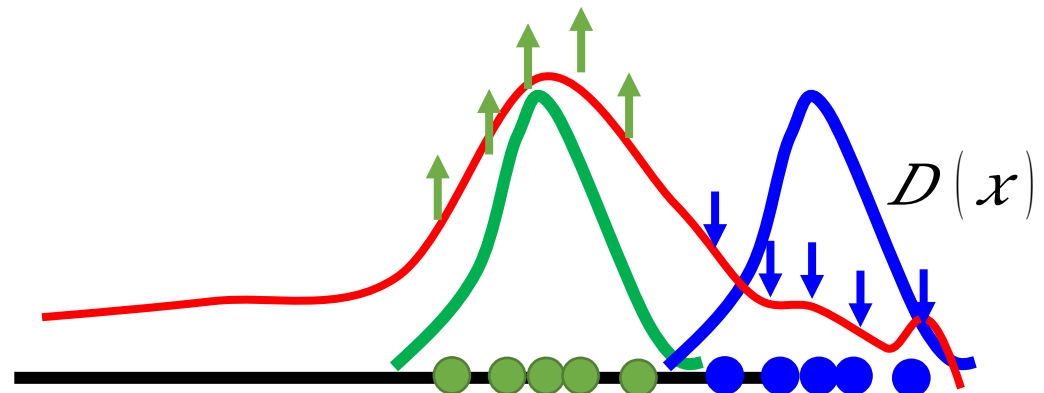
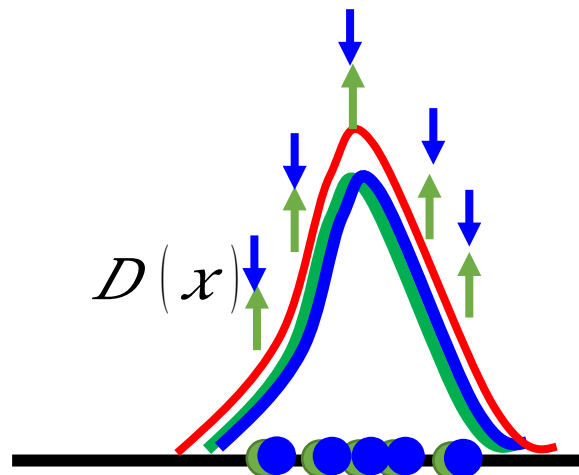
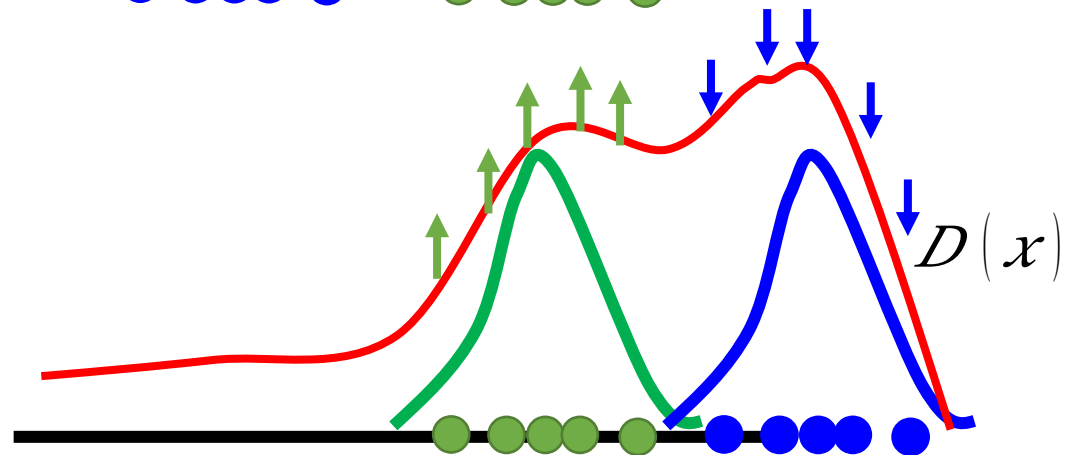
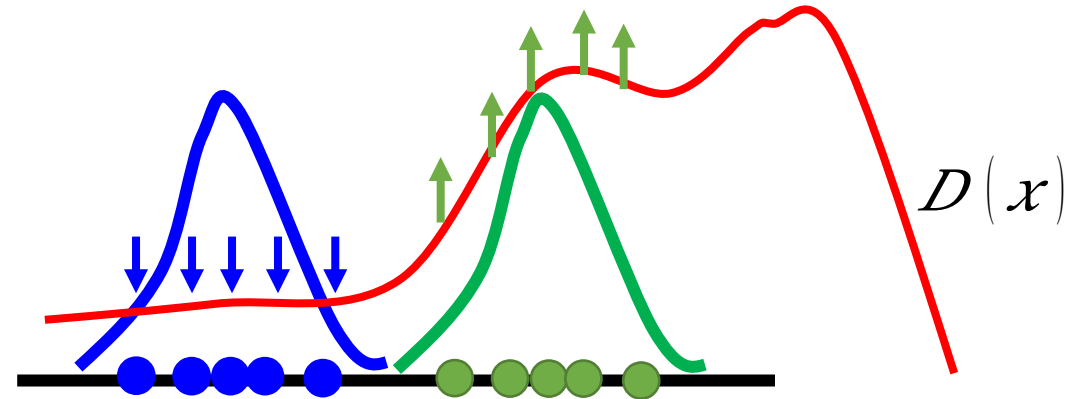


In practice, you cannot decrease all the  $x$  other than real examples.

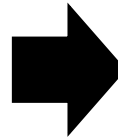
# Discriminator - Training



In the end .....

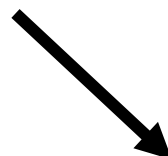
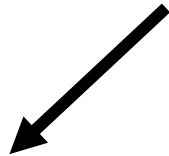


Structured Learning



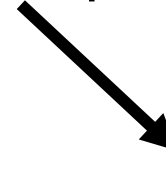
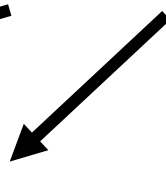
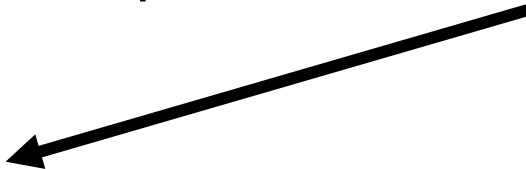
- Structured Perceptron
- Structured SVM
- Gibbs sampling
- Hidden information
- Application: sequence labelling, summarization

Graphical Model



Bayesian Network  
(Directed Graph)

Markov Random Field  
(Undirected Graph)



Conditional  
Random Field

Markov Logic  
Network

Boltzmann  
Machine



Segmental CRF

(Only list some of  
the approaches)



Restricted  
Boltzmann Machine

Energy-based  
Model:  
<http://www.cs.nyu.edu/~yann/research/ebm/>



# Generator v.s. Discriminator

- Generator

- Pros:

- Easy to generate even with deep model

- Cons:

- Imitate the appearance
- Hard to learn the correlation between components

- Discriminator

- Pros:

- Considering the big picture

- Cons:

- Generation is not always feasible
  - Especially when your model is deep
- How to do negative sampling?

# Generator + Discriminator

- General Algorithm

- Given a set of **positive examples**, randomly generate a set of **negative examples**.



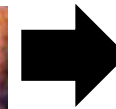
- In each iteration



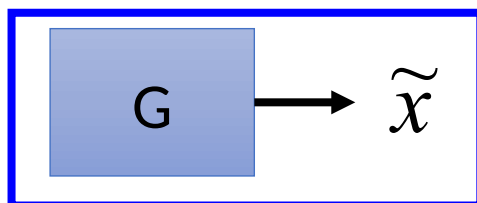
- Learn a discriminator D that can discriminate positive and negative examples.



v.s.



- Generate negative examples by discriminator D



=

$$\tilde{x} = \arg \max_{x \in X} D(x)$$

# Benefit of GAN

- From Discriminator's point of view
  - Using generator to generate negative samples

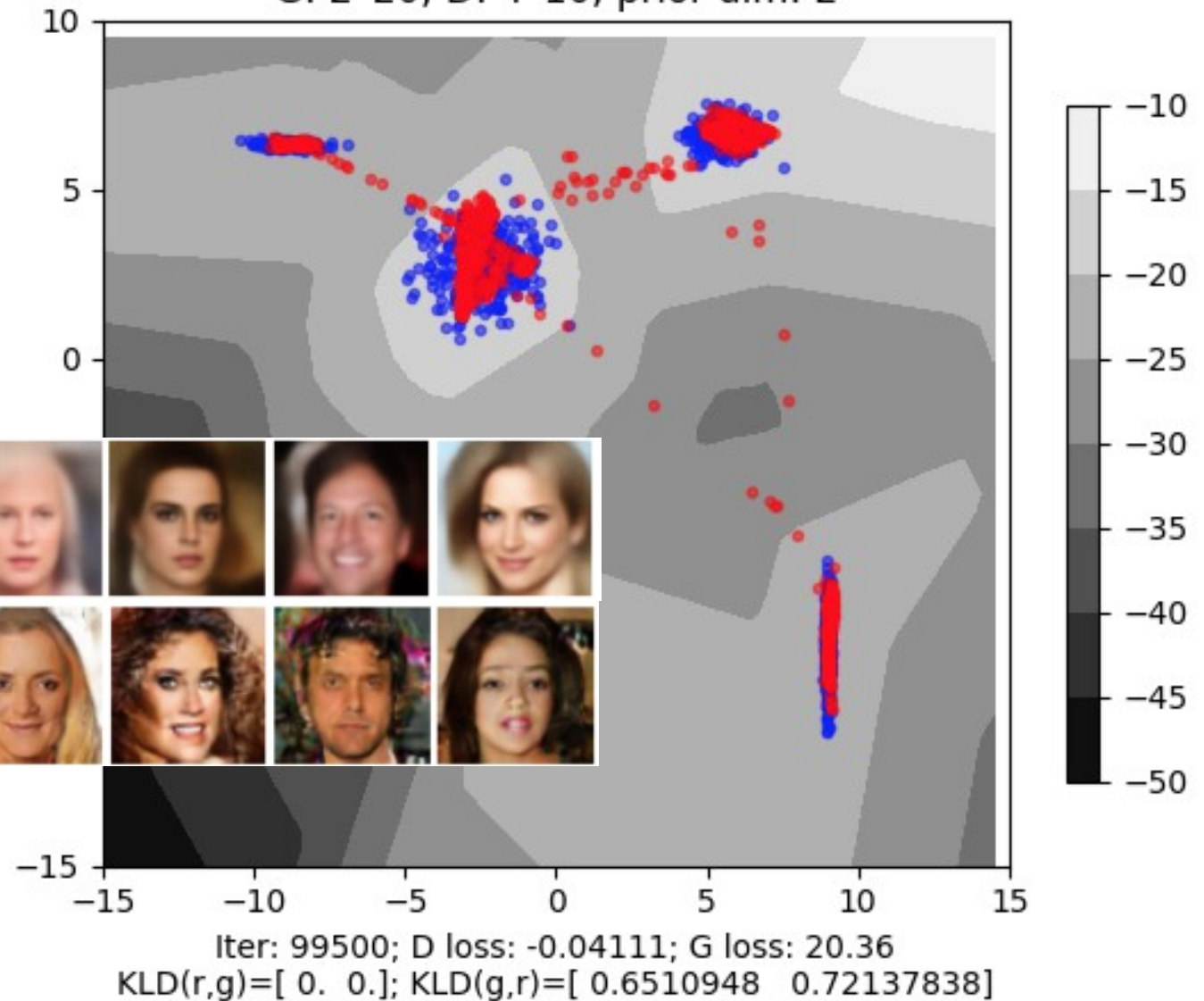
$$\boxed{\begin{array}{c} \text{G} \end{array} \rightarrow \tilde{x}} = \boxed{\tilde{x} = \arg \max_{x \in X} D(x)}$$

efficient

- From Generator's point of view
  - Still generate the object component-by-component
  - But it is learned from the discriminator with global view.

# GAN

wgan-gp-sub1000-gauss4  
Samples and Decision Boundary  
G: 2\*20; D: 4\*10; prior dim: 2

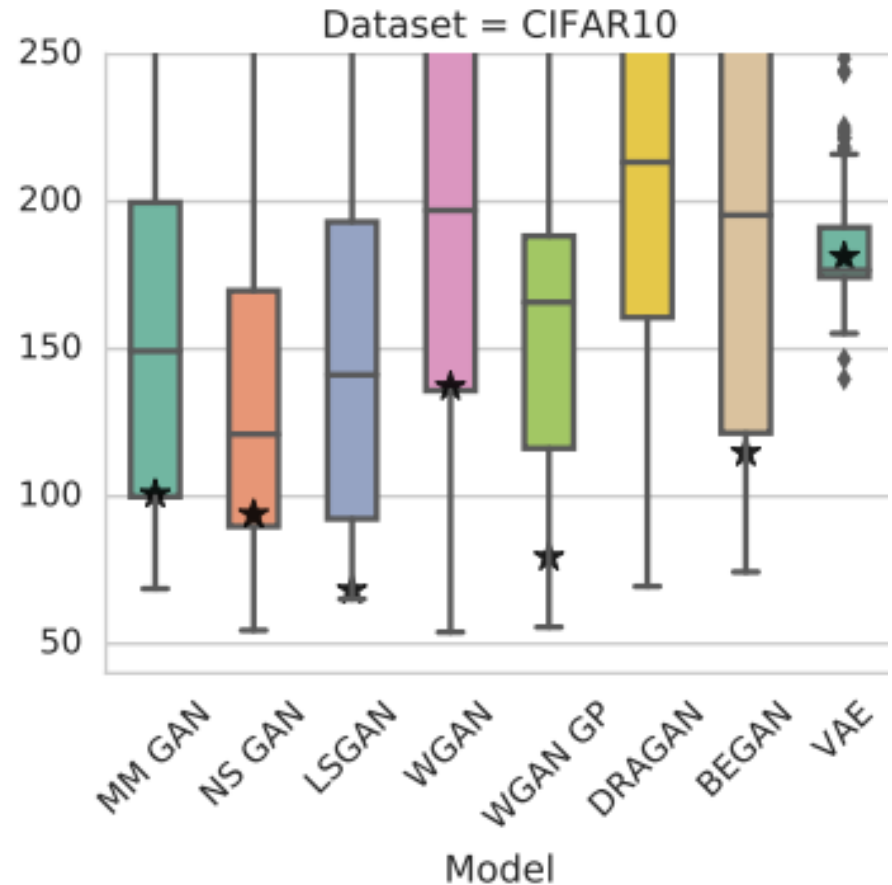
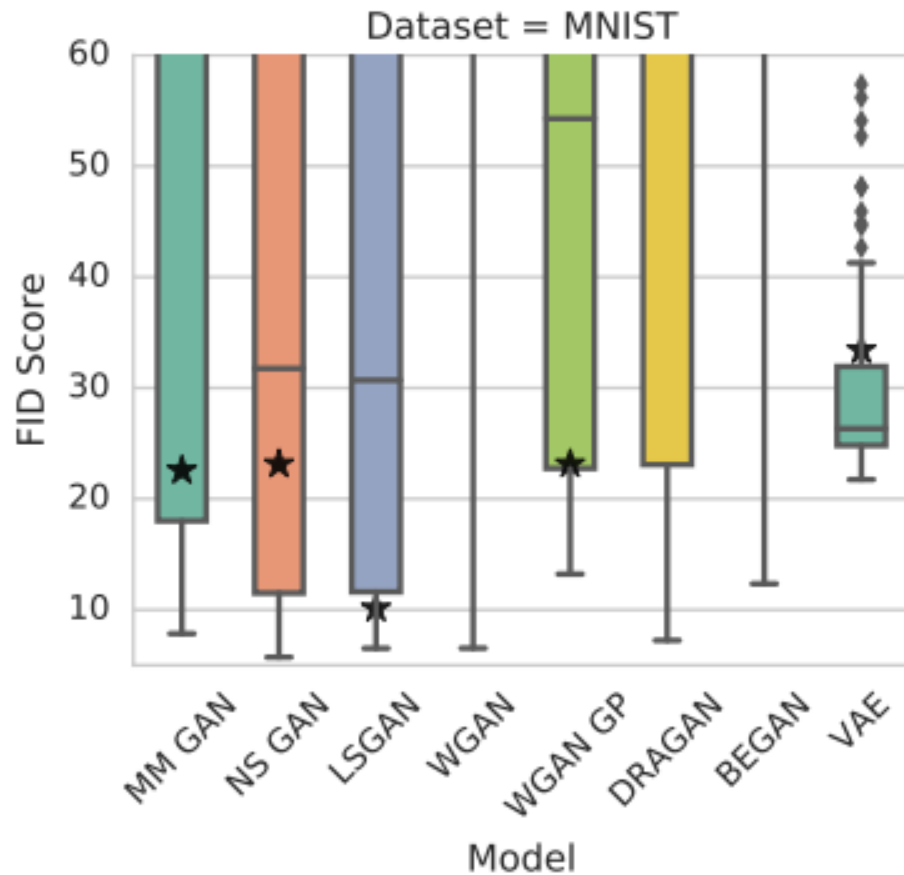


VAE

GAN

<https://arxiv.org/abs/1512.09300>

[Mario Lucic, et al. arXiv, 2017]



FID[Martin Heusel, et al., NIPS, 2017]: Smaller is better

# Next Time

- Preview
  - <https://youtu.be/0CKeqXI5IY0>
  - <https://youtu.be/KSN4QYgAtao>

# Outline

Basic Idea of GAN

GAN as structured learning

Can Generator learn by itself?

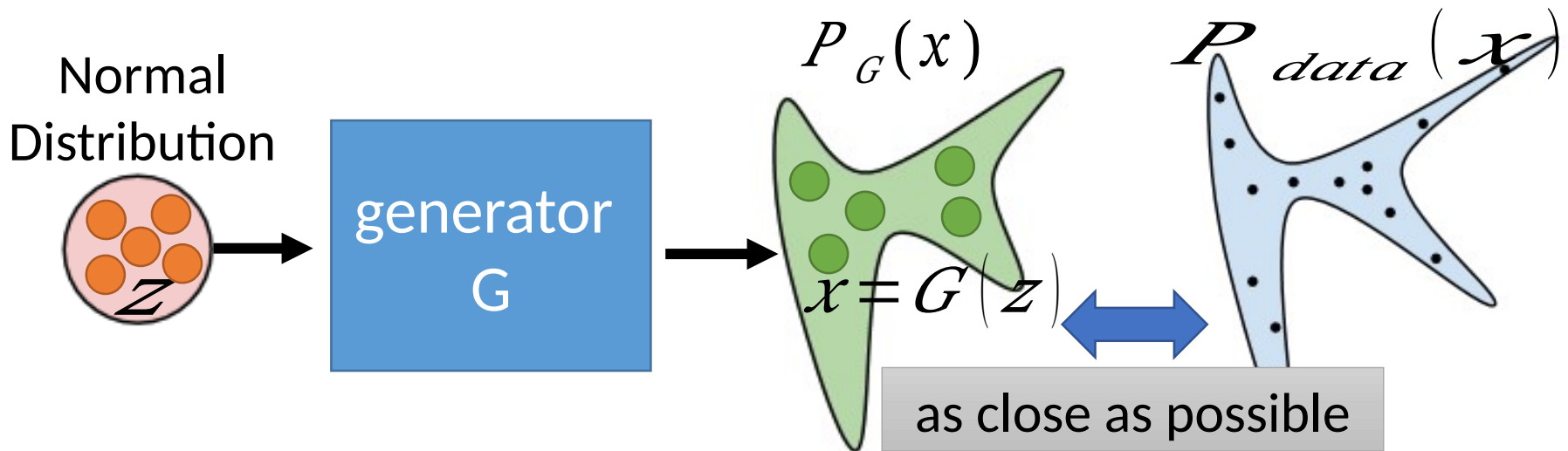
Can Discriminator generate?

A little bit theory

# Generator

: an image (a high-dimensional vector)

- A generator  $G$  is a network. The network defines a probability distribution



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

Divergence between distributions and

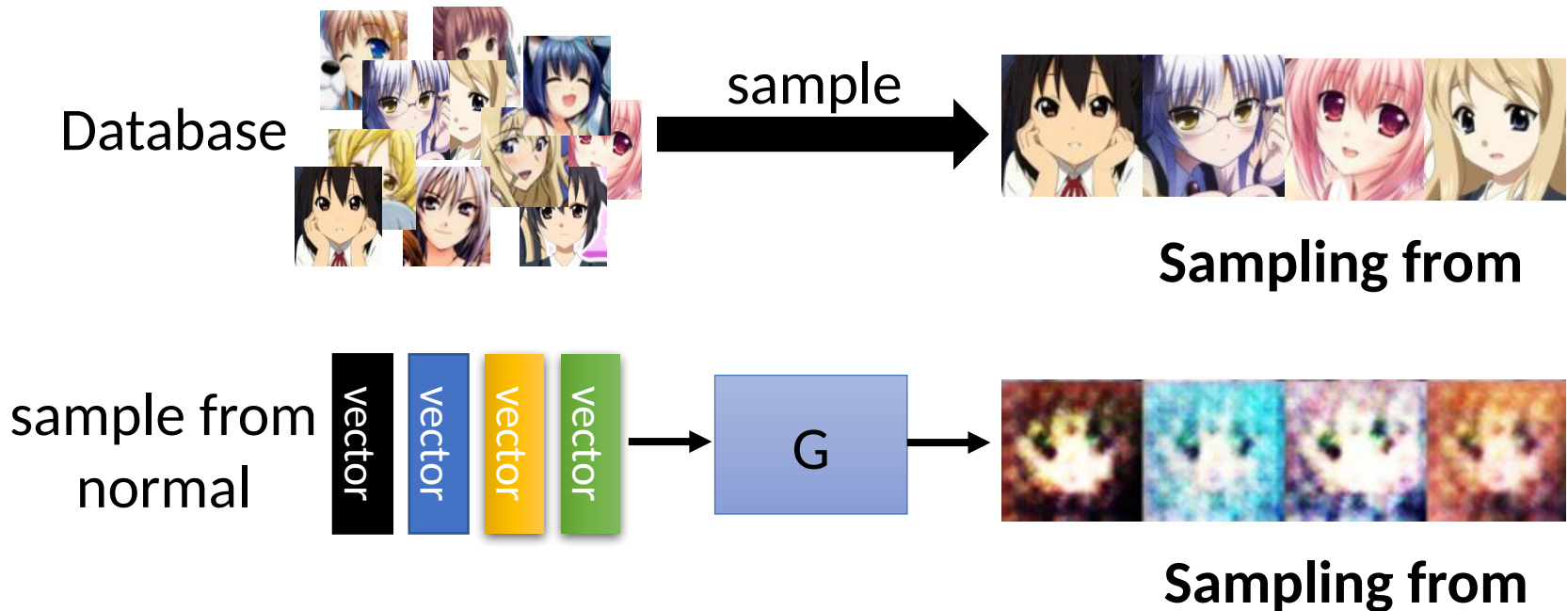
How to compute the divergence?



# Discriminator

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

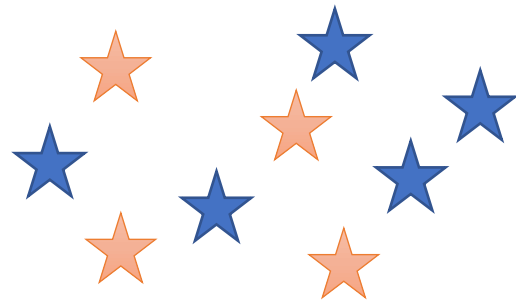
Although we do not know the distributions of  $P_G$  and  $P_{data}$ , we can sample from them.



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

★ : data sampled from

★ : data sampled from



train

Discriminator

Using the example objective function is exactly the same as training a binary classifier.

## Example Objective Function for D

$$V(G, D) = E_{x \sim P_{data}} [\log D(x)] + E_{x \sim P_G} [\log (1 - D(x))]$$



(G is fixed)

Training:  $D^* = \arg \max_D V(D, G)$

The maximum objective value is related to JS divergence.

# Discriminator

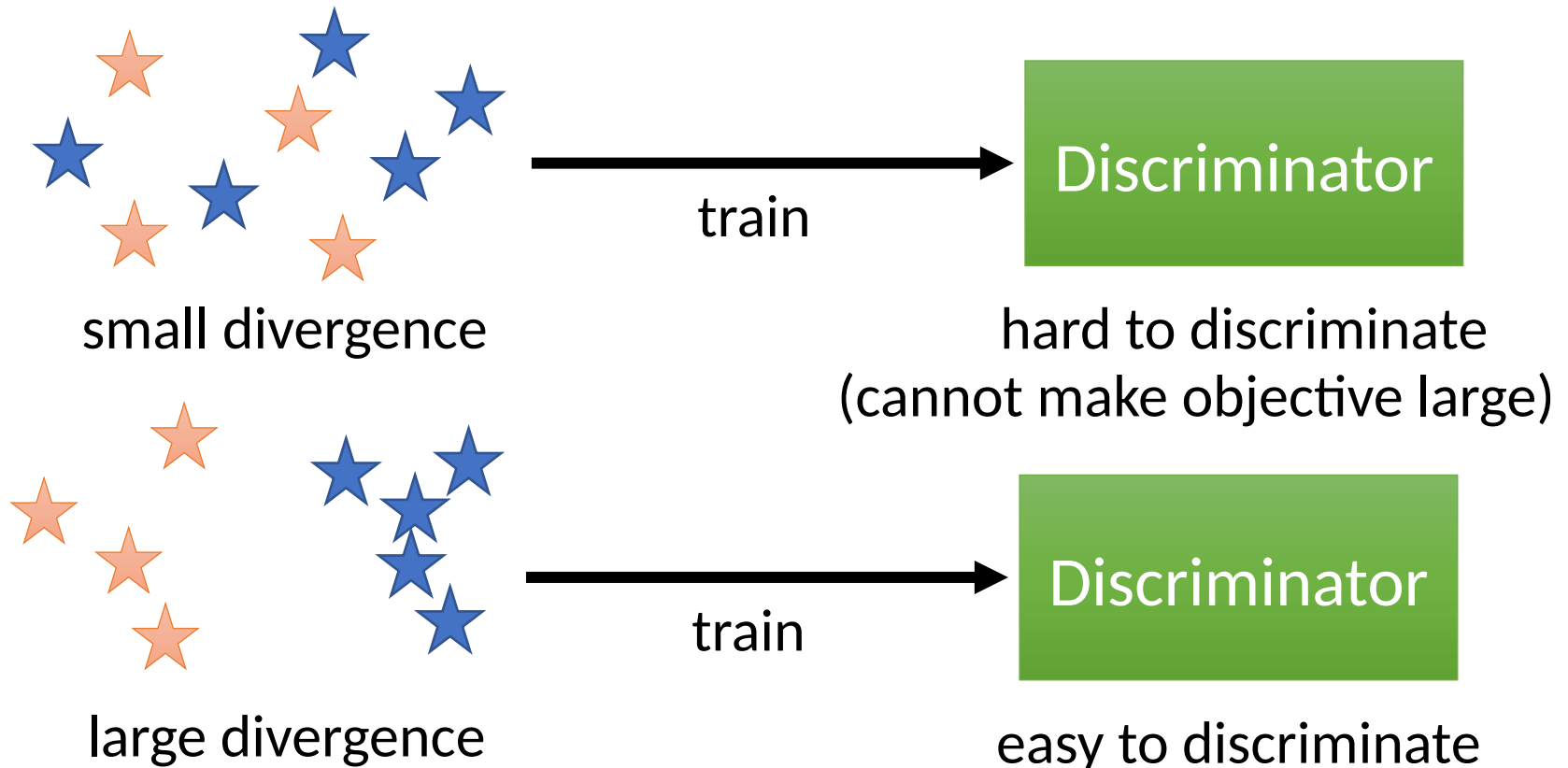
$$G^* = \arg \min_G \text{Div}(P_G, P_{data})$$

★ : data sampled from

★ : data sampled from

**Training:**

$$D^* = \arg \max_D V(D, G)$$



$$G^* = \arg \min_G \max_D V(G, D)$$

$$D^* = \arg \max_D V(D, G)$$

The maximum objective value is related to JS divergence.

- Initialize generator and discriminator
- In each training iteration:

**Step 1**: Fix generator  $G$ , and update discriminator  $D$

**Step 2**: Fix discriminator  $D$ , and update generator  $G$

# Can we use other divergence?

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

Using the divergence  
you like 😊

Name	Conjugate $f^*(t)$
Total variation	$t$
Kullback-Leibler (KL)	$\exp(t - 1)$
Reverse KL	$-1 - \log(-t)$
Pearson $\chi^2$	$\frac{1}{4}t^2 + t$
Neyman $\chi^2$	$2 - 2\sqrt{1 - t}$
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	$-\log(1 - \exp(t))$