Generative Adversarial Network
and its Applications to Speech Processing and Natural Language Processing

Hung-yi Lee

All Kinds of GAN …

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

GAN

ACGAN

BGAN

CGAN

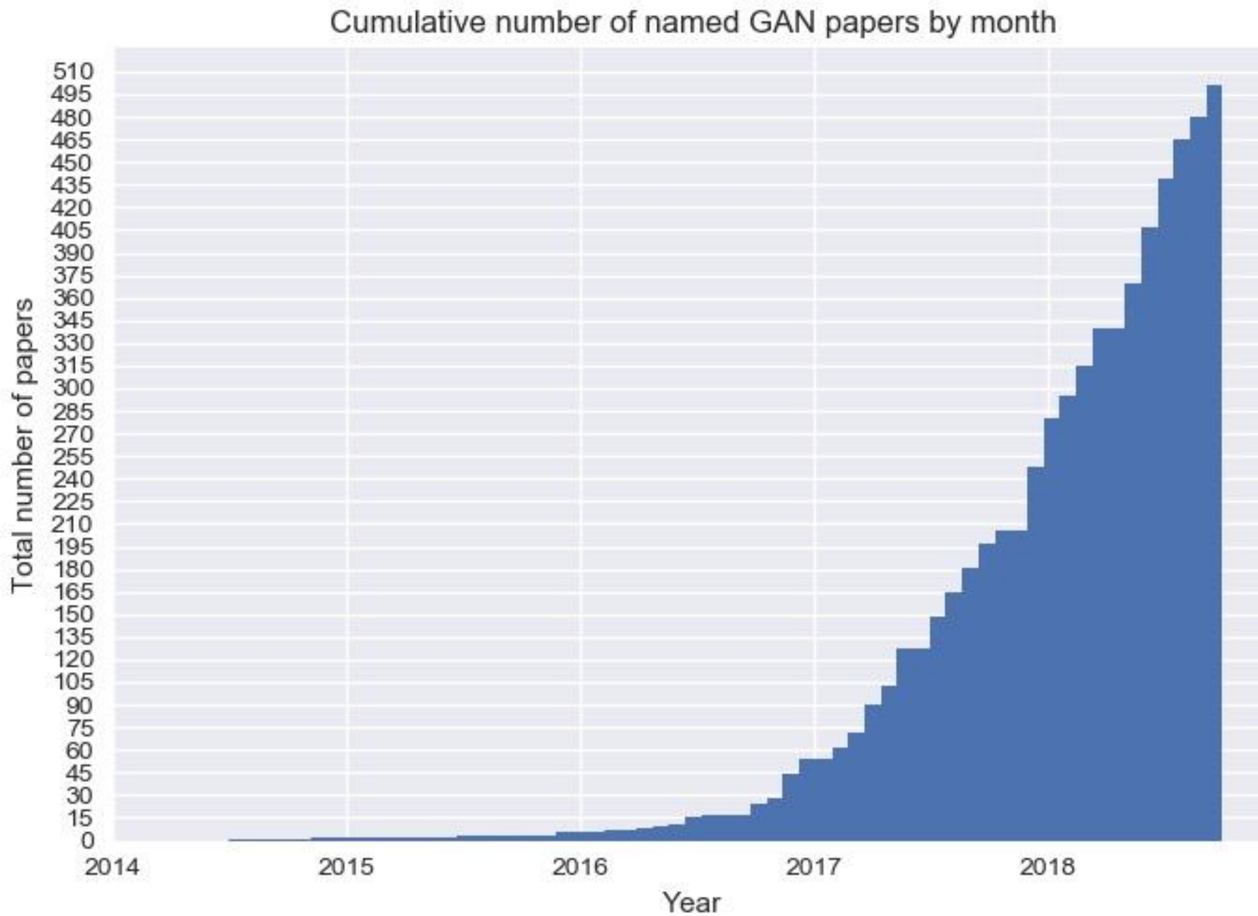
DCGAN

EBGAN

fGAN

GoGAN

⋮



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

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

Outline

Part I: General Introduction of Generative Adversarial Network (GAN)

Part II: Applications to Speech Processing

Part III: Applications to Natural Language Processing

Outline of Part 1

Generation by GAN

Conditional Generation

Unsupervised Conditional Generation

Relation to Reinforcement Learning

Outline of Part 1

Generation by GAN

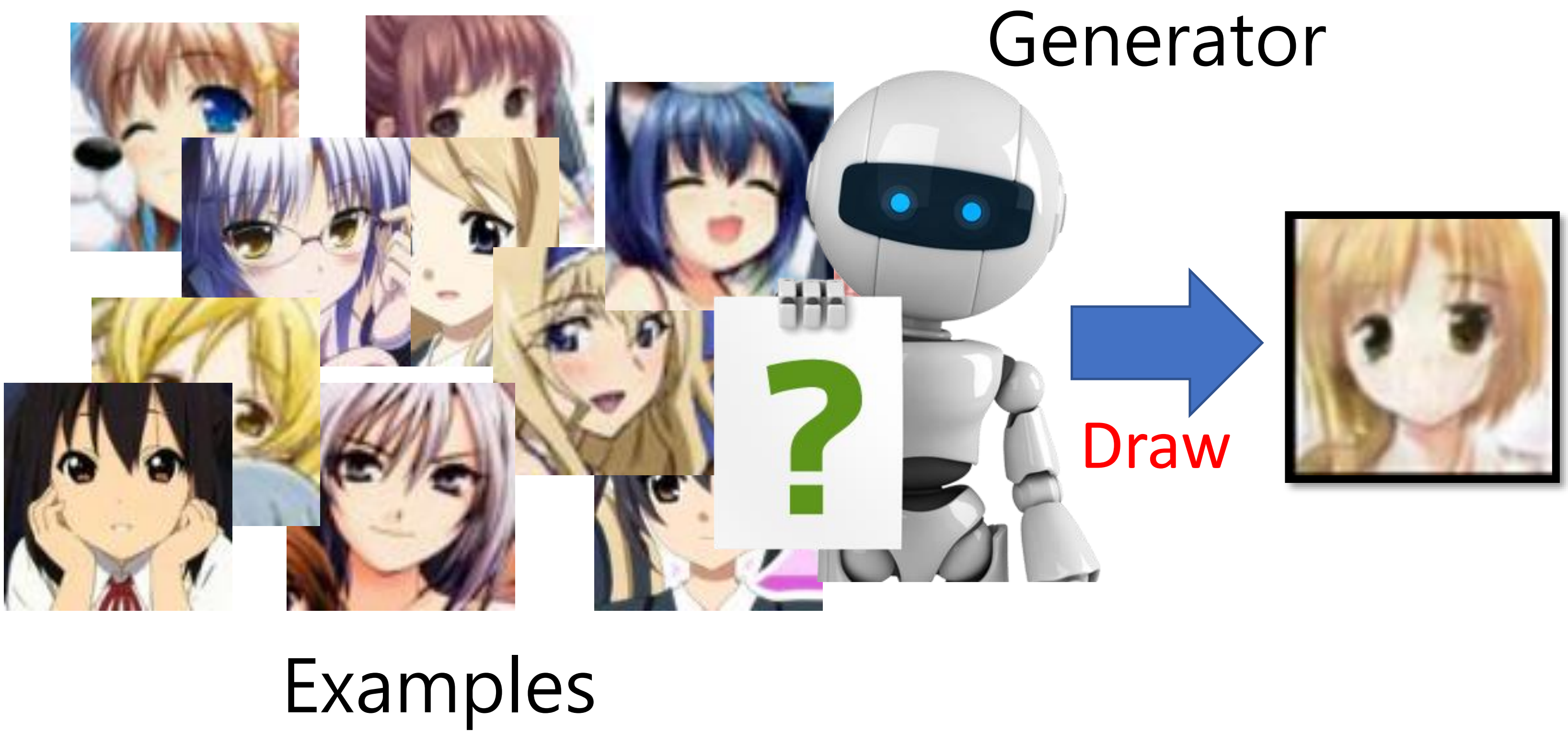
- Image Generation as Example
- Theory behind GAN
- Issues and Possible Solutions

Conditional Generation

Unsupervised Conditional Generation

Relation to Reinforcement Learning

Anime Face Generation



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

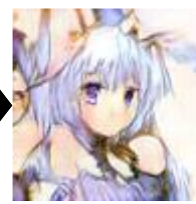
Basic Idea of GAN

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

vector → Generator → image

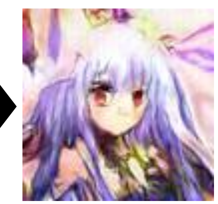
high dimensional vector

$$\begin{bmatrix} 0.1 \\ -3 \\ \vdots \\ 2.4 \\ 0.9 \end{bmatrix} \rightarrow \text{Generator} \rightarrow$$



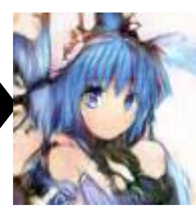
Each dimension of input vector represents some characteristics.

$$\begin{bmatrix} \boxed{3} \\ -3 \\ \vdots \\ 2.4 \\ 0.9 \end{bmatrix} \rightarrow \text{Generator} \rightarrow$$



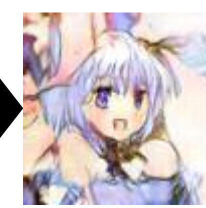
Longer hair

$$\begin{bmatrix} 0.1 \\ 2.1 \\ \vdots \\ \boxed{5.4} \\ 0.9 \end{bmatrix} \rightarrow \text{Generator} \rightarrow$$



blue hair

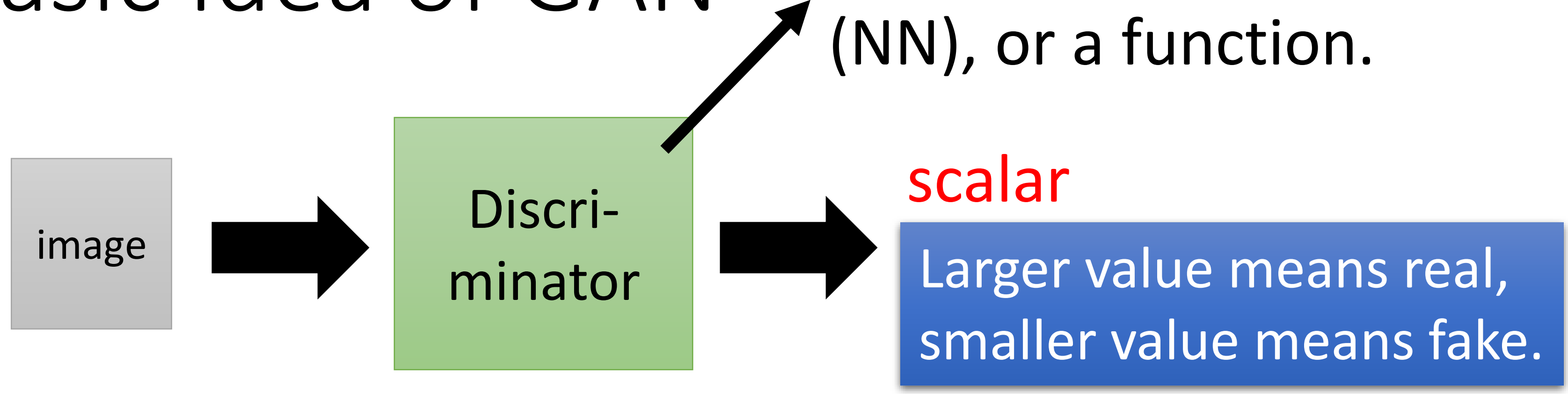
$$\begin{bmatrix} 0.1 \\ -3 \\ \vdots \\ 2.4 \\ \boxed{3.5} \end{bmatrix} \rightarrow \text{Generator} \rightarrow$$

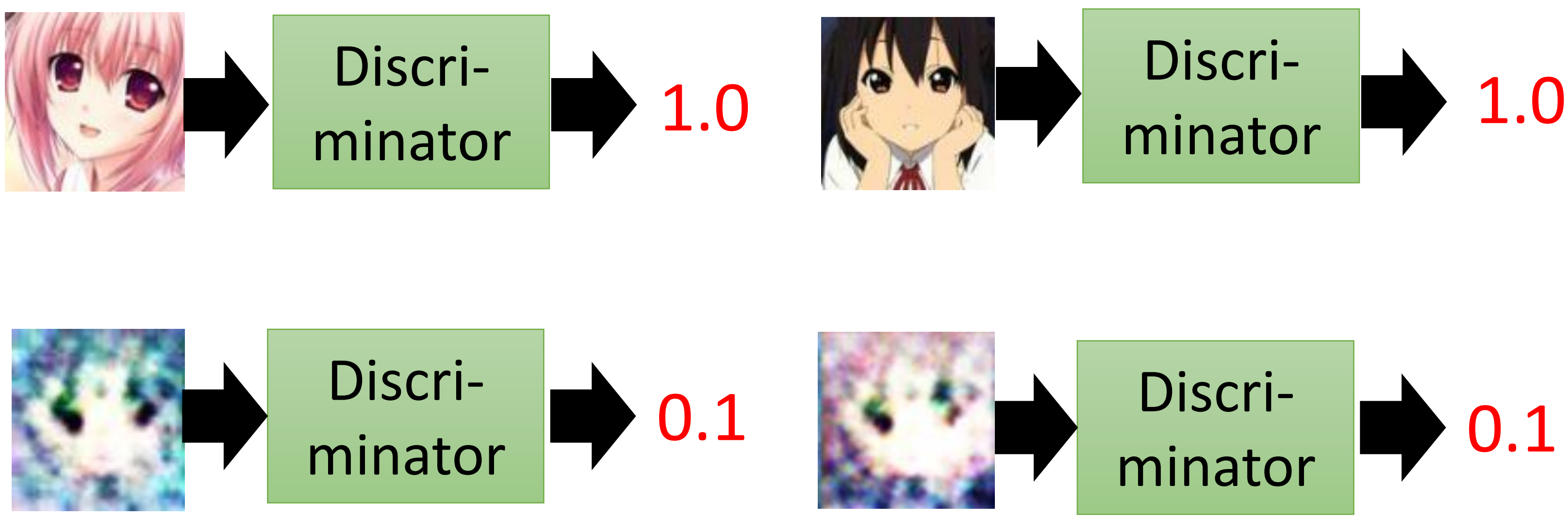


Open mouth

Basic Idea of GAN

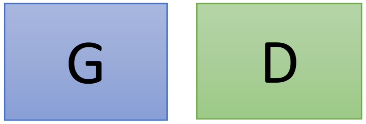
It is a neural network (NN), or a function.





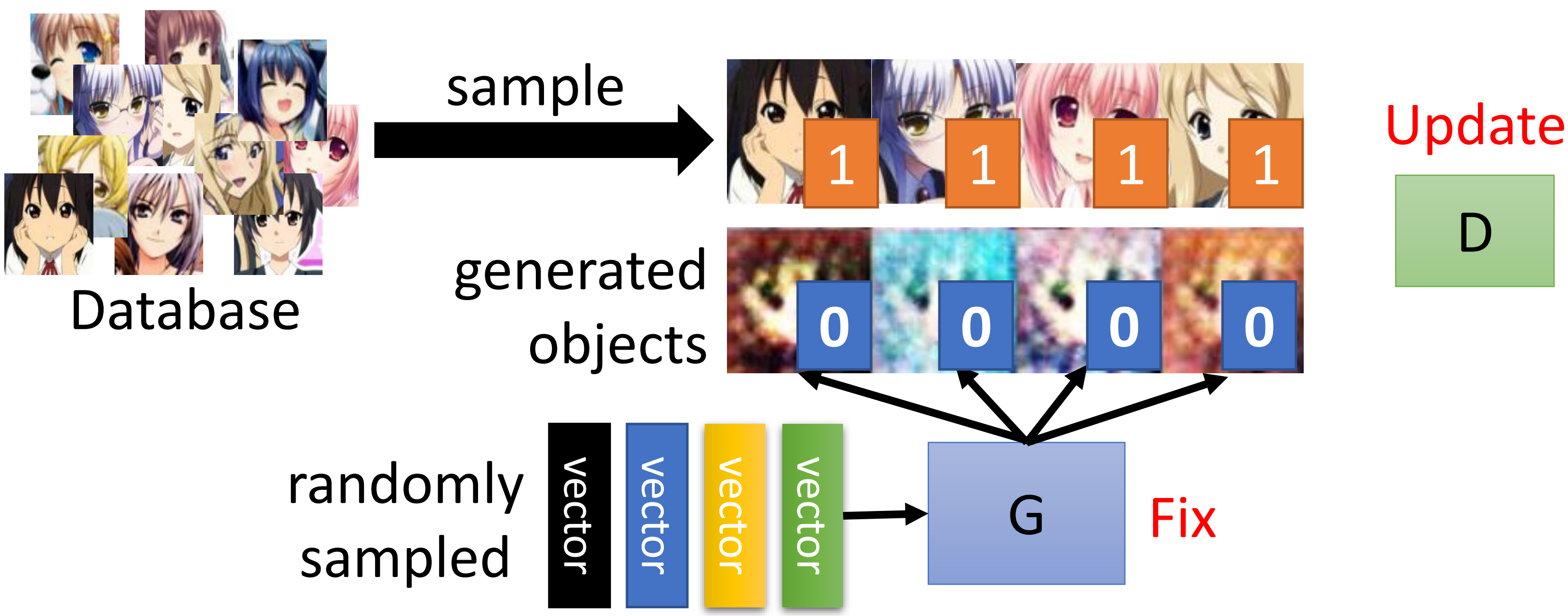
Algorithm

- Initialize generator and discriminator



- In each training iteration:

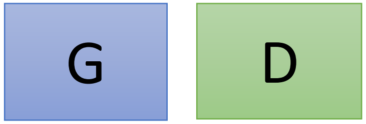
Step 1: Fix generator G, and update discriminator D



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

Algorithm

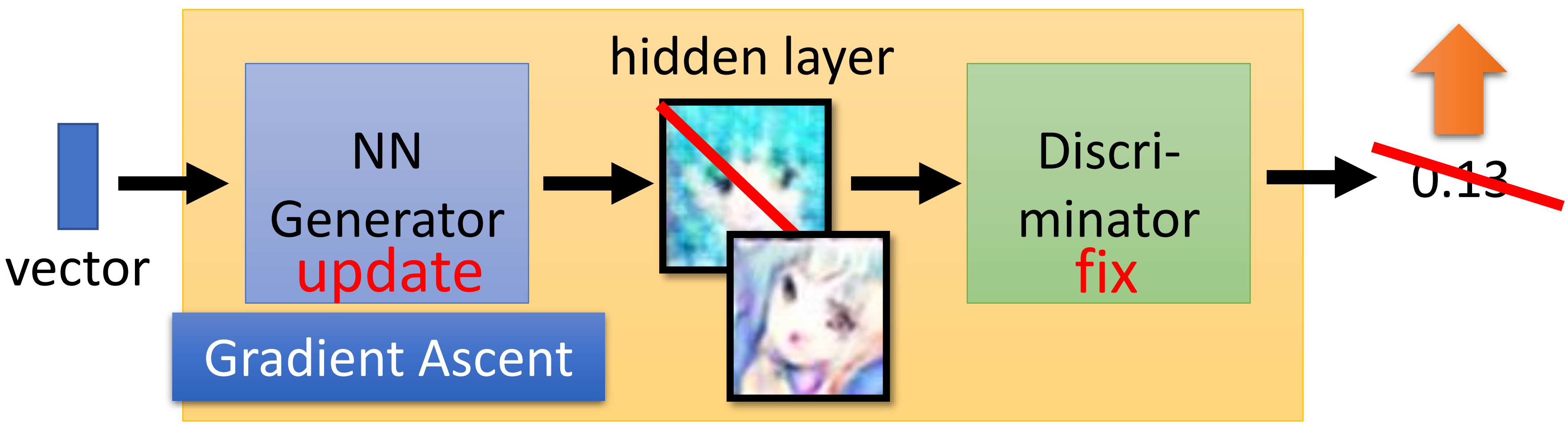
- Initialize generator and discriminator



- In each training iteration:

Step 2: Fix discriminator D, and update generator G

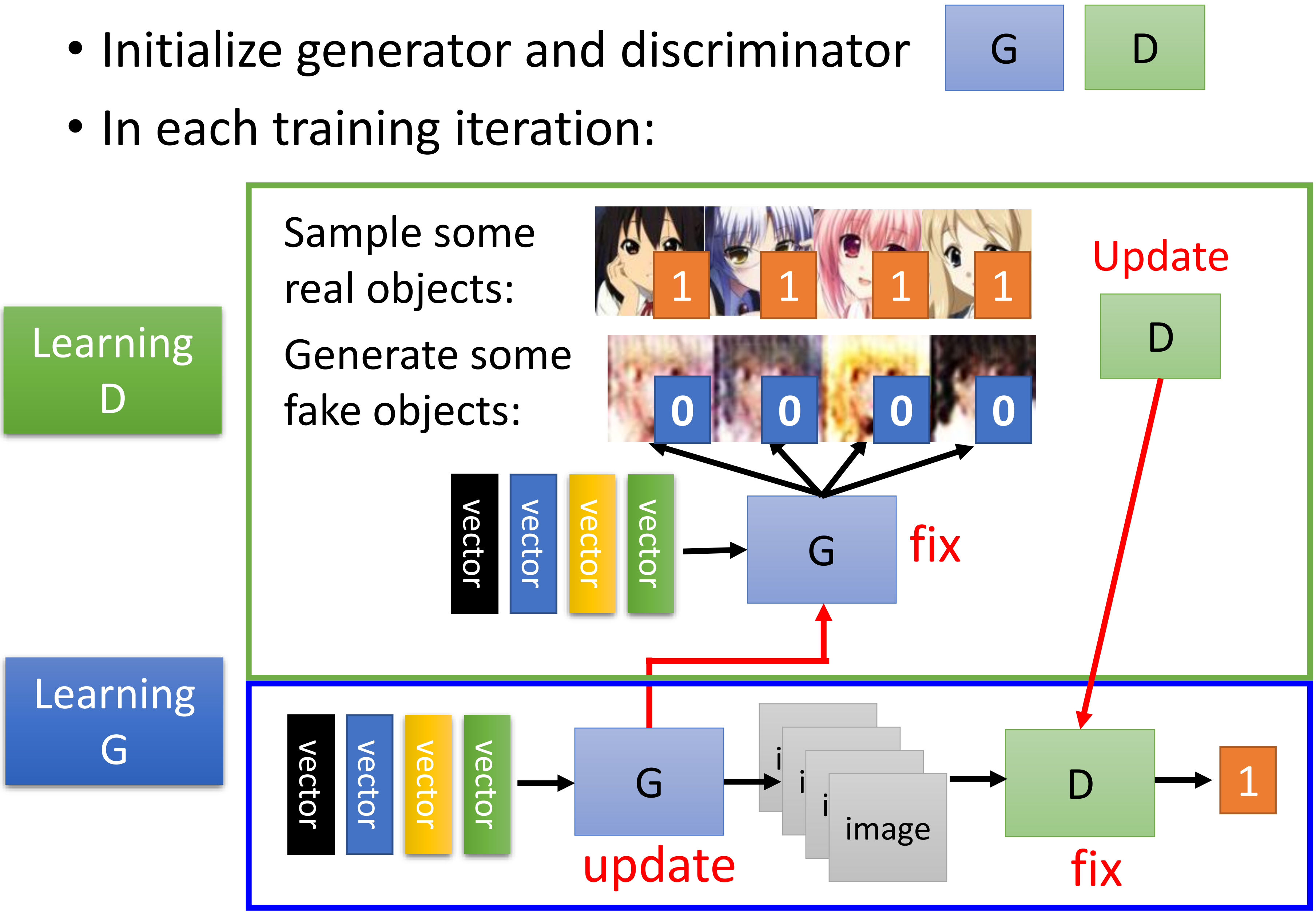
Generator learns to "fool" the discriminator



large network

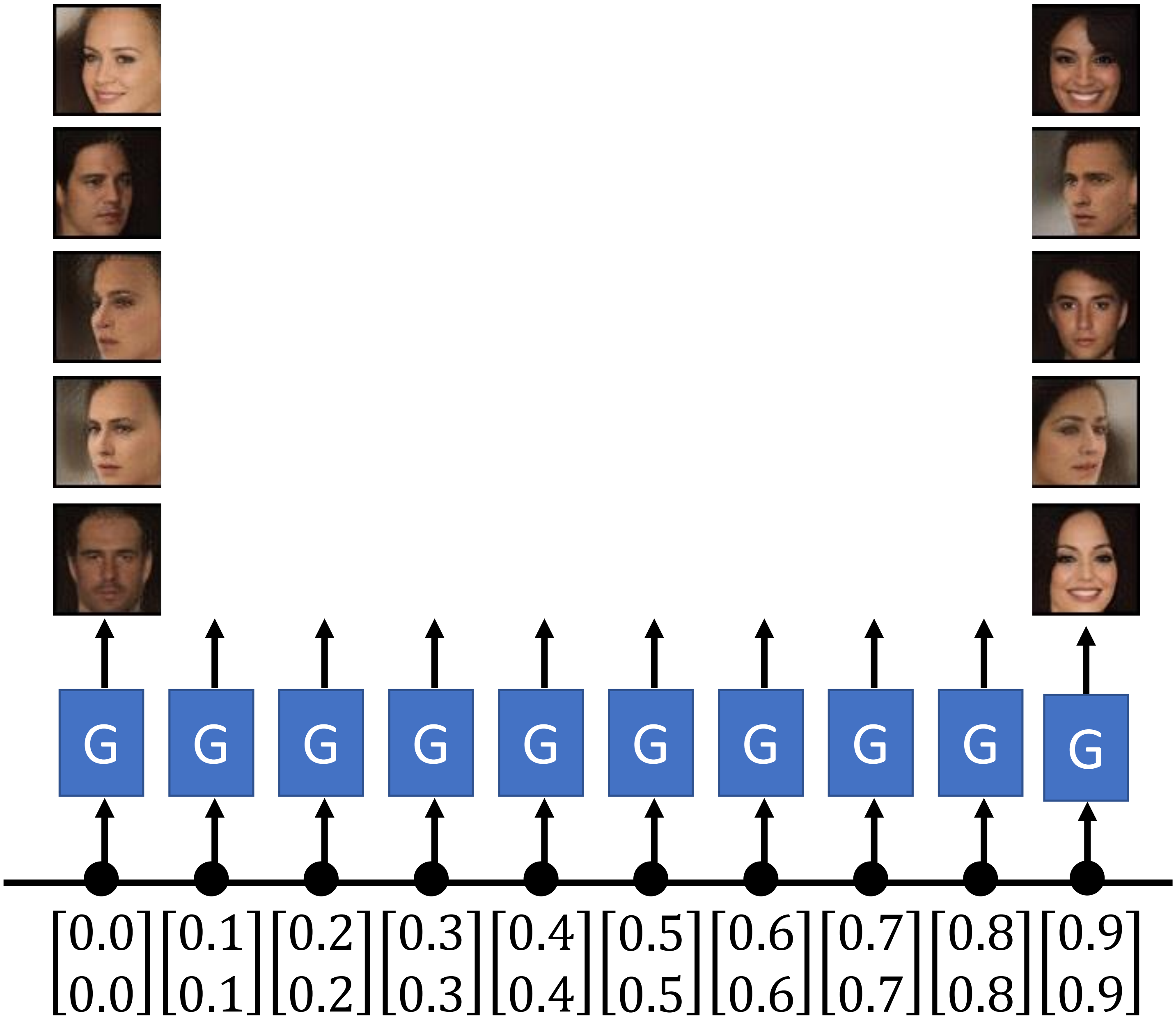
Algorithm

- Initialize generator and discriminator
- In each training iteration:



The faces generated by machine.

The images are generated by Yen-Hao Chen, Po-Chun Chien, Jun-Chen Xie, Tsung-Han Wu.


G
[0.0 0.0]
[0.1 0.1]
[0.2 0.2]
[0.3 0.3]
[0.4 0.4]
[0.5 0.5]
[0.6 0.6]
[0.7 0.7]
[0.8 0.8]
[0.9 0.9]

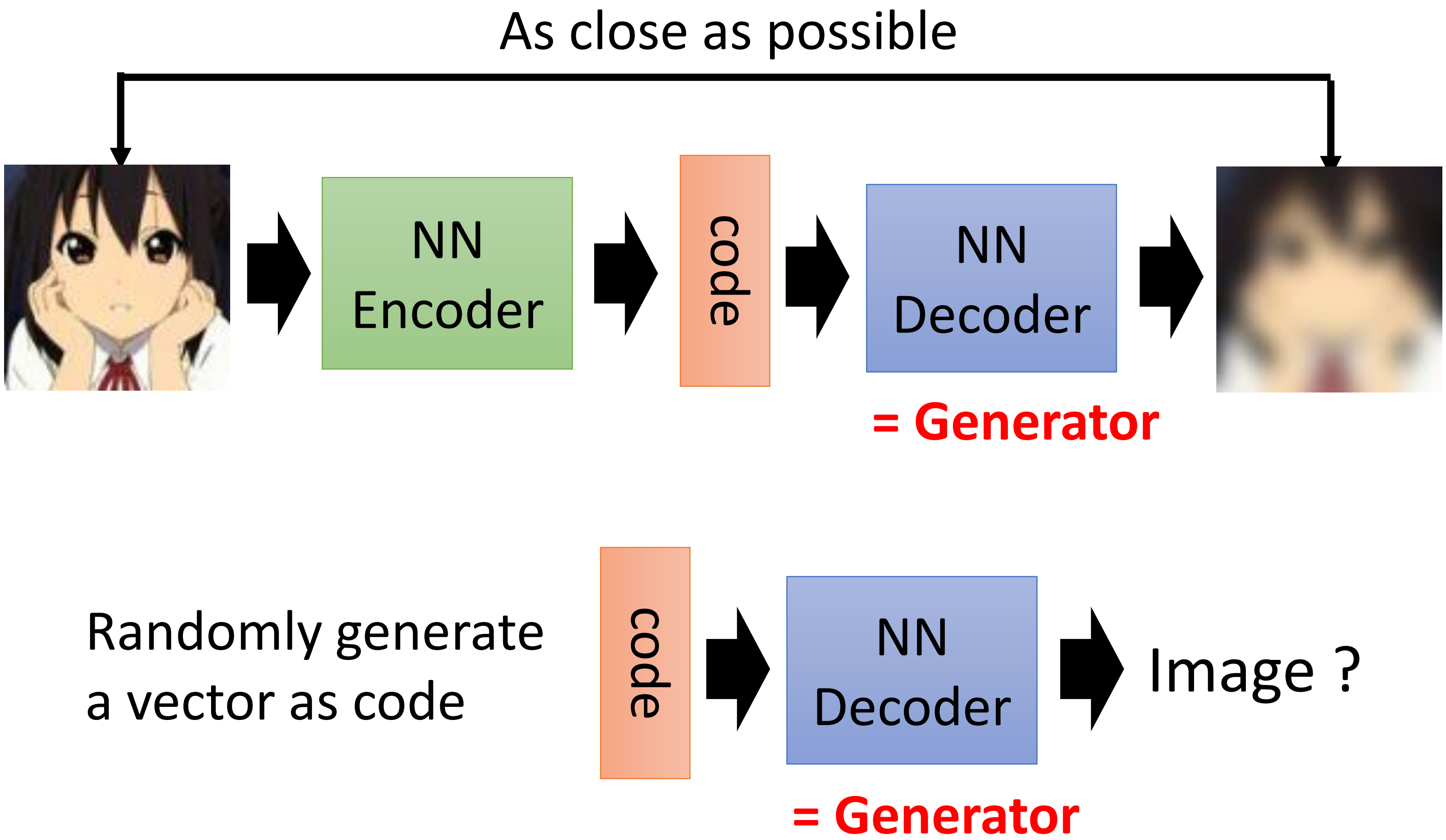
Amazing Results!

[Tero Karras, et al., ICLR, 2018]

Amazing Results!

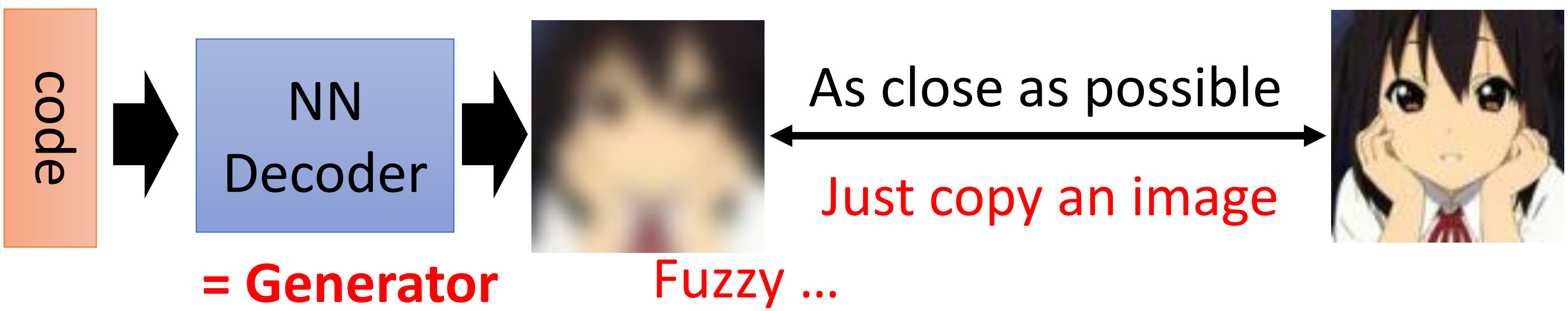
[Andrew Brock, et al., arXiv, 2018]

(Variational) Auto-encoder

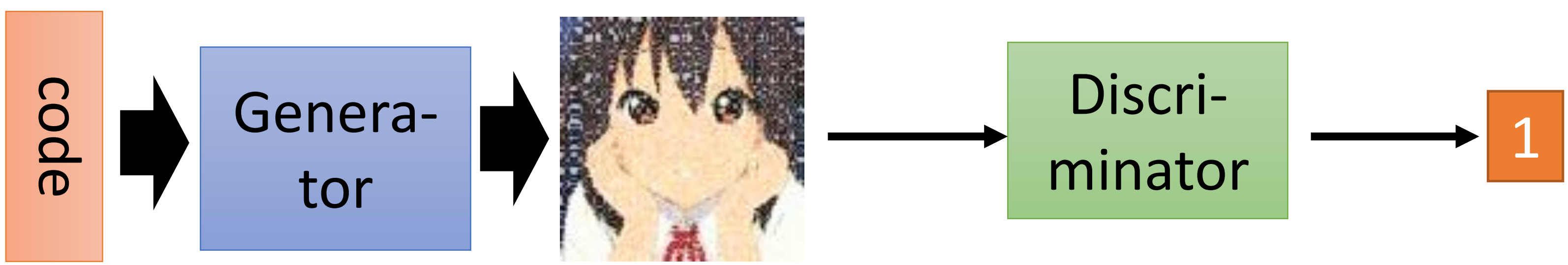


Auto-encoder v.s. GAN

Auto-encoder

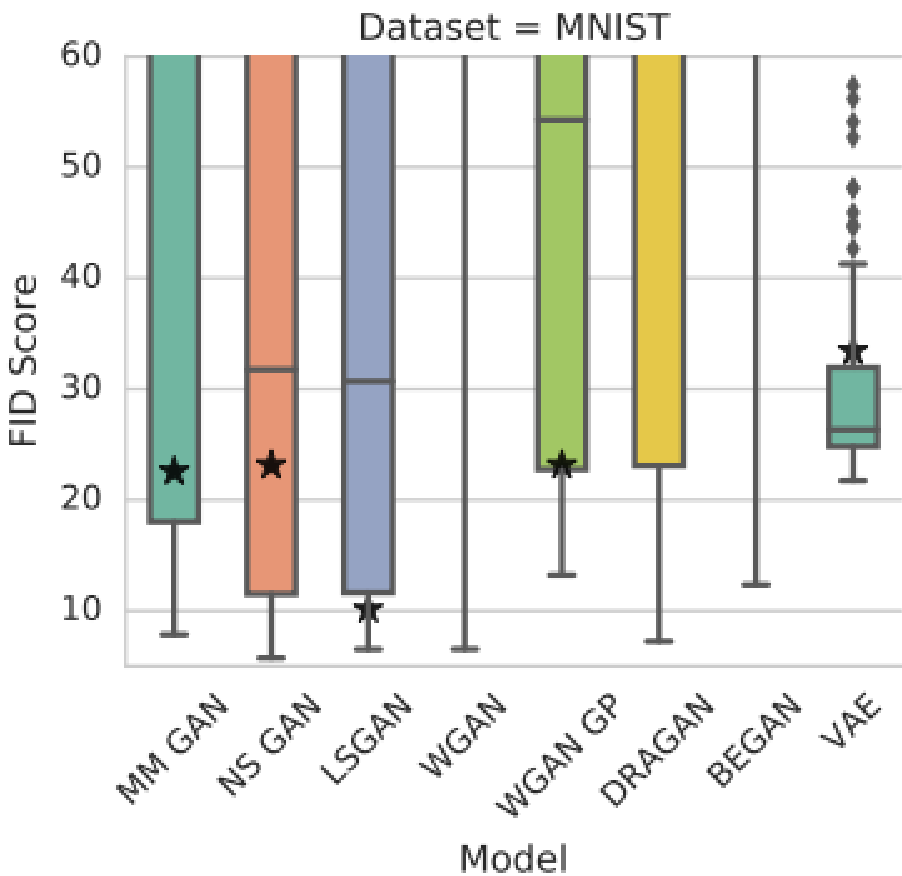


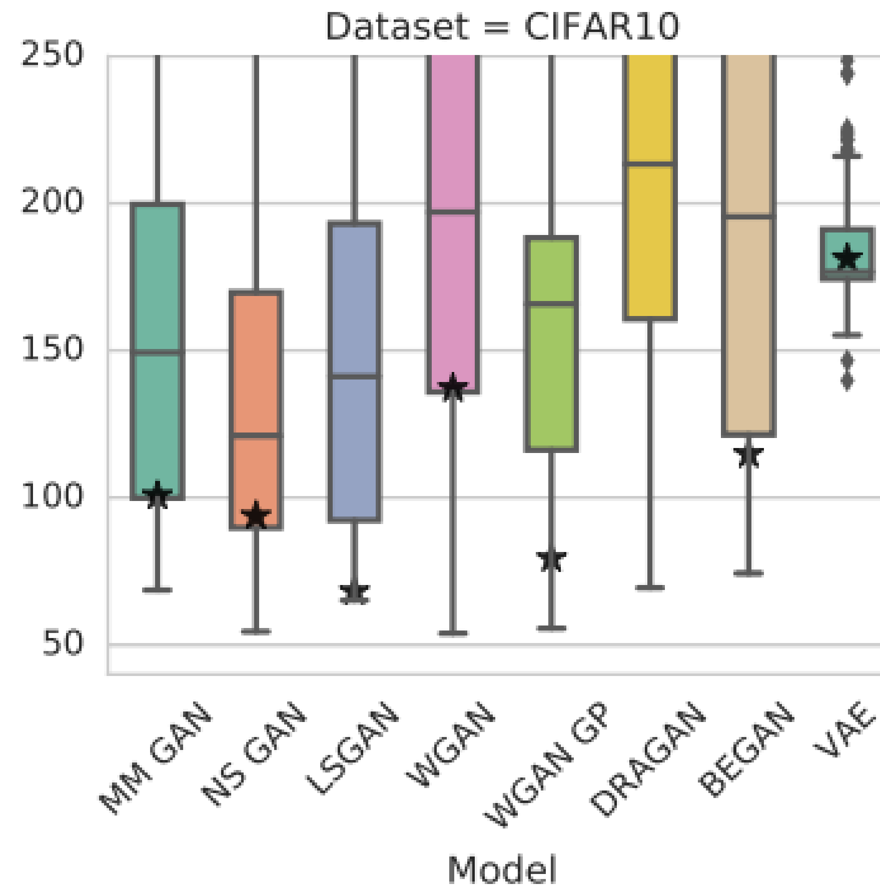
GAN



If discriminator does not simply memorize the images,
Generator learns the patterns of faces.

[Mario Lucic, et al. arXiv, 2017]





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

Outline of Part 1

Generation

- Image Generation as Example
- Theory behind GAN
- Issues and Possible Solutions

Conditional Generation

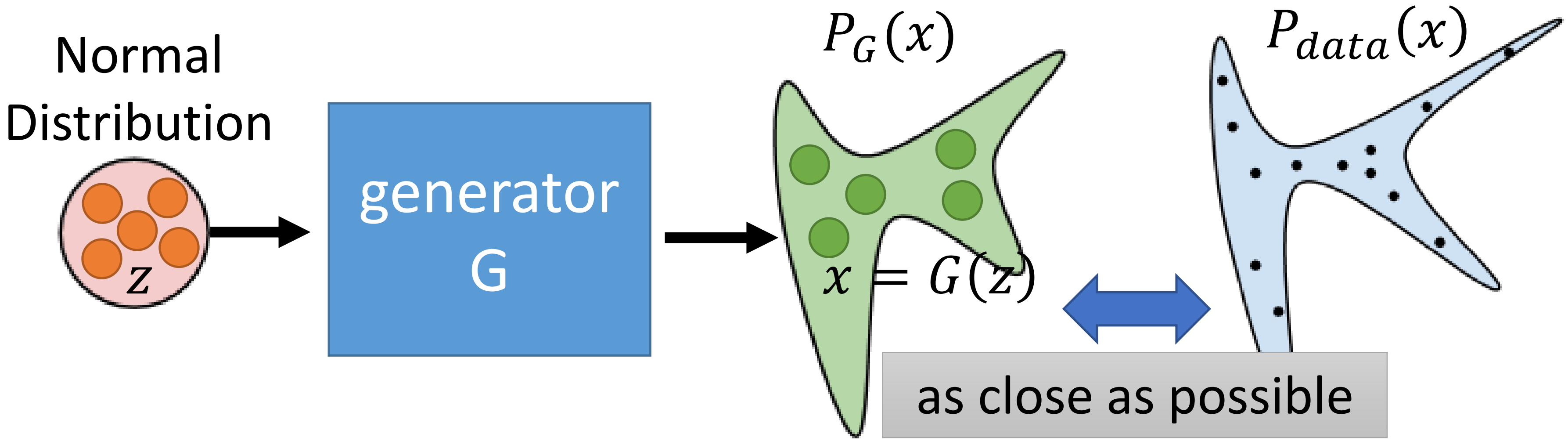
Unsupervised Conditional Generation

Relation to Reinforcement Learning

Generator

x: an image (a high-dimensional vector)

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



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

Divergence between distributions P_G and P_{data}

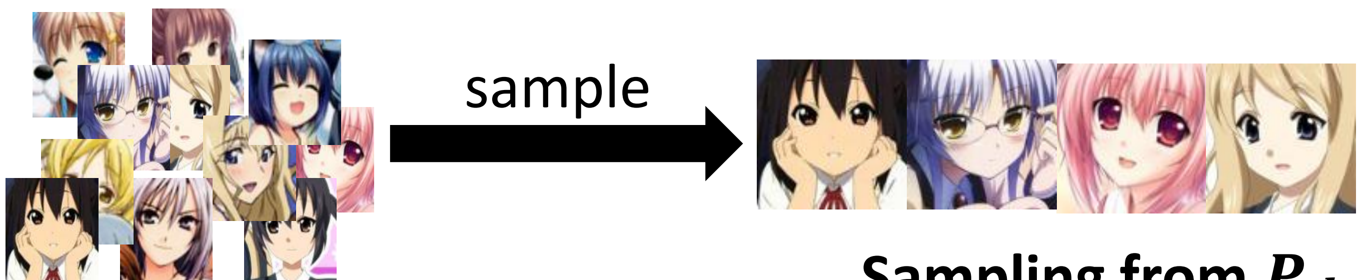
How to compute the divergence?

Discriminator

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

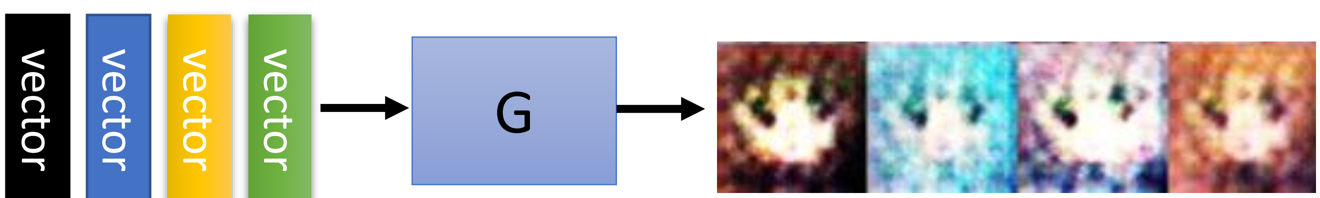
Although we do not know the distributions of P_G and P_{data}, we can sample from them.

Database



Sampling from P_{data}

sample from normal



Sampling from P_G

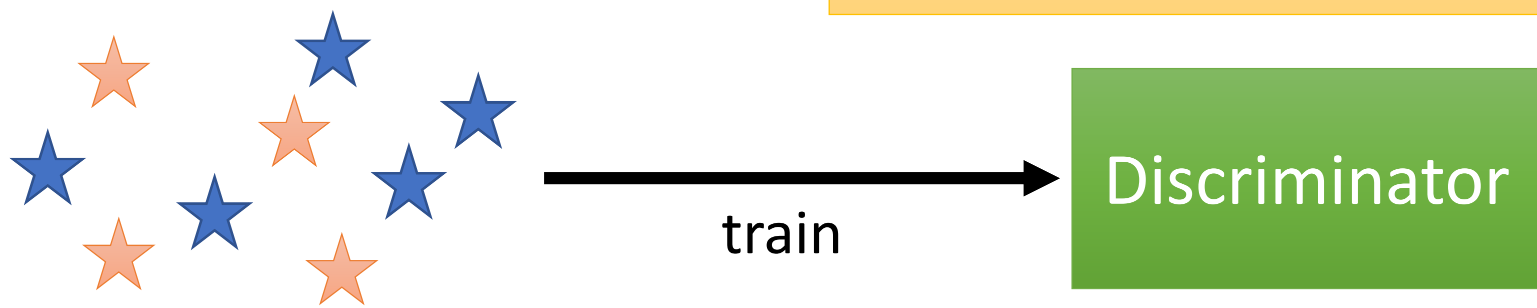
Discriminator

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

: data sampled from P_{data}

: data sampled from P_G

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



Example Objective Function for D

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

(G is fixed)

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

The maximum objective value is related to JS divergence.

[Goodfellow, et al., NIPS, 2014]

Discriminator

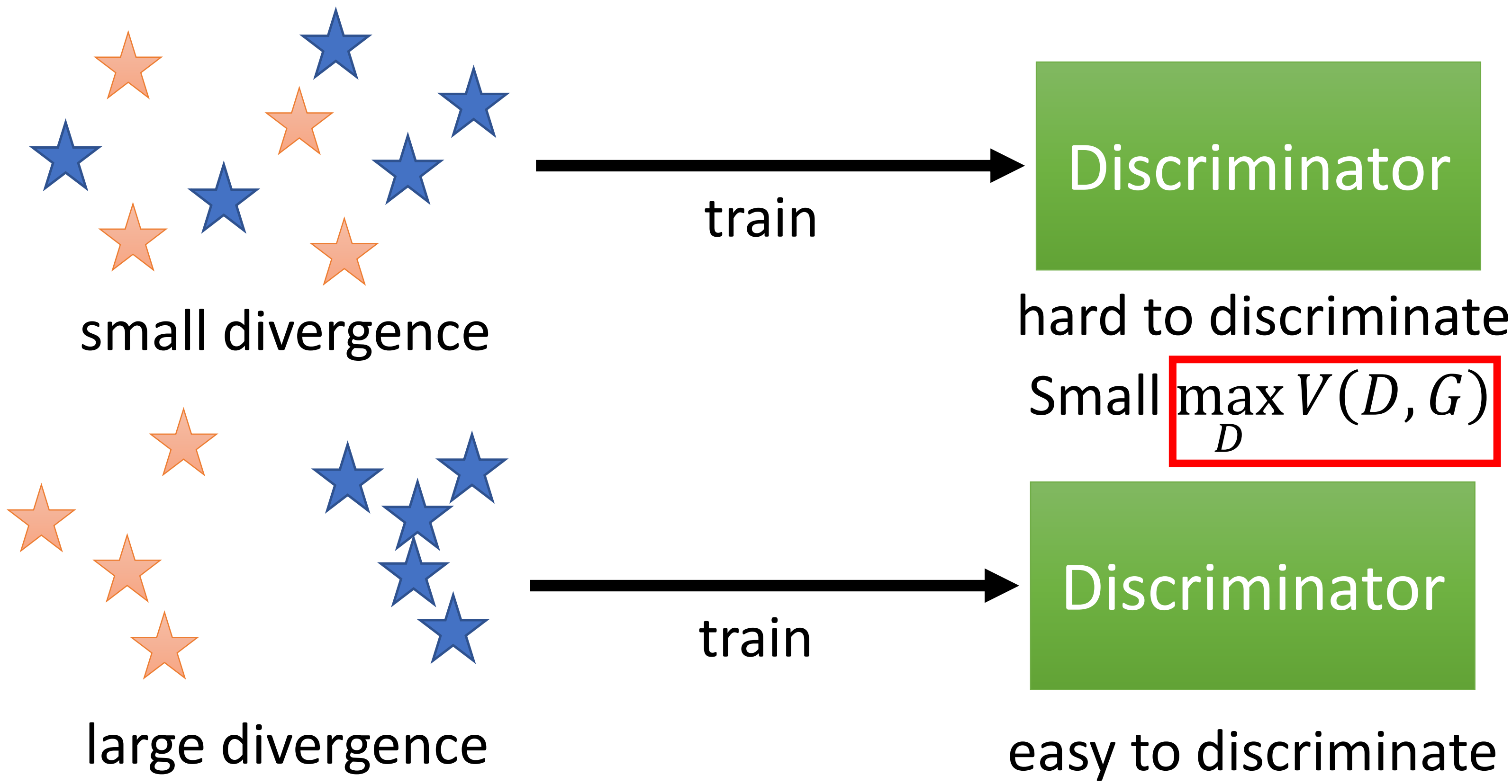
$$G^* = arg \min_G Div(P_G, P_{data})$$

: data sampled from P_{data}

: data sampled from P_G

Training:

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



[Goodfellow, et al., NIPS, 2014]

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

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

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	$\frac{1}{2}\int \lvert p(x)-q(x)\rvert \, dx$	$\frac{1}{2}\lvert u-1\rvert$
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 χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)}\, dx$	$(u-1)^2$
Neyman χ^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 ☺

[Sebastian Nowozin, et al., NIPS, 2016]

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

Outline of Part 1

Generation

- Image Generation as Example
- Theory behind GAN
- Issues and Possible Solutions

More tips and tricks:
https://github.com/soumith/ganhacks

Conditional Generation

Unsupervised Conditional Generation

Relation to Reinforcement Learning

GAN is hard to train ……

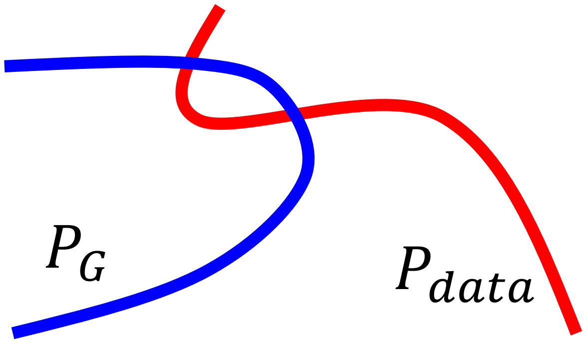
(I found this joke from 陳柏文's facebook.)

JS divergence is not suitable

- In most cases, P_G and P_{data} are not overlapped.
- 1. The nature of data

Both P_{data} and P_G are low-dim manifold in high-dim space.

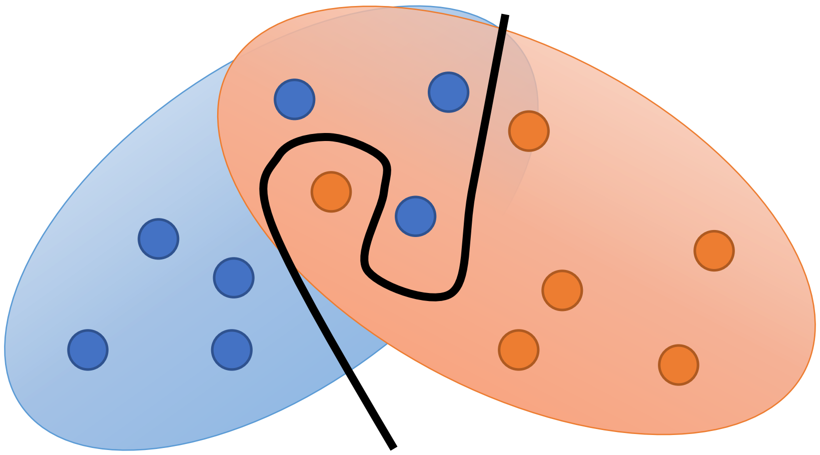
The overlap can be ignored.



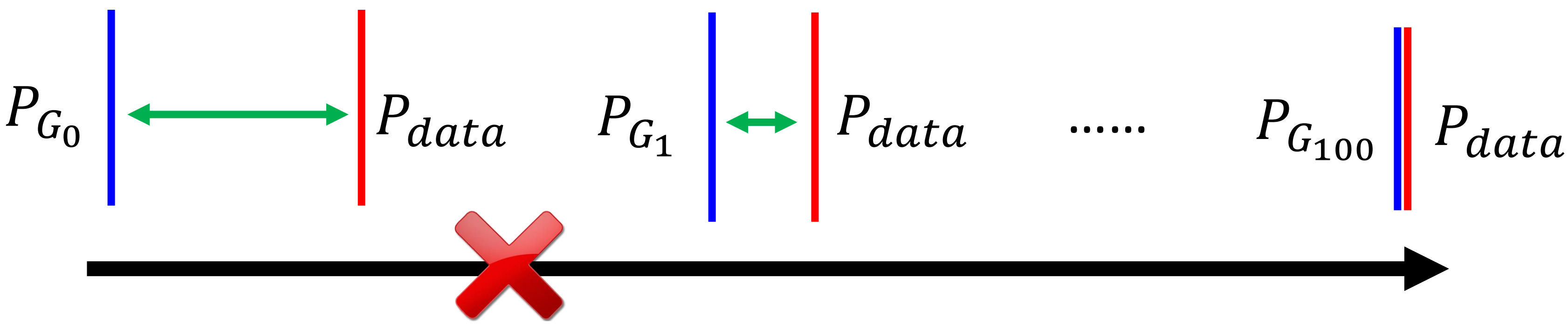
- 2. Sampling

Even though P_{data} and P_G have overlap.

If you do not have enough sampling ……



What is the problem of JS divergence?



Equally bad

$JS\left(P_{G_0}, P_{data}\right) = log2$ $\quad$ $JS\left(P_{G_1}, P_{data}\right) = log2$ $\quad$ ...... $\quad$ $JS\left(P_{G_{100}}, P_{data}\right) = 0$

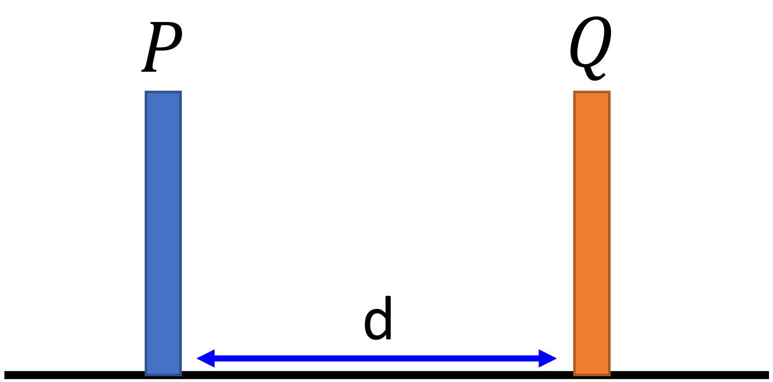
JS divergence is log2 if two distributions do not overlap.

Intuition: If two distributions do not overlap, binary classifier achieves 100% accuracy

➡ Same objective value is obtained. ➡ Same divergence

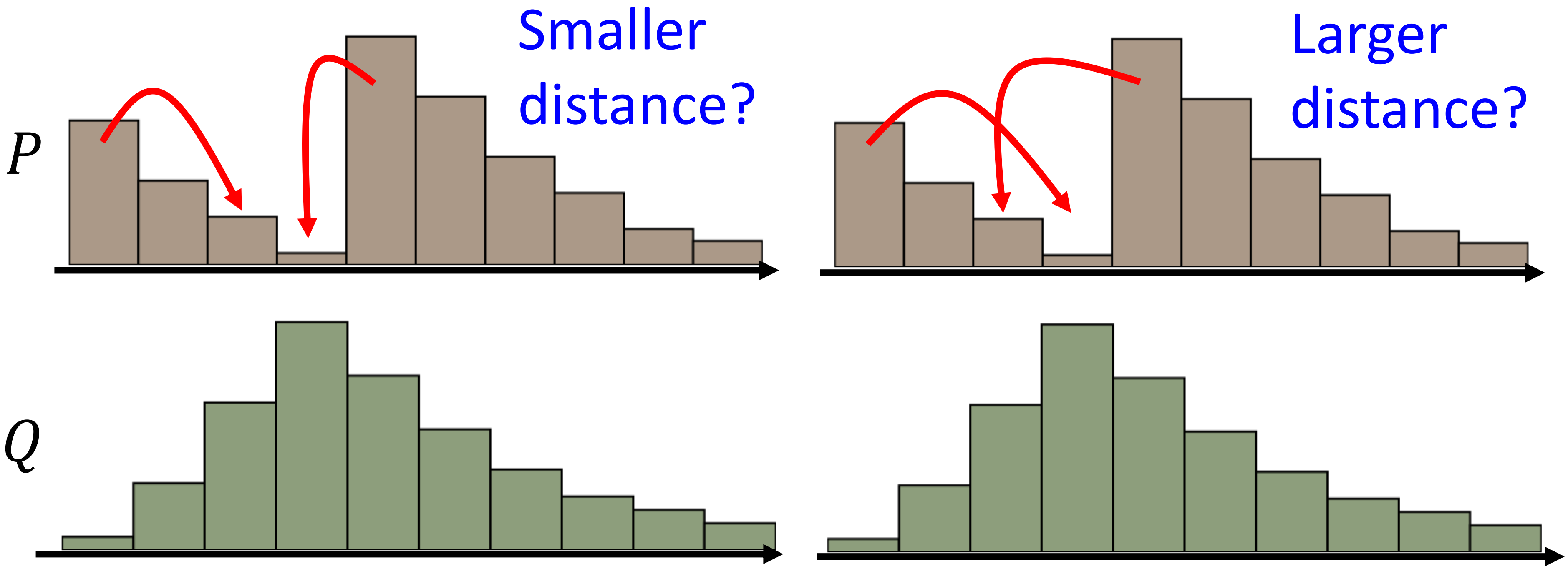
Wasserstein distance

- 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.



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

Wasserstein distance

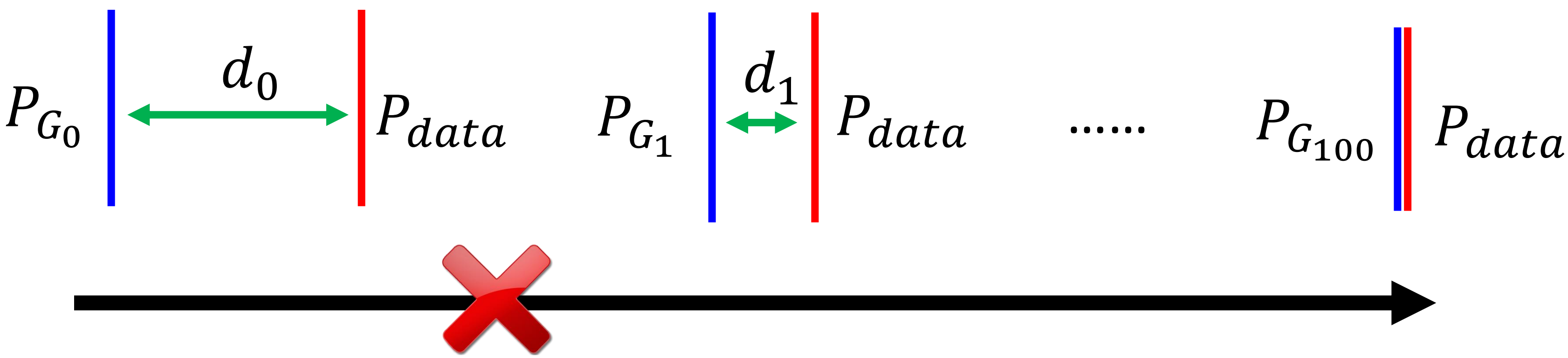


There are many possible “moving plans”.

Using the “moving plan” with the smallest average distance to define the Wasserstein distance.

Source of image: https://vincentherrmann.github.io/blog/wasserstein/

What is the problem of JS divergence?



$JS(P_{G_0}, P_{data}) = log2$ …… $JS(P_{G_1}, P_{data}) = log2$ …… $JS(P_{G_{100}}, P_{data}) = 0$

$W(P_{G_0}, P_{data}) = d_0$ …… $W(P_{G_1}, P_{data}) = d_1$ **Better!** …… $W(P_{G_{100}}, P_{data}) = 0$

[Martin Arjovsky, et al., arXiv, 2017]

WGAN

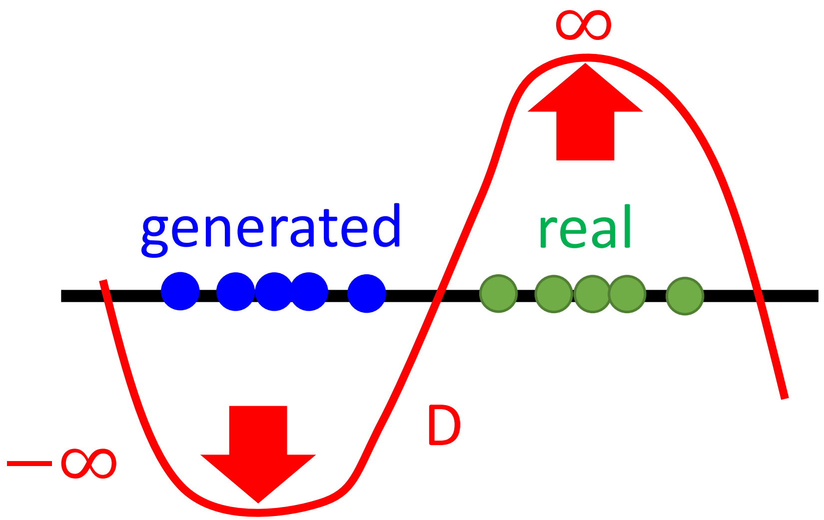
Evaluate wasserstein distance between P_{data} and P_G

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

D has to be smooth enough. How to fulfill this constraint?

Without the constraint, the training of D will not converge.

Keeping the D smooth forces D(x) become ∞ and $-\infty$



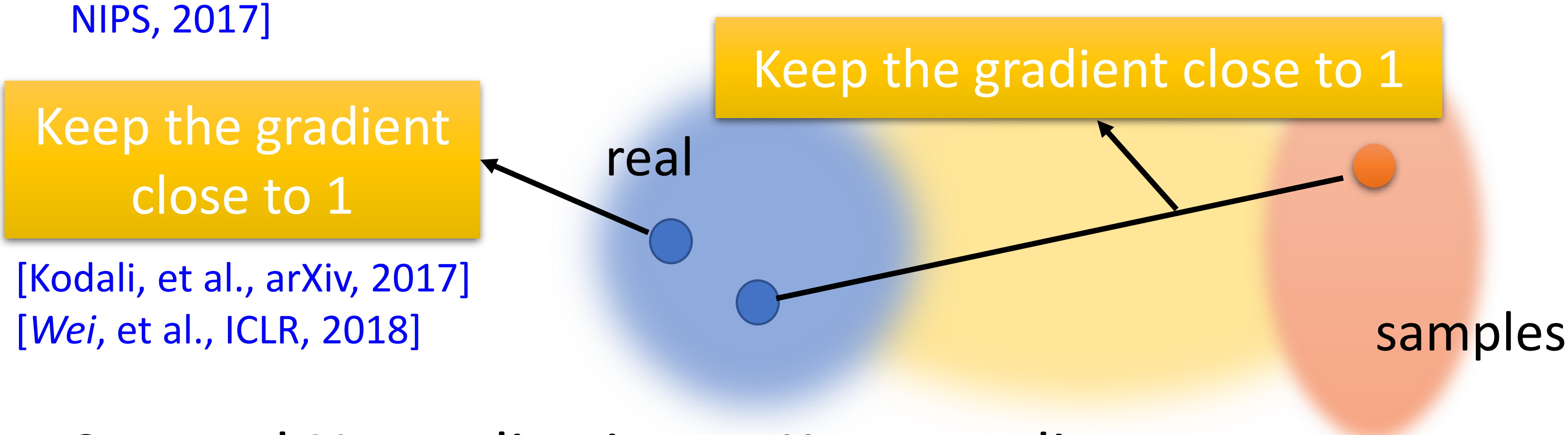
$$V(G,D) = \max_{D \in 1-Lipschitz} \left\{ E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)] \right\}$$

- Original WGAN → Weight Clipping [Martin Arjovsky, et al., arXiv, 2017]

  Force the parameters w between c and -c

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

- Improved WGAN → Gradient Penalty [Ishaan Gulrajani, NIPS, 2017]



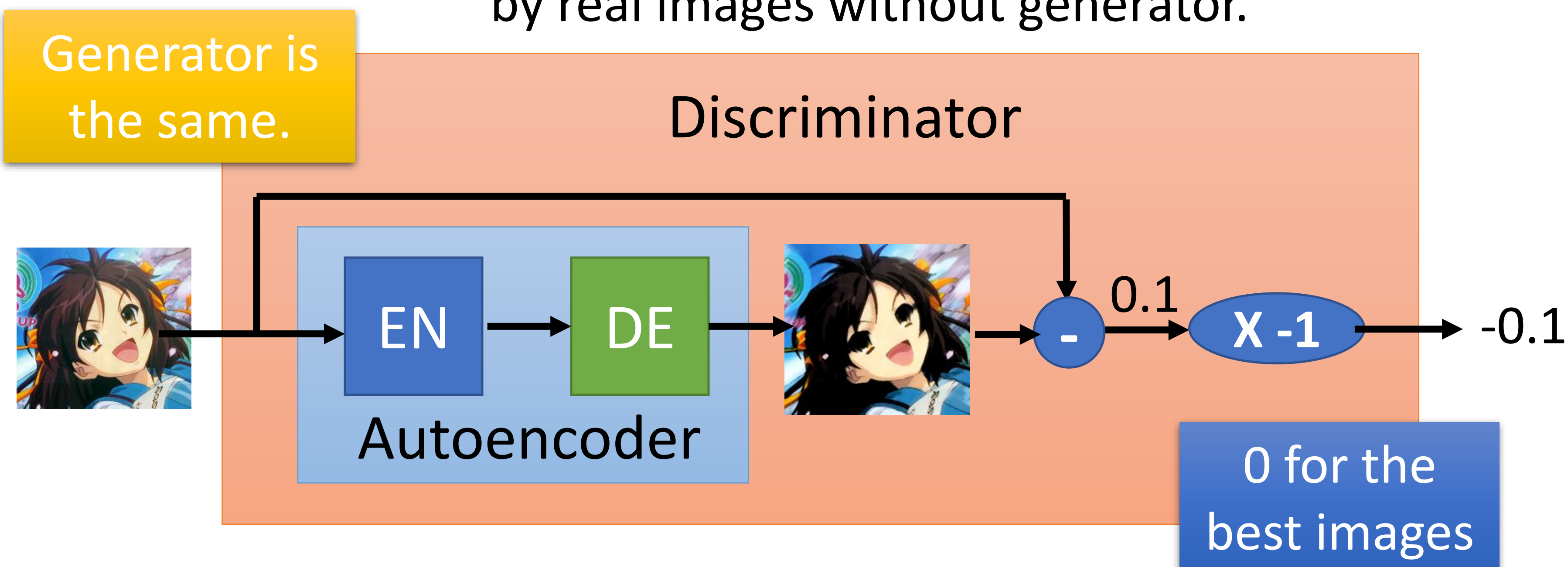
[Kodali, et al., arXiv, 2017]
[*Wei*, et al., ICLR, 2018]

- Spectral Normalization → Keep gradient norm smaller than 1 everywhere [Miyato, et al., ICLR, 2018]

[Junbo Zhao, et al., arXiv, 2016]

Energy-based GAN (EBGAN)

- Using an autoencoder as discriminator D
  - ➢ Using the negative reconstruction error of auto-encoder to determine the goodness
  - ➢ **Benefit**: The auto-encoder can be pre-train by real images without generator.

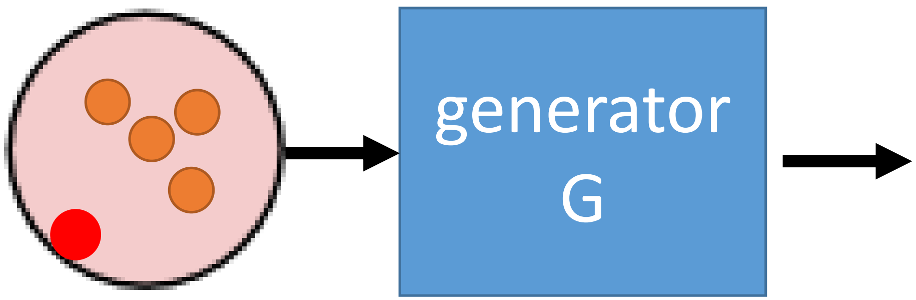


Tip: Improve Quality during Testing

本技巧由柯達方提供

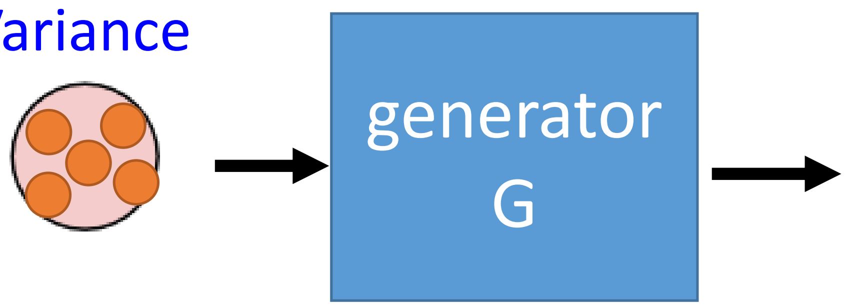
This tip is also used in [Andrew Brock, et al., arXiv, 2018]

Normal Distribution



Some samples are poor.

Smaller Variance

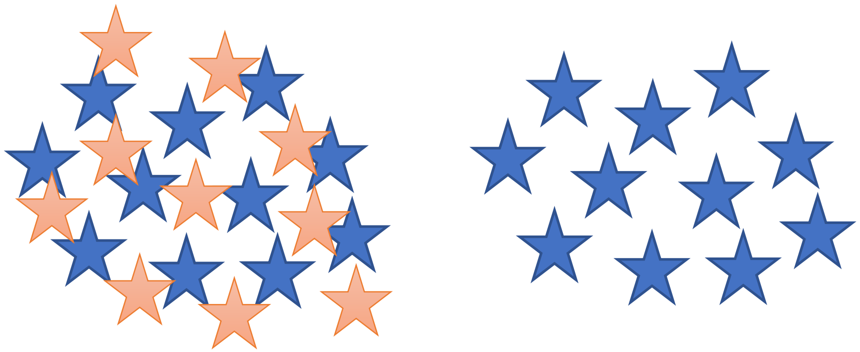


The output would be more stable, but sacrifice the diversity.

Mode Collapse

Training with too many iterations ……

Mode Dropping



Generator switches mode during training

Generator
at iteration t

Generator
at iteration t+1

Generator
at iteration t+2

BEGAN on CelebA

Tip: Ensemble

To generate an image

Random pick a generator G_i, and then use G_i to generate the image

......

Generator 1

Generator 2

......

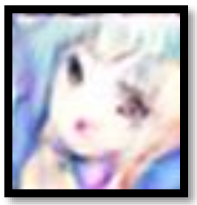
Train a set of generators: $\{G_1, G_2, \cdots, G_N\}$

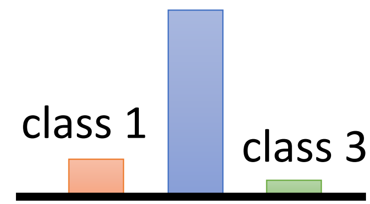
[Tim Salimans, et al., NIPS, 2016]

Objective Evaluation

x: image

y: class (output of CNN)

x  → Off-the-shelf Image Classifier → $P(y|x)$



e.g. Inception net, VGG, etc.

Concentrated distribution means higher visual quality

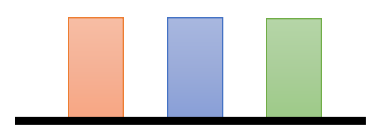
x^1 → CNN → $P(y^1|x^1)$

x^2 → CNN → $P(y^2|x^2)$

x^3 → CNN → $P(y^3|x^3)$

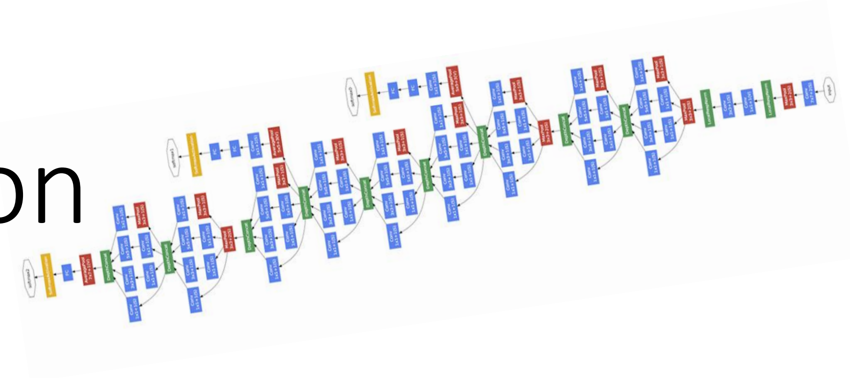
⋮

$$P(y) = \frac{1}{N}\sum_n P(y^n|x^n)$$

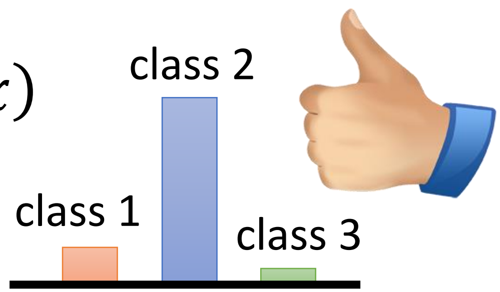


Uniform distribution means higher variety

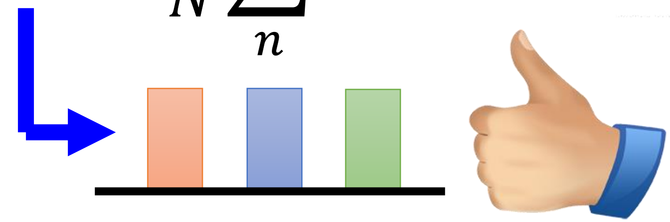
Objective Evaluation



$P(y|x)$



$$P(y) = \frac{1}{N}\sum_n P(y^n|x^n)$$



Inception Score [Tim Salimans, et al., NIPS 2016]

$$= \sum_x \sum_y \underline{P(y|x)logP(y|x)}$$

Negative entropy of P(y|x)

$$\underline{-\sum_y P(y)logP(y)}$$

Entropy of P(y)

Outline of Part 1

Generation

Conditional Generation

Unsupervised Conditional Generation

Relation to Reinforcement Learning

- ***Original Generator***

$P_G(x) \rightarrow P_{data}(x)$

Normal Distribution z → G → $x = G(z)$

- ***Conditional Generator***

$P_G(x|c) \rightarrow P_{data}(x|c)$

condition c → G

Normal Distribution z → G → $x = G(c, z)$

[Mehdi Mirza, et al., arXiv, 2014]

e.g. Text-to-Image

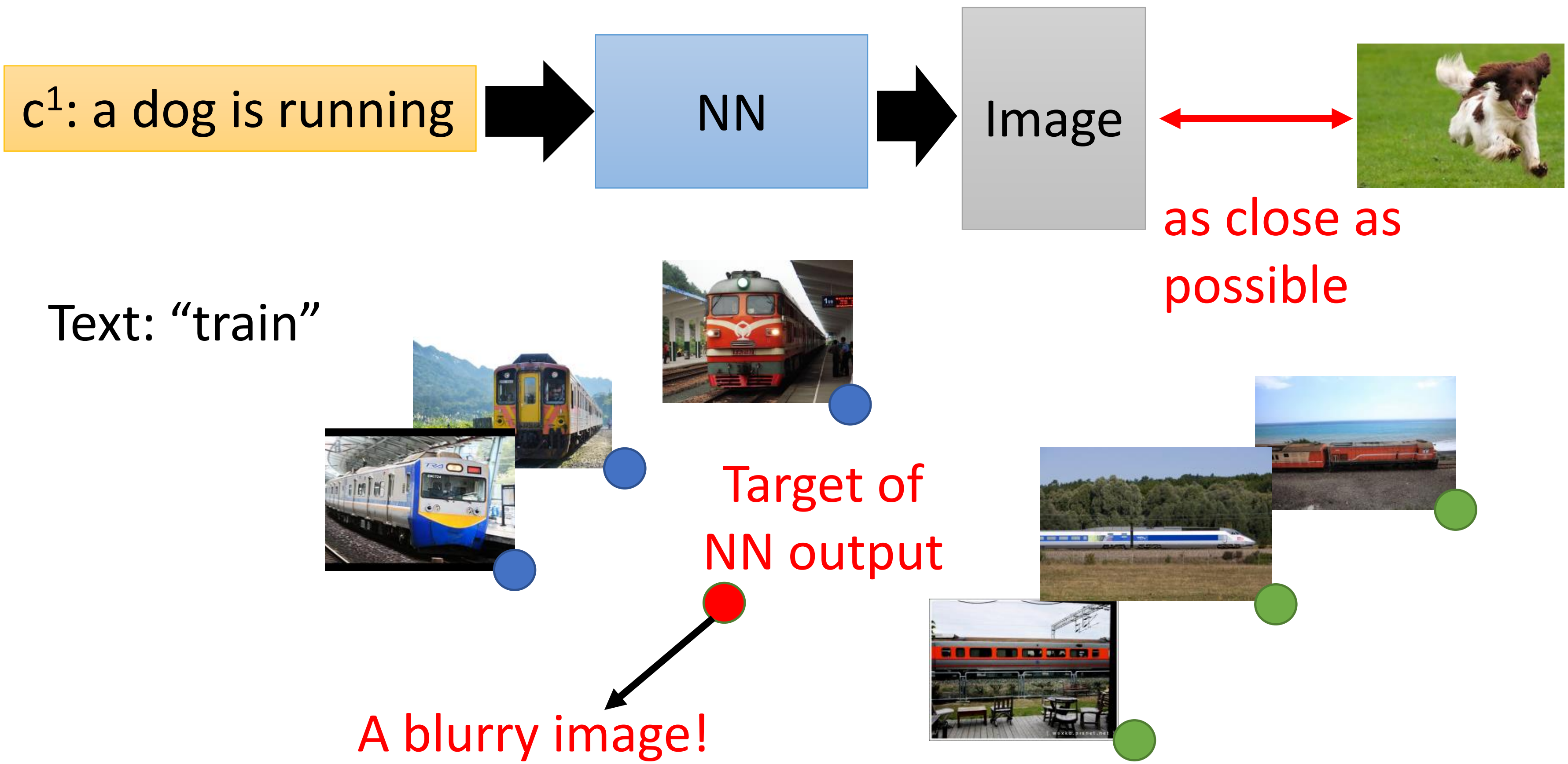
"Girl with red hair and red eyes"

"Girl with yellow ribbon"

→ NN Generator →


Text-to-Image

- **Traditional supervised approach**



[Scott Reed, et al, ICML, 2016]

Conditional GAN

c: train

Normal distribution z

G

Image

x = G(c,z)

x is real image or not

x

D (original)

scalar

Real images:  1

Generated images: Image 0

Generator will learn to generate realistic images ....

But completely ignore the input conditions.

[Scott Reed, et al, ICML, 2016]

Conditional GAN

c: train

Normal distribution z

G

Image

x = G(c,z)

c

x

D
(better)

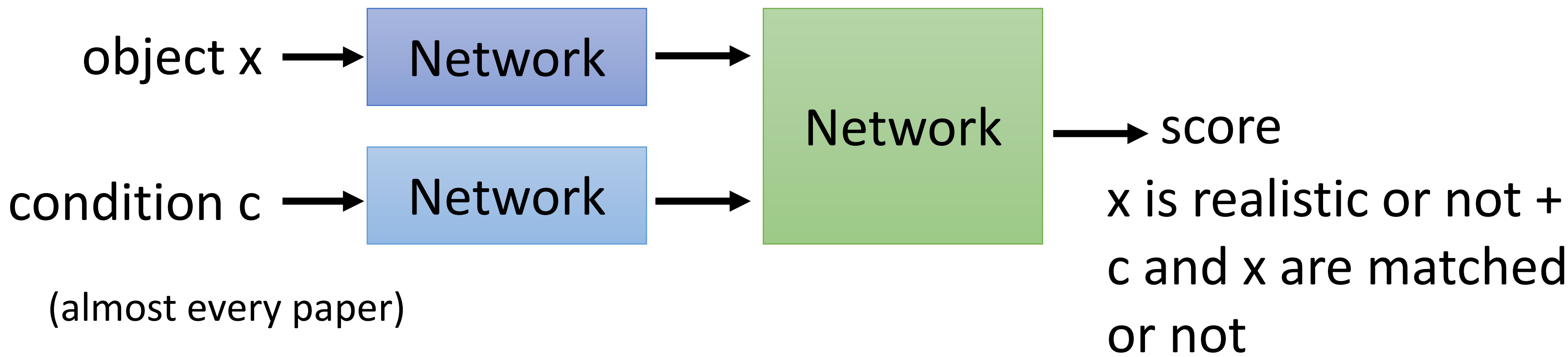
scalar

x is realistic or not +
c and x are matched or not

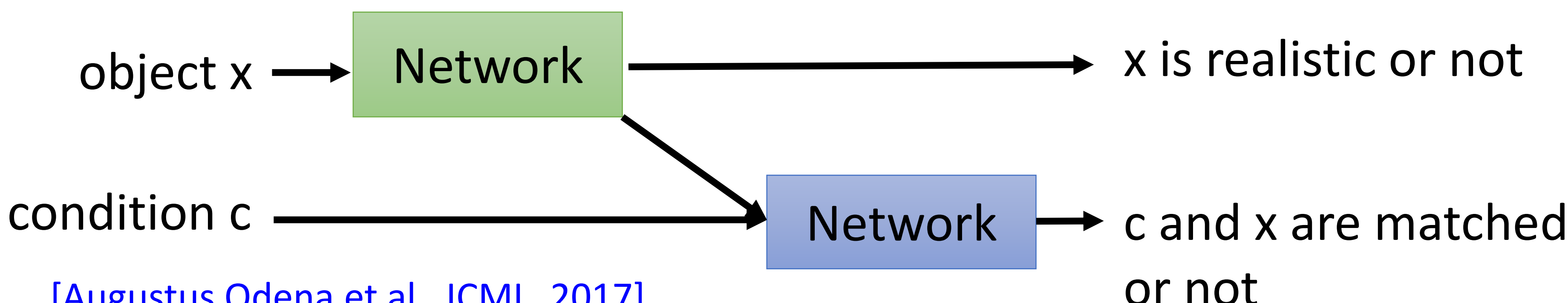
True text-image pairs: (train , ) 1

(cat , ) 0 (train , Image) 0

Conditional GAN - Discriminator



(almost every paper)



[Augustus Odena et al., ICML, 2017]

[Takeru Miyato, et al., ICLR, 2018]

[Han Zhang, et al., arXiv, 2017]

Conditional GAN

The images are generated by Yen-Hao Chen, Po-Chun Chien, Jun-Chen Xie, Tsung-Han Wu.

paired data

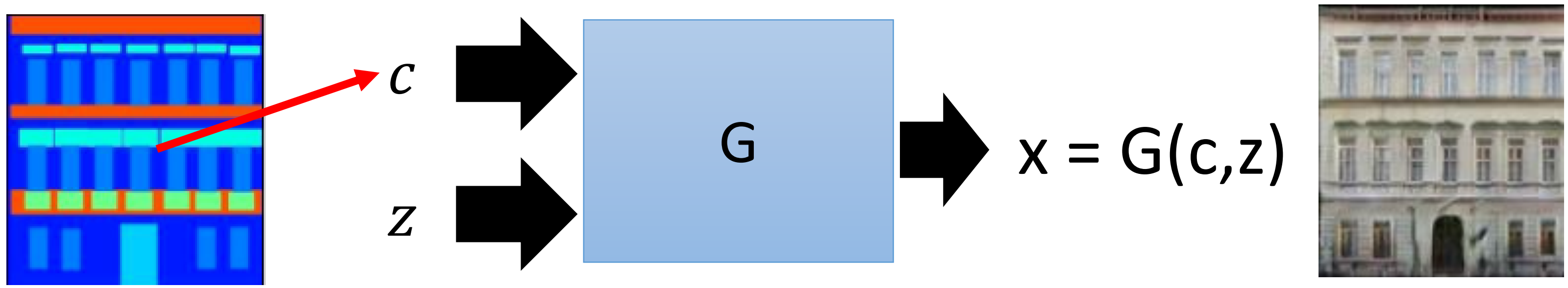
blue eyes
red hair
short hair

Collecting anime faces and the description of its characteristics

red hair, green eyes

blue hair, red eyes

Conditional GAN - Image-to-image



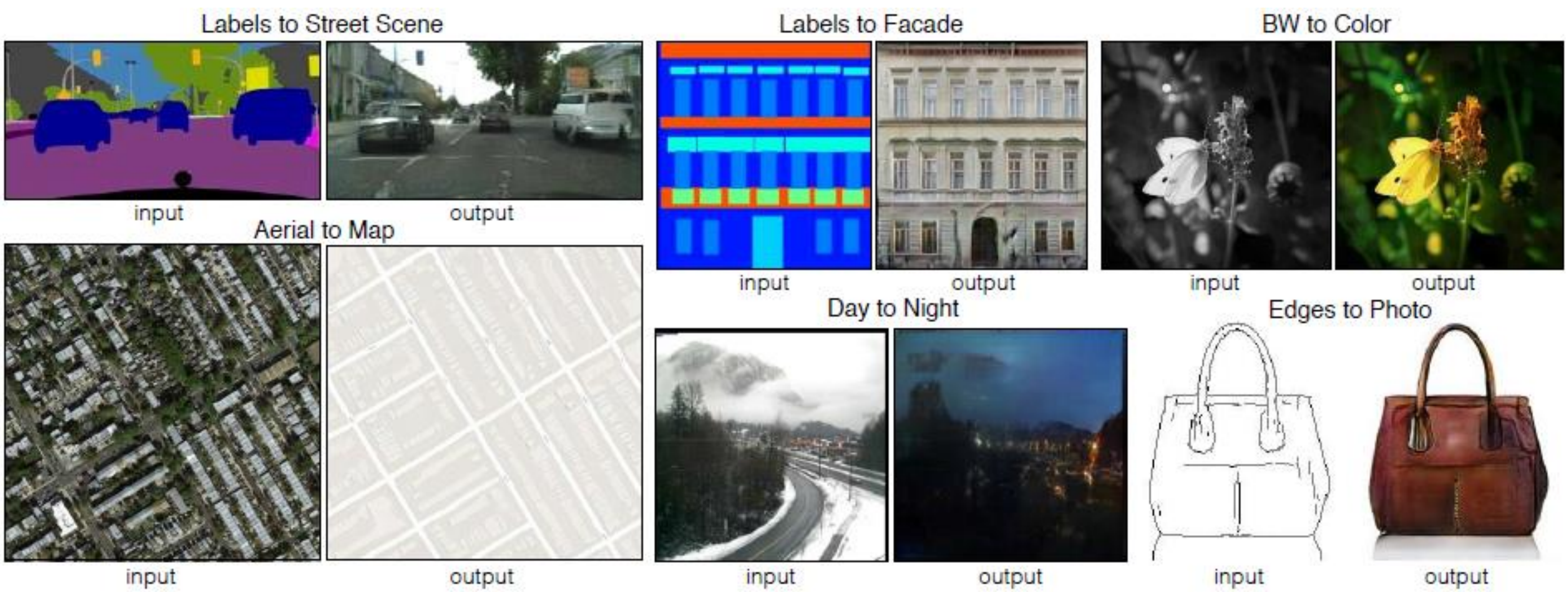
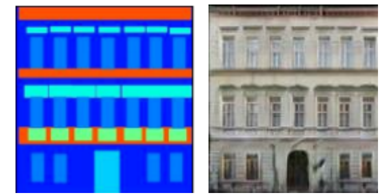


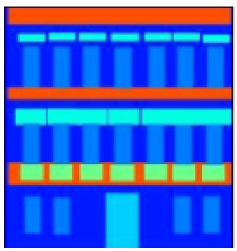
Image translation, or pix2pix

[Phillip Isola, et al., CVPR, 2017]

Conditional GAN - Image-to-image



- **Traditional supervised approach**



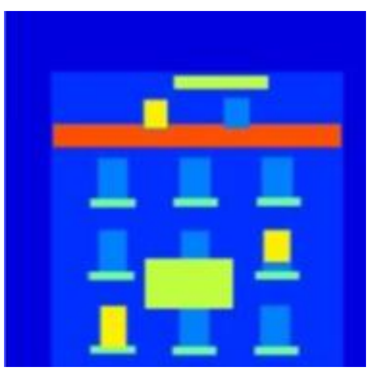
NN

Image

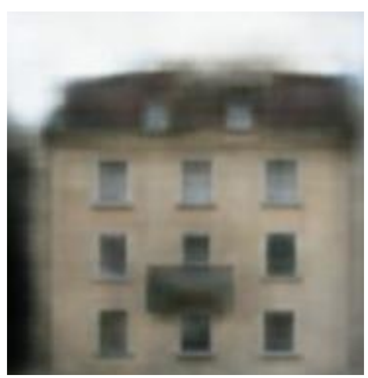
as close as possible

e.g. L1

Testing:



input

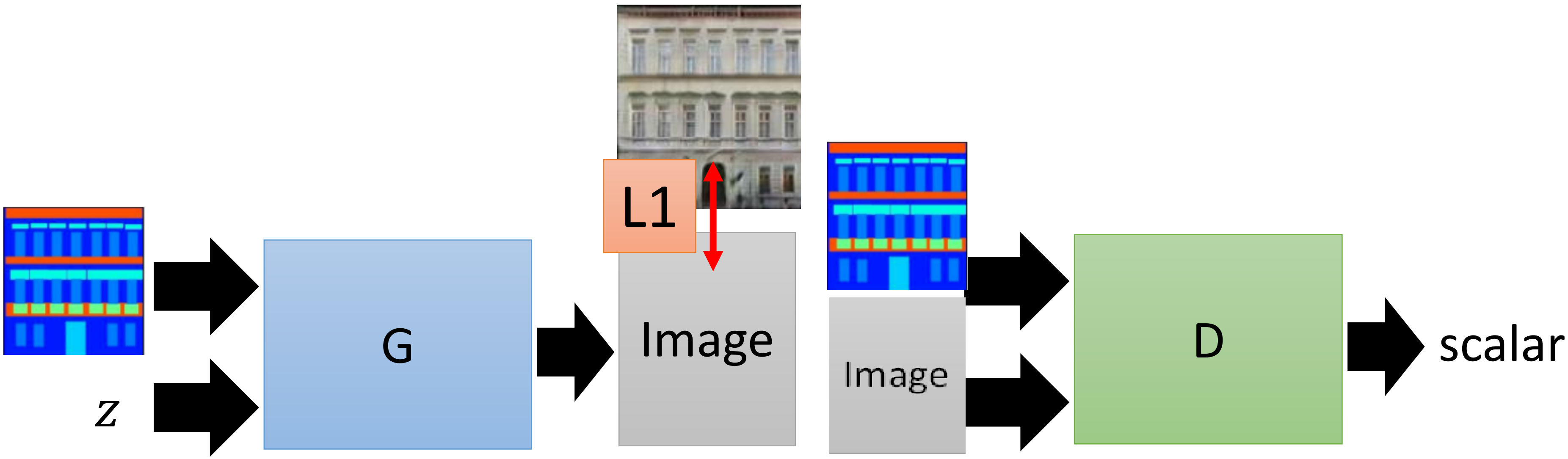


L1

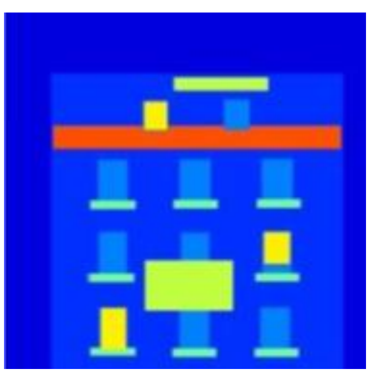
It is blurry.

[Phillip Isola, et al., CVPR, 2017]

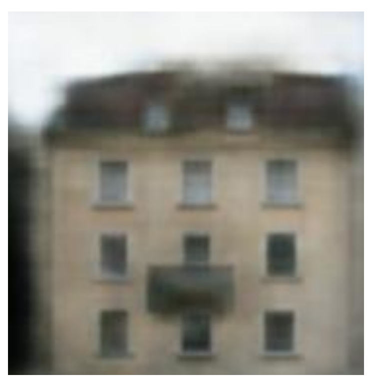
Conditional GAN - Image-to-image



Testing:



input



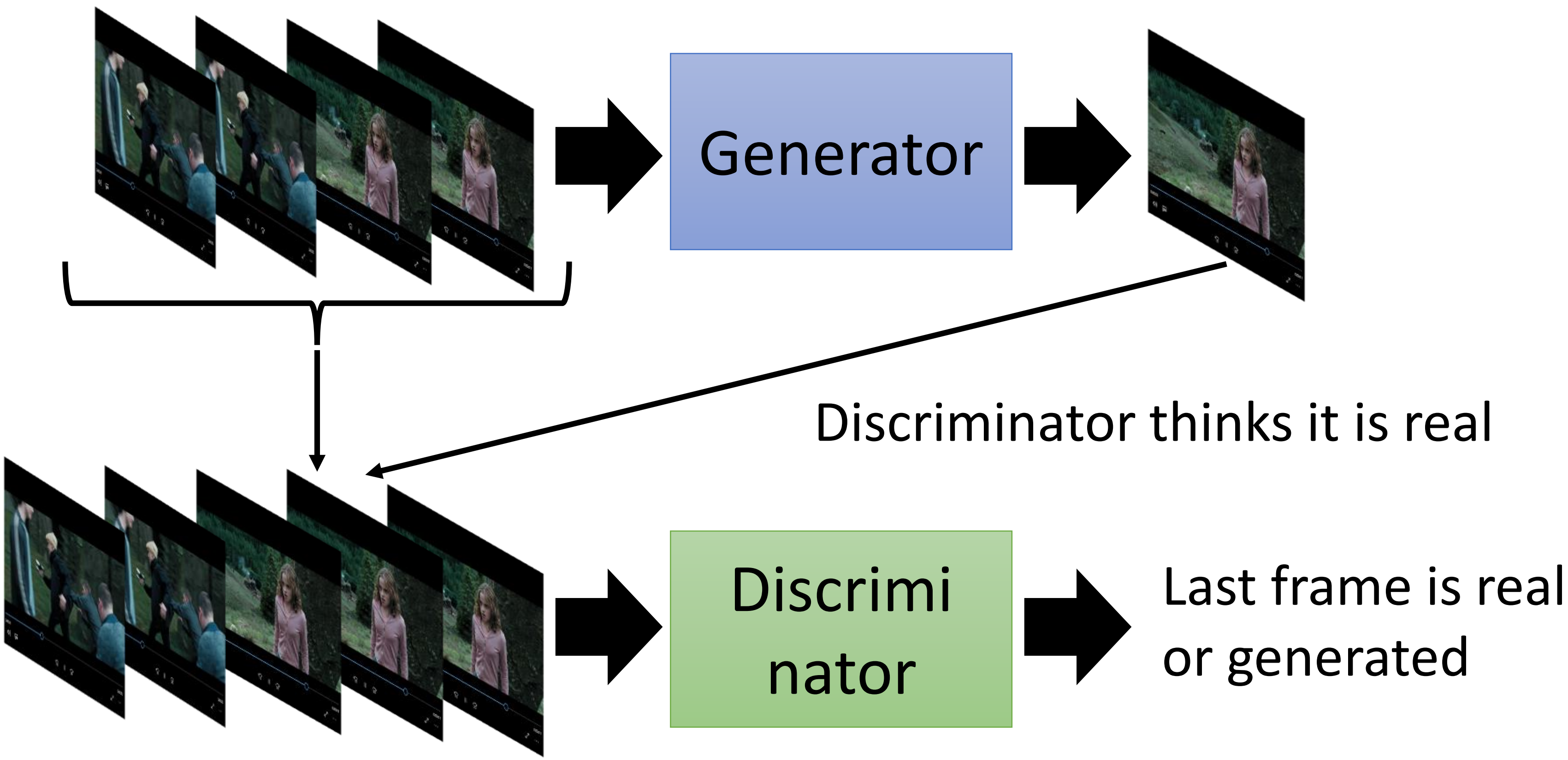
L1

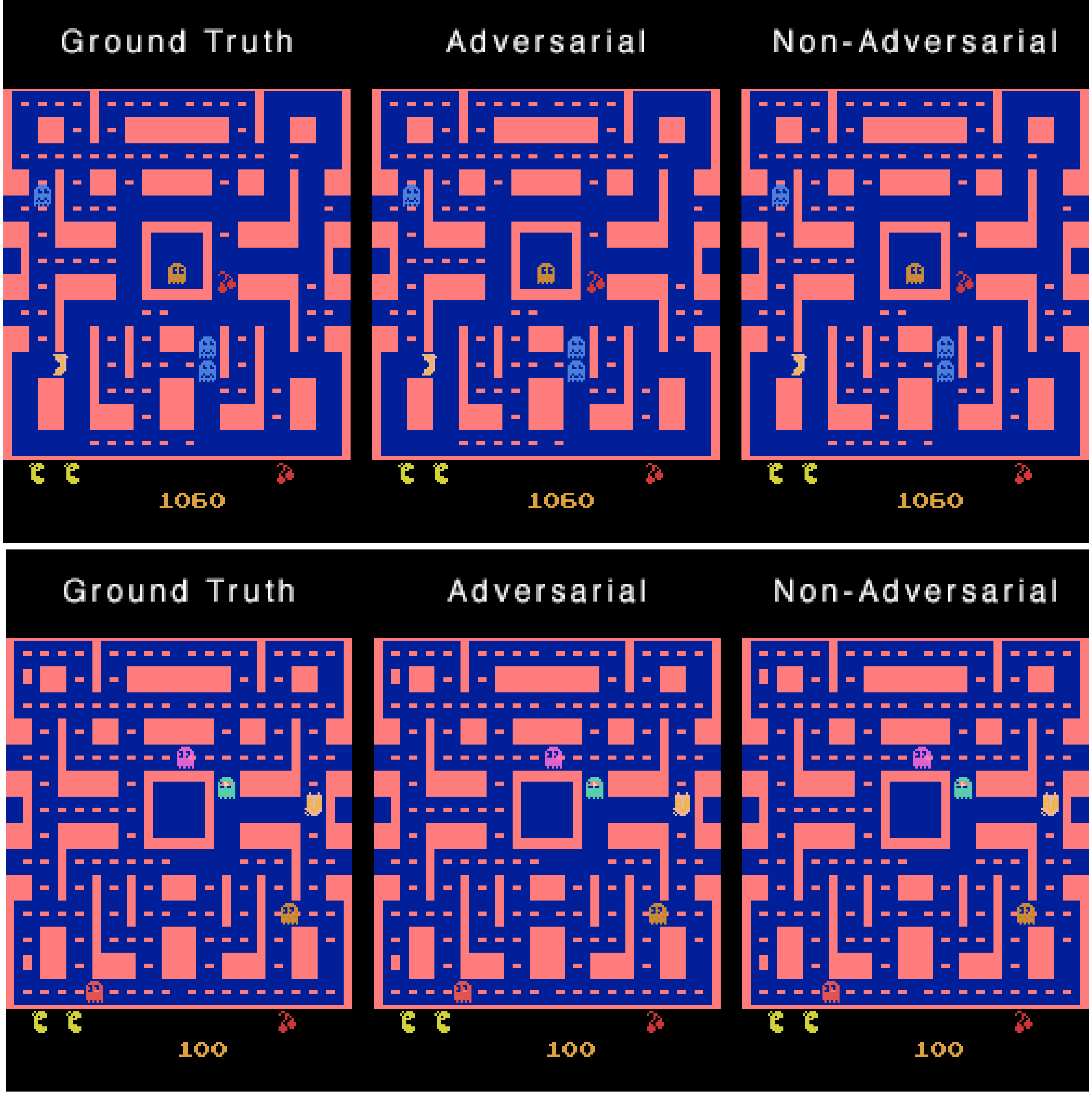
GAN

GAN + L1

Conditional GAN
- Video Generation

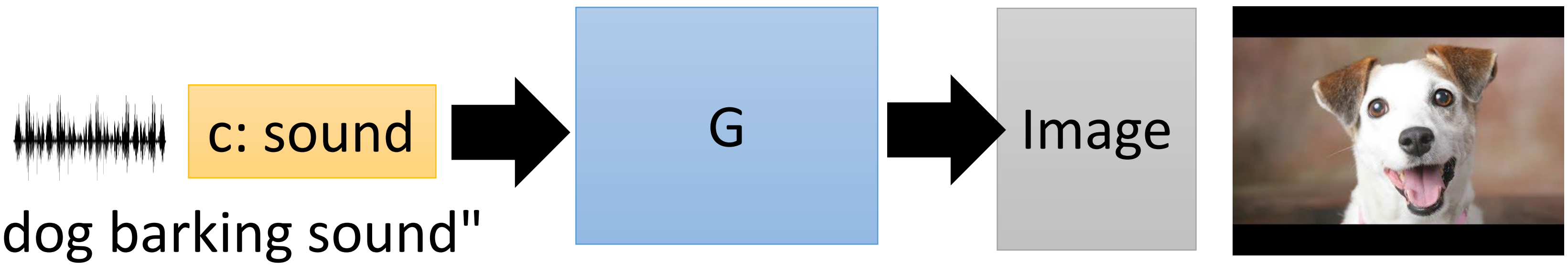
[Michael Mathieu, et al., arXiv, 2015]





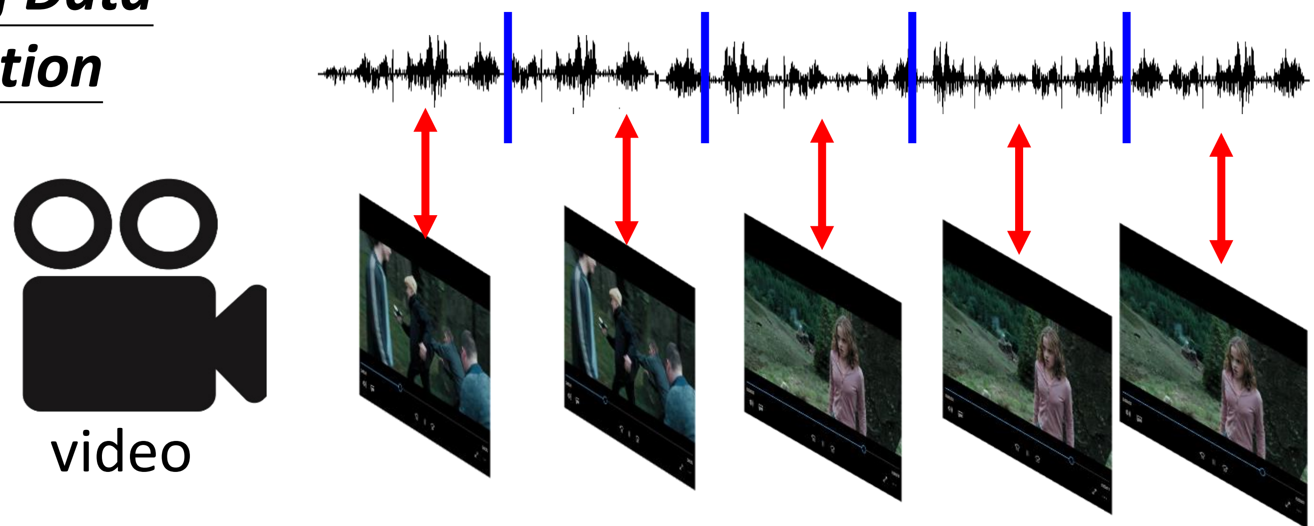
https://github.com/dyelax/Adversarial_Video_Generation

Conditional GAN
- Sound-to-image



"a dog barking sound"

Training Data Collection

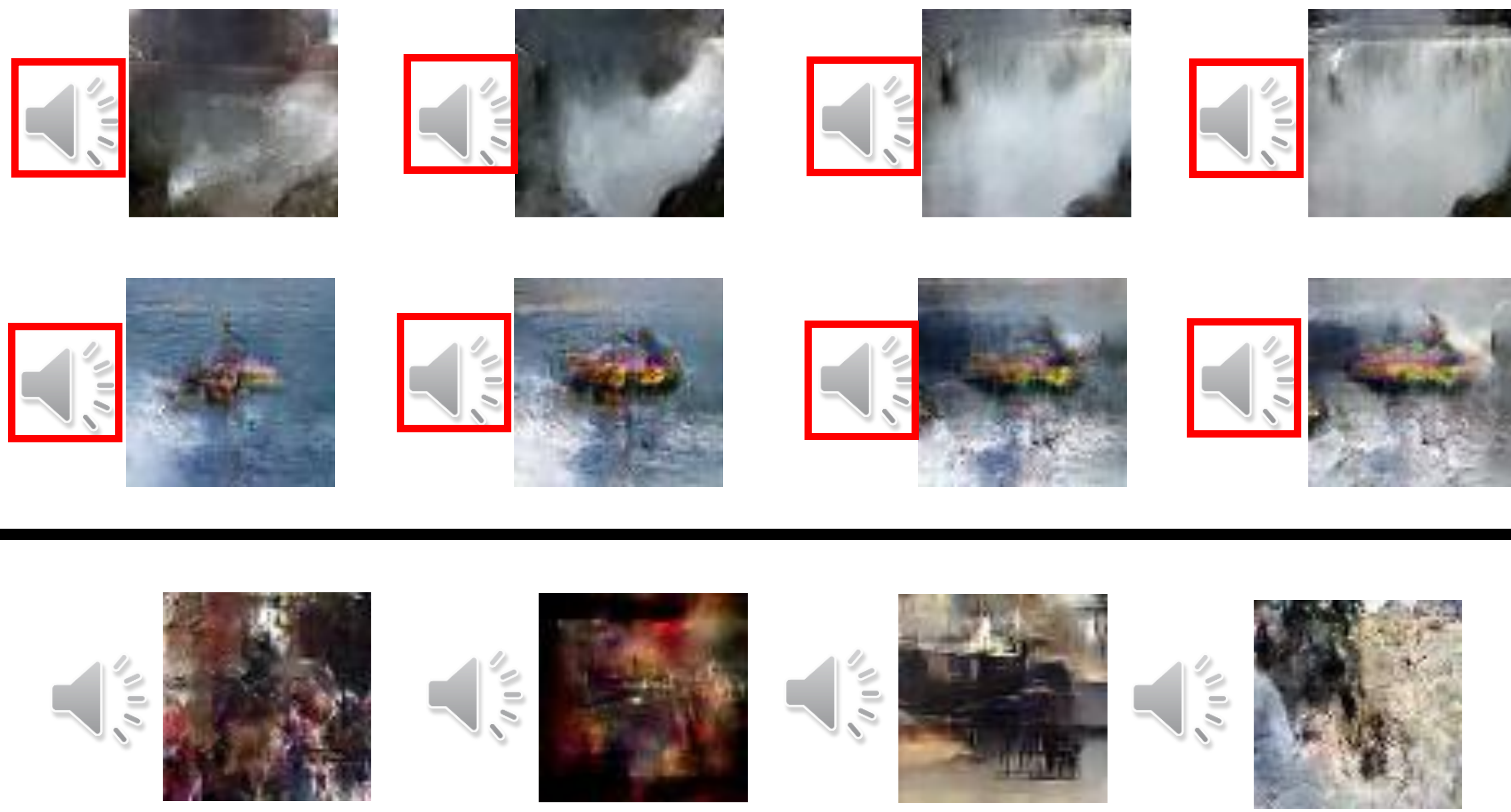


Conditional GAN
- Sound-to-image

The images are generated by Chia-Hung Wan and Shun-Po Chuang.
https://wjohn1483.github.io/audio_to_scene/index.html

- Audio-to-image

Louder



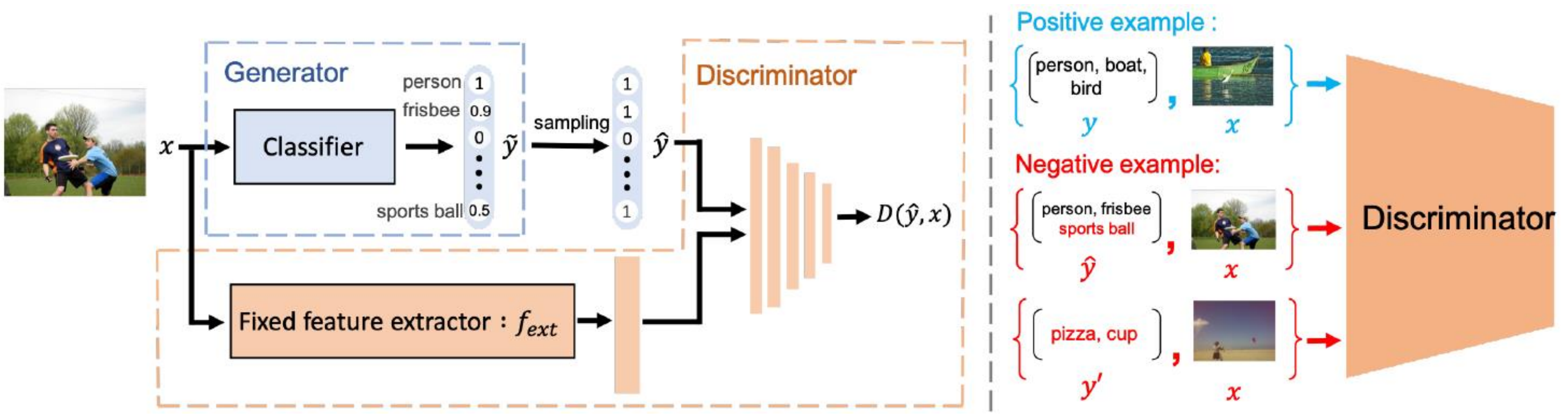
Conditional GAN - Image-to-label

Multi-label Image Classifier

person, sports ball,
baseball bat, baseball glove

Input condition

Generated output



Conditional GAN - Image-to-label

The classifiers can have different architectures.

The classifiers are trained as conditional GAN.

[Tsai, et al., submitted to ICASSP 2019]

F1	MS-COCO	NUS-WIDE
VGG-16	56.0	33.9
+ GAN	60.4	41.2
Inception	62.4	53.5
+GAN	63.8	55.8
Resnet-101	62.8	53.1
+GAN	64.0	55.4
Resnet-152	63.3	52.1
+GAN	63.9	54.1
Att-RNN	62.1	54.7
RLSD	62.0	46.9

Conditional GAN - Image-to-label

The classifiers can have different architectures.

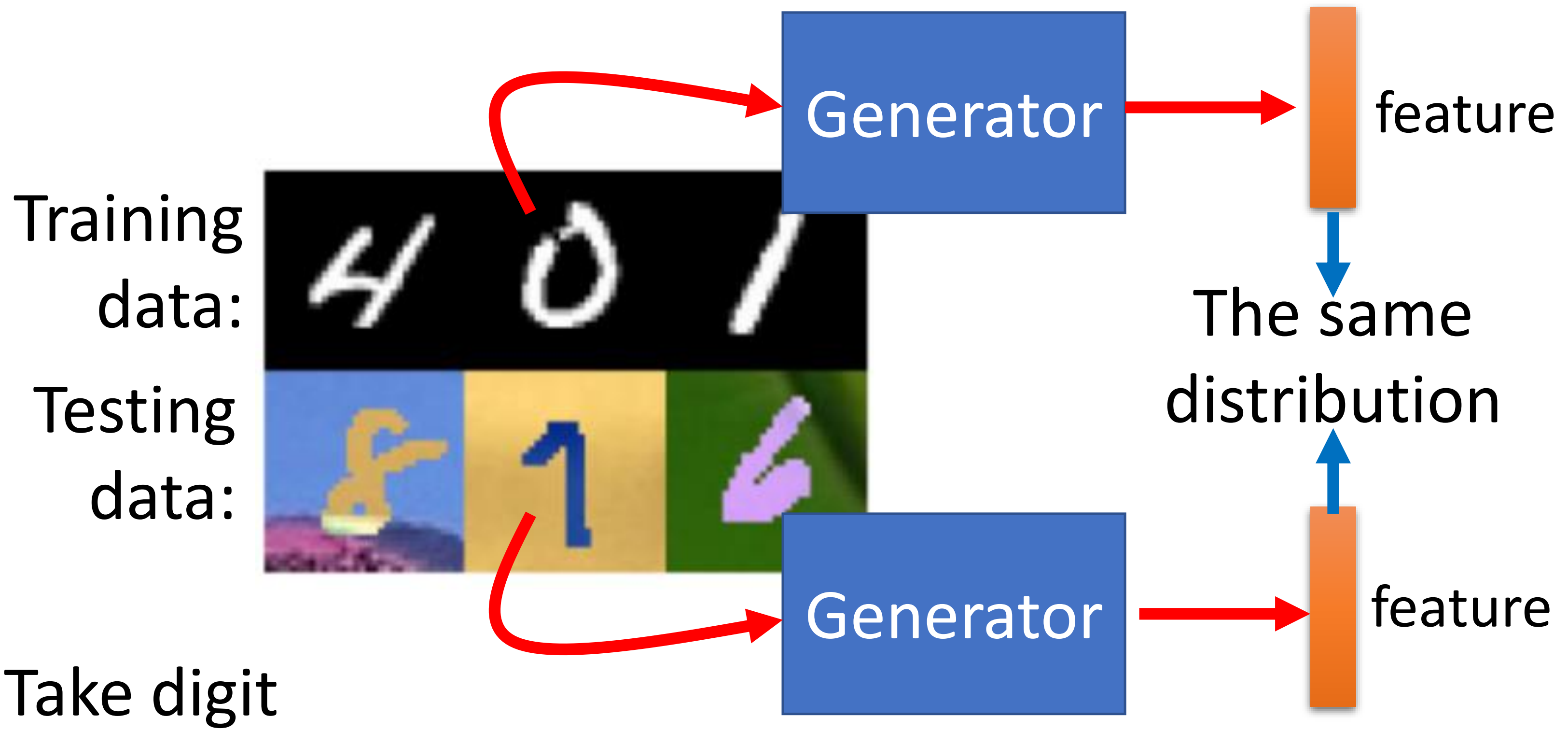
The classifiers are trained as conditional GAN.

Conditional GAN outperforms other models designed for multi-label.

F1	MS-COCO	NUS-WIDE
VGG-16	56.0	33.9
+ GAN	60.4	41.2
Inception	62.4	53.5
+GAN	63.8	55.8
Resnet-101	62.8	53.1
+GAN	64.0	55.4
Resnet-152	63.3	52.1
+GAN	63.9	54.1
Att-RNN	62.1	54.7
RLSD	62.0	46.9

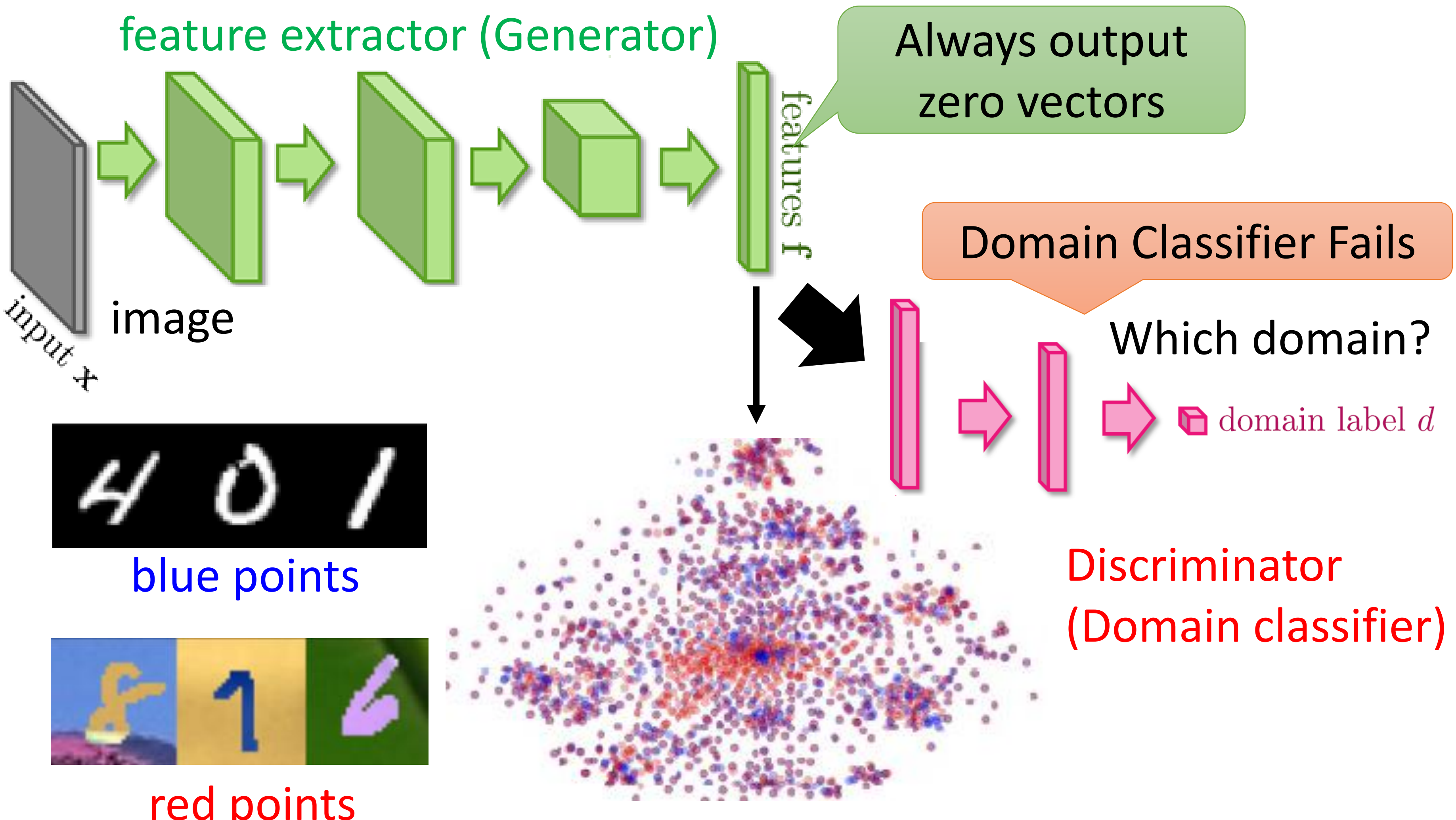
Domain Adversarial Training

- Training and testing data are in different domains



Take digit classification as example

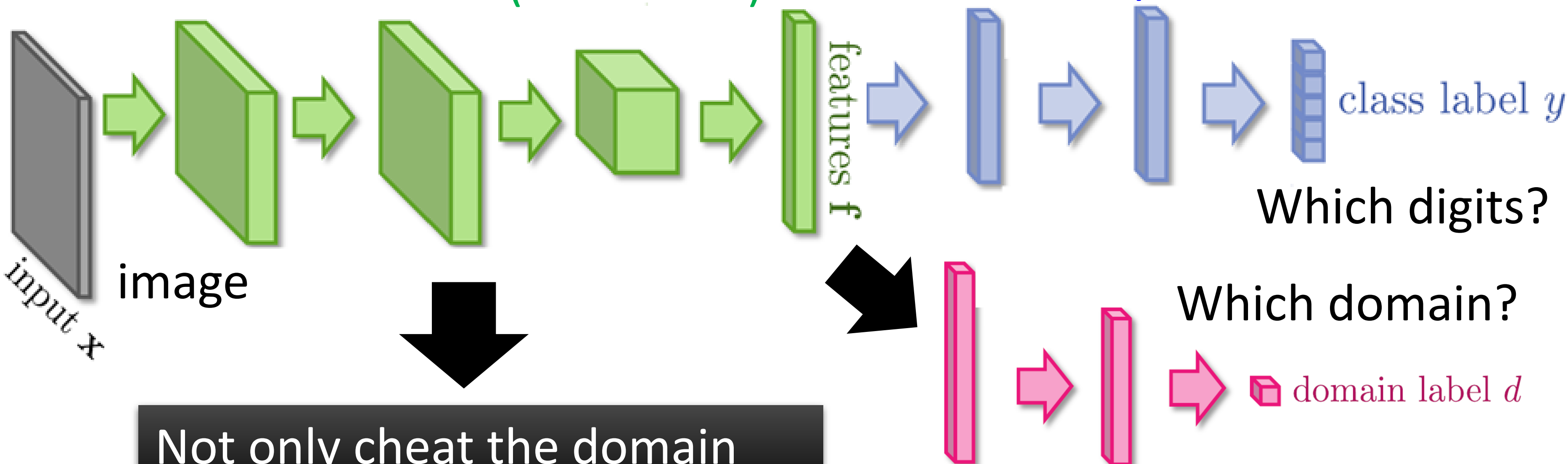
Domain Adversarial Training



Domain Adversarial Training

feature extractor (Generator)

Label predictor



image

Which digits?

Which domain?

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

Discriminator (Domain classifier)

Successfully applied on image classification

[Ganin et al, ICML, 2015][Ajakan et al. JMLR, 2016]

More speech-related applications in Part II.

Outline of Part 1

Generation

Conditional Generation

Unsupervised Conditional Generation

Relation to Reinforcement Learning

Unsupervised Conditional GAN

Condition
Object in Domain X → G → Generated Object
Object in Domain Y

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

Domain X

photos

Not Paired

Domain Y

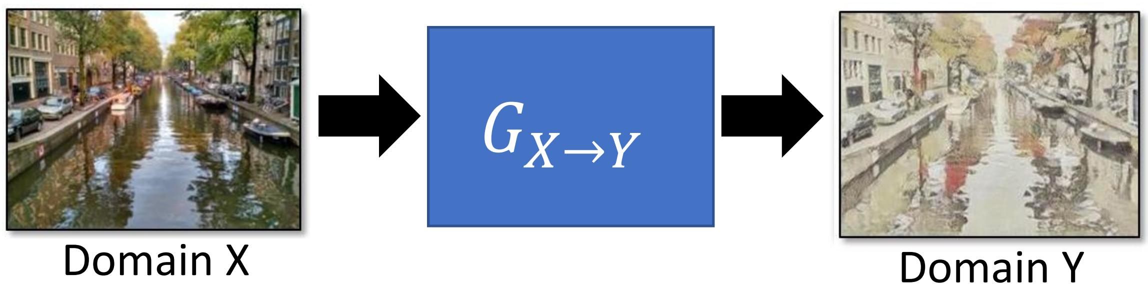
Vincent van Gogh’s paintings

Use image style transfer as example here

More Applications in Parts II and III

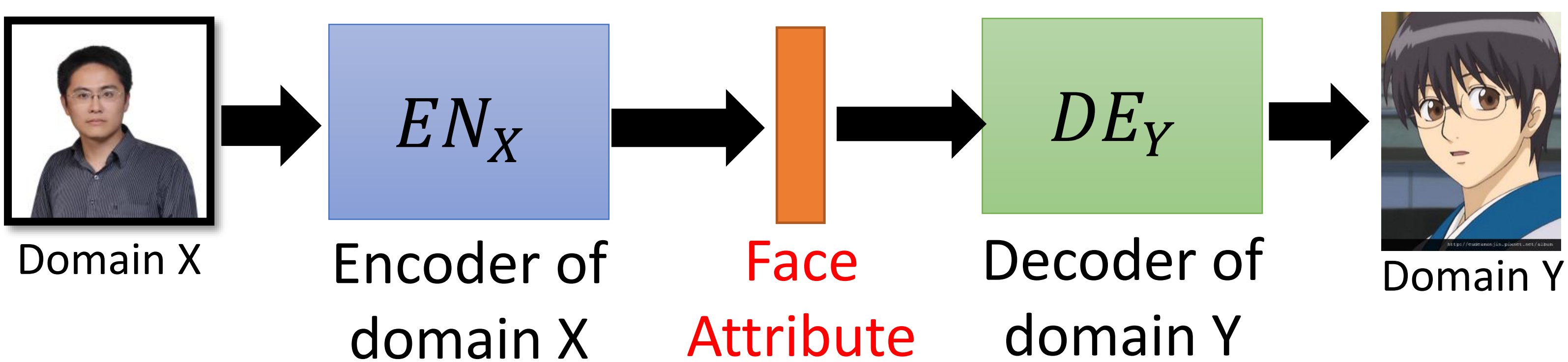
Unsupervised Conditional Generation

- Approach 1: Direct Transformation



For texture or color change

- Approach 2: Projection to Common Space



Larger change, only keep the semantics

Direct Transformation

Domain X

Domain Y

Domain X

$G_{X \rightarrow Y}$

Become similar to domain Y

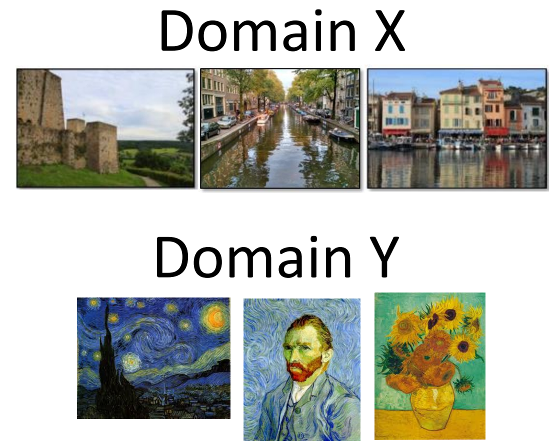
D_Y

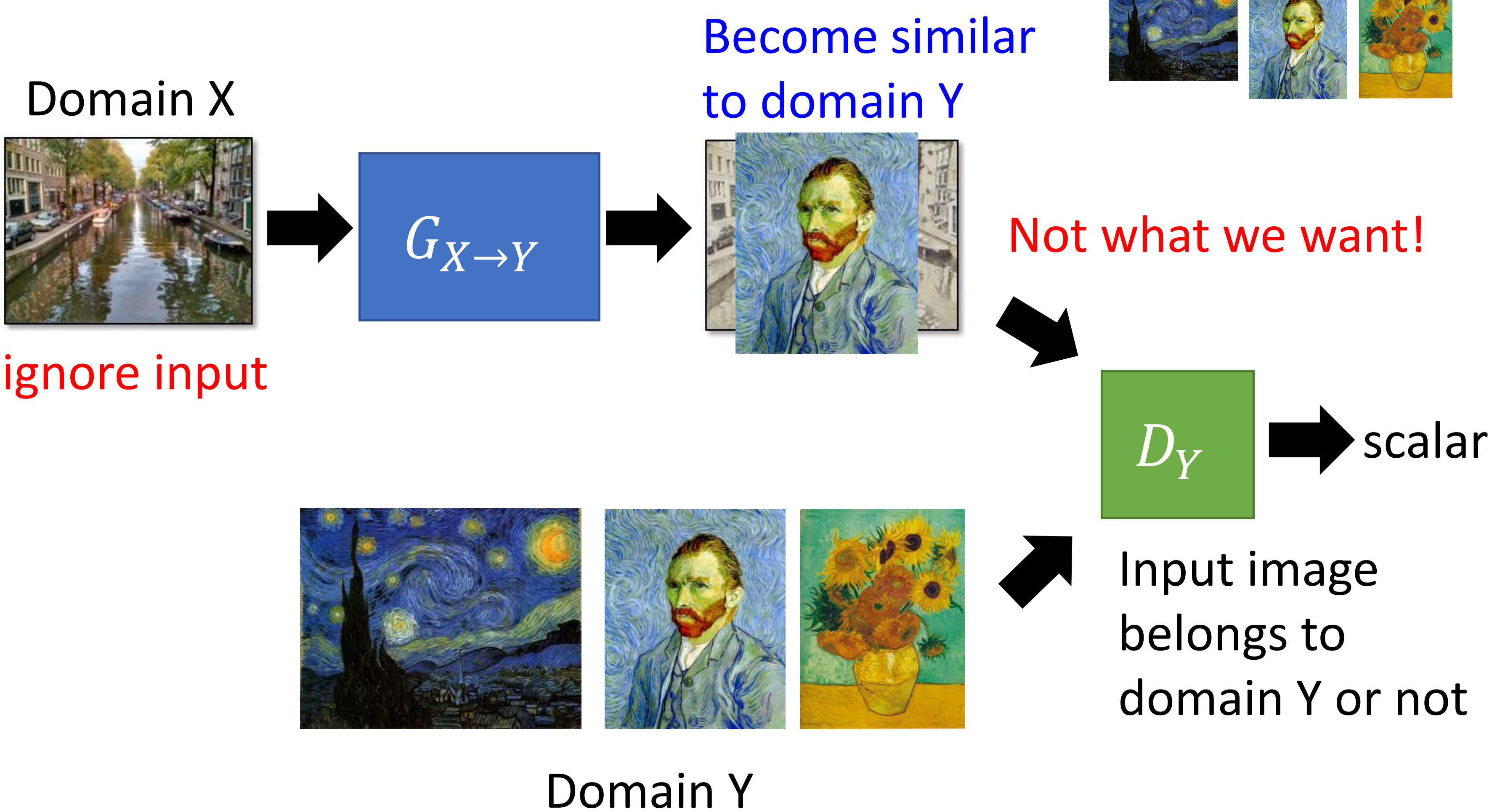
scalar

Input image belongs to domain Y or not

Domain Y

Direct Transformation





Direct Transformation

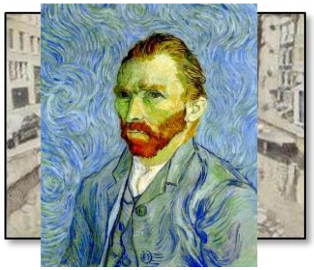
Domain X

Domain Y

Become similar to domain Y

Domain X

$G_{X \to Y}$



ignore input

Not what we want!

D_Y

scalar

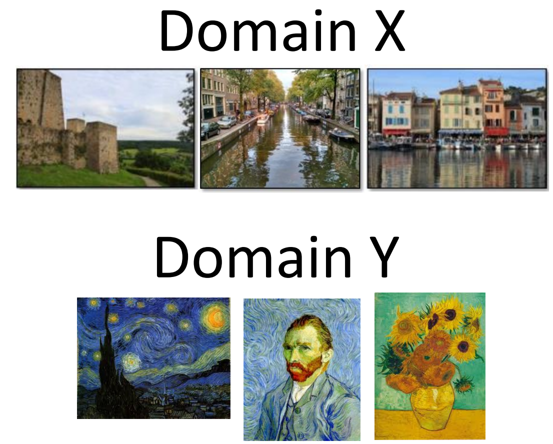
Input image belongs to domain Y or not

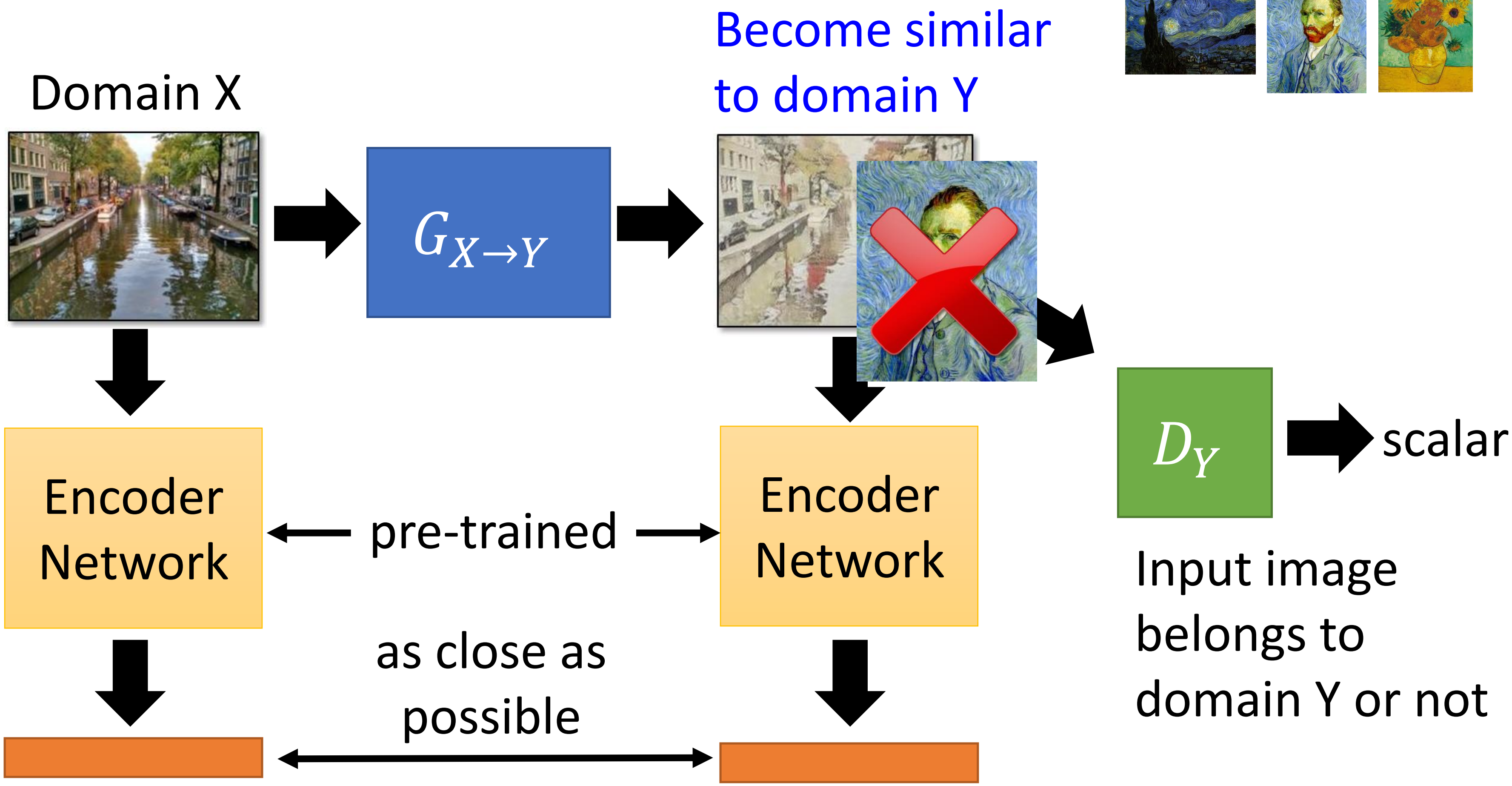
The issue can be avoided by network design.

Simpler generator makes the input and output more closely related.

[Tomer Galanti, et al. ICLR, 2018]

Direct Transformation

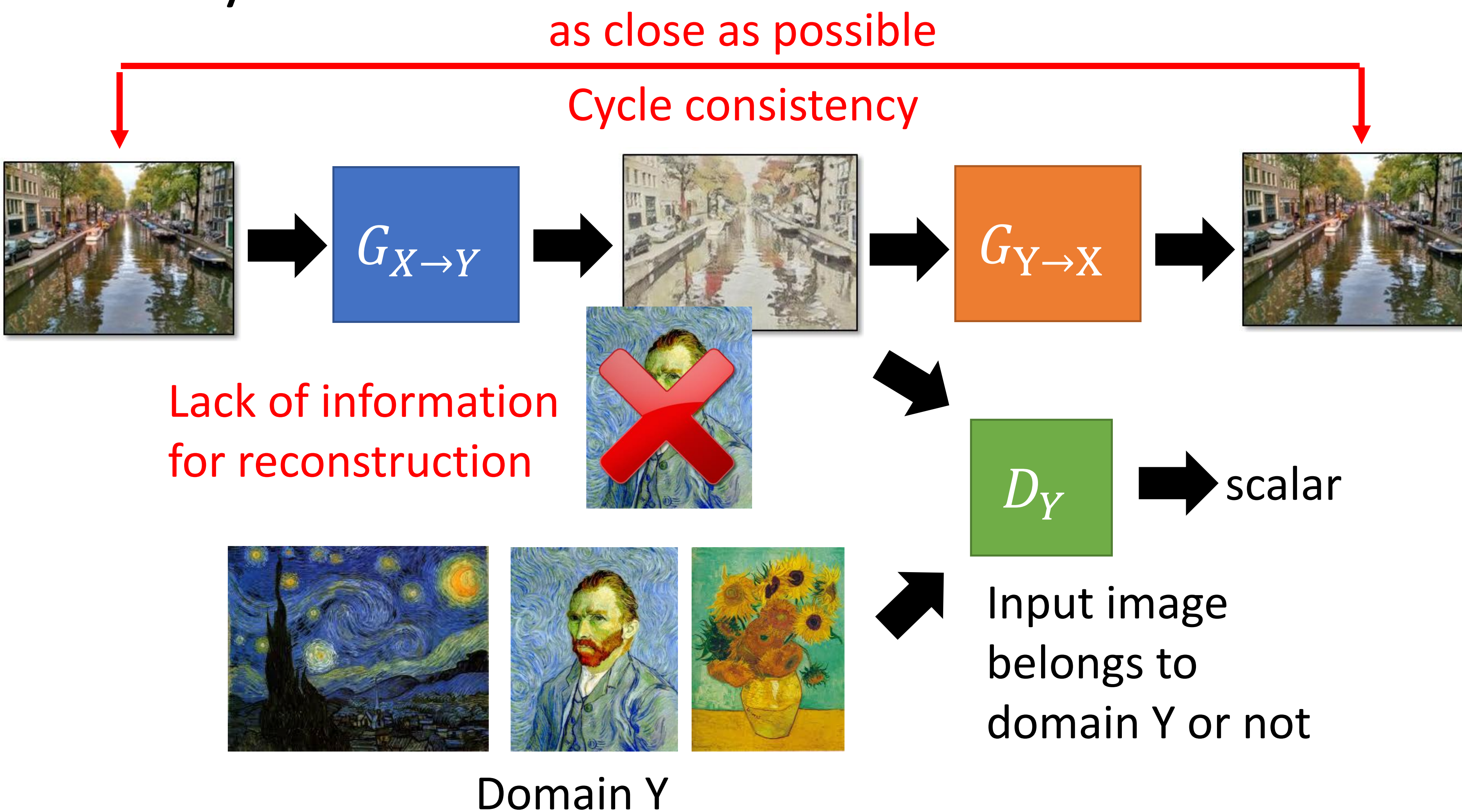




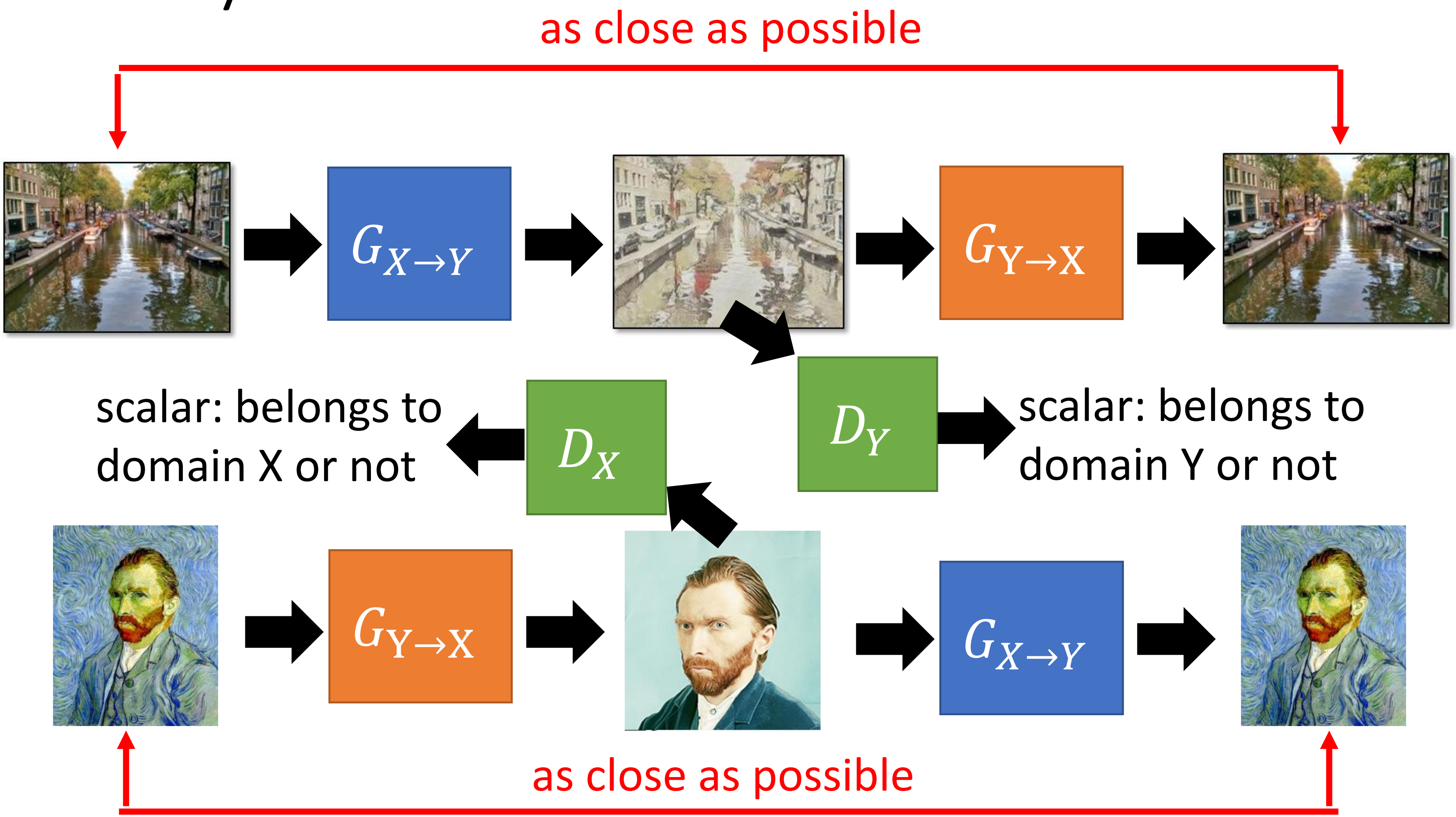
Baseline of DTN [Yaniv Taigman, et al., ICLR, 2017]

Direct Transformation – Cycle GAN

[Jun-Yan Zhu, et al., ICCV, 2017]



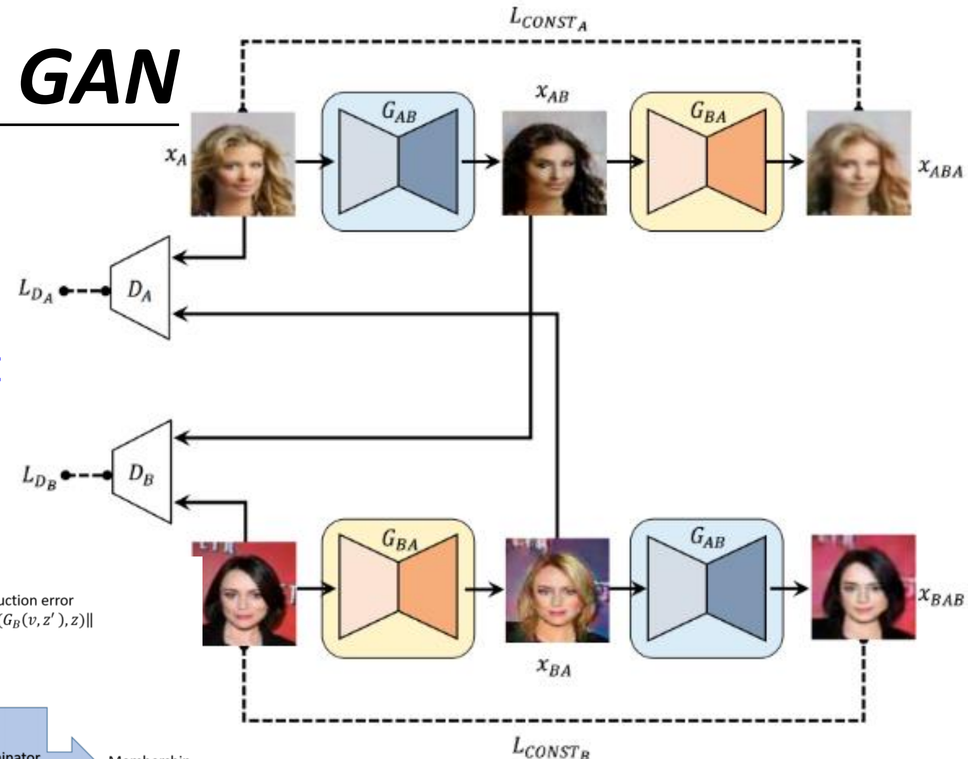
Direct Transformation – Cycle GAN



For multiple domains, considering starGAN

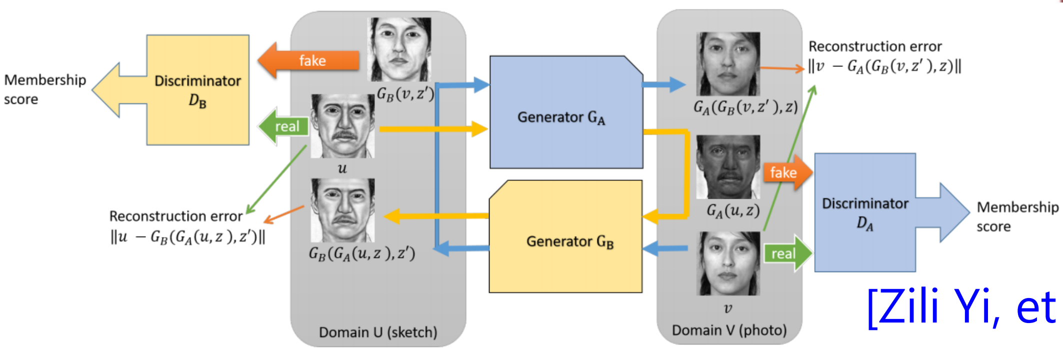
[Yunjey Choi, arXiv, 2017]

Disco GAN



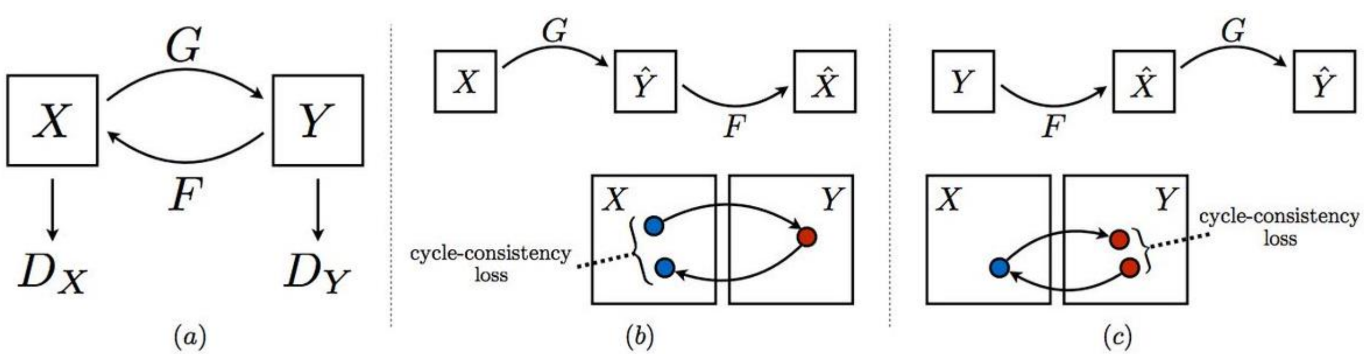
[Taeksoo Kim, et al., ICML, 2017]

Dual GAN



[Zili Yi, et al., ICCV, 2017]

Cycle GAN

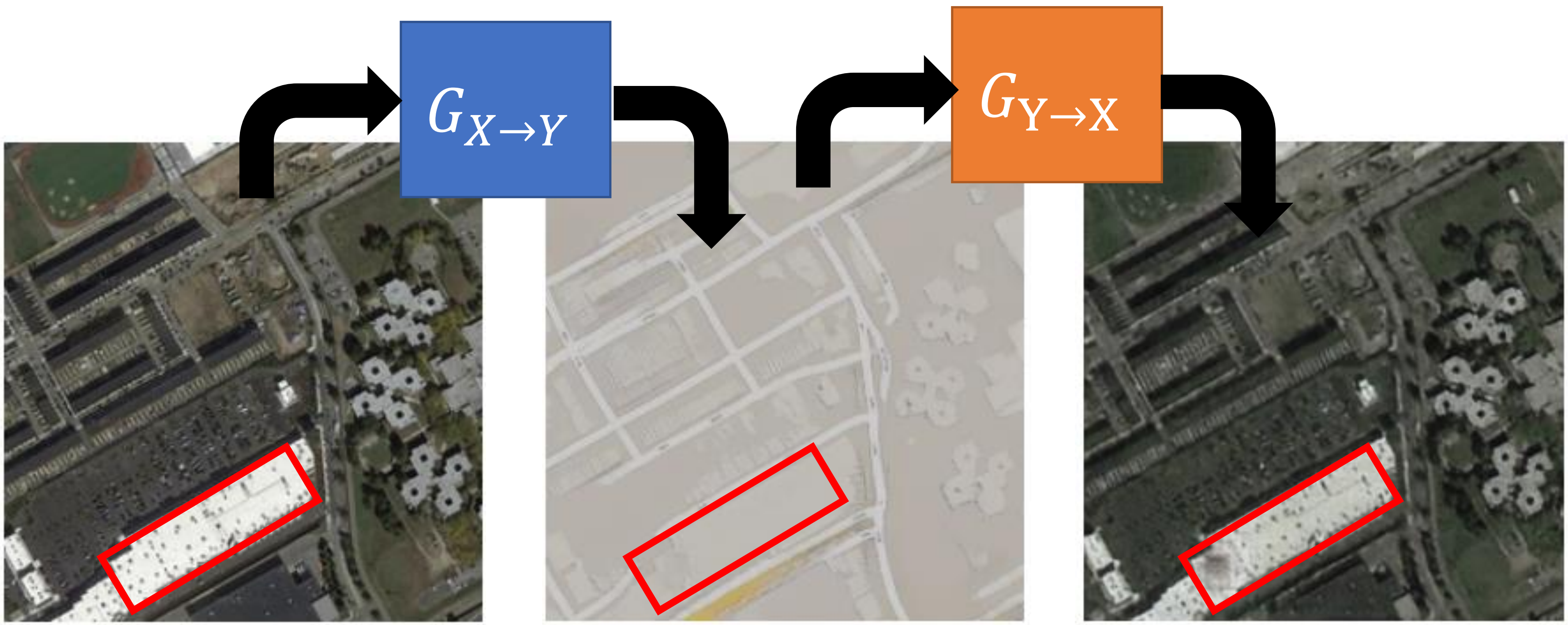


[Jun-Yan Zhu, et al., ICCV, 2017]

Issue of Cycle Consistency

- **CycleGAN: a Master of Steganography**

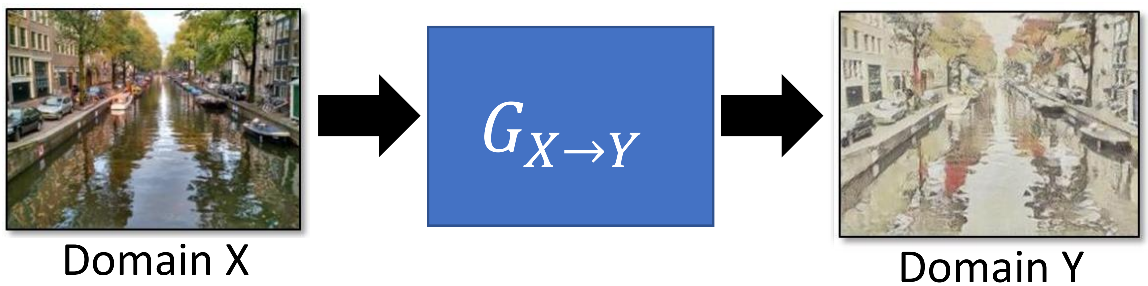
[Casey Chu, et al., NIPS workshop, 2017]



The information is hidden.

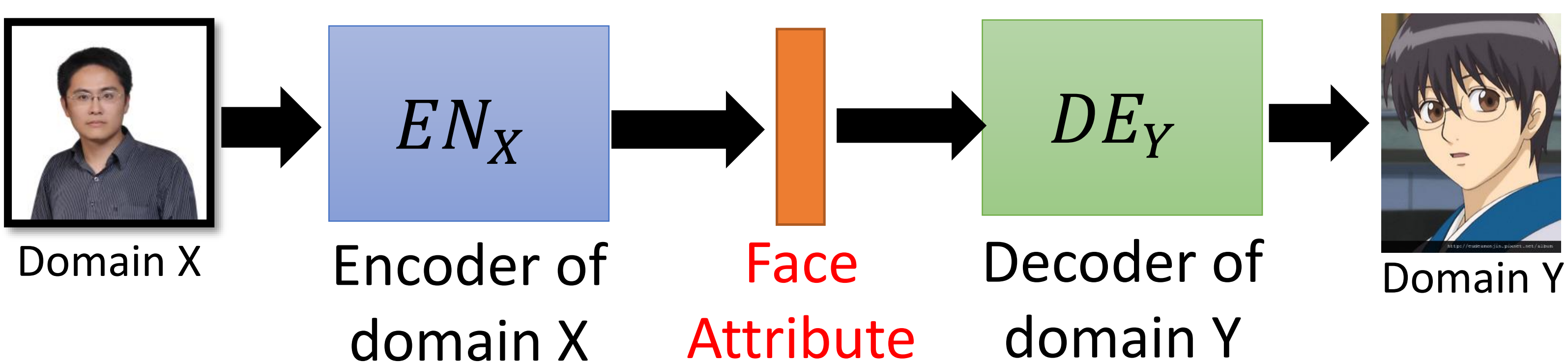
Unsupervised Conditional Generation

- Approach 1: Direct Transformation



For texture or color change

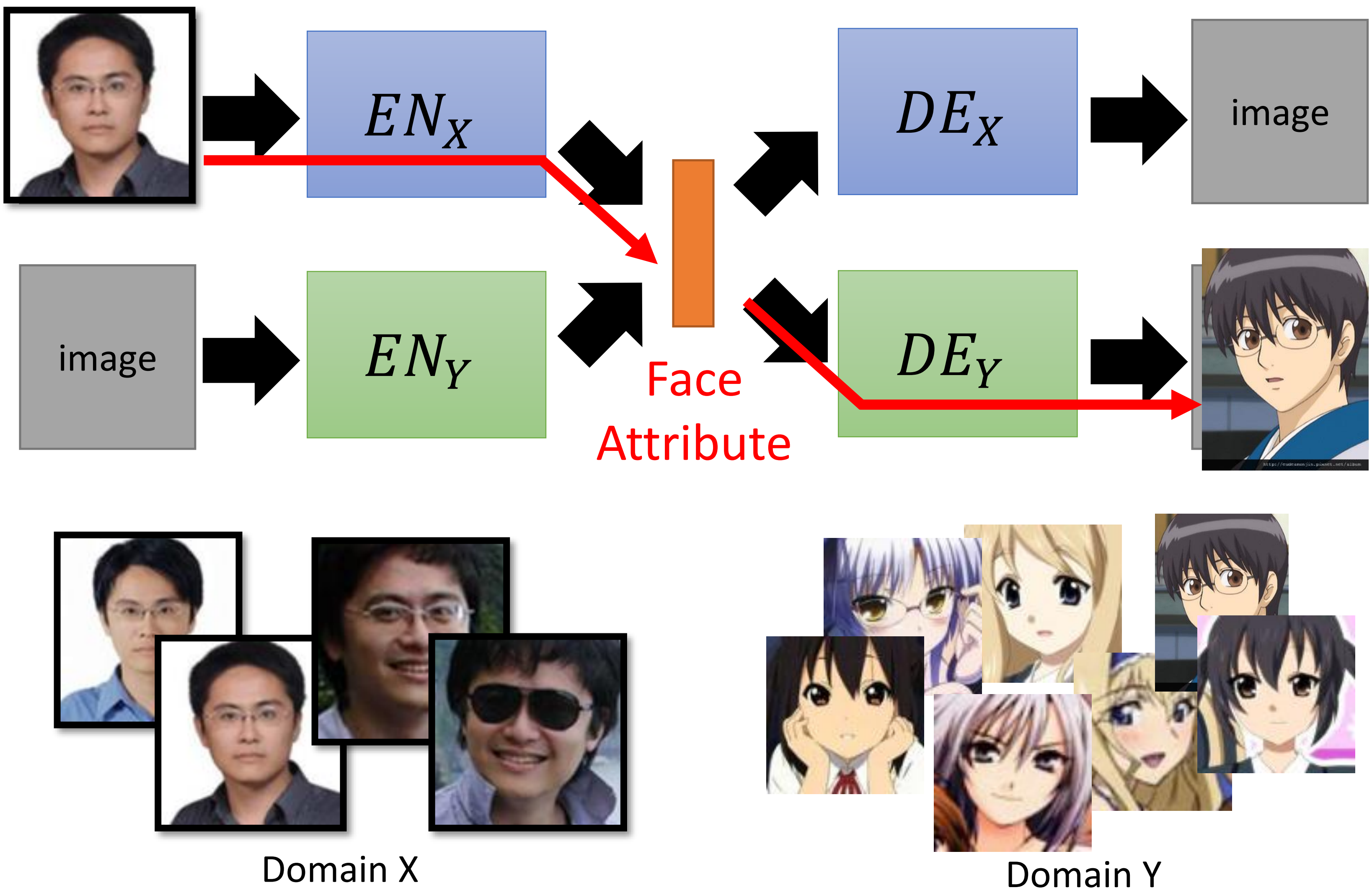
- Approach 2: Projection to Common Space



Larger change, only keep the semantics

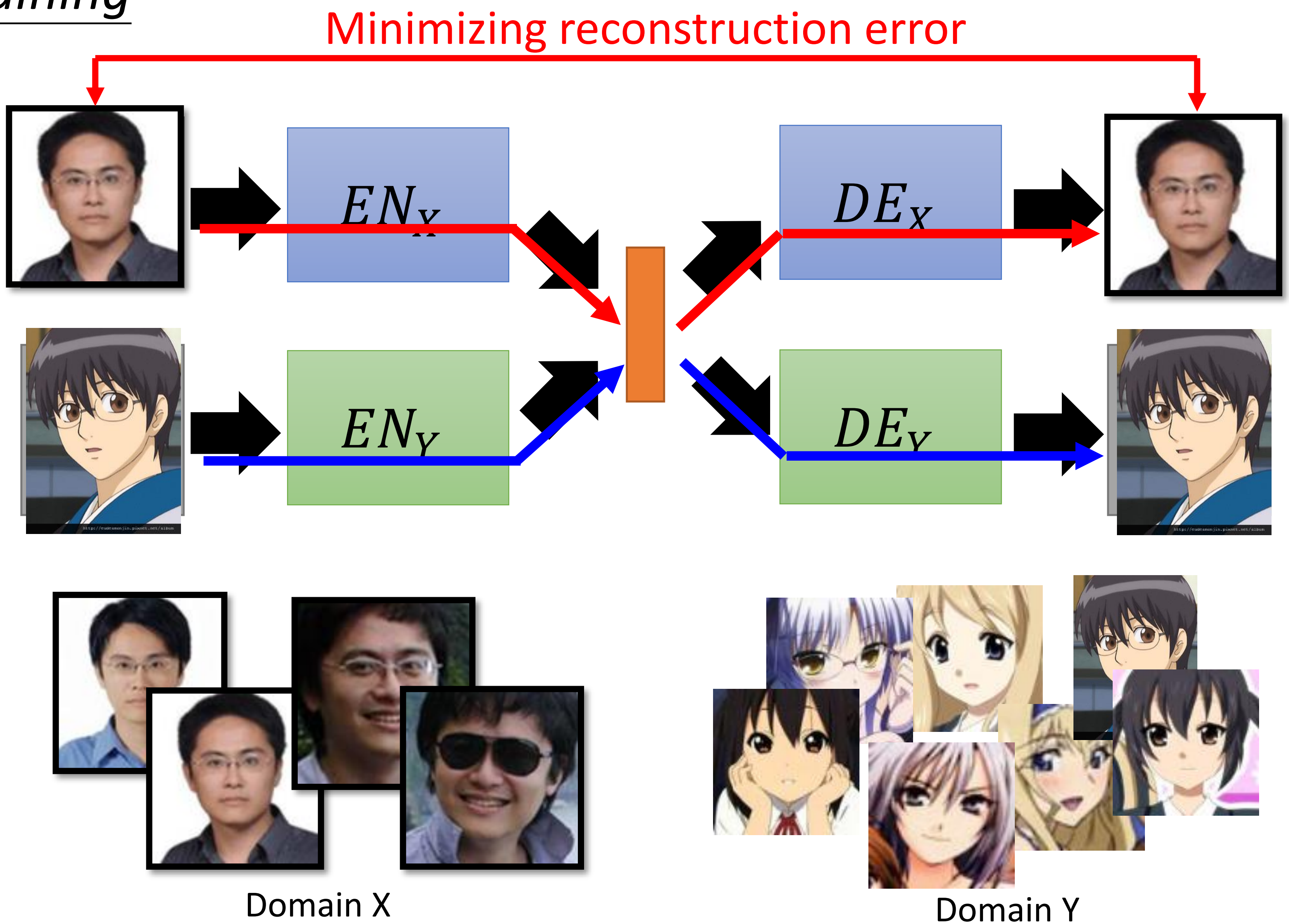
Projection to Common Space

Target



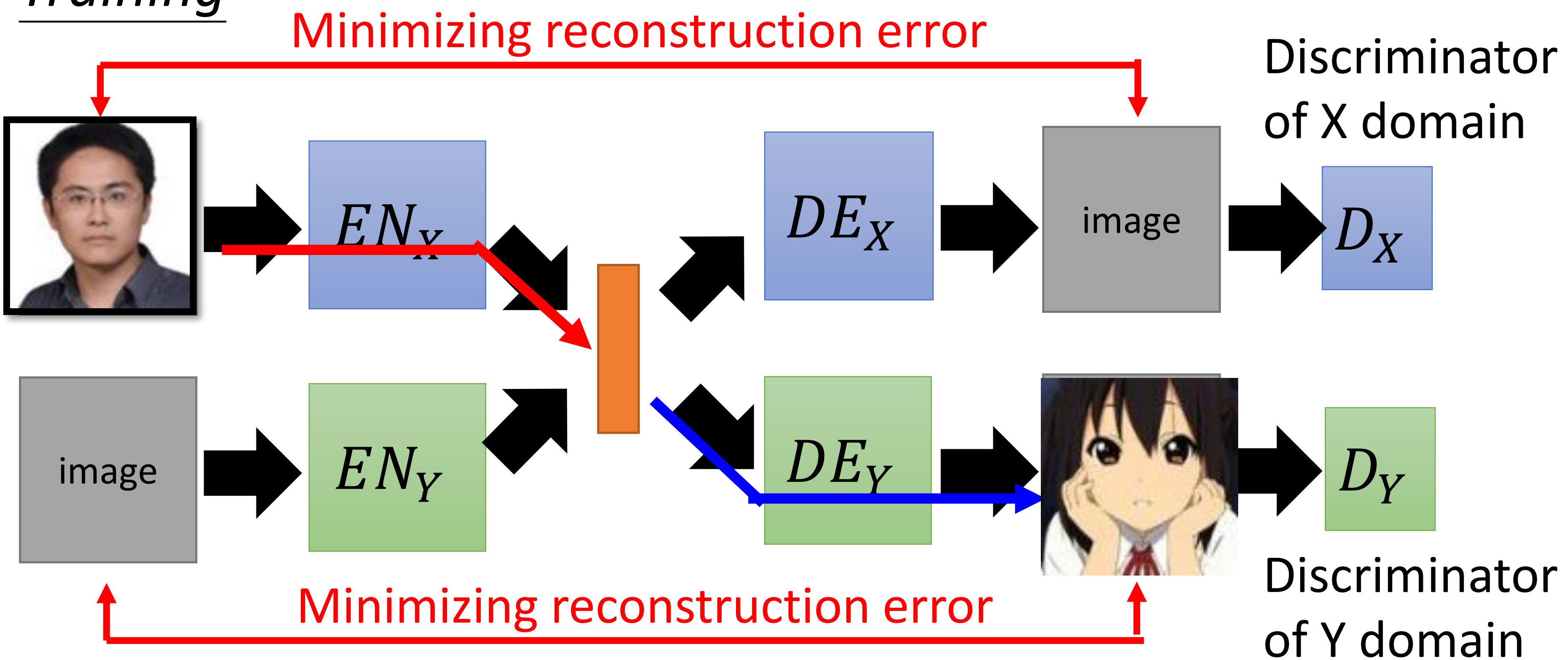
Projection to Common Space

Training



Projection to Common Space

Training

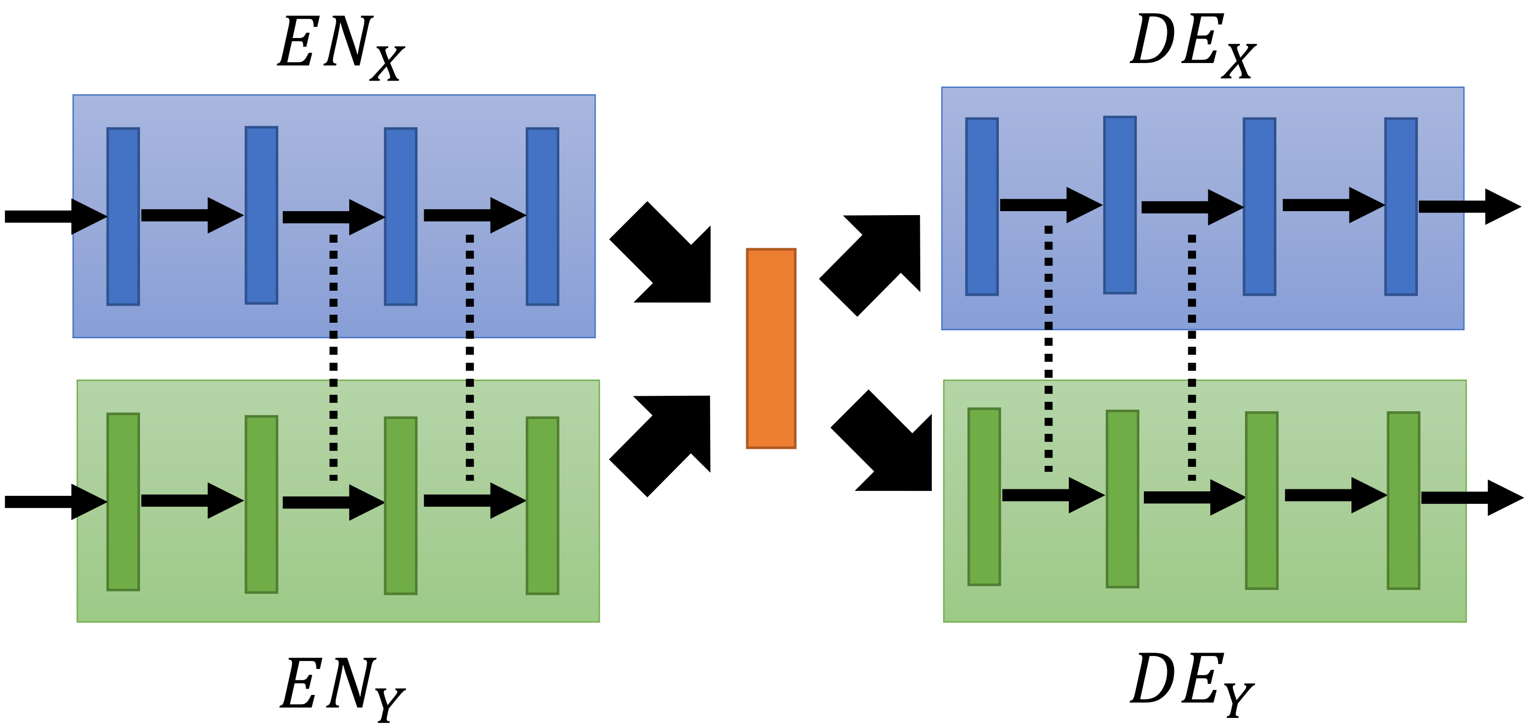


Because we train two auto-encoders separately …

The images with the same attribute may not project to the same position in the latent space.

Projection to Common Space

Training



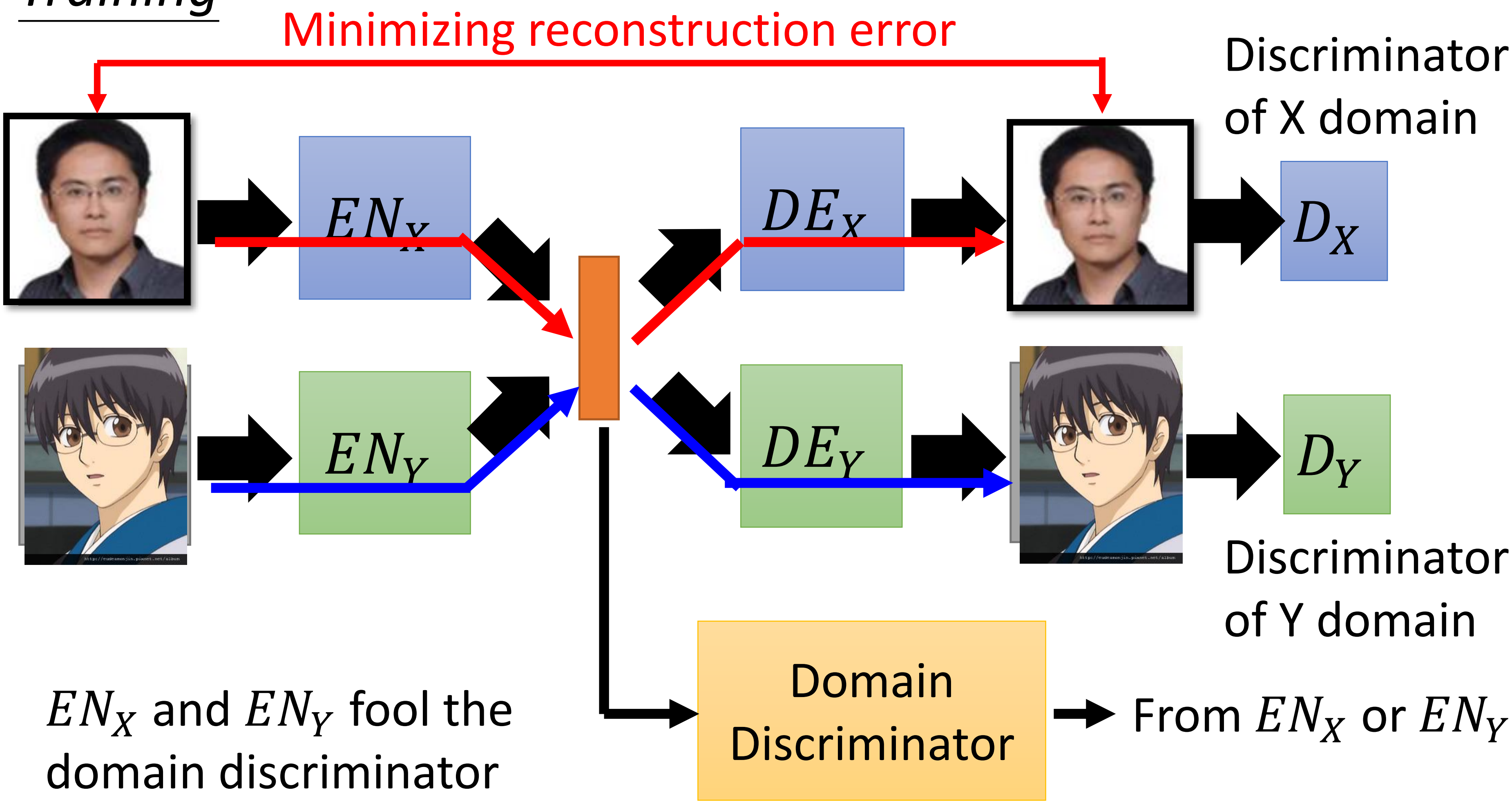
Sharing the parameters of encoders and decoders

Couple GAN[Ming-Yu Liu, et al., NIPS, 2016]

UNIT[Ming-Yu Liu, et al., NIPS, 2017]

Projection to Common Space

Training

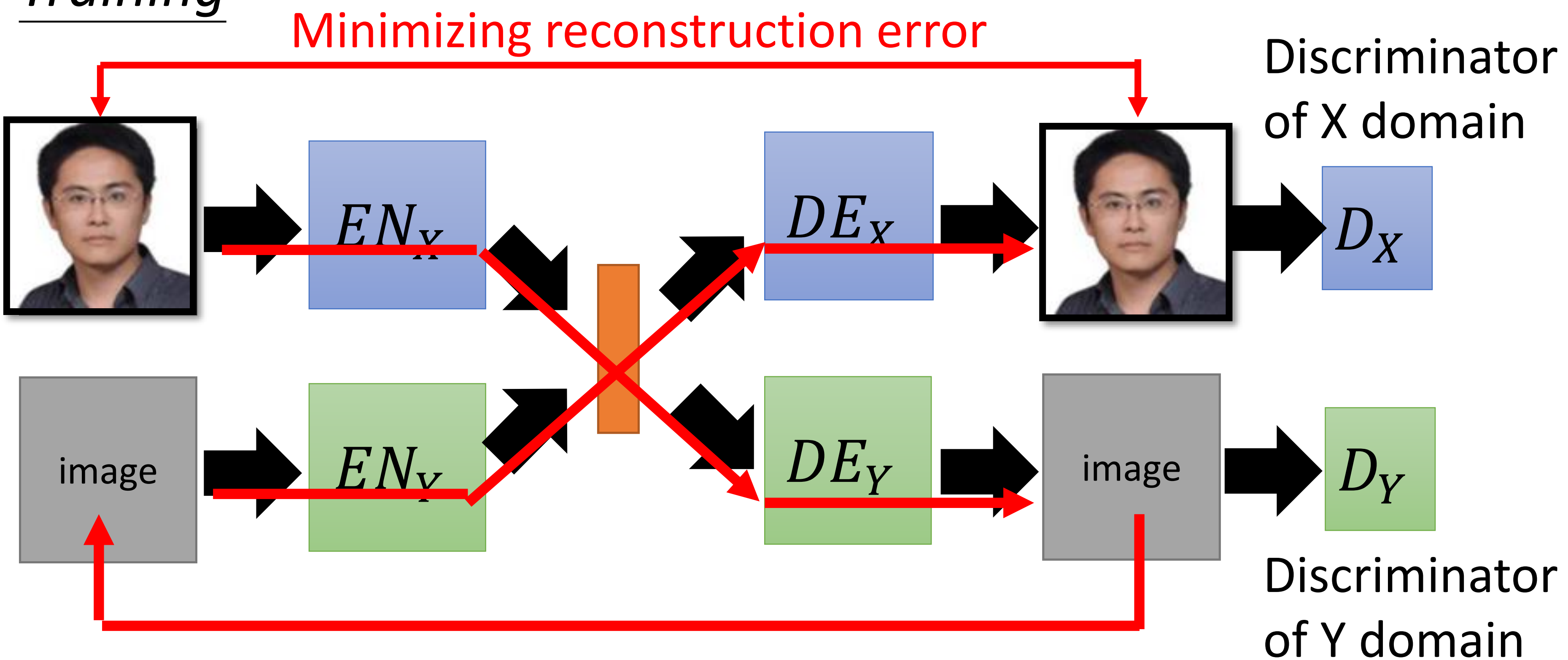


EN_X and EN_Y fool the domain discriminator

The domain discriminator forces the output of EN_X and EN_Y have the same distribution. [Guillaume Lample, et al., NIPS, 2017]

Projection to Common Space

Training

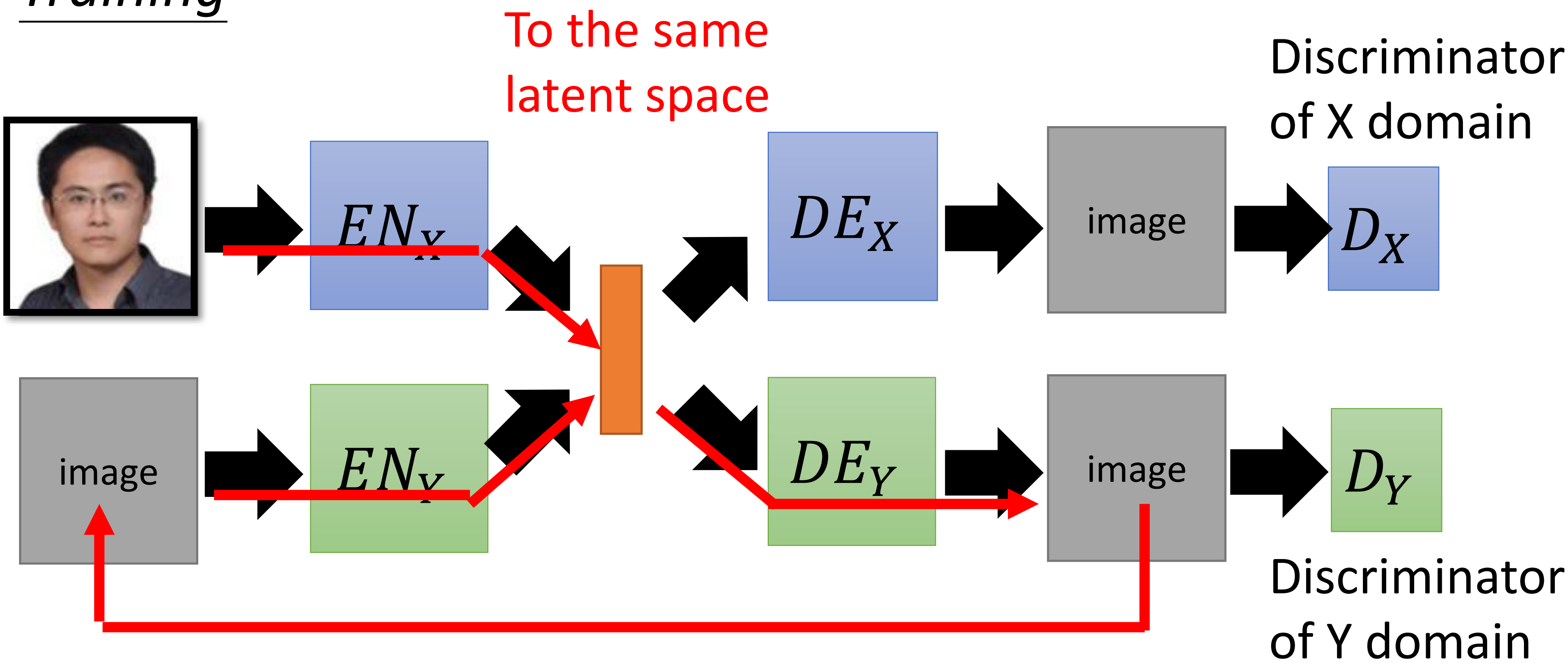


Cycle Consistency:

Used in ComboGAN [Asha Anoosheh, et al., arXiv, 017]

Projection to Common Space

Training



Semantic Consistency:

Used in DTN [Yaniv Taigman, et al., ICLR, 2017] and XGAN [Amélie Royer, et al., arXiv, 2017]

Outline of Part 1

Generation

Conditional Generation

Unsupervised Conditional Generation

Relation to Reinforcement Learning

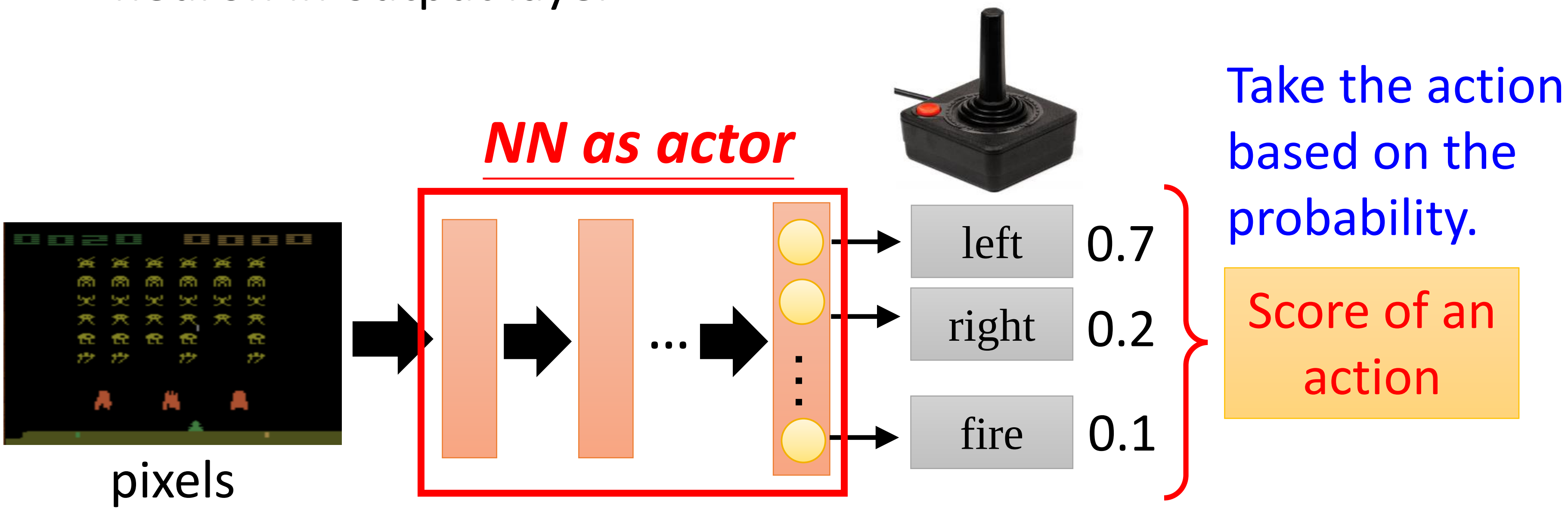
Basic Components

You cannot control

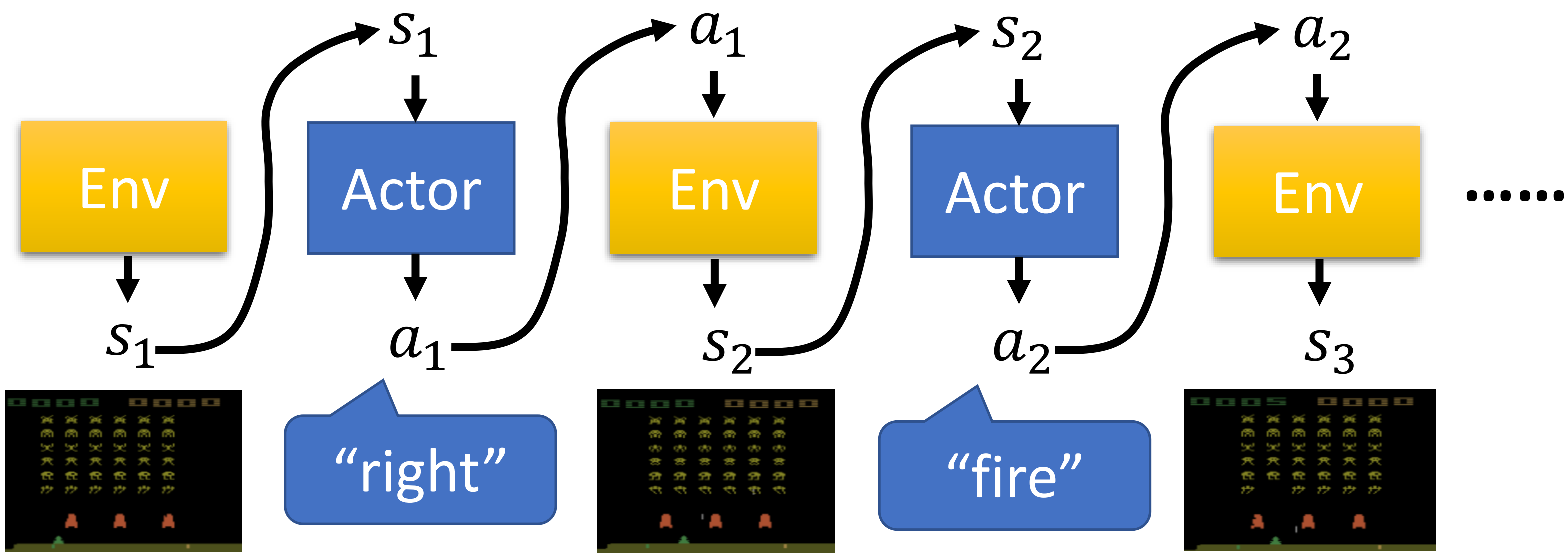
	Actor	Env	Reward Function
Video Game			Get 20 scores when killing a monster
Go	AlphaGo Google DeepMind		The rule of GO

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



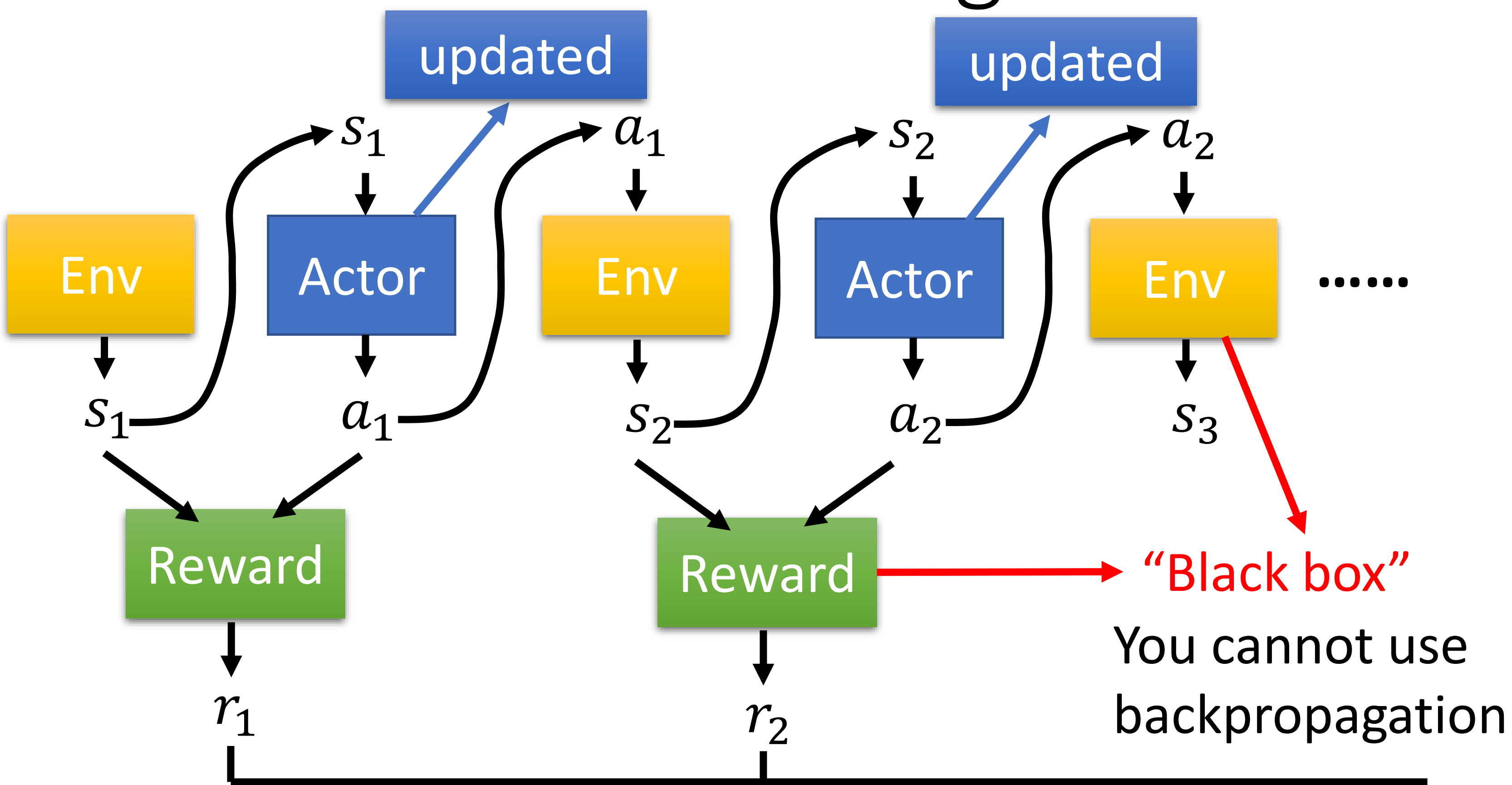
Actor, Environment, Reward



Trajectory

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

Reinforcement Learning v.s. GAN



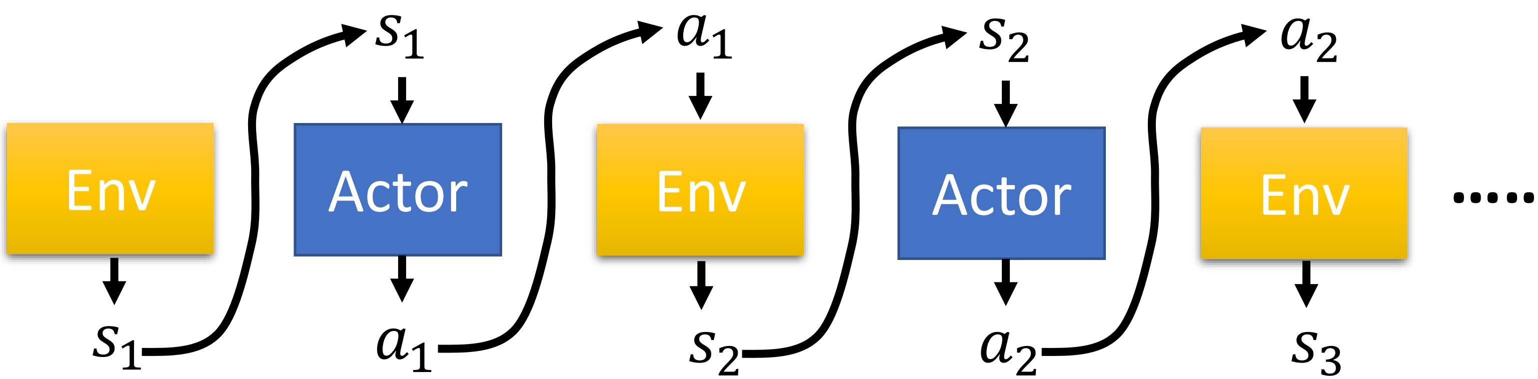
Actor → Generator

Reward Function → Discriminator

Fixed

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

Imitation Learning

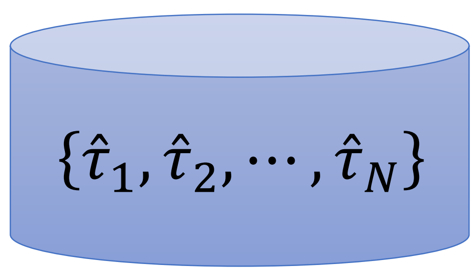


reward function is not available

We have demonstration of the expert.

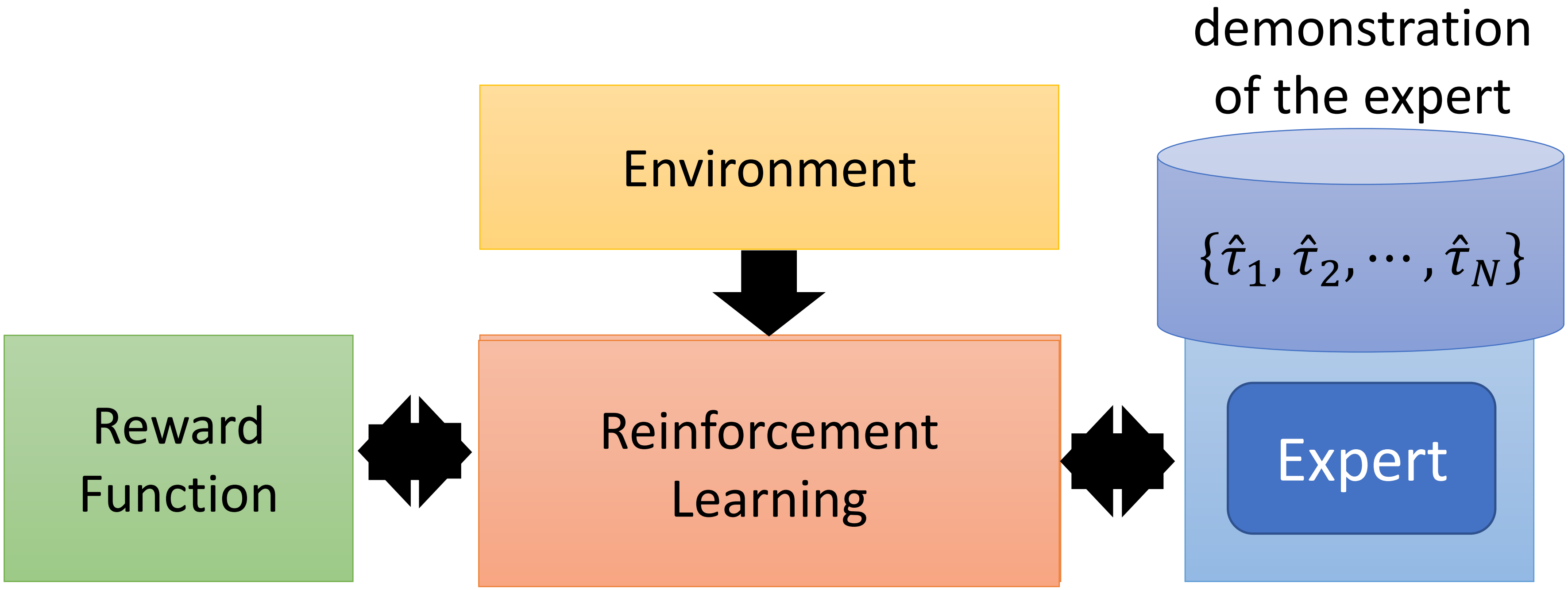
Self driving: record human drivers

Robot: grab the arm of robot



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

Inverse Reinforcement Learning

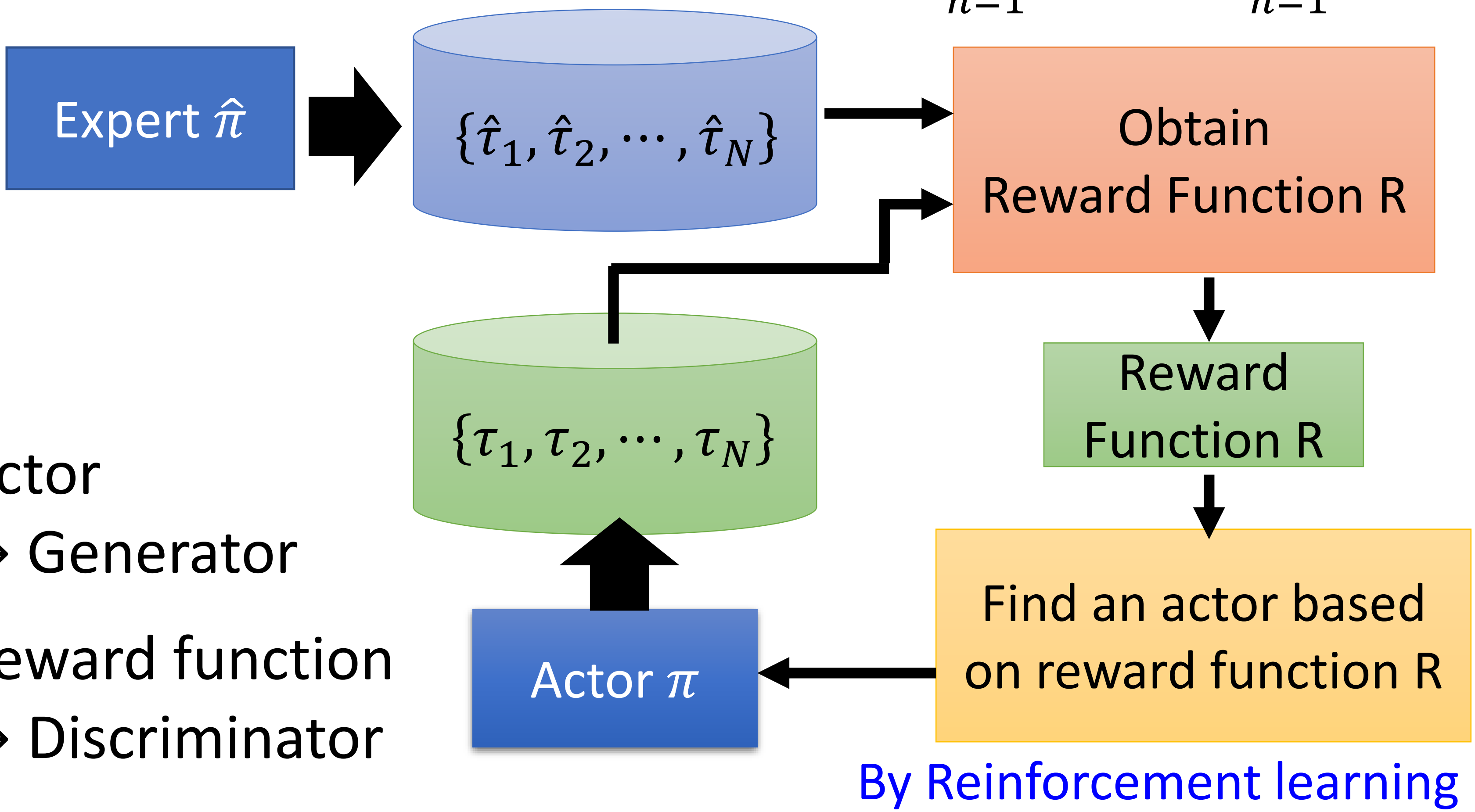


- Using the reward function to find the ***optimal actor***.
- Modeling reward can be easier. Simple reward function can lead to complex policy.

Framework of IRL

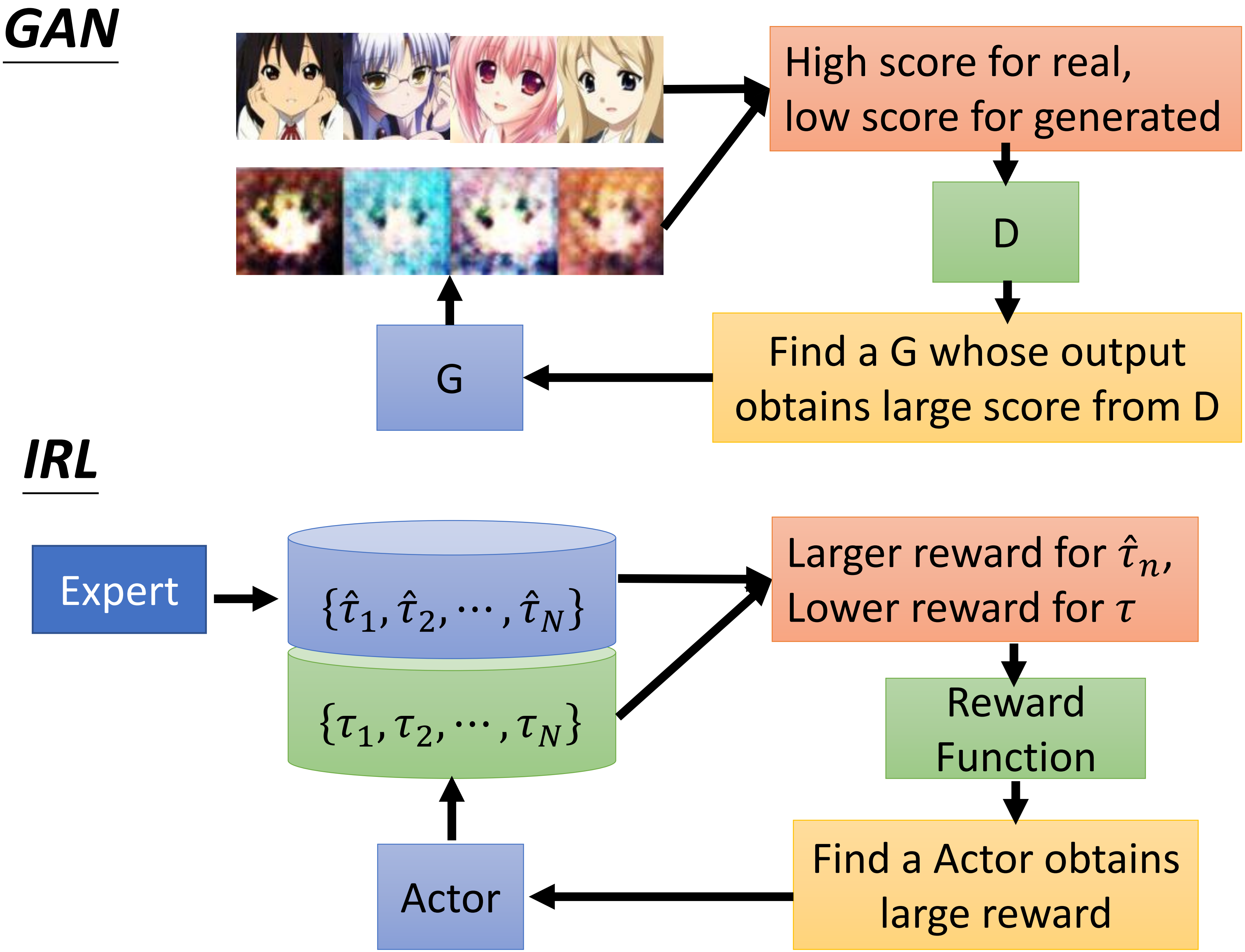
The expert is always the best.

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



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

Concluding Remarks

Generation

Conditional Generation

Unsupervised Conditional Generation

Relation to Reinforcement Learning

Reference

- **Generation**

  - Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative Adversarial Networks, NIPS, 2014
  - Sebastian Nowozin, Botond Cseke, Ryota Tomioka, “f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization”, NIPS, 2016
  - Martin Arjovsky, Soumith Chintala, Léon Bottou, Wasserstein GAN, arXiv, 2017
  - Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville, Improved Training of Wasserstein GANs, NIPS, 2017
  - Junbo Zhao, Michael Mathieu, Yann LeCun, Energy-based Generative Adversarial Network, arXiv, 2016
  - Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, Olivier Bousquet, “Are GANs Created Equal? A Large-Scale Study”, arXiv, 2017
  - Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen Improved Techniques for Training GANs, NIPS, 2016
  - Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Sepp Hochreiter, GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, NIPS, 2017

Reference

- **Generation**

  - Naveen Kodali, Jacob Abernethy, James Hays, Zsolt Kira, "On Convergence and Stability of GANs", arXiv, 2017
  - Xiang Wei, Boqing Gong, Zixia Liu, Wei Lu, Liqiang Wang, Improving the Improved Training of Wasserstein GANs: A Consistency Term and Its Dual Effect, ICLR, 2018
  - Takeru Miyato, Toshiki Kataoka, Masanori Koyama, Yuichi Yoshida, Spectral Normalization for Generative Adversarial Networks, ICLR, 2018
  - Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen, Progressive Growing of GANs for Improved Quality, Stability, and Variation, ICLR, 2018
  - Andrew Brock, Jeff Donahue, Karen Simonyan, Large Scale GAN Training for High Fidelity Natural Image Synthesis, arXiv, 2018

Reference

- **Conditional Generation**

  - Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee, Generative Adversarial Text to Image Synthesis, ICML, 2016
  - Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros, Image-to-Image Translation with Conditional Adversarial Networks, CVPR, 2017
  - Michael Mathieu, Camille Couprie, Yann LeCun, Deep multi-scale video prediction beyond mean square error, arXiv, 2015
  - Mehdi Mirza, Simon Osindero, Conditional Generative Adversarial Nets, arXiv, 2014
  - Takeru Miyato, Masanori Koyama, cGANs with Projection Discriminator, ICLR, 2018
  - Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dimitris Metaxas, StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks, arXiv, 2017
  - Augustus Odena, Christopher Olah, Jonathon Shlens, Conditional Image Synthesis With Auxiliary Classifier GANs, ICML, 2017

Reference

- **Conditional Generation**
  - 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
  - Che-Ping Tsai, Hung-Yi Lee, Adversarial Learning of Label Dependency: A Novel Framework for Multi-class Classification, submitted to ICASSP 2019

Reference

- **Unsupervised Conditional Generation**
  - Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros, Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, ICCV, 2017
  - Zili Yi, Hao Zhang, Ping Tan, Minglun Gong, DualGAN: Unsupervised Dual Learning for Image-to-Image Translation, ICCV, 2017
  - Tomer Galanti, Lior Wolf, Sagie Benaim, The Role of Minimal Complexity Functions in Unsupervised Learning of Semantic Mappings, ICLR, 2018
  - Yaniv Taigman, Adam Polyak, Lior Wolf, Unsupervised Cross-Domain Image Generation, ICLR, 2017
  - Asha Anoosheh, Eirikur Agustsson, Radu Timofte, Luc Van Gool, ComboGAN: Unrestrained Scalability for Image Domain Translation, arXiv, 2017
  - Amélie Royer, Konstantinos Bousmalis, Stephan Gouws, Fred Bertsch, Inbar Mosseri, Forrester Cole, Kevin Murphy, XGAN: Unsupervised Image-to-Image Translation for Many-to-Many Mappings, arXiv, 2017

Reference

- **Unsupervised Conditional Generation**
  - Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, Marc'Aurelio Ranzato, Fader Networks: Manipulating Images by Sliding Attributes, NIPS, 2017
  - Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, Jiwon Kim, Learning to Discover Cross-Domain Relations with Generative Adversarial Networks, ICML, 2017
  - Ming-Yu Liu, Oncel Tuzel, “Coupled Generative Adversarial Networks”, NIPS, 2016
  - Ming-Yu Liu, Thomas Breuel, Jan Kautz, Unsupervised Image-to-Image Translation Networks, NIPS, 2017
  - Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo, StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation, arXiv, 2017

Outline

National Taiwan University

Part I: General Introduction of Generative Adversarial Network (GAN)

Part II: Applications to Natural Language Processing

Part III: Applications to Speech Processing

Unsupervised Conditional Generation

Image Style Transfer

photos

Not Paired

Vincent van Gogh's paintings

Text Style Transfer

It is good.
It's a good day.
I love you.

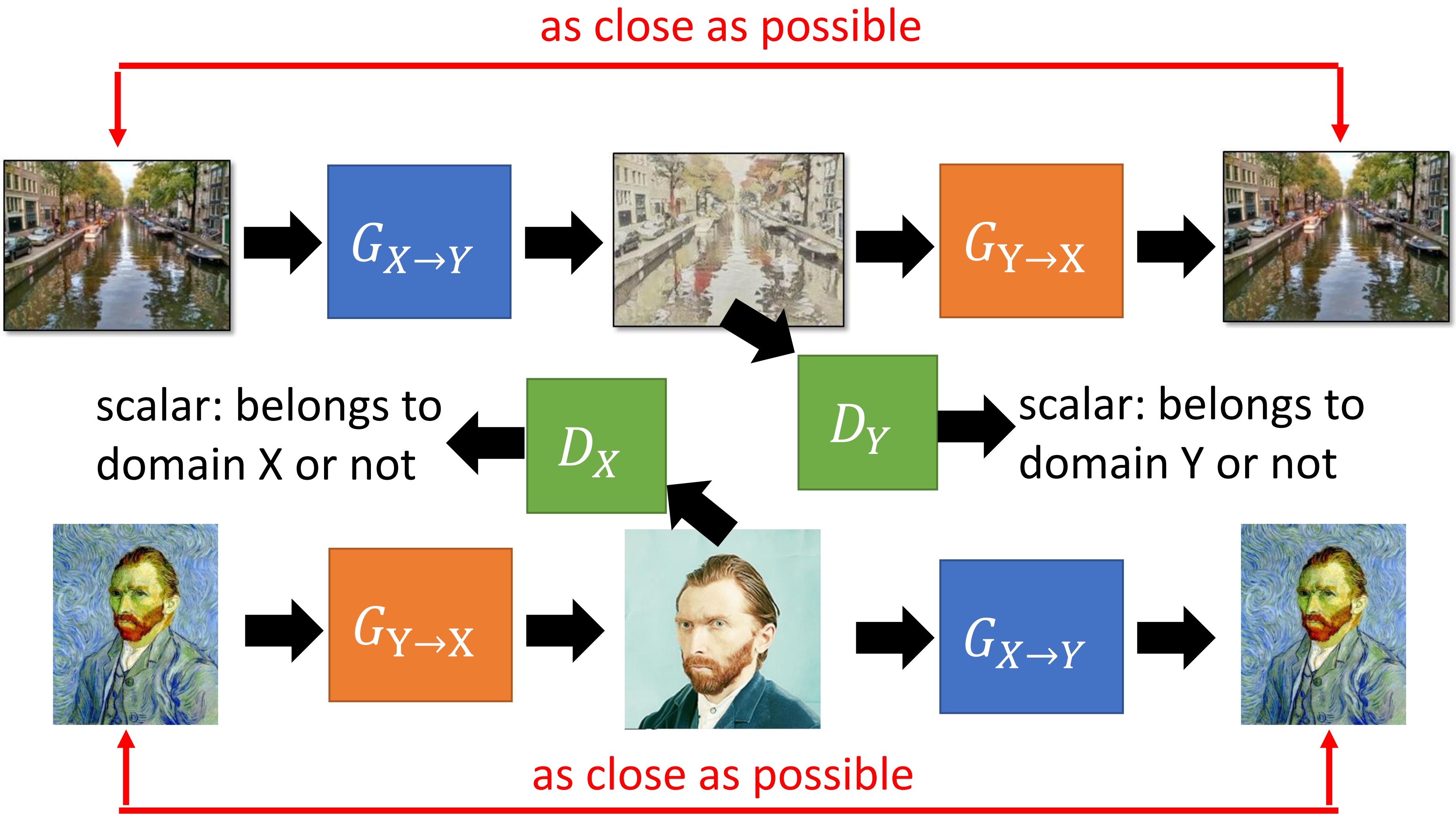
positive

Not Paired

It is bad.
It's a bad day.
I don't love you.

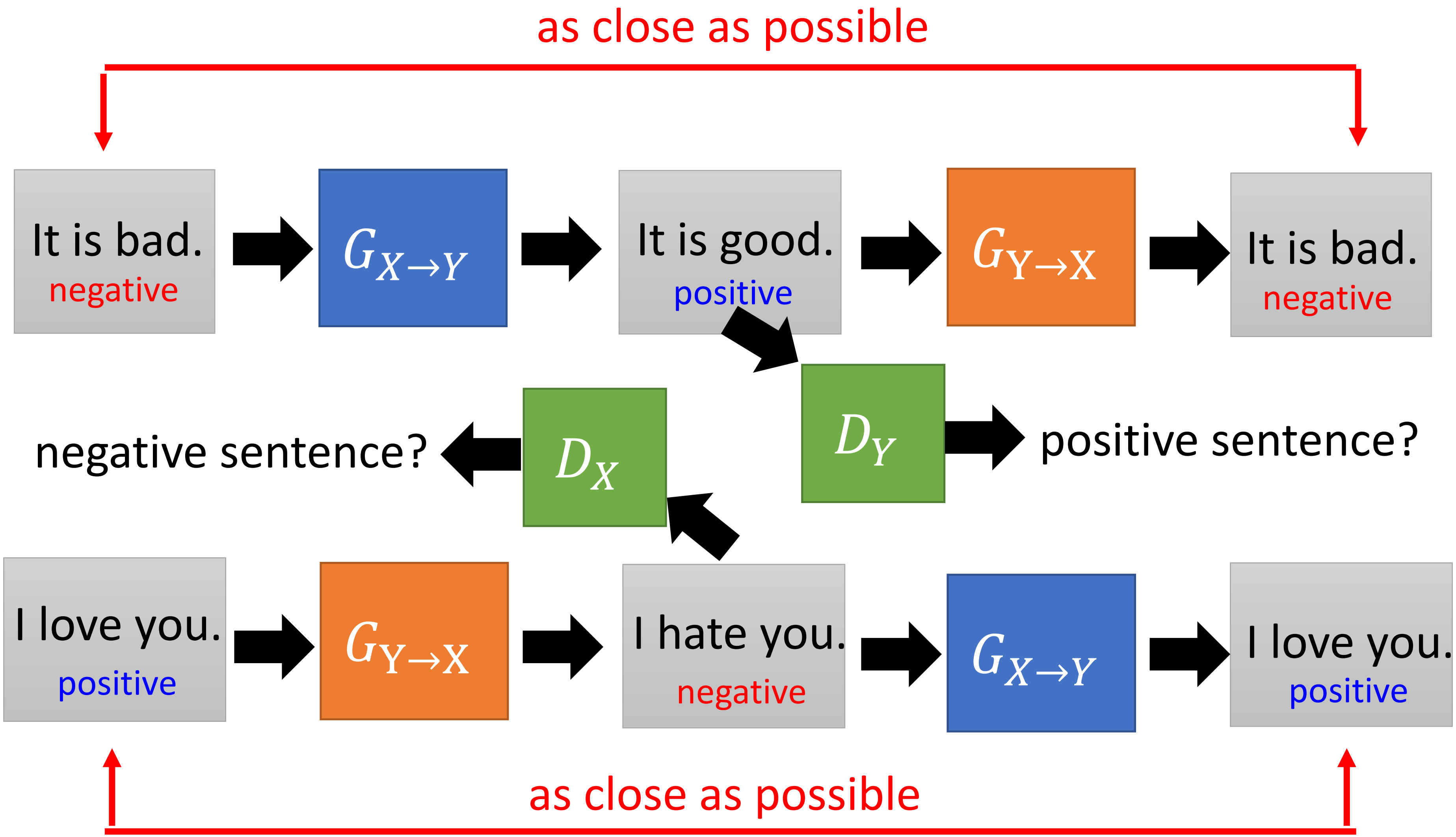
negative

Cycle GAN

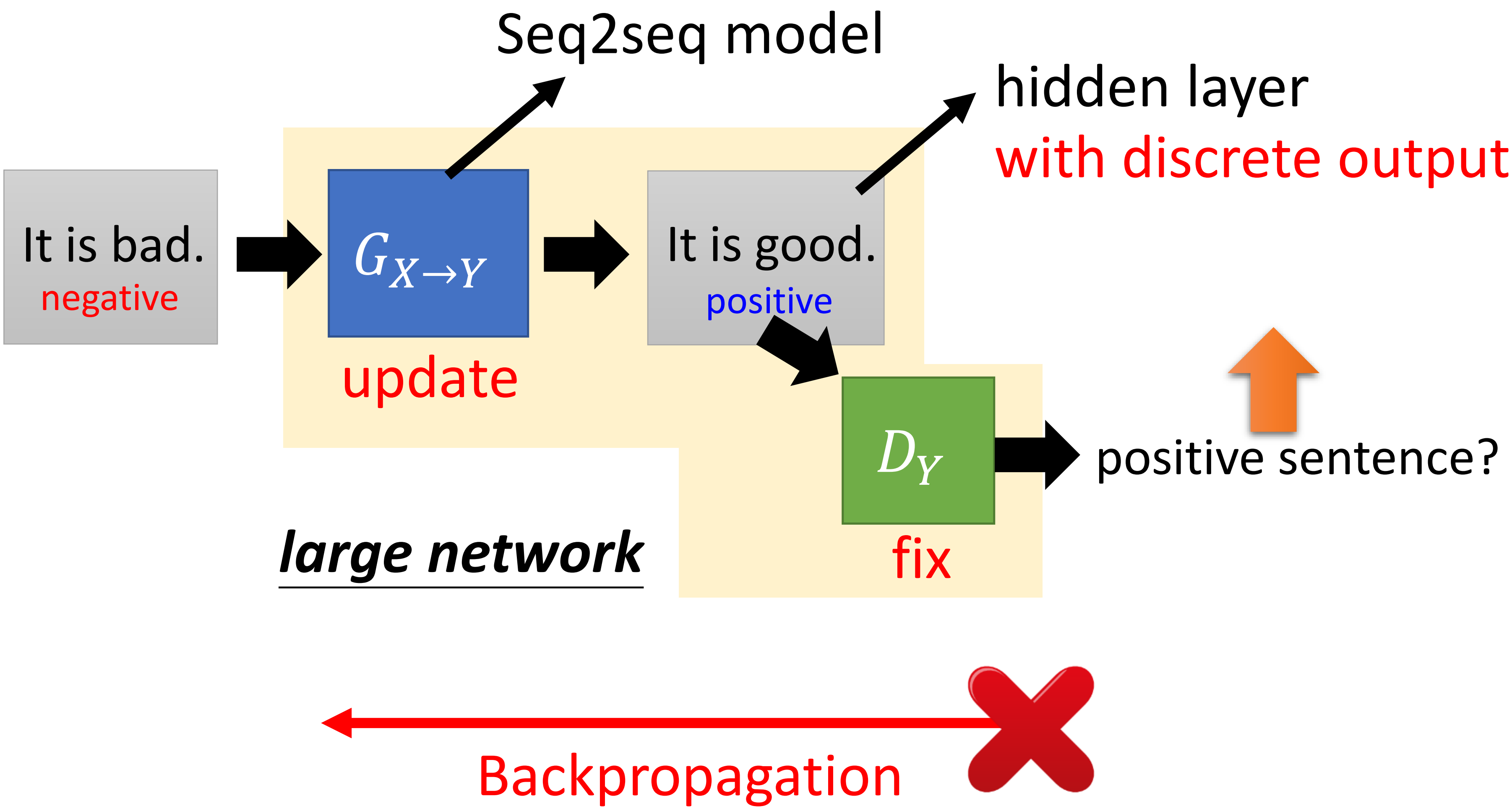


as close as possible
$G_{X\to Y}$
$G_{Y\to X}$
D_Y
scalar: belongs to domain Y or not
D_X
scalar: belongs to domain X or not
$G_{Y\to X}$
$G_{X\to Y}$
as close as possible

Cycle GAN


as close as possible
It is bad.
negative
$G_{X \to Y}$
It is good.
positive
$G_{Y \to X}$
It is bad.
negative
D_Y
positive sentence?
D_X
negative sentence?
I love you.
positive
$G_{Y \to X}$
I hate you.
negative
$G_{X \to Y}$
I love you.
positive
as close as possible

Discrete Issue



Three Categories of Solutions

Gumbel-softmax

- [Matt J. Kusner, et al, arXiv, 2016]

Continuous Input for Discriminator

- [Sai Rajeswar, et al., arXiv, 2017][Ofir Press, et al., ICML workshop, 2017][Zhen Xu, et al., EMNLP, 2017][Alex Lamb, et al., NIPS, 2016][Yizhe Zhang, et al., ICML, 2017]

“Reinforcement Learning”

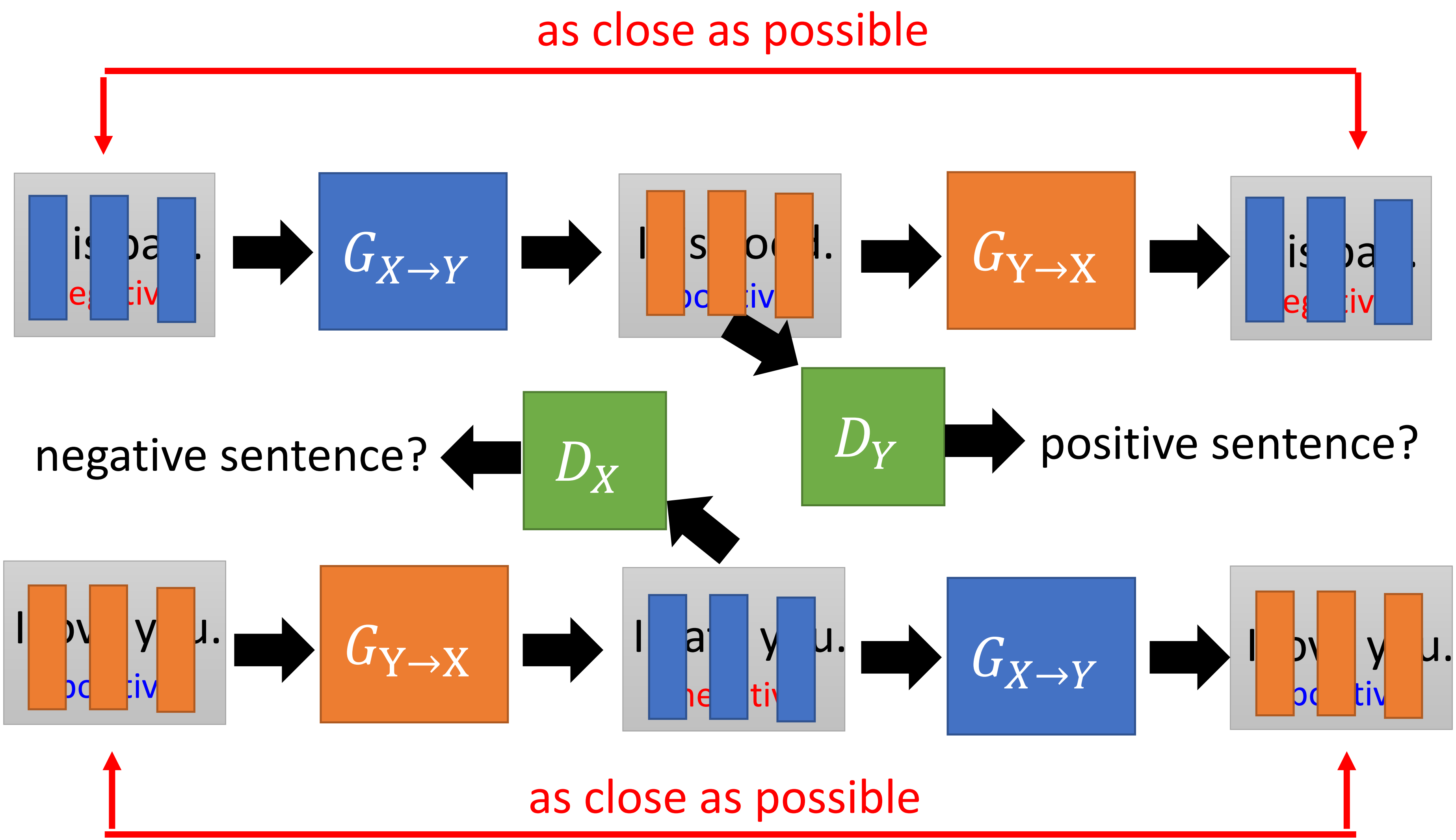
- [Yu, et al., AAAI, 2017][Li, et al., EMNLP, 2017][Tong Che, et al, arXiv, 2017][Jiaxian Guo, et al., AAAI, 2018][Kevin Lin, et al, NIPS, 2017][William Fedus, et al., ICLR, 2018]

Cycle GAN

Discrete?

Word embedding

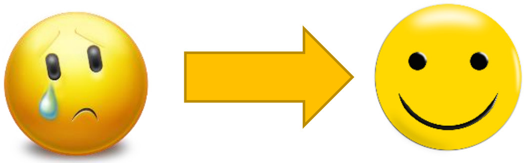
[Lee, et al., ICASSP, 2018]



Thinks Yau-Shian Wang for providing the results.

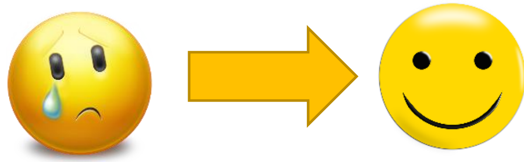
Cycle GAN

- 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 dummy 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
  my little doggy is sick → my little doggy is my little doggy



感謝 張瓊之 同學提供實驗結果

Cycle GAN



Negative sentence to positive sentence:

胃疼，沒睡醒，各種不舒服 -> 生日快樂，睡醒，超級舒服

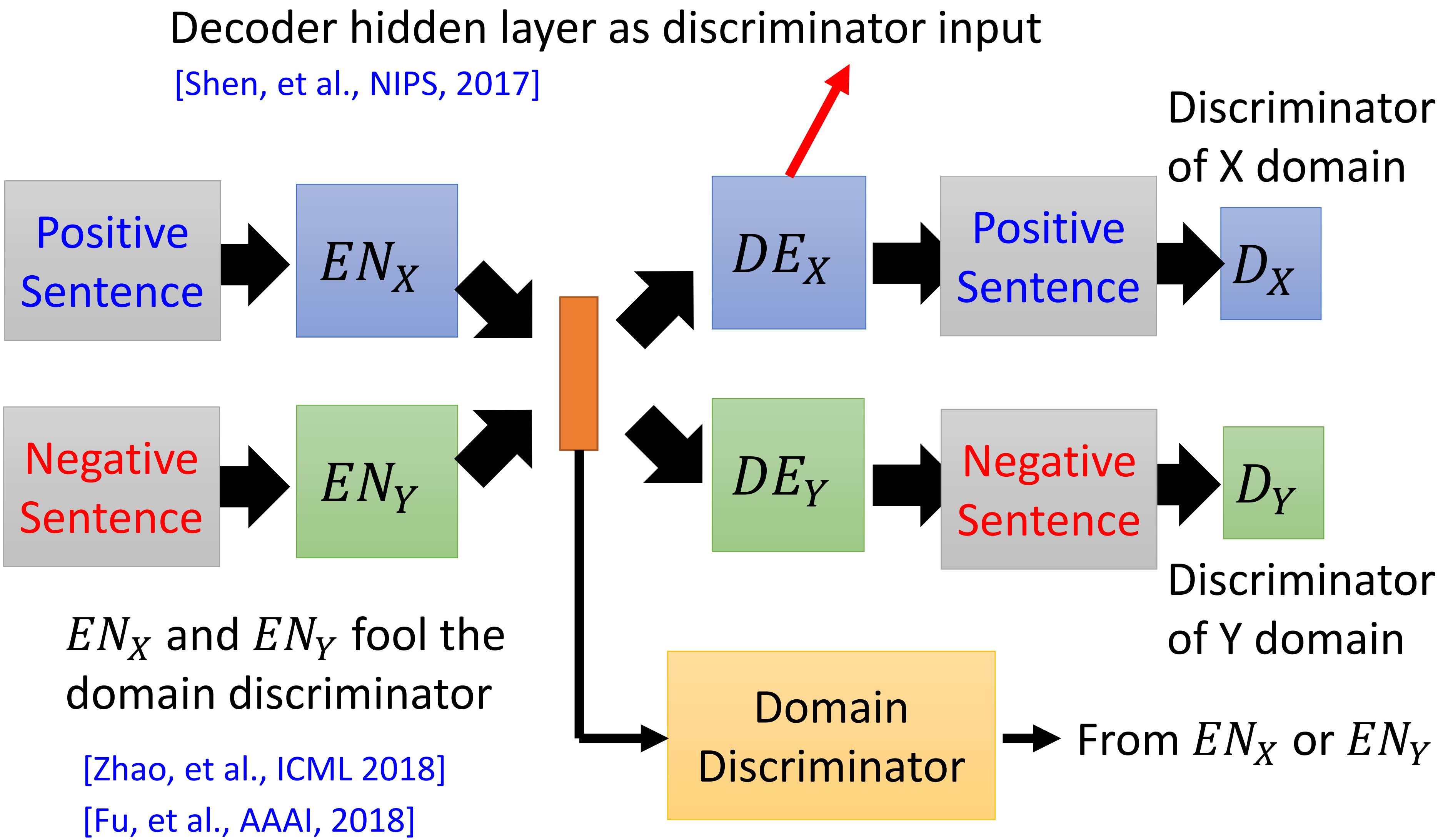
我都想去上班了, 真夠賤的! -> 我都想去睡了, 真帥的 !

暈死了, 吃燒烤、竟然遇到個變態狂 -> 哈哈好 ~ , 吃燒烤 ~ 竟然遇到帥狂

我肚子痛的厲害 -> 我生日快樂厲害

感冒了, 難受的說不出話來了 ! -> 感冒了, 開心的說不出話來 !

Projection to Common Space



Unsupervised Conditional Generation

Image Style Transfer

Not Paired

photos

Vincent van Gogh's paintings

Text Style Transfer

document

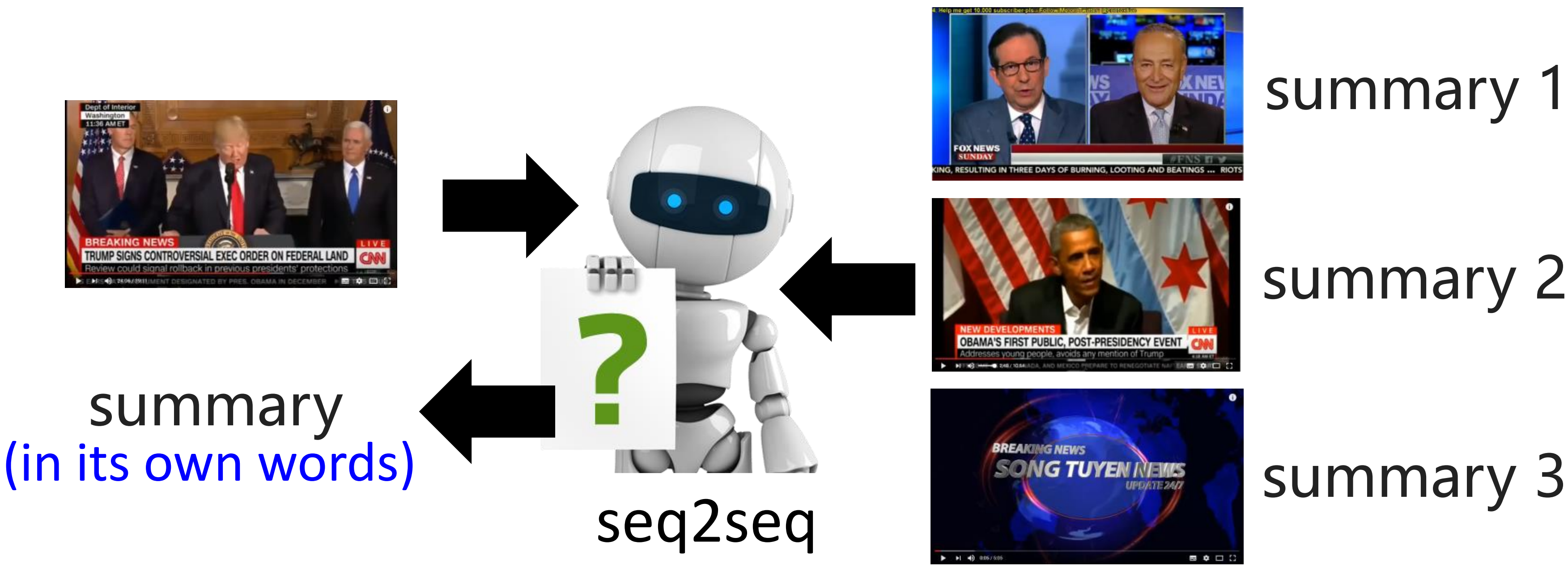
Not Paired

summary

This is **unsupervised abstractive summarization**.

Abstractive Summarization

- Now machine can do **abstractive summary** by seq2seq (write summaries in its own words)

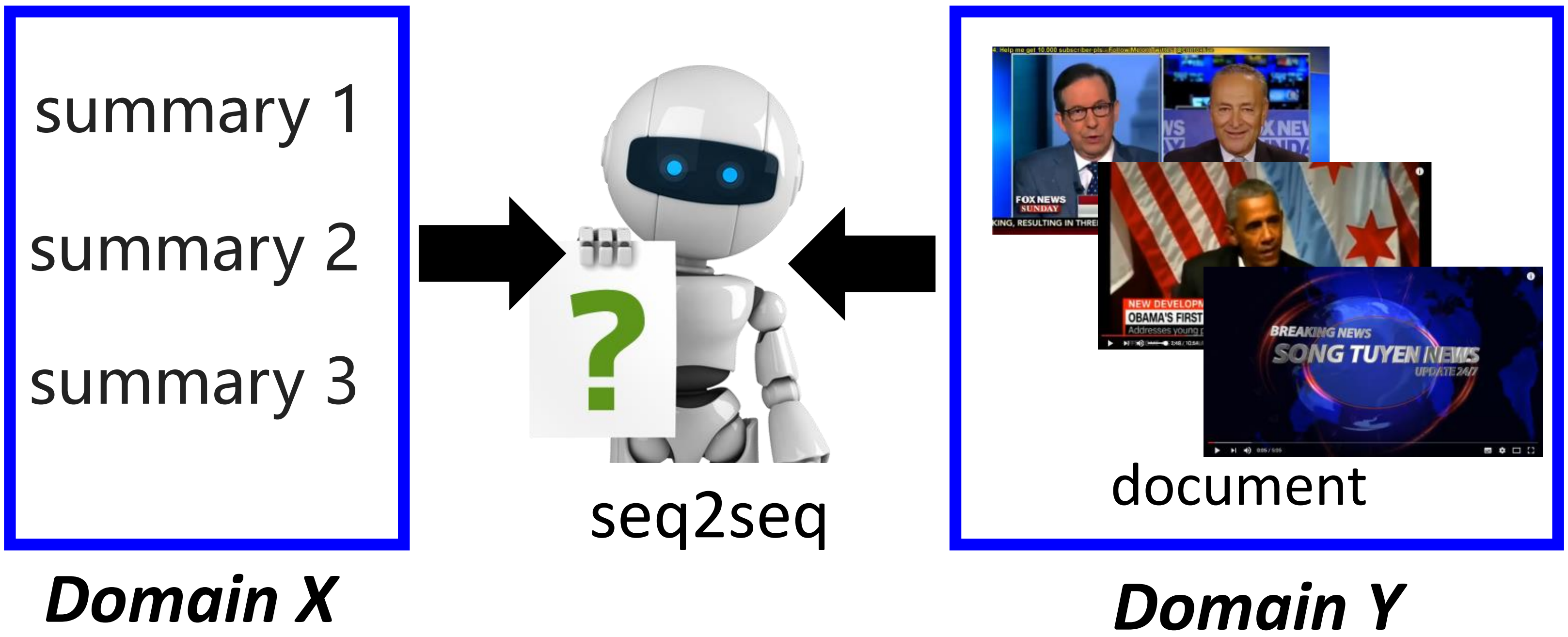


Supervised: We need lots of labelled training data.

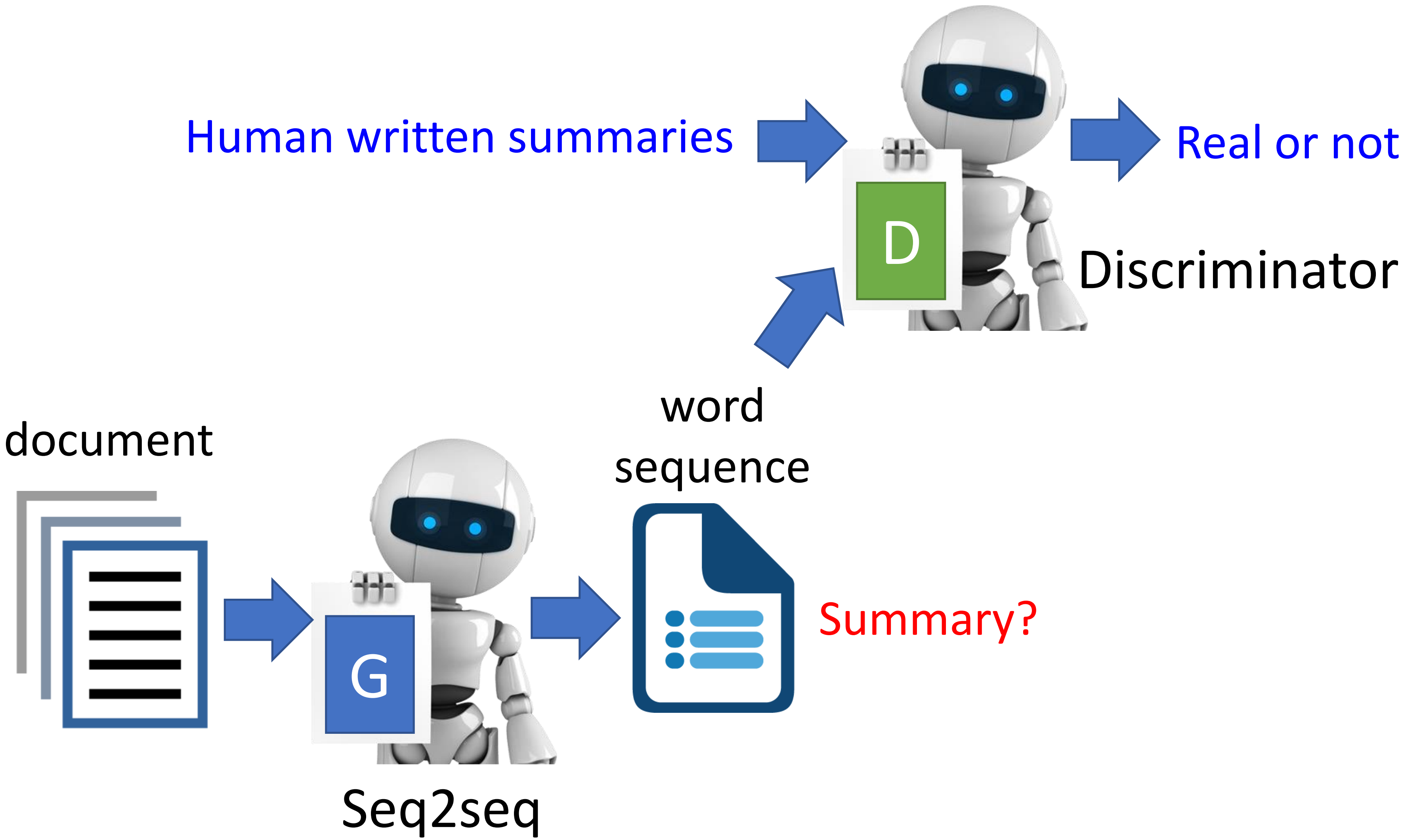
Training Data

Unsupervised Abstractive Summarization

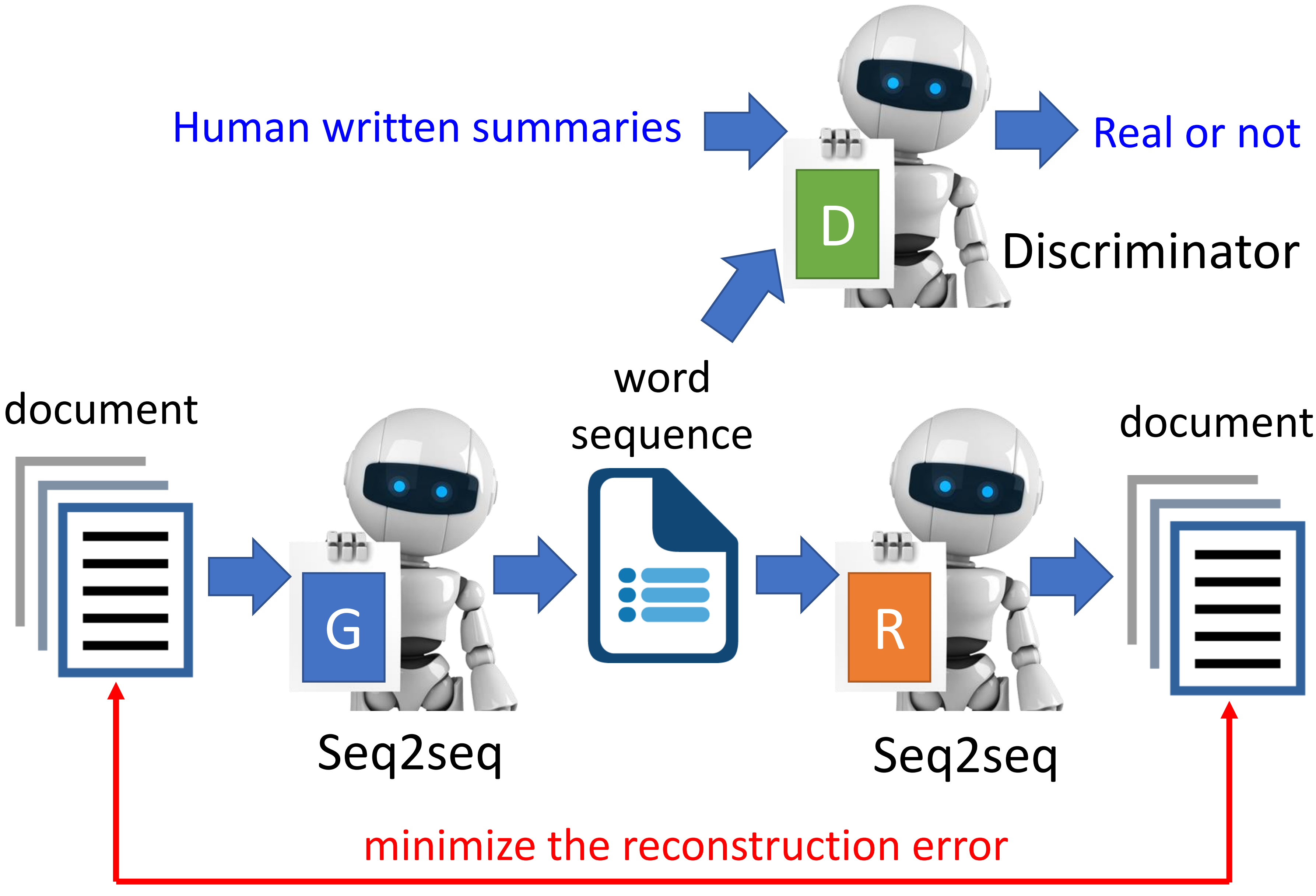
- Now machine can do **abstractive summary** by seq2seq (write summaries in its own words)



Unsupervised Abstractive Summarization



Unsupervised Abstractive Summarization

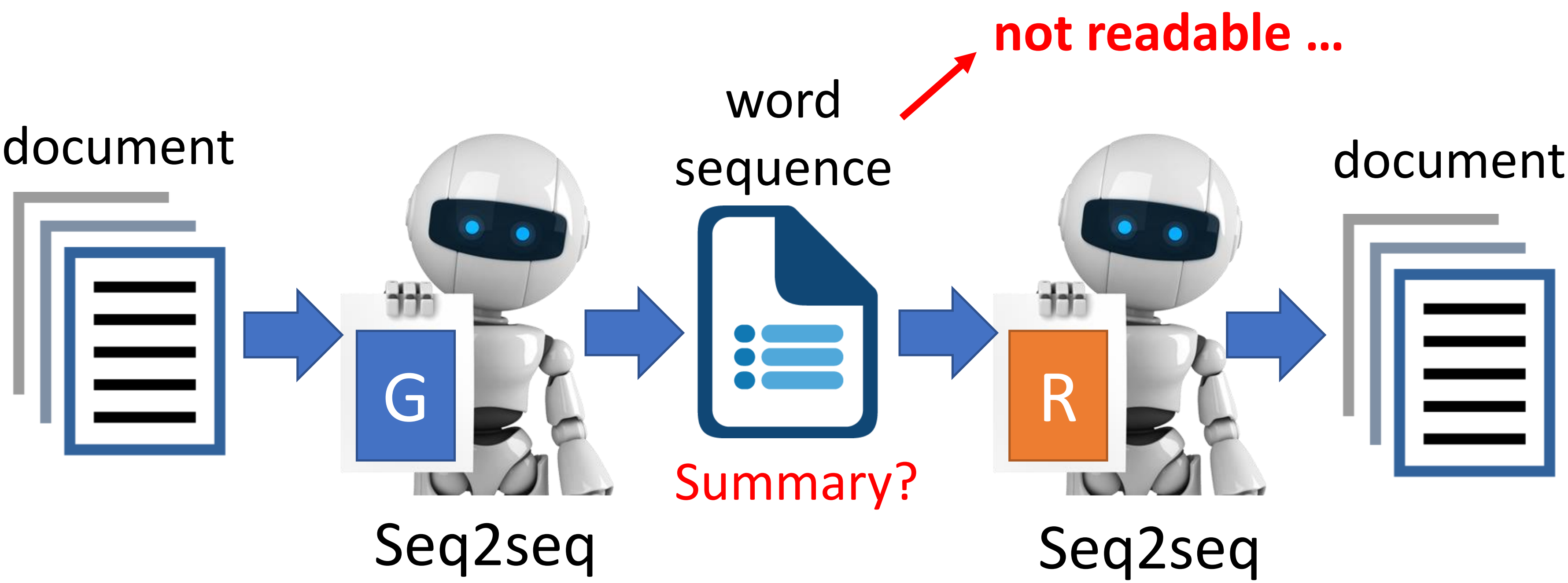


Unsupervised Abstractive Summarization

Only need a lot of documents to train the model

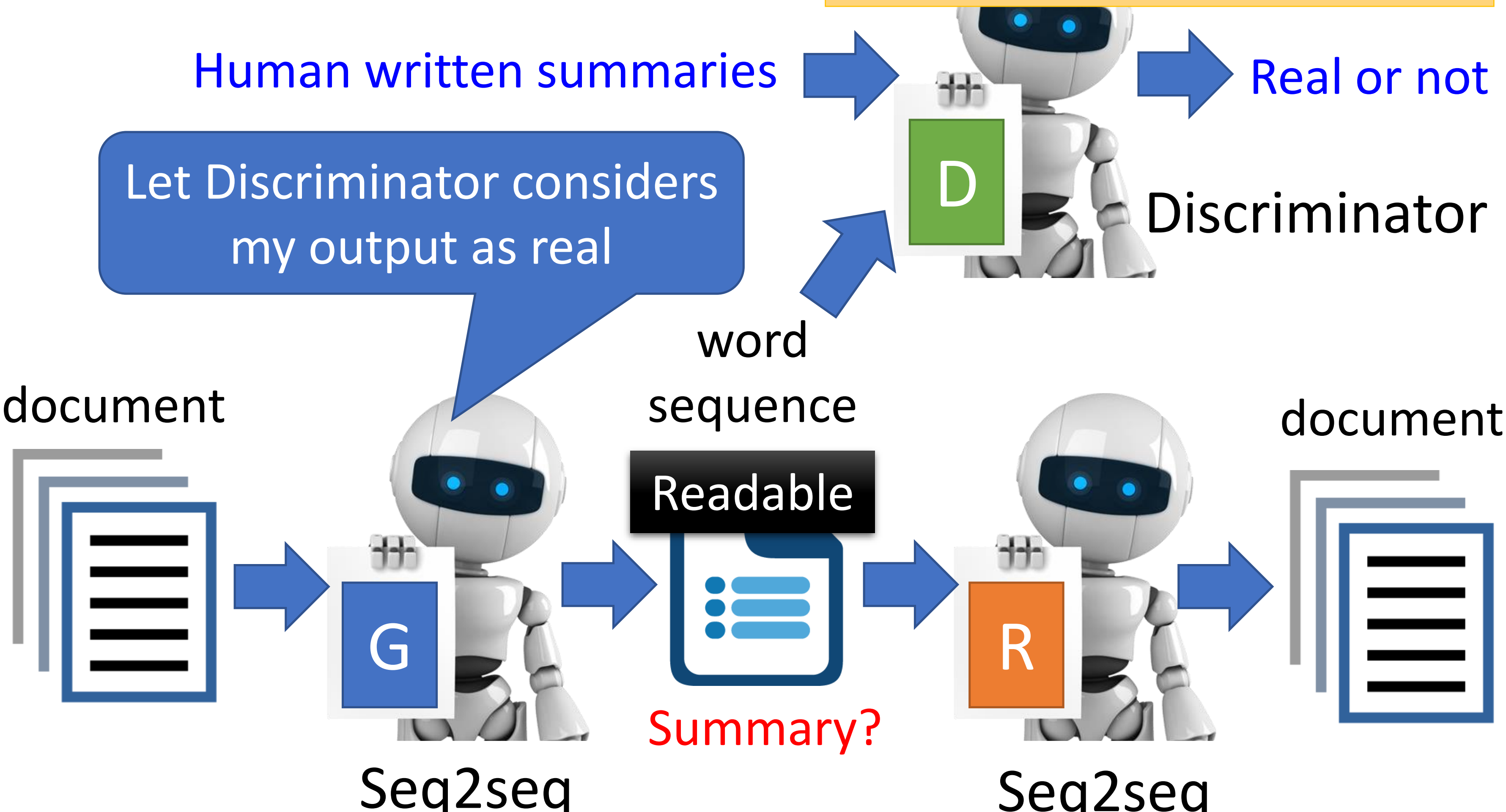
This is a ***seq2seq2seq auto-encoder***.

Using a sequence of words as latent representation.



Unsupervised Abstractive Summarization

REINFORCE algorithm to deal with the discrete issue



Experimental results

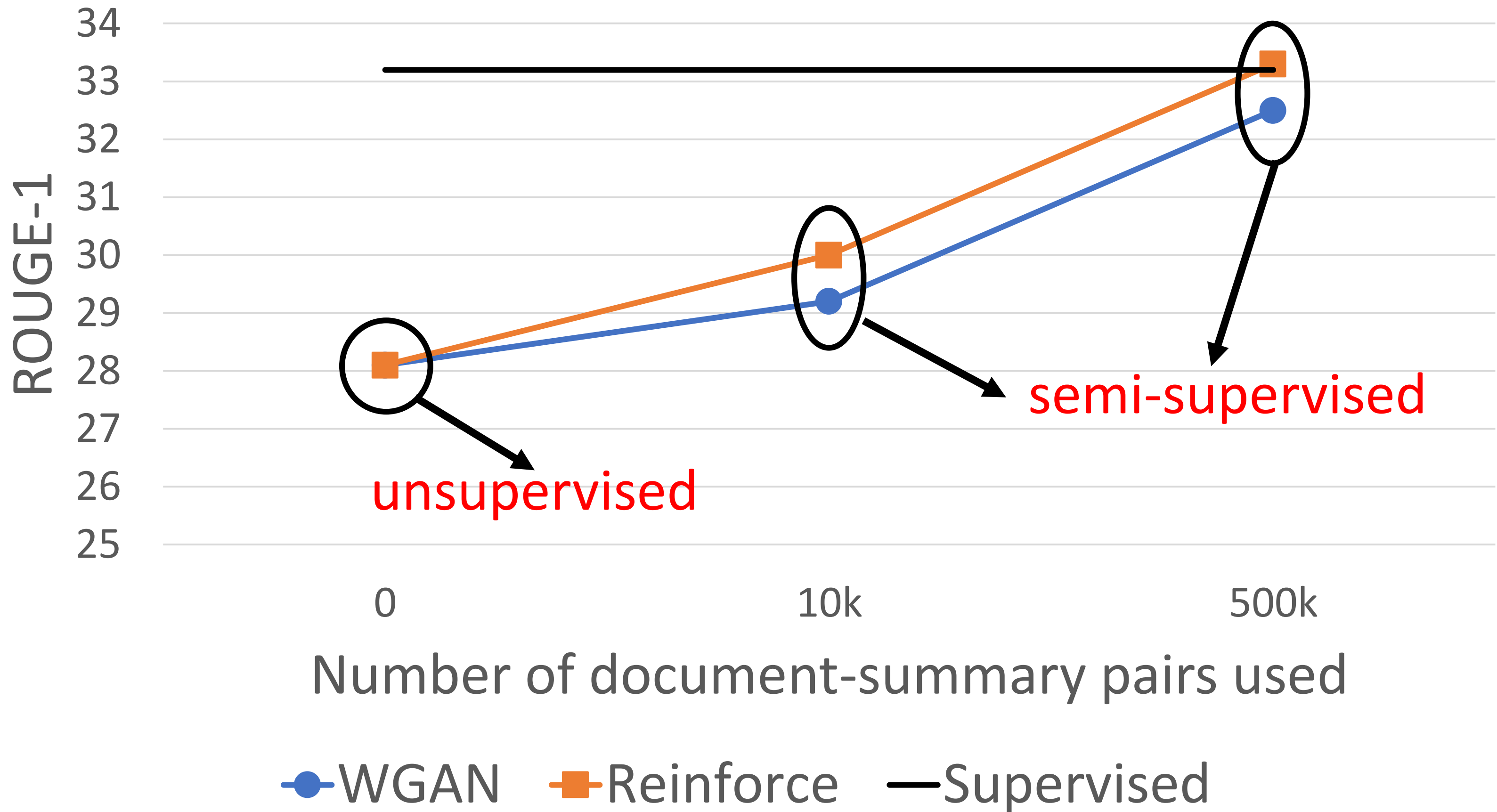
English Gigaword (Document title as summary)

	ROUGE-1	ROUGE-2	ROUGE-L
Supervised	33.2	14.2	30.5
Trivial	21.9	7.7	20.5
Unsupervised (matched data)	28.1	10.0	25.4
Unsupervised (no matched data)	27.2	9.1	24.1

- Matched data: using the title of English Gigaword to train Discriminator
- No matched data: using the title of CNN/Diary Mail to train Discriminator

Semi-supervised Learning

Using matched data



Approaches to deal with the discrete issue.

3.8M pairs are used.

感謝 王耀賢 同學提供實驗結果

Unsupervised Abstractive Summarization

- ***Document***:澳大利亞今天與13個國家簽署了反興奮劑雙邊協議,旨在加強體育競賽之外的藥品檢查並共享研究成果 ……
- ***Summary***:
  - Human:澳大利亞與13國簽署反興奮劑協議
  - Unsupervised:澳大利亞加強體育競賽之外的藥品檢查
- ***Document***:中華民國奧林匹克委員會今天接到一九九二年冬季奧運會邀請函,由於主席張豐緒目前正在中南美洲進行友好訪問,因此尚未決定是否派隊赴賽 ……
- ***Summary***:
  - Human:一九九二年冬季奧運會函邀我參加
  - Unsupervised:奧委會接獲冬季奧運會邀請函

感謝 王耀賢 同學提供實驗結果

Unsupervised Abstractive Summarization

- ***Document***:據此間媒體27日報道,印度尼西亞蘇門答臘島的兩個省近日來連降暴雨,洪水泛濫導致塌方,到26日為止至少已有60人喪生,100多人失蹤 ……
- ***Summary***:
  - Human:印尼水災造成60人死亡
  - Unsupervised:印尼門洪水泛濫導致塌雨
- ***Document***:安徽省合肥市最近為領導幹部下基層做了新規定:一律輕車簡從,不準搞迎來送往、不準搞層層陪同 ……
- ***Summary***:
  - Human:合肥規定領導幹部下基層活動從簡
  - Unsupervised:合肥領導幹部下基層做搞迎來送往規定:一律簡

Outline

National Taiwan University

Part I: General Introduction of Generative Adversarial Network (GAN)

Part II: Applications to Natural Language Processing

Part III: Applications to Speech Processing

Unsupervised Conditional Generation

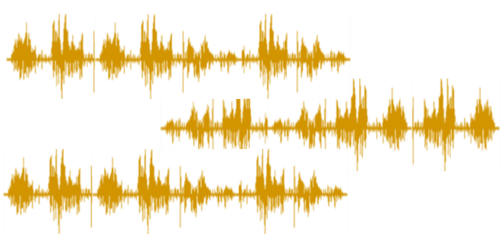
Image Style Transfer

photos

Not Paired

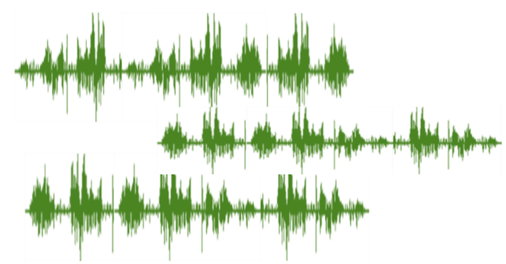
Vincent van Gogh's paintings

Speech Style Transfer



Speaker A

Not Paired



Speaker B

This is **unsupervised voice conversion**.

Voice Conversion

Voice Conversion

- The same sentence has different impact when it is said by different people.

Do you want to study a PhD?

Go away!

Student

新垣結衣
(Aragaki Yui)

Do you want to study a PhD?

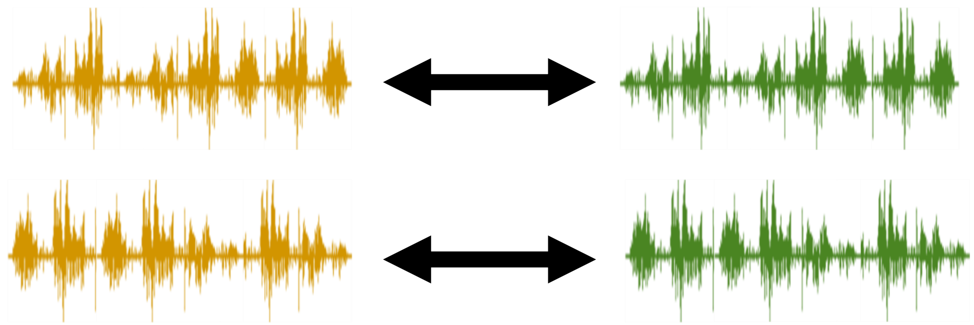
Student

In the past

Speaker A | Speaker B

How are you? | How are you?

Good morning | Good morning

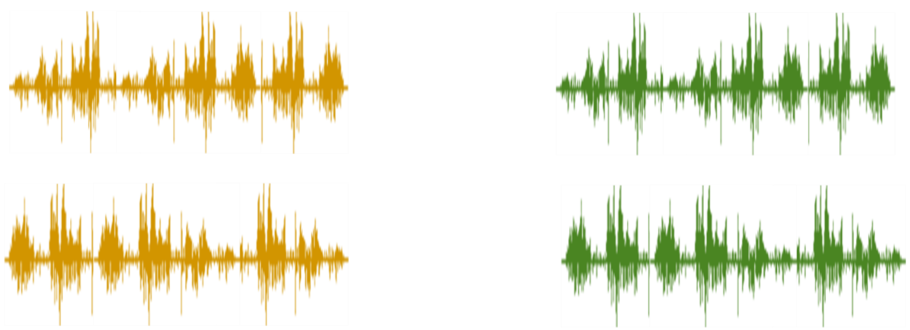


With GAN

Speaker A | Speaker B

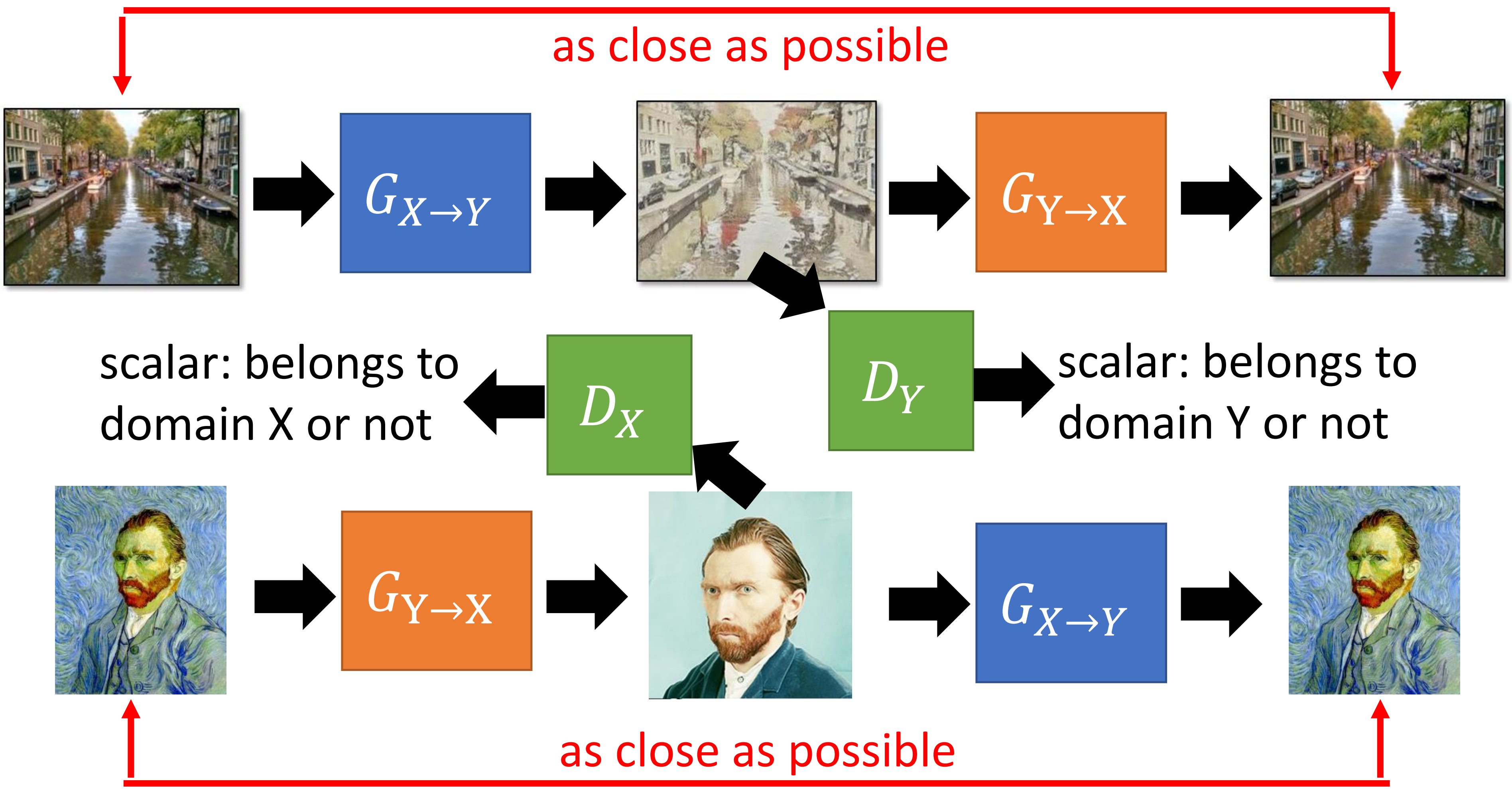
天氣真好 | How are you?

再見囉 | Good morning



Speakers A and B are talking about completely different things.

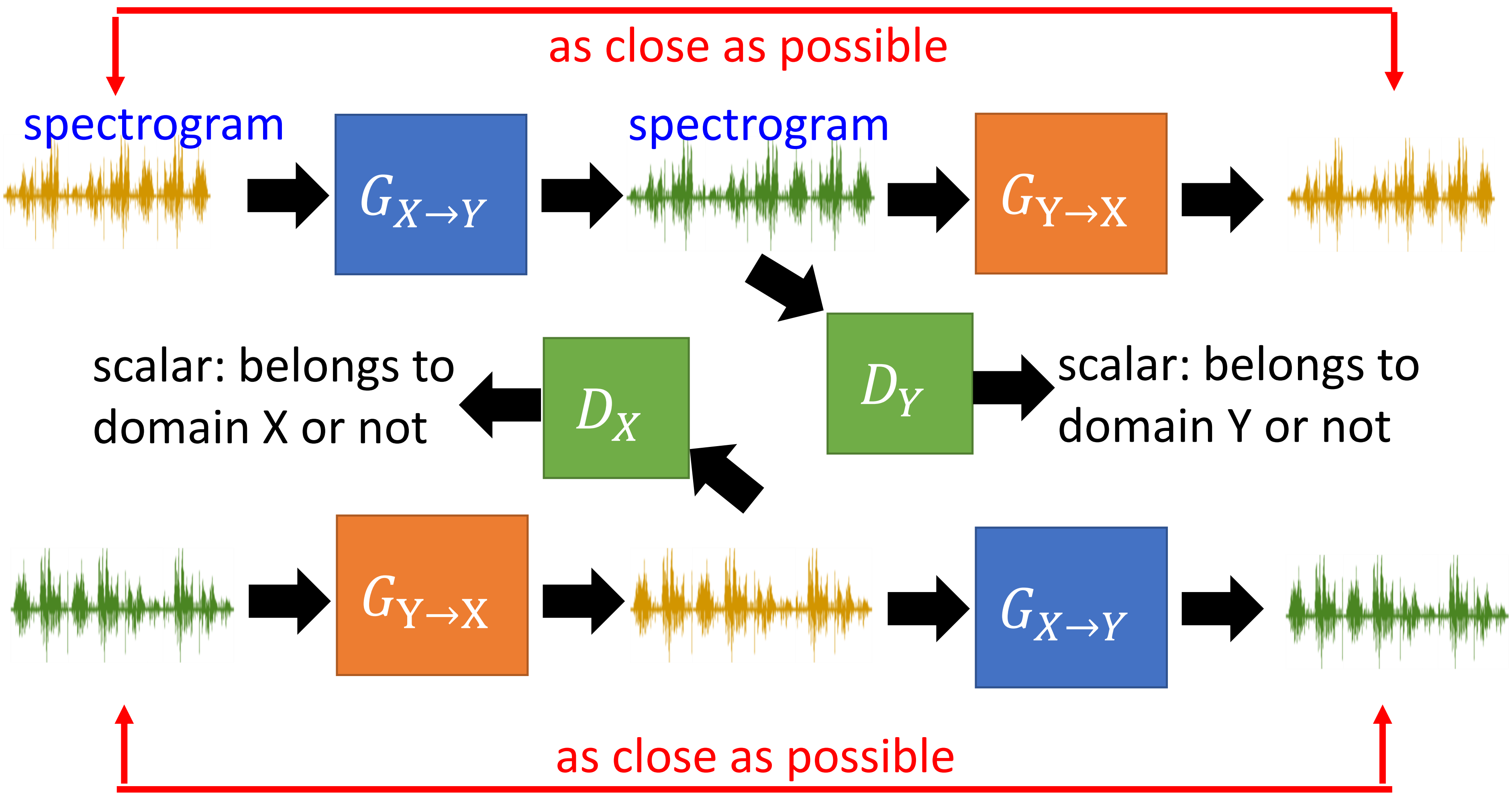
Cycle GAN


as close as possible
$G_{X \to Y}$
$G_{Y \to X}$
D_X
D_Y
scalar: belongs to domain X or not
scalar: belongs to domain Y or not
$G_{Y \to X}$
$G_{X \to Y}$
as close as possible

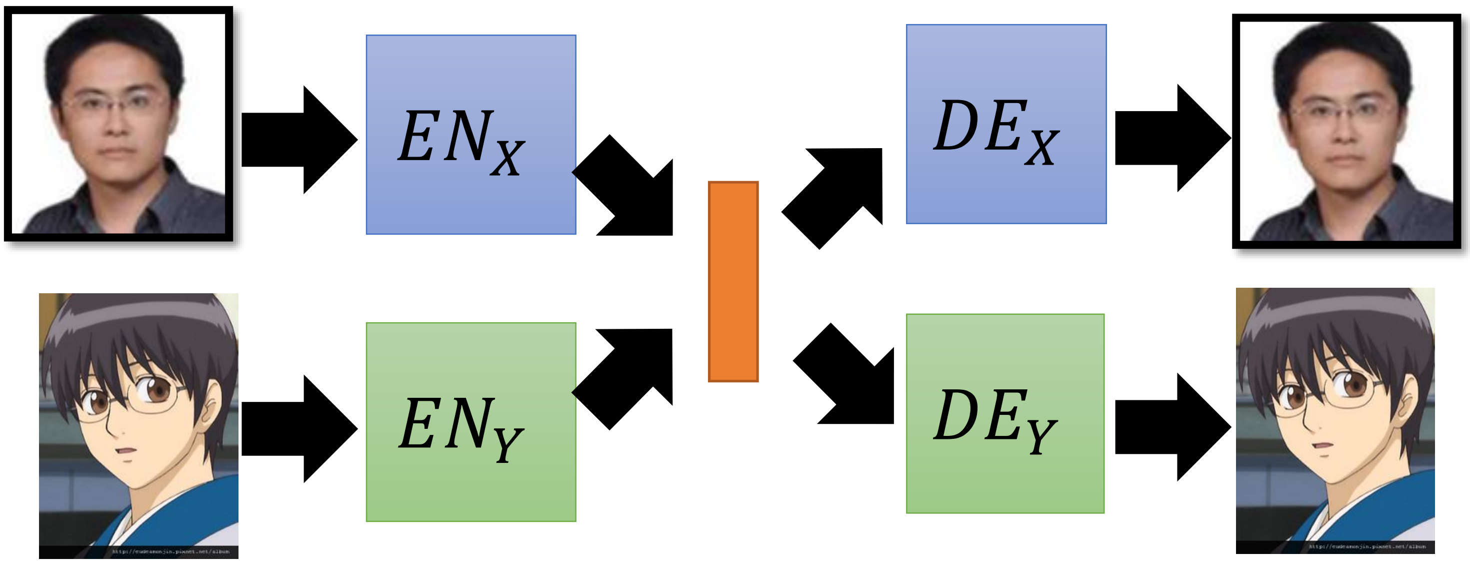
Cycle GAN for Voice Conversion

X: Speaker A, Y: Speaker B

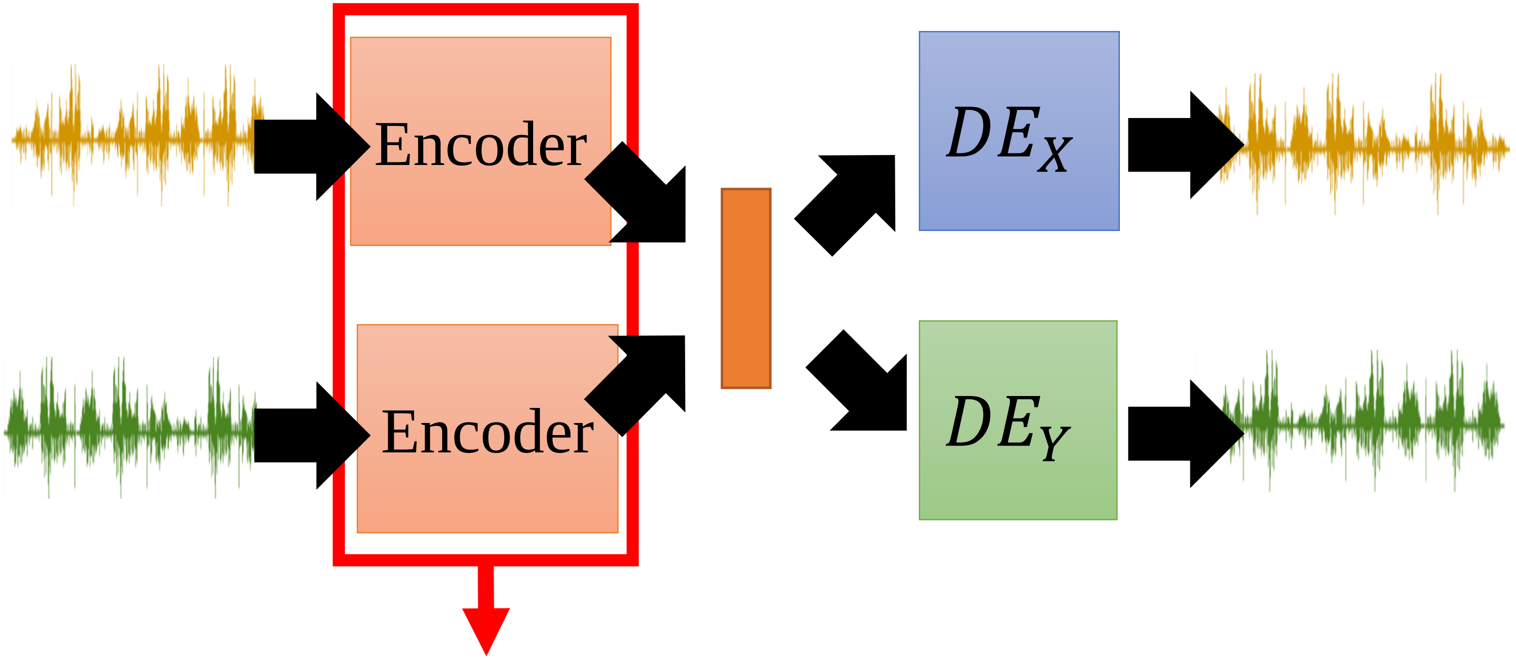
[Takuhiro Kaneko, et. al, arXiv, 2017][Fuming Fang, et. al, ICASSP, 2018][Yang Gao, et. al, ICASSP, 2018]



Projection to Common Space

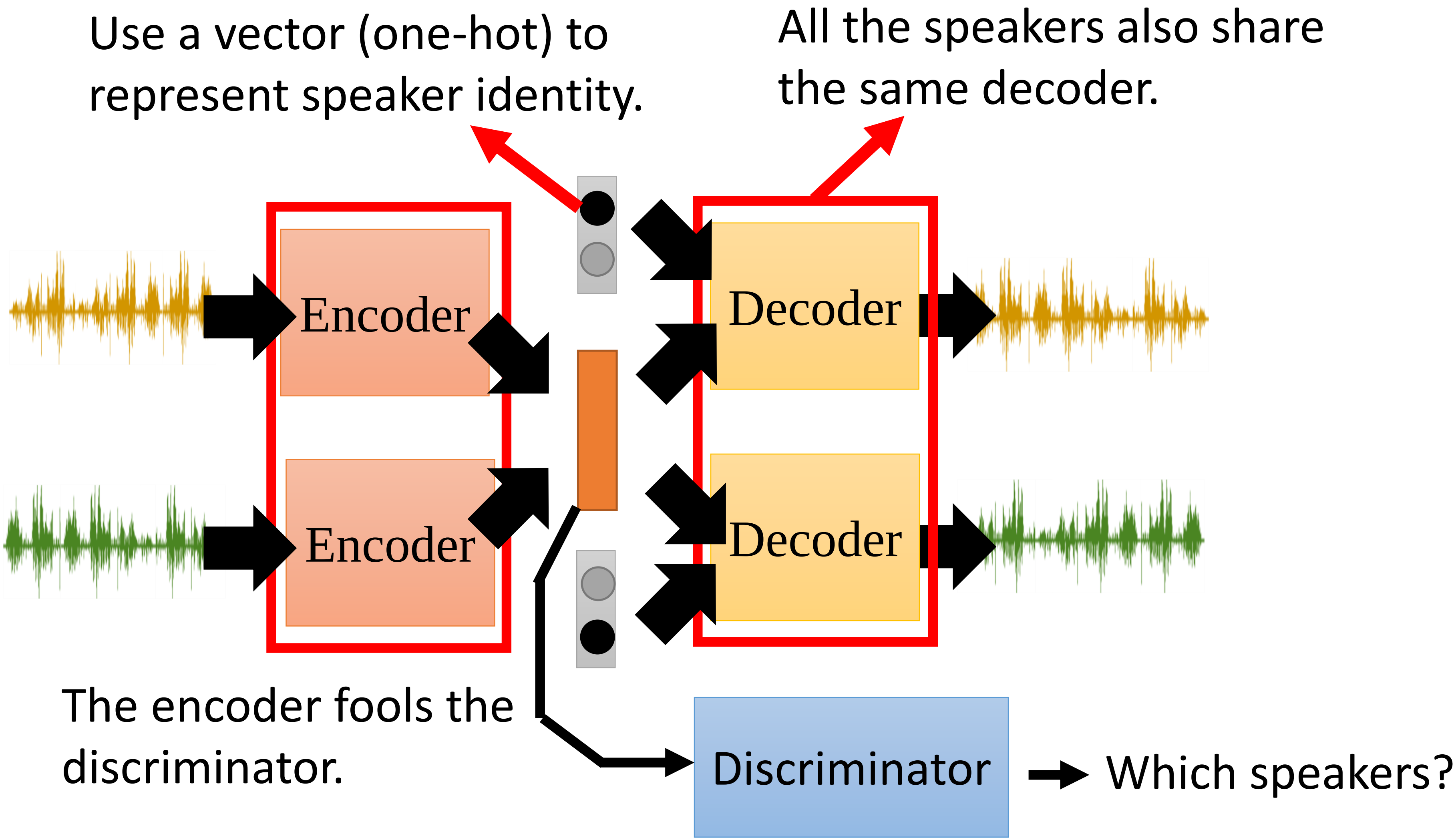


Projection to Common Space



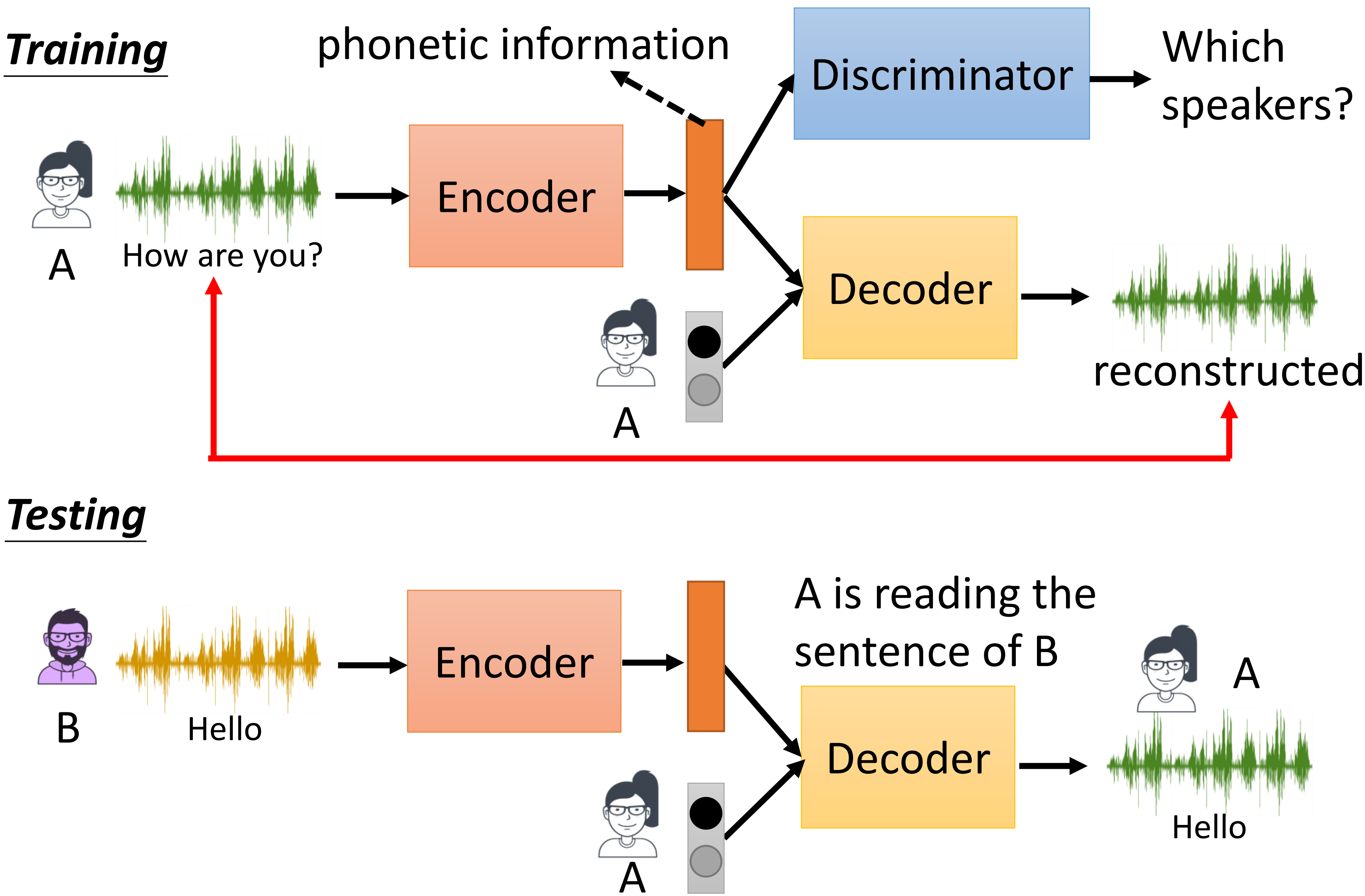
- All the speakers share the same encoder.
- The model can deal with the speakers never seen during training.

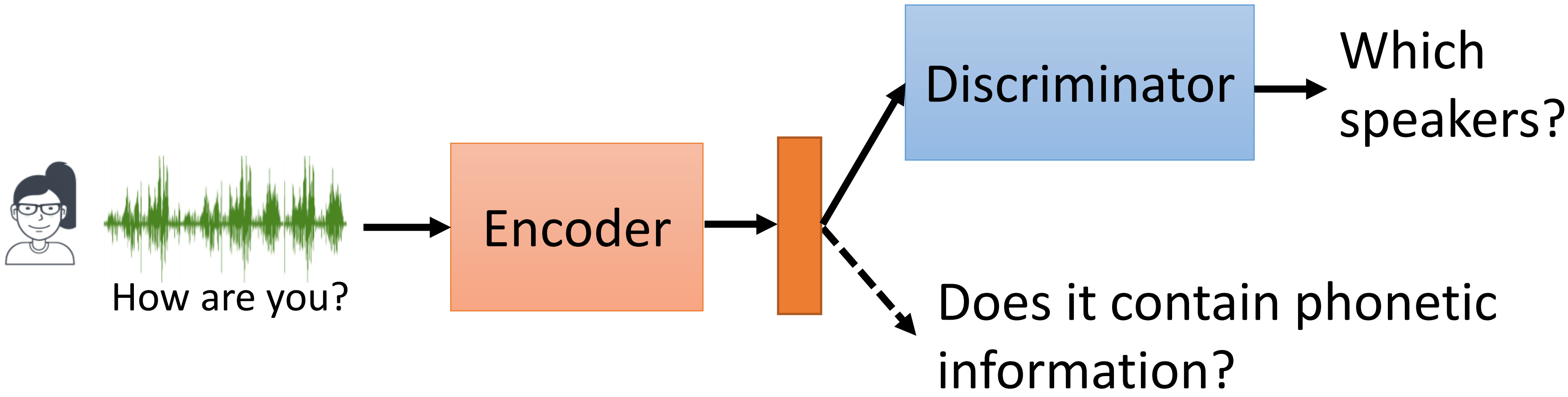
Projection to Common Space

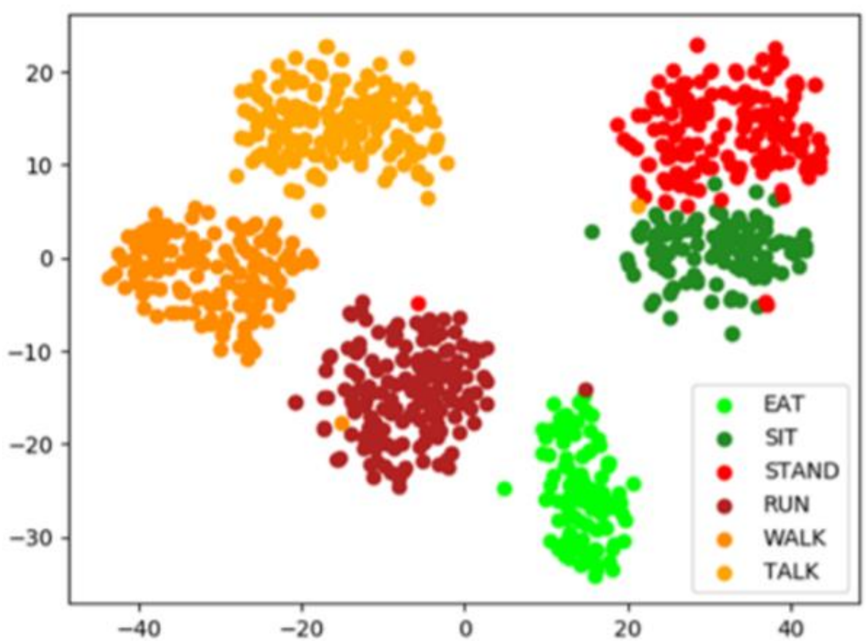


We hope that encoder can extract the phonetic information while removing the speaker information.

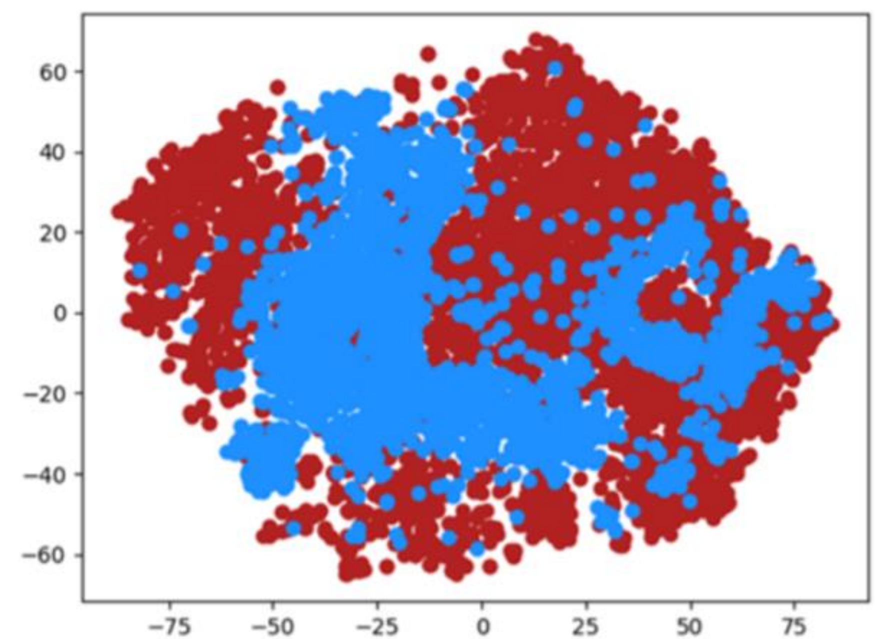
Projection to Common Space



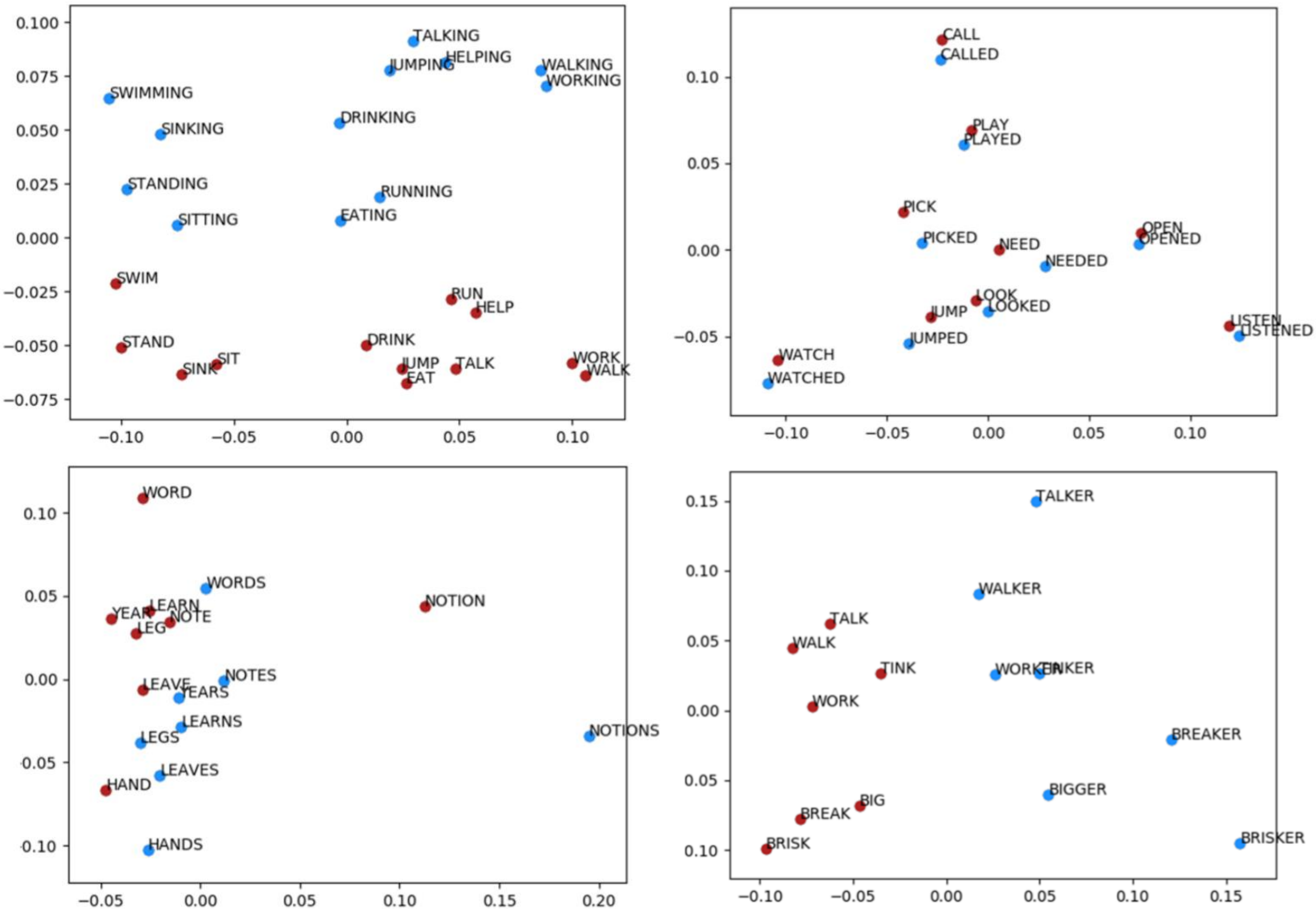




Different colors:
different words

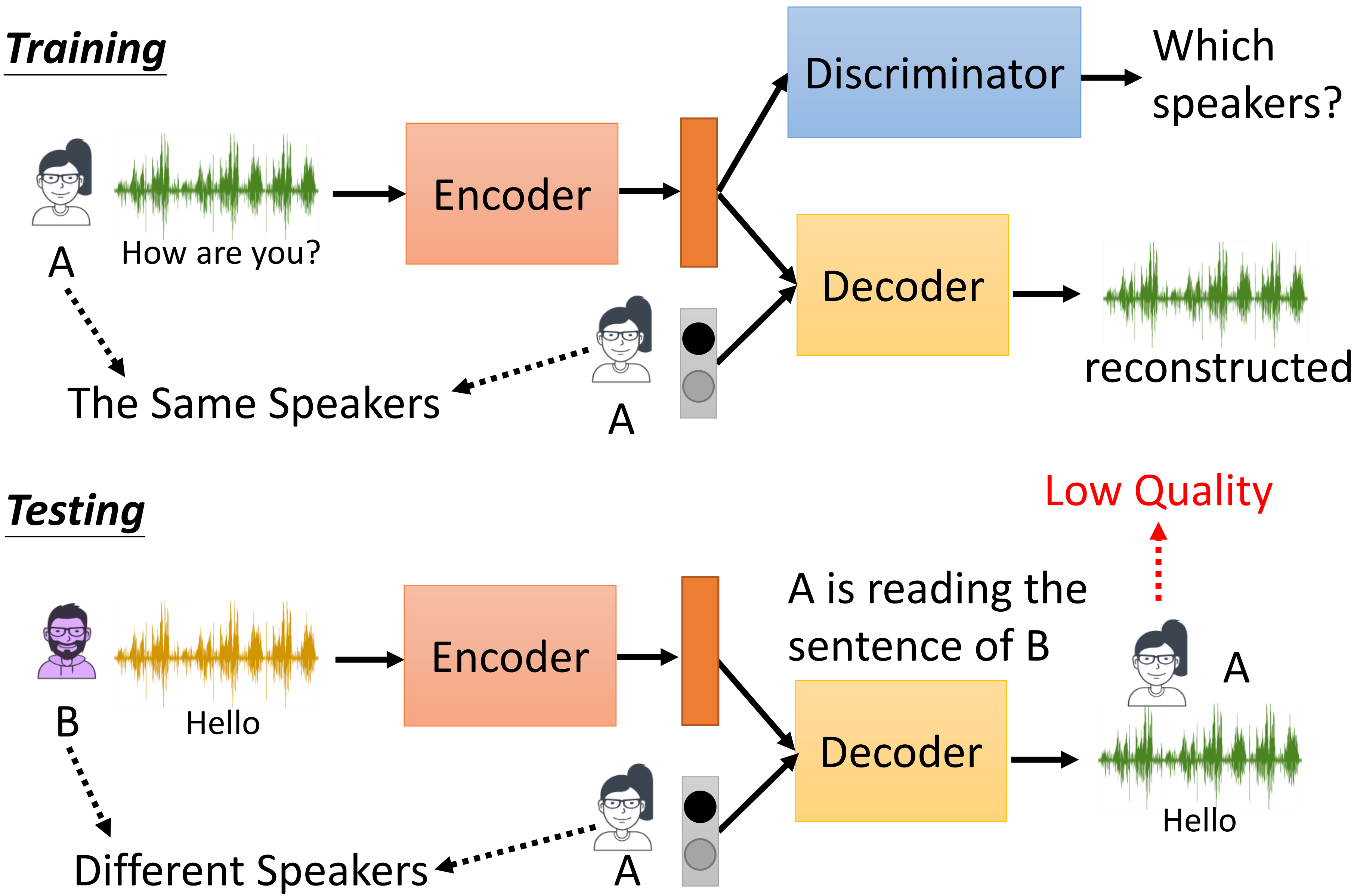


Different colors:
different speakers

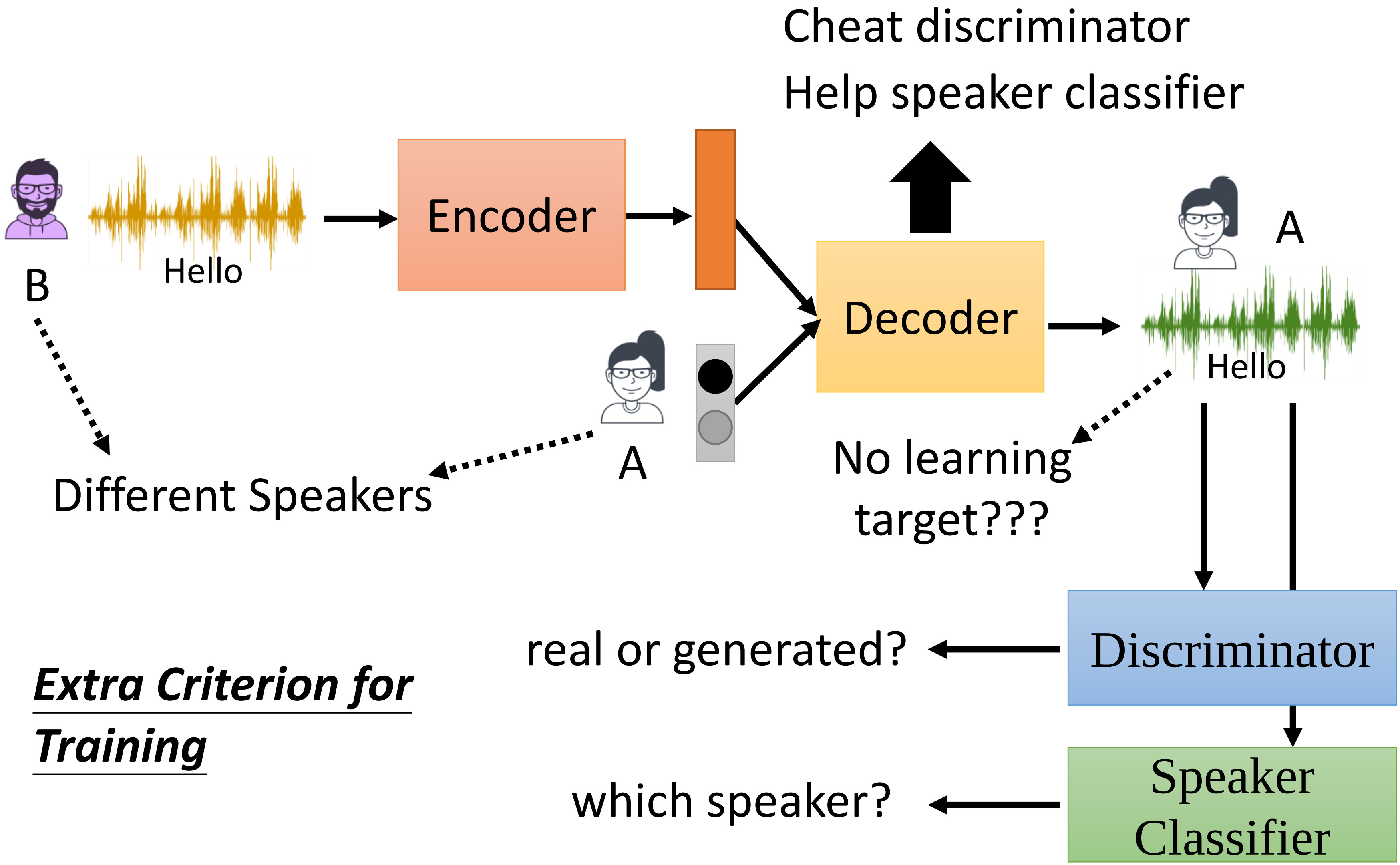


“Audio” Word to Vector

Issues


Training
A
How are you?
Encoder
Discriminator
Which speakers?
Decoder
reconstructed
The Same Speakers
A
Testing
Low Quality
B
Hello
Encoder
A is reading the sentence of B
A
Decoder
Hello
Different Speakers
A

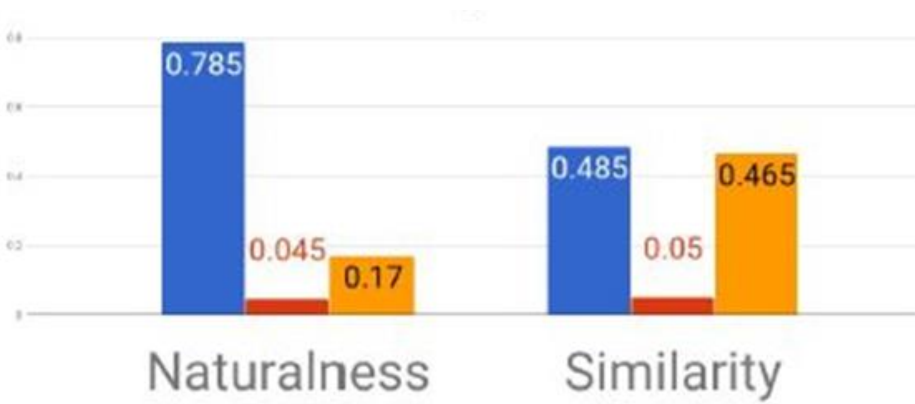
2nd Stage Training

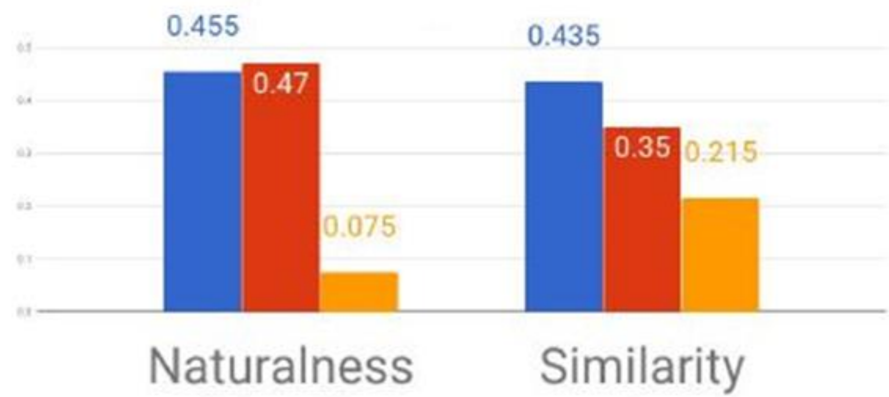


Extra Criterion for Training

Experimental Results

- Subjective evaluations(20 speakers in VCTK)

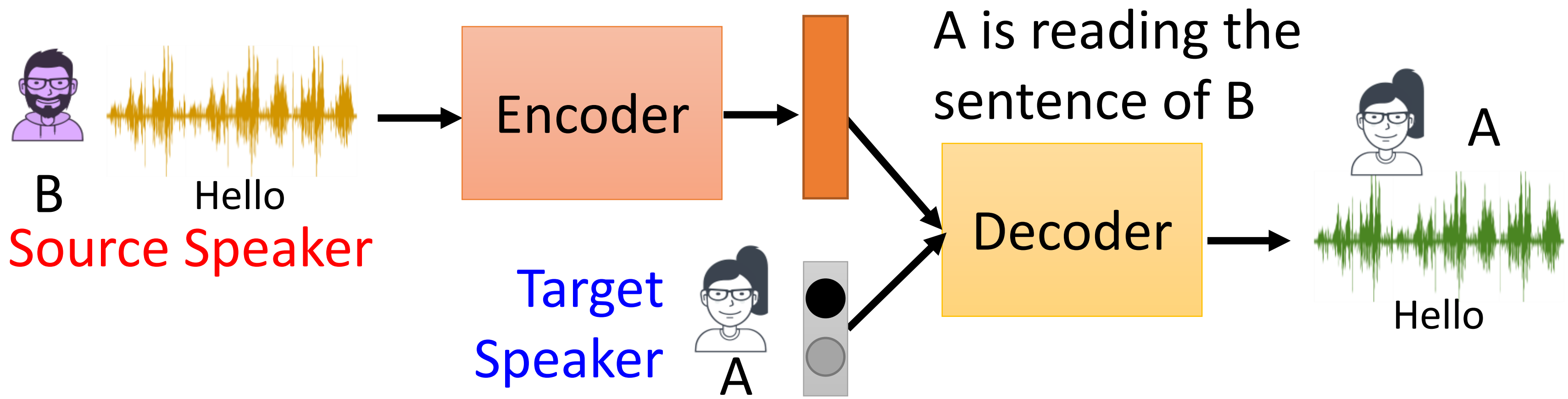




“Two stages” is better
“One stage” is better
Indistinguishable

“Projection” is better
“Cycle GAN” is better
Indistinguishable

Demo



Source:

Target:

Source to Target:

Thanks Ju-chieh Chou for providing the results.

https://jjery2243542.github.io/voice_conversion_demo/

Target Speaker

Source Speaker

(Never seen during training!)

Source to Target

Me

Me

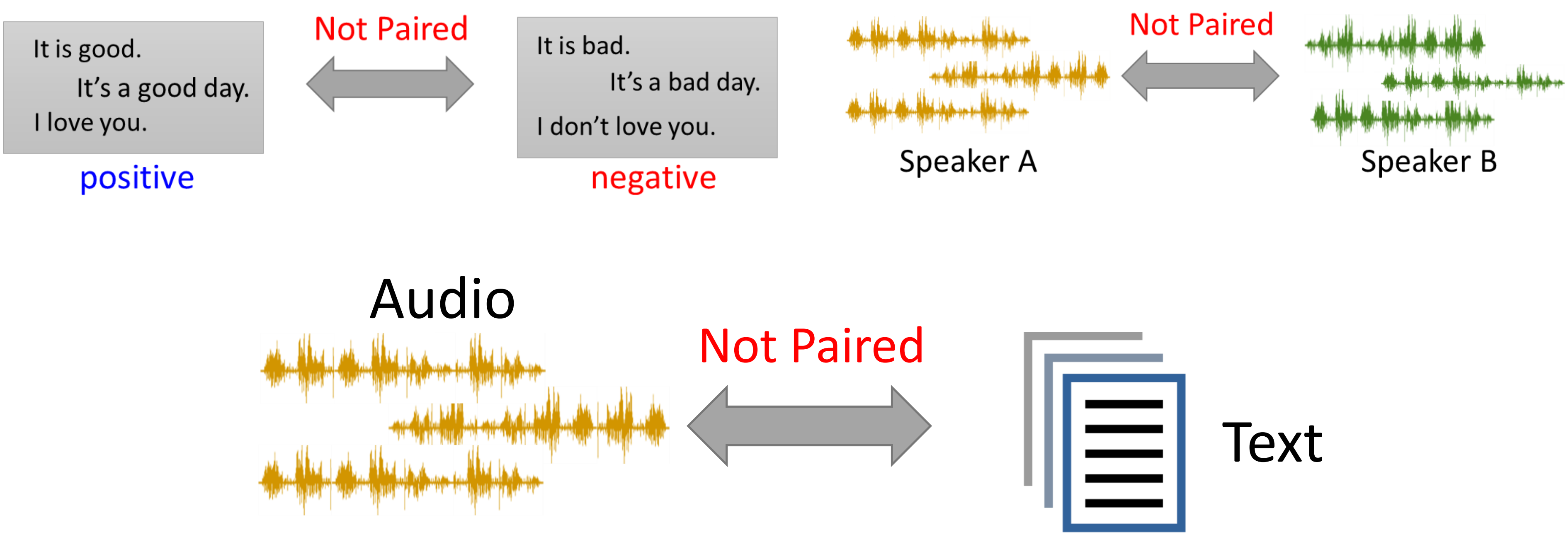
Me

Me

Thanks Ju-chieh Chou for providing the results.

https://jjery2243542.github.io/voice_conversion_demo/

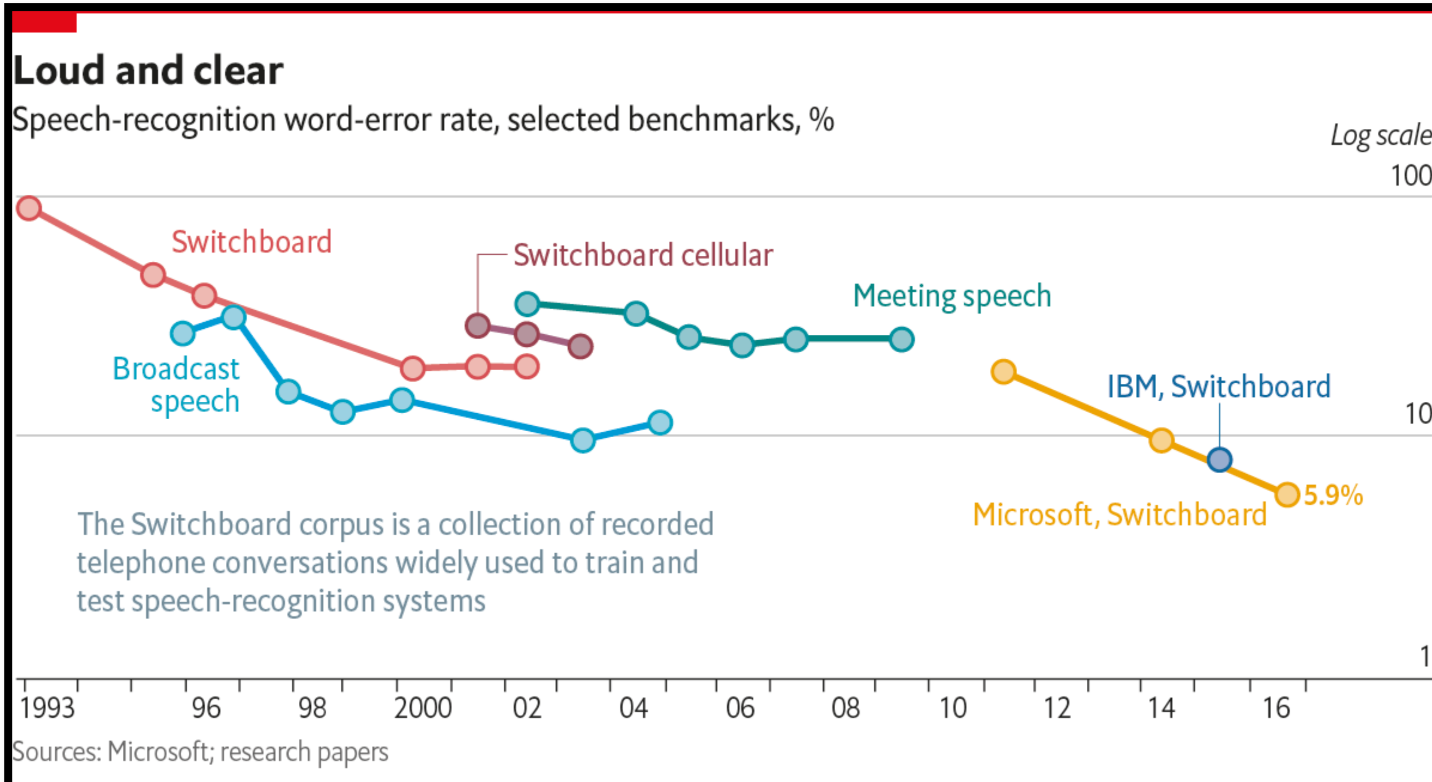
Unsupervised
Conditional Generation



This is **unsupervised speech recognition**.

https://devopedia.org/images/article/102/9180.1532710057.png

Supervised Speech Recognition



(I believe you have seen similar figures before.)

- Supervised learning needs lots of annotated speech.
- However, most of the languages are low resourced.

http://www.parenting.com/article/teach-baby-to-talk

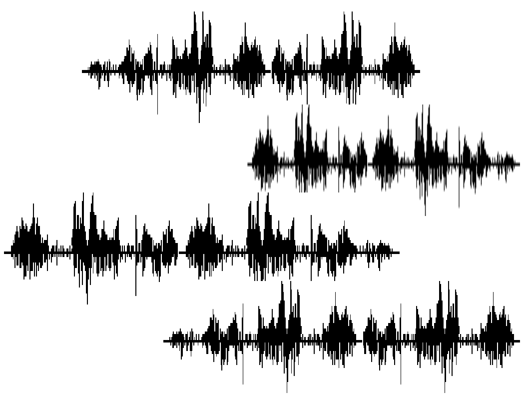
Speech Recognition in the Future

Learning human language with very little supervision

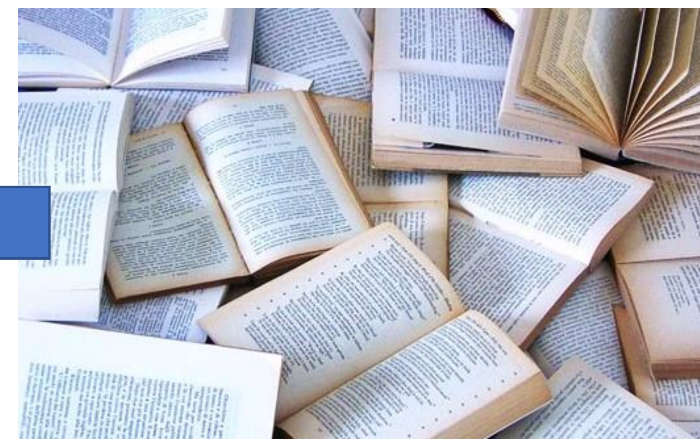
[Liu, et al., INTERSPEECH, 2018]
[Chen, et al., arXiv, 2018]

Unsupervised Speech Recognition

- Machine learns to recognize speech from unparallel speech and text.



audio collection
(without text annotation)

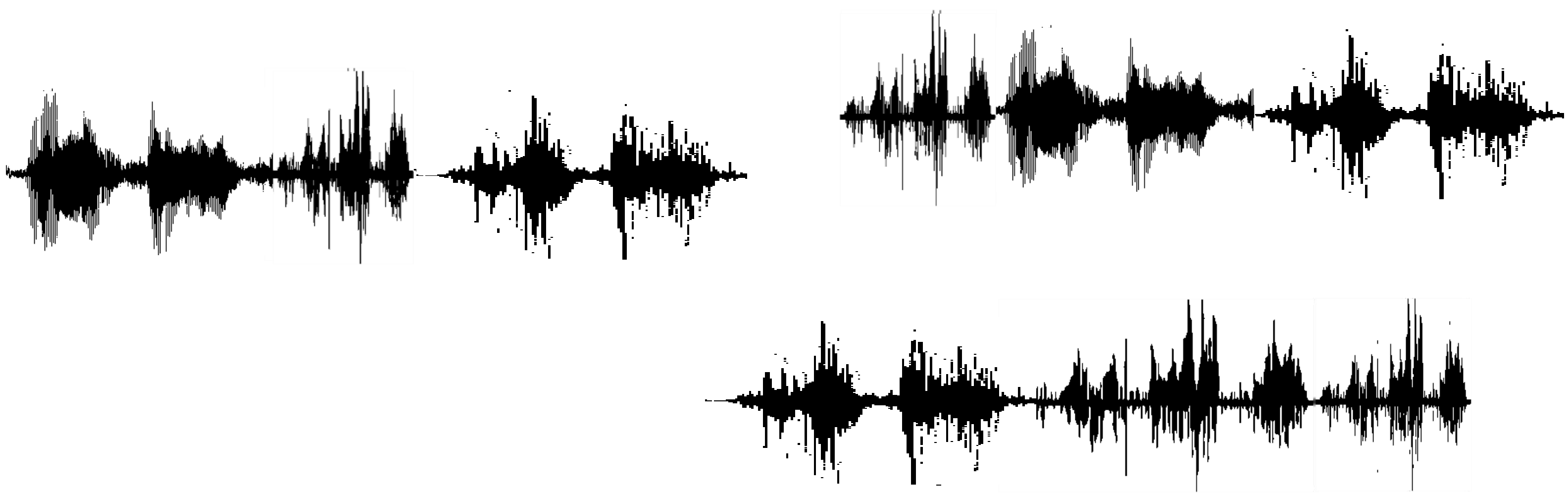


text documents
(not parallel to audio)

This idea was too crazy to be realized in the past.

However, it becomes possible with GAN recently.

Acoustic Token Discovery

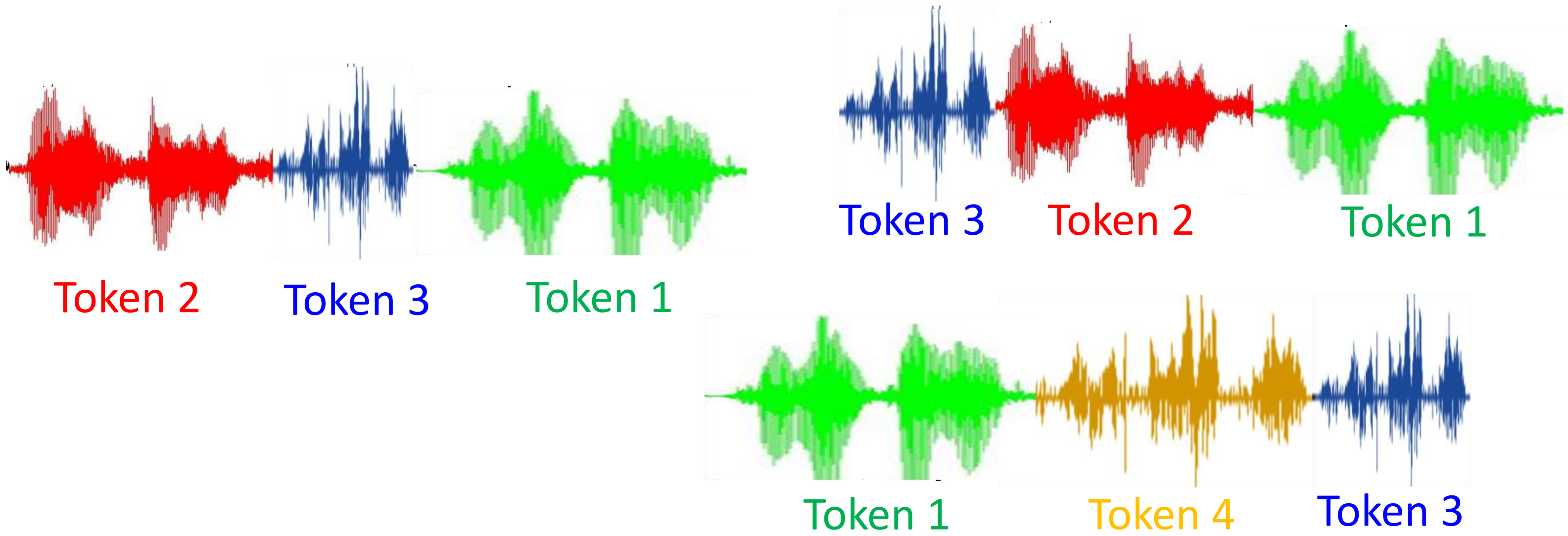


Acoustic tokens can be discovered from audio collection without text annotation.

Acoustic tokens: chunks of acoustically similar audio segments with token IDs

[Zhang & Glass, ASRU 09]
[Huijbregts, ICASSP 11]
[Chan & Lee, Interspeech 11]

Acoustic Token Discovery



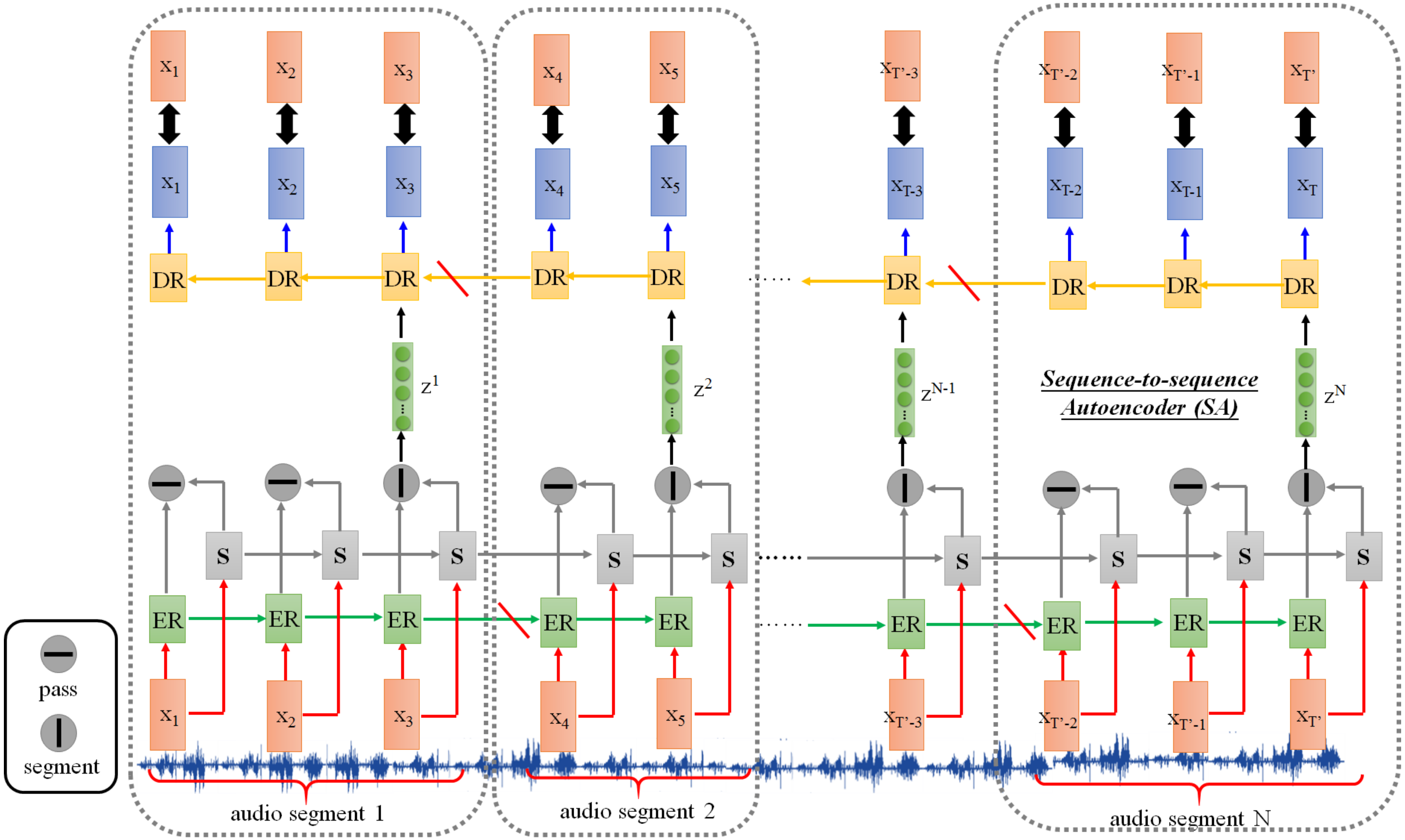
Acoustic tokens can be discovered from audio collection without text annotation.

Acoustic tokens: chunks of acoustically similar audio segments with token IDs

[Zhang & Glass, ASRU 09]
[Huijbregts, ICASSP 11]
[Chan & Lee, Interspeech 11]

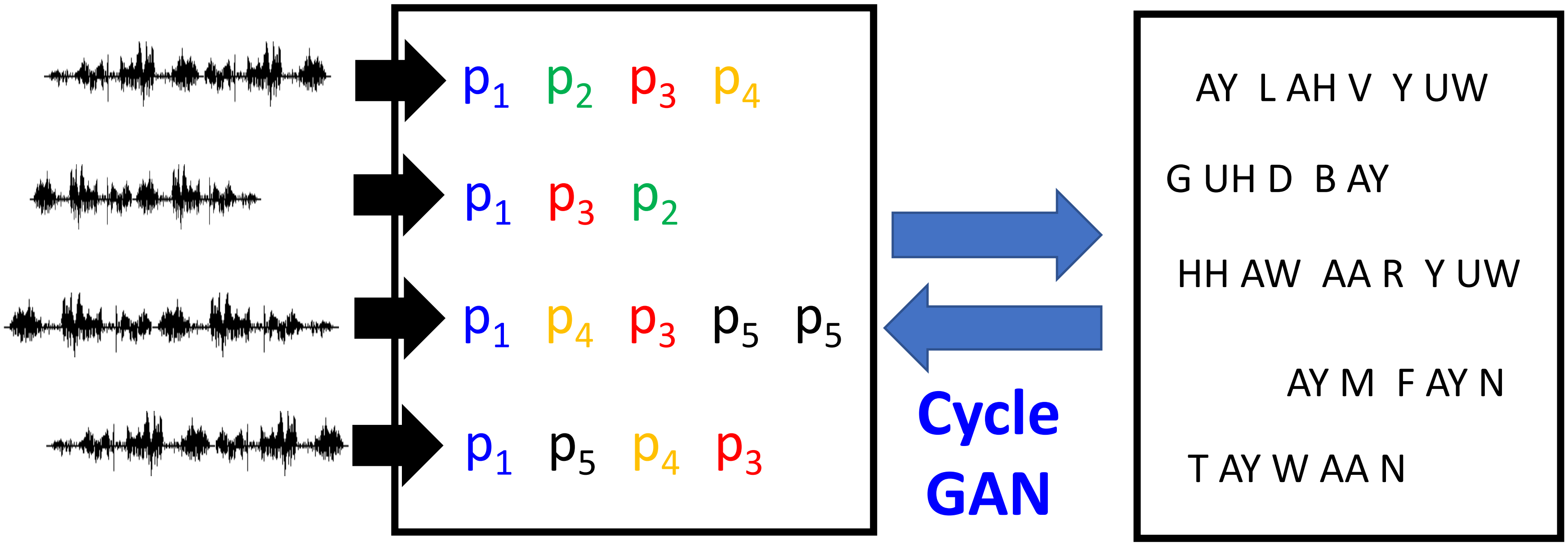
[Wang, et al., ICASSP, 2018]

Acoustic Token Discovery



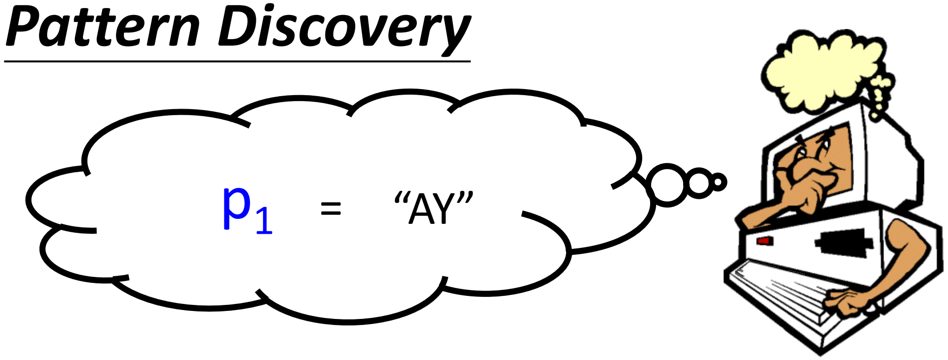
Phonetic-level acoustic tokens are obtained by segmental sequence-to-sequence autoencoder.

Unsupervised Speech Recognition



Phone-level Acoustic Pattern Discovery

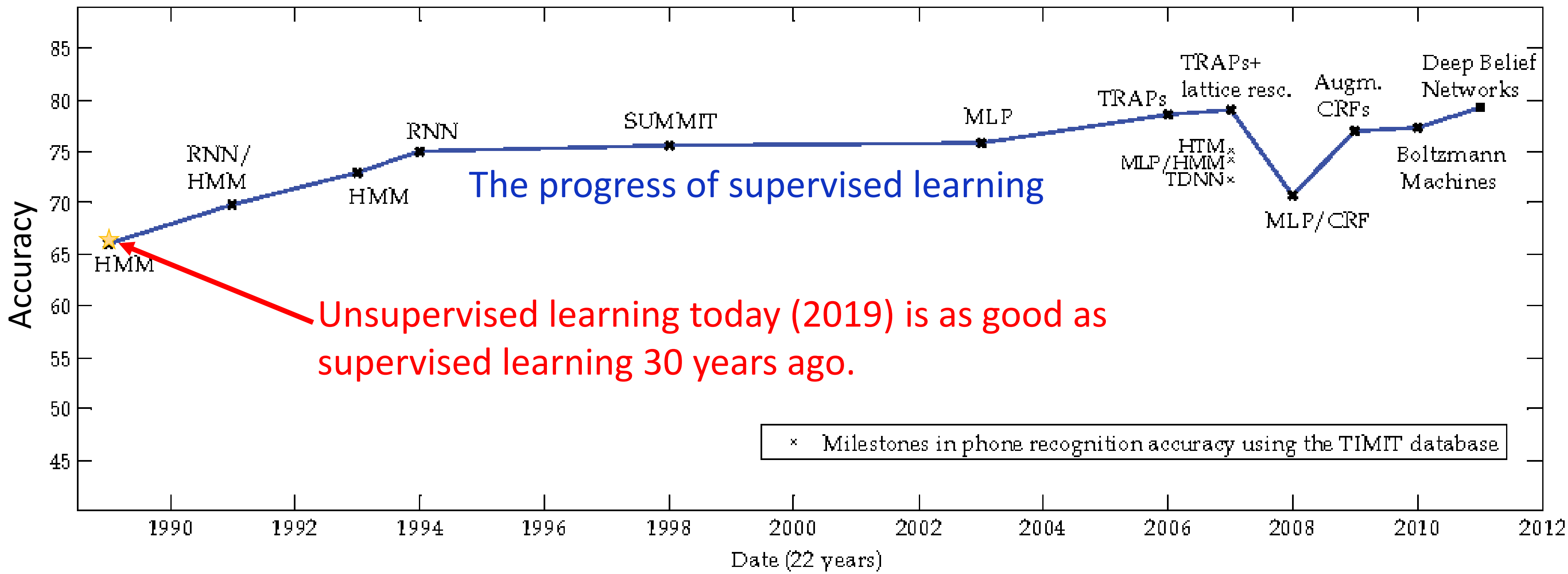
Phoneme sequences from Text



[Liu, et al., INTERSPEECH, 2018]

[Chen, et al., arXiv, 2018]

Approaches			Matched (all 4000)		Nonmatched (3000/1000)	
			FER	PER	FER	PER
(I) Supervised (labeled)						
(a) RNN Transducer [23]			-	17.7	-	-
(b) standard HMMs			-	21.5	-	-
(c) Phoneme classifier			27.0	28.9	-	-
(II) Unsupervised (with oracle boundaries)						
(d) Relationship mapping GAN [22]			40.5	40.2	43.6	43.4
(e) Segmental Emperical-ODM [23]			33.3	32.5	40.0	40.1
(f) Proposed: GAN			27.6	28.5	32.7	34.3
(III) Completely unsupervised (no label at all)						
(g) Segmental Emperical-ODM [23]			-	36.5	-	41.6
Proposed	iteration 1	(h) GAN	48.3	48.6	50.3	50.0
		(i) GAN/HMM	-	30.7	-	39.5
	iteration 2	(j) GAN	41.0	41.0	44.3	44.3
		(k) GAN/HMM	-	27.0	-	35.5
	iteration 3	(l) GAN	39.7	38.4	45.0	44.2
		(m) GAN/HMM	-	26.1	-	33.1



The image is modified from: Phone recognition on the TIMIT database Lopes, C. and Perdigão, F., 2011. Speech Technologies, Vol 1, pp. 285--302.

Concluding Remarks

Part I: General Introduction of Generative Adversarial Network (GAN)

Part II: Applications to Natural Language Processing

Part III: Applications to Speech Processing

To Learn More …

You can learn more from the YouTube Channel

https://www.youtube.com/playlist?list=PLJV_el3uVTsMd2G9ZjcpJn1YfnM9wVOBf

(in Mandarin)