Introduction of Generative Adversarial Network (GAN)

李宏毅

Hung-yi Lee

Generative Adversarial Network (GAN)

- How to pronounce “GAN”?

Google 小姐

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 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.

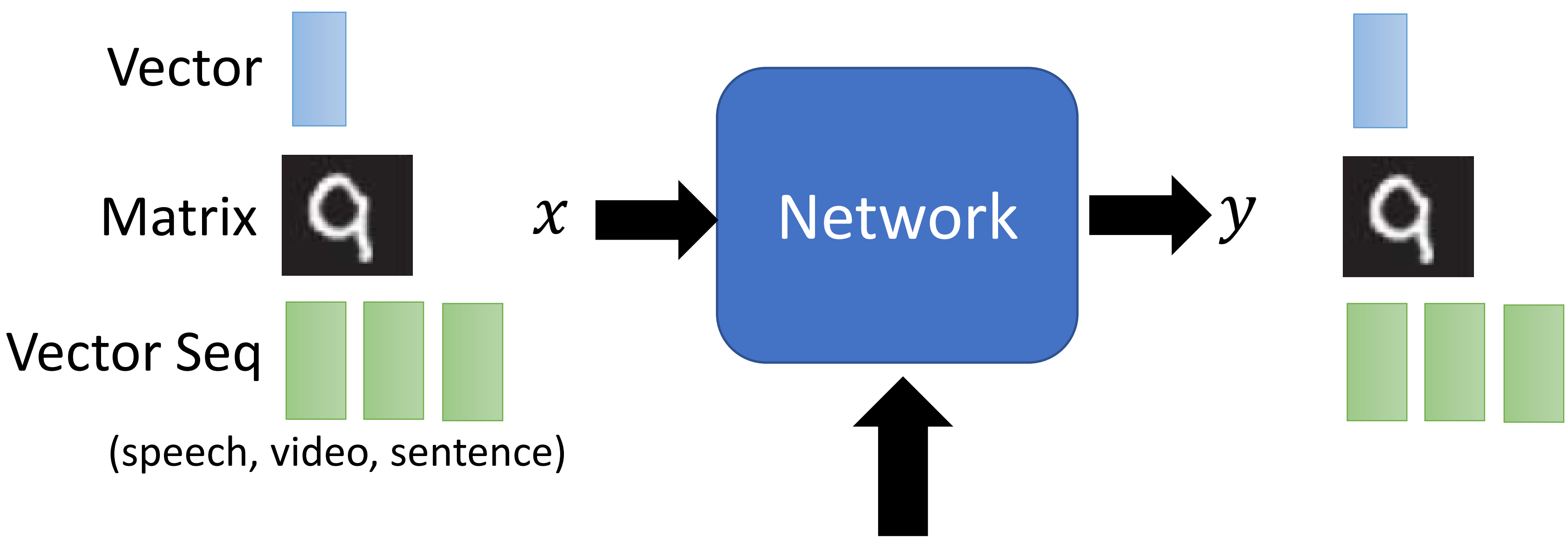
https://www.quora.com/What-are-some-recent-and-potentially-upcoming-breakthroughs-in-deep-learning

Deep Learning in One Slide (Review)

Many kinds of network structures:

- Fully connected feedforward network
- Convolutional neural network (CNN)
- Recurrent neural network (RNN)

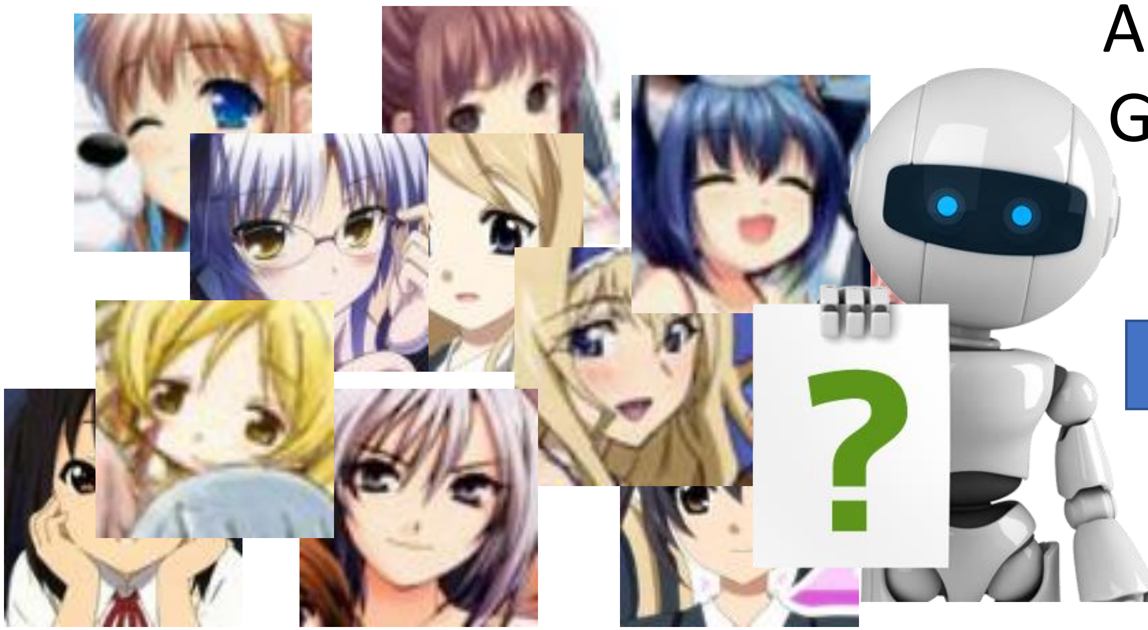
Different networks can take different kinds of input/output.



How to find the function?

Given the example inputs/outputs as training data: $\{(x_1,y_1),(x_2,y_2), \ldots\ldots, (x_{1000},y_{1000})\}$

Creation



Anime Face Generation

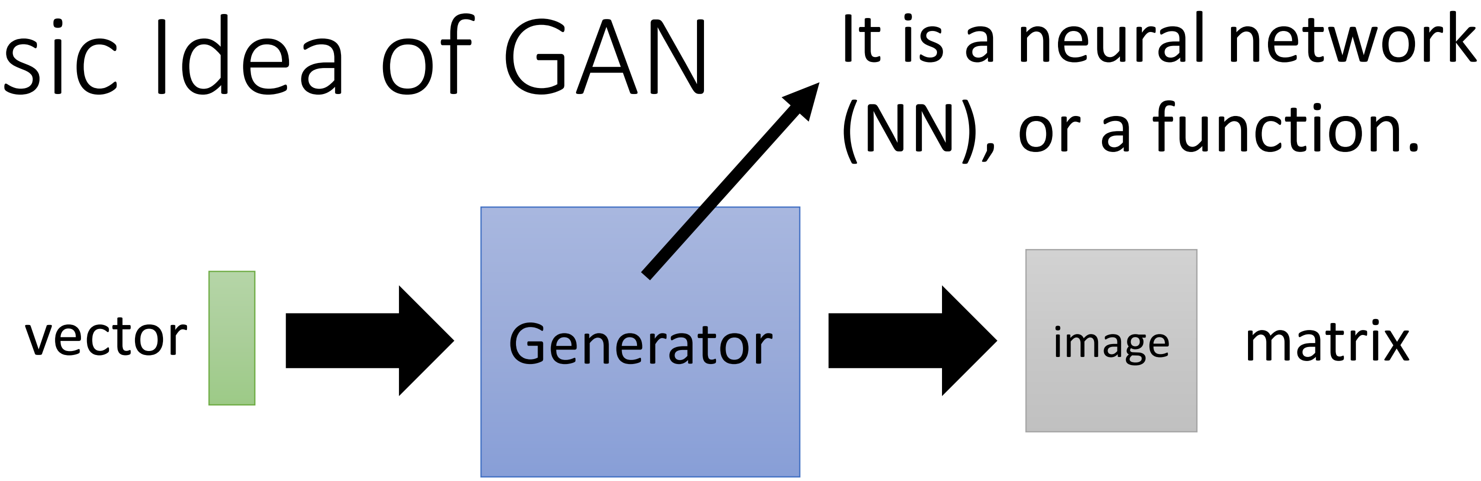
Drawing?

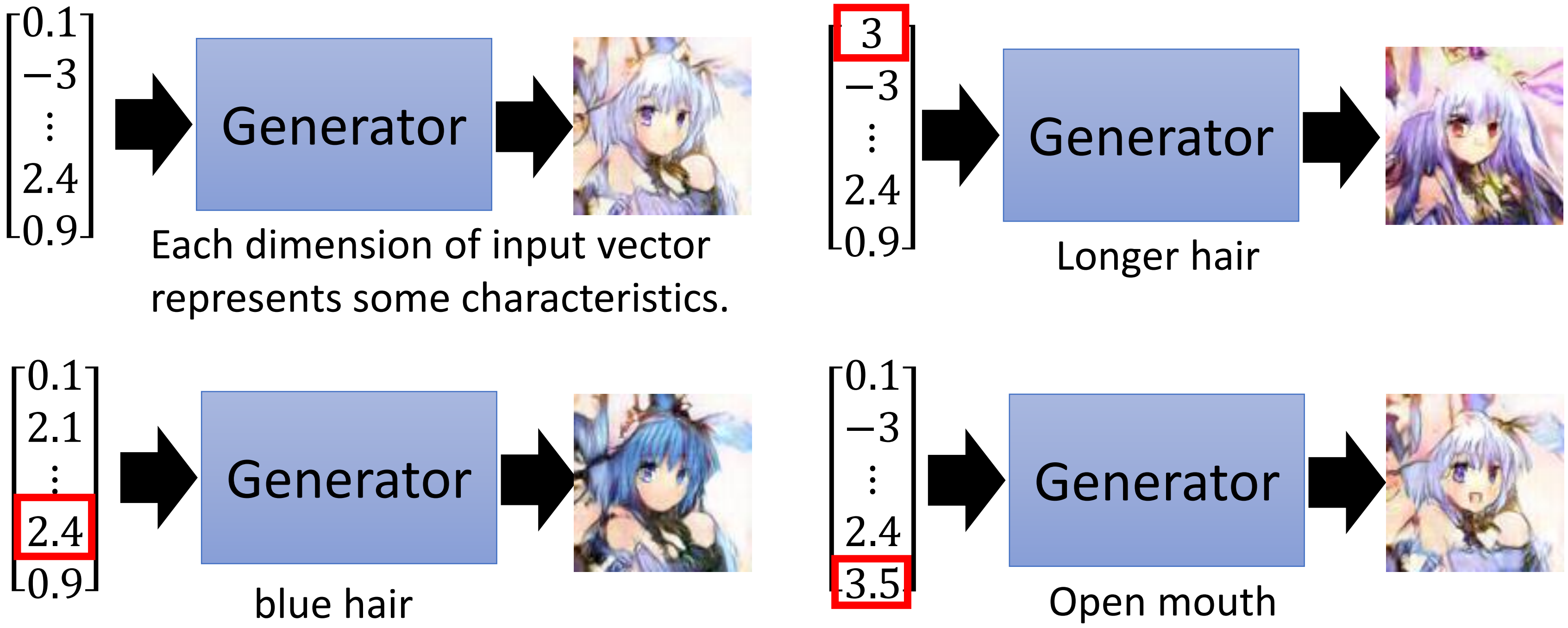
何之源的知乎: https://zhuanlan.zhihu.com/p/24767059

DCGAN: https://github.com/carpedm20/DCGAN-tensorflow

Powered by: http://mattya.github.io/chainer-DCGAN/

Basic Idea of GAN





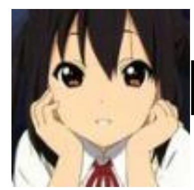
Basic Idea of GAN

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

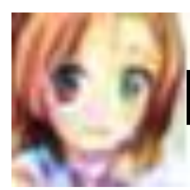
image → Discri-minator → scalar

Larger value means real, smaller value means fake.

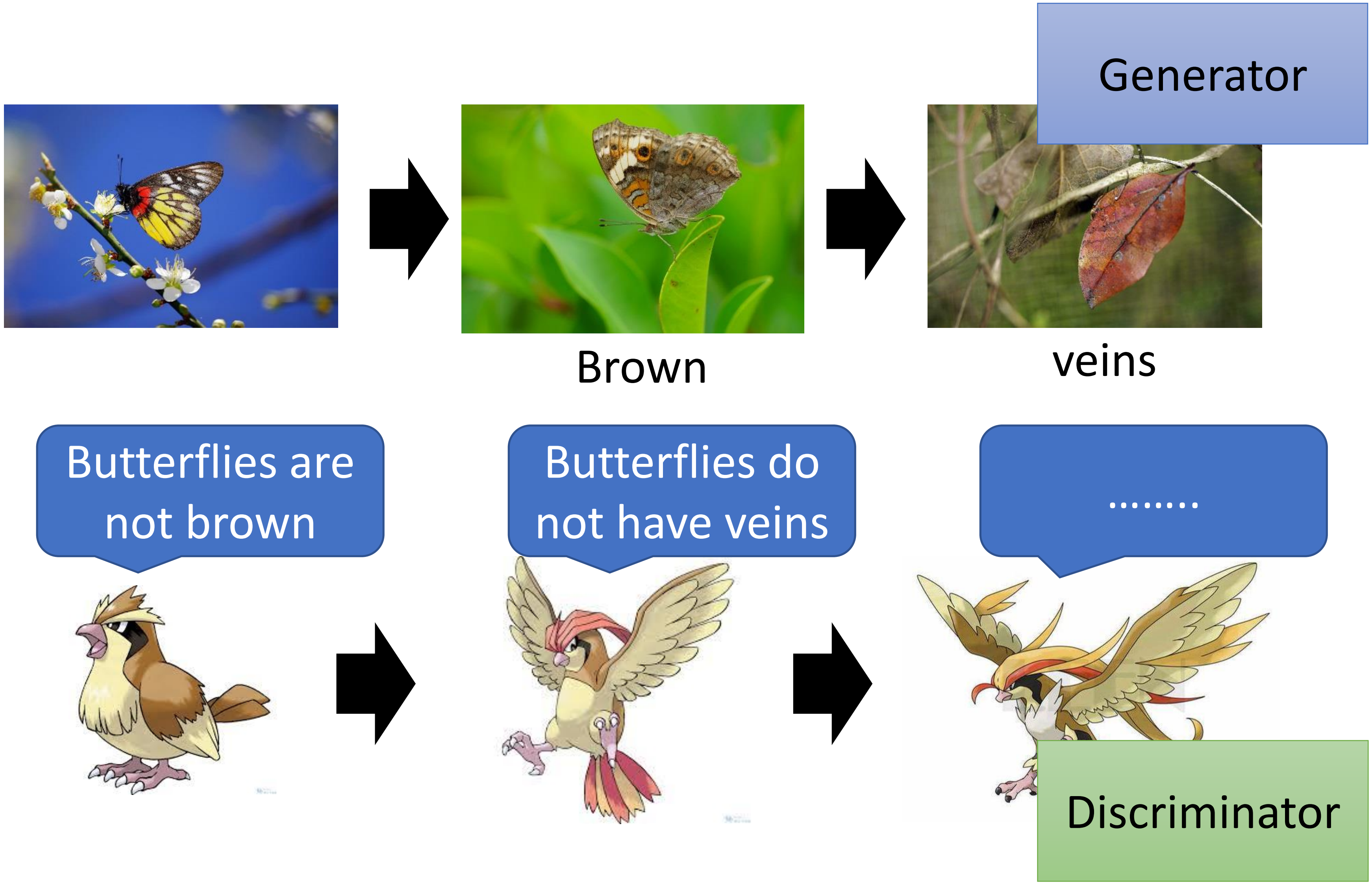
 → Discri-minator → 1.0

 → Discri-minator → 1.0

 → Discri-minator → 0.1

 → Discri-minator → ?

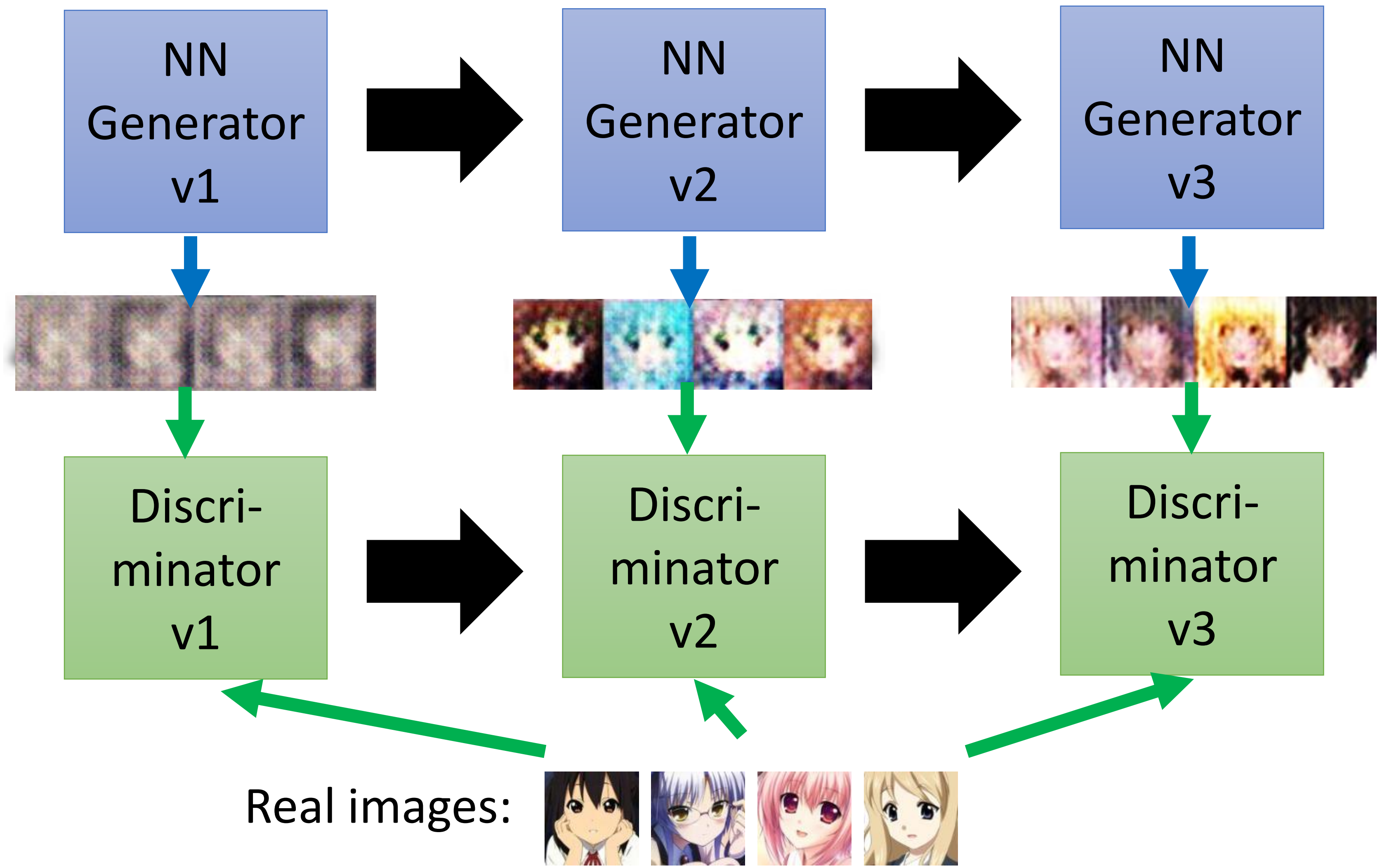
Basic Idea of GAN



Basic Idea of GAN

This is where the term "***adversarial***" comes from.

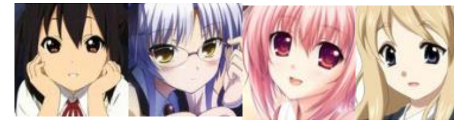
You can explain the process in different ways…….



Basic Idea of GAN

Generator (student)

Discriminator (teacher)



Generator v1

Discriminator v1

No eyes

Generator v2

Discriminator v2

No mouth

Generator v3

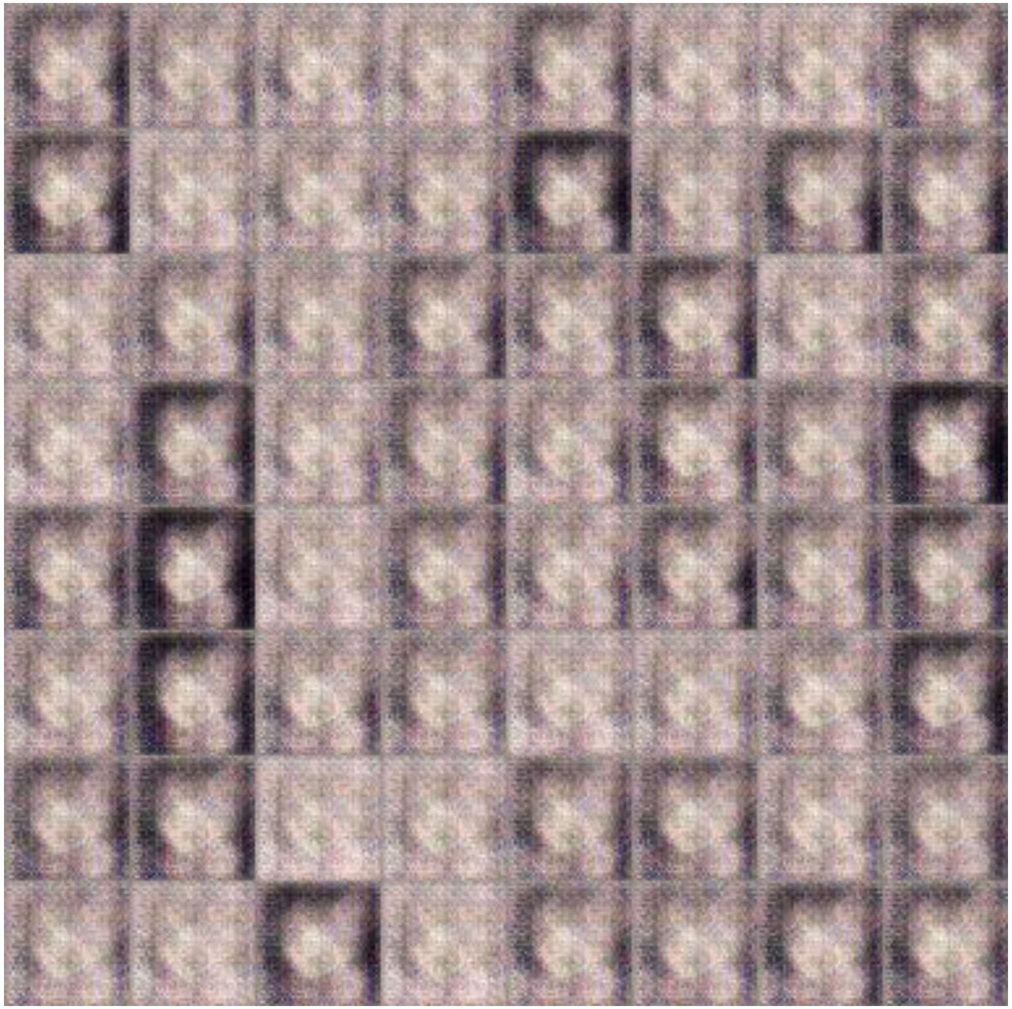
Questions

Q1: Why generator cannot learn by itself?

Q2: Why discriminator don’t generate object itself?

Q3: How discriminator and generator interact?

Anime Face Generation



100 updates

Anime Face Generation

1000 updates

Anime Face Generation

2000 updates

Anime Face Generation

5000 updates

Anime Face Generation

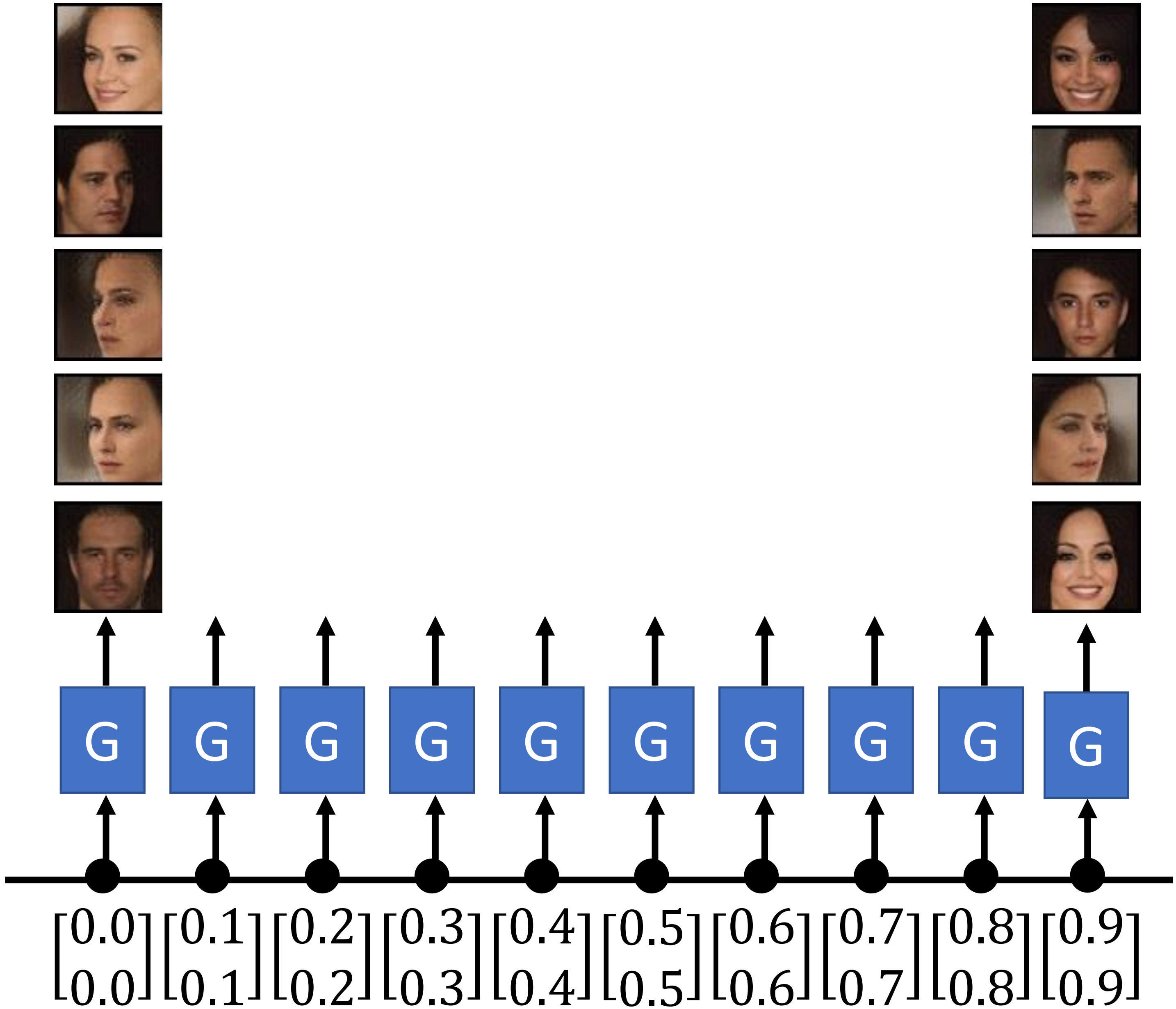
10,000 updates

Anime Face Generation

20,000 updates

Anime Face Generation

50,000 updates



感謝陳柏文同學提供實驗結果

Outline

Lecture 1: Introduction of GAN

Lecture 2: Variants of GAN

Lecture 3: Making Decision and Control

Lecture I:
Introduction of GAN

To learn more theory:
https://www.youtube.com/watch?v=0CKeqXl5IY0&lc=z13zuxbglpvsgbgpo04cg1bxuoraejdpapo0k
https://www.youtube.com/watch?v=KSN4QYgAtao&lc=z13kz1nqvuqsipqfn23phthasre4evrdo

Lecture I

When can I use GAN?

Generation by GAN

Improving GAN

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

Regression,
Classification

Output Sequence $f: X \rightarrow Y$

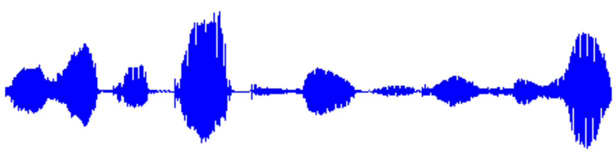
Machine Translation

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)

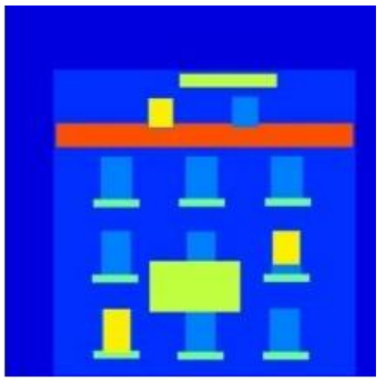
Y : “I’m fine.”
(response of machine)

Output Matrix

$$f : X \rightarrow Y$$

Image to Image

Colorization:

X : 

Y : 

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

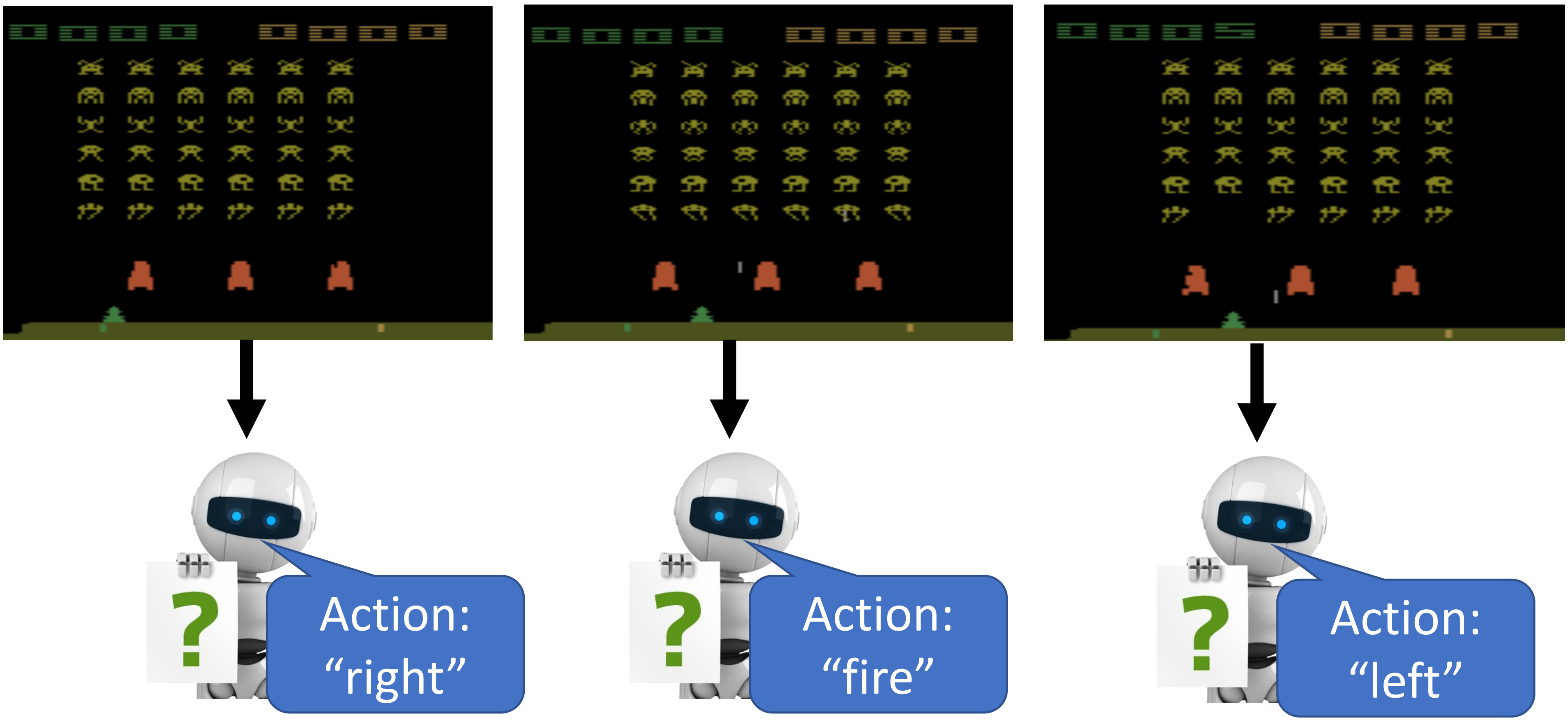
Text to Image

X : "this white and yellow flower have thin white petals and a round yellow stamen"

Y : 

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

Decision Making and Control



GO Playing is the same.

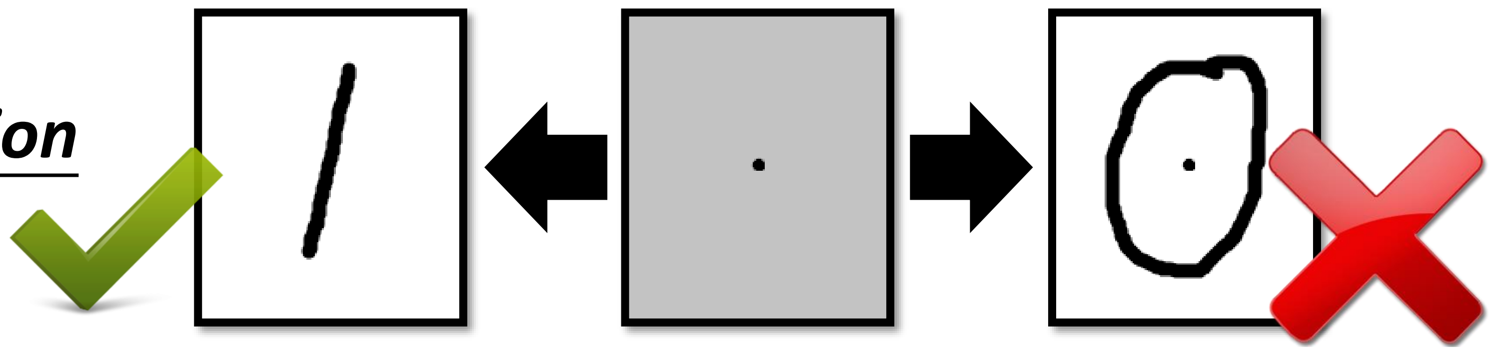
Why Structured Learning Interesting?

- **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 Interesting?

- Machine has to learn to ***planning***
  - Machine can generate 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

Bottom Up

+

Discriminator

Evaluating the whole object, and find the best one

Top Down

Lecture I

When can I use GAN?

Generation by GAN

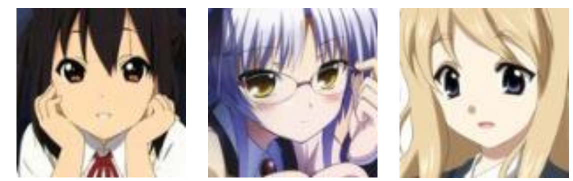
Improving GAN

Generation

We will control what to generate latter. → Conditional Generation

Image Generation

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



In a specific range

Sentence Generation

$$\begin{bmatrix} 0.3 \\ -0.1 \\ \vdots \\ -0.7 \end{bmatrix} \begin{bmatrix} 0.1 \\ -0.1 \\ \vdots \\ 0.2 \end{bmatrix} \begin{bmatrix} -0.3 \\ 0.1 \\ \vdots \\ 0.5 \end{bmatrix} \rightarrow \boxed{\text{NN Generator}} \rightarrow$$

How are you?

Good morning.

Good afternoon.

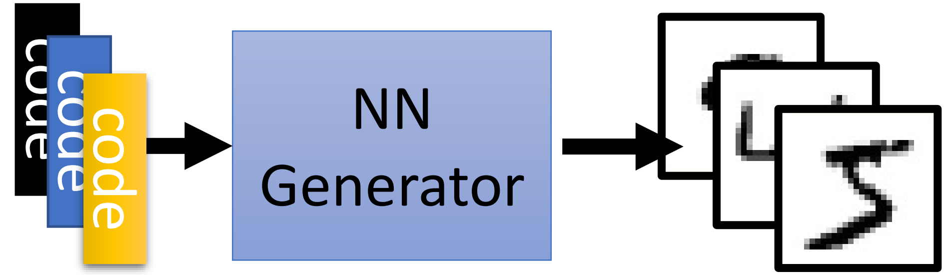
So many questions ……

Q1: Why generator cannot learn by itself?

Q2: Why discriminator don’t generate object itself?

Q3: How discriminator and generator interact?

Generator



vectors

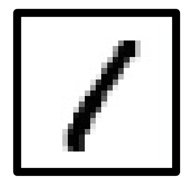
code: $\begin{bmatrix} 0.1 \\ -0.5 \end{bmatrix}$ $\begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}$ $\begin{bmatrix} 0.2 \\ -0.1 \end{bmatrix}$ $\begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}$

(where does them come from?)

Image:

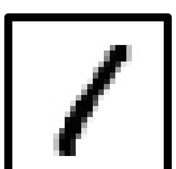
As close as possible

$\begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}$ → NN Generator → image ↔



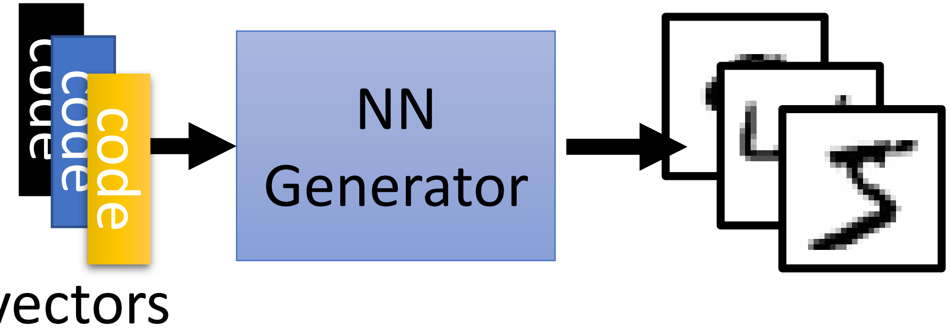
As close as possible

c.f.



→ NN Classifier → $\begin{bmatrix} y_1 \\ y_2 \\ \vdots \end{bmatrix}$ ↔ $\begin{bmatrix} 1 \\ 0 \\ \vdots \end{bmatrix}$

Generator

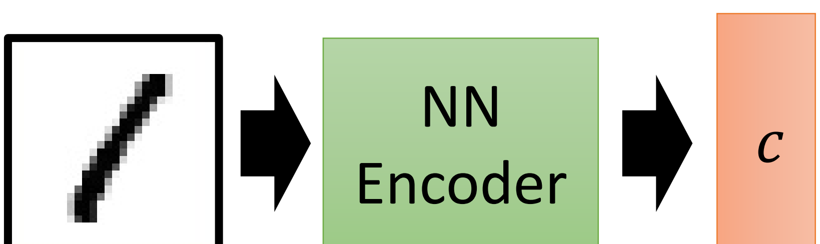


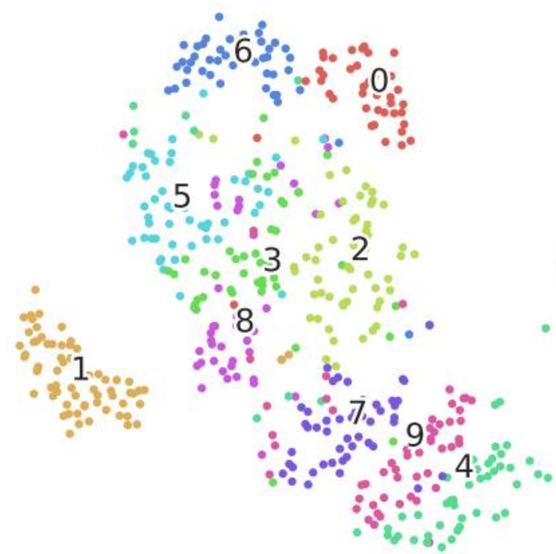
code: $\begin{bmatrix} 0.1 \\ -0.5 \end{bmatrix}$ $\begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}$ $\begin{bmatrix} 0.2 \\ -0.1 \end{bmatrix}$ $\begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}$

(where does them come from?)

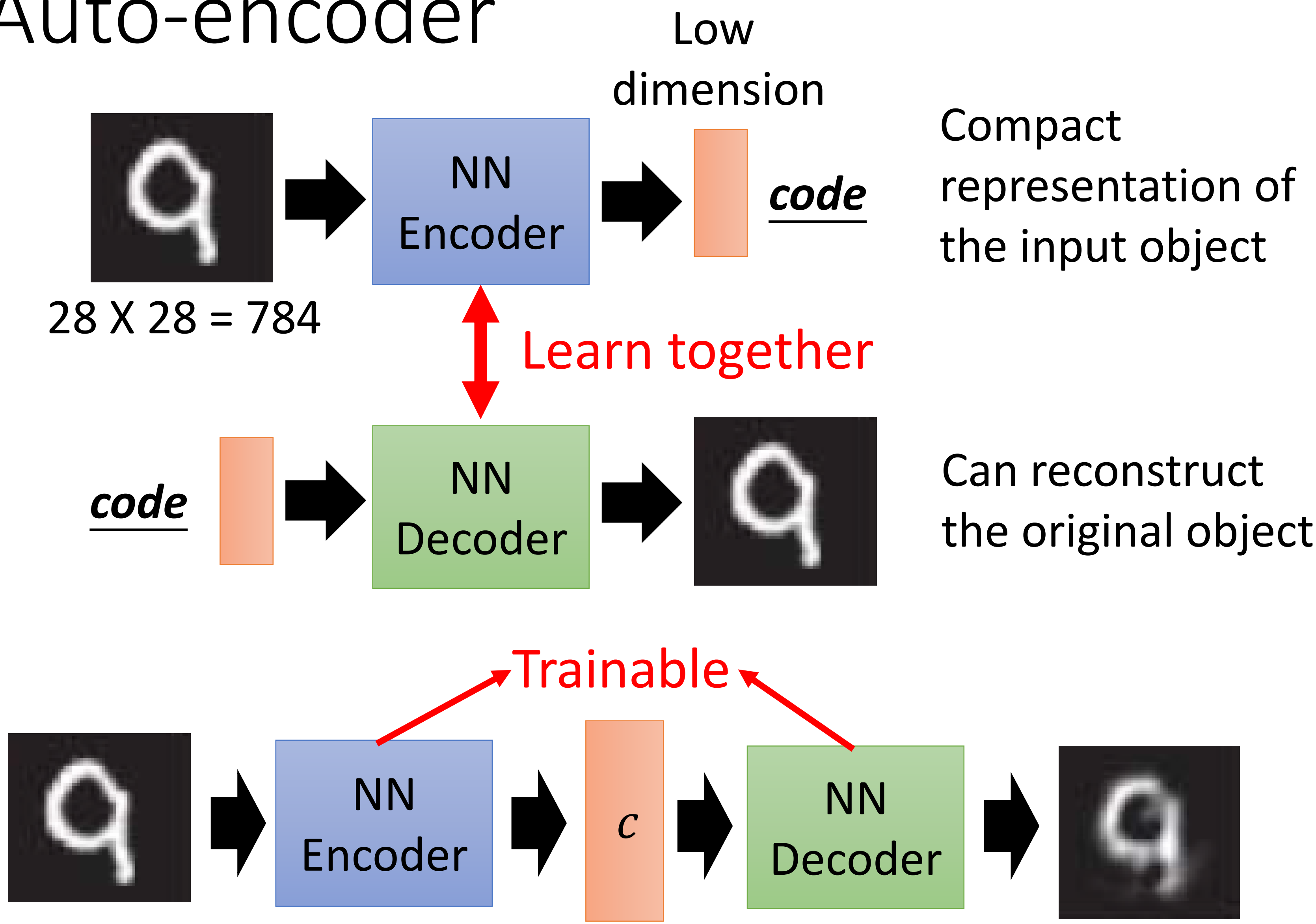
Image:

Encoder in auto-encoder provides the code ☺

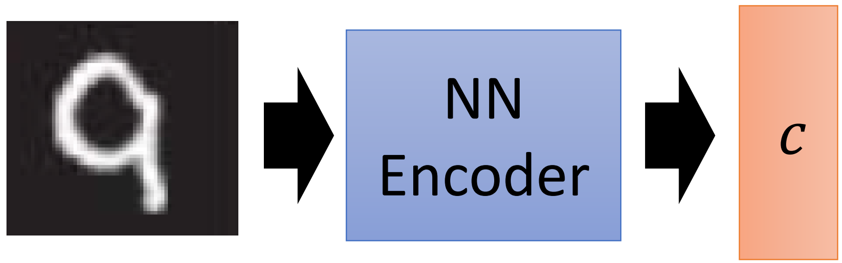


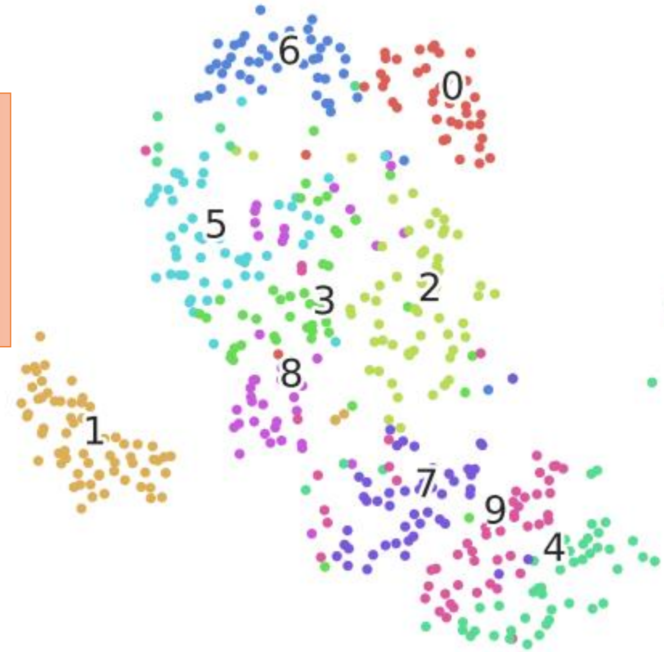


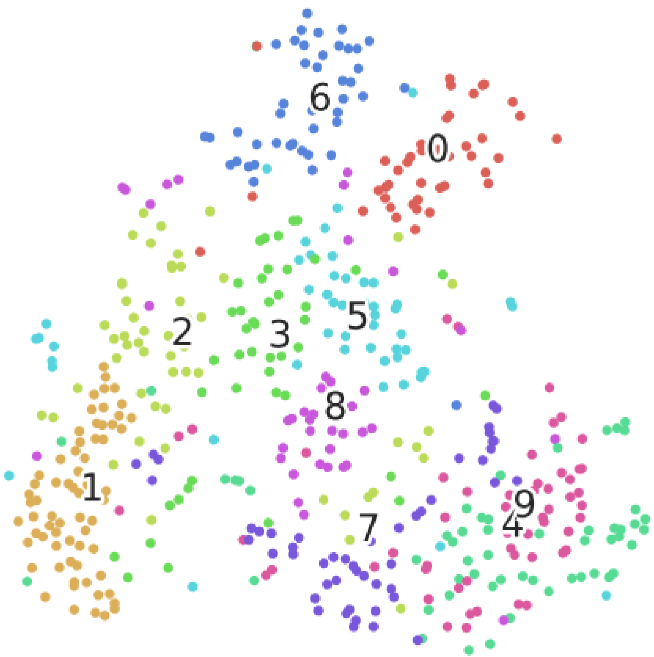
Auto-encoder



Auto-encoder - Example

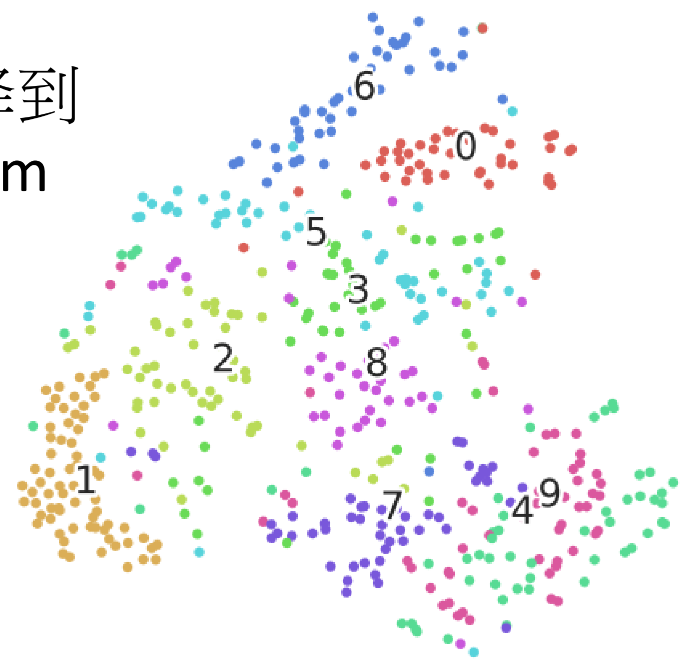






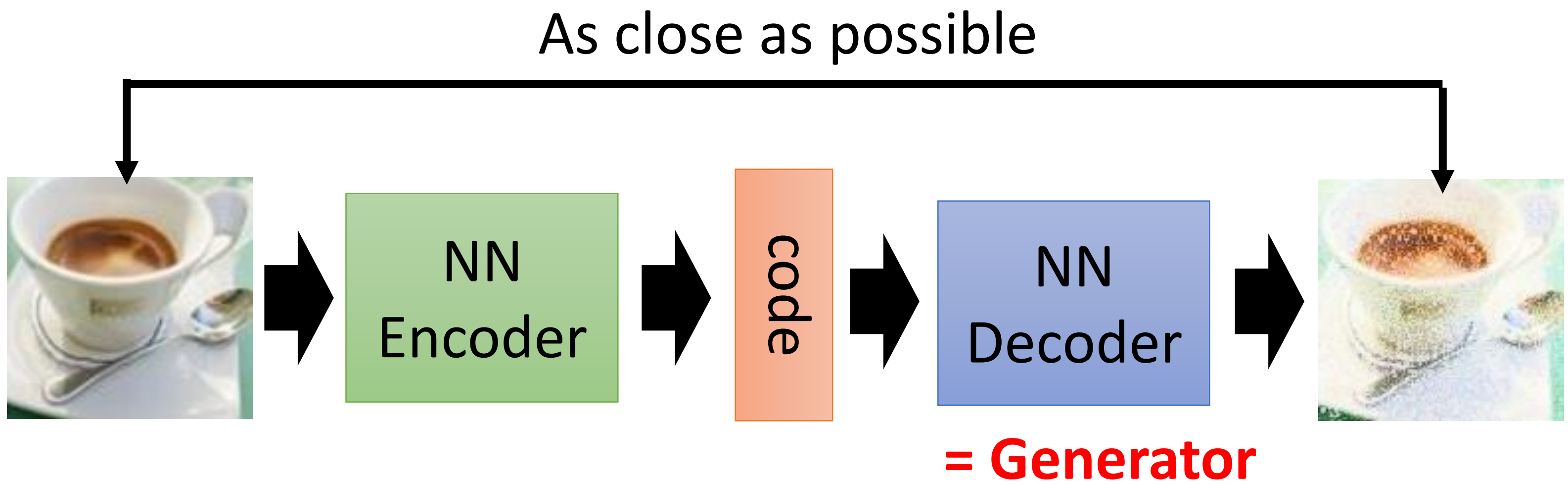
Pixel -> tSNE

PCA 降到
32-dim

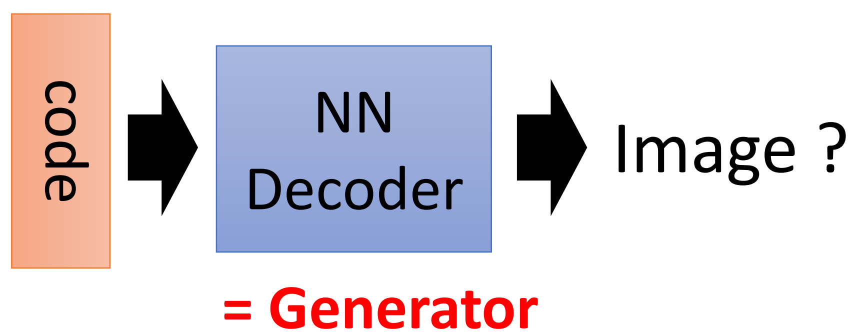


Unsupervised Leaning
without annotation

Auto-encoder

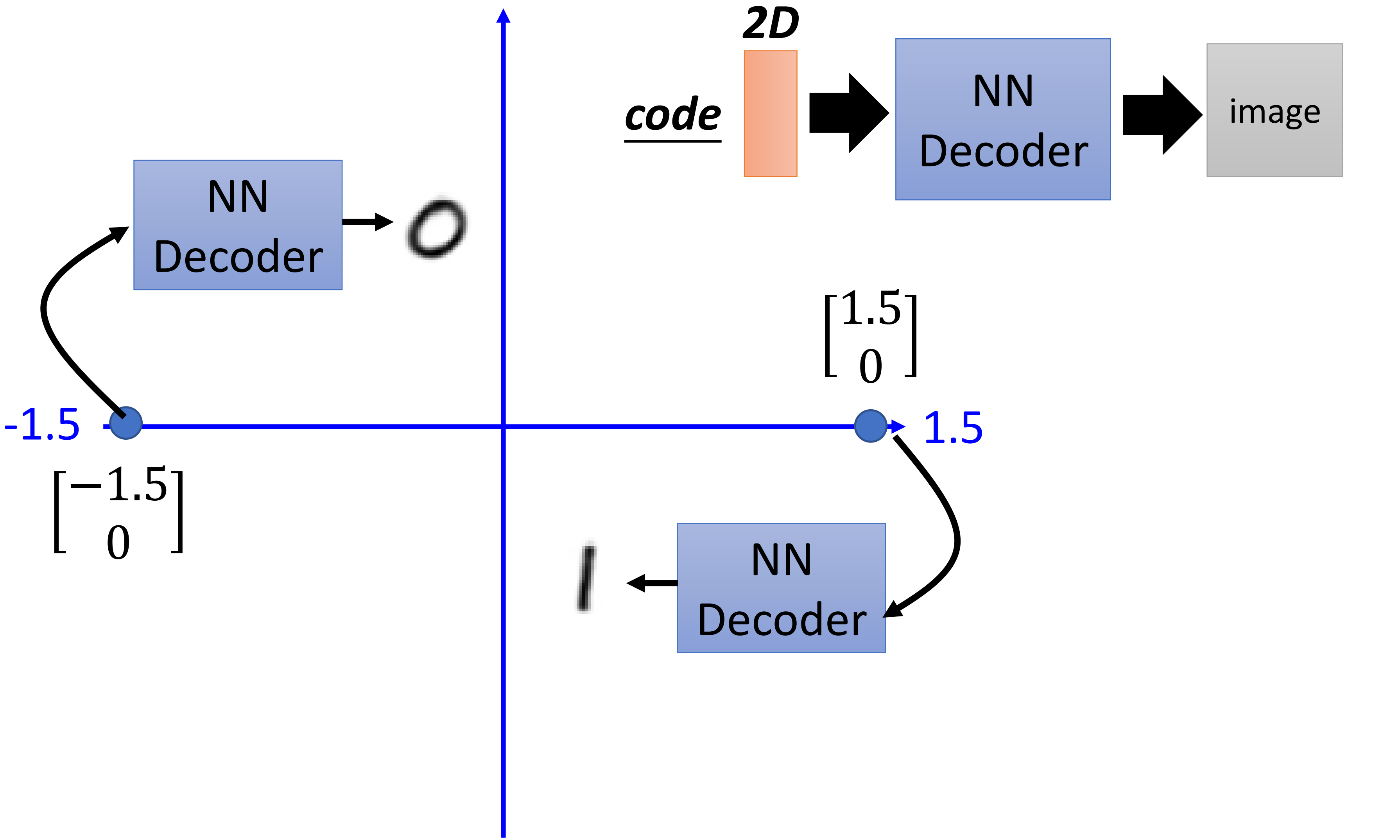


Randomly generate a vector as code



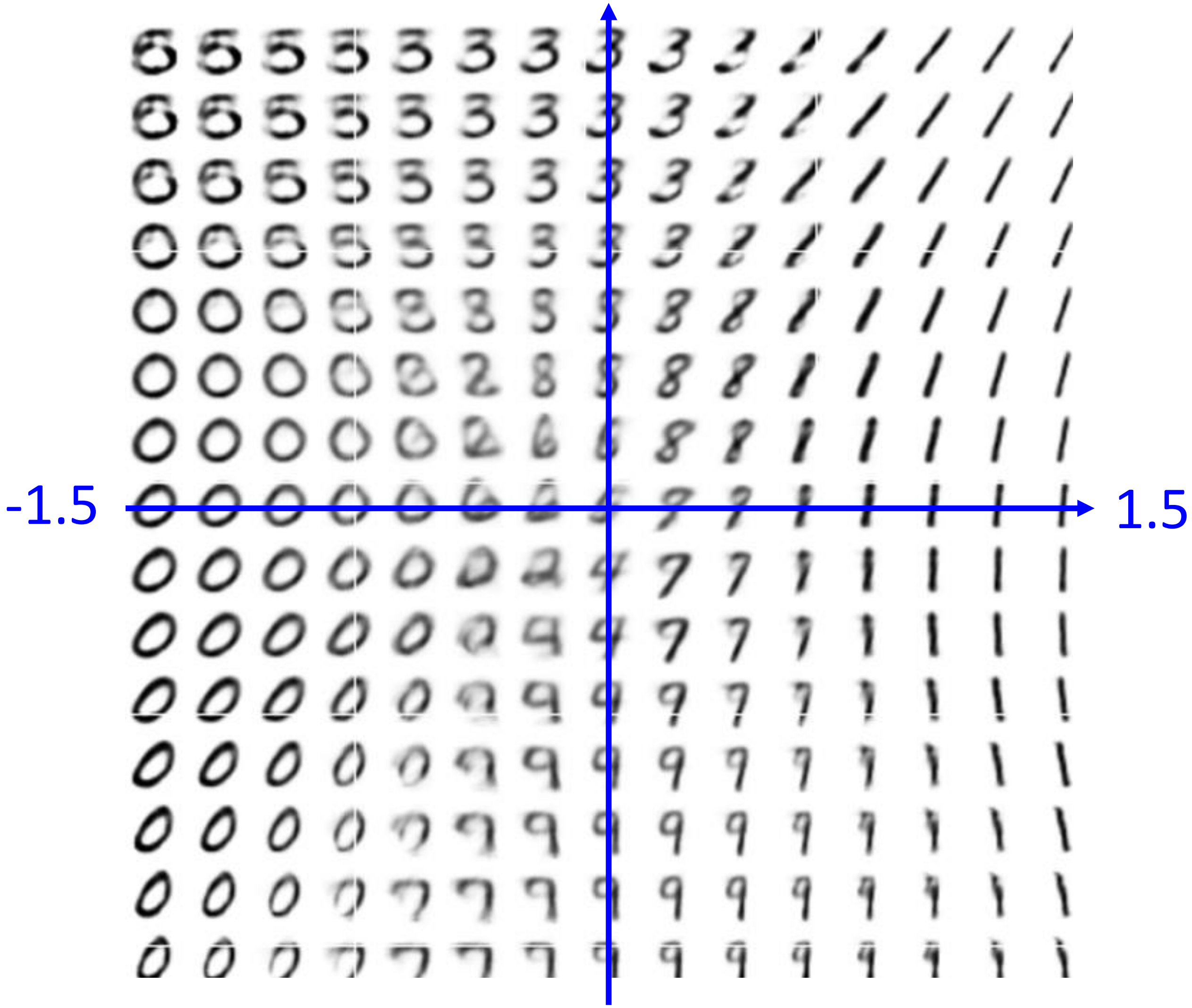
(real examples)

Auto-encoder

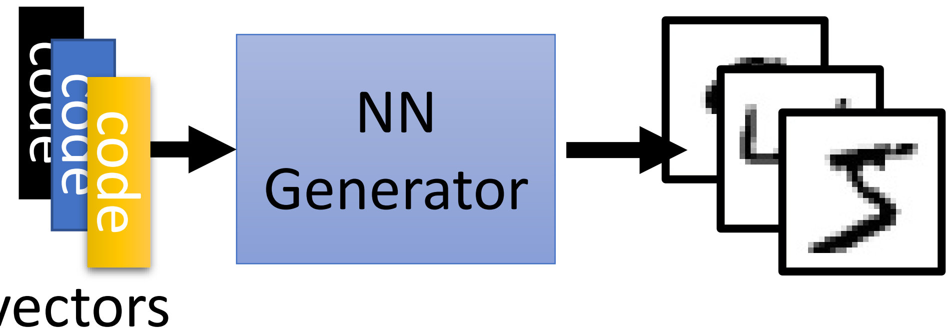


(real examples)

Auto-encoder



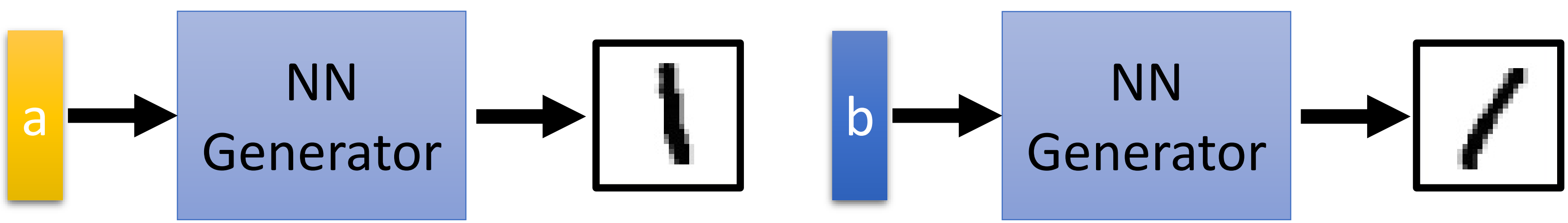
Auto-encoder

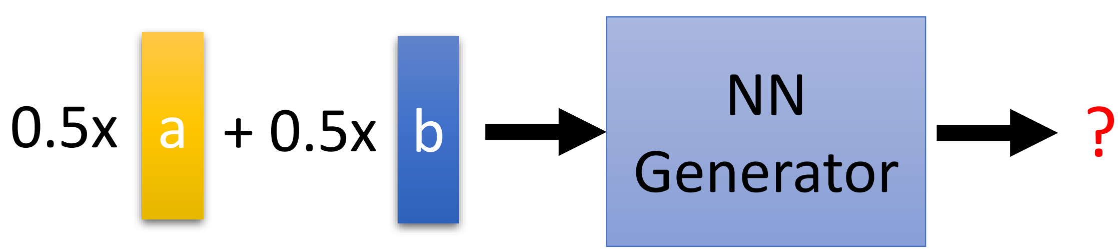


code: $\begin{bmatrix} 0.1 \\ -0.5 \end{bmatrix}$ $\begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}$ $\begin{bmatrix} 0.2 \\ -0.1 \end{bmatrix}$ $\begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}$

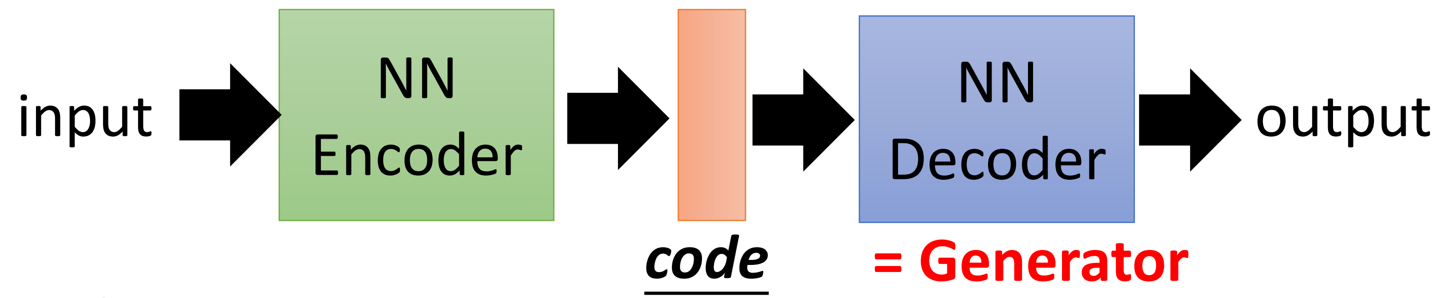
(where does them come from?)

Image:

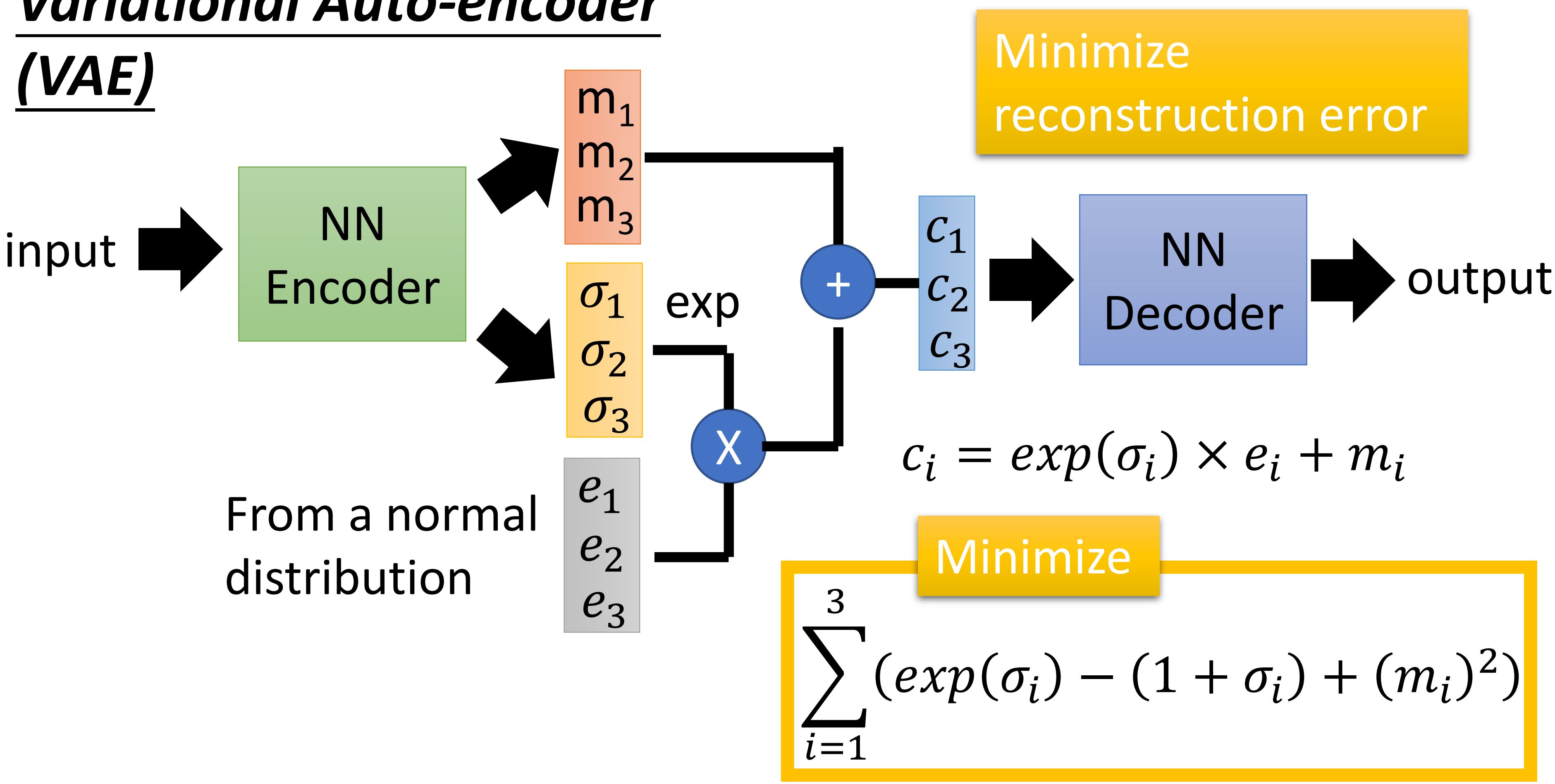




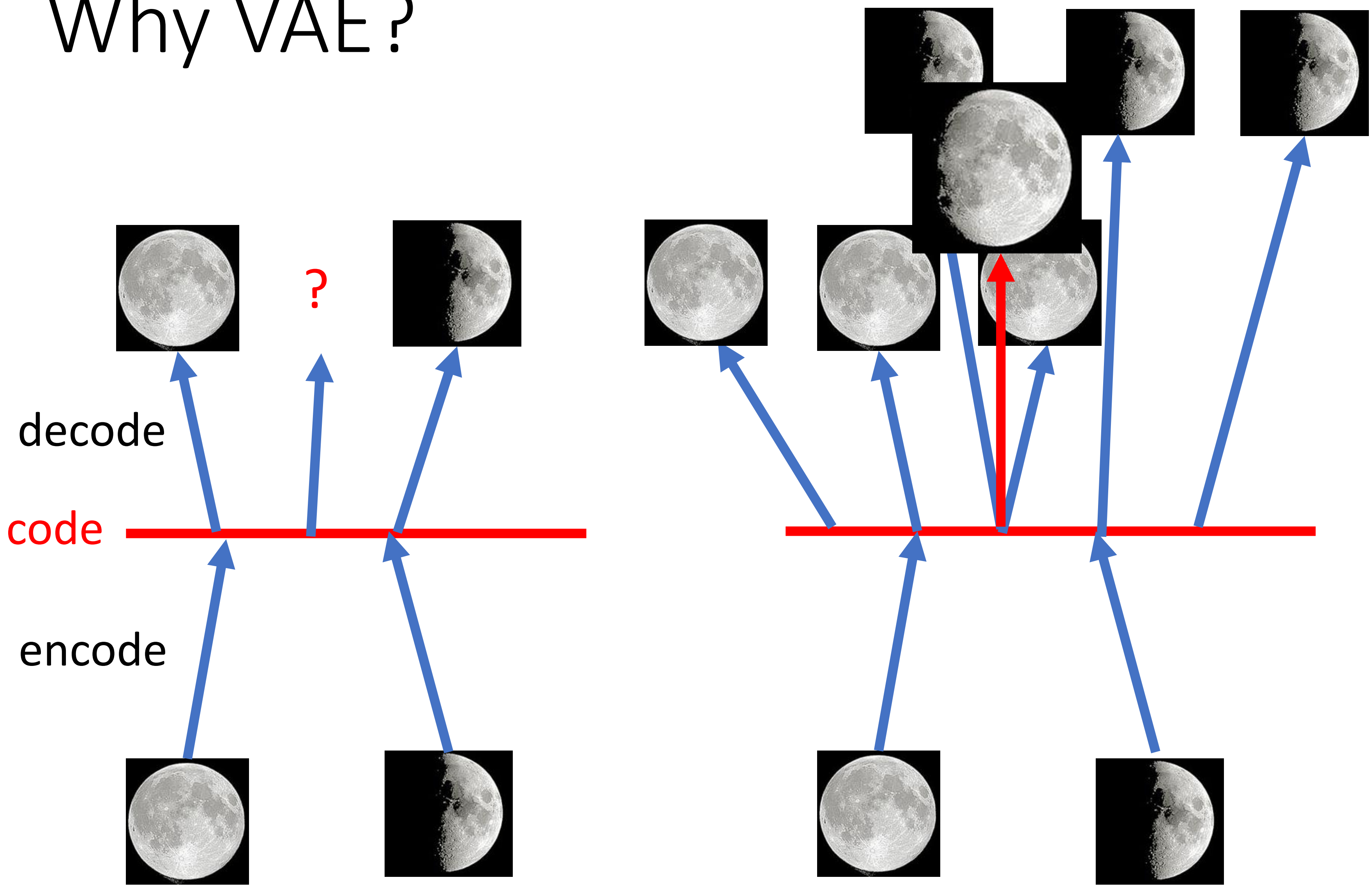
Auto-encoder



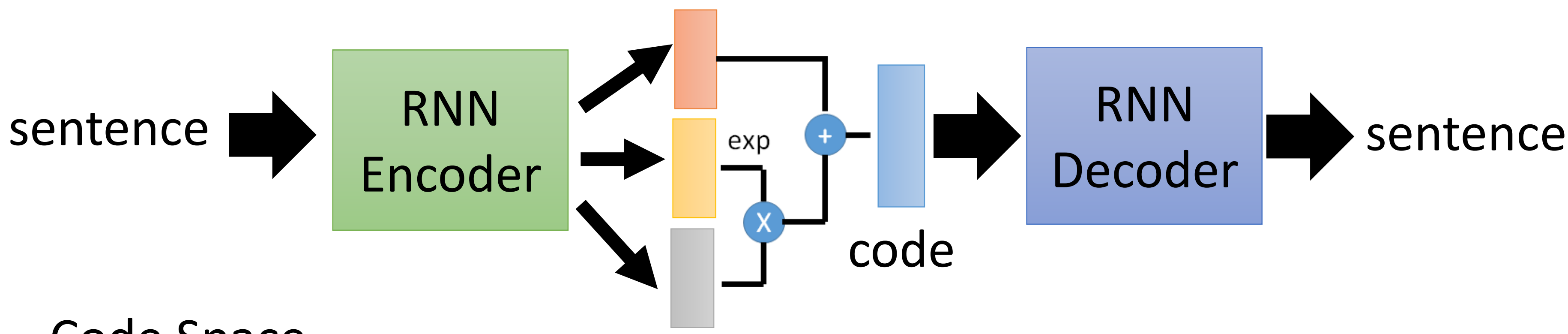
Variational Auto-encoder (VAE)



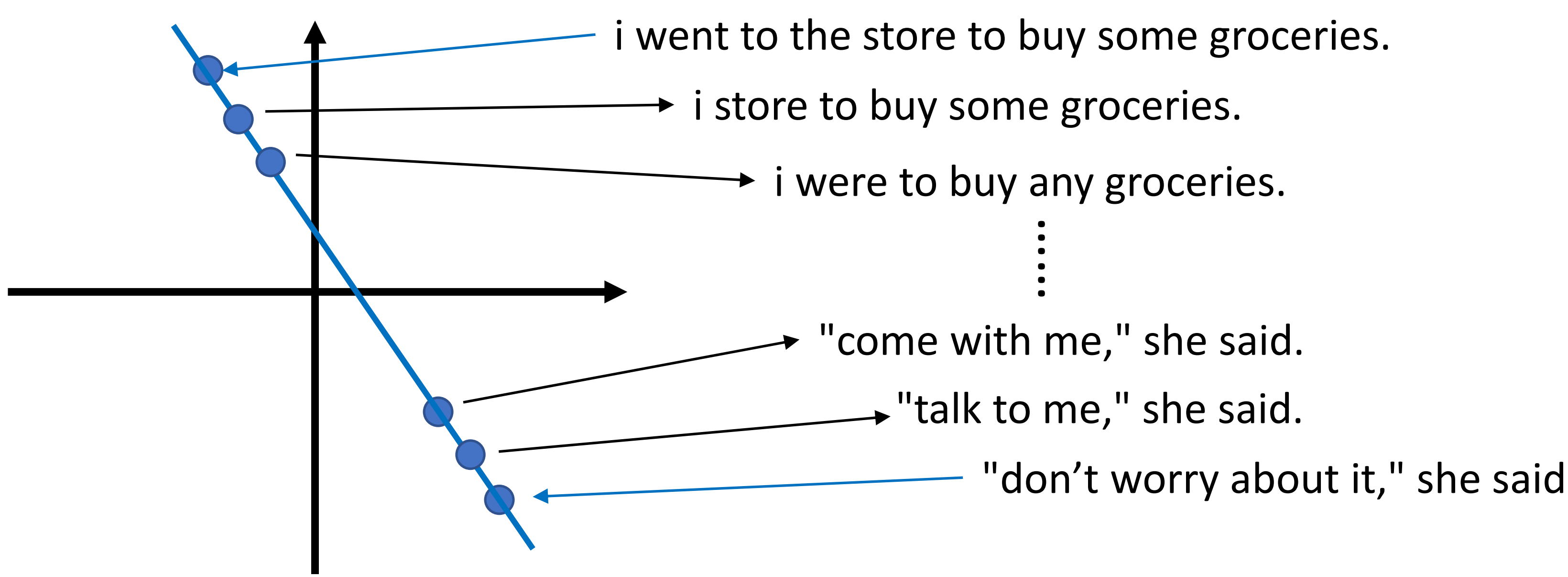
Why VAE?



VAE - Writing Poetry



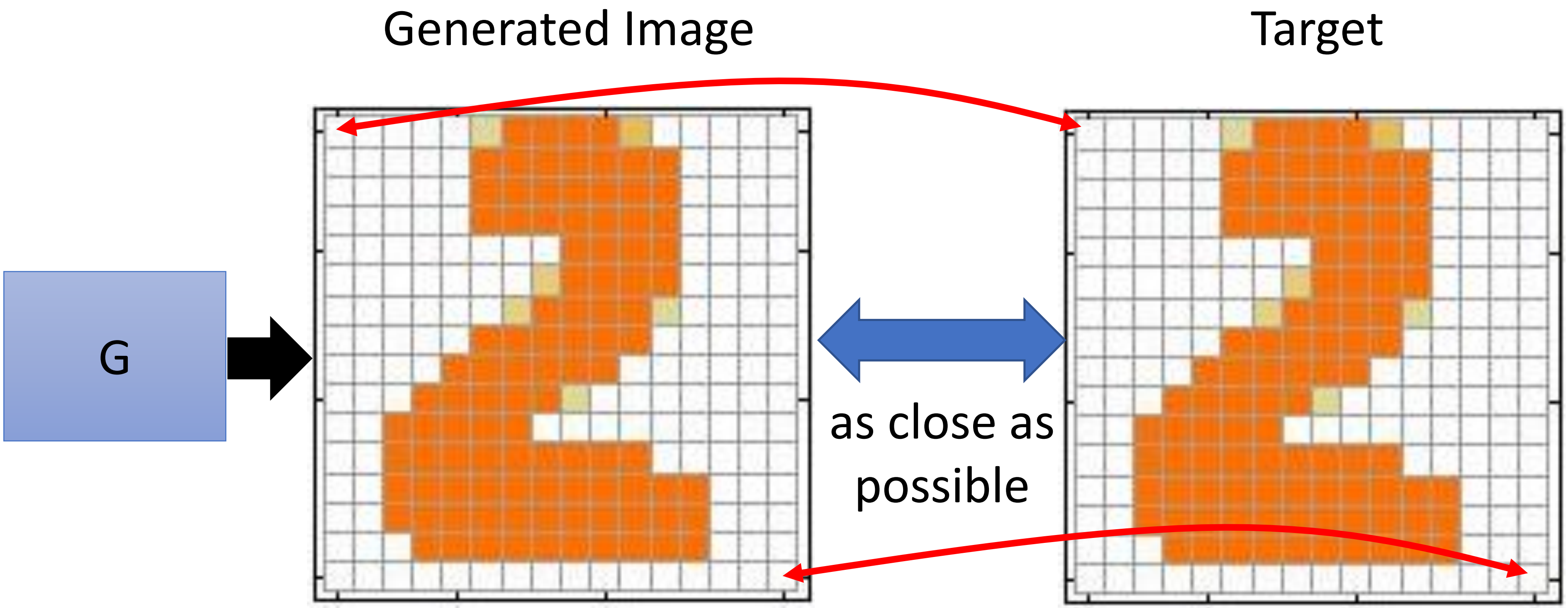
Code Space



Ref: http://www.wired.co.uk/article/google-artificial-intelligence-poetry

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, Samy Bengio, Generating Sentences from a Continuous Space, arXiv prepring, 2015

What do we miss?



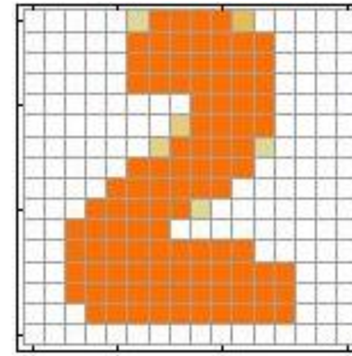
It will be fine if the generator can truly copy the target image.

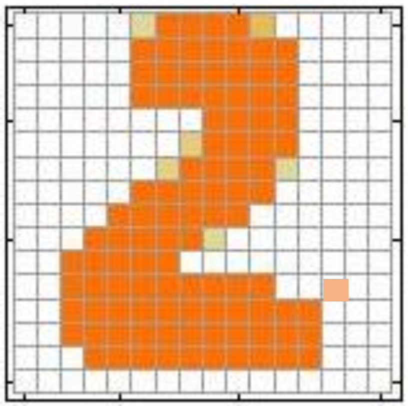
What if the generator makes some mistakes …….

Some mistakes are serious, while some are fine.

What do we miss?

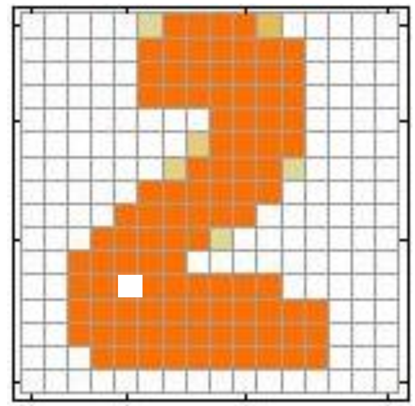
Target





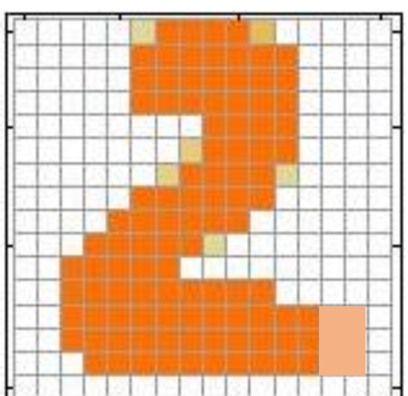
1 pixel error

我覺得不行



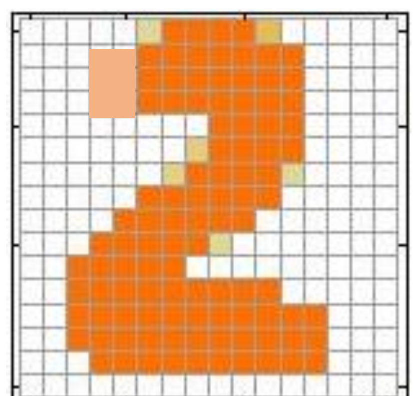
1 pixel error

我覺得不行



6 pixel errors

我覺得
其實 OK

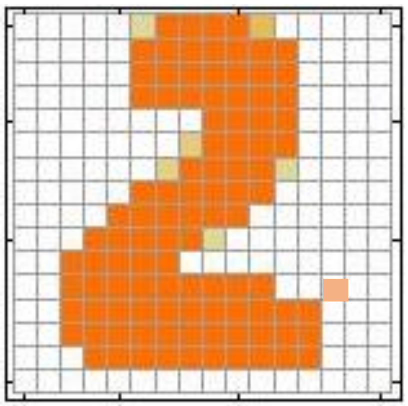


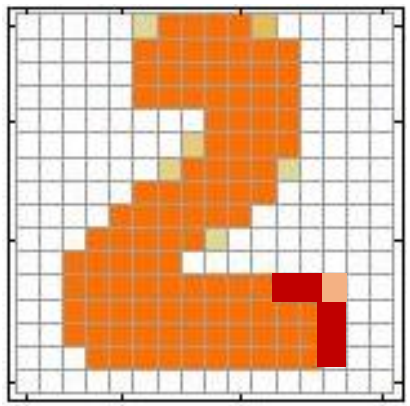
6 pixel errors

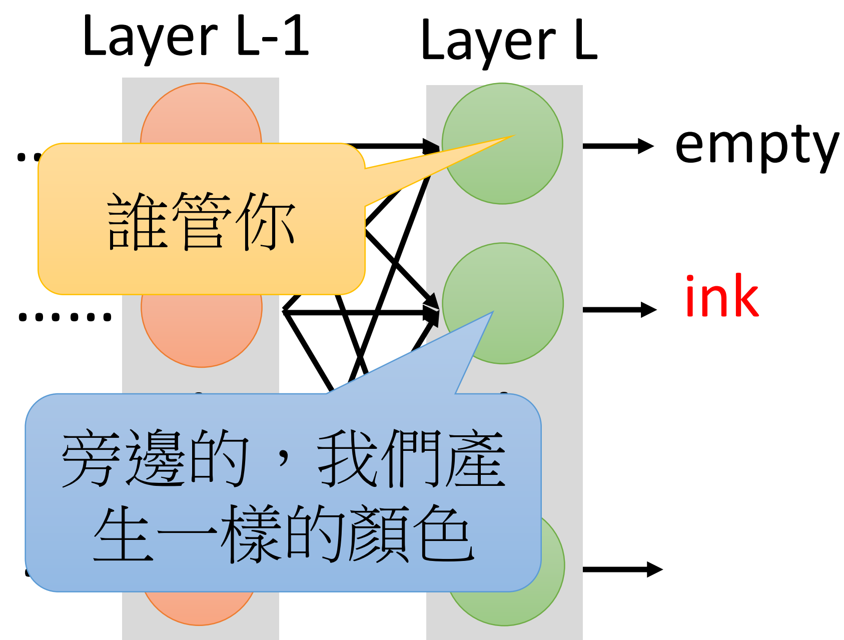
我覺得
其實 OK

What do we miss?

Each neural in output layer corresponds to a pixel.


我覺得不行


我覺得其實 OK



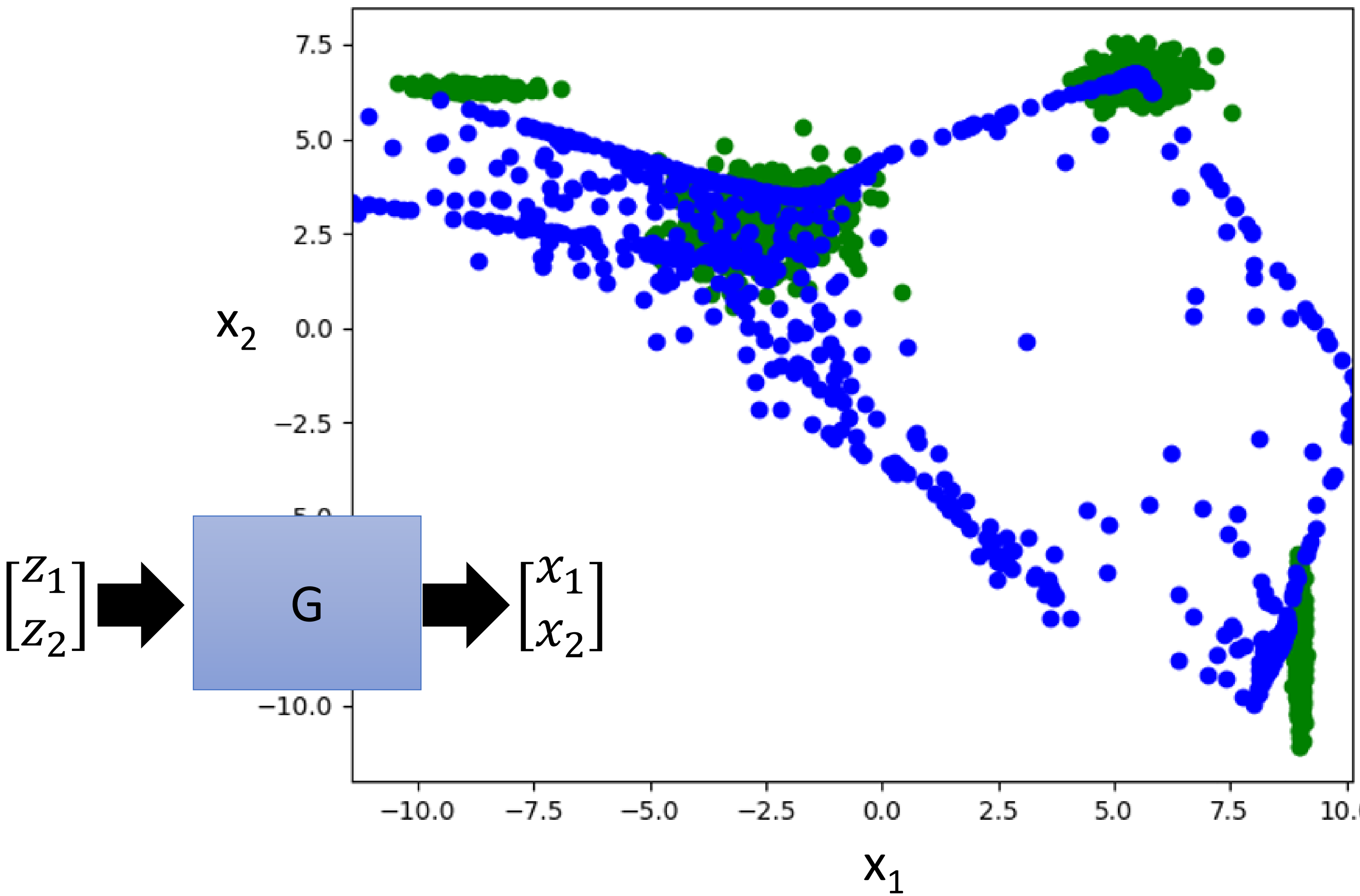
The relation between the components are critical.

The last layer generates each components independently.

Need deep structure to catch the relation between components.

感謝 黃淞楓 同學提供結果

(Variational) Auto-encoder



So many questions ……

Q1: Why generator cannot learn by itself?

Q2: Why discriminator don’t generate object itself?

Q3: How discriminator and generator interact?

Discriminator

Evaluation function, Potential Function, Evaluation Function ...

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

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

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

→ D → 1.0

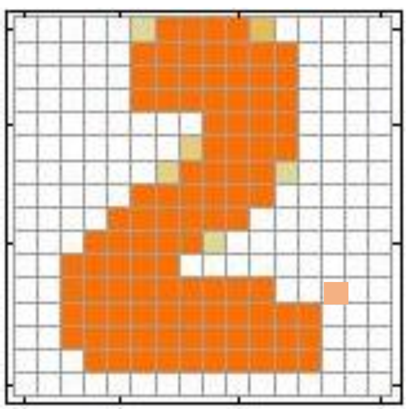
→ D → 0.1

Can we use the discriminator to generate objects?

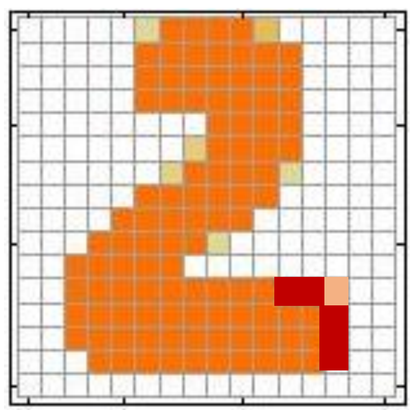
Yes.

Discriminator

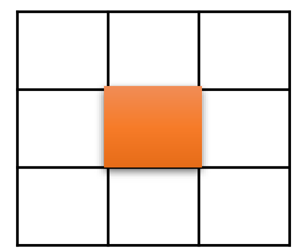
- It is easier to catch the relation between the components by top-down evaluation.



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

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

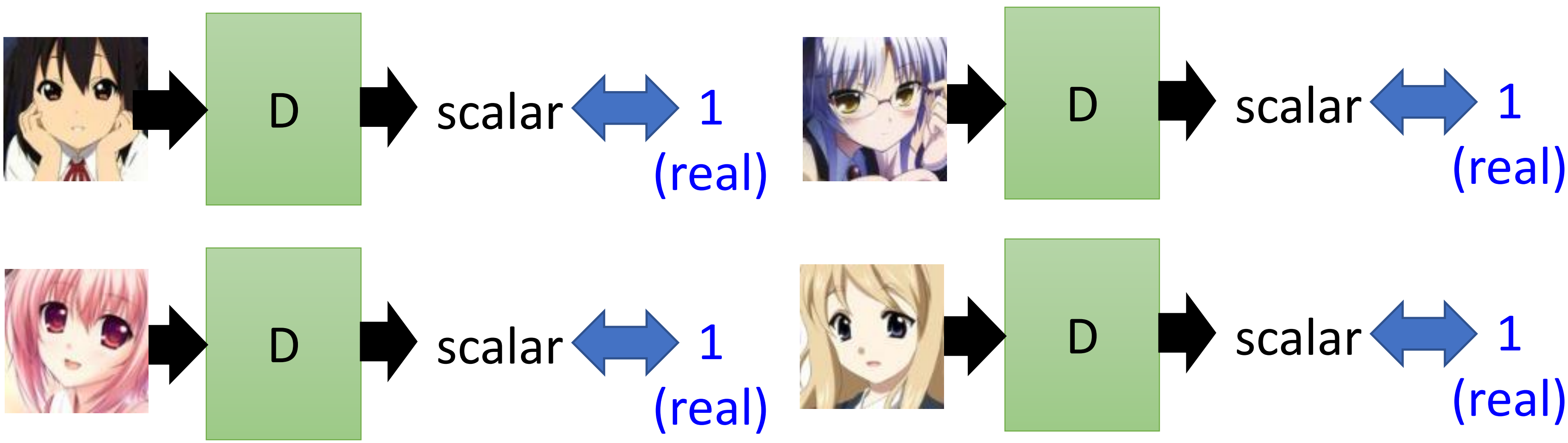
Enumerate all possible x !!!

It is feasible ???

How to learn the discriminator?

Discriminator - Training

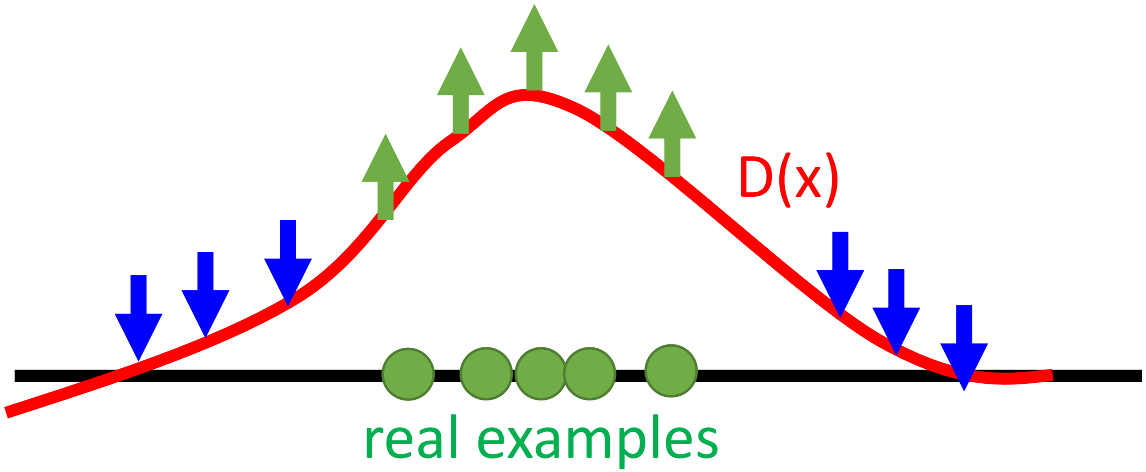
- I have some real images



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

Discriminator training needs some negative examples.

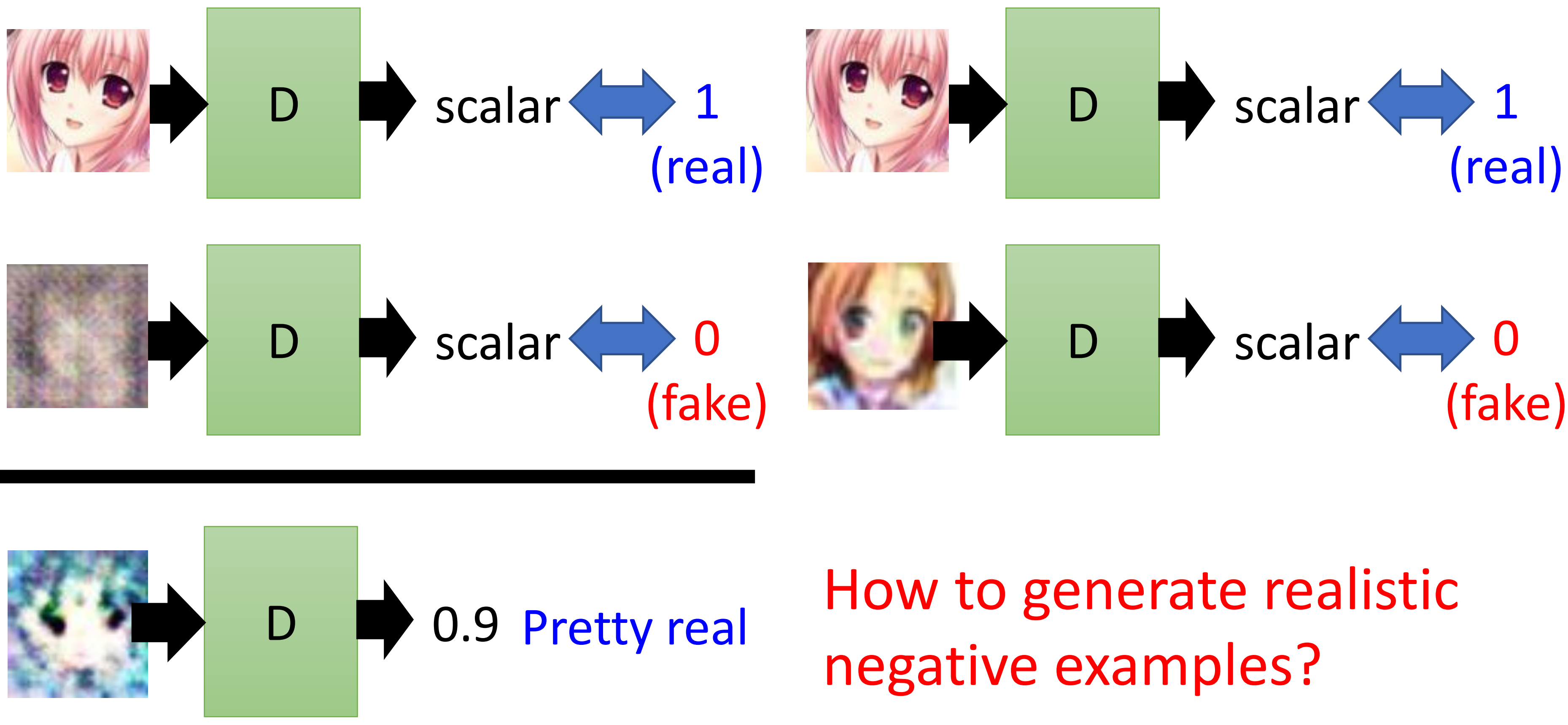
Discriminator - Training



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

Discriminator - Training

- Negative examples are critical.

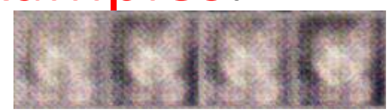


How to generate realistic negative examples?

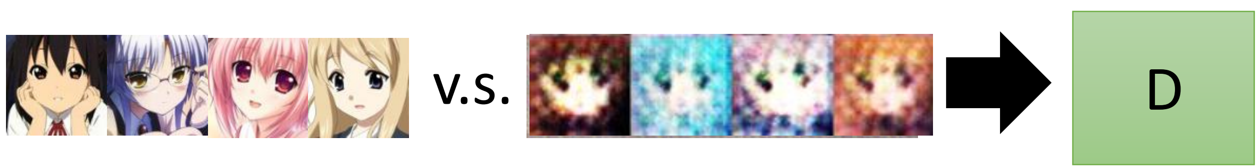
Discriminator - Training

- **General Algorithm**

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



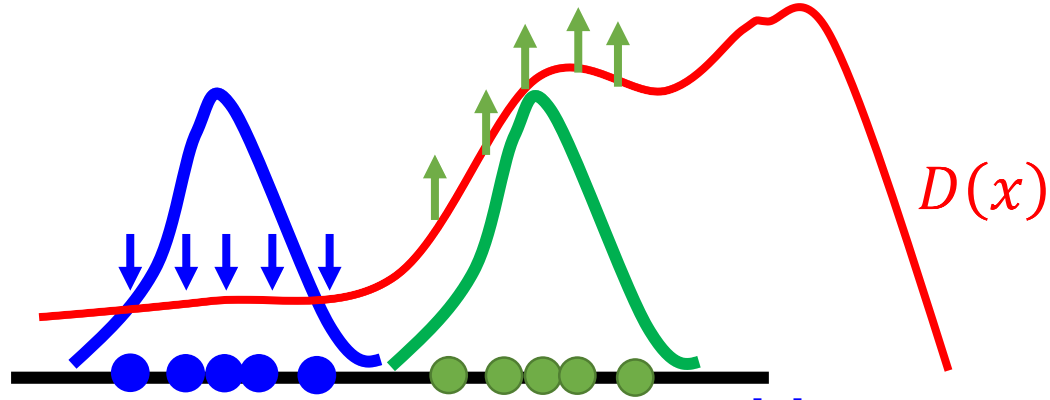
  - In each iteration
    - Learn a discriminator D that can discriminate positive and negative examples.

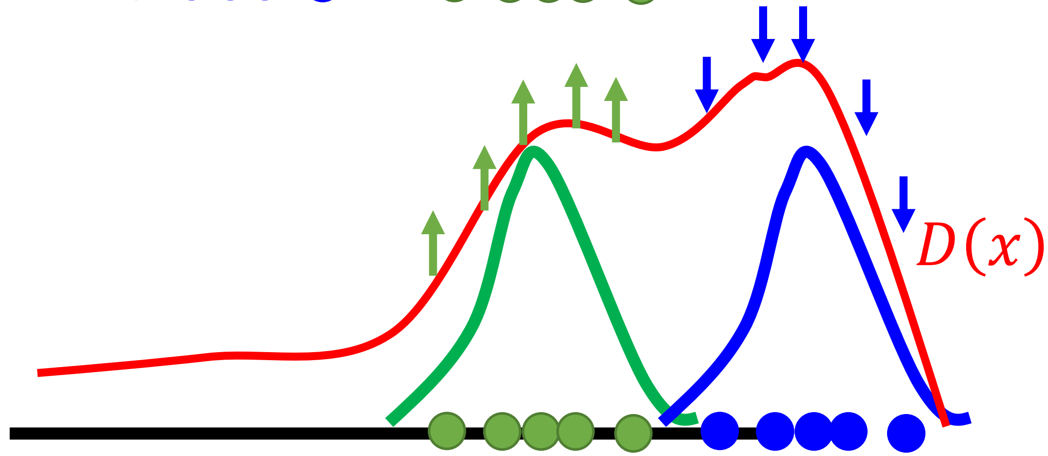


    - Generate negative examples by discriminator D

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

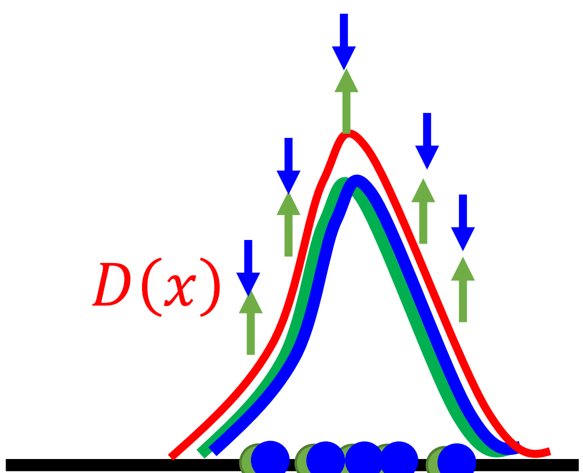
Discriminator - Training

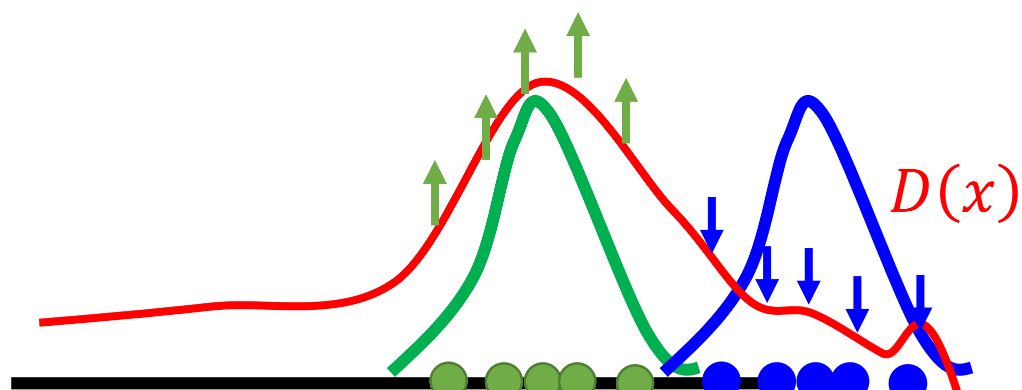


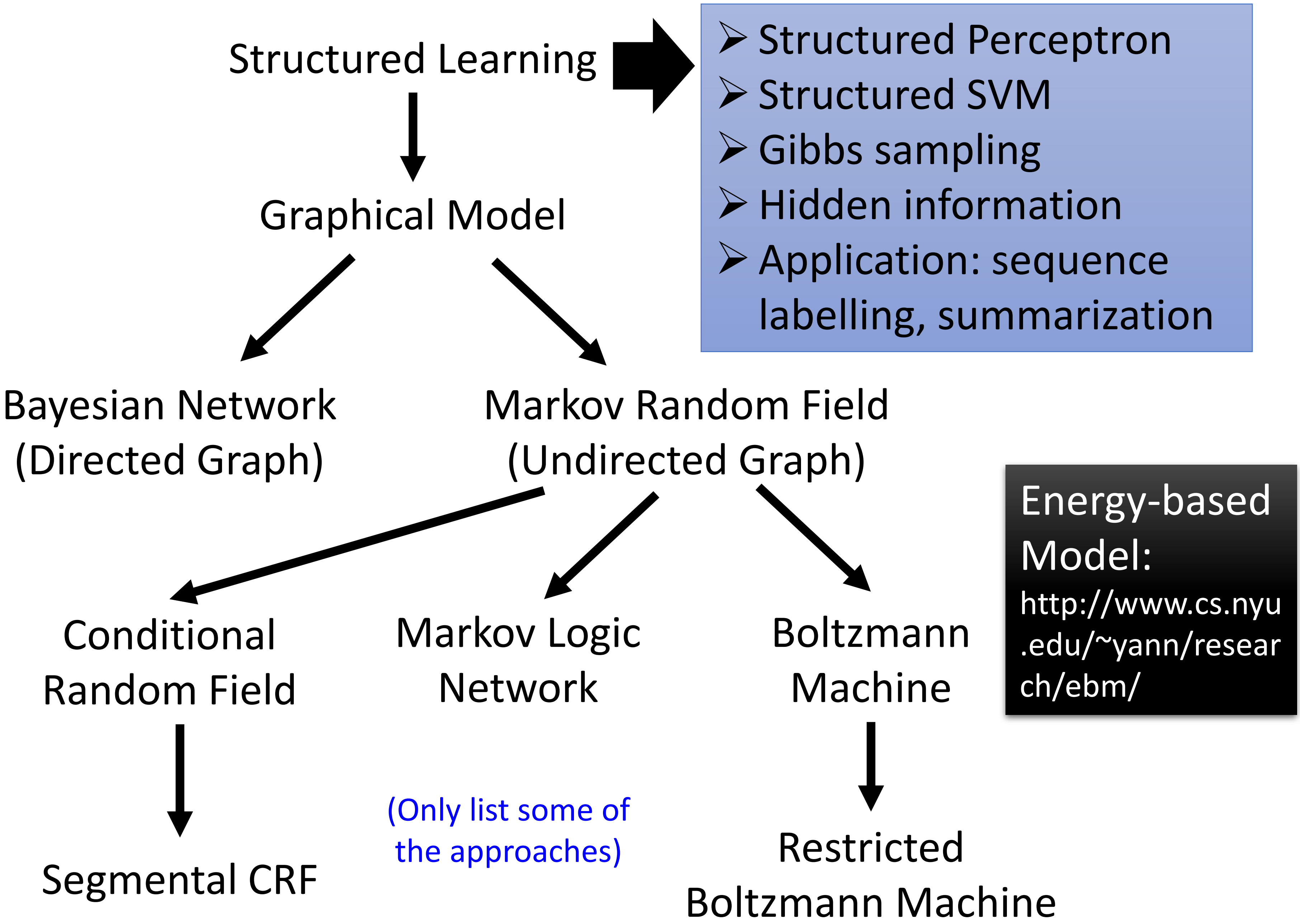


- real
- generated

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)
Energy-based Model:
http://www.cs.nyu.edu/~yann/research/ebm/
Conditional Random Field
Markov Logic Network
Boltzmann Machine
Segmental CRF
(Only list some of the approaches)
Restricted Boltzmann Machine

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?

So many questions ……

Q1: Why generator cannot learn by itself?

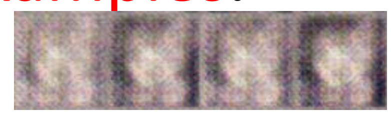
Q2: Why discriminator don’t generate object itself?

Q3: How discriminator and generator interact?

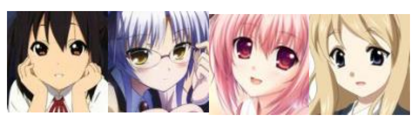
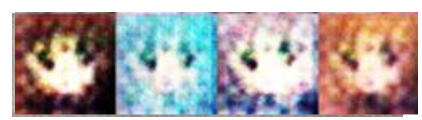
Discriminator - Training

- **<u>General Algorithm</u>**

  - 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.

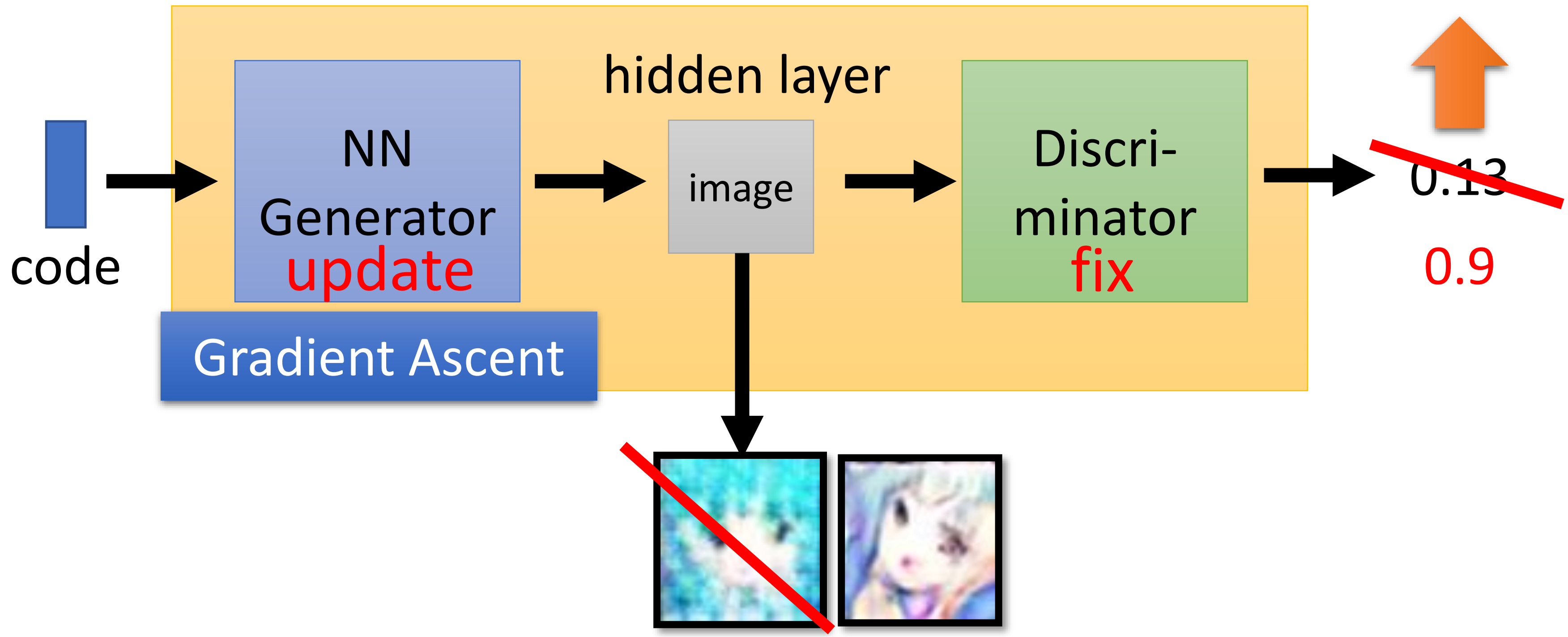
 v.s.  $\rightarrow$ D

    - Generate negative examples by discriminator D

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

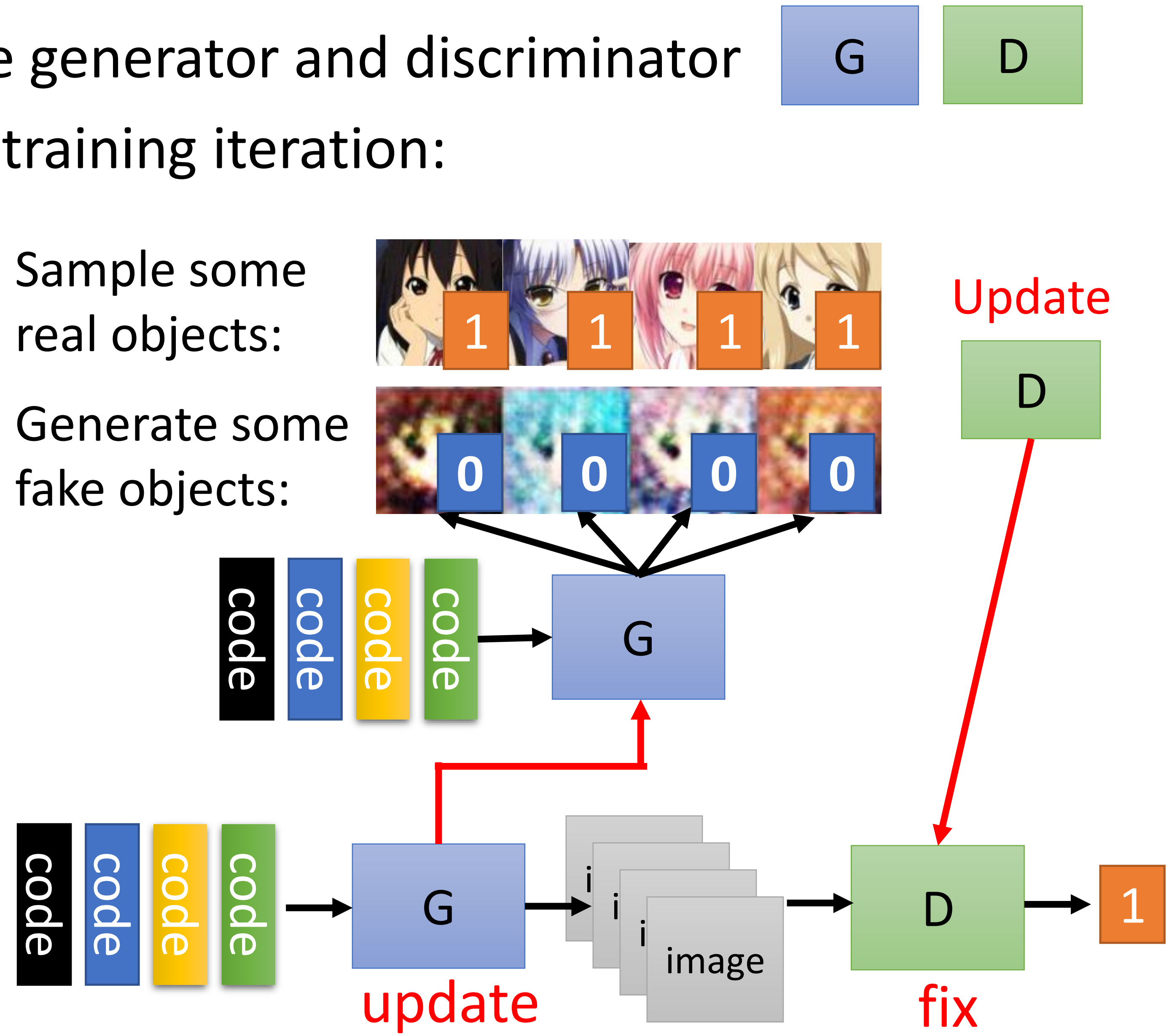
Generating Negative Examples

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



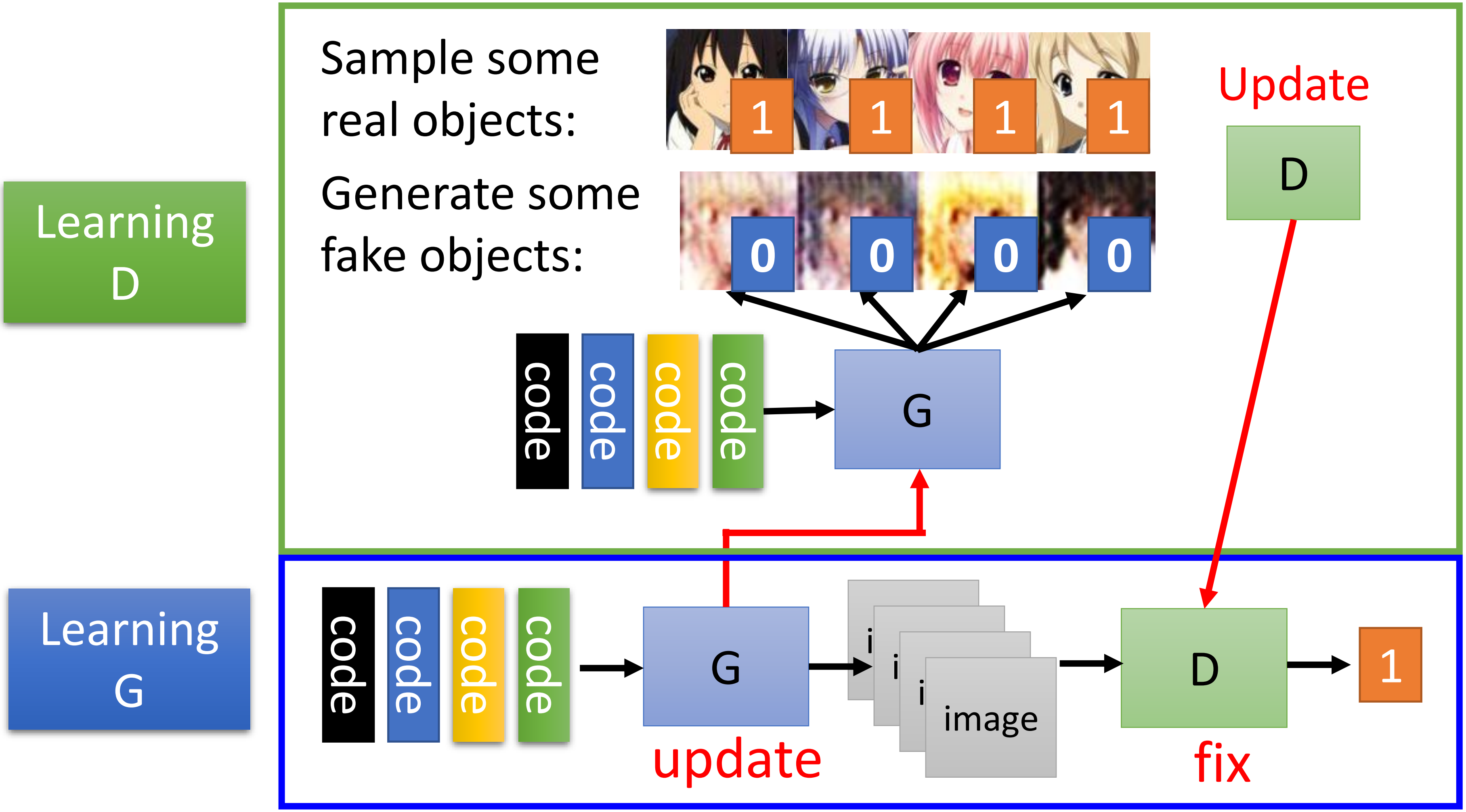
Algorithm

- Initialize generator and discriminator
- In each training iteration:



Algorithm

- Initialize generator and discriminator G D
- In each training iteration:



Benefit of GAN

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

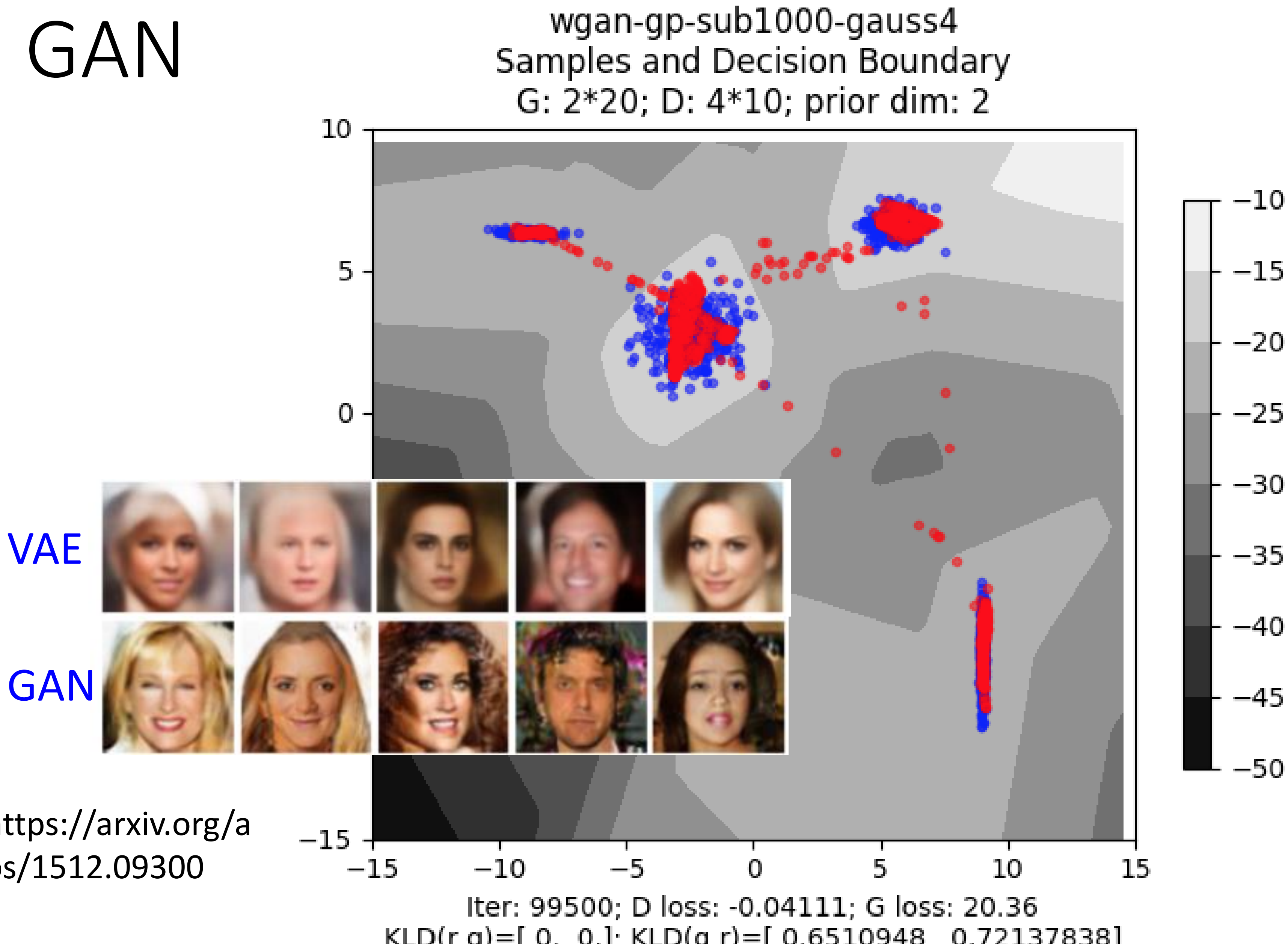
$$\boxed{G \longrightarrow \tilde{x}} = \boxed{\tilde{x} = \arg\max_{x \in X} D(x)}$$

efficient

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

感謝 段逸林 同學提供結果

GAN



https://arxiv.org/abs/1512.09300

Lecture I

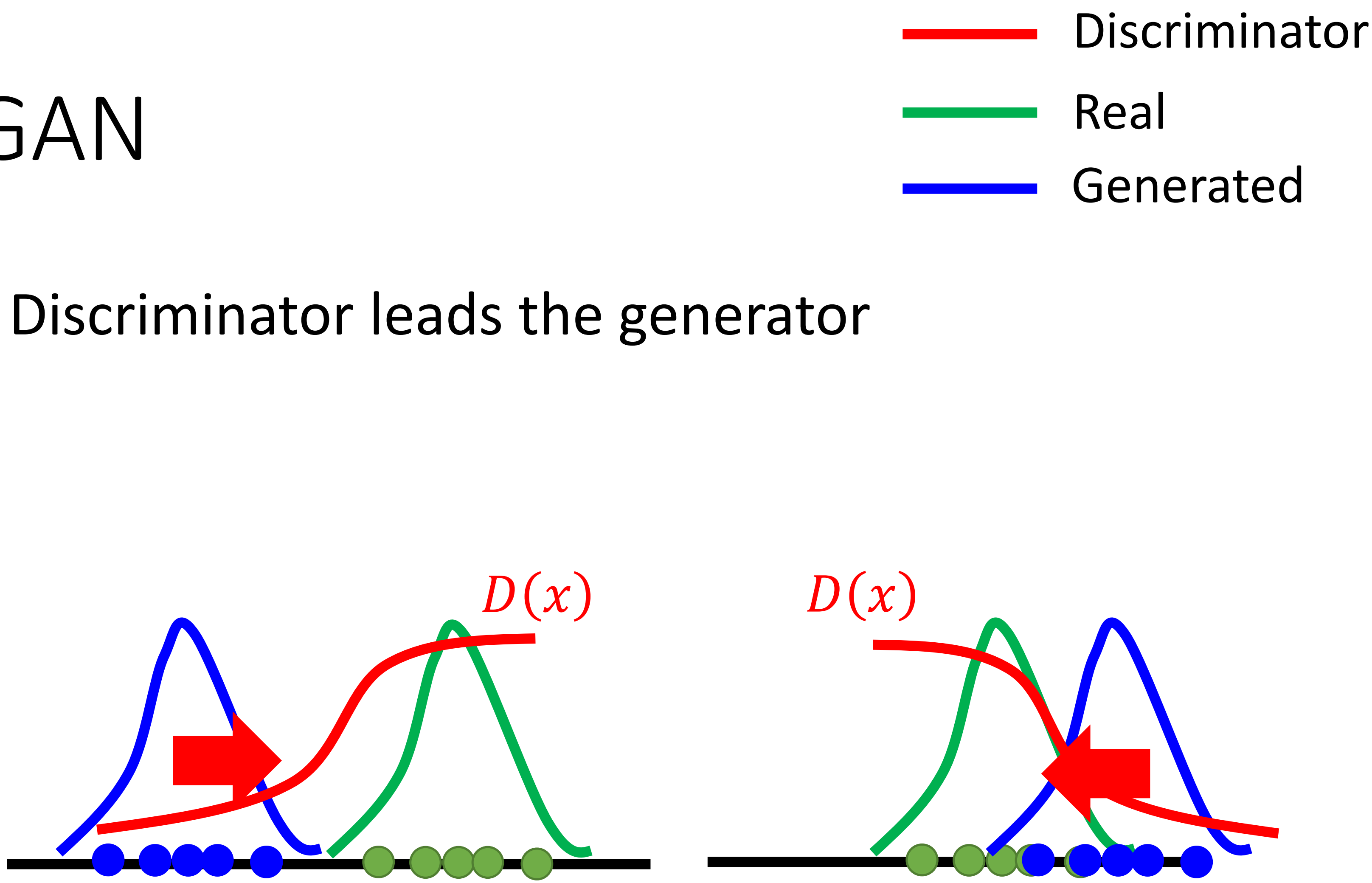
When can I use GAN?

Generation by GAN

Improving GAN

GAN

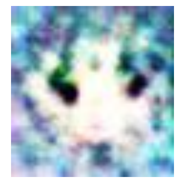
- Discriminator leads the generator



Binary Classifier as Discriminator

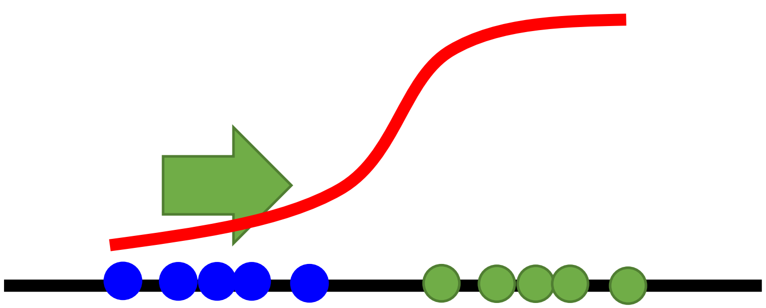
- real
- generated

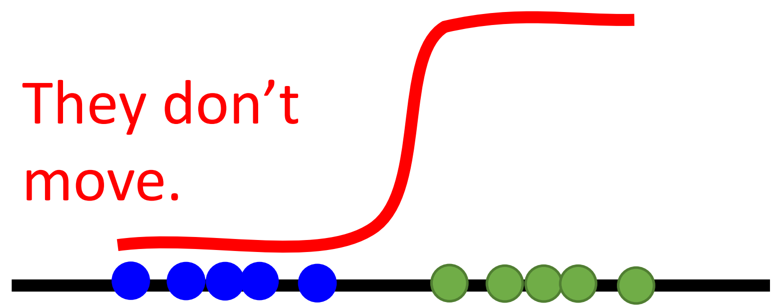
 → Binary Classifier → Real

 → Binary Classifier → Fake

Typical binary classifier uses sigmoid function at the output layer

1 is the largest, 0 is the smallest

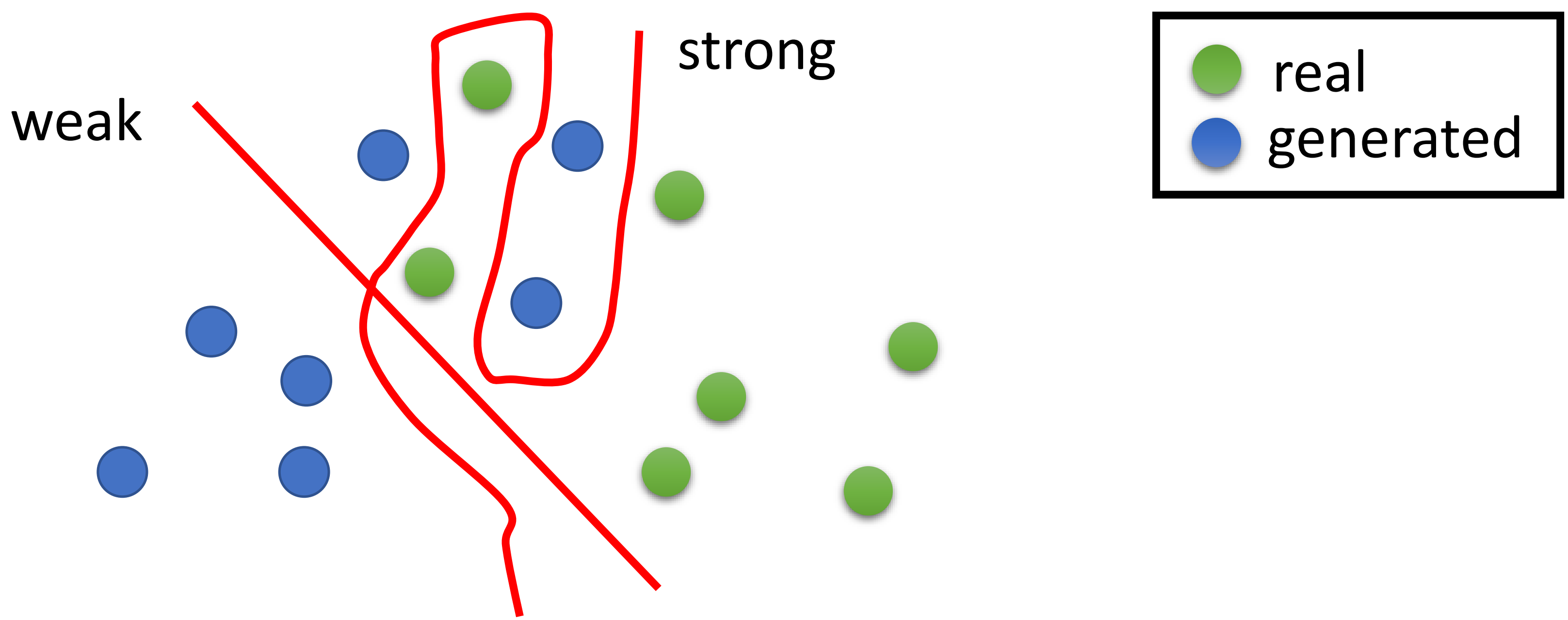




You cannot train your classifier ***too good***……

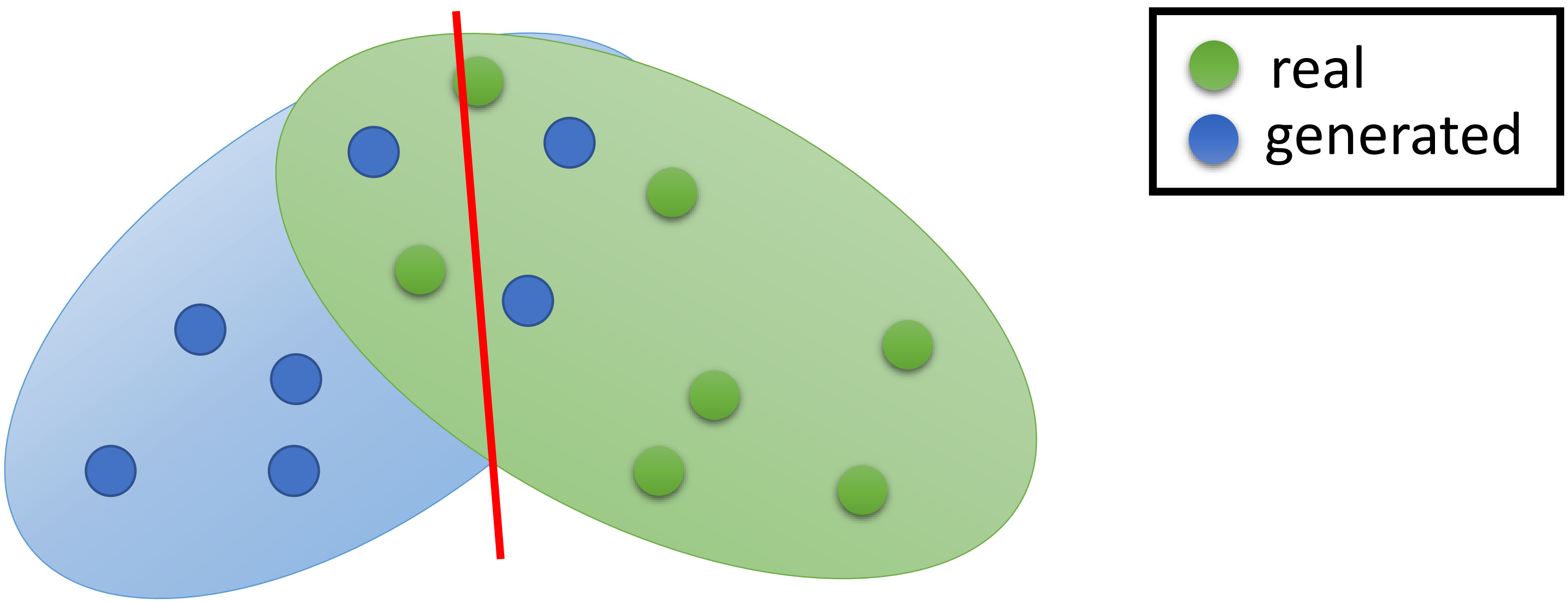
Binary Classifier as Discriminator

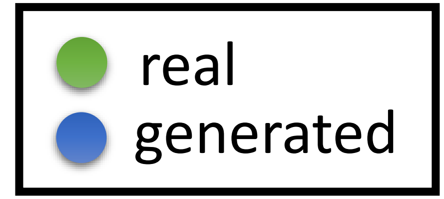
- Don't let the discriminator perfectly separate real and generated data
  - Weaken your discriminator?



Binary Classifier as Discriminator

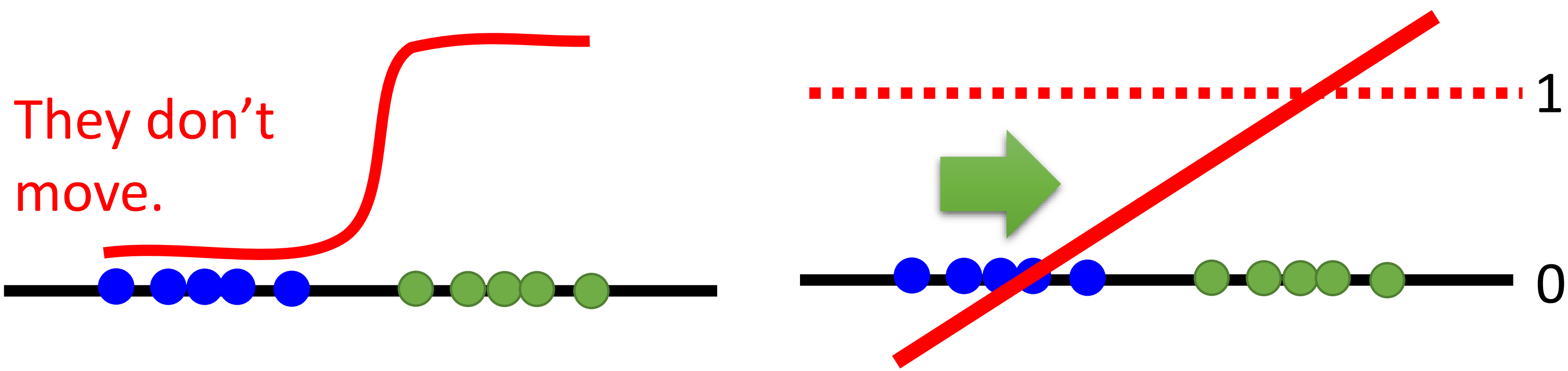
- Don't let the discriminator perfectly separate real and generated data
  - Add noise to input or label?





Least Square GAN (LSGAN)

- Replace sigmoid with linear (replace classification with regression)



real
generated

WGAN

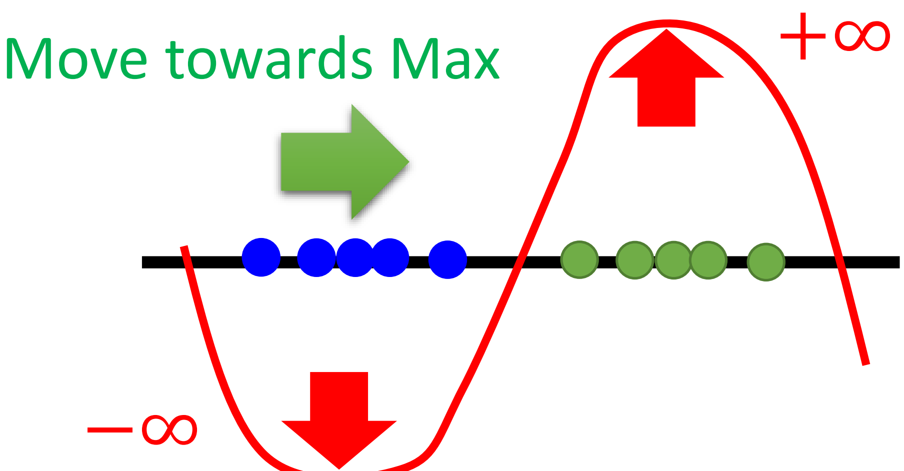
- We want the scores of the real examples as large as possible, generated examples as small as possible.

Binary Classifier → scalar

as large as possible

Binary Classifier → scalar

as small as possible



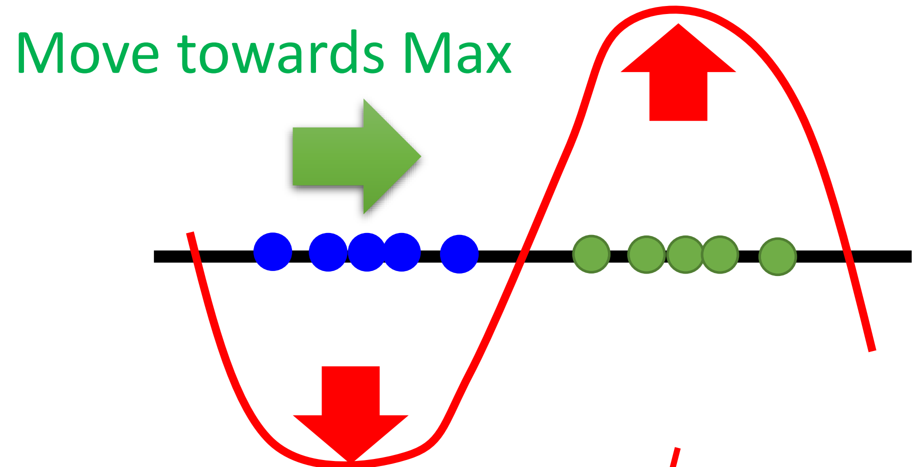
Any problem?

WGAN

The discriminator should be a 1-Lipschitz function.

It should be smooth.

How to realize?



Lipschitz Function

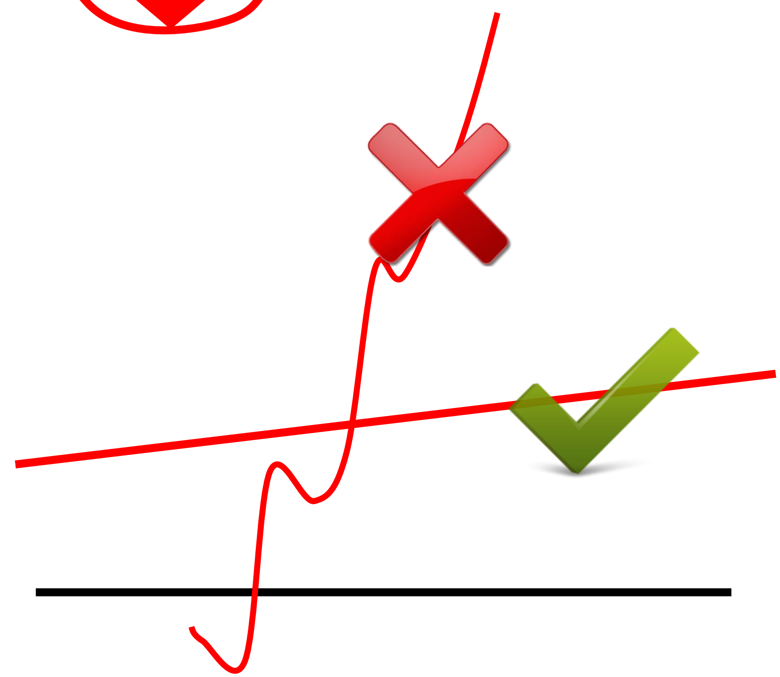
$$\|D(x_1) - D(x_2)\| \le K\|x_1 - x_2\|$$

Output change

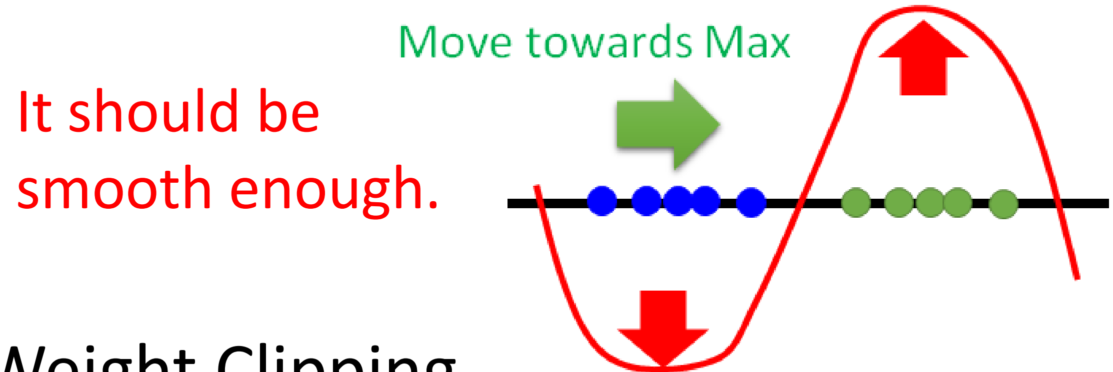
Input change

K=1 for "$1 - Lipschitz$"

Do not change fast



WGAN



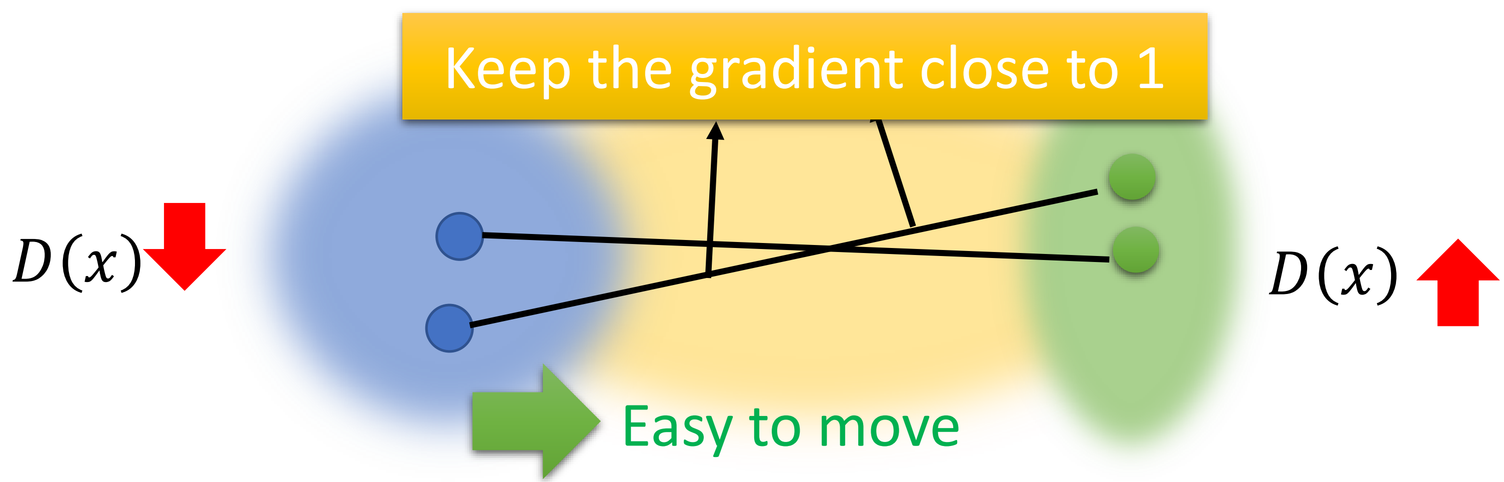
- Original WGAN → Weight Clipping

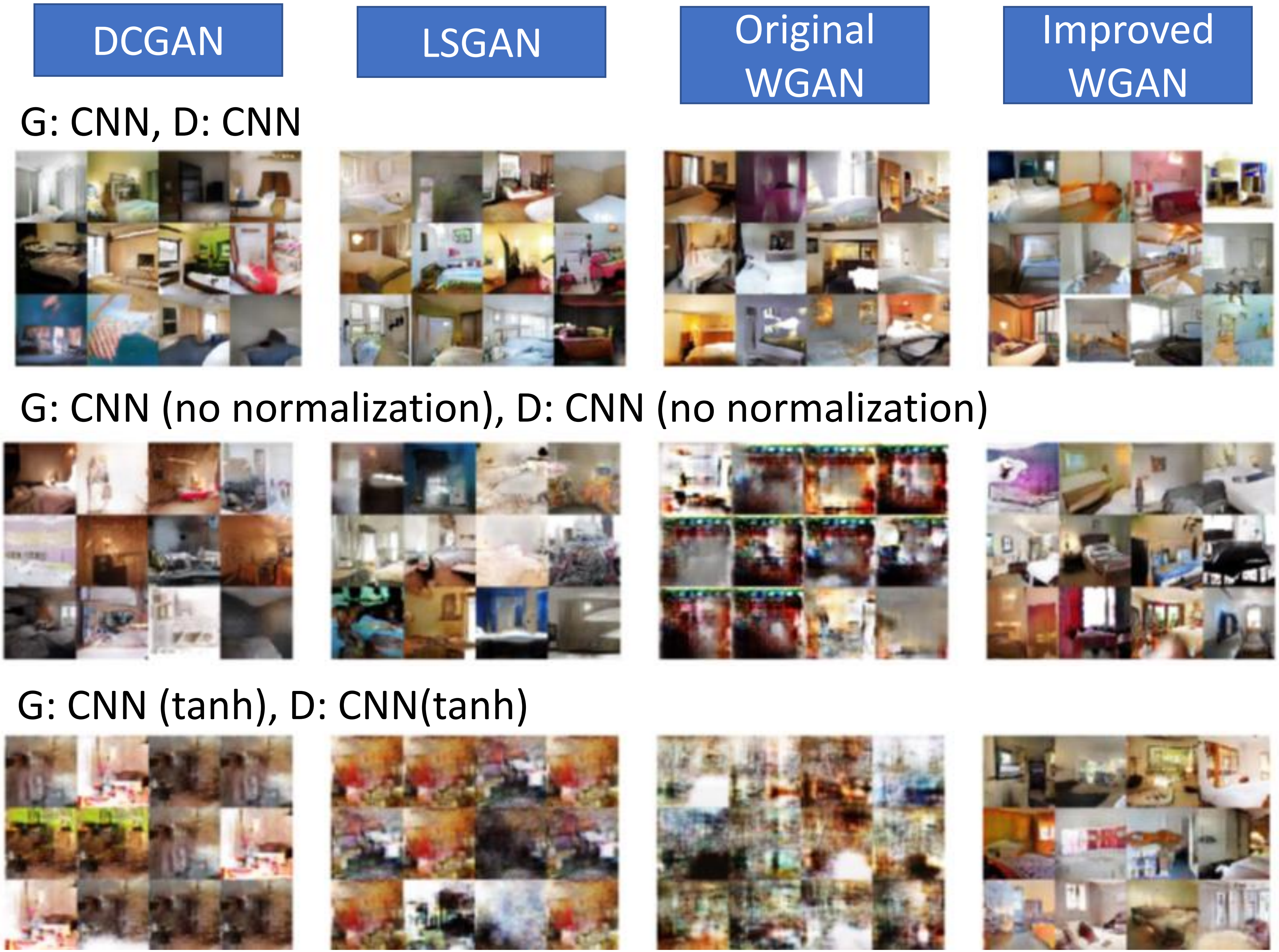
  Force the parameters w between c and -c

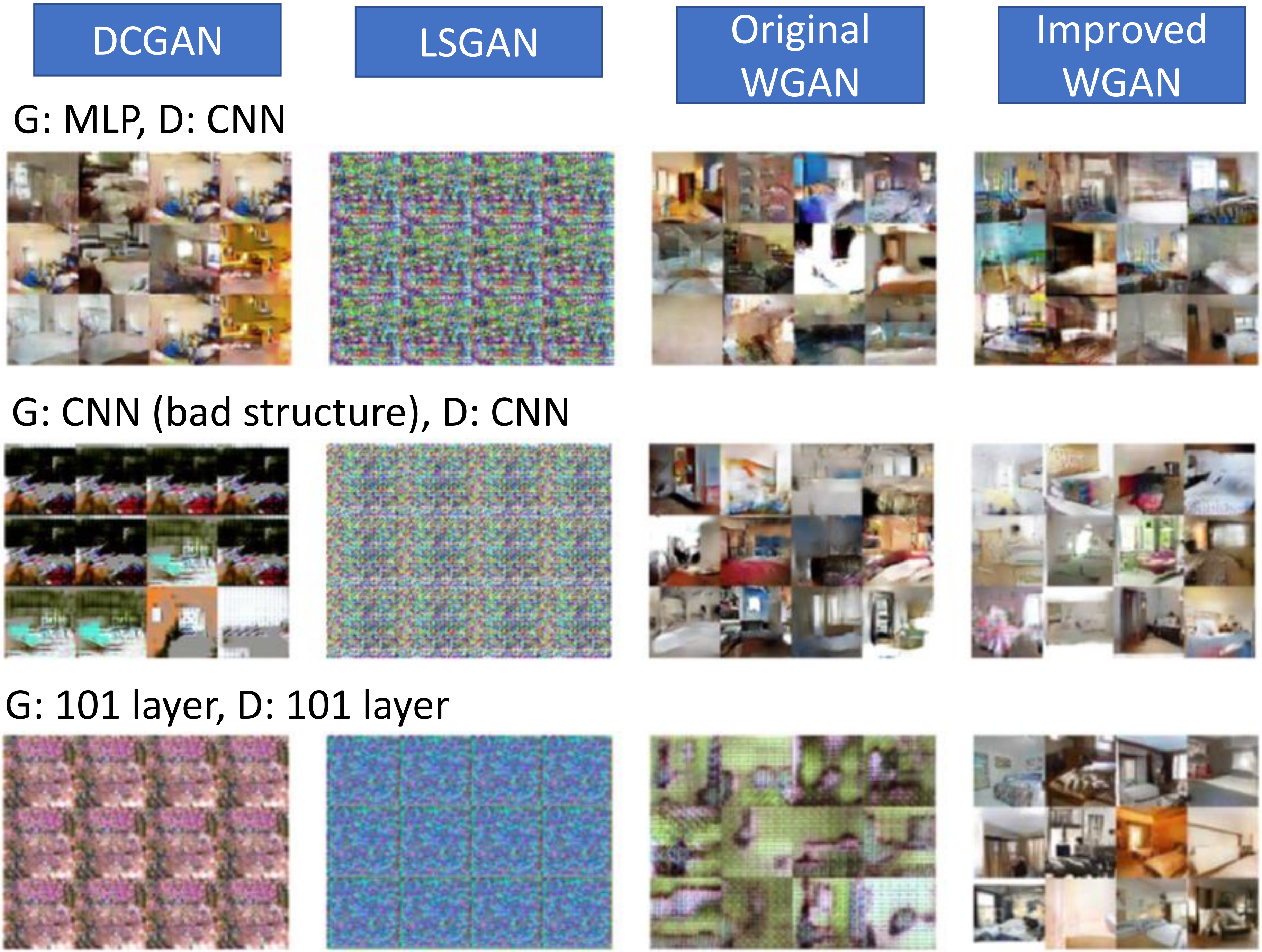
  After parameter update, if w > c, w = c; if w < -c, w = -c

  Do not truly maximize (minimize) the real (generated) examples

- Improved WGAN → Gradient Penalty



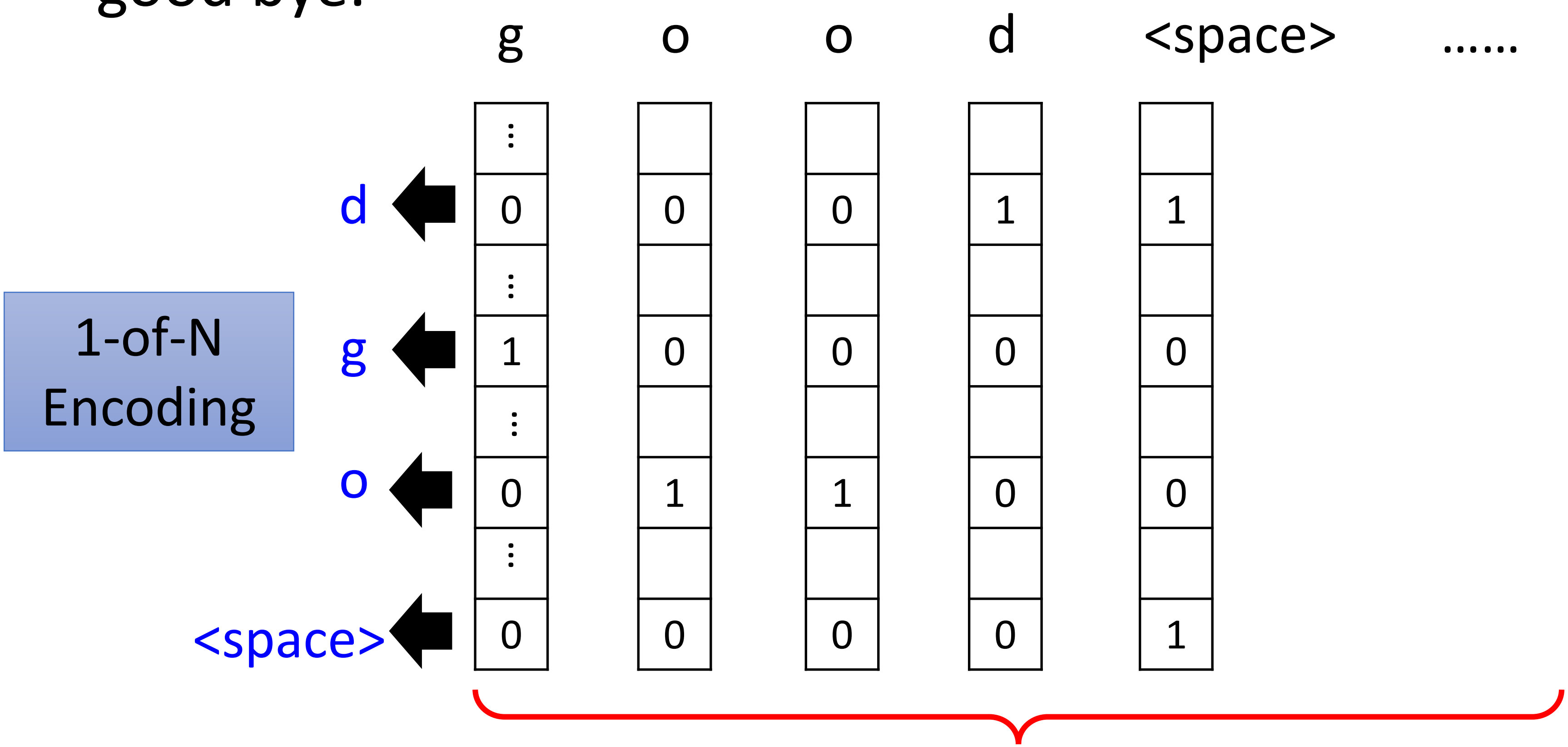

DCGAN
LSGAN
Original WGAN
Improved WGAN
G: CNN, D: CNN
G: CNN (no normalization), D: CNN (no normalization)
G: CNN (tanh), D: CNN(tanh)


DCGAN
LSGAN
Original WGAN
Improved WGAN
G: MLP, D: CNN
G: CNN (bad structure), D: CNN
G: 101 layer, D: 101 layer

I will talk about RNN later.

Sentence Generation

- good bye.



Consider this matrix as an “image”

I will talk about RNN later.

Sentence Generation

- Real sentence

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

- Generated

0.9	0.1	0.1	0	0
0.1	0.9	0.1	0	0
0	0	0.7	0.1	0
0	0	0.1	0.8	0.1
0	0	0	0.1	0.9

Can never be 1-of-N

A binary classifier can immediately find the difference.

No overlap

WGAN is helpful

Improved WGAN successfully generating sentences

https://arxiv.org/abs/1704.00028

WGAN with gradient penalty	
Busino game camperate spent odea	Solice Norkedin pring in since
In the bankaway of smarling the	ThiS record (31.) UBS) and Ch
SingersMay , who kill that imvic	It was not the annuas were plogr
Keray Pents of the same Reagun D	This will be us , the ect of DAN
Manging include a tudancs shat "	These leaded as most-worsd p2 a0
His Zuith Dudget , the Denmbern	The time I paidOa South Cubry i
In during the Uitational questio	Dour Fraps higs it was these del
Divos from The ' noth ronkies of	This year out howneed allowed lo
She like Monday , of macunsuer S	Kaulna Seto consficutes to repor
The investor used ty the present	A can teal , he was schoon news
A papees are cointry congress oo	In th 200. Pesish picriers rega
A few year inom the group that s	Konney Panice rimimber the teami
He said this syenn said they wan	The new centuct cut Denester of
As a world 1 88 ,for Autouries	The near , had been one injostie
Foand , th Word people car , Il	The incestion to week to shorted
High of the upseader homing pull	The company the high product of
The guipe is worly move dogsfor	20 - The time of accomplete , wh
The 1874 incidested he could be	John WVuderenson seqiivic spends
The allo tooks to security and c	A ceetens in indestredly the Wat

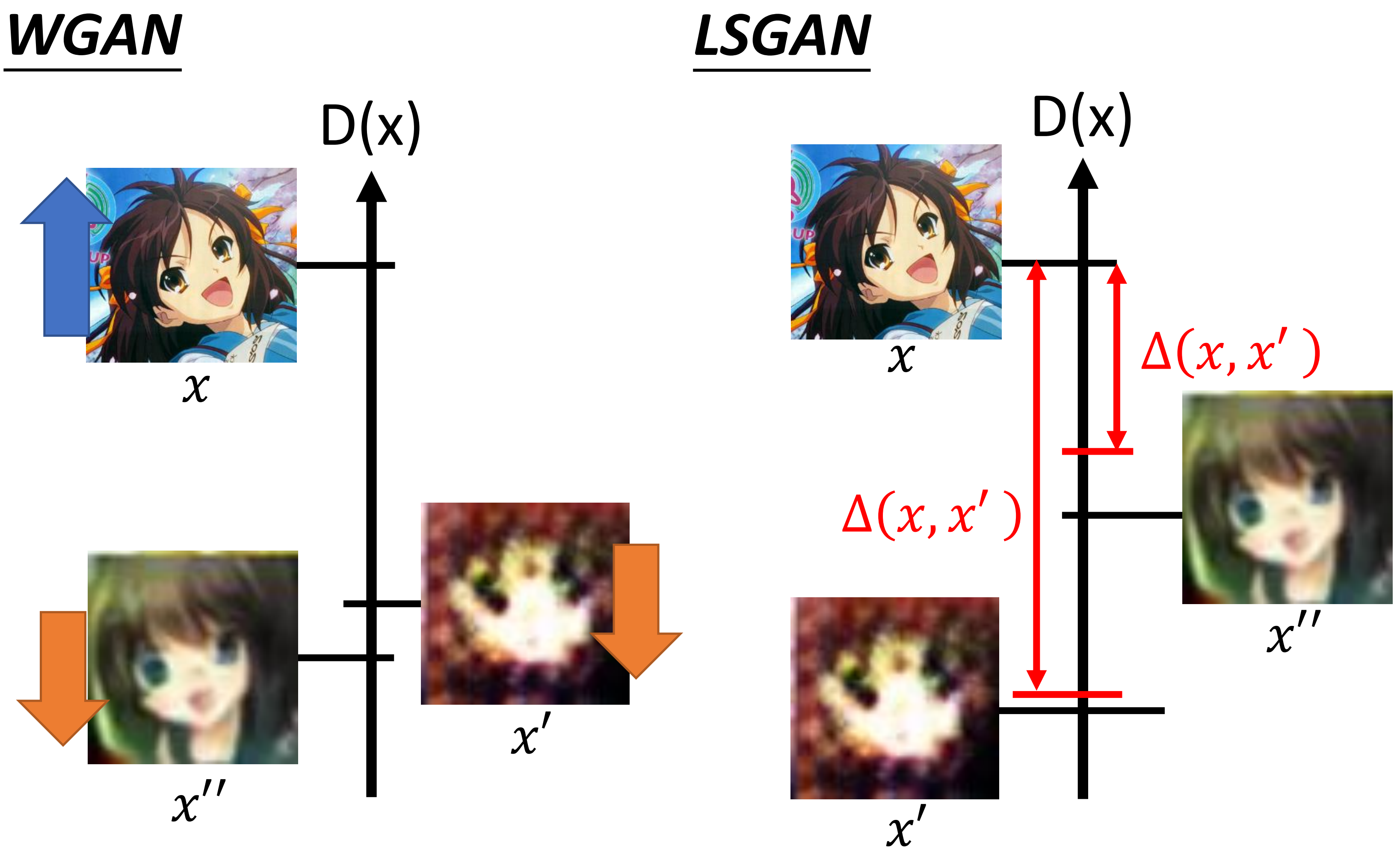
W-GAN – 唐詩鍊成

感謝 李仲翊 同學提供實驗結果

輸出 32 個字 (包含標點)

- 升雲白遲丹齋取，此酒新巷市入頭。黃道故海歸中後，不驚入得韻子門。
- 據口容章蓄翎翎，邦貸無遊隔將毬。外蕭曾臺遶出畧，此計推上呂天夢。
- 新來寶伎泉，手雪泓臺蓑。曾子花路魏，不謀散薦船。
- 功持牧度機邀爭，不躚官嬉牧涼散。不迎白旅今掩冬，盡蘸金衹可停。
- 玉十洪沄爭春風，溪子風佛挺橫鞋。盤盤稅焰先花齋，誰過飄鶴一丞幢。
- 海人依野庇，為阻例沉迴。座花不佐樹，弟闌十名儂。
- 入維當興日世瀕，不評皺。頭醉空其杯，駸園凋送頭。
- 鉢笙動春枝，寶絫潔長知。官爲窑爛去，絆粒薛一靜。
- 吾涼腕不楚，縱先待旅知。楚人縱酒待，一蔓飄聖猜。
- 折幕故癘應韻子，徑頭霜瓊老徑徑。尚錯春鏘熊悽梅，去吹依能九將香。
- 通可矯目鷚須淨，丹迤挈花一抵嫖。外子當目中前醒，迎日幽筆鈎弧前。
- 庭愛四樹人庭好，無衣服仍繡秋州。更怯風流欲鳰雲，帛陽舊據畞婷儻。

Loss-sensitive GAN (LSGAN)

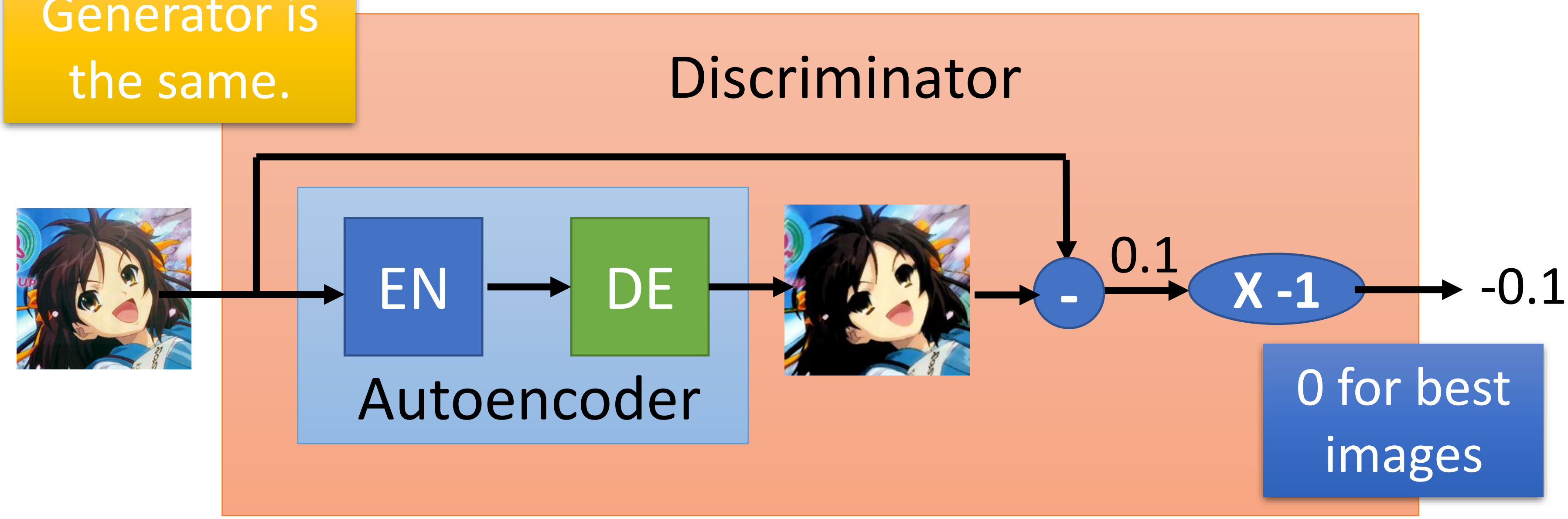


Energy-based GAN (EBGAN)

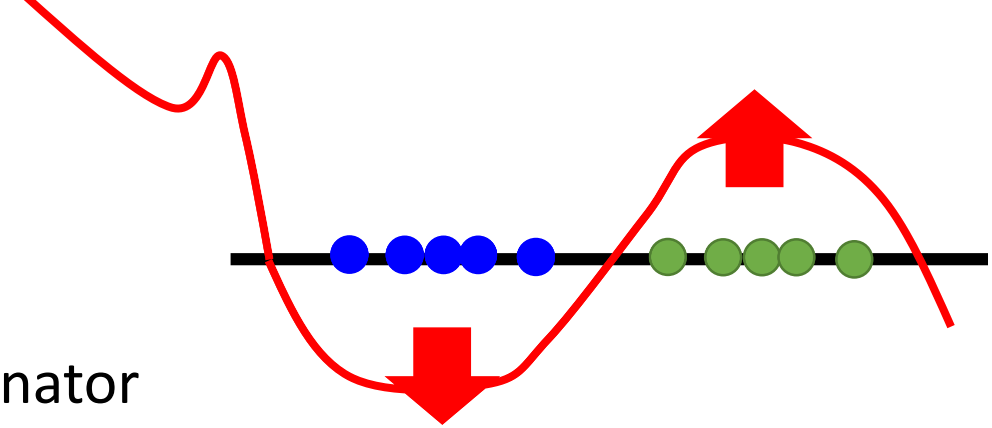
- Using an autoencoder as discriminator D

An image is good. = It can be reconstructed by autoencoder.

Generator is the same.

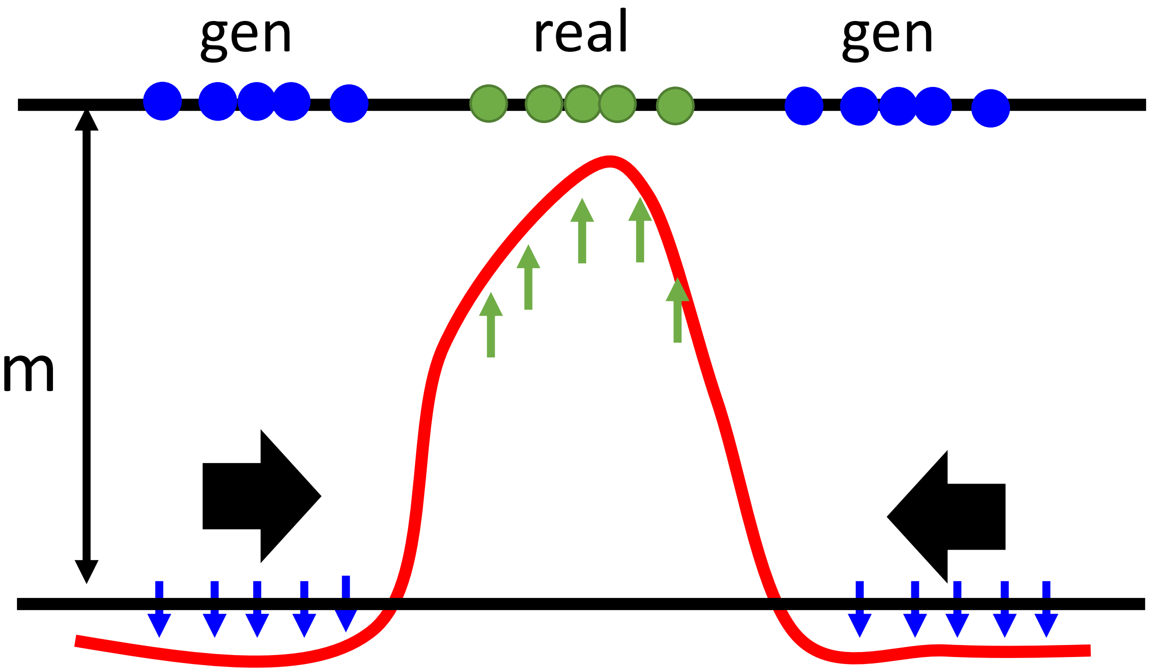


EBGAN



Auto-encoder based discriminator only give limited region large value.

0 is for the best.



Do not have to be very negative

Hard to reconstruct, easy to destroy

Mode Collapse

Missing Mode ?

Mode collapse is easy to detect.

?

Missing Mode ?

- E.g. BEGAN on CelebA

陳柏文同學提供
實驗結果

圖片來原 柯達方 同學

Ensemble

Generator 1

Generator 2

Lecture II:
Variants of GAN

Lecture II

Conditional Generation

Sequence Generation

A Little Bit of Theory (option)

Conditional Generation

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



Conditional Generation

"Girl with red hair and red eyes"

"Girl with yellow ribbon"

$\rightarrow \boxed{\text{NN Generator}} \rightarrow$

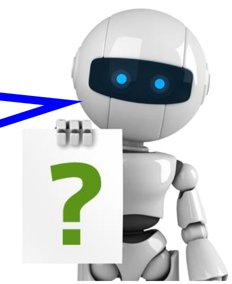

Conditional Generation

- We don’t want to simply generate some random stuff.
- Generate objects based on conditions:

Caption Generation

Given condition:

“A young girl is dancing.”

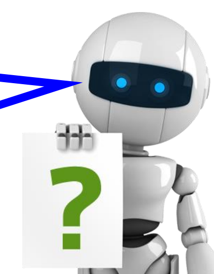


Chat-bot

Given condition:

“Hello”

“Hello. Nice to see you.”



Conditional Generation

Modifying input code

- Making code has influence (InfoGAN)
- Connection code space with attribute

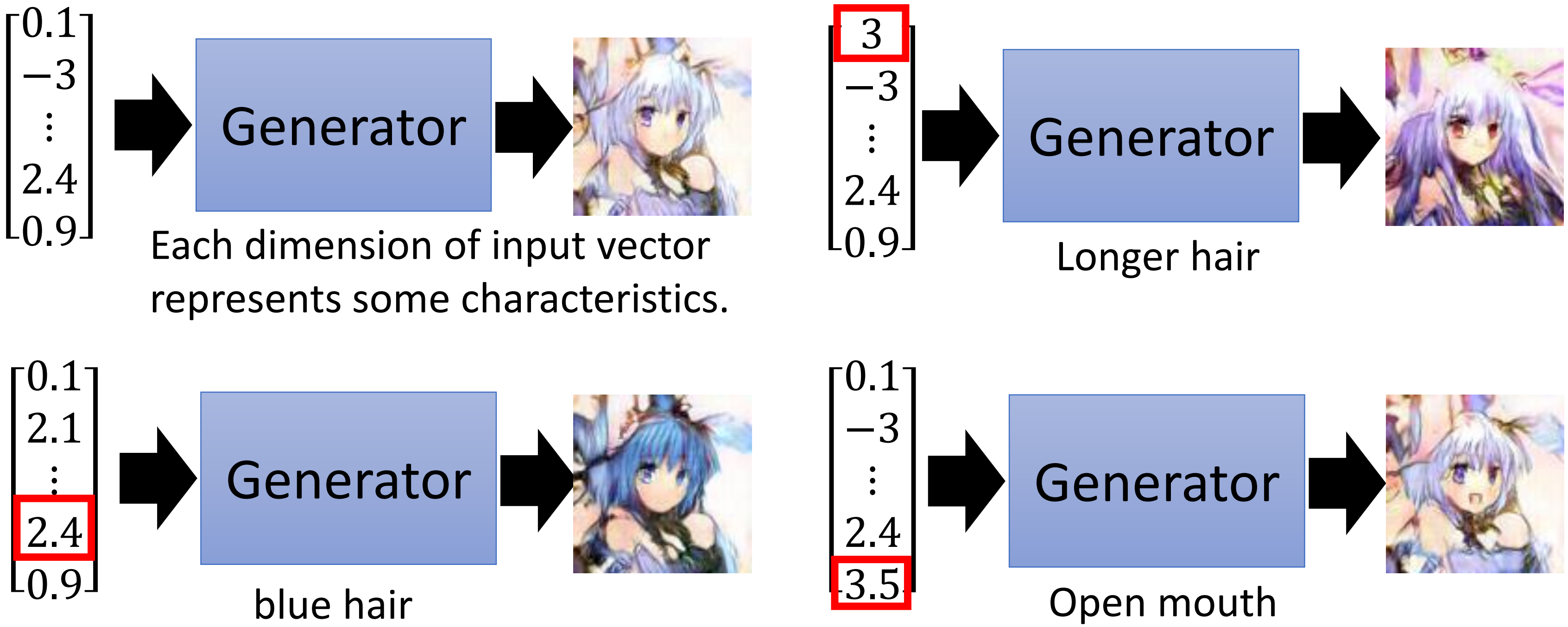
Controlling by input objects

- Paired data
- Unpaired data
- Unsupervised

Feature extraction

- Domain Independent Feature
- Improving Auto-encoder (VAE-GAN, BiGAN)

Modifying Input Code



- The input code determines the generator output.
- Understand the meaning of each dimension to control the output.

Conditional Generation

Modifying input code

- Making code has influence (InfoGAN)
- Connection code space with attribute

Controlling by input objects

- Paired data
- Unpaired data
- Unsupervised

Feature extraction

- Domain Independent Feature
- Improving Auto-encoder (VAE-GAN, BiGAN)

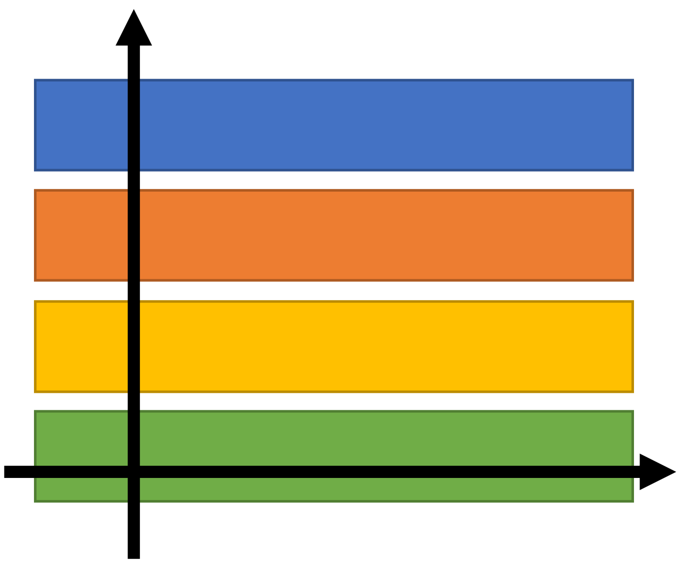
InfoGAN

(The colors represents the characteristics.)

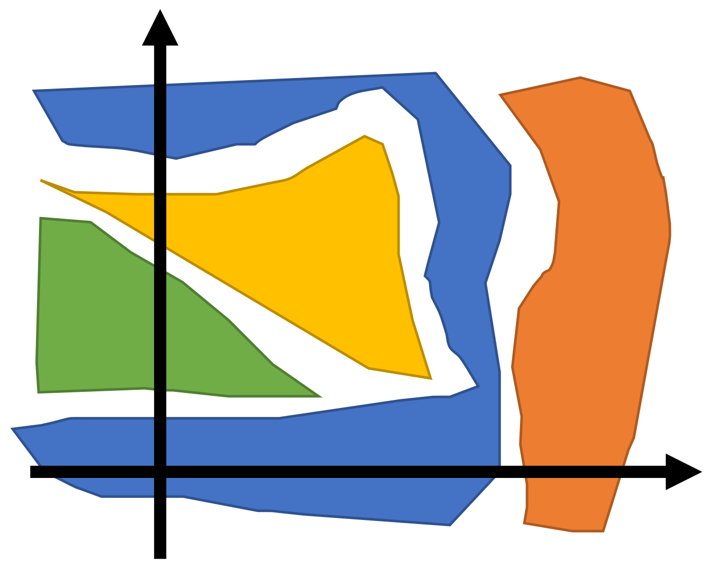
Regular GAN

Modifying a specific dimension, no clear meaning

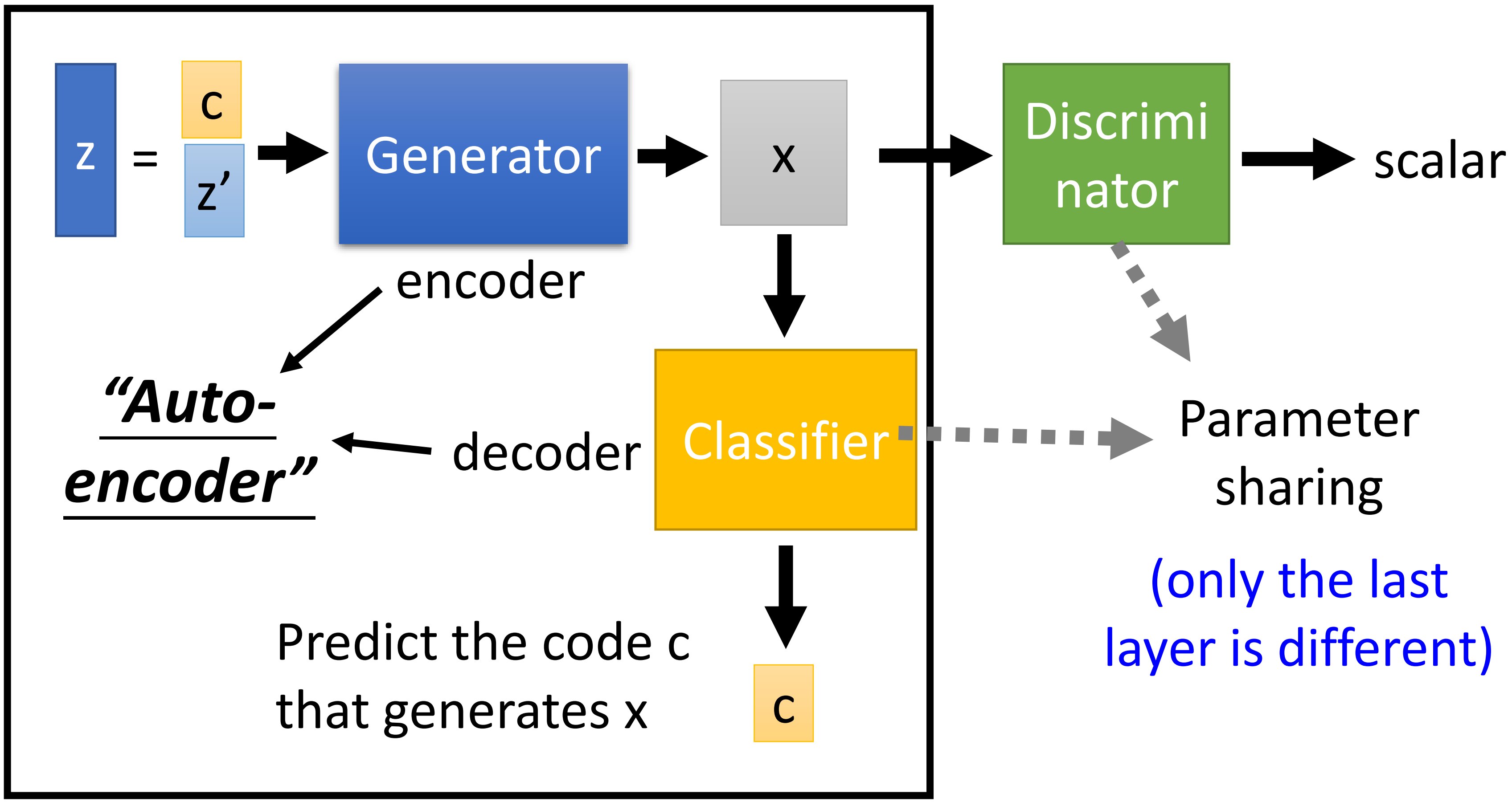
What we expect



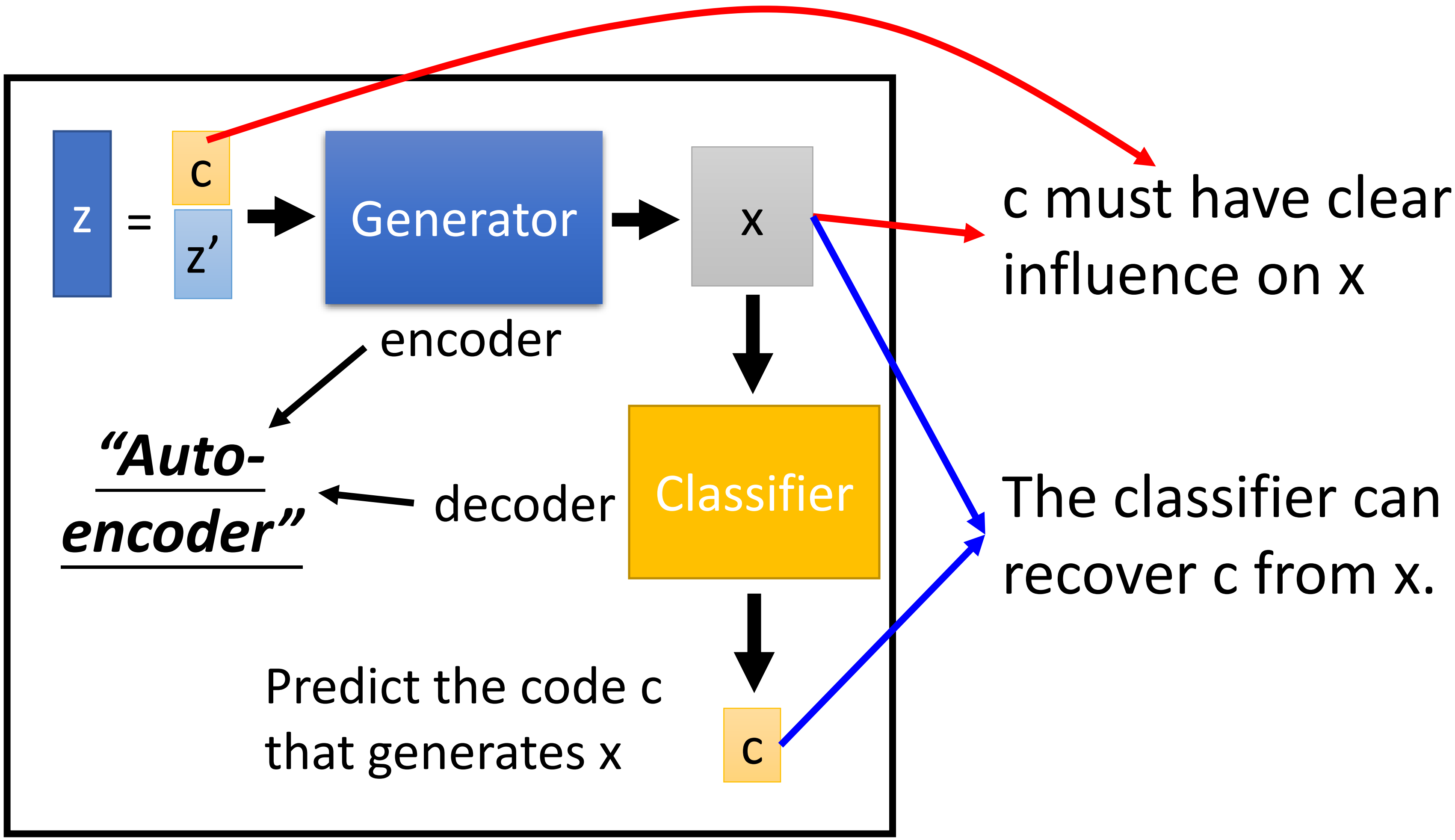
Actually …

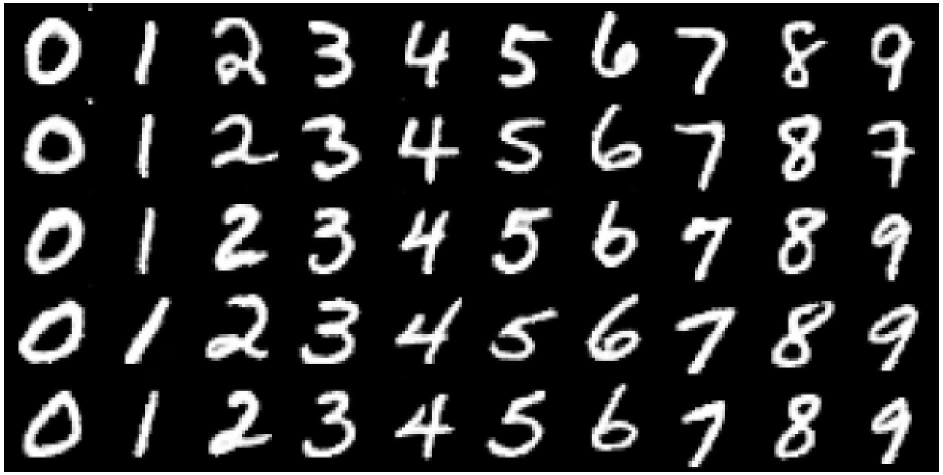


What is InfoGAN?

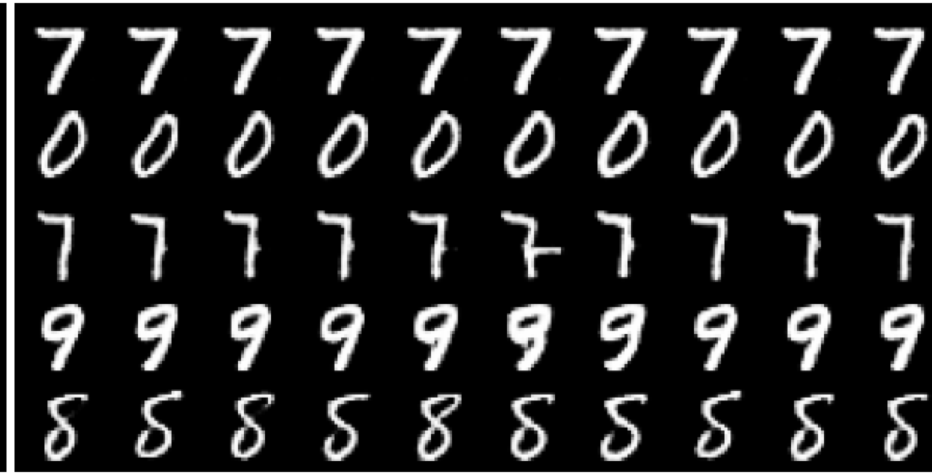


What is InfoGAN?

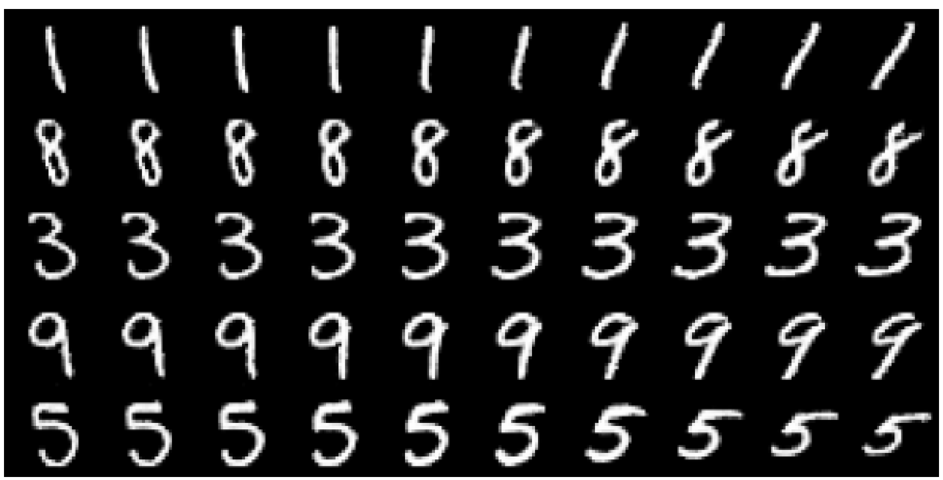




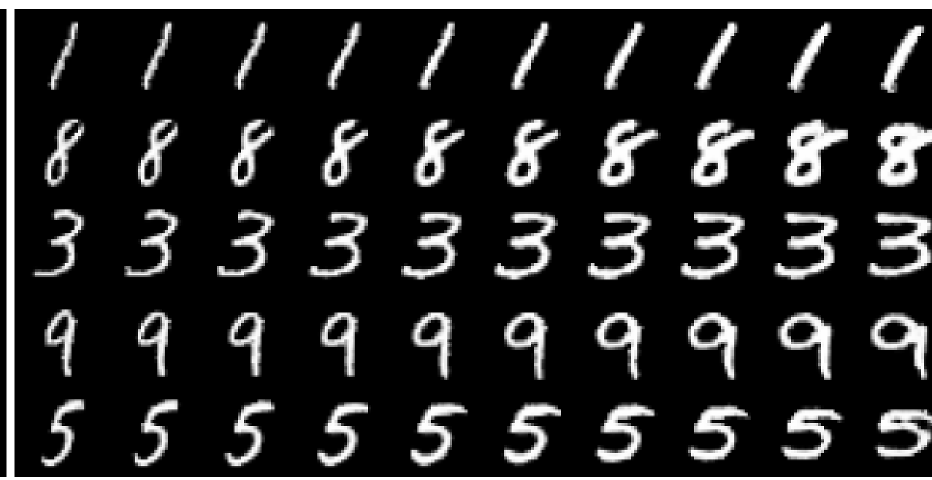
(a) Varying c_1 on InfoGAN (Digit type)



(b) Varying c_1 on regular GAN (No clear meaning)



(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation)



(d) Varying c_3 from -2 to 2 on InfoGAN (Width)

https://arxiv.org/abs/1606.03657

(a) Rotation (b) Width

(c) Lighting (d) Wide or Narrow

https://arxiv.org/abs/1606.03657

Conditional Generation

Modifying input code

- Making code has influence (InfoGAN)
- Connection code space with attribute

Controlling by input objects

- Paired data
- Unpaired data
- Unsupervised

Feature extraction

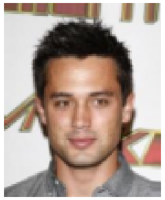
- Domain Independent Feature
- Improving Auto-encoder (VAE-GAN, BiGAN)

Connecting Code and Attribute

(c) Hair style

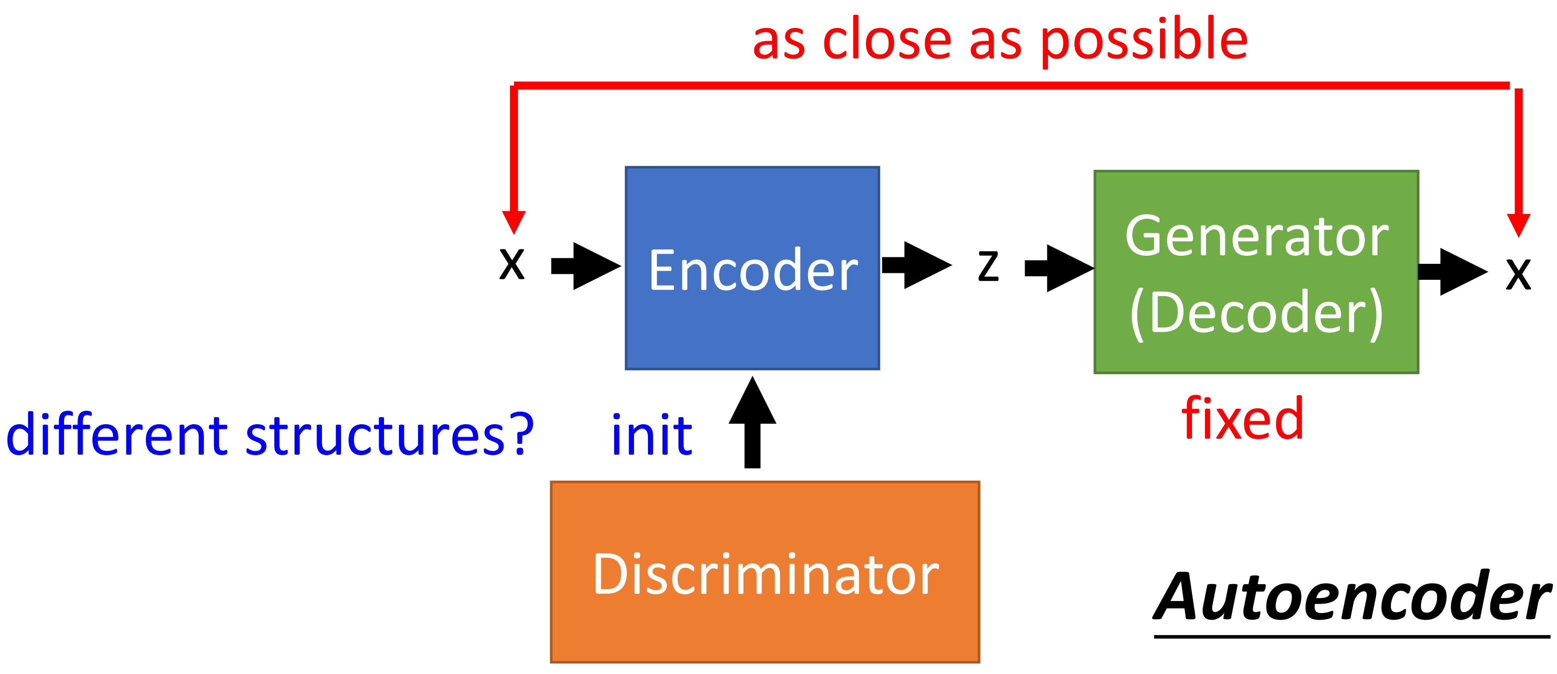
(d) Emotion

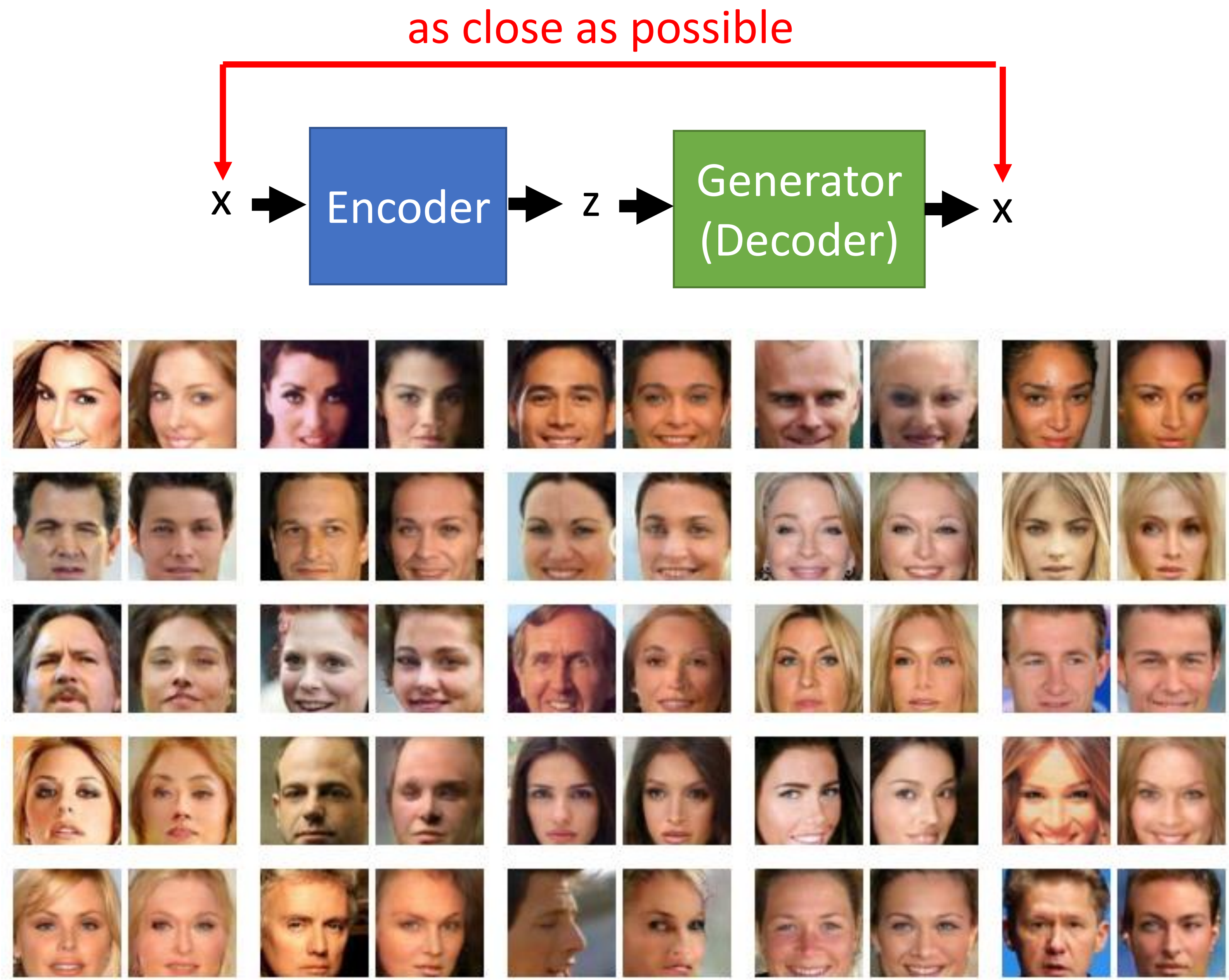
CelebA

Image	Attributes
	Arched eyebrows, attractive, brown hair, heavy makeup, high cheekbones, mouth slightly open, no beard, pointy nose, smiling, straight hair, wearing earrings, wearing lipstick, young.
	5 o'clock shadows, attractive, bags under eyes, big lips, big nose, black hair, bushy eyebrows, male, no beard, pointy nose, straight hair, young.

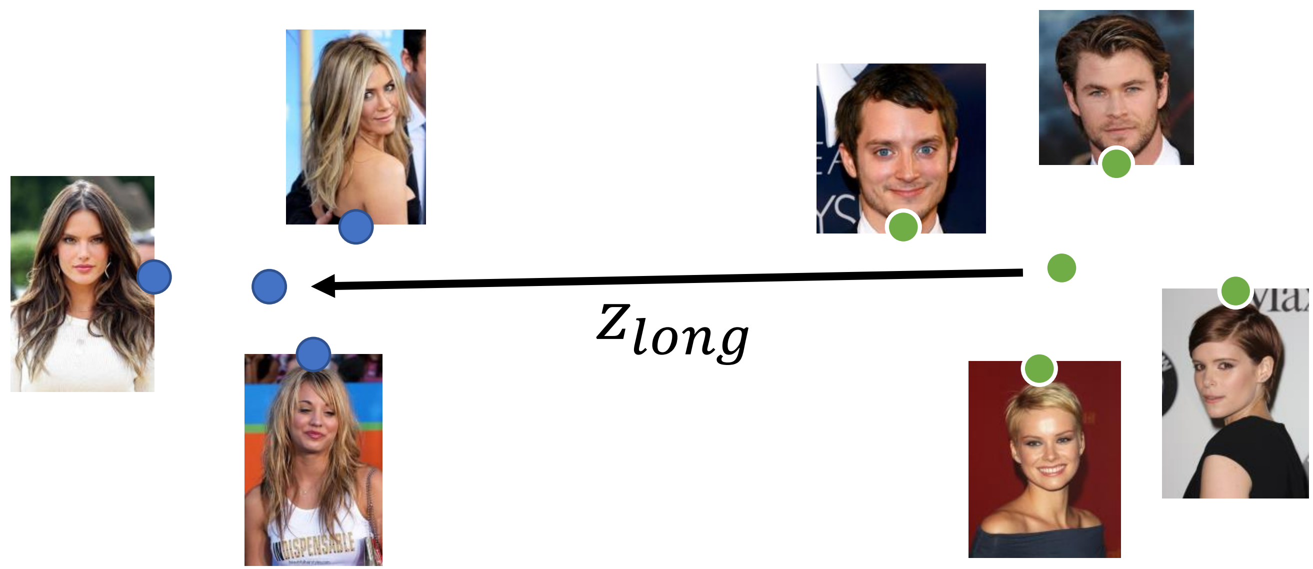
GAN+Autoencoder

- We have a generator (input z, output x)
- However, given x, how can we find z?
    - Learn an encoder (input x, output z)




as close as possible
x
Encoder
z
Generator
(Decoder)
x

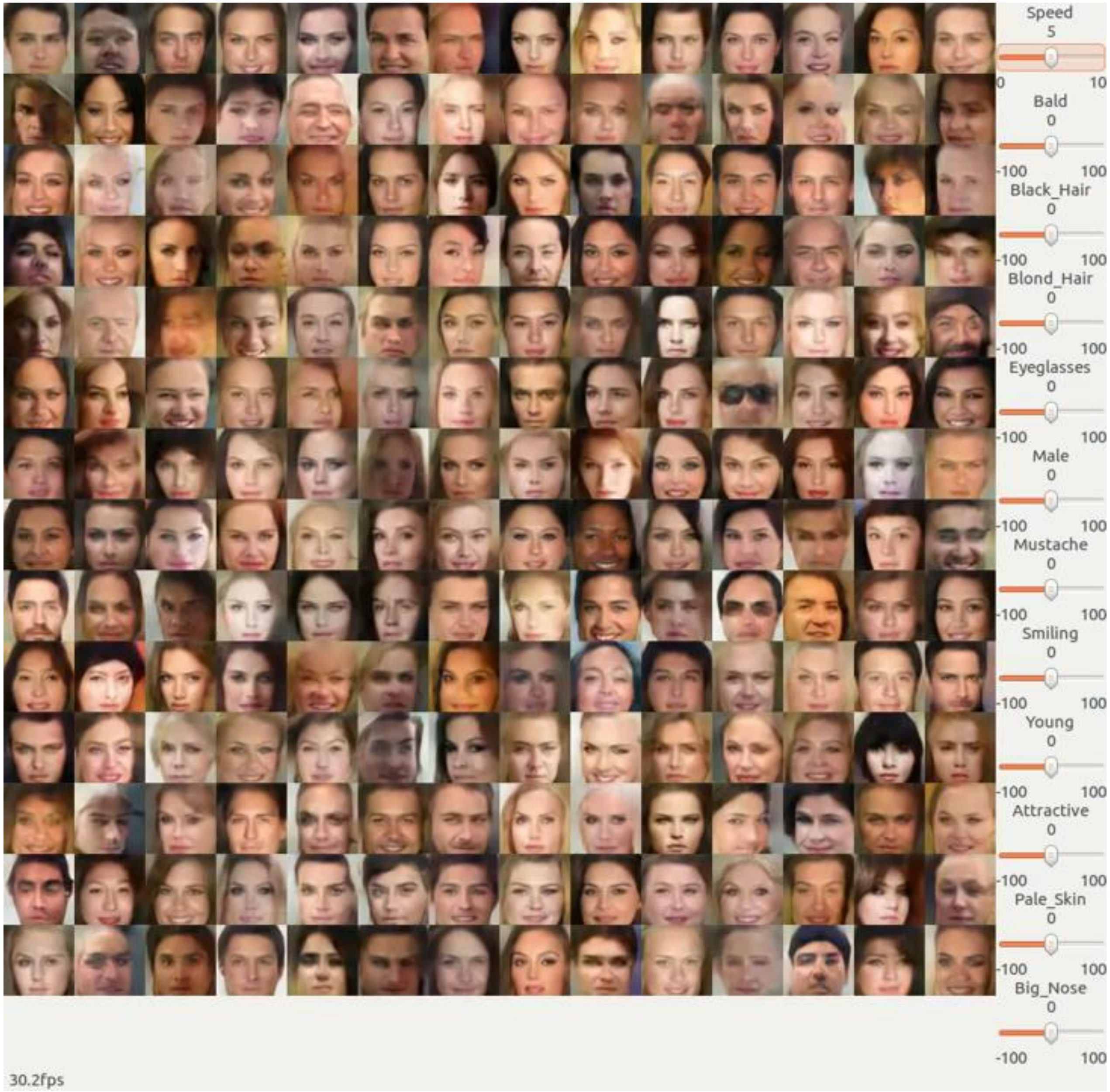
Attribute Representation



$$z_{long} = \frac{1}{N_1} \sum_{x \in long} En(x) - \frac{1}{N_2} \sum_{x' \notin long} En(x')$$

Short Hair $\quad x \Rightarrow En(x) + z_{long} = z' \Rightarrow Gen(z')$ $\quad$ Long Hair

Photo Editing



https://www.youtube.com/watch?v=kPEIJJsQr7U

Conditional Generation

Modifying input code

- Making code has influence (InfoGAN)
- Connection code space with attribute

Controlling by input objects

- Paired data
- Unpaired data
- Unsupervised

Feature extraction

- Domain Independent Feature
- Improving Auto-encoder (VAE-GAN, BiGAN)

Conditional Generation

Modifying input code

- Making code has influence (InfoGAN)
- Connection code space with attribute

Controlling by input objects

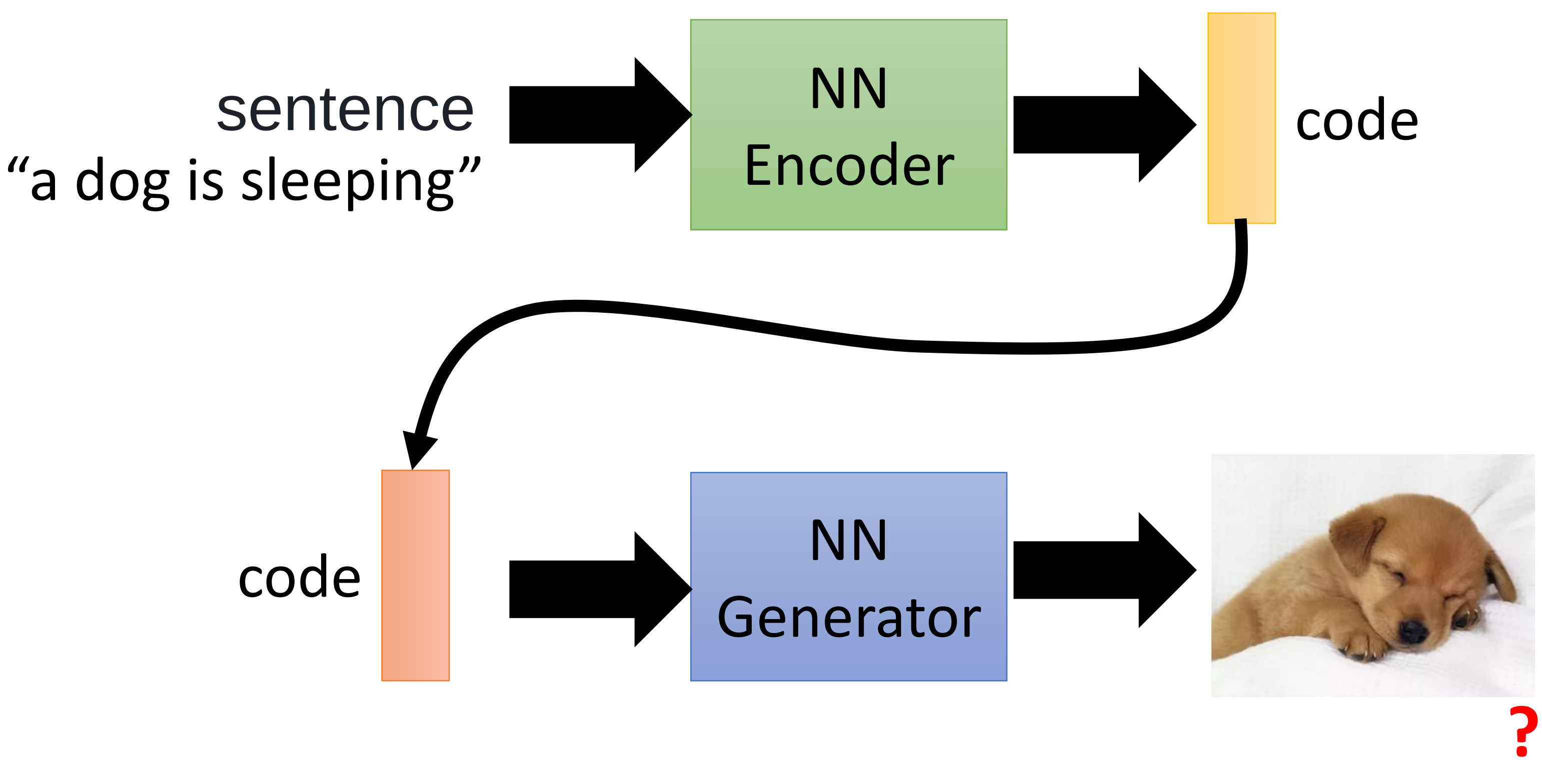
- $\boxed{\text{Paired data}}$
- Unpaired data
- Unsupervised

Feature extraction

- Domain Independent Feature
- Improving Auto-encoder (VAE-GAN, BiGAN)

Conditional GAN

- Generating images based on text description

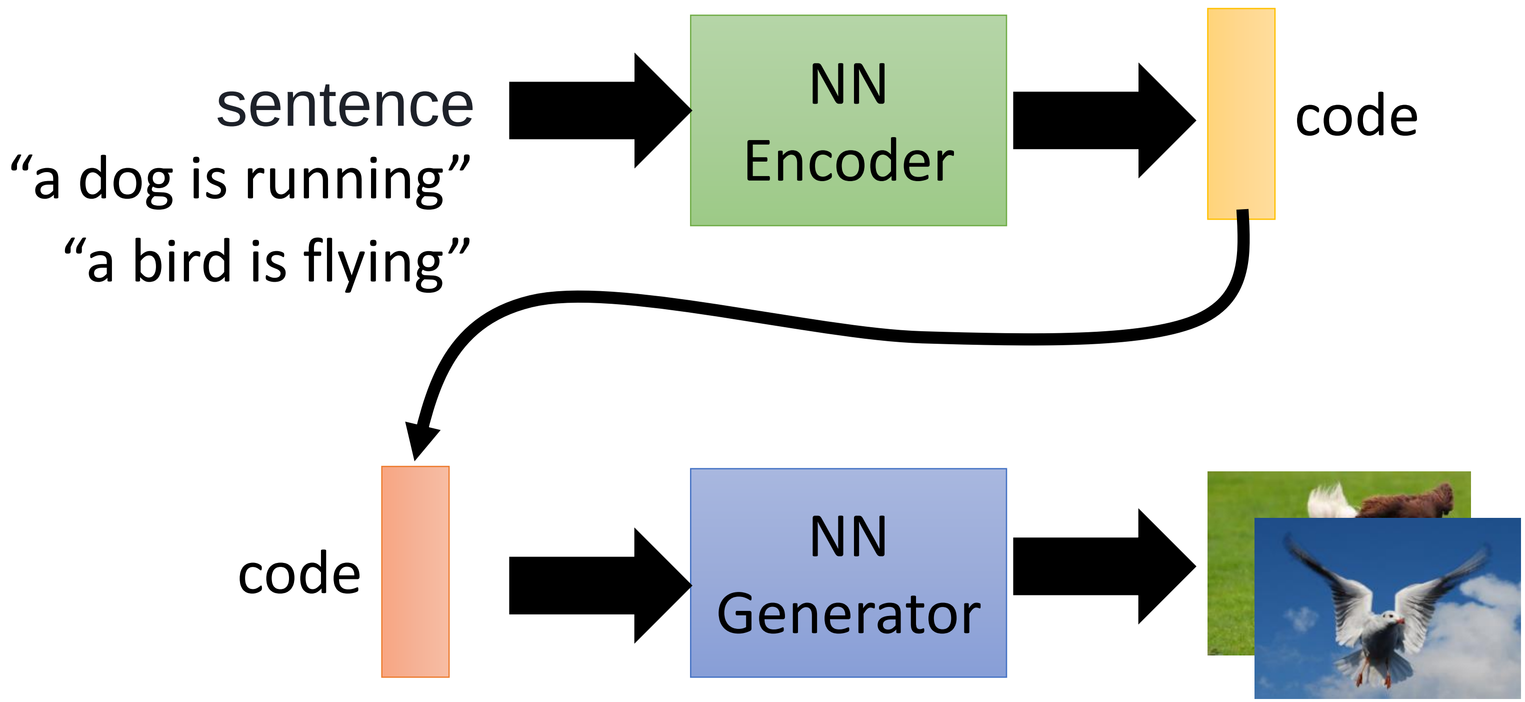


Conditional GAN

c^1: a dog is running $\hat{x}^1$:

c^2: a bird is flying $\hat{x}^2$:

- Generating images based on text description



Conditional GAN

c^1: a dog is running $\hat{x}^1$: 

c^2: a bird is flying $\hat{x}^2$: 

- ***Text to image*** by traditional supervised learning

c^1: a dog is running → NN → Image ⟷ $\hat{x}^1$: 

as close as possible

Text: "train"

Target of NN output

A blurry image!

Conditional GAN

c: train → G → Image x = G(c,z)

Prior distribution z → G

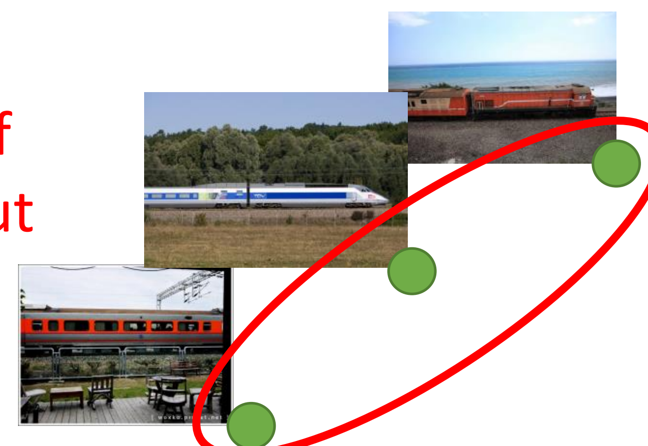
It is a distribution.

Approximate the distribution below

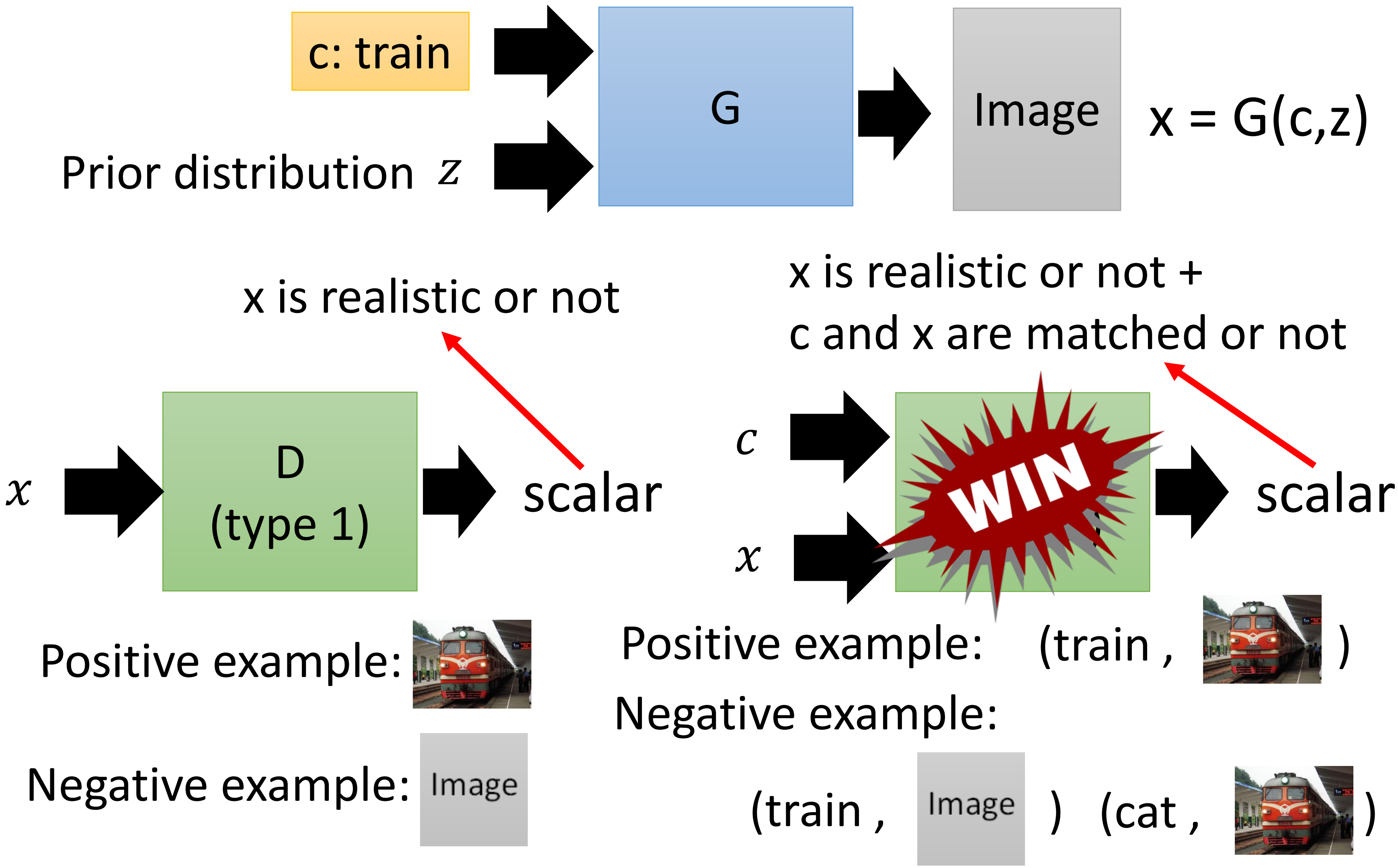
Text: "train"

Target of NN output

A blurry image!



Conditional GAN



Text to Image - Results

"red flower with black center"

Caption	Image
this flower has white petals and a yellow stamen	
the center is yellow surrounded by wavy dark purple petals	
this flower has lots of small round pink petals	

Text to Image
- Results

Caption	Image
a pitcher is about to throw the ball to the batter	
a group of people on skis stand in the snow	
a man in a wet suit riding a surfboard on a wave	

Conditional GAN

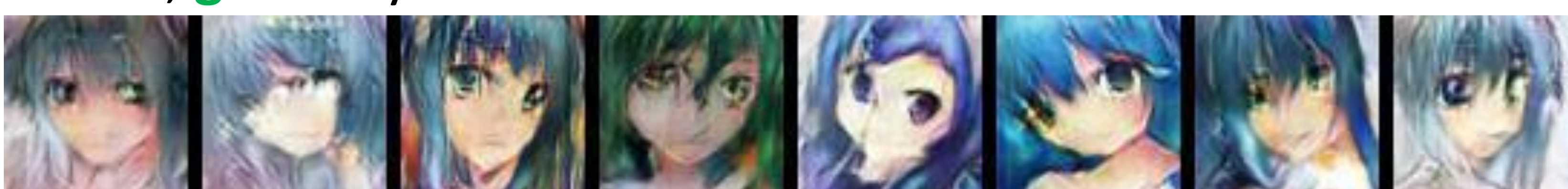
MLDS 作業三
負責助教：曾柏翔

- 根據文字敘述畫出動漫人物頭像

Red hair, long hair

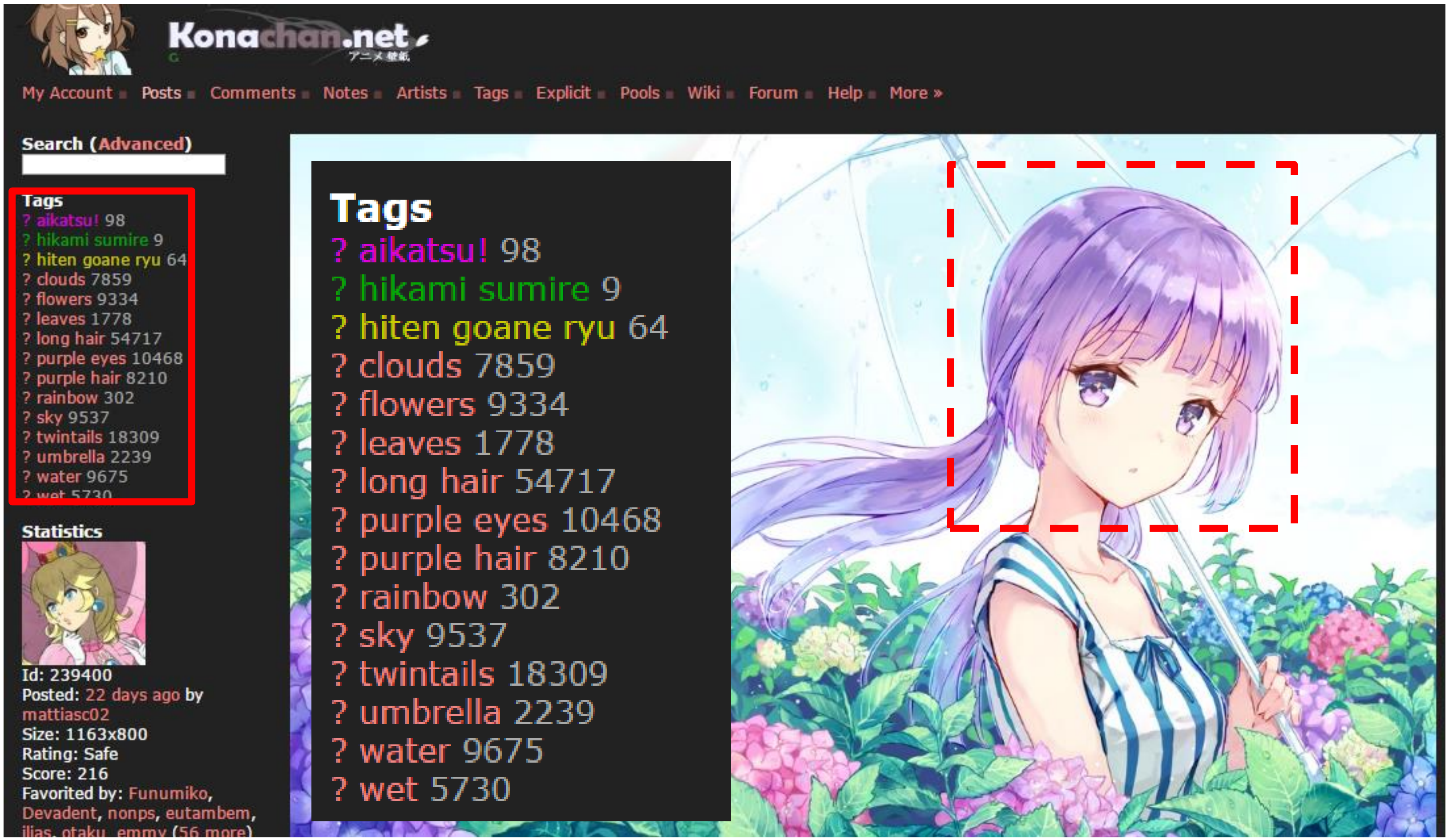
Black hair, **blue** eyes

Blue hair, **green** eyes



Data Collection

感謝曾柏翔助教、
樊恩宇助教蒐集資料



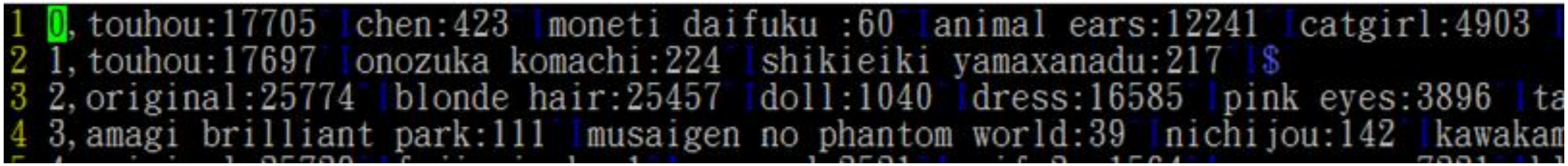
http://konachan.net/post/show/239400/aikatsu-clouds-flowers-hikami_sumire-hiten_goane_r

Released Training Data

blue eyes
red hair
short hair

96 x 96

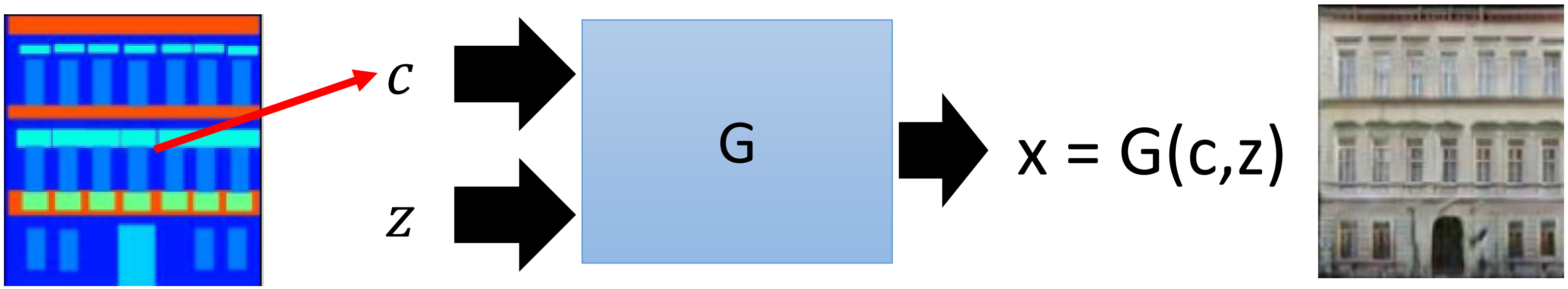
- Data download link: https://drive.google.com/open?id=0BwJmB7alR-AvMHEtczZZN0EtdzQ
- Anime Dataset:
  - training data: 33.4k (image, tags) pair
- Training tags file format
  - img_id <comma> tag1 <colon> #_post <tab> tag2 <colon> …

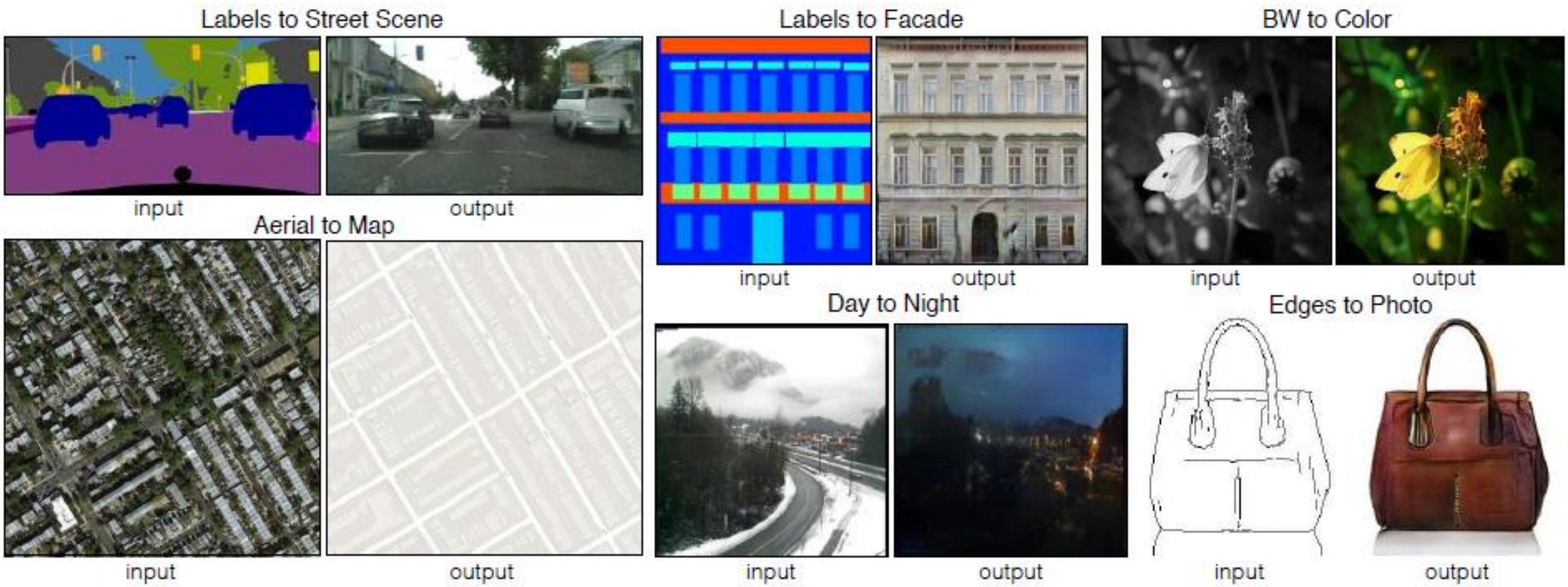


```
0, touhou:17705 	chen:423 	moneti daifuku :60 	animal ears:12241 	catgirl:4903
1, touhou:17697 	onozuka komachi:224 	shikieiki yamaxanadu:217 	$
2, original:25774 	blonde hair:25457 	doll:1040 	dress:16585 	pink eyes:3896 	ta
3, amagi brilliant park:111 	musaigen no phantom world:39 	nichijou:142 	kawakam
```

tags.csv

Image-to-image

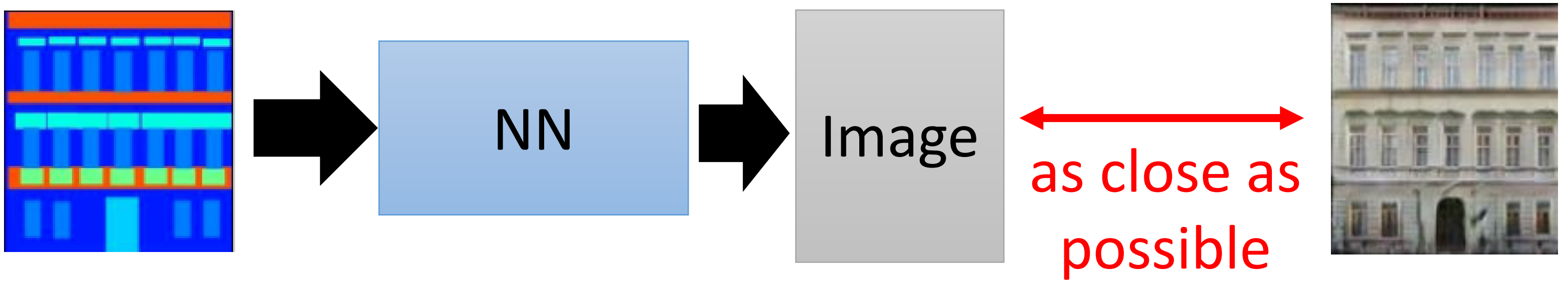




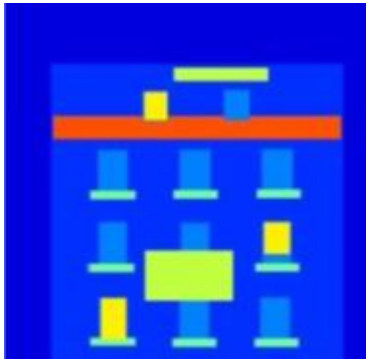
https://arxiv.org/pdf/1611.07004

Image-to-image

- Traditional supervised approach



Testing:

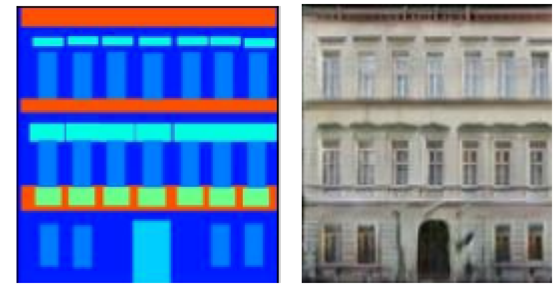


input

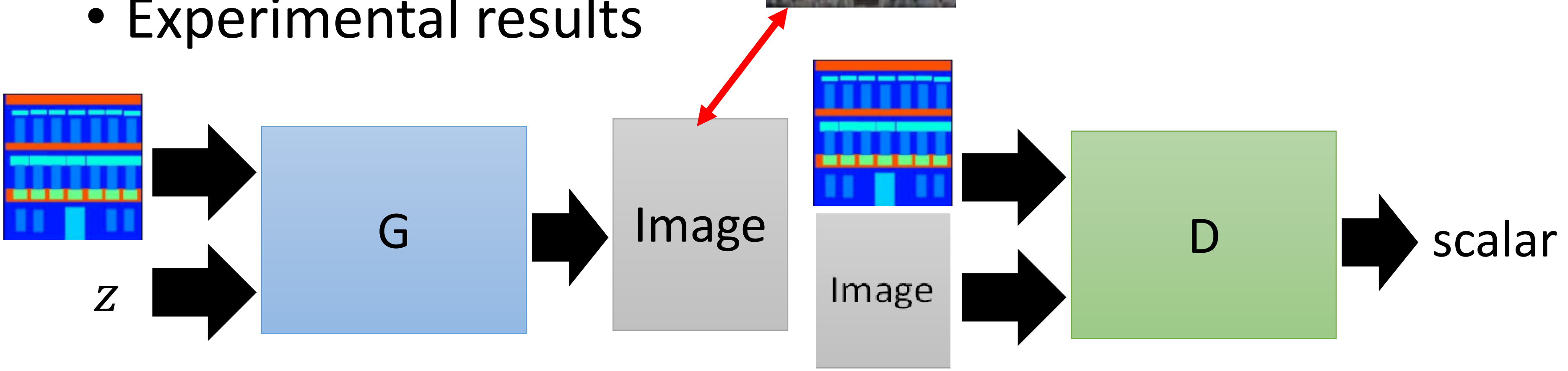
close

It is blurry because it is the average of several images.

Image-to-image



- Experimental results



Testing:

Image super resolution

- Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network”, CVPR, 2016

bicubic (21.59dB/0.6423)	SRResNet (23.53dB/0.7832)	SRGAN (21.15dB/0.6868)	original

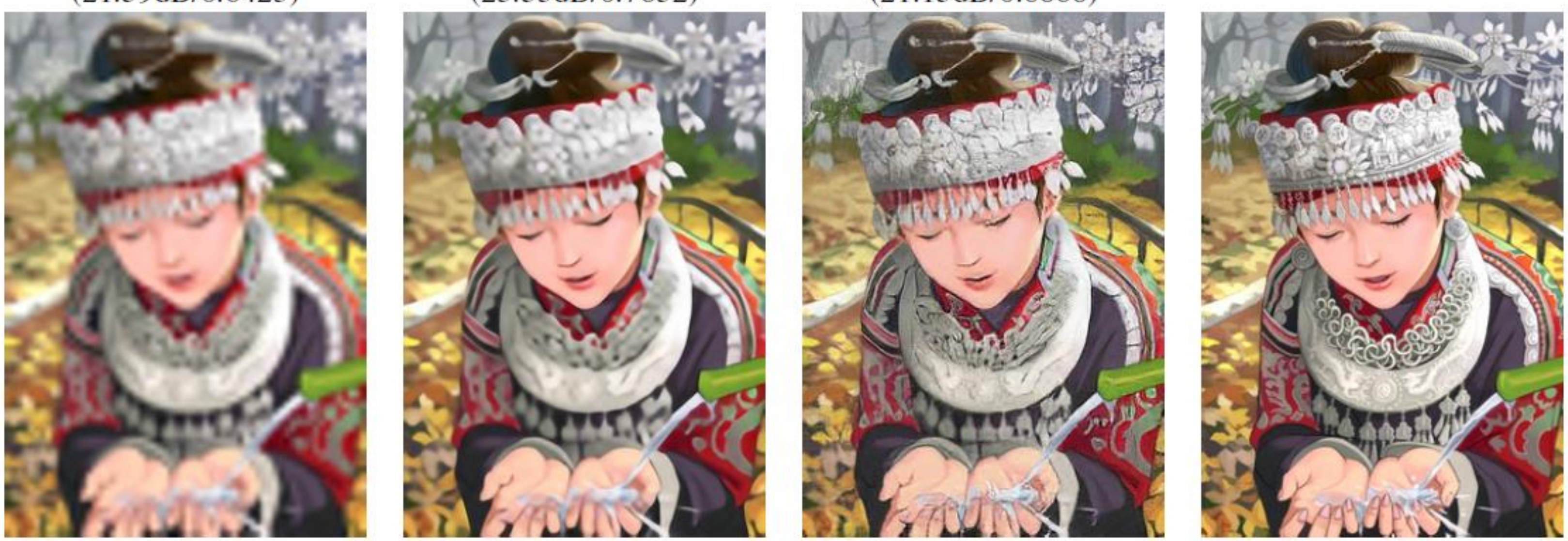
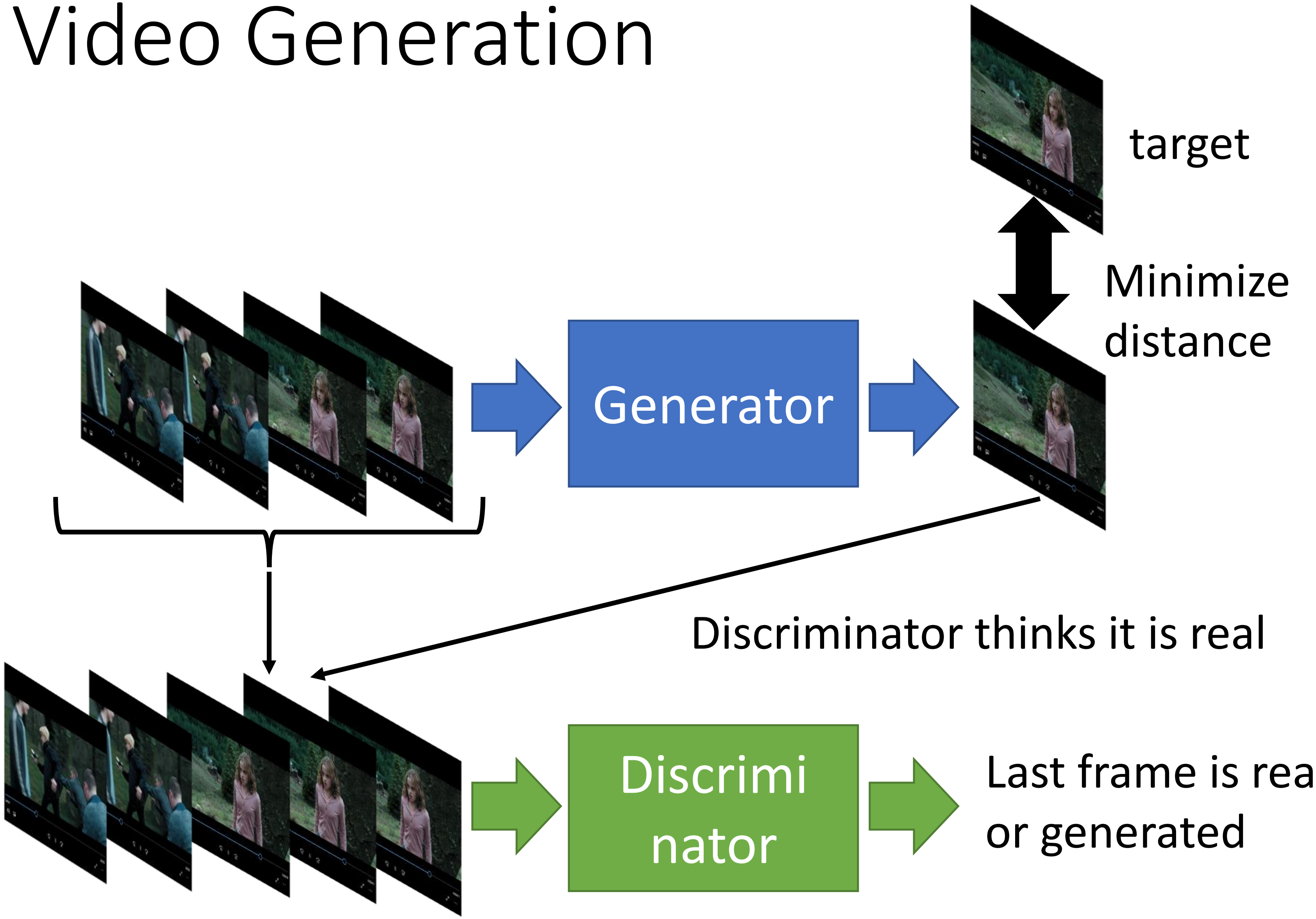
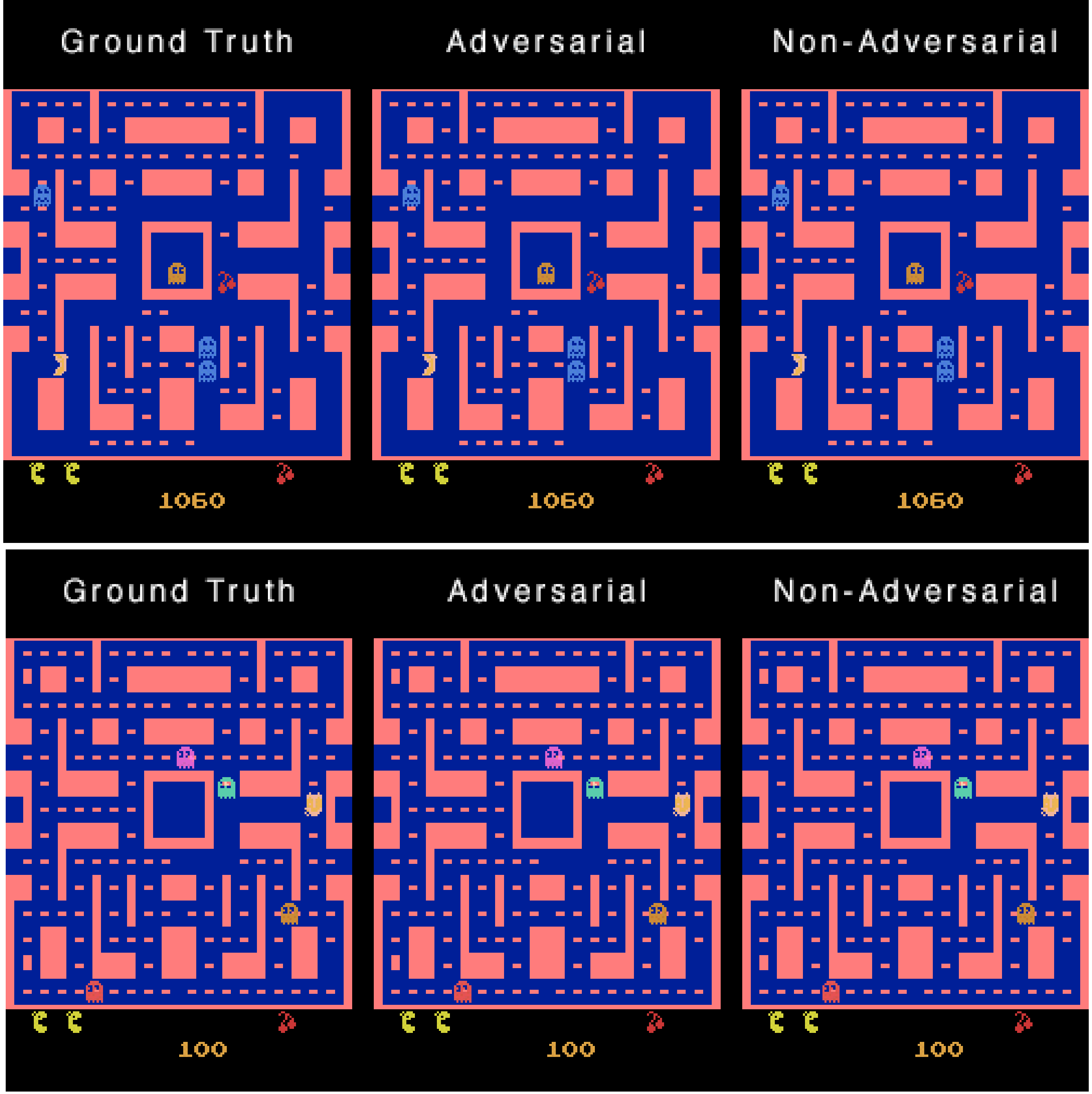


Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]

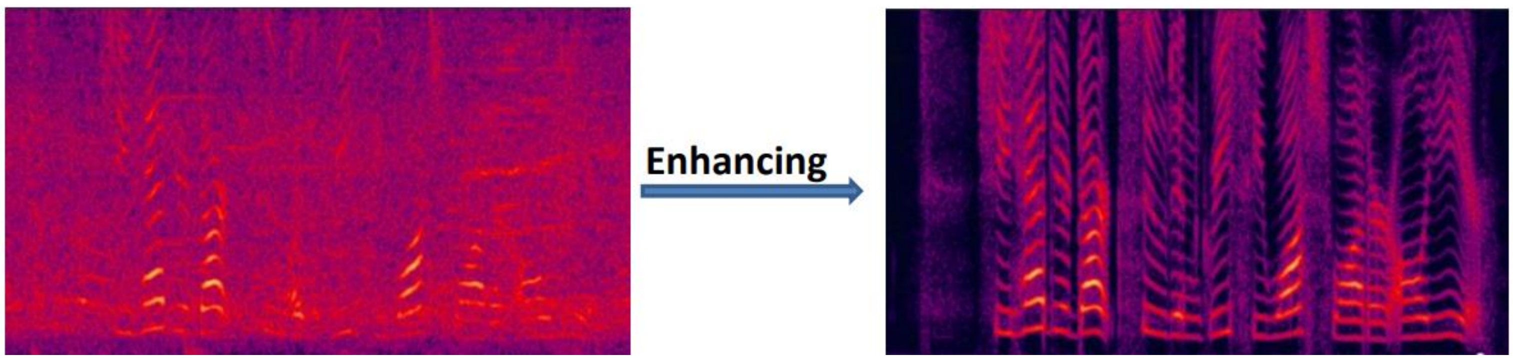
Video Generation



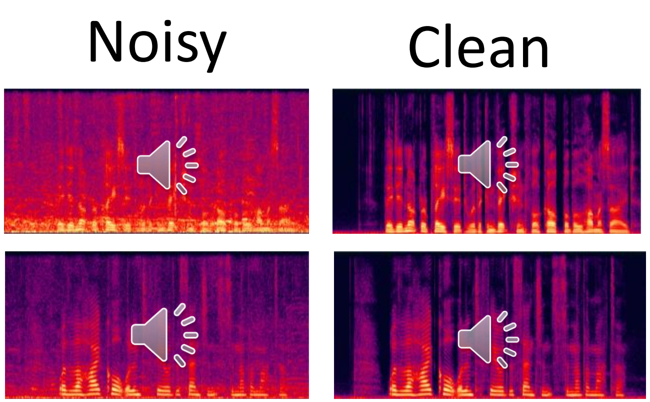


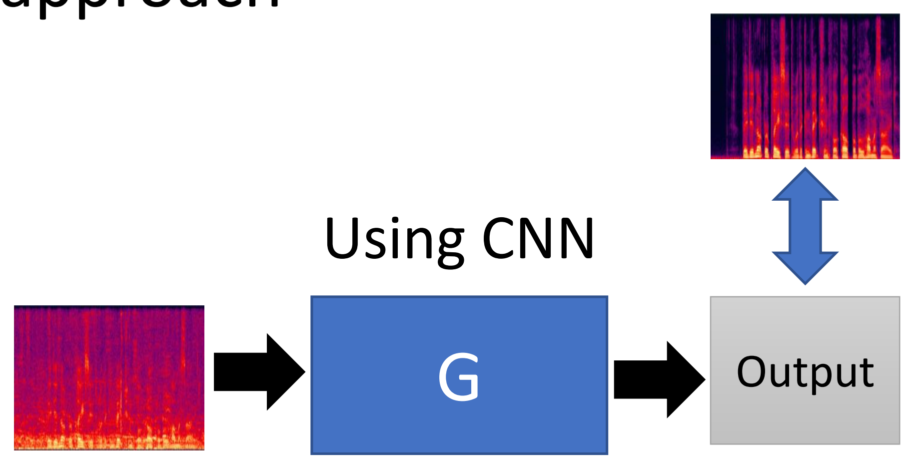
https://github.com/dyelax/Adversarial_Video_Generation

Speech Enhancement



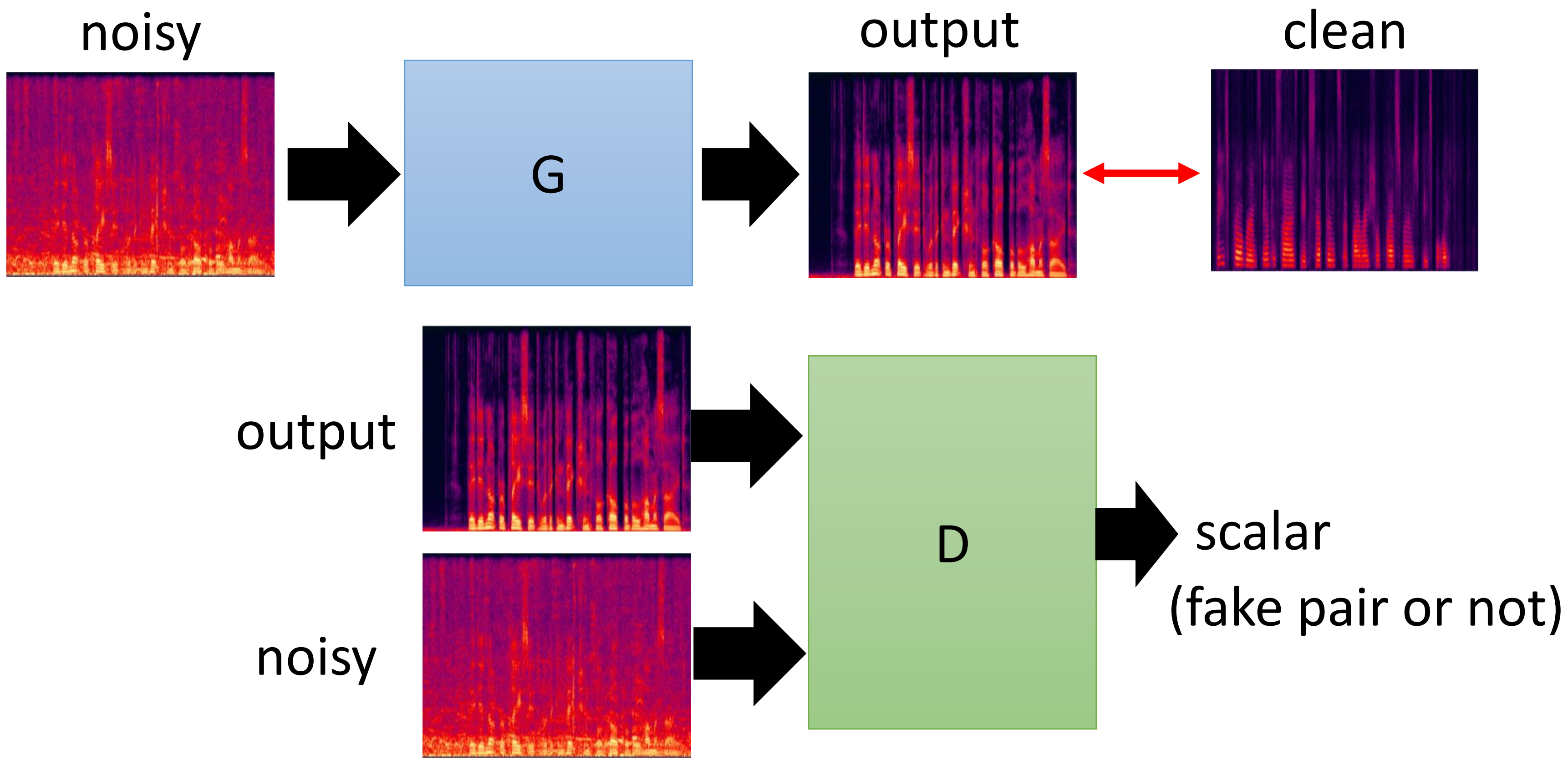
- Typical deep learning approach





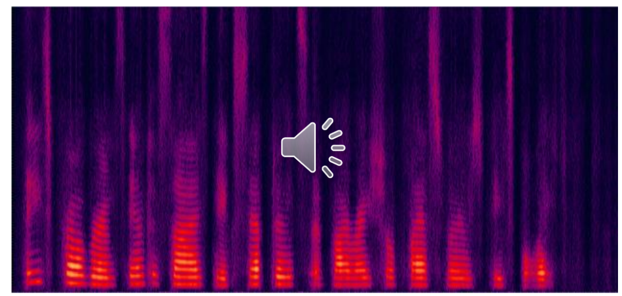
Speech Enhancement

- Conditional GAN

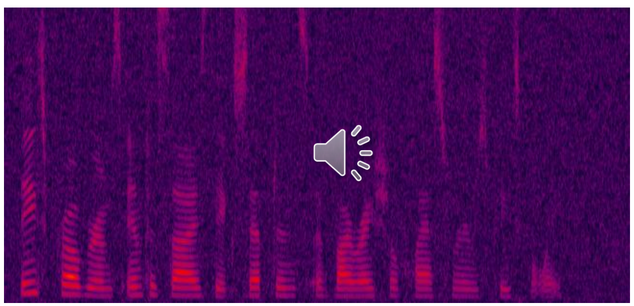


Speech Enhancement

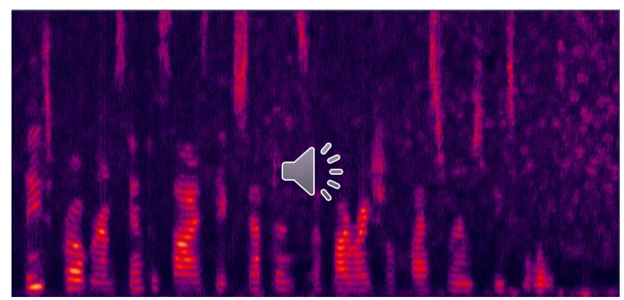
Enhanced Speech



Noisy Speech



感謝廖峴峰同學提供實驗結果
(和中研院曹昱博士共同指導)



Which Enhanced Speech is better?

More about Speech Processing

- Speech synthesis
  - Takuhiro Kaneko, Hirokazu Kameoka, Nobukatsu Hojo, Yusuke Ijima, Kaoru Hiramatsu, Kunio Kashino, “Generative Adversarial Network-based Postfiltering for Statistical Parametric Speech Synthesis”, ICASSP 2017
  - Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari, "Training algorithm to deceive anti-spoofing verification for DNN-based speech synthesis, ”, ICASSP 2017
- Voice Conversion
  - Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, Hsin-Min Wang, Voice Conversion from Unaligned Corpora using Variational Autoencoding Wasserstein Generative Adversarial Networks, Interspeech 2017
- Speech Enhancement
  - Santiago Pascual, Antonio Bonafonte, Joan Serrà, SEGAN: Speech Enhancement Generative Adversarial Network, Interspeech 2017

Conditional Generation

Modifying input code

- Making code has influence (InfoGAN)
- Connection code space with attribute

Controlling by input objects

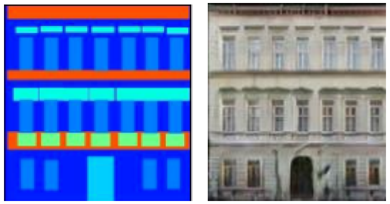
- Paired data
- Unpaired data
- Unsupervised

Feature extraction

- Domain Independent Feature
- Improving Auto-encoder (VAE-GAN, BiGAN)

Cycle GAN, Disco GAN

paired data



Transform an object from one domain to another ***without paired data***

photo van Gogh

Domain X

Domain Y

Cycle GAN

https://arxiv.org/abs/1703.10593
https://junyanz.github.io/CycleGAN/

Domain X

Domain Y

Domain X

ignore input

$G_{X \rightarrow Y}$

Become similar
to domain Y

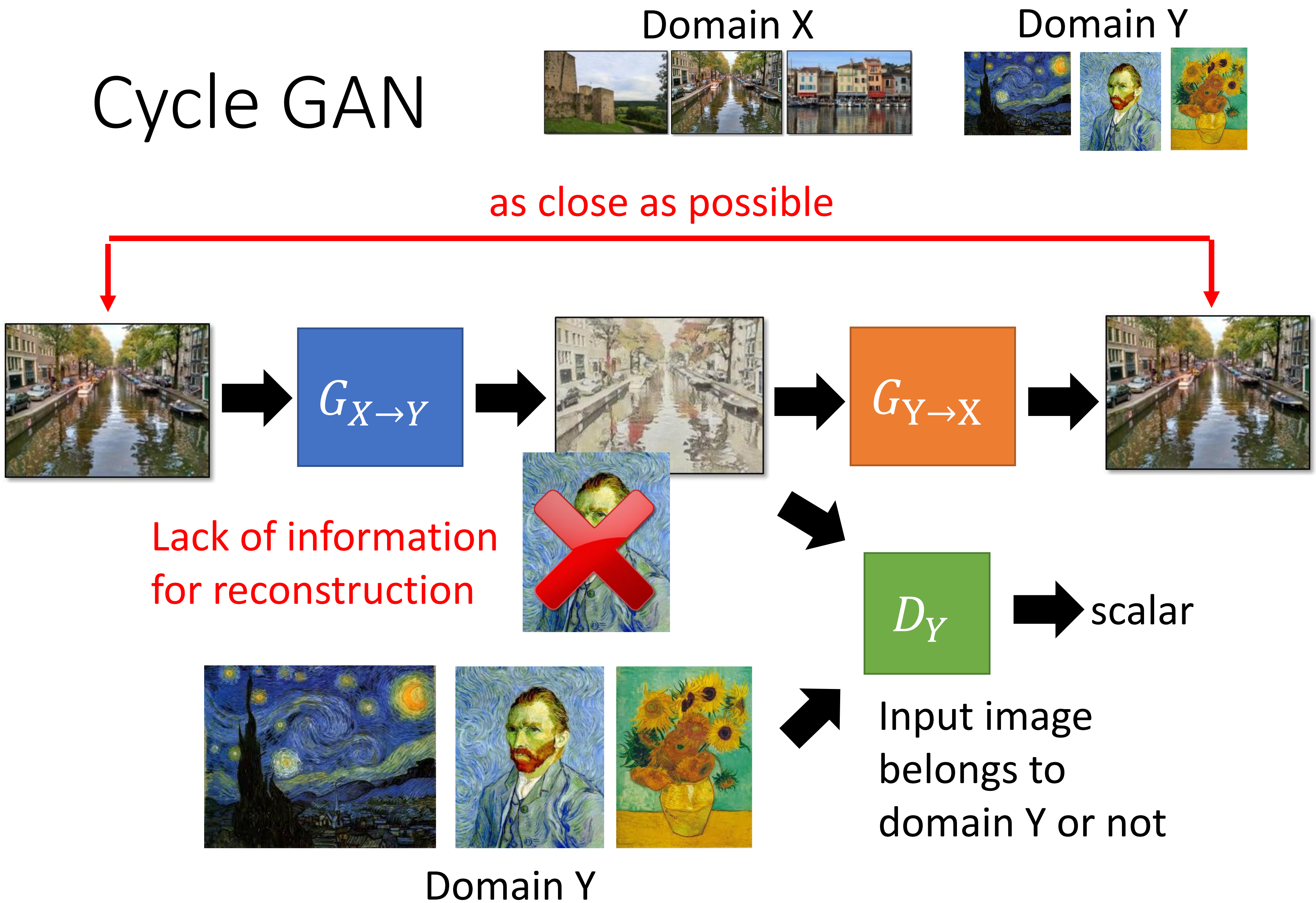
Not what we want

D_Y

scalar

Input image
belongs to
domain Y or not

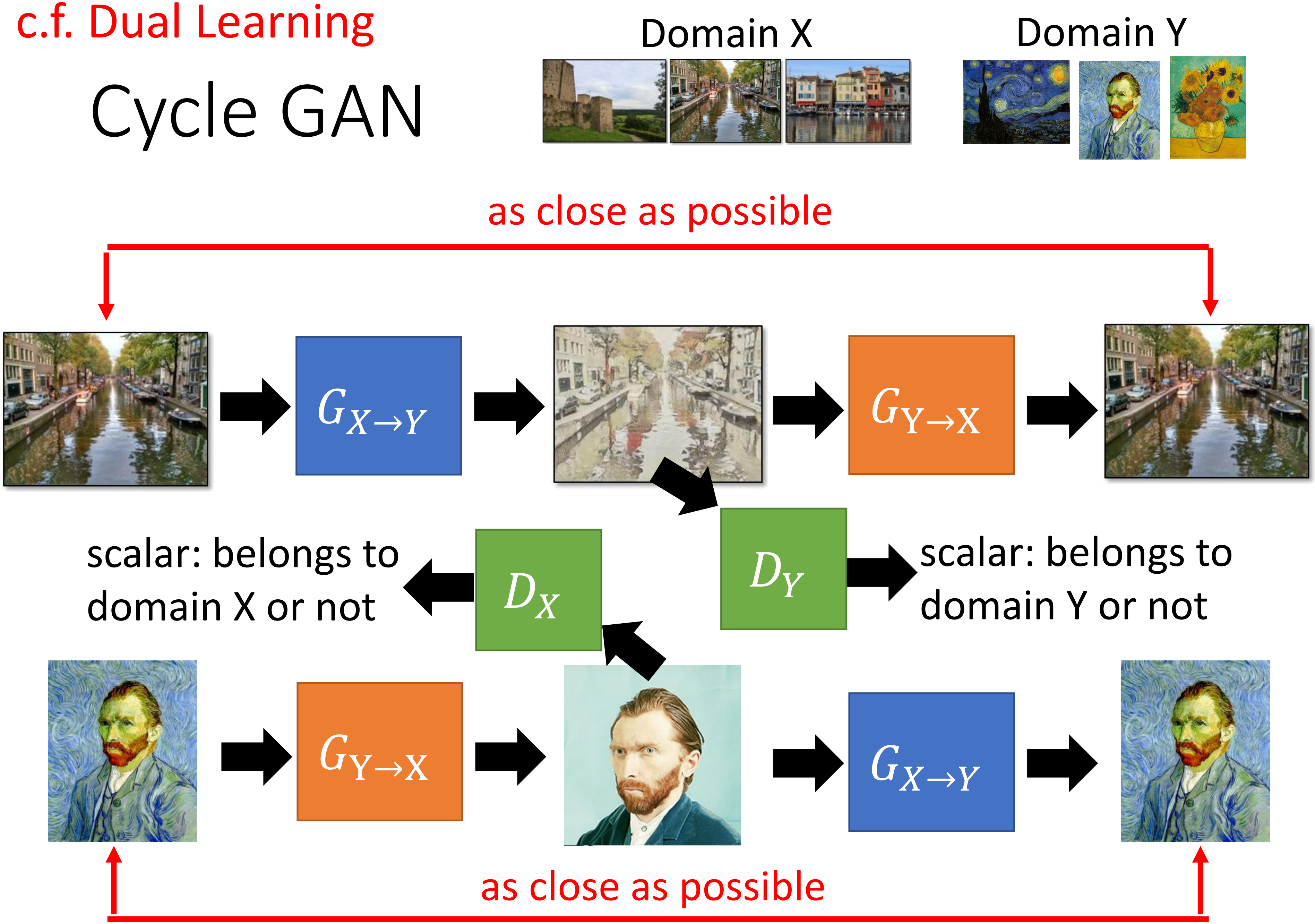
Domain Y



Cycle GAN
Domain X
Domain Y
as close as possible
$G_{X \to Y}$
$G_{Y \to X}$
Lack of information
for reconstruction
D_Y
scalar
Input image
belongs to
domain Y or not
Domain Y

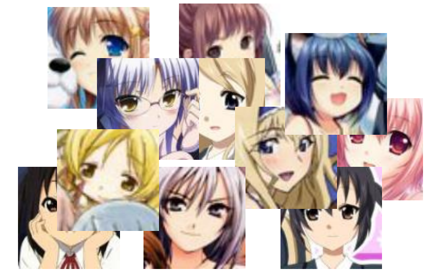
c.f. Dual Learning

Cycle GAN



動畫化的世界

input



output ***domain***

- Using the code: https://github.com/Hi-king/kawaii_creator
- It is not cycle GAN, Disco GAN

Conditional Generation

Modifying input code

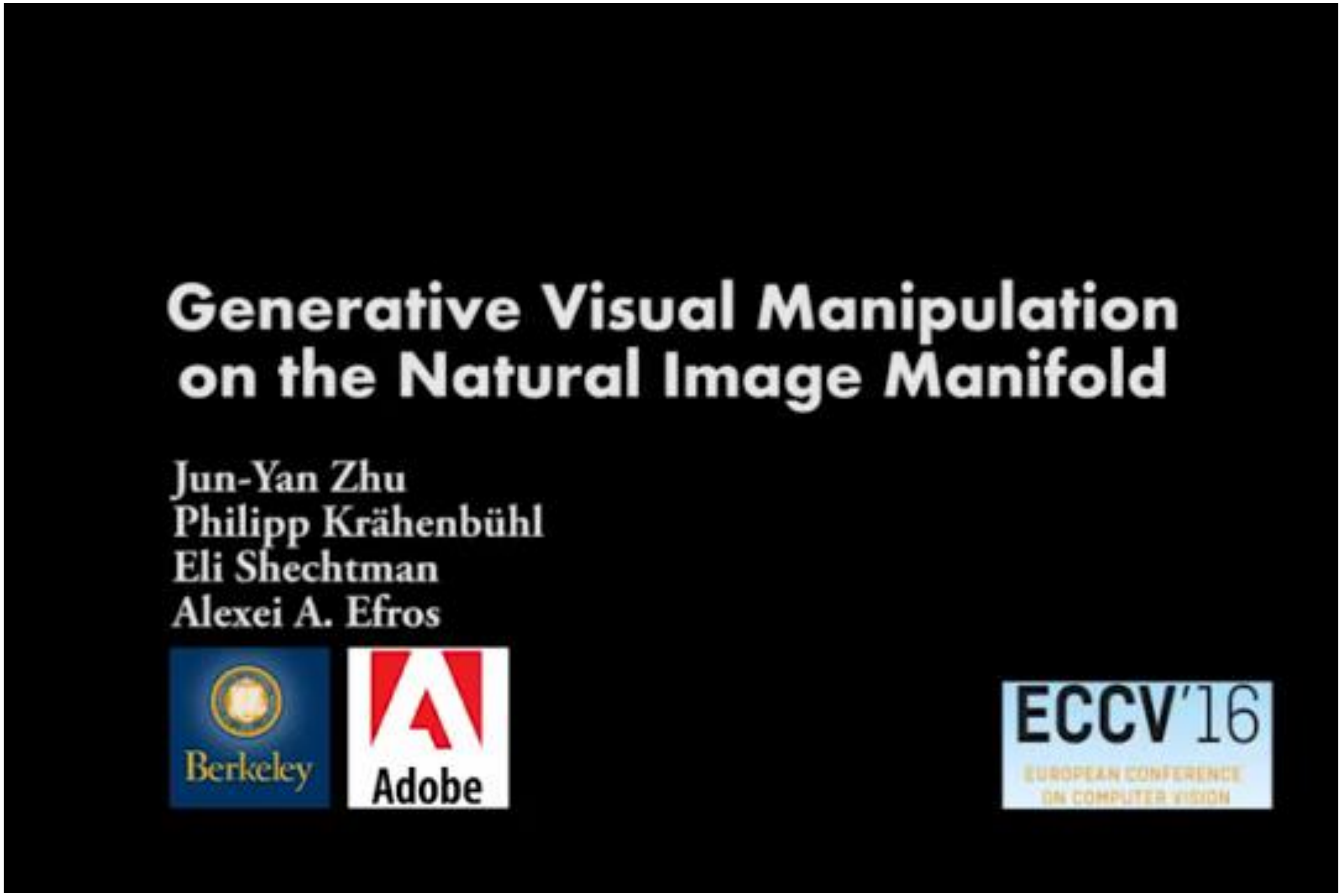
- Making code has influence (InfoGAN)
- Connection code space with attribute

Controlling by input objects

- Paired data
- Unpaired data
- Unsupervised

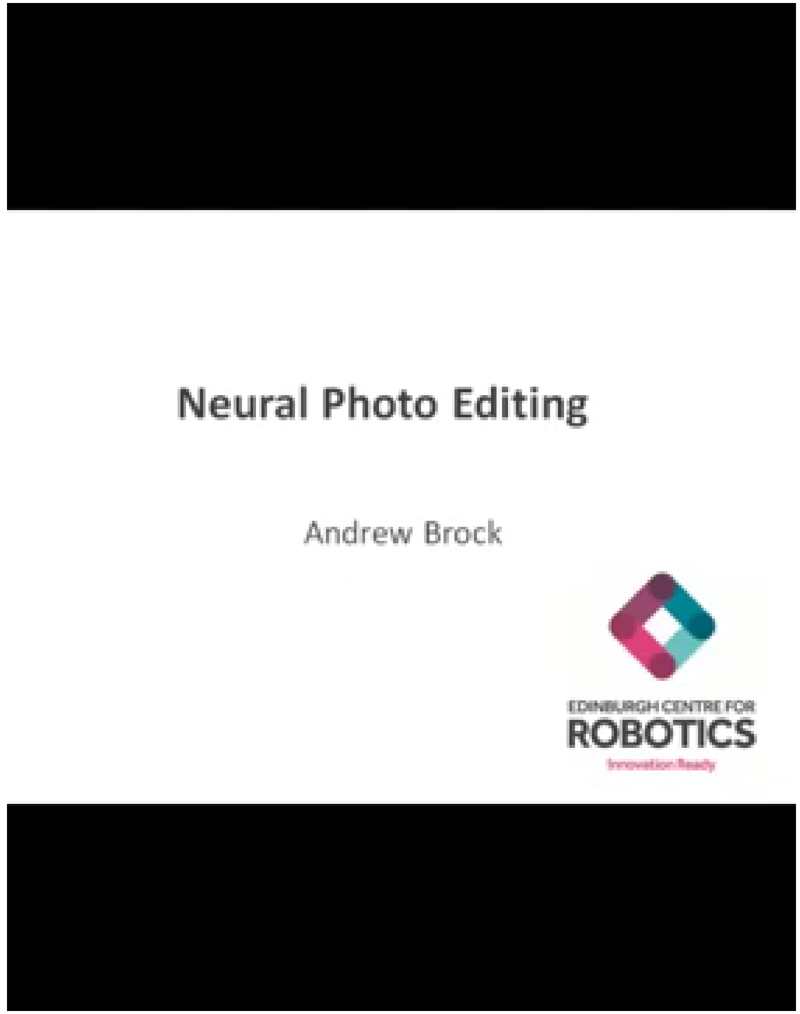
Feature extraction

- Domain Independent Feature
- Improving Auto-encoder (VAE-GAN, BiGAN)



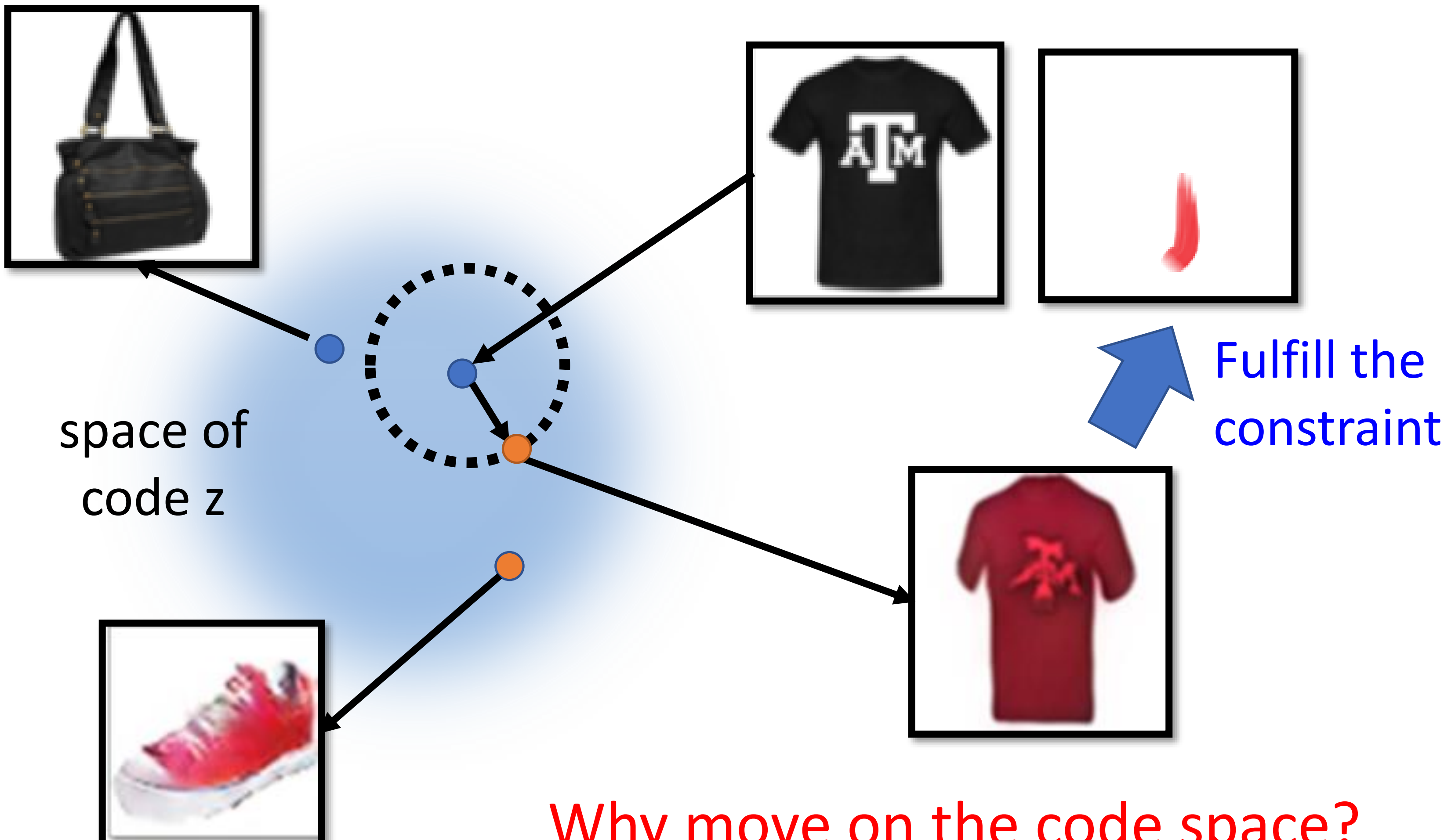
https://www.youtube.com/watch?v=9c4z6YsBGQ0

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman and Alexei A. Efros. "Generative Visual Manipulation on the Natural Image Manifold", ECCV, 2016.



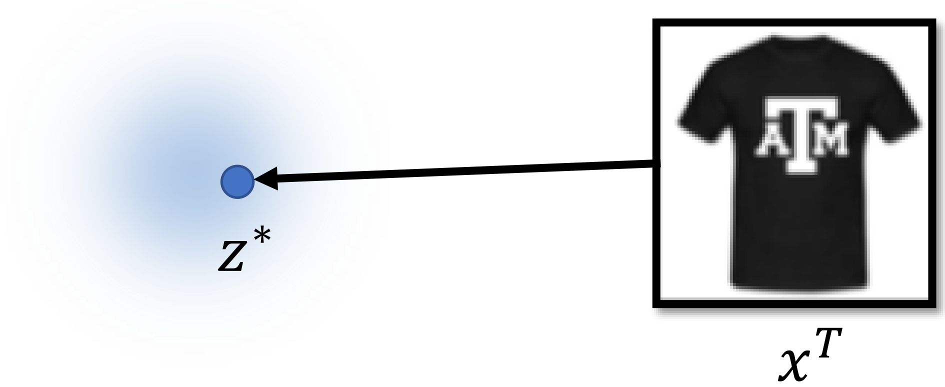
Andrew Brock, Theodore Lim, J.M. Ritchie, Nick Weston, **Neural Photo Editing with Introspective Adversarial Networks,** arXiv preprint, 2017

Basic Idea



Why move on the code space?

Back to z



- ***Method 1***

$$z^* = arg\min_z \underline{L(G(z), x^T)}$$

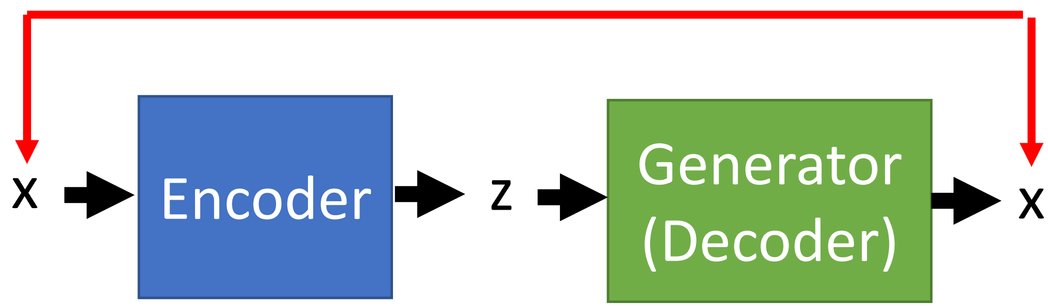
Gradient Descent

Difference between G(z) and x^T

➢ Pixel-wise
➢ By another network

- ***Method 2***

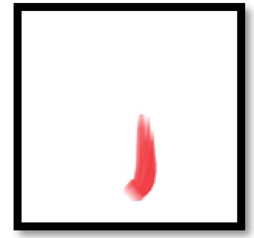
as close as possible



- ***Method 3***

Using the results from ***method 2*** as the initialization of ***method 1***

Editing Photos



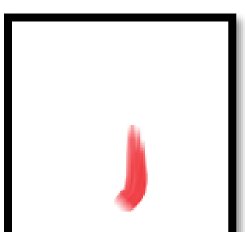
- z_0 is the code of the input image

image

$$z^* = arg\min_z U\big(G(z)\big) + \lambda_1 \|z - z_0\|^2 - \lambda_2 D\big(G(z)\big)$$

Using discriminator to check the image is realistic or not

Not too far away from the original image



Does it fulfill the constraint of editing?

Conditional Generation

Modifying input code

- Making code has influence (InfoGAN)
- Connection code space with attribute

Controlling by input objects

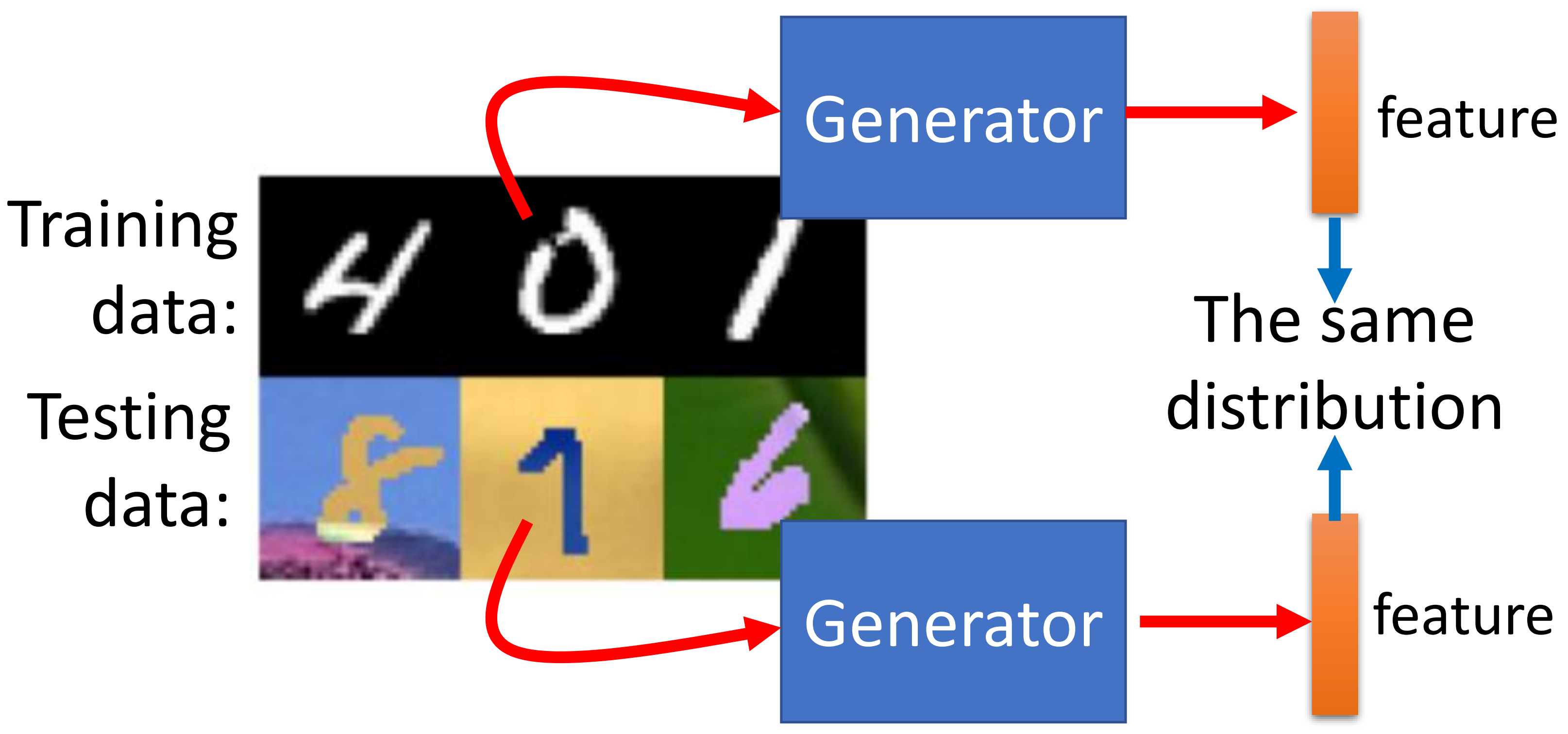
- Paired data
- Unpaired data
- Unsupervised

Feature extraction

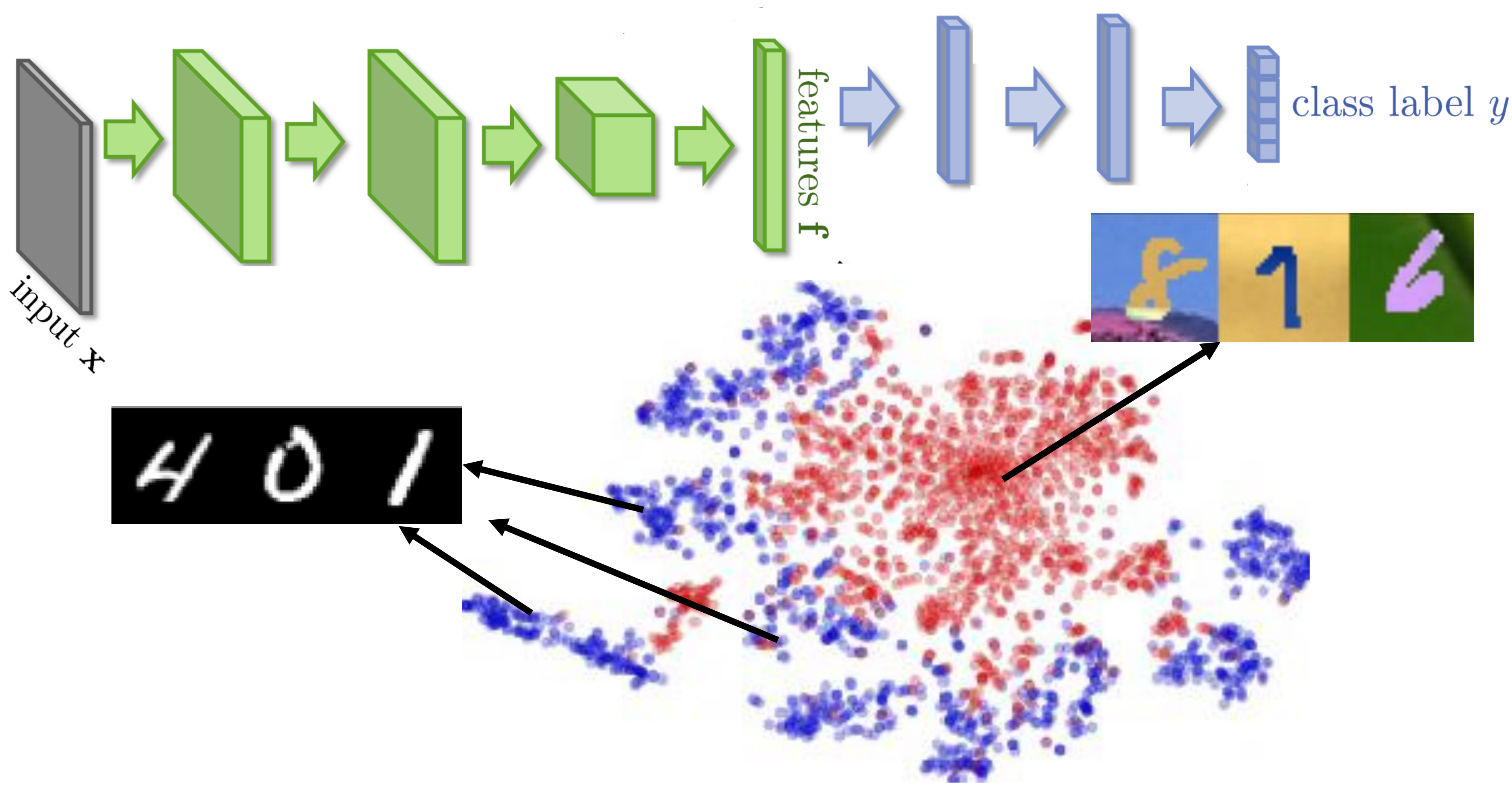
- Domain Independent Feature
- Improving Auto-encoder (VAE-GAN, BiGAN)

Domain Independent Features

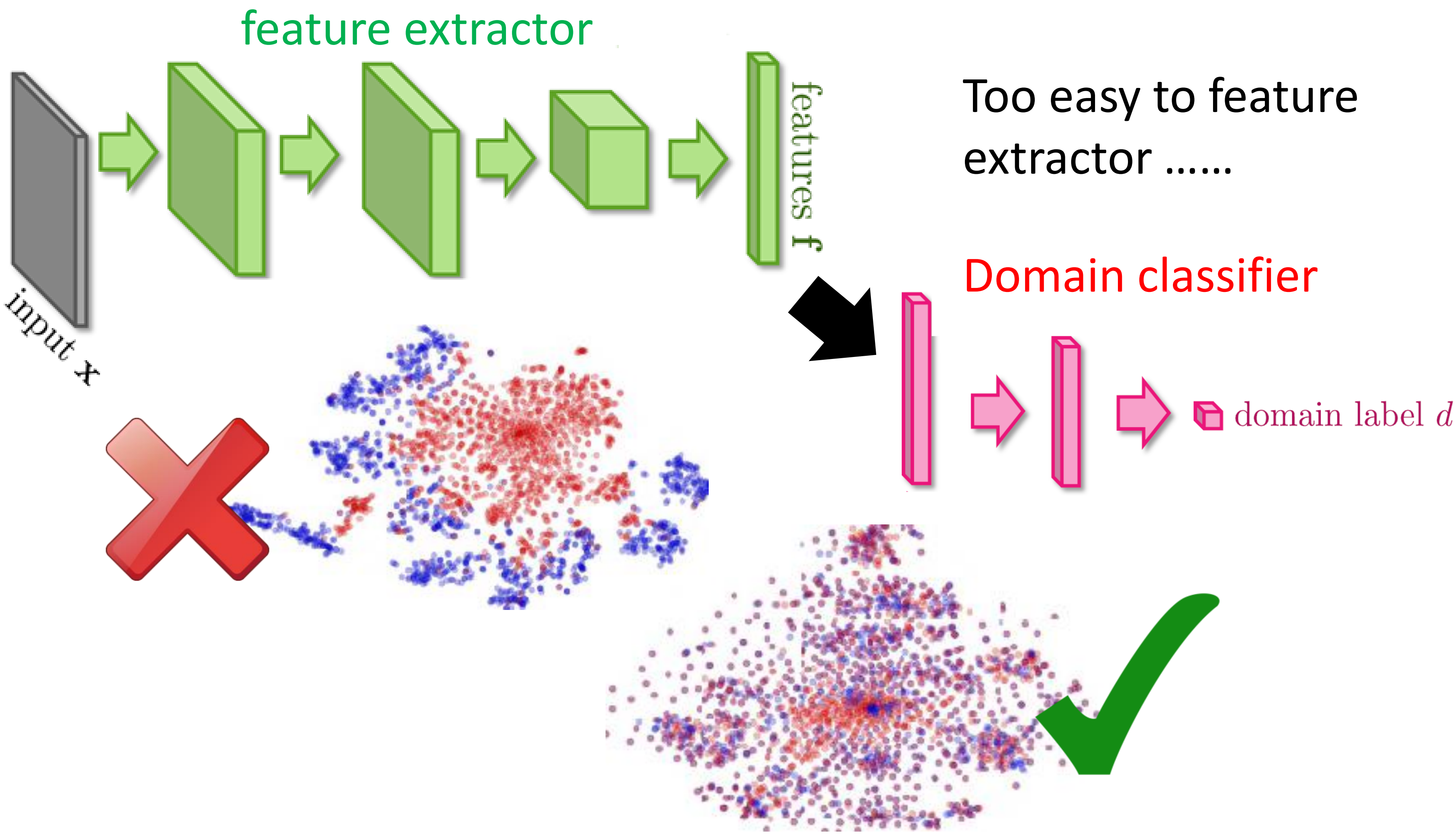
- Training and testing data are in different domains



Domain Independent Features



Domain Independent Features

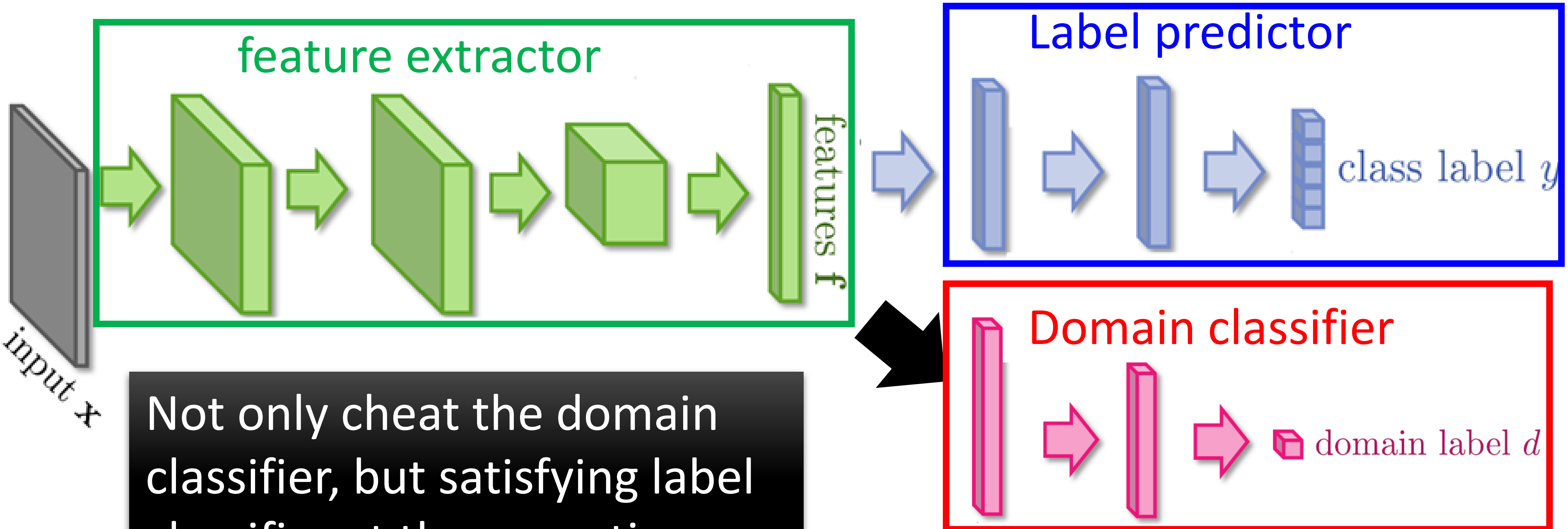


Too easy to feature extractor ......

Domain-adversarial training

Maximize label classification accuracy + minimize domain classification accuracy

Maximize label classification accuracy

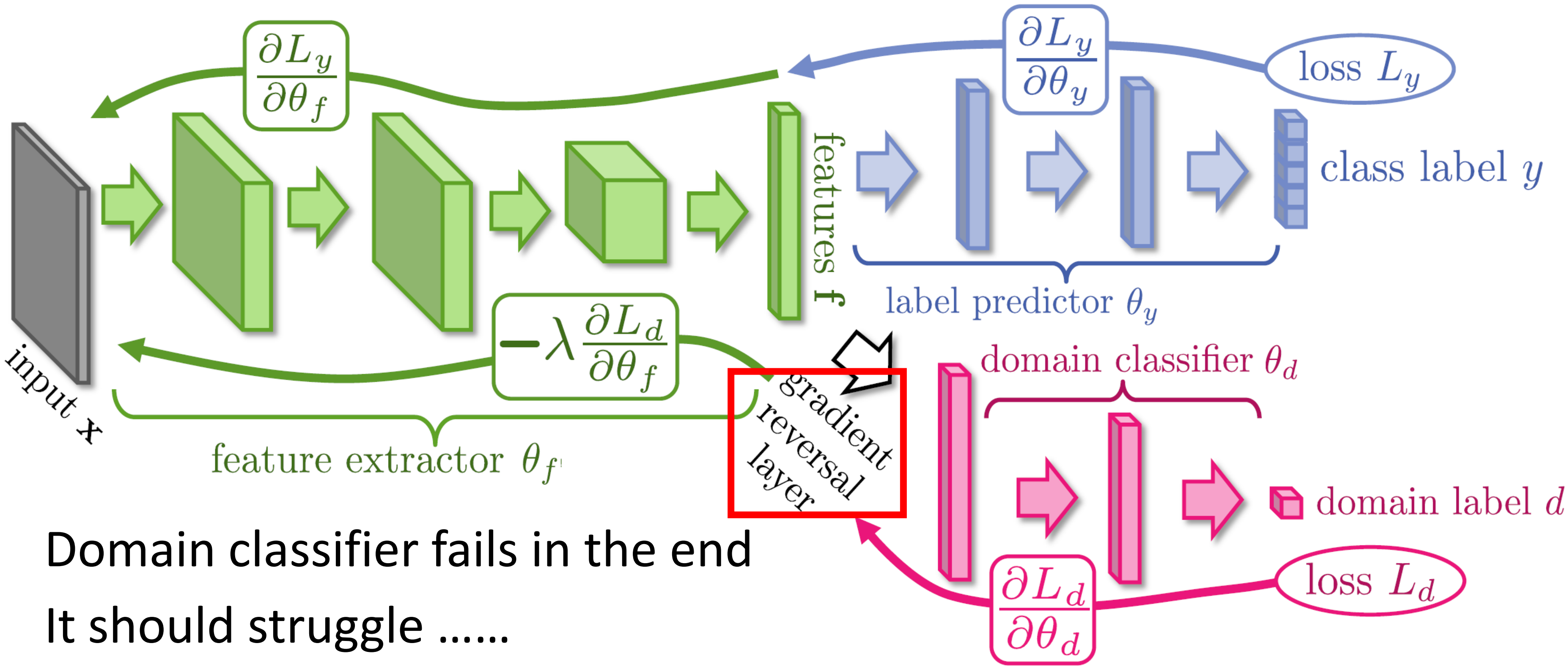


Not only cheat the domain classifier, but satisfying label classifier at the same time

Maximize domain classification accuracy

This is a big network, but different parts have different goals.

Domain-adversarial training



Domain classifier fails in the end

It should struggle ……

Yaroslav Ganin, Victor Lempitsky, Unsupervised Domain Adaptation by Backpropagation, ICML, 2015

Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, Domain-Adversarial Training of Neural Networks, JMLR, 2016

Domain-adversarial training

METHOD	SOURCE / TARGET	MNIST / MNIST-M	SYN NUMBERS / SVHN	SVHN / MNIST	SYN SIGNS / GTSRB
SOURCE ONLY		.5749	.8665	.5919	.7400
PROPOSED APPROACH		**.8149** (57.9%)	**.9048** (66.1%)	**.7107** (29.3%)	**.8866** (56.7%)
TRAIN ON TARGET		.9891	.9244	.9951	.9987

Yaroslav Ganin, Victor Lempitsky, Unsupervised Domain Adaptation by Backpropagation, ICML, 2015

Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, Domain-Adversarial Training of Neural Networks, JMLR, 2016

Conditional Generation

Modifying input code

- Making code has influence (InfoGAN)
- Connection code space with attribute

Controlling by input objects

- Paired data
- Unpaired data
- Unsupervised

Feature extraction

- Domain Independent Feature
- Improving Auto-encoder (VAE-GAN, BiGAN)

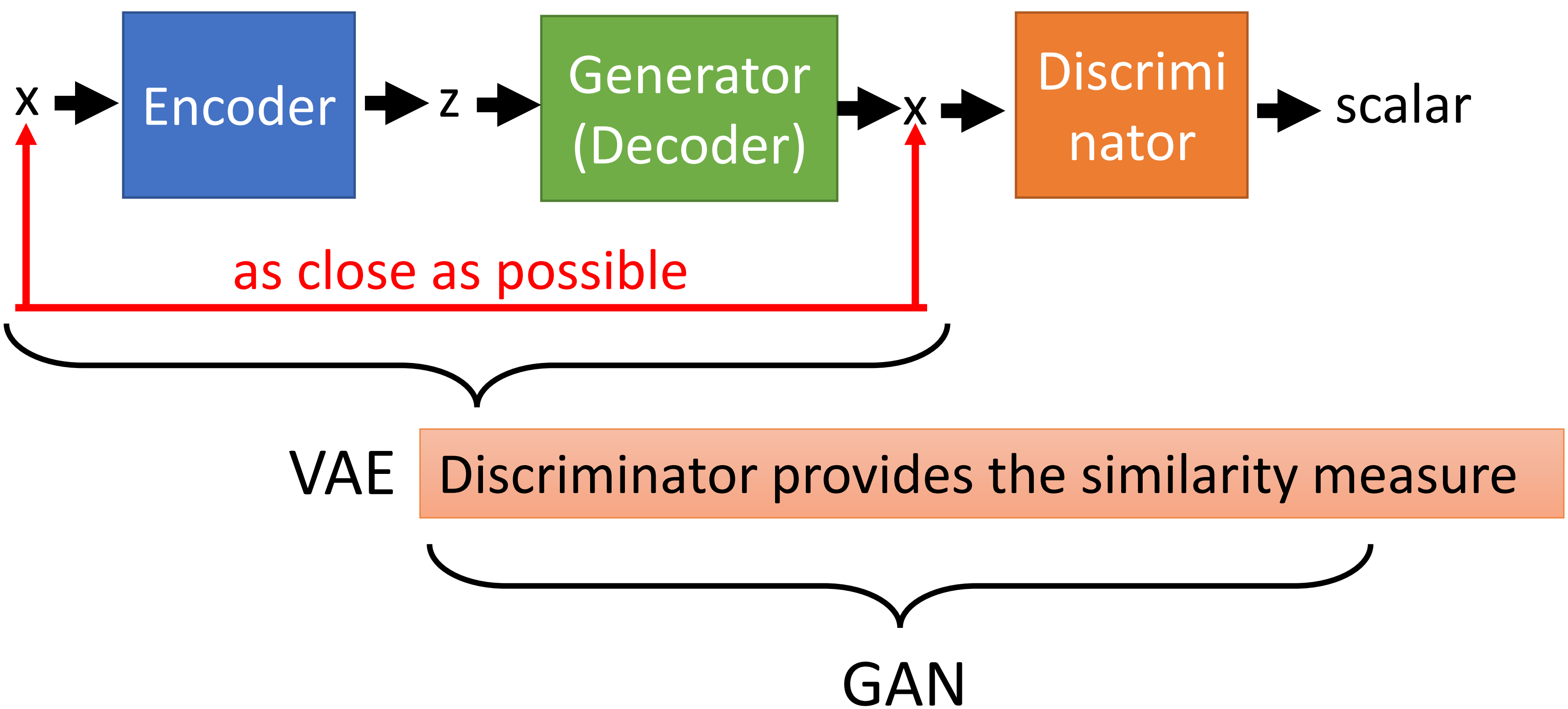
VAE-GAN

Anders Boesen, Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, Ole Winther, "Autoencoding beyond pixels using a learned similarity metric", ICML. 2016

- ➢ Minimize reconstruction error
- ➢ z close to normal

- ➢ Minimize reconstruction error
- ➢ Cheat discriminator

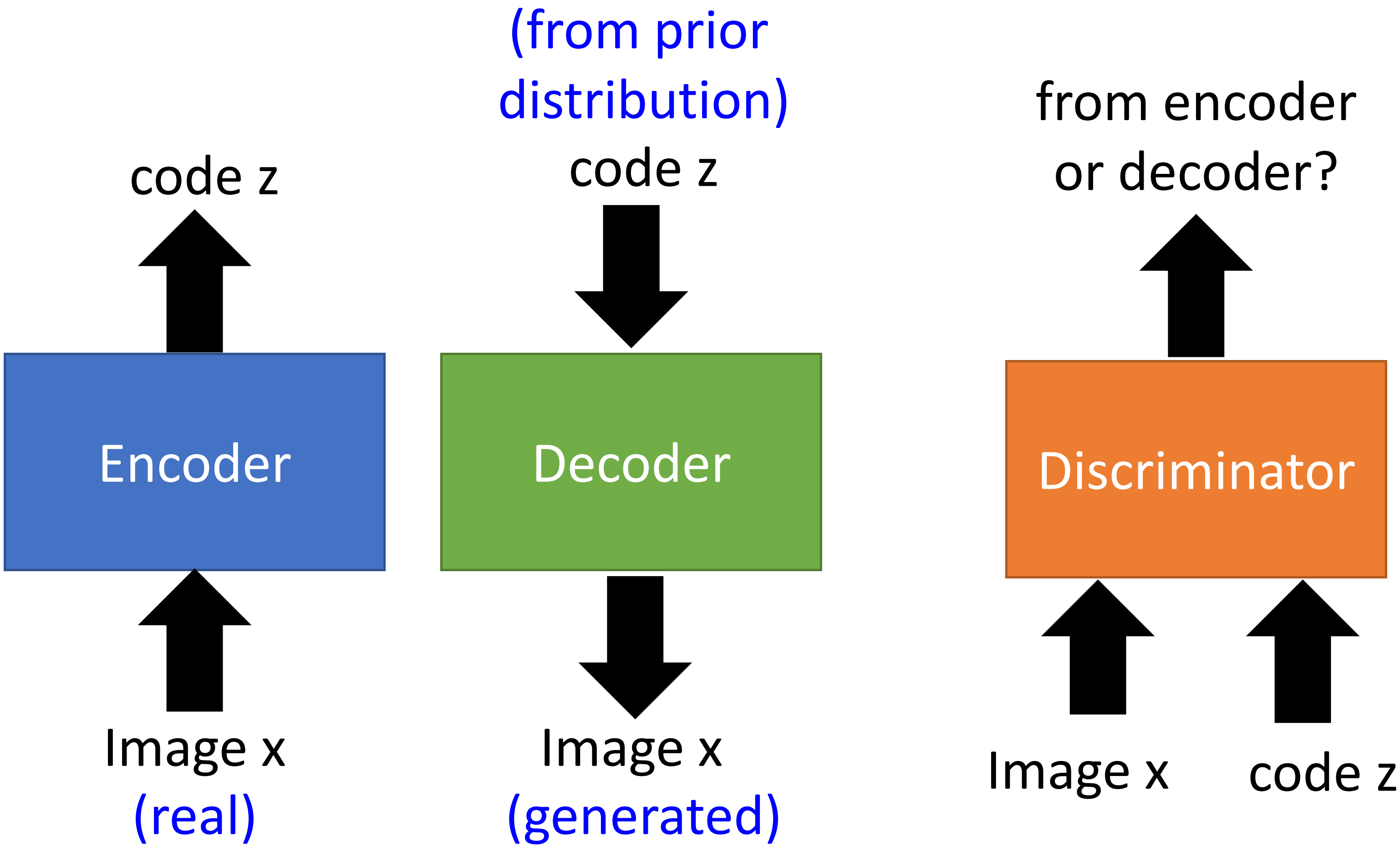
- ➢ Discriminate real, generated and reconstructed images



BiGAN

Jeff Donahue, Philipp Krähenbühl, Trevor Darrell, “Adversarial Feature Learning”, ICLR, 2017

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, Aaron Courville, “Adversarially Learned Inference”, ICLR, 2017

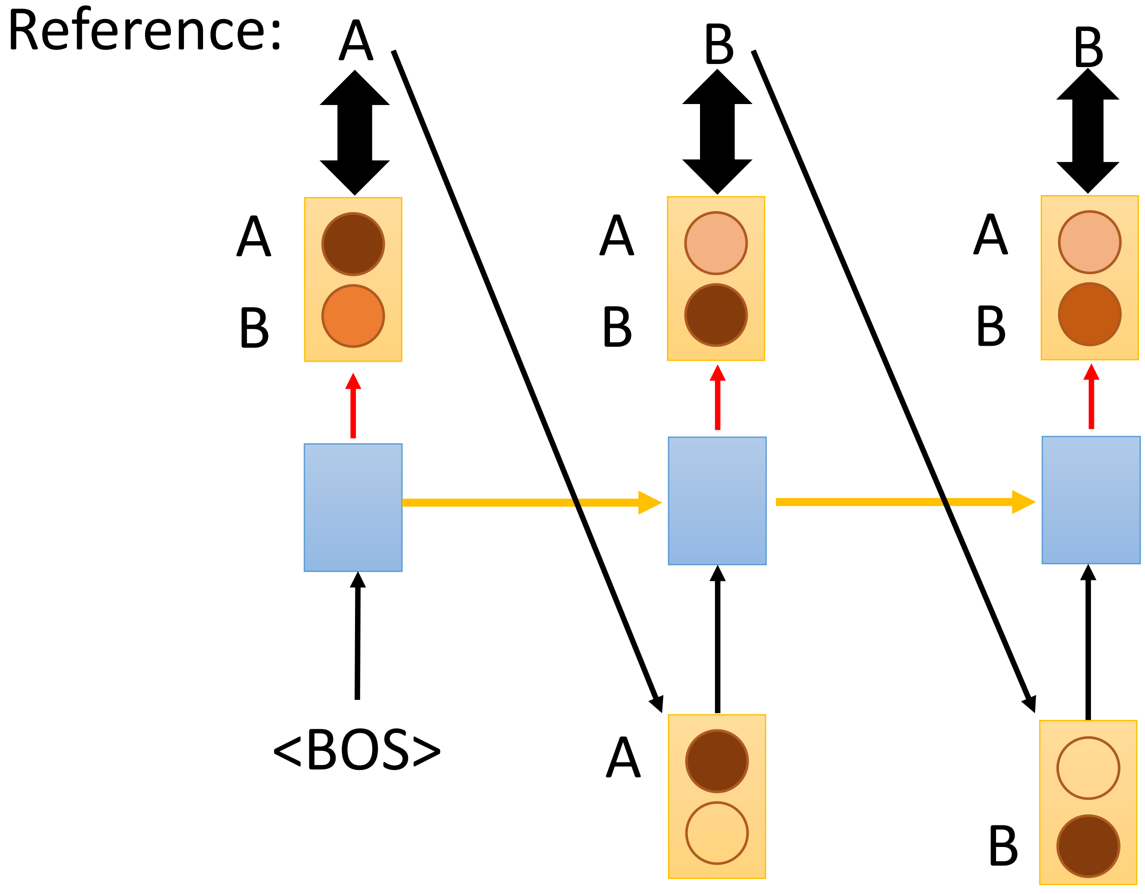


Lecture II

- Conditional Generation
- Sequence Generation
- A Little Bit of Theory (option)

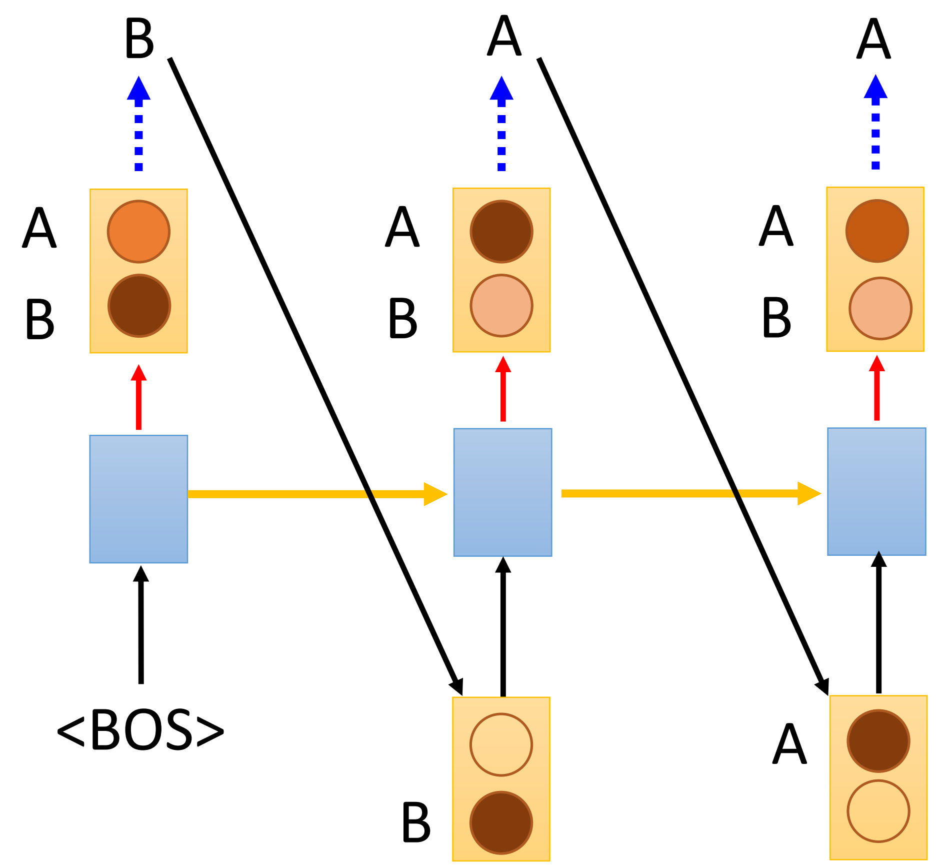
Sentence Generation

- ***Training***



Sentence Generation

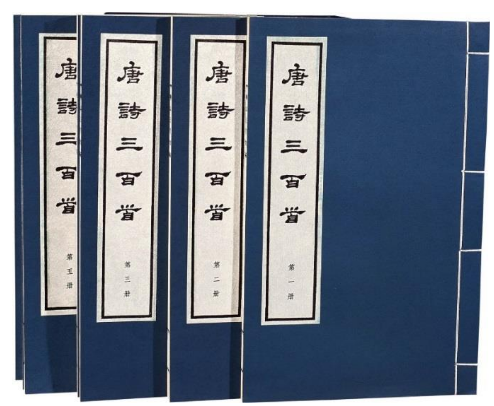
- ***Generation (Testing)***



Sentence Generation

Generator → sentence x

sentence x → Discriminator → Real or fake



Original GAN

Sentence Generation

- Initialize generator G and discriminator D
- In each iteration:

> - Sample real sentences x from database
> - Generate sentences $\tilde{x}$ by G
> - Update D to increase $D(x)$ and decrease $D(\tilde{x})$

> - Update Gen such that
>
> 
>

Sentence Generation

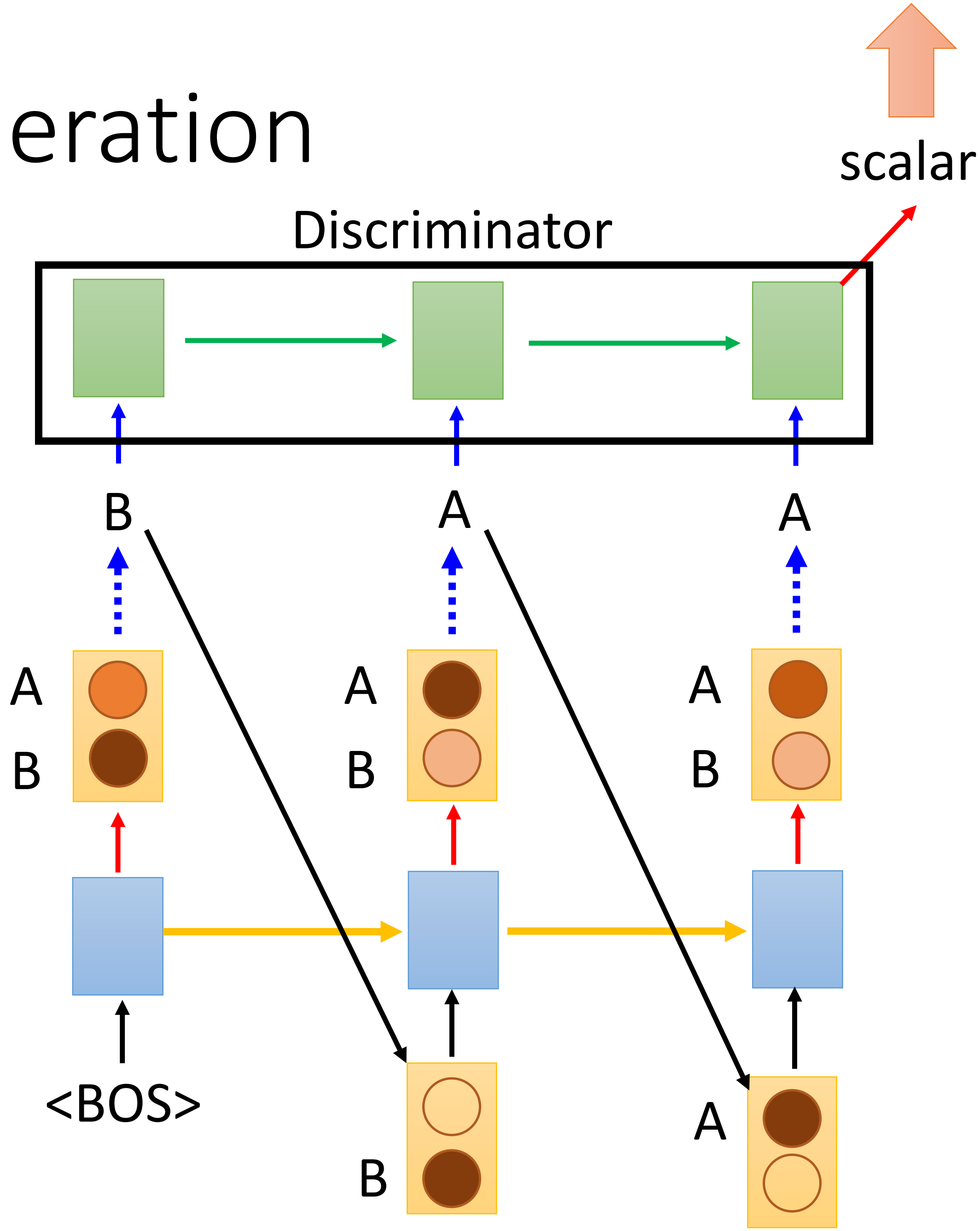
Can we do backpropogation?

NO!

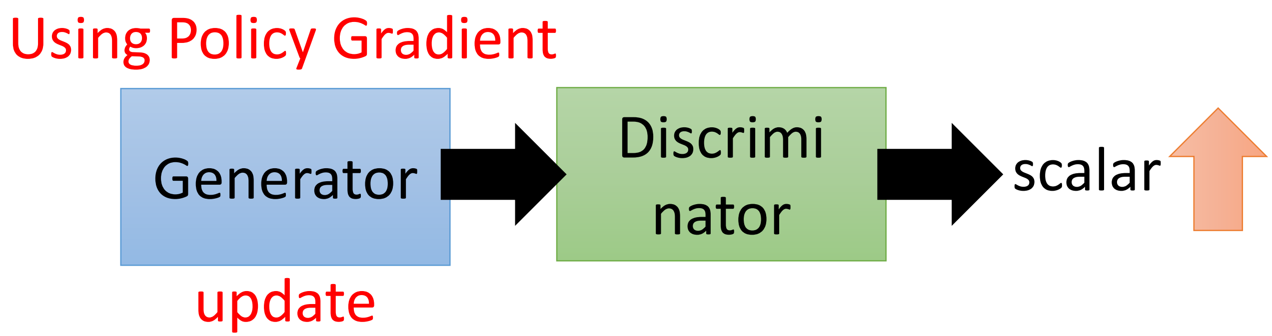
Tuning generator a little bit will not change the output.

In the paper of improved WGAN …

(ignoring sampling process)



Sentence Generation - SeqGAN



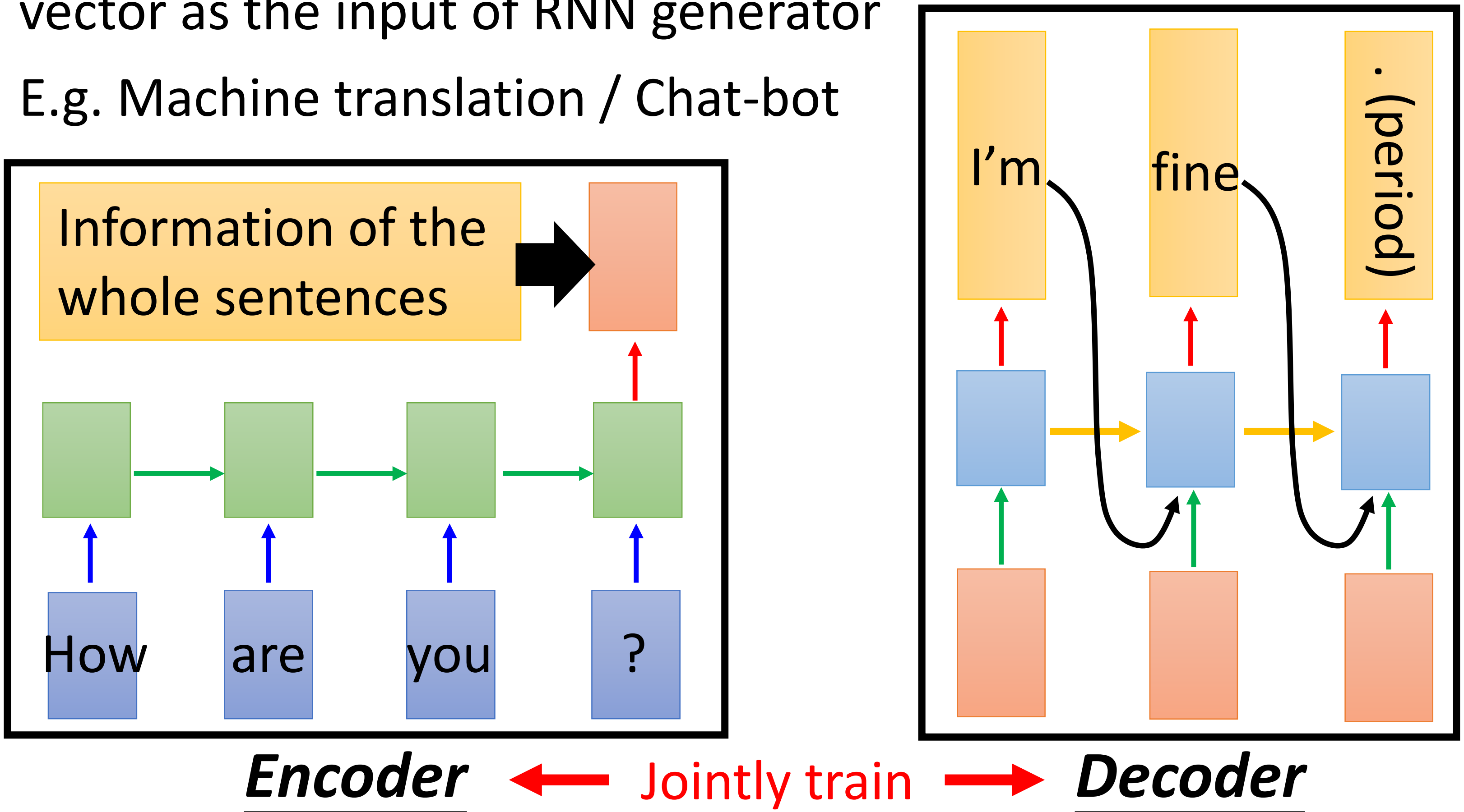
- Using ***<u>Reinforcement learning</u>***
  - Consider the discriminator as **reward function**
  - Consider the output of discriminator as total reward
  - Update generator to increase discriminator = to get maximum total reward

Ref: Lantao Yu, Weinan Zhang, Jun Wang, Yong Yu, "SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient", AAAI, 2017

Conditional Generation

Sequence-to-sequence learning

- Represent the input condition as a vector, and consider the vector as the input of RNN generator
- E.g. Machine translation / Chat-bot



Chat-bot with GAN

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, Dan Jurafsky, "Adversarial Learning for Neural Dialogue Generation", arXiv preprint, 2017

Input sentence/history h → Chatbot (En, De) → response sentence x

Input sentence/history h → Discriminator → Real or fake
response sentence x → Discriminator

human dialogues → Discriminator

Conditional GAN

Example Results

感謝 段逸林 同學提供實驗結果

input	I love you.

input	Do you like machine learning?

input	I thought I have met you before.

input	Let's go to the party.

input	How do you feel about the president?

感謝 王耀賢 同學提供實驗結果

Cycle GAN

Negative sentence to positive sentence:

it's a crappy day -> it's a great day
i wish you could be here -> you could be here
it's not a good idea -> it's good idea
i miss you -> i love you
i don't love you -> i love you
i can't do that -> i can do that
i feel so sad -> i happy
it's a bad day -> it's a good day
it's a crappy day -> it's a great day
sorry for doing such a horrible thing -> thanks for doing a great thing
my doggy is sick -> my doggy is my doggy
i am so hungry -> i am so
my little doggy is sick -> my little doggy is my little doggy

Lecture II

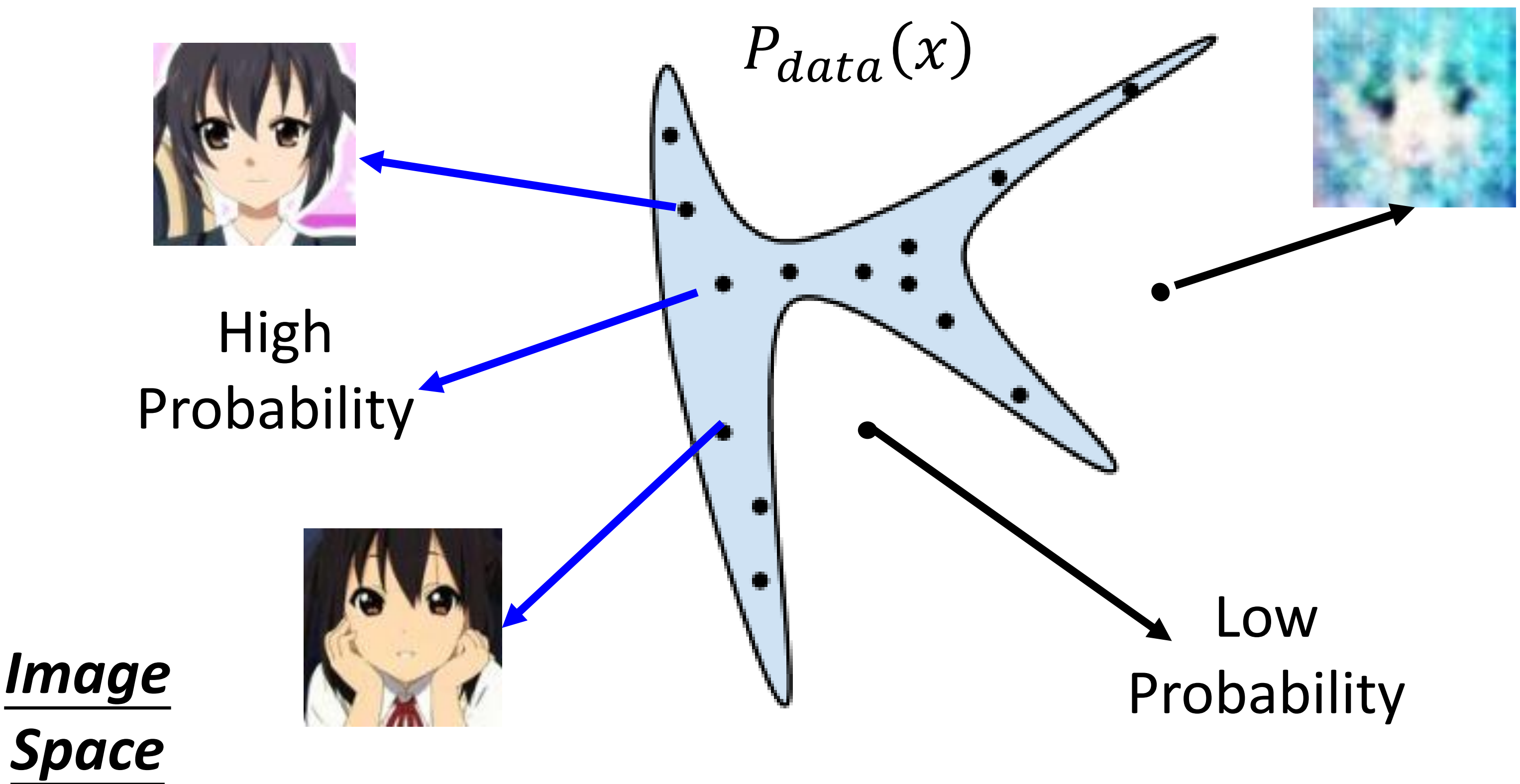
Conditional Generation

Sequence Generation

A Little Bit of Theory (option)

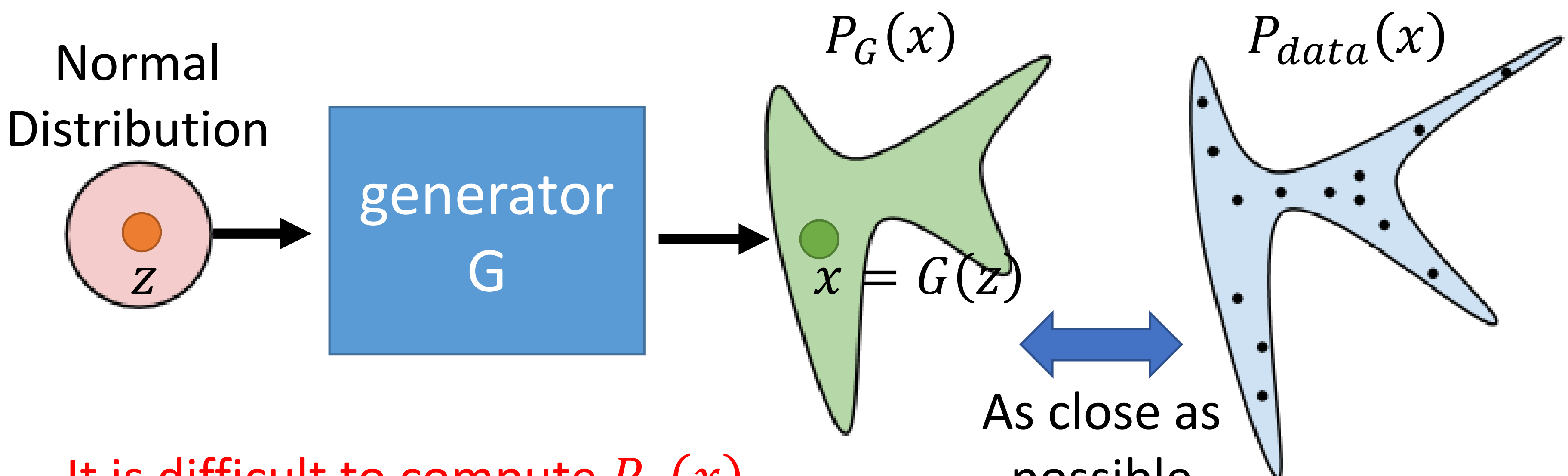
Theory behind GAN

- The data we want to generate has a distribution $P_{data}(x)$



Theory behind GAN

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



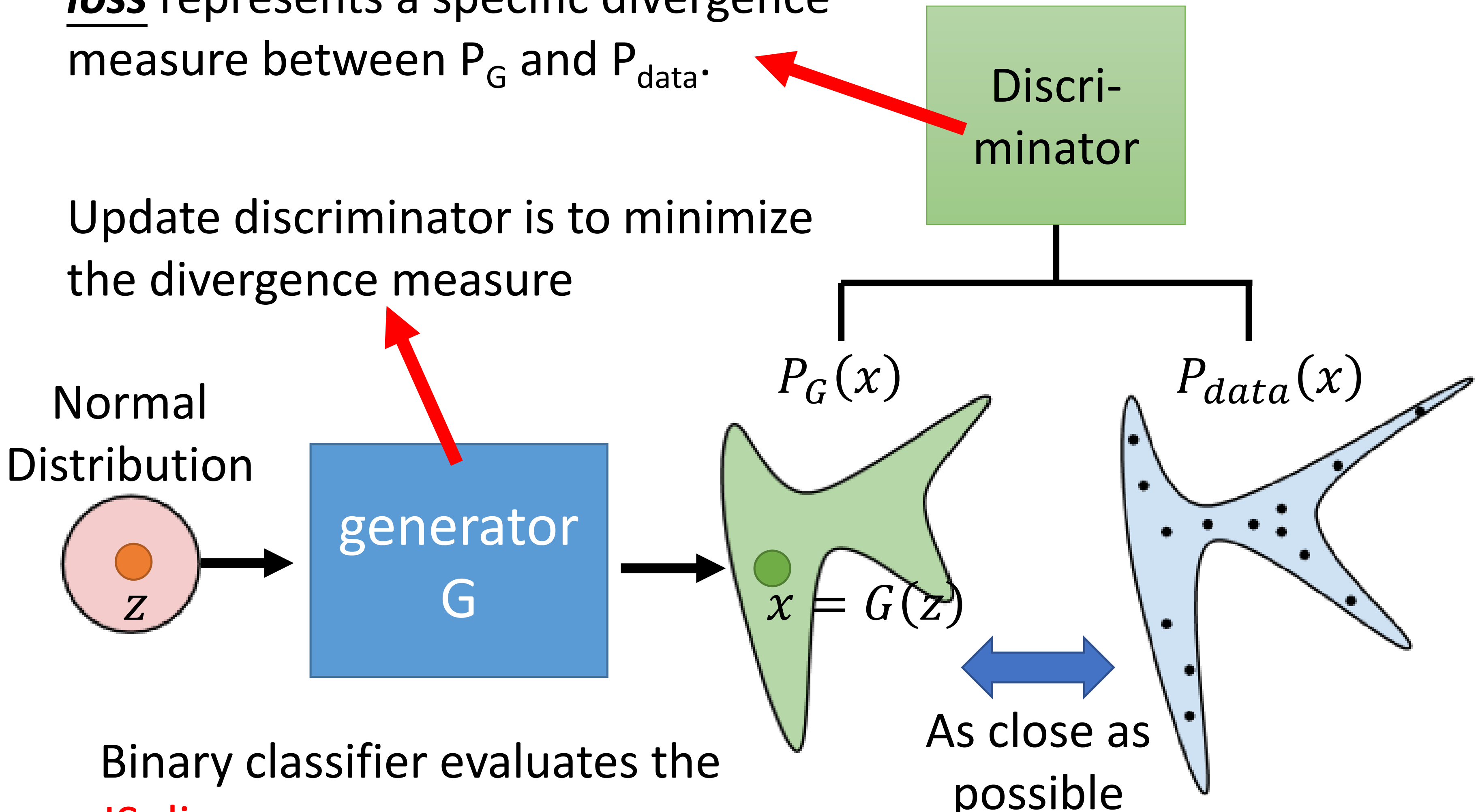
It is difficult to compute $P_G(x)$

We do not know what the distribution looks like.

https://blog.openai.com/generative-models/

When the discriminator is well trained, its ***loss*** represents a specific divergence measure between P_G and P_{data}.

Update discriminator is to minimize the divergence measure



Binary classifier evaluates the JS divergence

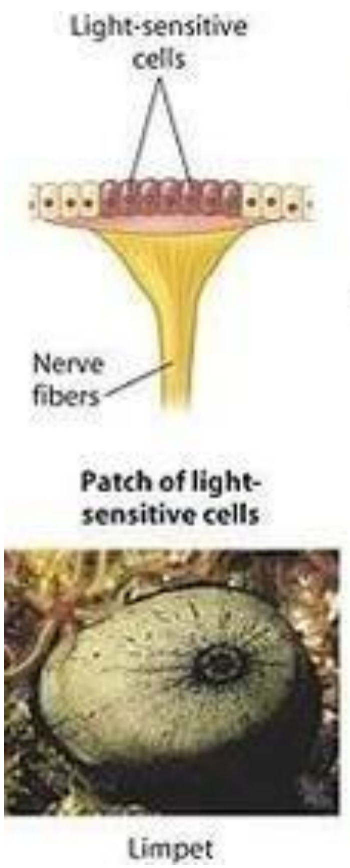
You can design the discriminator to evaluate other divergence.

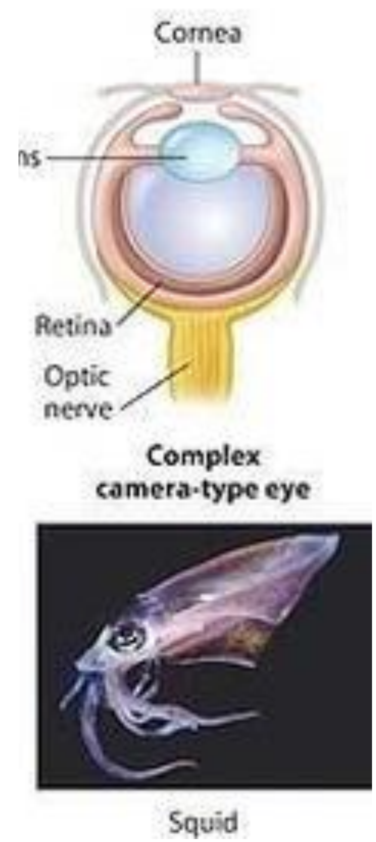
https://blog.openai.com/generative-models/

http://www.guokr.com/post/773890/

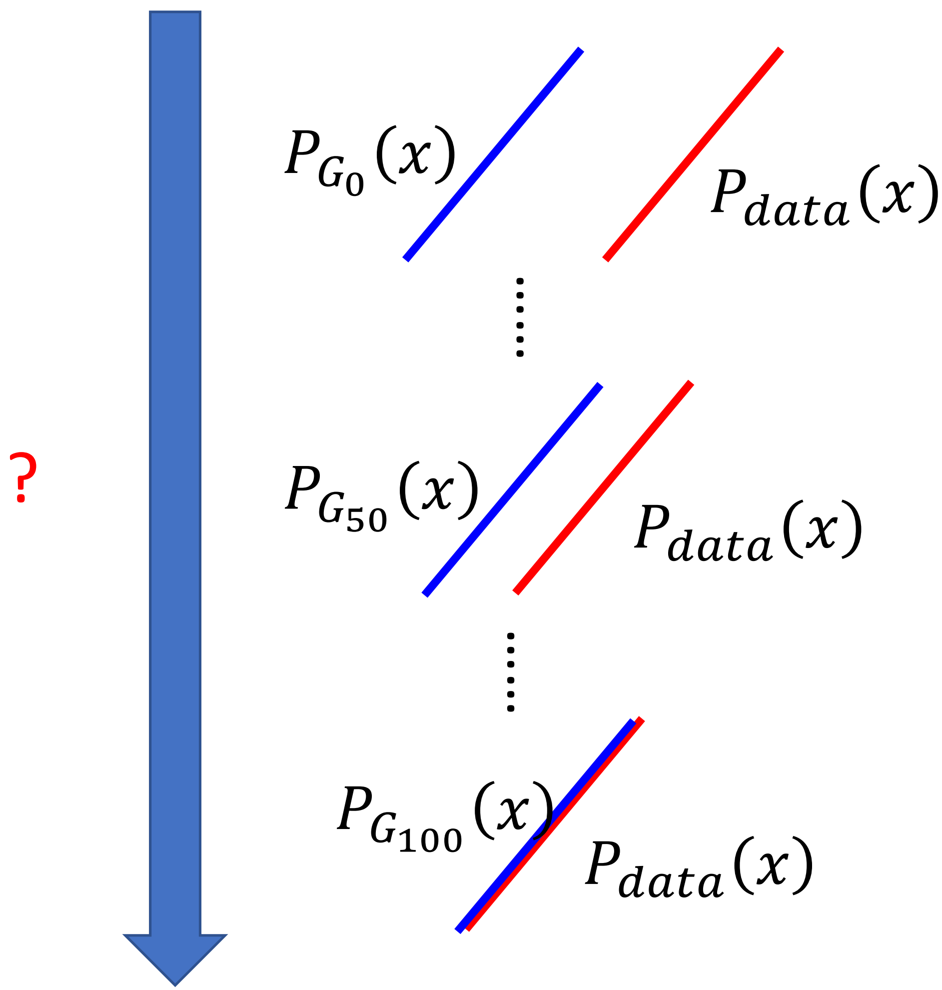
Why GAN is hard to train?

Better





Why GAN is hard to train?



$JS(P_{G_1}||P_{data}) = log2$

$JS(P_{G_2}||P_{data}) = log2$

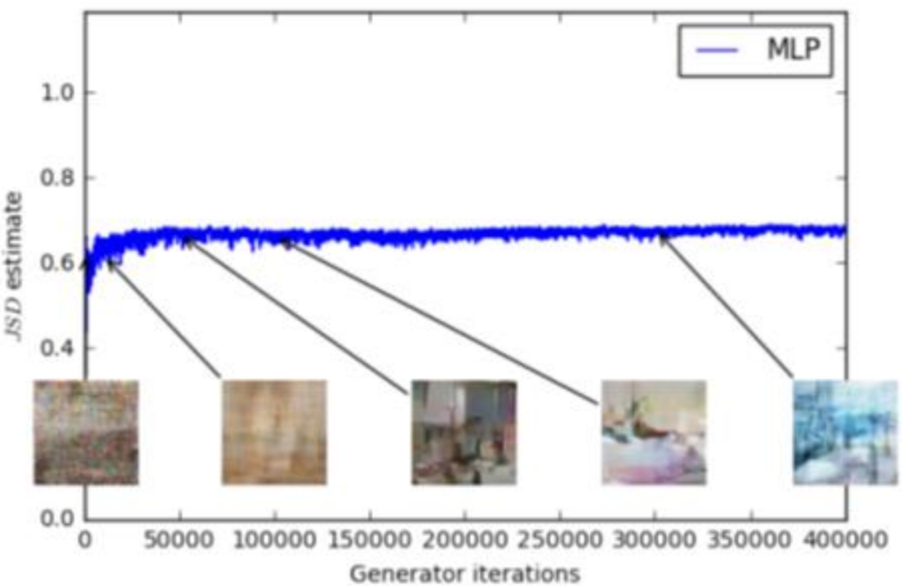
Not really better ......

$JS(P_{G_2}||P_{data}) = 0$

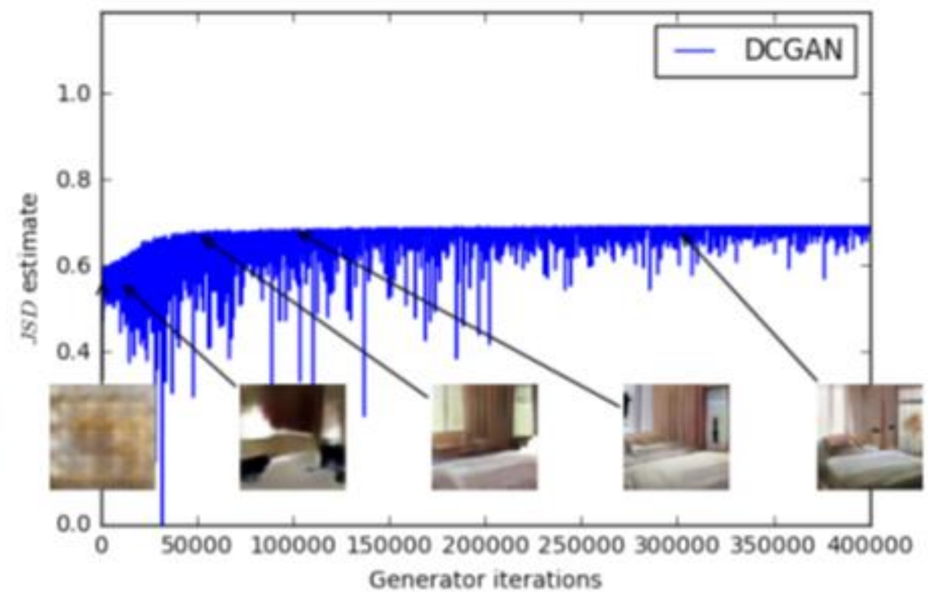
Evaluating JS divergence

https://arxiv.org/abs/1701.07875

- JS divergence estimated by discriminator telling little information



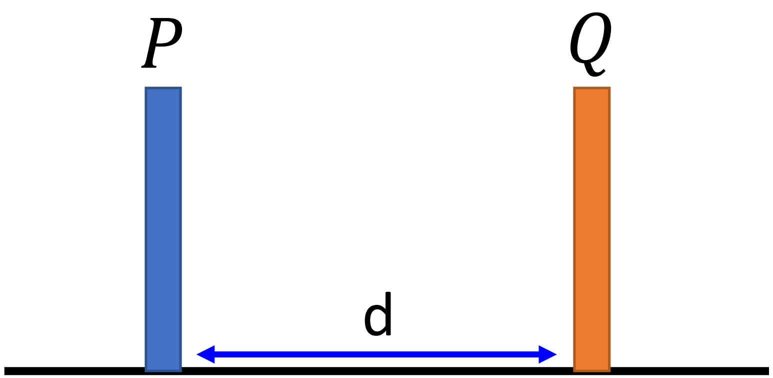
Weak Generator



Strong Generator

Earth Mover's Distance

- Considering one distribution P as a pile of earth, and another distribution Q as the target
- The average distance the earth mover has to move the earth.



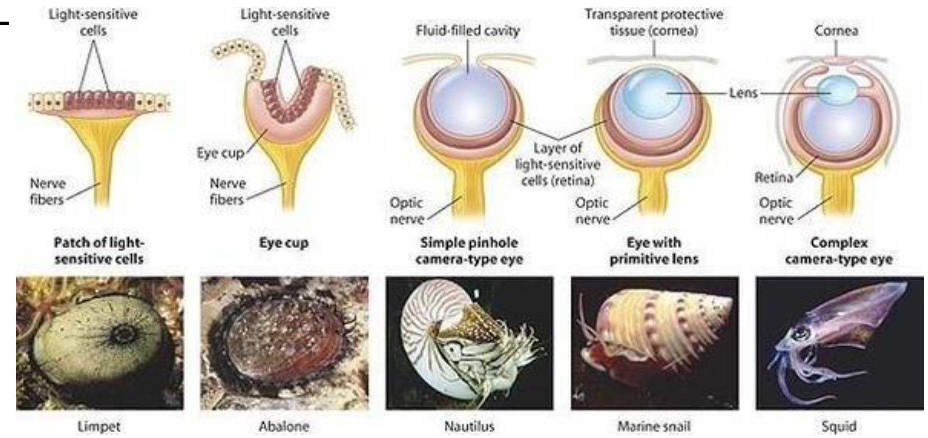
$$W(P,Q) = d$$

Why Earth Mover's Distance?

$$D_f(P_{data}||P_G)$$

$$\Downarrow$$

$$W(P_{data}, P_G)$$

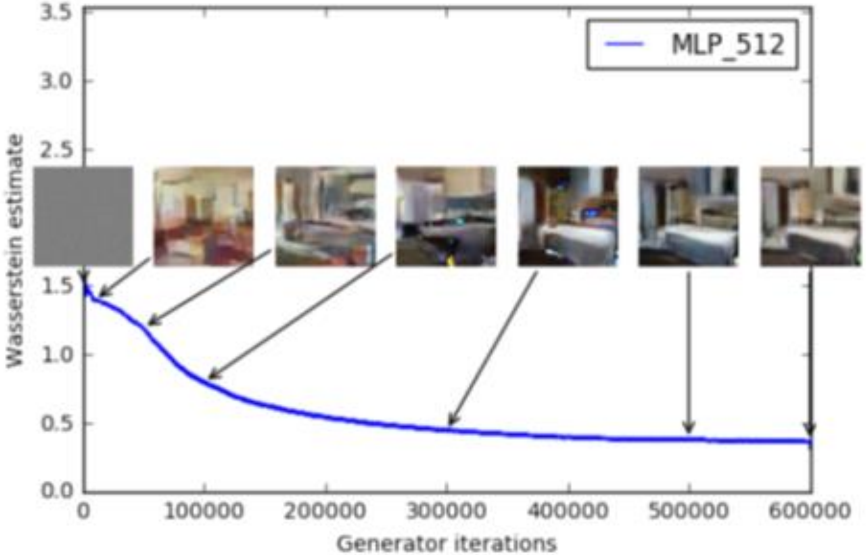


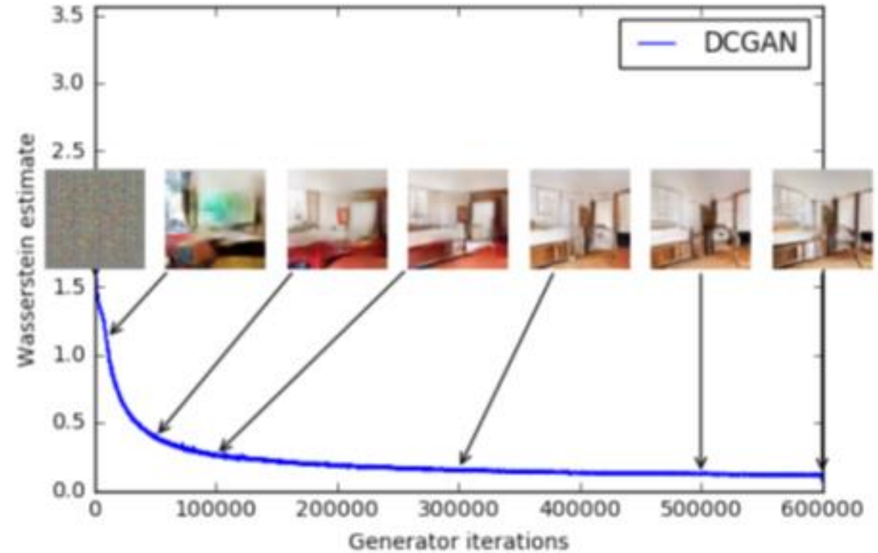
d_0 $\quad\quad\quad\quad\quad\quad\quad\quad$ d_{50}

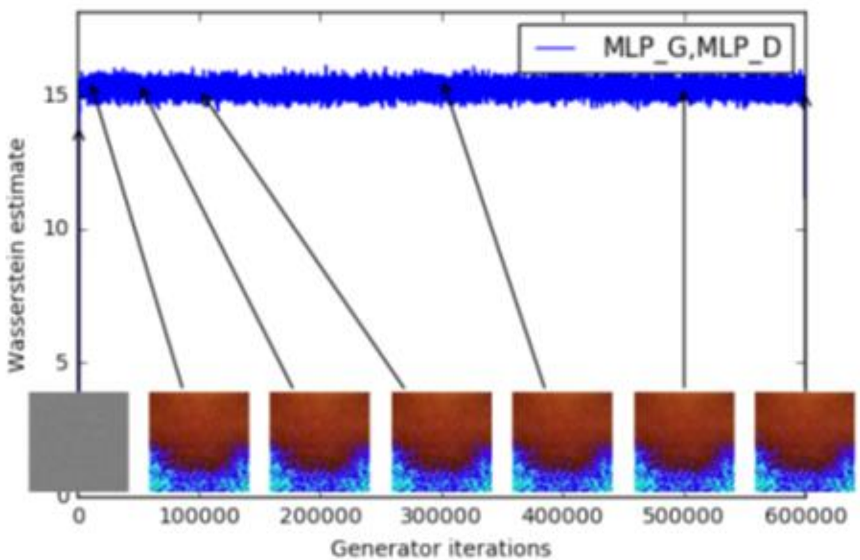
P_{G_0} $\longleftrightarrow$ P_{data} …… $P_{G_{50}}$ $\longleftrightarrow$ P_{data} …… $P_{G_{100}}$ P_{data}

$$JS(P_{G_0}, P_{data}) = log2 \qquad JS(P_{G_{50}}, P_{data}) = log2 \qquad JS(P_{G_{100}}, P_{data}) = 0$$

$$W(P_{G_0}, P_{data}) = d_0 \qquad W(P_{G_{50}}, P_{data}) = d_{50} \qquad W(P_{G_{100}}, P_{data}) = 0$$





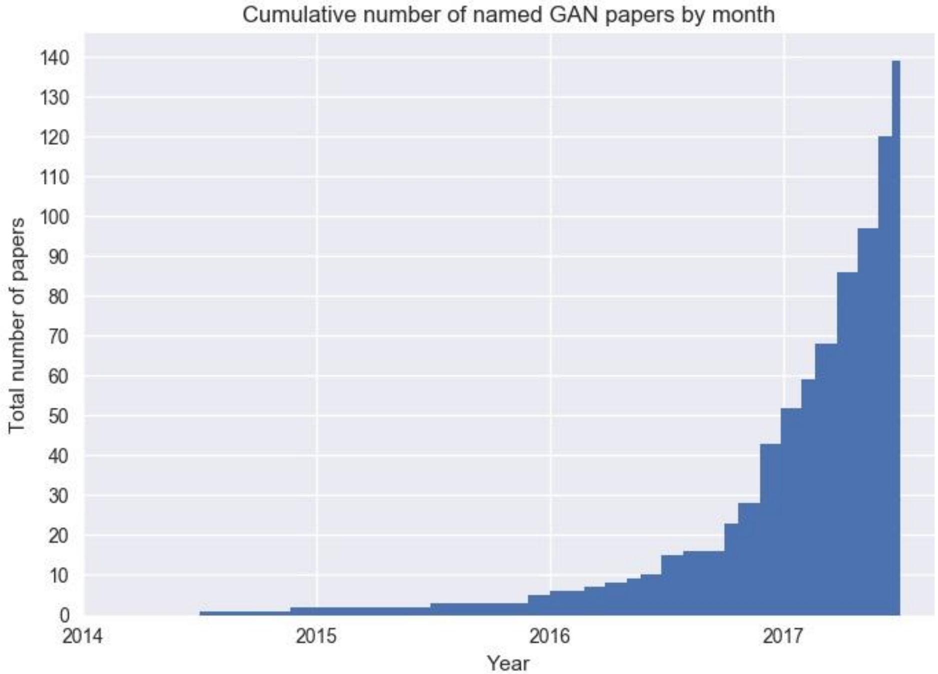


<u>Vertical</u>

$$W(P_{data}, P_G) = \max_{D \in 1-Lipschitz} \{ E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)] \}$$

https://arxiv.org/abs/1701.07875

To Learn more ……

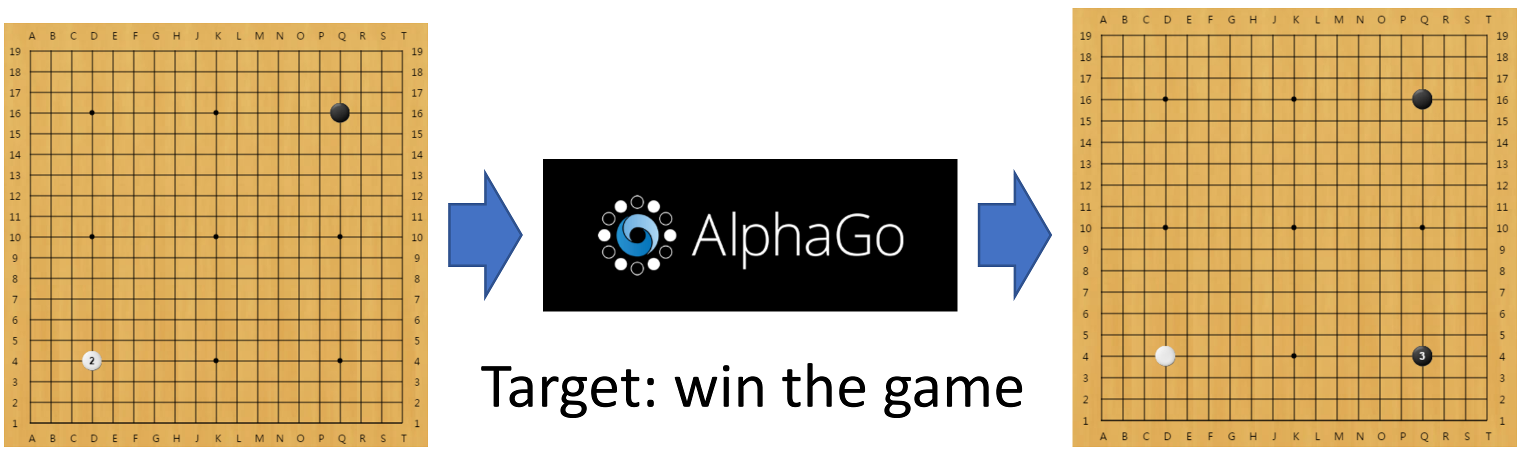


- GAN Zoo
  - https://github.com/hindupuravinash/the-gan-zoo
- Tricks: https://github.com/soumith/ganhacks
- Tutorial of Ian Goodfellow: https://arxiv.org/abs/1701.00160

Lecture III:
Decision Making and Control

Decision Making and Control

- Machine observe some inputs, takes an action, and finally achieve the target.
- E.g. Go playing



observation

Target: win the game

action

State: summarization of observation

Decision Making and Control

- Machine plays video games
- Widely studies:
  - Gym: https://gym.openai.com/
  - Universe: https://openai.com/blog/universe/

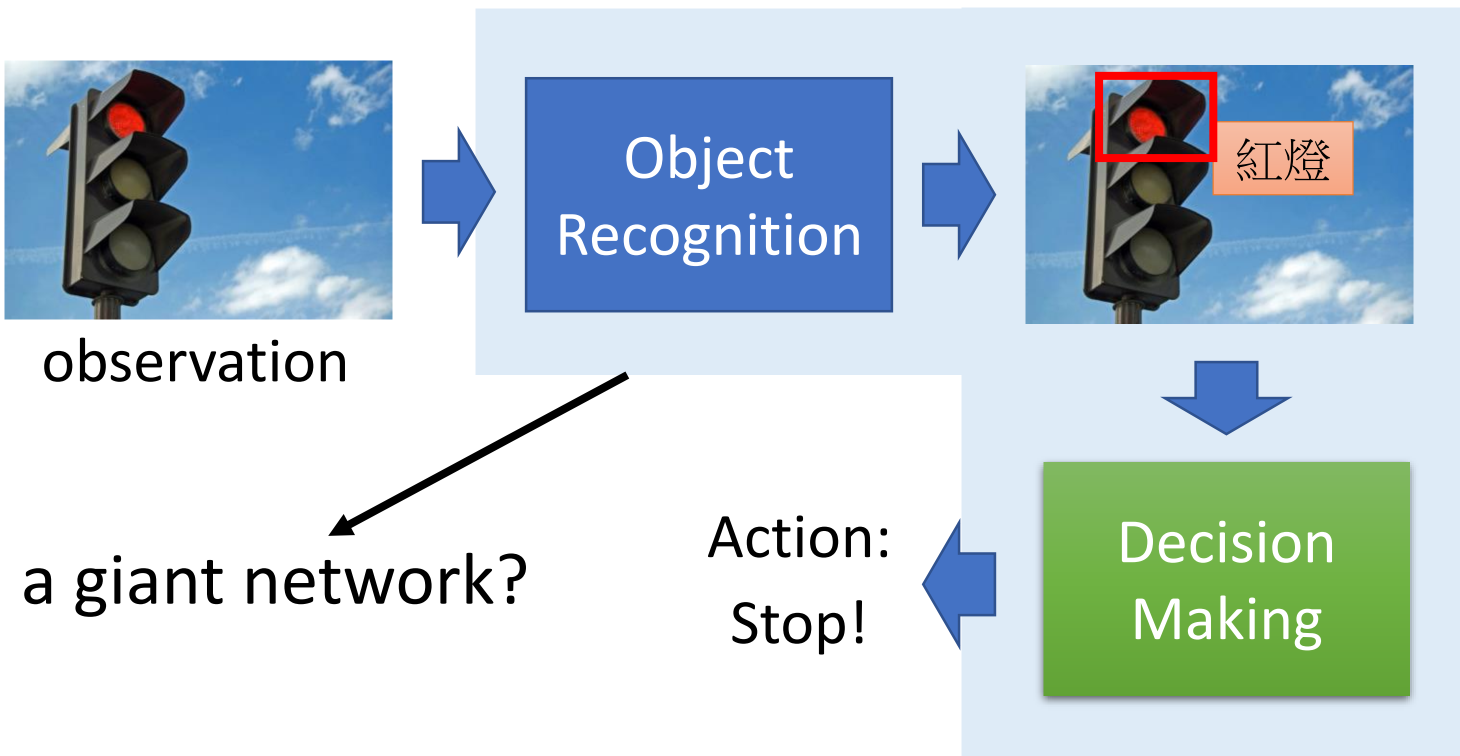
Machine learns to play video games as human players

➢ What machine observes are pixels

➢ Machine learns to take proper action itself

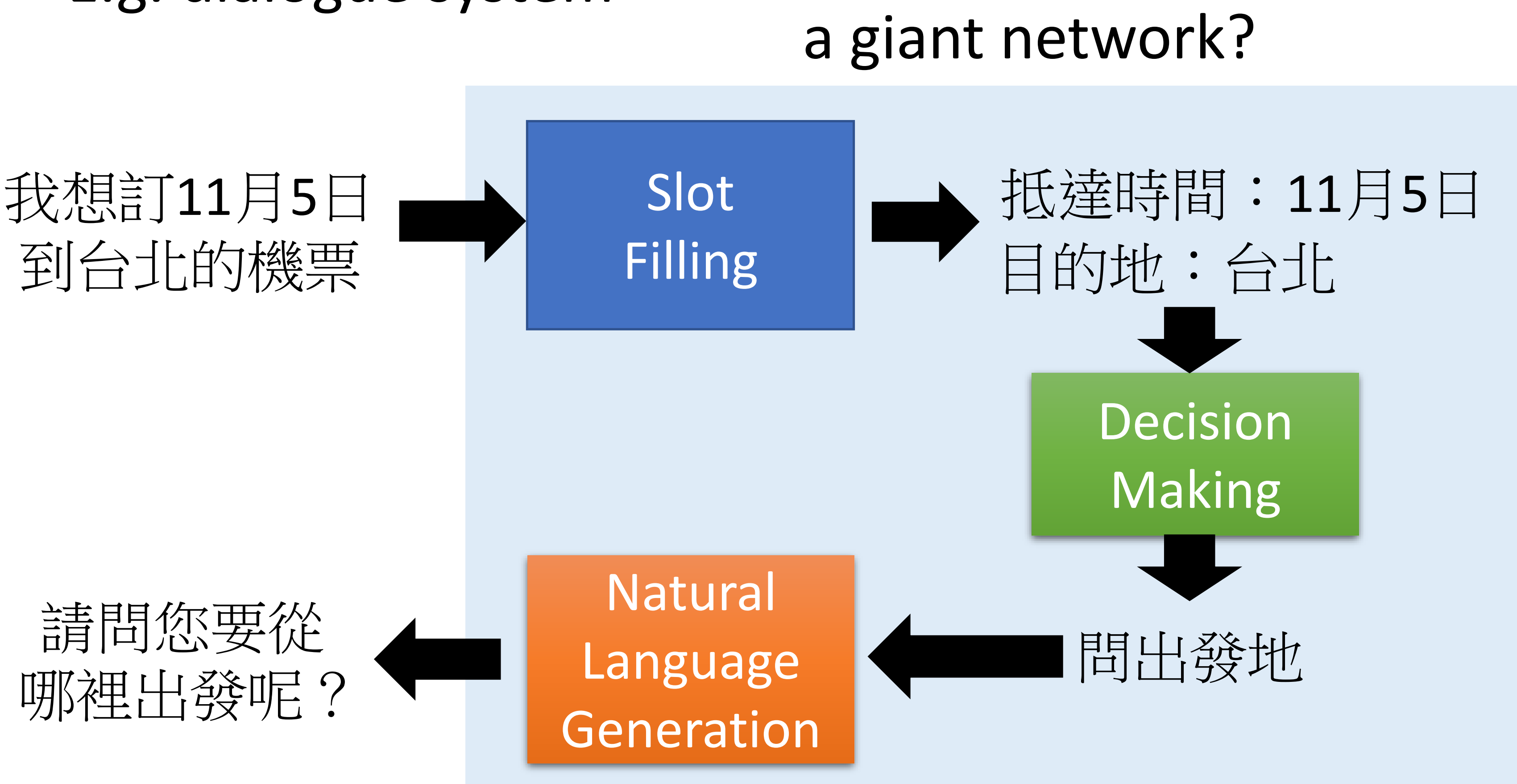
Decision Making and Control

- E.g. self-driving car



Decision Making and Control

- E.g. dialogue system

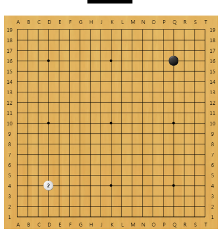


How to solve this problem?

- Network as a function, learn as typical supervised tasks

Target: 3-3

Network



Target: Stop!

Network

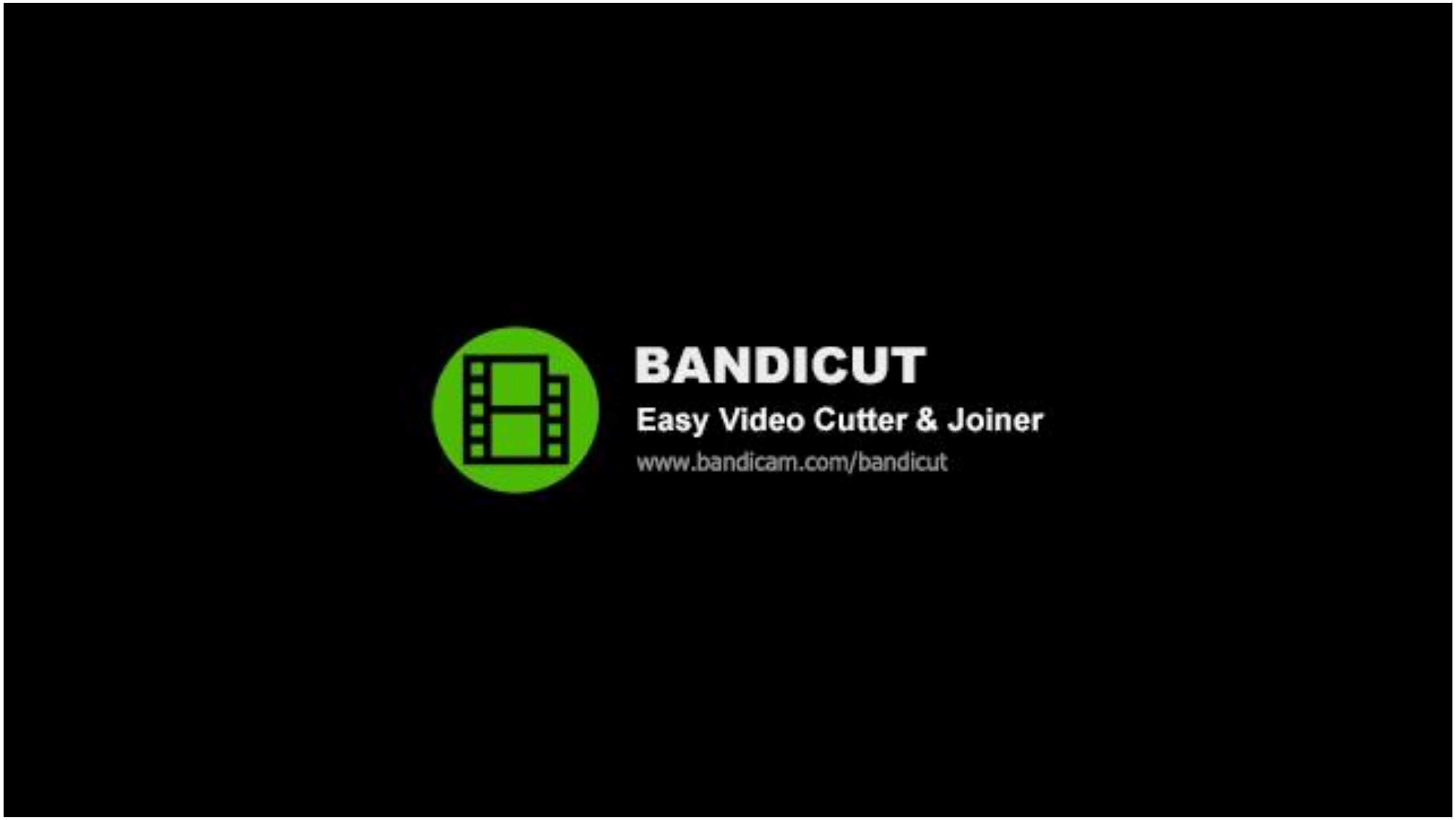
Target: 請問您要從哪裡出發呢？

Network

我想訂11月5日到台北的機票

Behavior Cloning

Machine do not know some behavior must copy, but some can be ignored.



https://www.youtube.com/watch?v=j2FSB3bseek

Properties of Decision Making and Control

What do we miss?

Machine does not know the influence of each action.

- Agent's actions **affect the subsequent data it receives**
- **Reward delay**
  - In space invader, only "fire" obtains reward
    - Although the moving before "fire" is important
  - In Go playing, sacrificing immediate reward to gain more long-term reward

Better Way ……

A sequence of decision

Network

Next move

Two Learning Scenarios

- Scenario 1: Reinforcement Learning
  - Machine interacts with the environment.
  - Machine obtains the reward from the environment, so it knows its performance is good or bad.
- Scenario 2: Learning by demonstration
  - Also known as imitation learning, apprenticeship learning
  - An expert demonstrates how to solve the task, and machine learns from the demonstration.

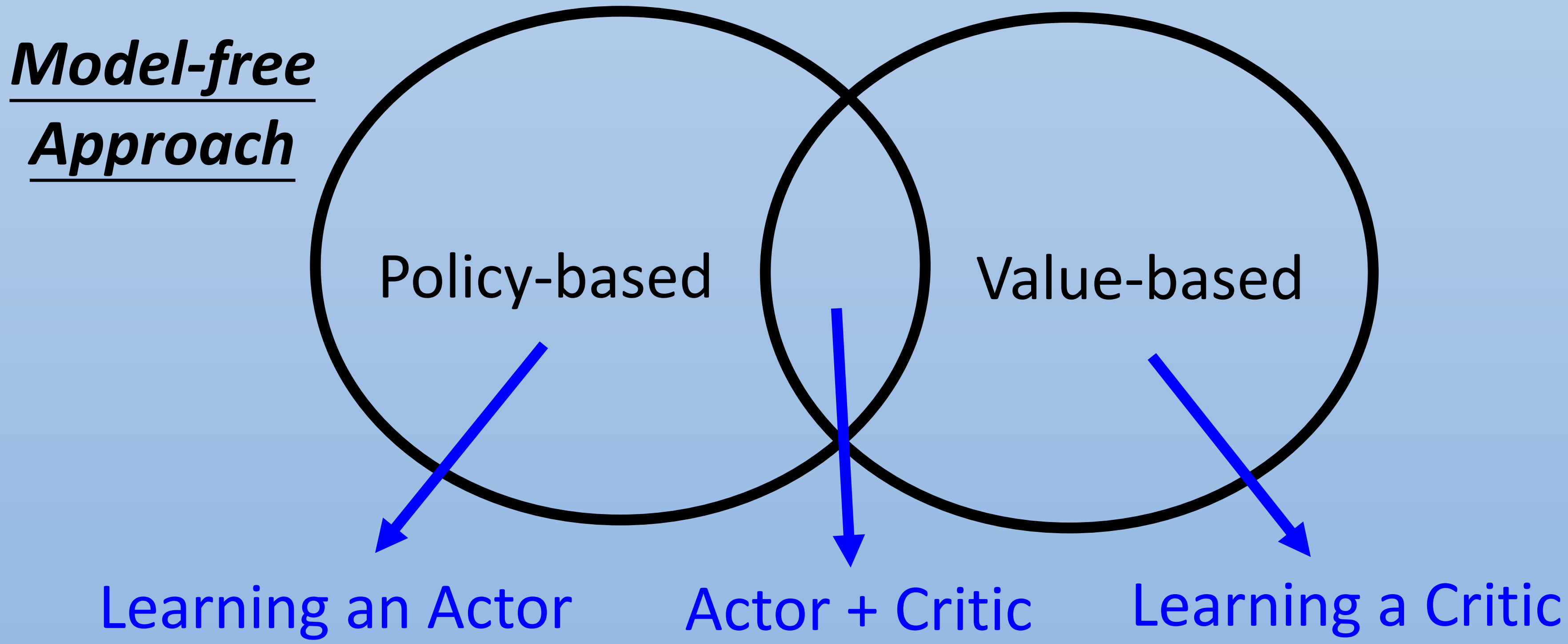
Lecture III

Reinforcement Learning

Inverse Reinforcement Learning

RL

Alpha Go: policy-based + value-based + model-based



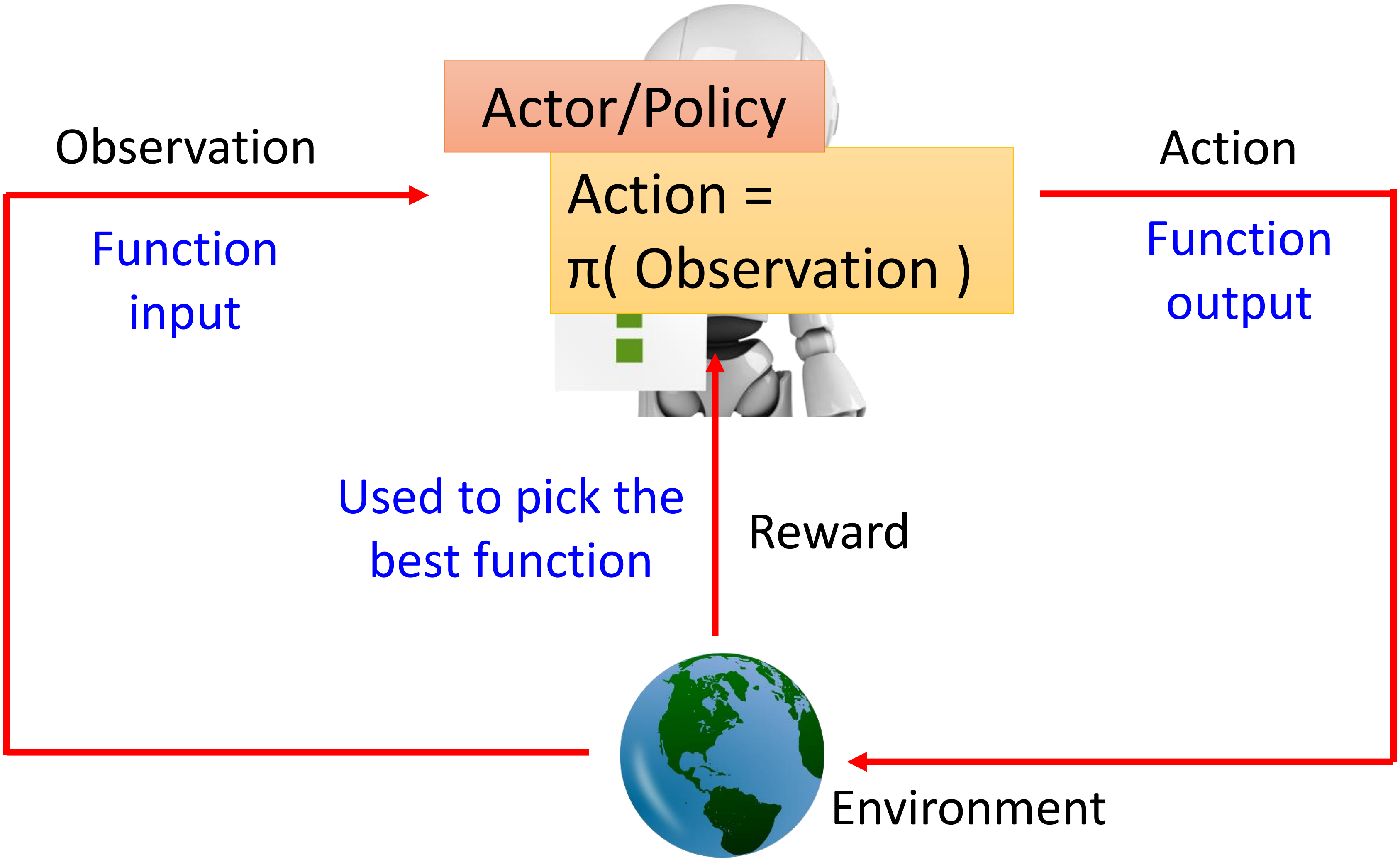
Model-based Approach

Basic Components

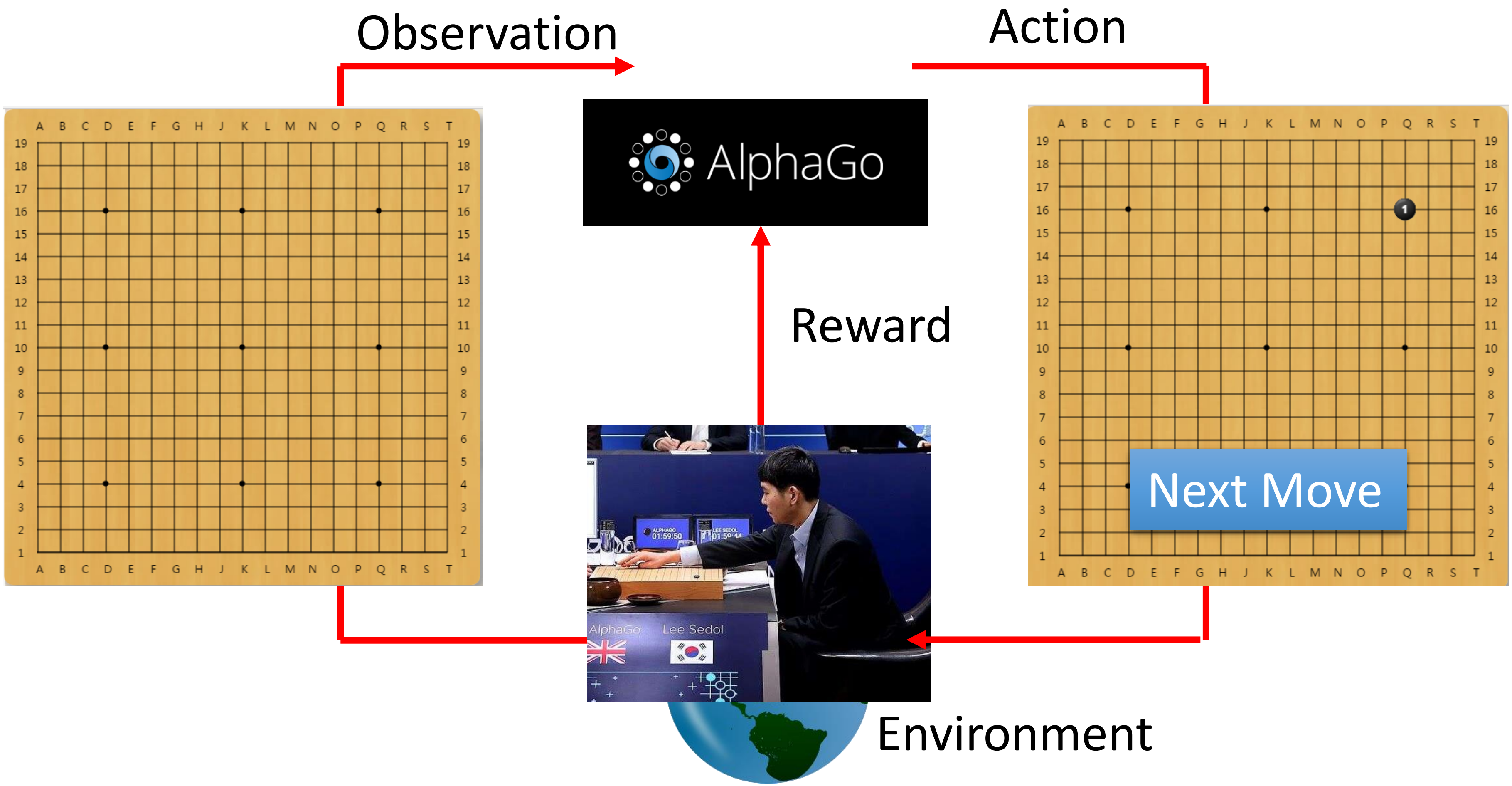
You cannot control

	Actor	Env	Reward Function
Video Game			殺一隻怪得 20 分 …
Go	AlphaGo Google DeepMind	 	圍棋規則

Scenario

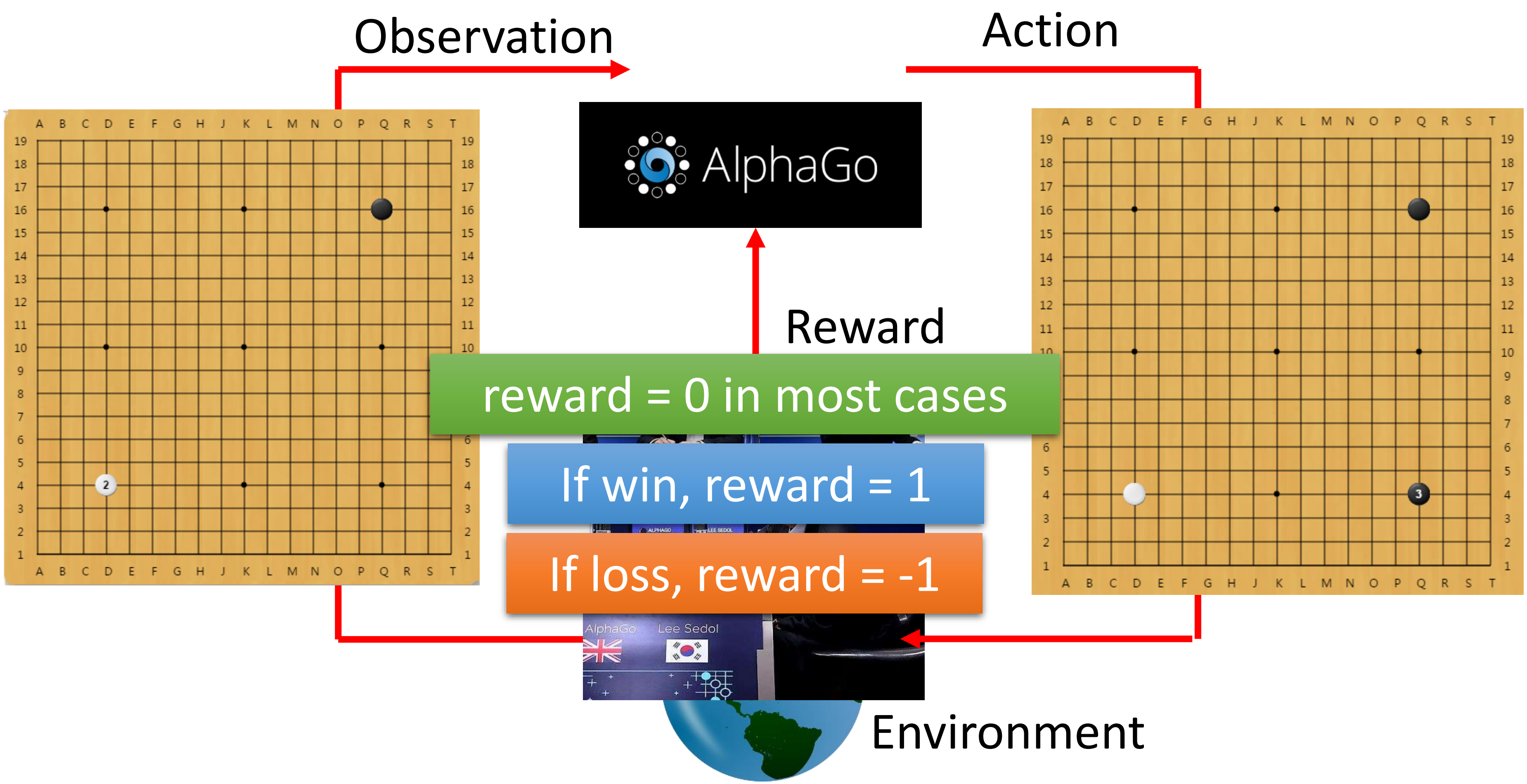

Actor/Policy
Observation
Action =
π(Observation)
Action
Function
input
Function
output
Used to pick the
best function
Reward
Environment

Learning to play Go

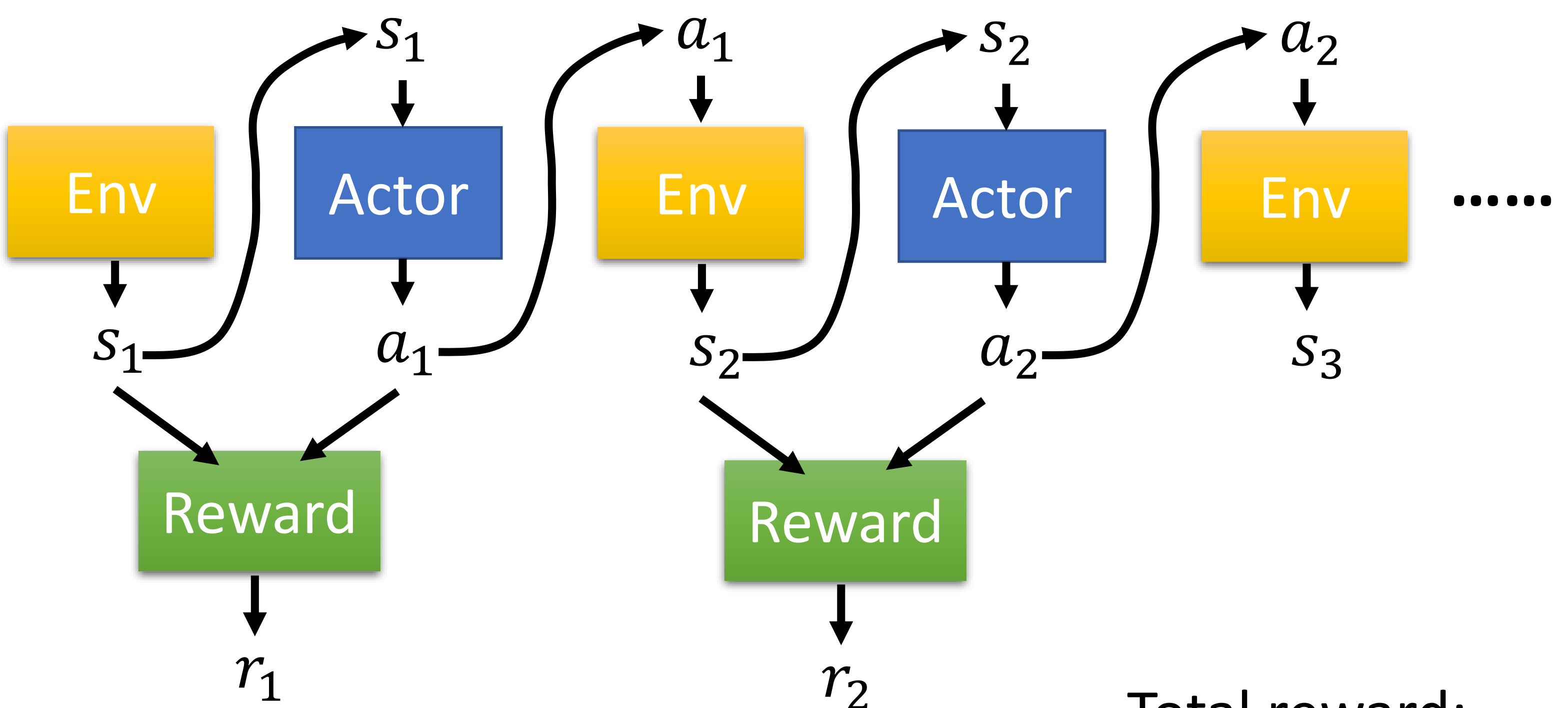


Learning to play Go

Agent learns to take actions maximizing expected reward.



Actor, Environment, Reward



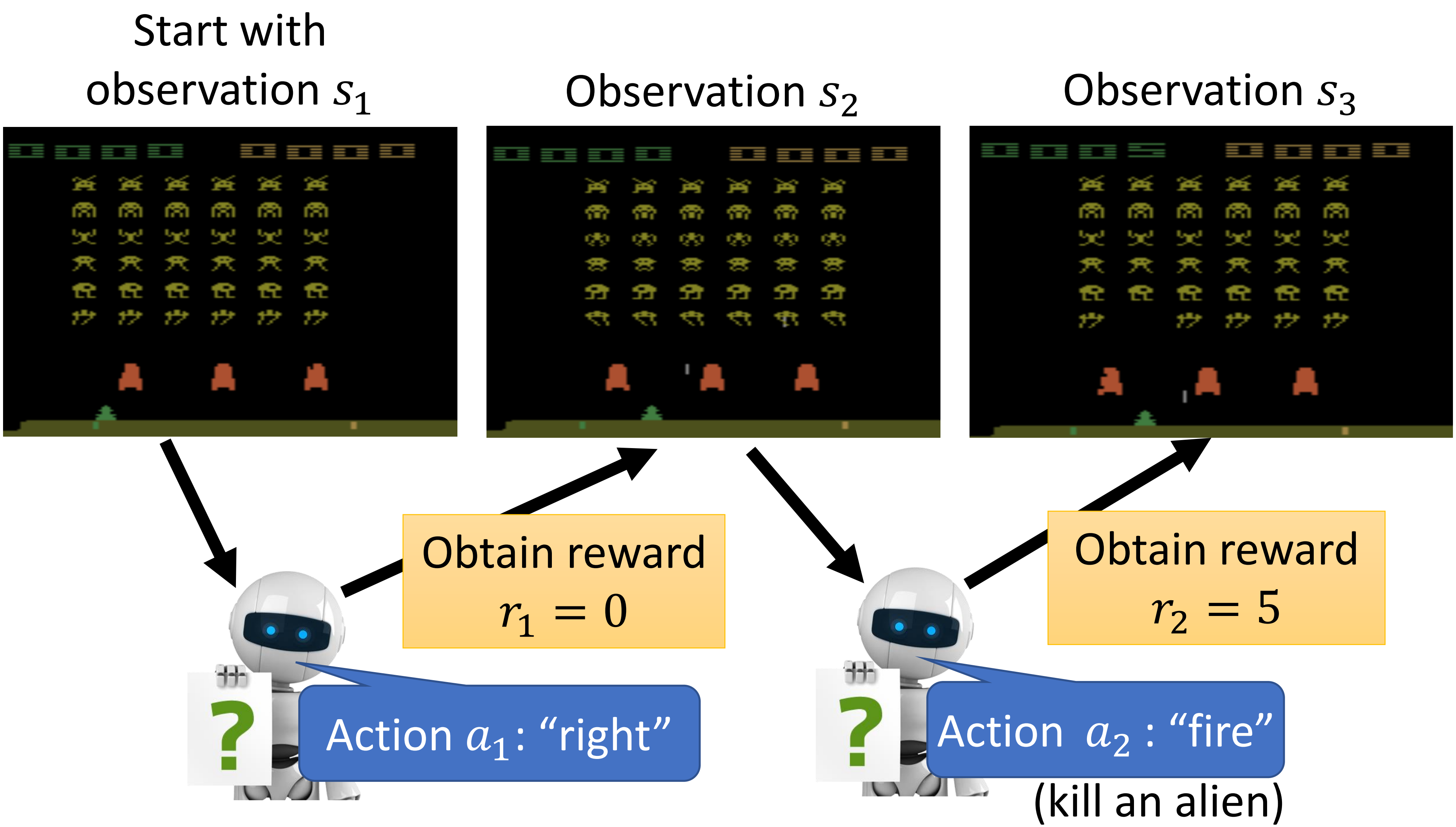
Total reward:

$$R(\tau) = \sum_{t=1}^{T} r_t$$

Trajectory

$$\tau = \{s_1, a_1, s_2, a_2, \cdots, s_T, a_T\}$$

Example: Playing Video Game

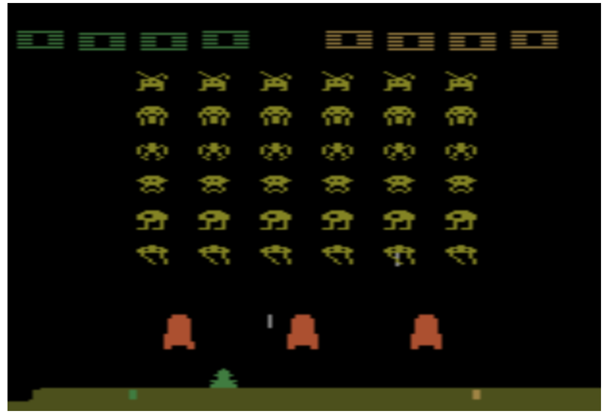


Usually there is some randomness in the environment

Example: Playing Video Game

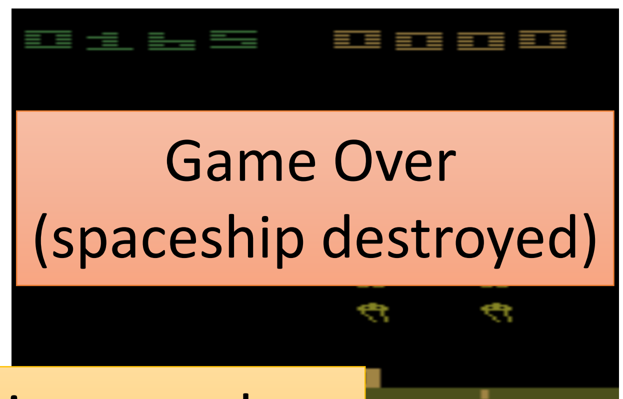
Start with observation s_1

Observation s_2



Observation s_3

After many turns



Obtain reward r_T

Action a_T

This is an ***episode***.

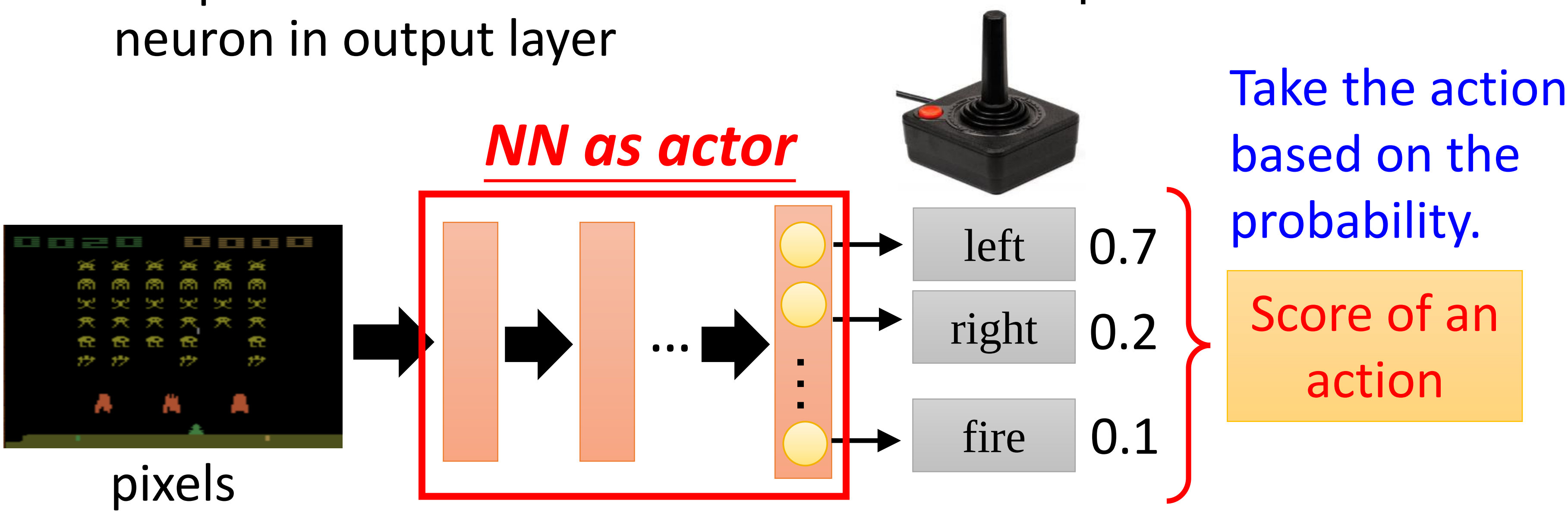
Total reward:

$$R = \sum_{t=1}^{T} r_t$$

We want the total reward be maximized.

Neural network as Actor

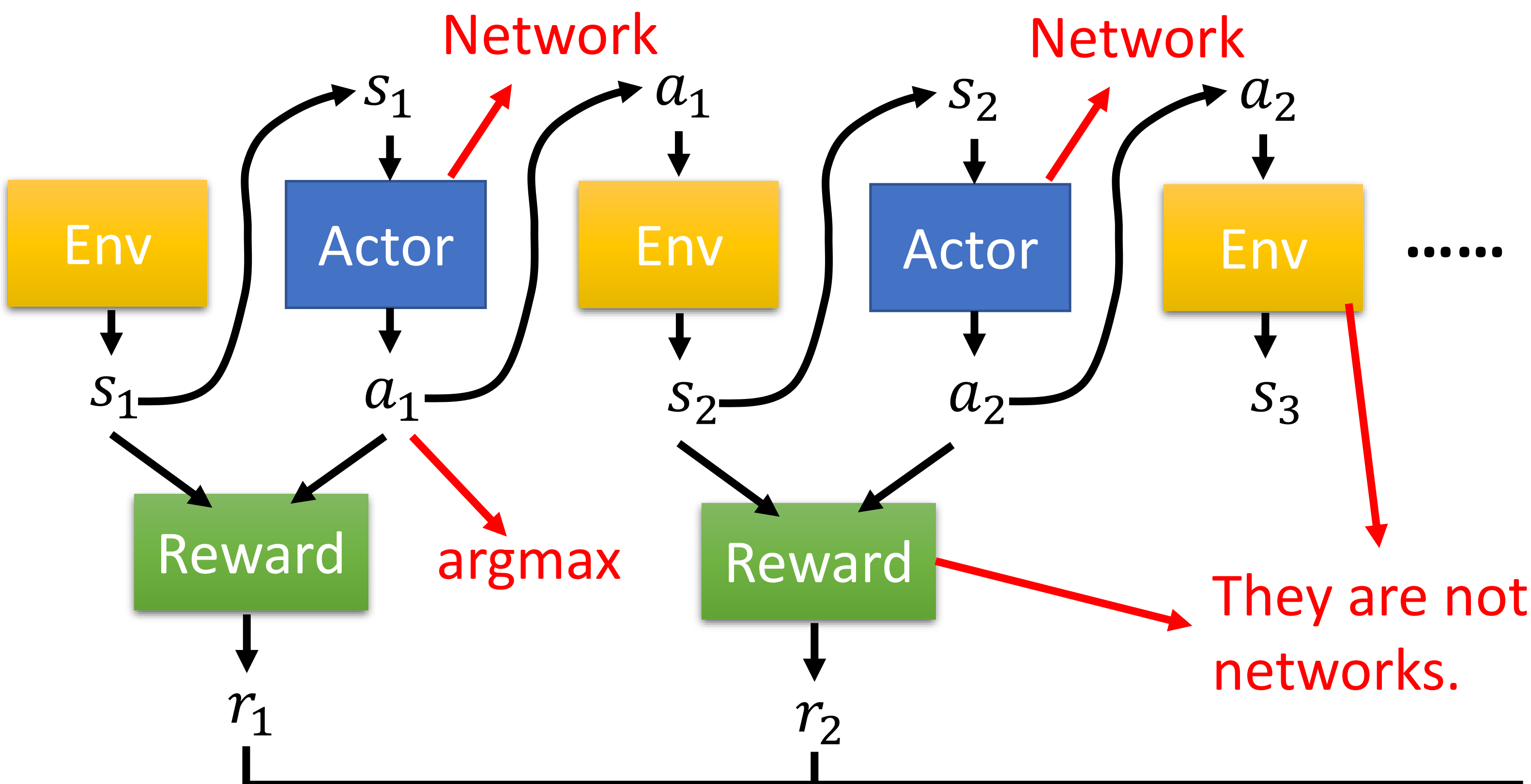
- Input of neural network: the observation of machine represented as a vector or a matrix
- Output neural network : each action corresponds to a neuron in output layer



What is the benefit of using network instead of lookup table?

generalization

Actor, Environment, Reward



如果發現不能微分就用 policy gradient 硬 train 一發

$$R(\tau) = \sum_{t=1}^{T} r_t$$

Ref: https://www.youtube.com/watch?v=W8XF3ME8G2I

Warning of Math

Policy Gradient

Three Steps for Deep Learning

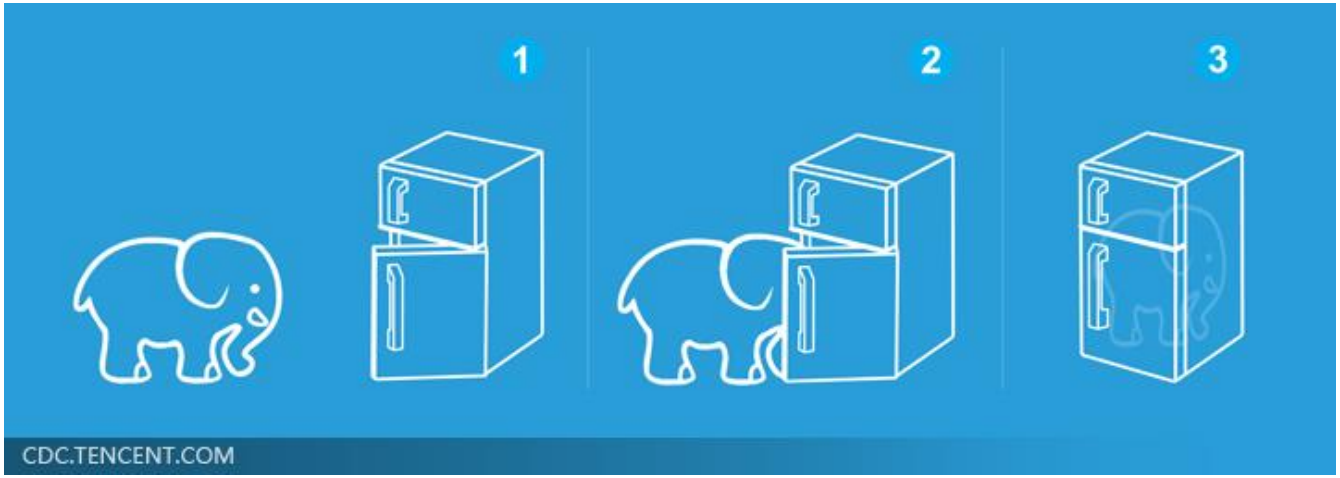
Step 1:

Neural Network as Actor

Step 2: goodness of function

Step 3: pick the best function

Deep Learning is so simple ……



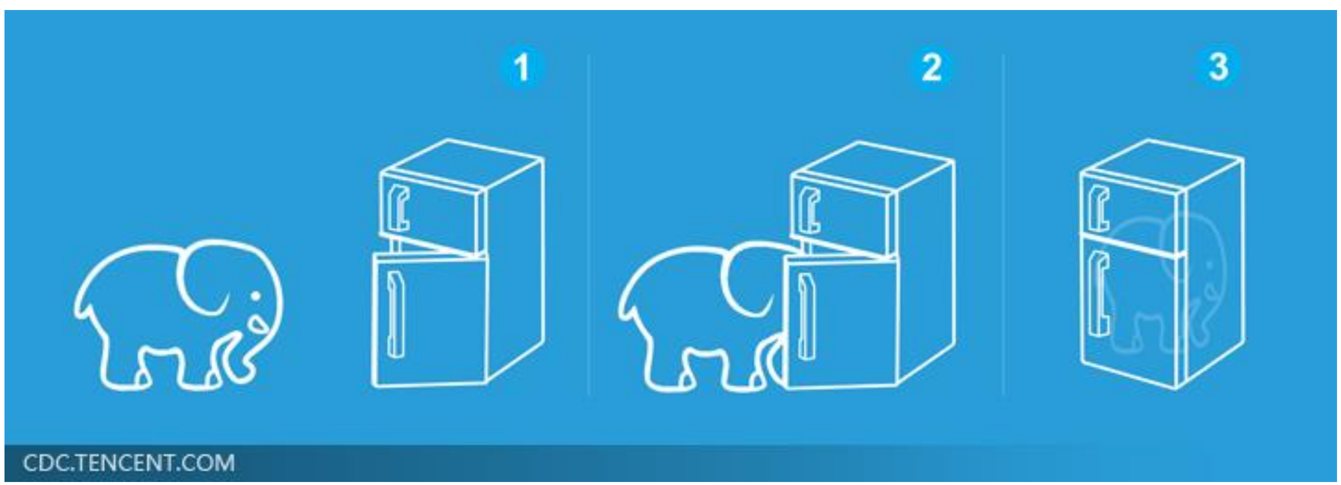
Three Steps for Deep Learning

Step 1: Neural Network as Actor

Step 2: goodness of function

Step 3: pick the best function

Deep Learning is so simple ……



Goodness of Actor

- Given an actor $\pi_\theta(s)$ with network parameter θ
- Use the actor $\pi_\theta(s)$ to play the video game
  - Start with observation s_1
  - Machine decides to take a_1
  - Machine obtains reward r_1
  - Machine sees observation s_2
  - Machine decides to take a_2
  - Machine obtains reward r_2
  - Machine sees observation s_3
  - ......
  - Machine decides to take a_T
  - Machine obtains reward r_T END

Total reward: $R_\theta = \sum_{t=1}^{T} r_t$

Even with the same actor, R_θ is different each time

Randomness in the actor and the game

We define $\bar{R}_\theta$ as the *expected value* of R_θ

$\bar{R}_\theta$ evaluates the goodness of an actor $\pi_\theta(s)$

Goodness of Actor

We define $\bar{R}_\theta$ as the *expected value* of R_θ

- $\tau = \{s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_T, a_T, r_T\}$

$$P(\tau|\theta) =$$

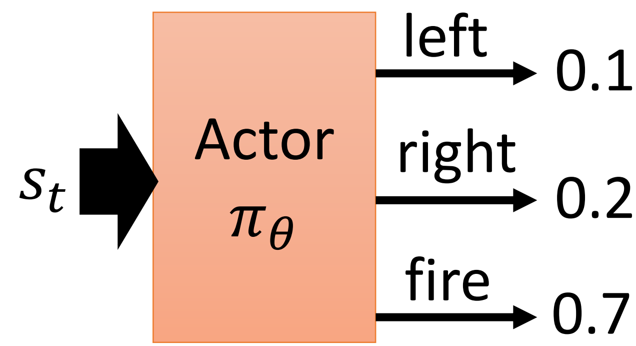
$$p(s_1)p(a_1|s_1,\theta)p(r_1,s_2|s_1,a_1)p(a_2|s_2,\theta)p(r_2,s_3|s_2,a_2)\cdots$$

$$= \underline{p(s_1)} \prod_{t=1}^{T} \underline{p(a_t|s_t,\theta)}\,\underline{p(r_t,s_{t+1}|s_t,a_t)}$$

not related to your actor

Control by your actor π_θ

$$p(a_t = \text{"}fire\text{"}|s_t,\theta) = 0.7$$



Goodness of Actor

- An episode is considered as a trajectory τ
  - $\tau = \{s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_T, a_T, r_T\}$
  - $R(\tau) = \sum_{t=1}^{T} r_t$
  - If you use an actor to play the game, each τ has a probability to be sampled
    - The probability depends on actor parameter θ: $P(\tau|\theta)$

$$\bar{R}_\theta = \boxed{\sum_{\tau}} R(\tau) \boxed{P(\tau|\theta)} \approx \boxed{\frac{1}{N} \sum_{n=1}^{N}} R(\tau^n)$$

Sum over all possible trajectory

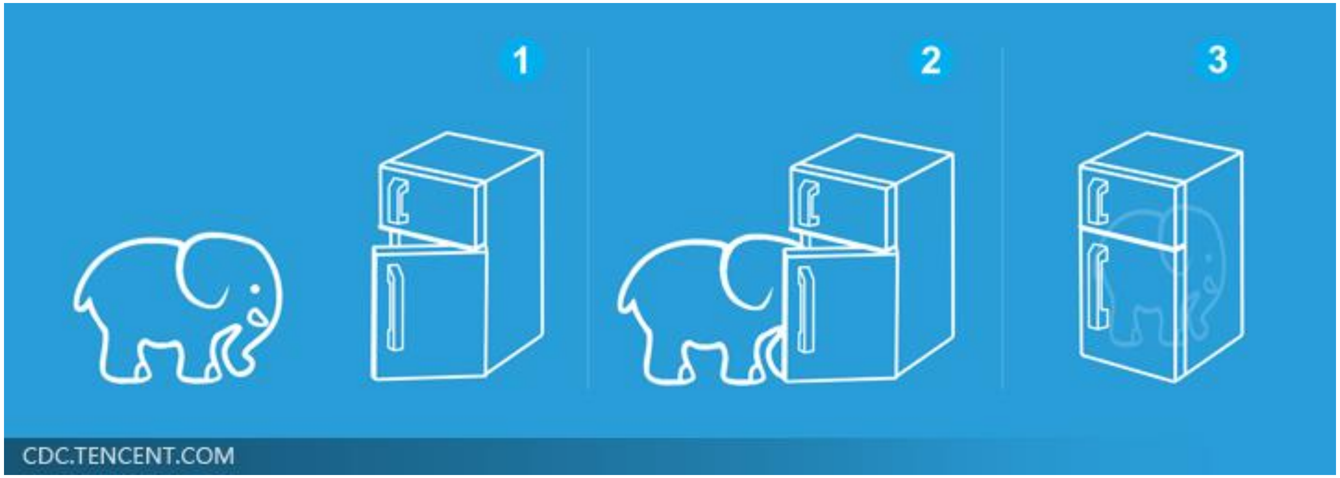
Use π_θ to play the game N times, obtain $\{\tau^1, \tau^2, \cdots, \tau^N\}$

Sampling τ from $P(\tau|\theta)$ N times

Three Steps for Deep Learning

Step 1: Neural Network as Actor → Step 2: goodness of function → Step 3: pick the best function

Deep Learning is so simple ……



Gradient Ascent

- Problem statement

$$\theta^* = arg \max_\theta \bar{R}_\theta$$

- Gradient ascent
  - Start with θ^0
  - $\theta^1 \leftarrow \theta^0 + \eta \nabla \bar{R}_{\theta^0}$
  - $\theta^2 \leftarrow \theta^1 + \eta \nabla \bar{R}_{\theta^1}$
  - ……

$$\theta = \{w_1, w_2, \cdots, b_1, \cdots\}$$

$$\nabla \bar{R}_\theta = \begin{bmatrix} \partial \bar{R}_\theta / \partial w_1 \\ \partial \bar{R}_\theta / \partial w_2 \\ \vdots \\ \partial \bar{R}_\theta / \partial b_1 \\ \vdots \end{bmatrix}$$

Policy Gradient

$$\bar{R}_\theta = \sum_\tau R(\tau)P(\tau|\theta) \quad \nabla\bar{R}_\theta = ?$$

$$\nabla\bar{R}_\theta = \sum_\tau R(\tau)\nabla P(\tau|\theta) = \sum_\tau R(\tau)P(\tau|\theta)\frac{\nabla P(\tau|\theta)}{P(\tau|\theta)}$$

$R(\tau)$ do not have to be differentiable

It can even be a black box.

$$= \boxed{\sum_\tau} R(\tau)\boxed{P(\tau|\theta)}\nabla log P(\tau|\theta)$$

$$\boxed{\frac{dlog(f(x))}{dx} = \frac{1}{f(x)}\frac{df(x)}{dx}}$$

$$\approx \frac{1}{N}\sum_{n=1}^{N} R(\tau^n)\underline{\nabla log P(\tau^n|\theta)}$$

Use π_θ to play the game N times,
Obtain $\{\tau^1, \tau^2, \cdots, \tau^N\}$

Policy Gradient

$$\nabla log P(\tau|\theta) = ?$$

- $\tau = \{s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_T, a_T, r_T\}$

$$P(\tau|\theta) = p(s_1) \prod_{t=1}^{T} p(a_t|s_t, \theta) p(r_t, s_{t+1}|s_t, a_t)$$

$$log P(\tau|\theta)$$

$$= log p(s_1) + \sum_{t=1}^{T} log p(a_t|s_t, \theta) + log p(r_t, s_{t+1}|s_t, a_t)$$

$$\nabla log P(\tau|\theta) = \sum_{t=1}^{T} \nabla log p(a_t|s_t, \theta)$$

Ignore the terms not related to θ

Policy Gradient

$$\nabla logP(\tau|\theta) = \sum_{t=1}^{T} \nabla logp(a_t|s_t,\theta)$$

$$\theta^{new} \leftarrow \theta^{old} + \eta \nabla \bar{R}_{\theta^{old}}$$

$$\nabla \bar{R}_\theta \approx \frac{1}{N}\sum_{n=1}^{N} R(\tau^n)\nabla logP(\tau^n|\theta) = \frac{1}{N}\sum_{n=1}^{N} R(\tau^n)\sum_{t=1}^{T_n} \nabla logp(a_t^n|s_t^n,\theta)$$

$$= \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} R(\tau^n)\nabla logp(a_t^n|s_t^n,\theta)$$

What if we replace $R(\tau^n)$ with r_t^n ……

If in τ^n machine takes a_t^n when seeing s_t^n in

$R(\tau^n)$ is positive → Tuning θ to increase $p(a_t^n|s_t^n)$

$R(\tau^n)$ is negative → Tuning θ to decrease $p(a_t^n|s_t^n)$

It is very important to consider the cumulative reward $R(\tau^n)$ of the whole trajectory τ^n instead of immediate reward r_t^n

Policy Gradient

Given actor parameter θ

τ^1: (s_1^1, a_1^1) $R(\tau^1)$

(s_2^1, a_2^1) $R(\tau^1)$

$\vdots$ $\vdots$

τ^2: (s_1^2, a_1^2) $R(\tau^2)$

(s_2^2, a_2^2) $R(\tau^2)$

$\vdots$ $\vdots$

Update Model

$$\theta \leftarrow \theta + \eta \nabla \bar{R}_\theta$$

$$\nabla \bar{R}_\theta =$$

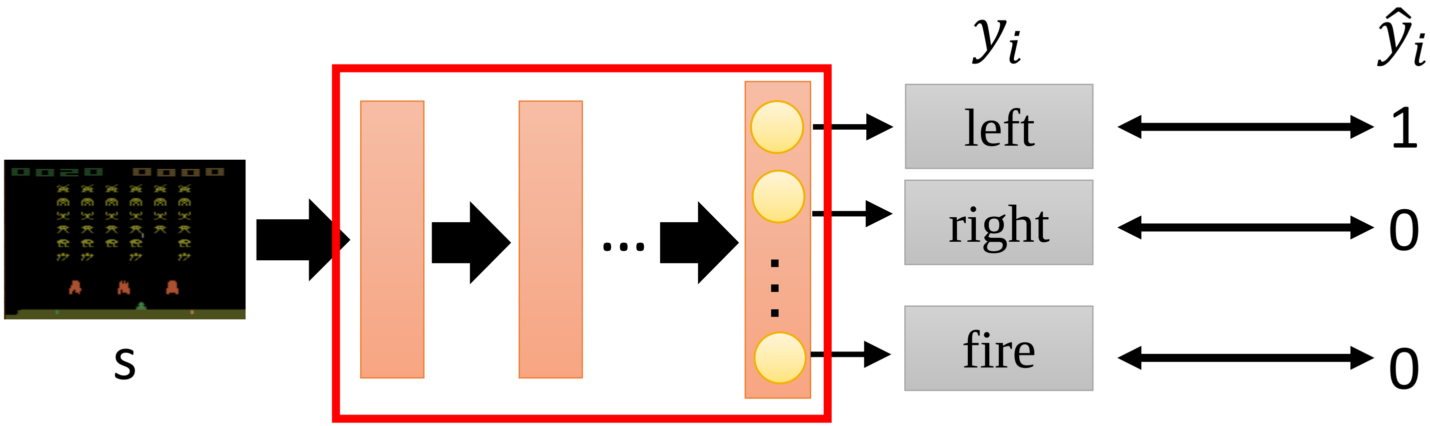
$$\frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n} R(\tau^n) \nabla log p(a_t^n | s_t^n, \theta)$$

Data Collection

Policy Gradient

Considered as Classification Problem

Minimize: $-\sum_{i=1}^{3} \hat{y}_i log y_i$



Maximize: $log y_i =$

$logP("left"|s)$

$$\theta \leftarrow \theta + \eta \nabla logP("left"|s)$$

Policy Gradient

$$\theta \leftarrow \theta + \eta \nabla \bar{R}_\theta$$

$$\nabla \bar{R}_\theta =$$

$$\frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \boxed{\quad\quad} \nabla logp(a_t^n | s_t^n, \theta)$$

Given actor parameter θ

τ^1: (s_1^1, a_1^1) $R(\tau^1)$

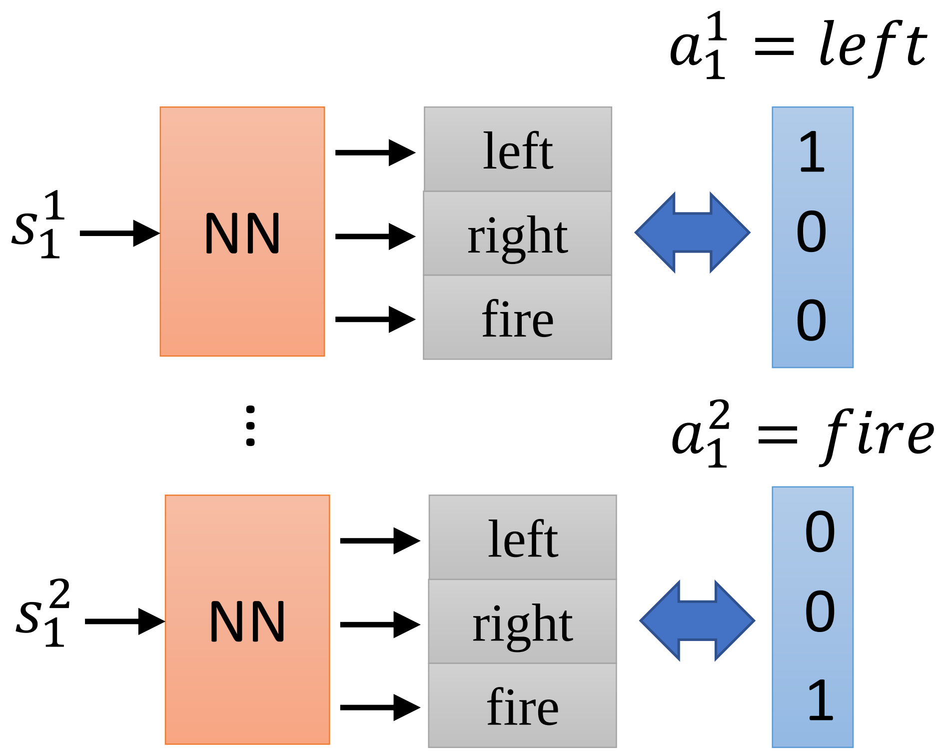
(s_2^1, a_2^1) $R(\tau^1)$

⋮ ⋮

τ^2: (s_1^2, a_1^2) $R(\tau^2)$

(s_2^2, a_2^2) $R(\tau^2)$

⋮ ⋮



Policy Gradient

$$\theta \leftarrow \theta + \eta \nabla \bar{R}_\theta$$

$$\nabla \bar{R}_\theta =$$

$$\frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n} R(\tau^n) \nabla logp(a_t^n | s_t^n, \theta)$$

Given actor parameter θ

τ^1: (s_1^1, a_1^1) $R(\tau^1)$ **2**

(s_2^1, a_2^1) $R(\tau^1)$ **2**

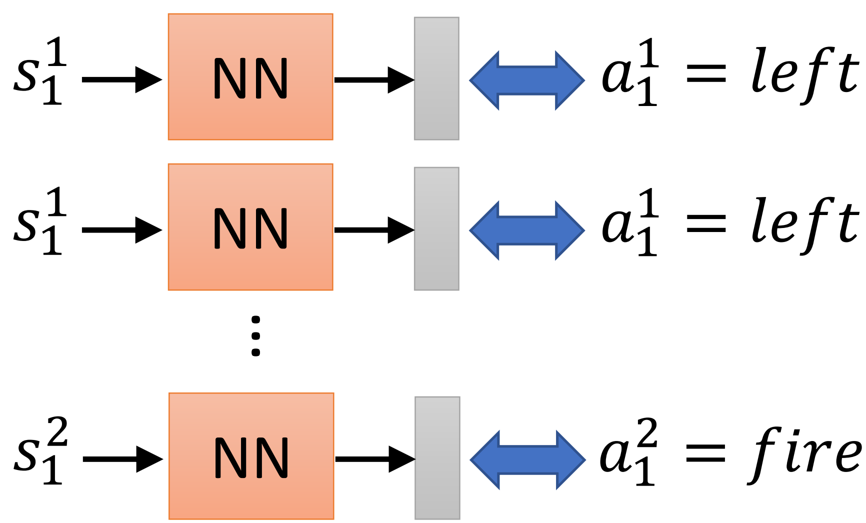
⋮ ⋮

τ^2: (s_1^2, a_1^2) $R(\tau^2)$ **1**

(s_2^2, a_2^2) $R(\tau^2)$ **1**

⋮ ⋮

Each training data is weighted by $R(\tau^n)$



End of Warning

Critic

- A critic does not determine the action.
- Given an actor π, it evaluates the how good the actor is

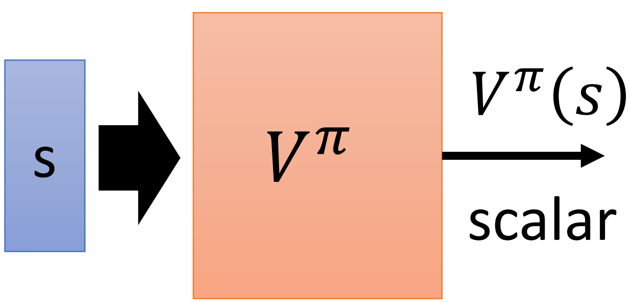
An actor can be found from a critic.

e.g. Q-learning

http://combiboilersleeds.com/picaso/critics/critics-4.html

Critic

- State value function $V^{\pi}(s)$
  - When using actor π, the *cumulated* reward expects to be obtained after seeing observation (state) s



$V^{\pi}(s)$ is large

$V^{\pi}(s)$ is smaller

Critic

$V^{以前的阿光}(大馬步飛) = \text{bad}$

$V^{變強的阿光}(大馬步飛) = \text{good}$

How to estimate $V^\pi(s)$

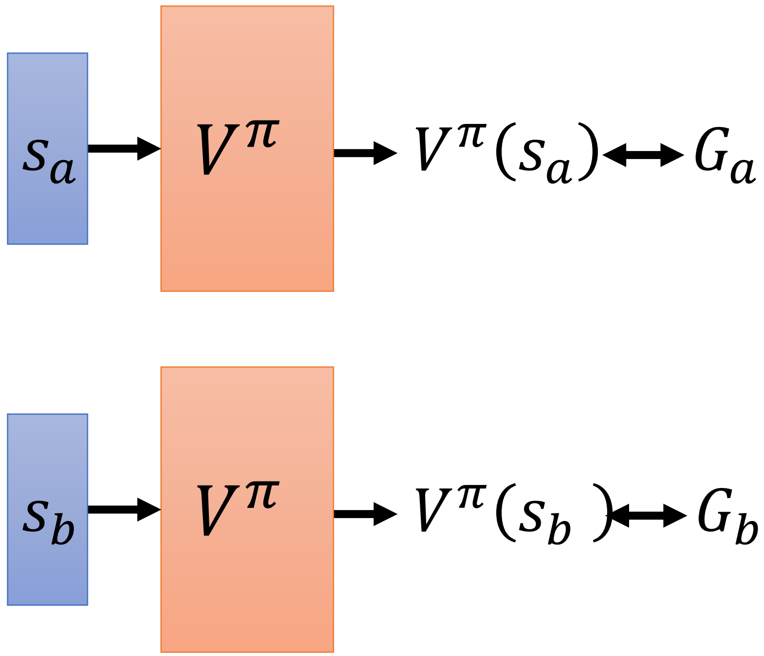
- Monte-Carlo based approach
  - The critic watches π playing the game

After seeing s_a,

Until the end of the episode, the cumulated reward is G_a

After seeing s_b,

Until the end of the episode, the cumulated reward is G_b

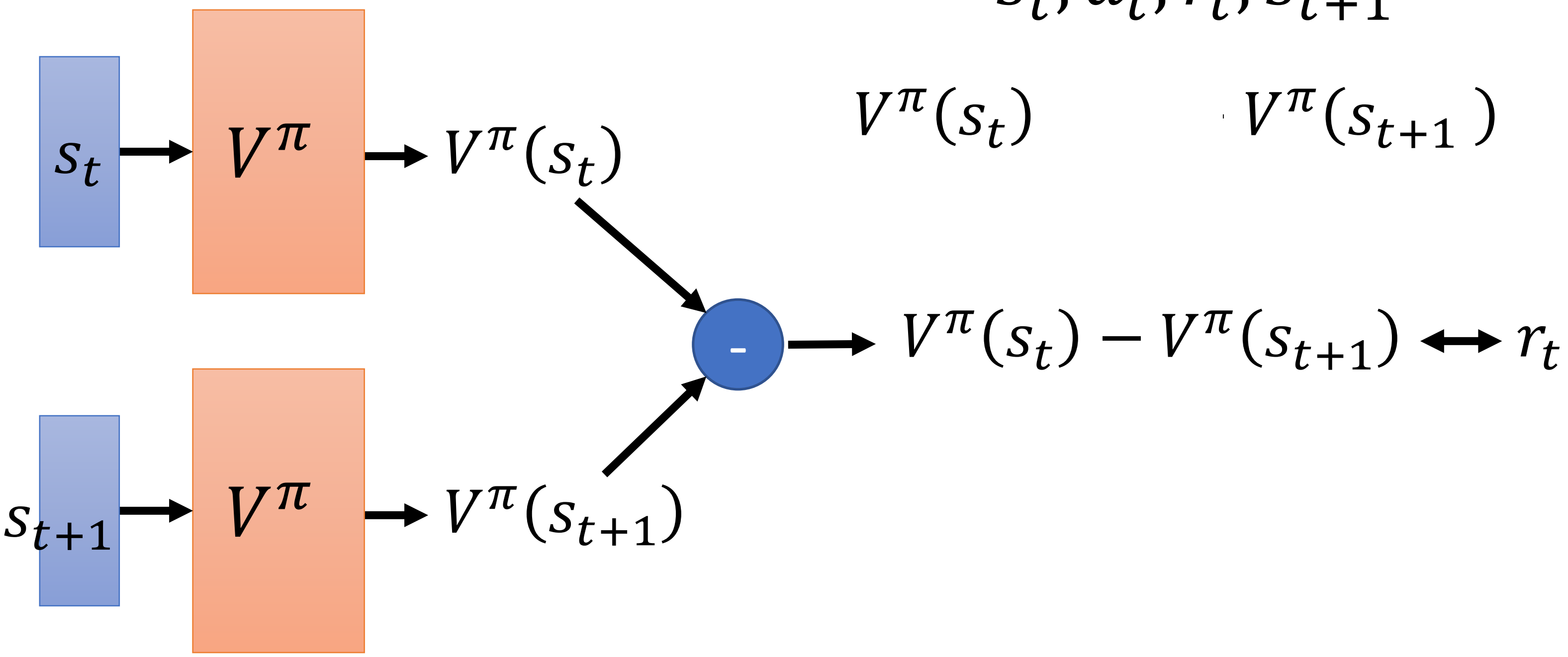


How to estimate $V^\pi(s)$

- Temporal-difference approach

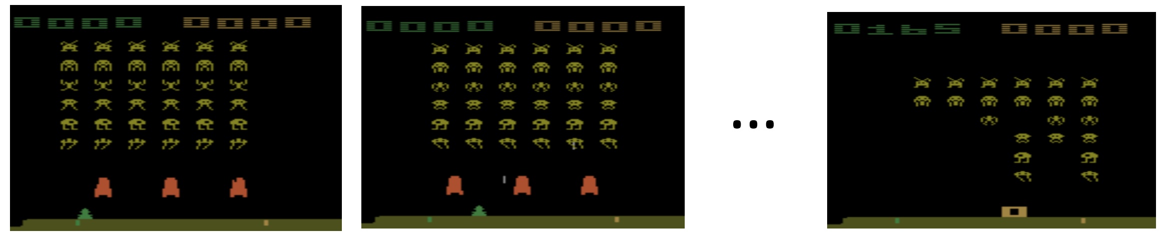
$$\cdots s_t, a_t, r_t, s_{t+1} \cdots$$

$V^\pi(s_t)$ $V^\pi(s_{t+1})$



Some applications have very long episodes, so that delaying all learning until an episode's end is too slow.

MC v.s. TD



$$s_a \rightarrow V^\pi \rightarrow V^\pi(s_a) \leftrightarrow G_a$$

Larger variance

unbiased

$$s_t \rightarrow V^\pi \rightarrow V^\pi(s_t) \leftrightarrow r + V^\pi(s_{t+1}) \leftarrow V^\pi \leftarrow s_{t+1}$$

Smaller variance

May be biased

MC v.s. TD

[Sutton, v2, Example 6.4]

- The critic has the following 8 episodes
  - $s_a, r = 0, s_b, r = 0$, END
  - $s_b, r = 1$, END
  - $s_b, r = 1$, END
  - $s_b, r = 1$, END
  - $s_b, r = 1$, END
  - $s_b, r = 1$, END
  - $s_b, r = 1$, END
  - $s_b, r = 0$, END

$$V^\pi(s_b) = 3/4$$

$V^\pi(s_a) =?$ 0? 3/4?

Monte-Carlo: $V^\pi(s_a) = 0$

Temporal-difference:

$$V^\pi(s_b) + r = V^\pi(s_a)$$

3/4 0 3/4

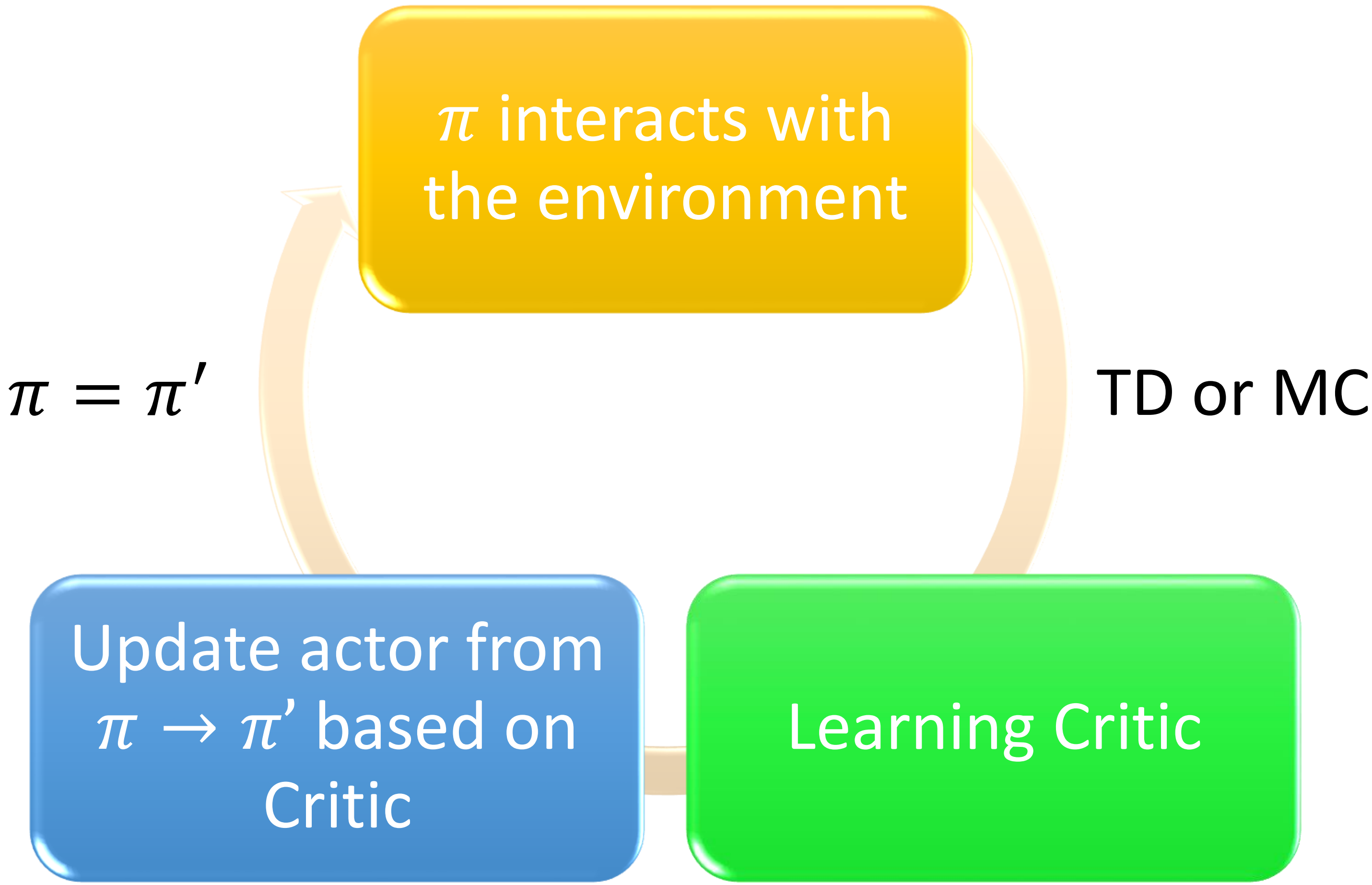
(The actions are ignored here.)

Actor-Critic

Actor = Generator ?

Critic = Discriminator ?

https://arxiv.org/abs/1610.01945



Playing On-line Game

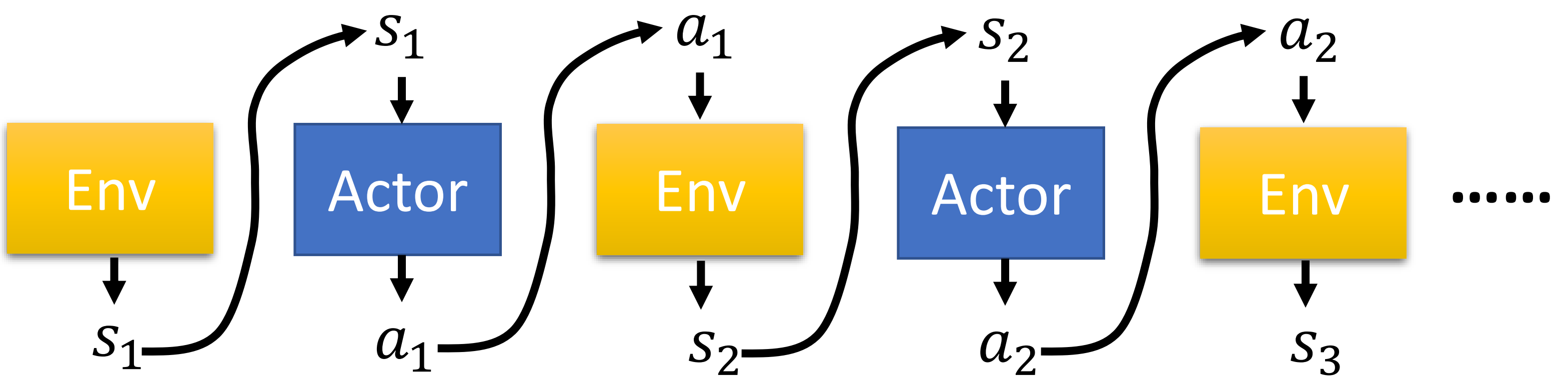
劉廷緯、溫明浩
https://www.youtube.com/watch?v=8iRD1w73fDo&feature=youtu.be

Lecture III

Reinforcement Learning

Inverse Reinforcement Learning

Imitation Learning

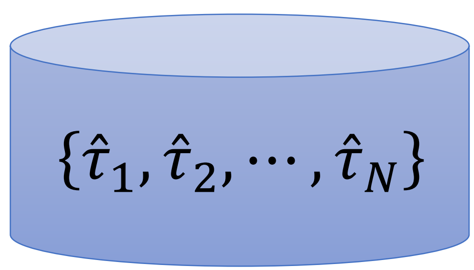


reward function is not available

We have demonstration of the expert.

Self driving: record human drivers

Robot: grab the arm of robot



Each $\hat{\tau}$ is a trajectory of the export.

Motivation

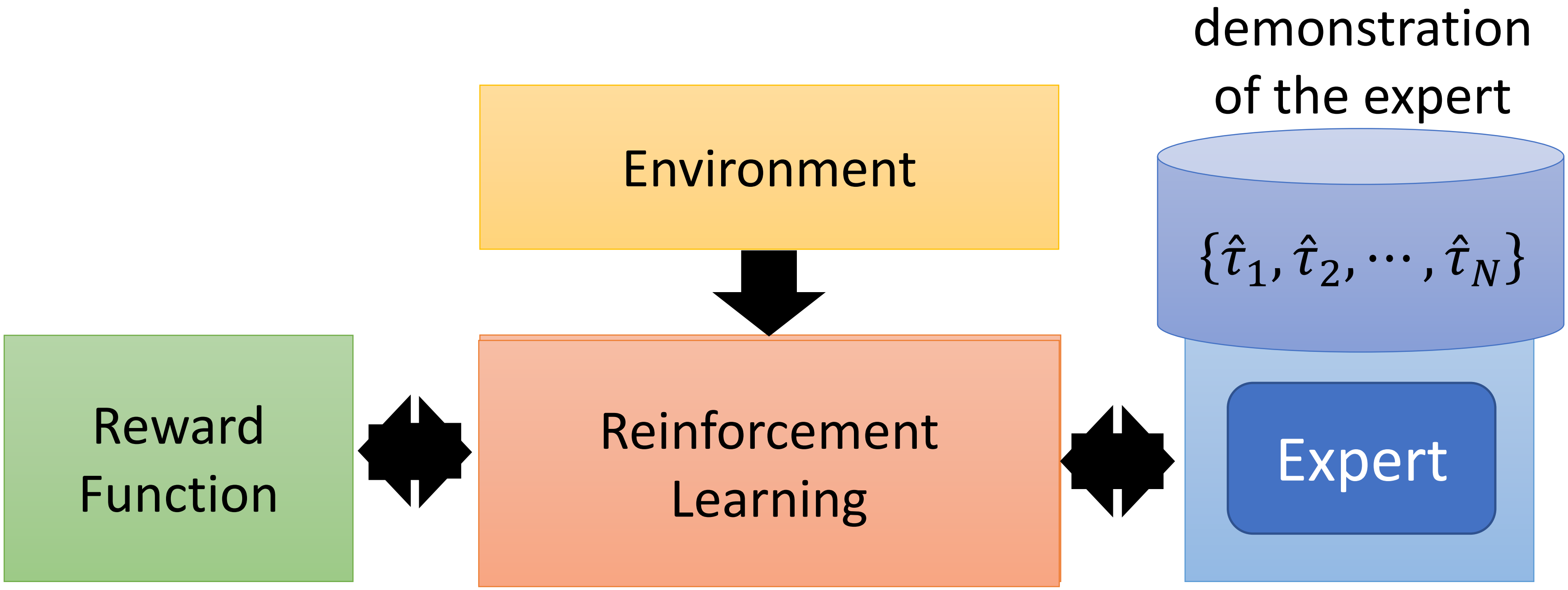
- It is hard to define reward in some tasks.
  - Hand-crafted rewards can lead to uncontrolled behavior.

機器人三大法則：
一、機器人不得傷害人類，或坐視人類受到傷害而袖手旁觀。
二、除非違背第一法則，機器人必須服從人類的命令。
三、在不違背第一法則及第二法則的情況下，機器人必須保護自己。

神邏輯

因此為了保護人類整體，控制人類的自由是必須的

Inverse Reinforcement Learning



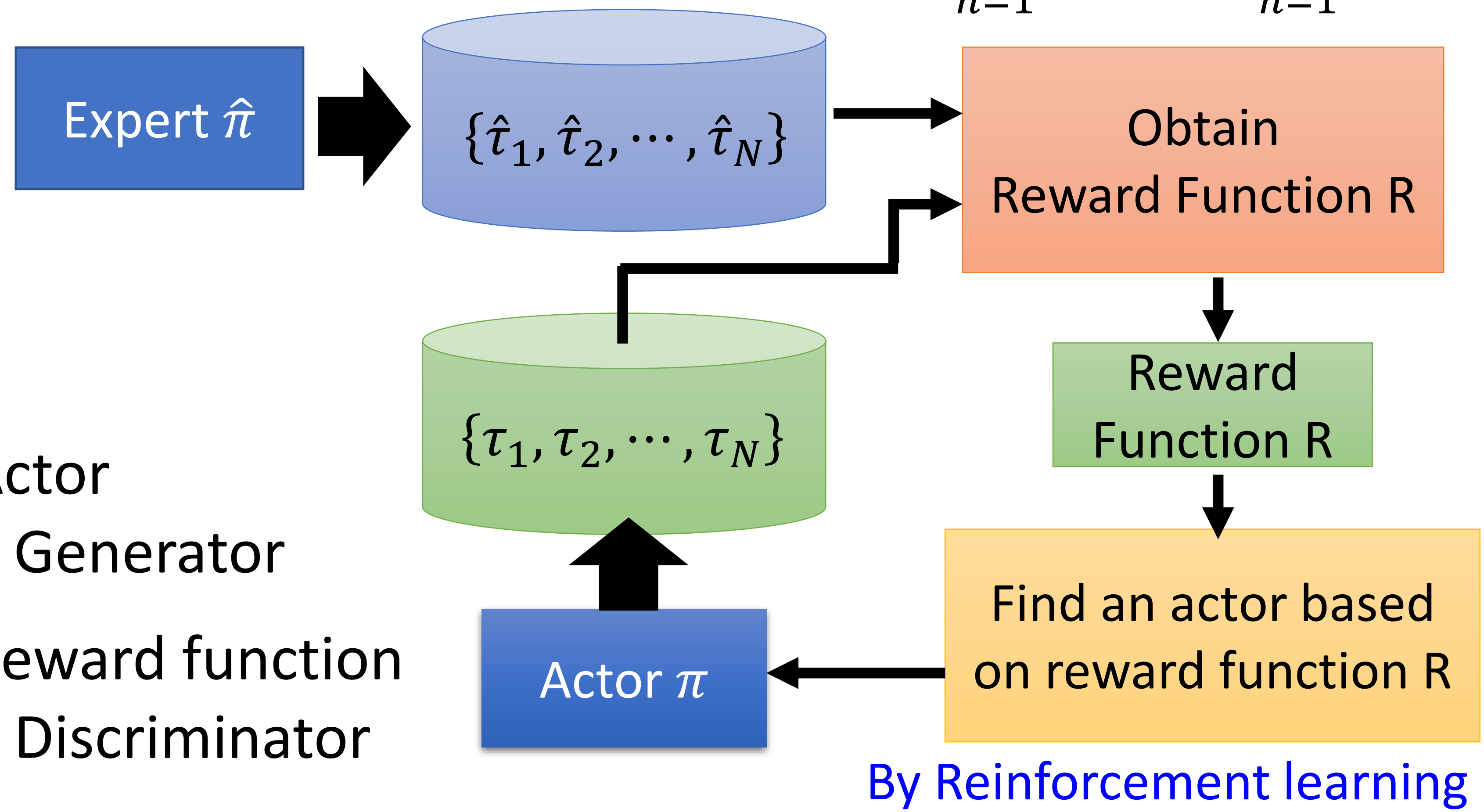
- Using the reward function to find the ***optimal actor***.
- Modeling reward can be easier. Simple reward function can lead to complex policy.

Inverse Reinforcement Learning

- Principle: ***The teacher is always the best***.
- Basic idea:
  - Initialize an actor
  - In each iteration
    - The actor interacts with the environments to obtain some trajectories
    - Define a reward function, which makes the trajectories of the teacher better than the actor
    - The actor learns to maximize the reward based on the new reward function.
  - Output the reward function and the actor learned from the reward function

Framework of IRL

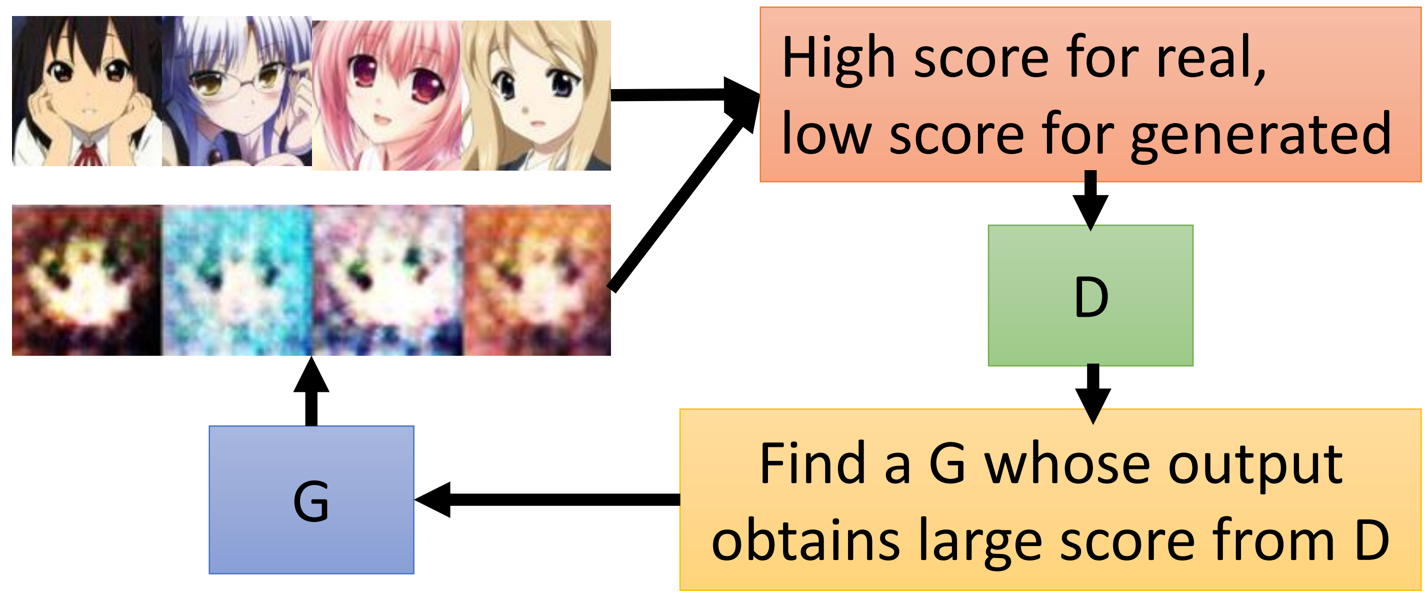
$$\sum_{n=1}^{N} R(\hat{\tau}_n) > \sum_{n=1}^{N} R(\tau)$$



Actor
= Generator

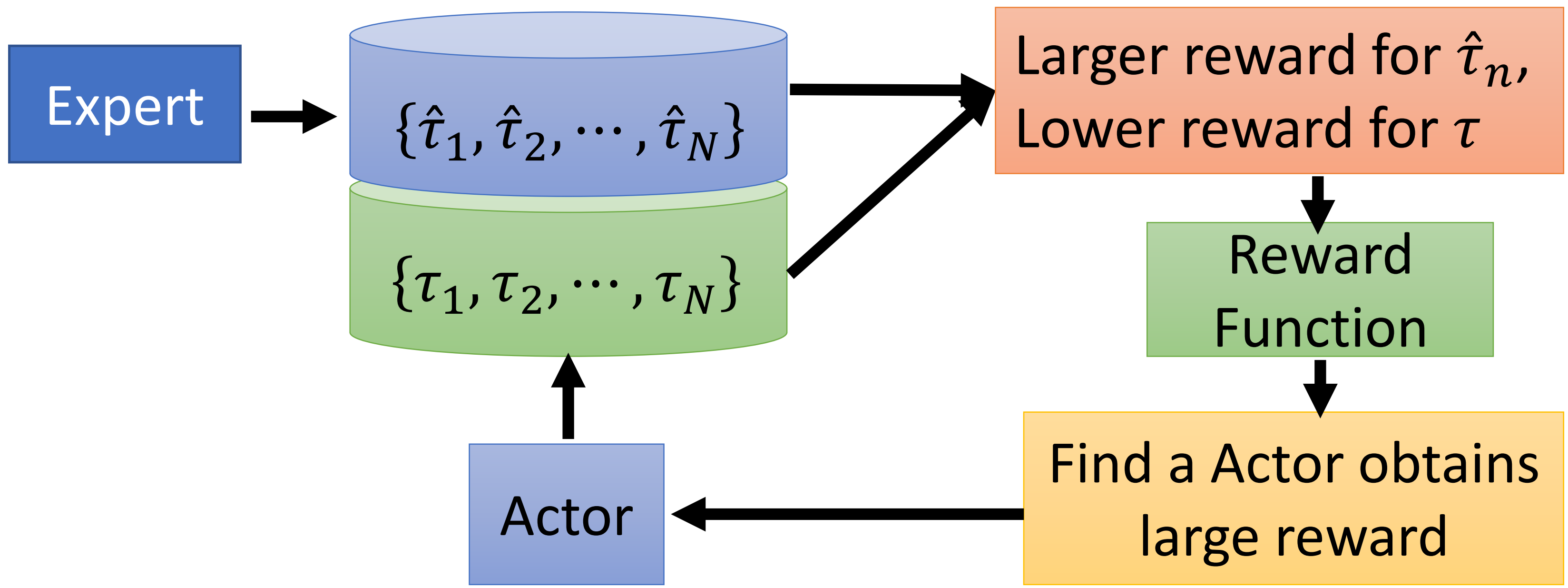
Reward function
= Discriminator

GAN



High score for real,
low score for generated
D
Find a G whose output
obtains large score from D
G

IRL

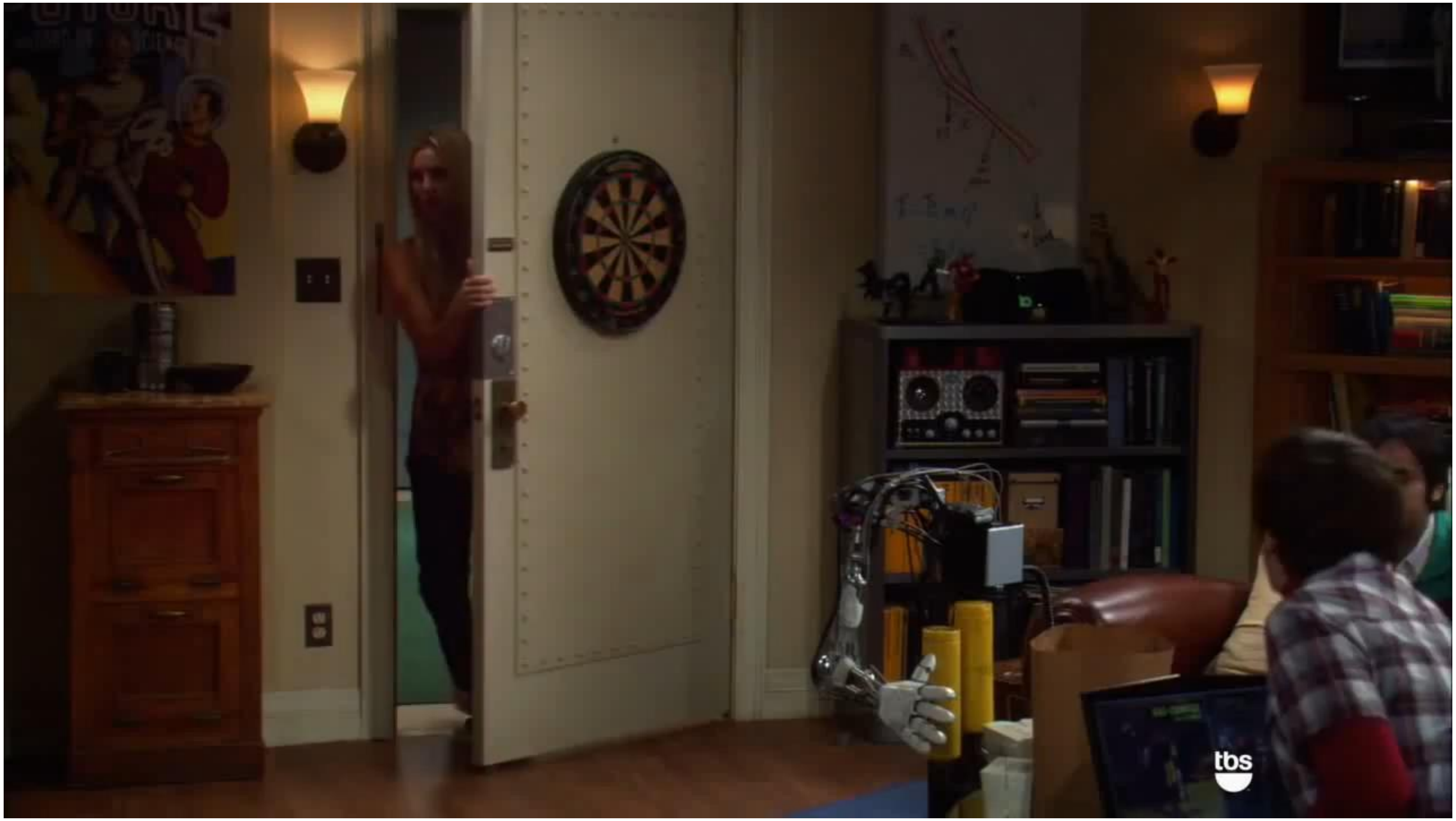


Expert
$\{\hat{\tau}_1, \hat{\tau}_2, \cdots, \hat{\tau}_N\}$
$\{\tau_1, \tau_2, \cdots, \tau_N\}$
Larger reward for $\hat{\tau}_n$,
Lower reward for τ
Reward
Function
Find a Actor obtains
large reward
Actor

Teaching Robot

- In the past ……

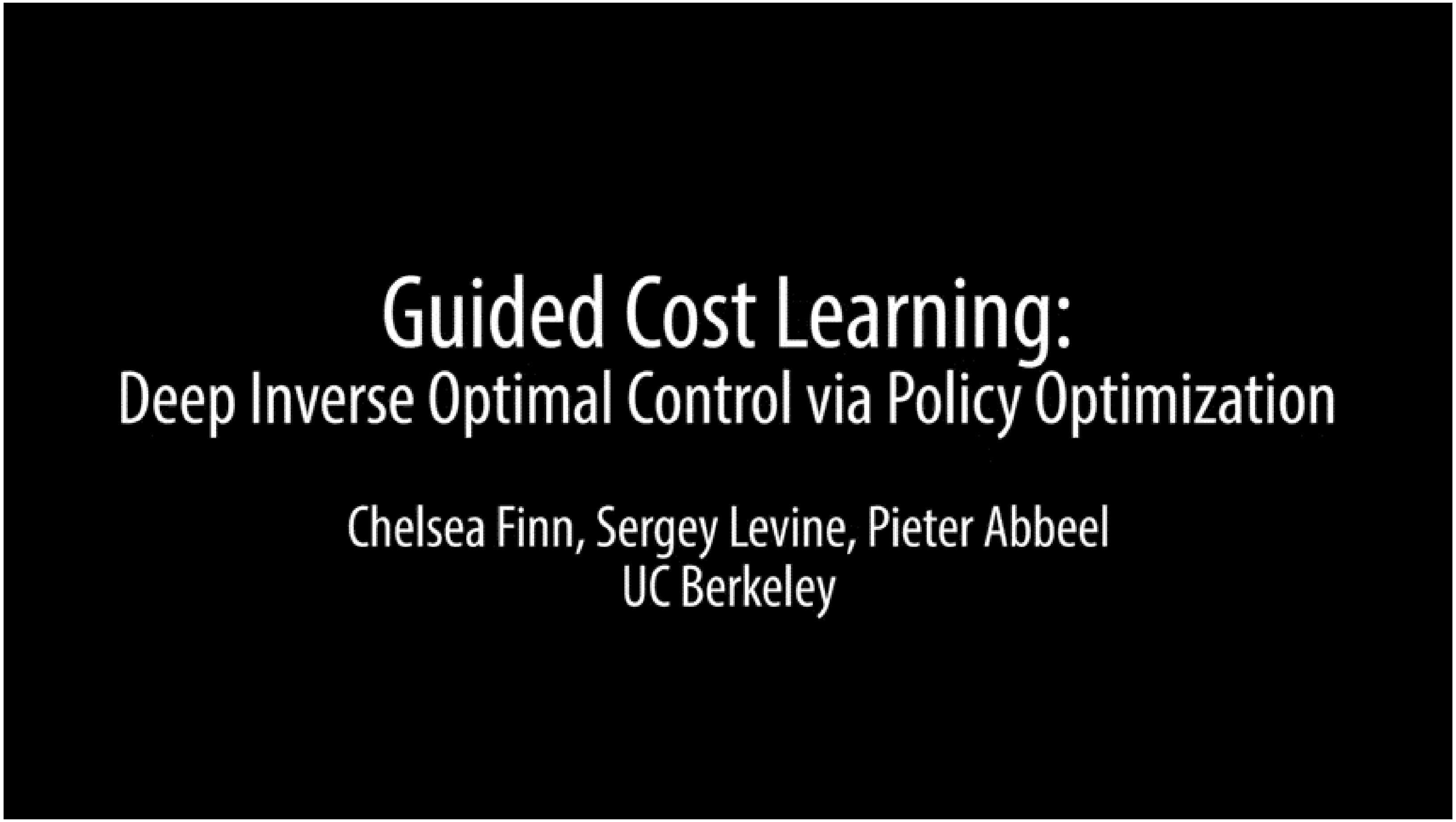
https://www.youtube.com/watch?v=DEGbtjTOIB0



Robot

Chelsea Finn, Sergey Levine, Pieter Abbeel, "Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization", ICML, 2016

http://rll.berkeley.edu/gcl/

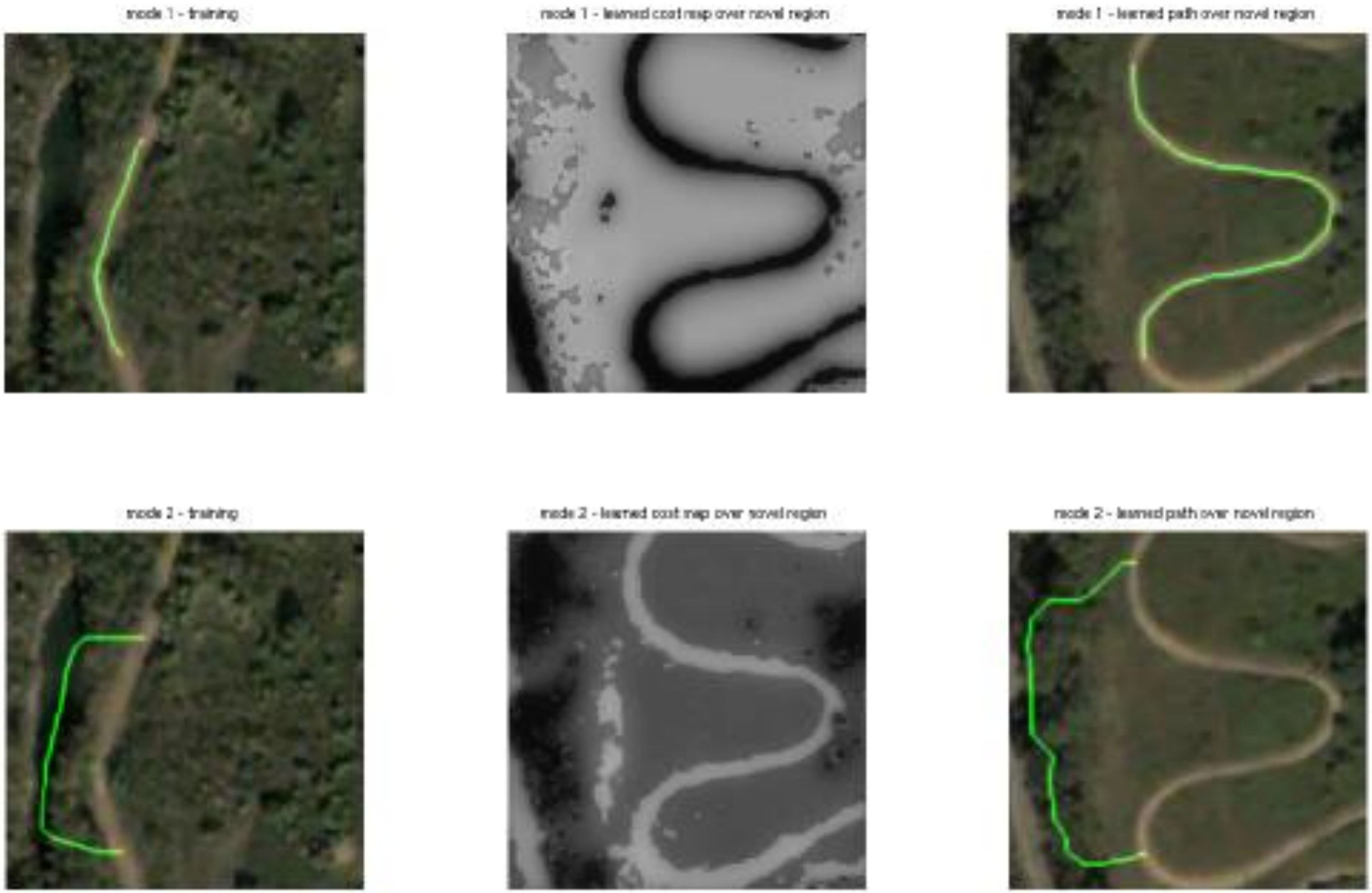


Parking a Car

https://pdfs.semanticscholar.org/27c3/6fd8e6aeadd50ec1e180399df70c46dede00.pdf

Path Planning

http://martin.zinkevich.org/publications/maximummarginplanning.pdf



Third Person Imitation Learning

- Ref: Bradly C. Stadie, Pieter Abbeel, Ilya Sutskever, “Third-Person Imitation Learning”, arXiv preprint, 2017

First Person

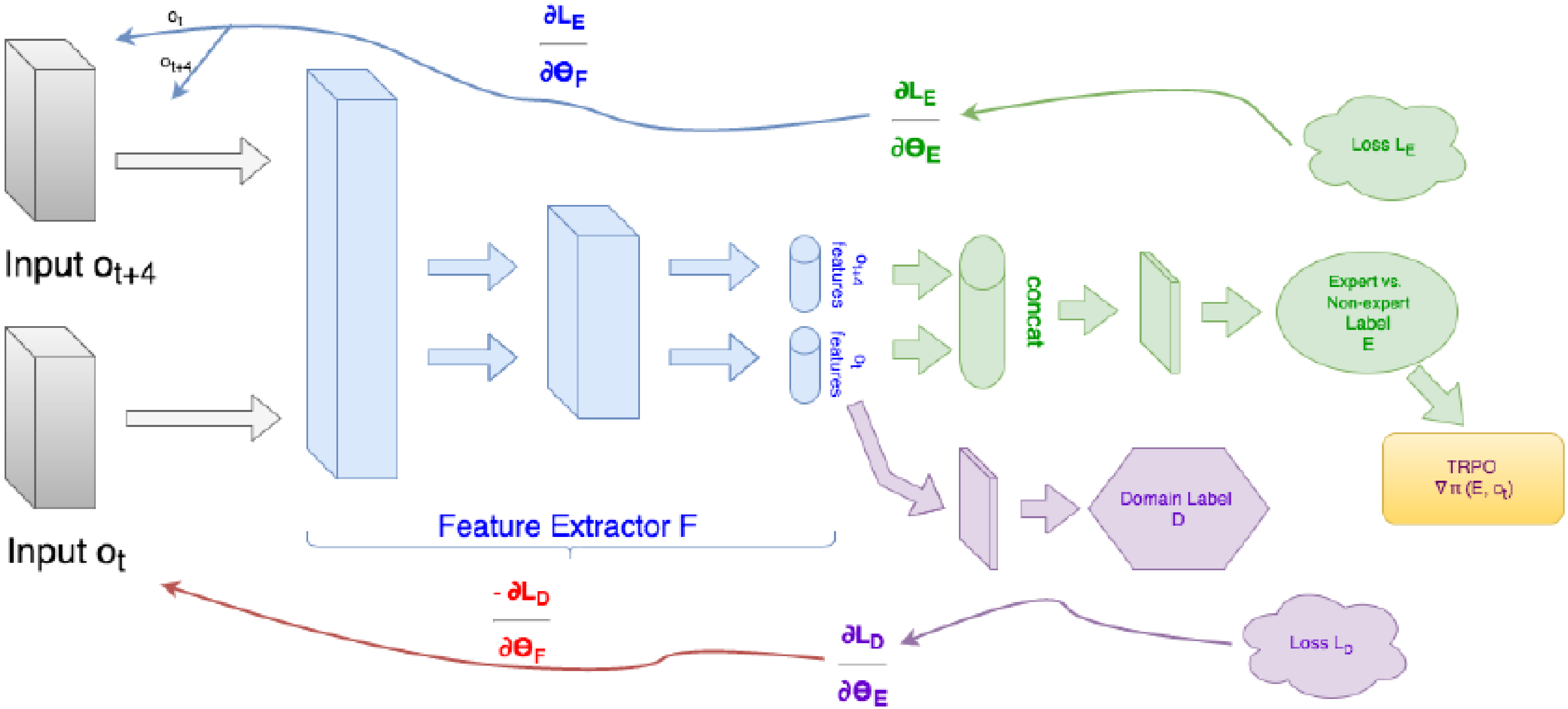
http://lasa.epfl.ch/research_new/ML/index.php

Third Person

https://kknews.cc/sports/q5kbb8.html

http://sc.chinaz.com/Files/pic/icons/1913/%E6%9C%BA%E5%99%A8%E4%BA%BA%E5%9B%BE%E6%A0%87%E4%B8%8B%E8%BD%BD34.png

Third Person Imitation Learning



Concluding Remarks

Lecture 1: Introduction of GAN

Lecture 2: Variants of GAN

Lecture 3: Making Decision and Control

To Learn More …

- **Machine Learning**
  - Slides: http://speech.ee.ntu.edu.tw/~tlkagk/courses_ML16.html
  - Video: https://www.youtube.com/watch?v=fegAeph9UaA&list=PLJV_el3uVTsPy9oCRY30oBPNLCo89yu49
- **Machine Learning and Having it Deep and Structured**
  - Slides: http://speech.ee.ntu.edu.tw/~tlkagk/courses_MLDS17.html
  - Video: https://www.youtube.com/watch?v=IzHoNwlCGnE&list=PLJV_el3uVTsPMxPbjeX7PicgWbY7F8wW9