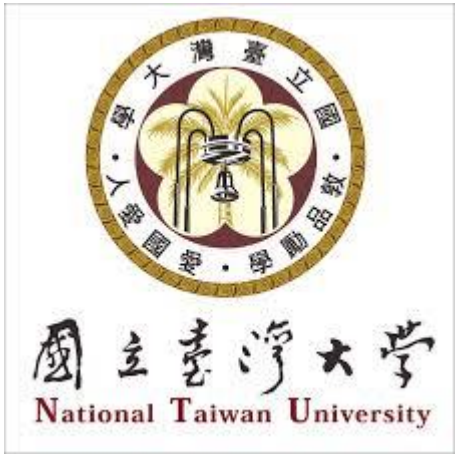
# Generative Adversarial Network

## and its Applications to Speech Processing and Natural Language Processing

Hung-yi Lee and Yu Tsao




# Outline

Part I: Basic Idea of Generative Adversarial Network (GAN)

Part II: A little bit theory

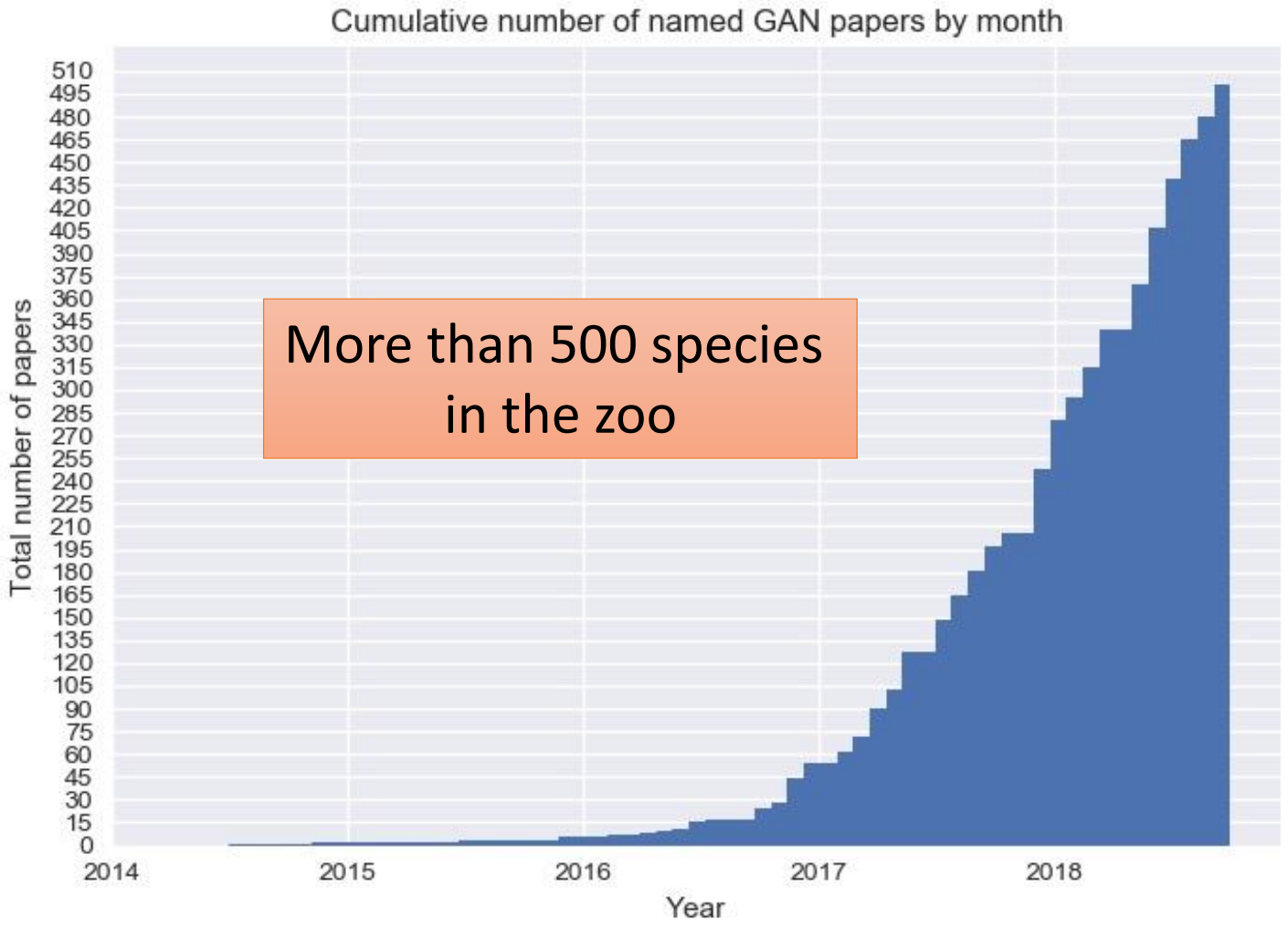
Take a break

Part III: Applications to Speech Processing

Part IV: Applications to Natural Language Processing

# *All Kinds of GAN …*

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




(not updated since 2018.09)

# *All Kinds of GAN …*

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

GAN

ACGAN

BGAN

CGAN

DCGAN

EBGAN

fGAN

GoGAN

⋮

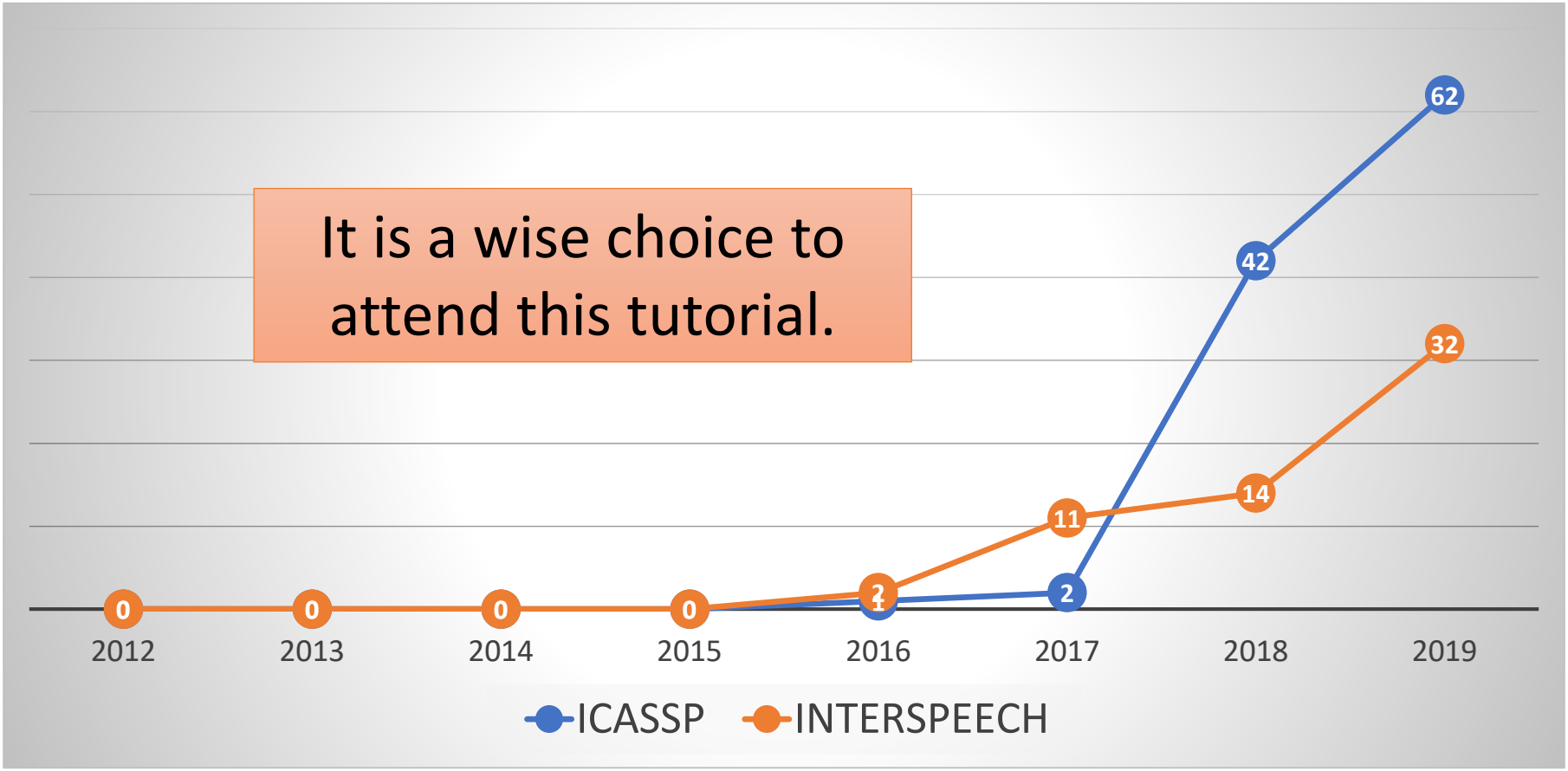
- SeUDA - Semantic-Aware Generative Adversarial Nets for Unsupervised Domain Adapt Segmentation
- SG-GAN - Semantic-aware Grad-GAN for Virtual-to-Real Urban Scene Adaption (githu
- SG-GAN - Sparsely Grouped Multi-task Generative Adversarial Networks for Facial Attr
- SGAN - Texture Synthesis with Spatial Generative Adversarial Networks
- SGAN - Stacked Generative Adversarial Networks (github)
- SGAN - Steganographic Generative Adversarial Networks
- SGAN - SGAN: An Alternative Training of Generative Adversarial Networks
- SGAN - CT Image Enhancement Using Stacked Generative Adversarial Networks and T Segmentation Improvement
- sGAN - Generative Adversarial Training for MRA Image Synthesis Using Multi-Contrast
- SiftingGAN - SiftingGAN: Generating and Sifting Labeled Samples to Improve the Rem Classification Baseline in vitro
- SiGAN - SiGAN: Siamese Generative Adversarial Network for Identity-Preserving Face H
- SimGAN - Learning from Simulated and Unsupervised Images through Adversarial Trai
- SisGAN - Semantic Image Synthesis via Adversarial Learning

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

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

# INTERSPEECH & ICASSP

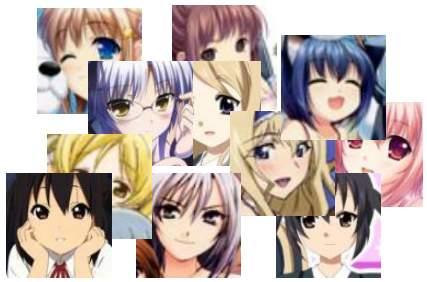
How many papers have "*adversarial*" in their titles?



# Part I: Basic Idea

# Three Categories of GAN

## 1. Generation



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

random vector

Generator

image

## 2. Conditional Generation

blue eyes,
red hair,
short hair

*paired data*

"Girl with red hair"

text

Generator

image

## 3. Unsupervised Conditional Generation

domain x

domain y

*unpaired data*

x

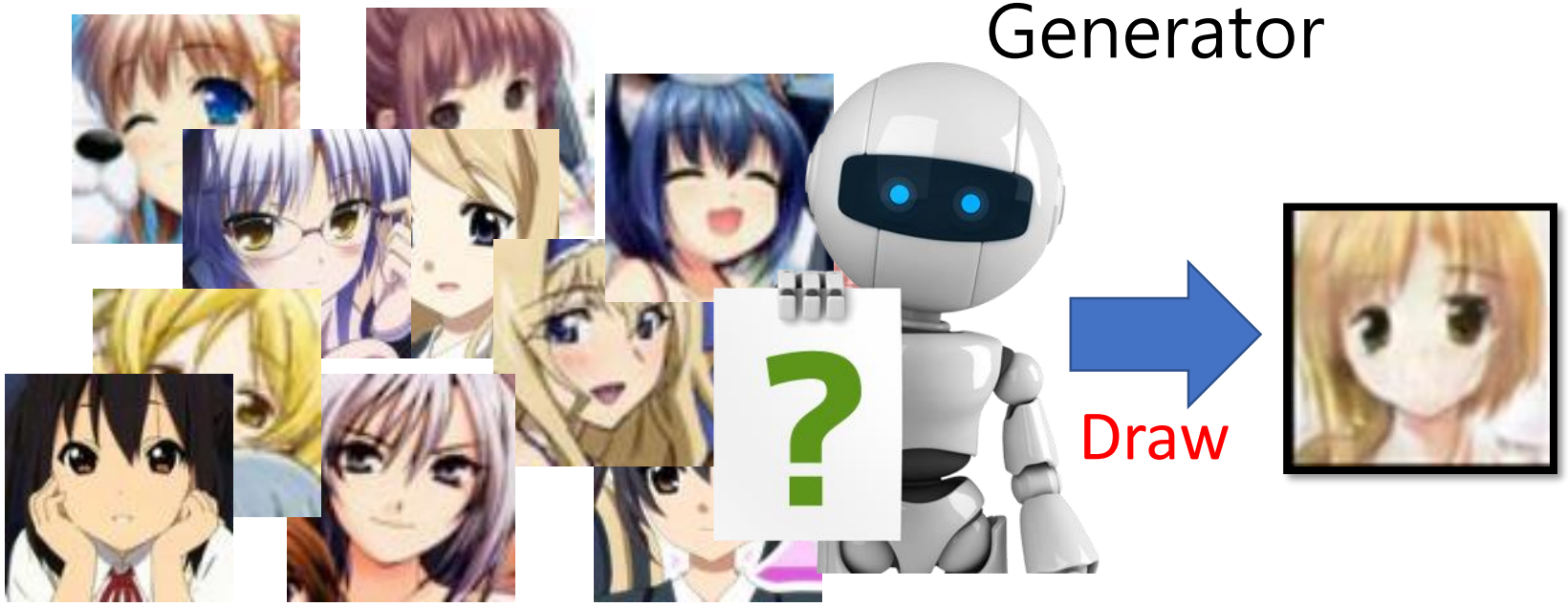
Photo

Generator

y

Vincent van Gogh's style

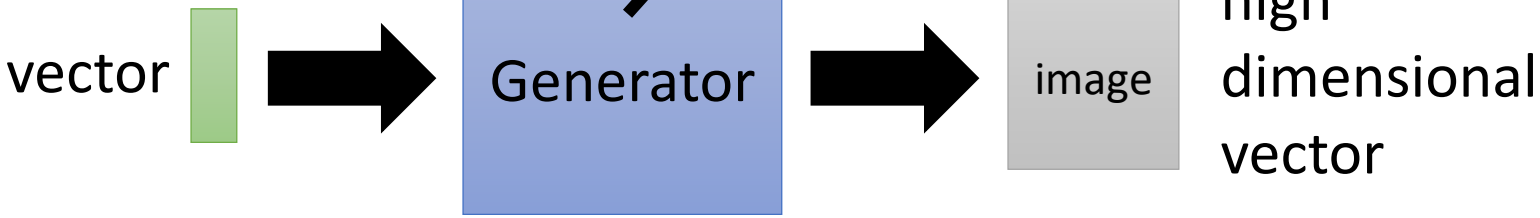
# Anime Face Generation

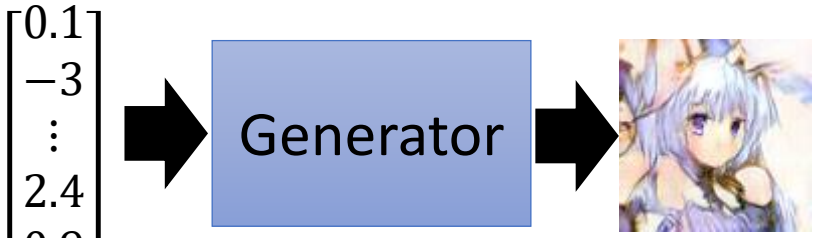


Powered by: http://mattya.github.io/chainer-DCGAN/

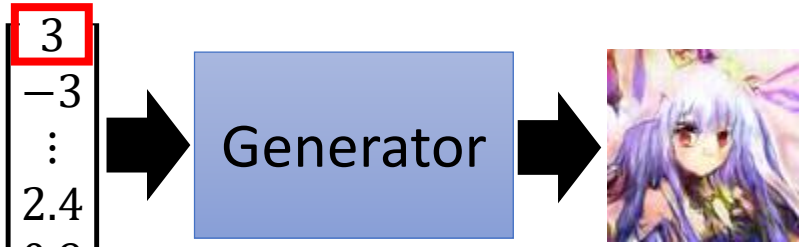
# Basic Idea of GAN

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

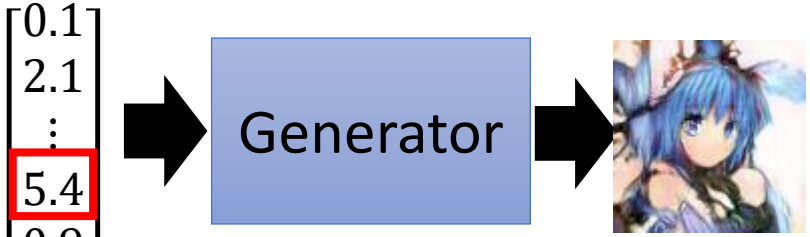


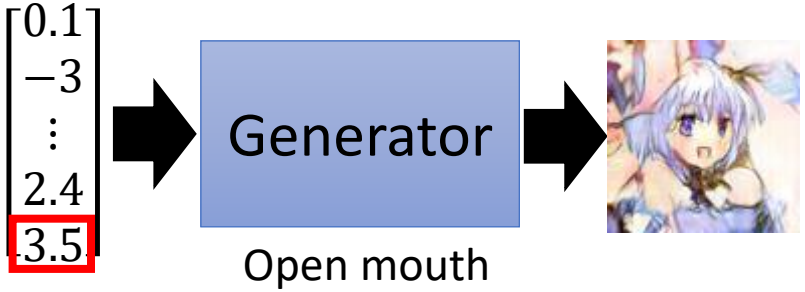

Each dimension of input vector represents some characteristics.




Longer hair




blue hair




Open mouth

# 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

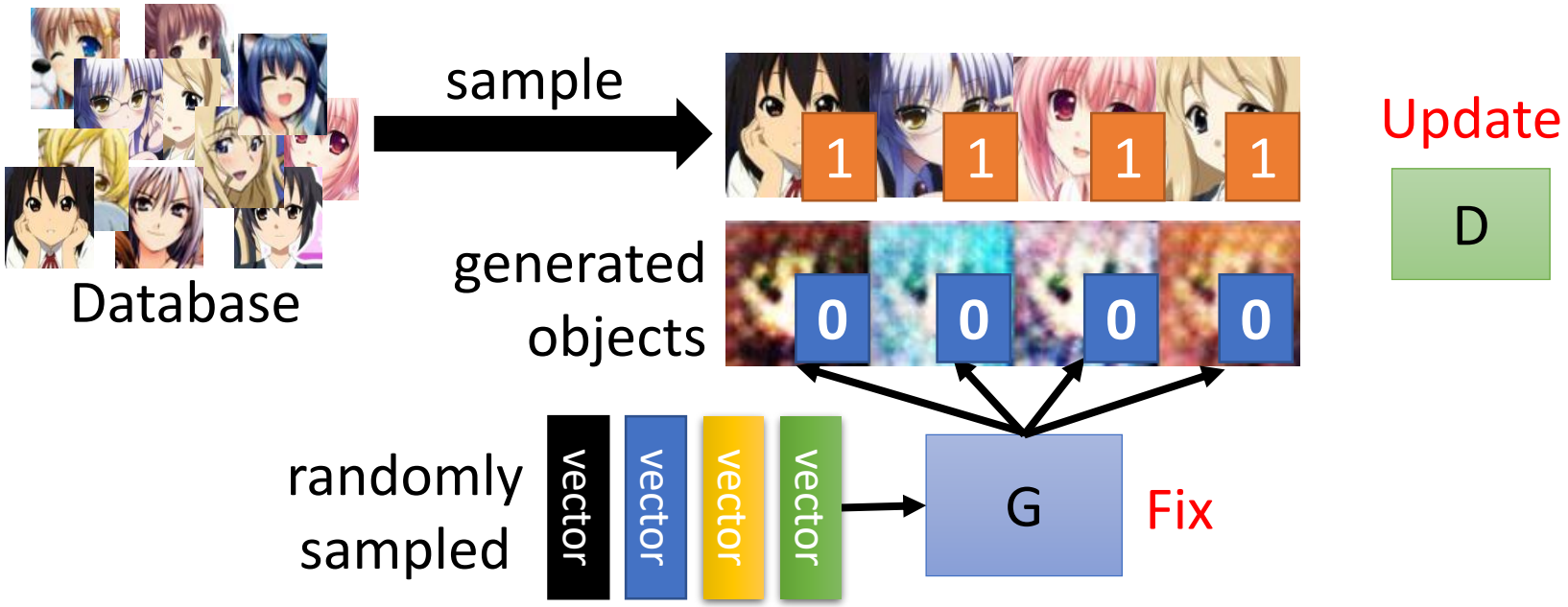
 → Discri-minator → 0.1

# 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.

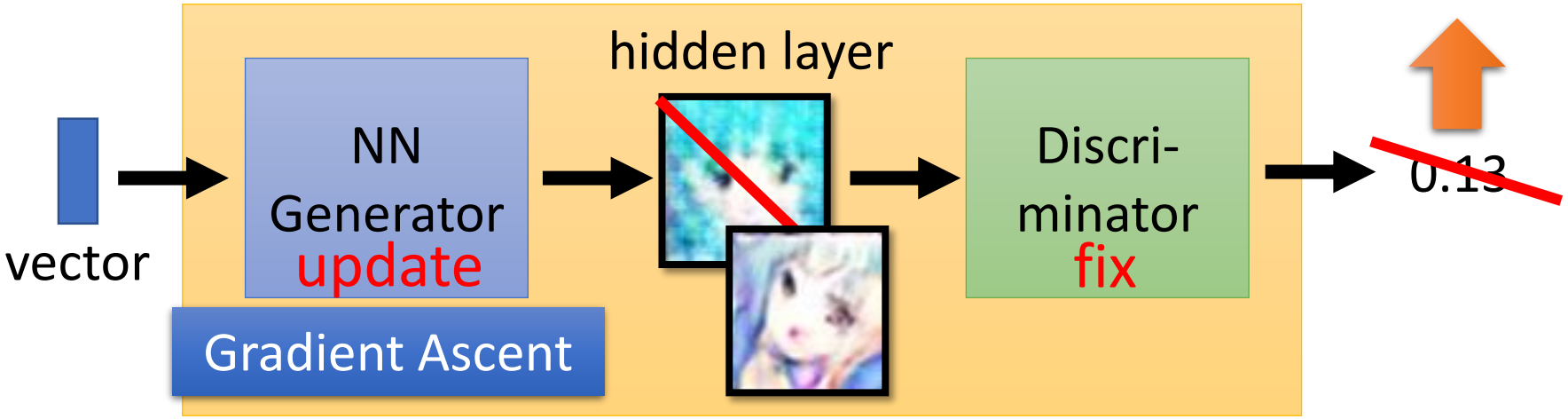
# Algorithm

- Initialize generator and discriminator
- In each training iteration:

G D

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

Generator learns to "fool" the discriminator

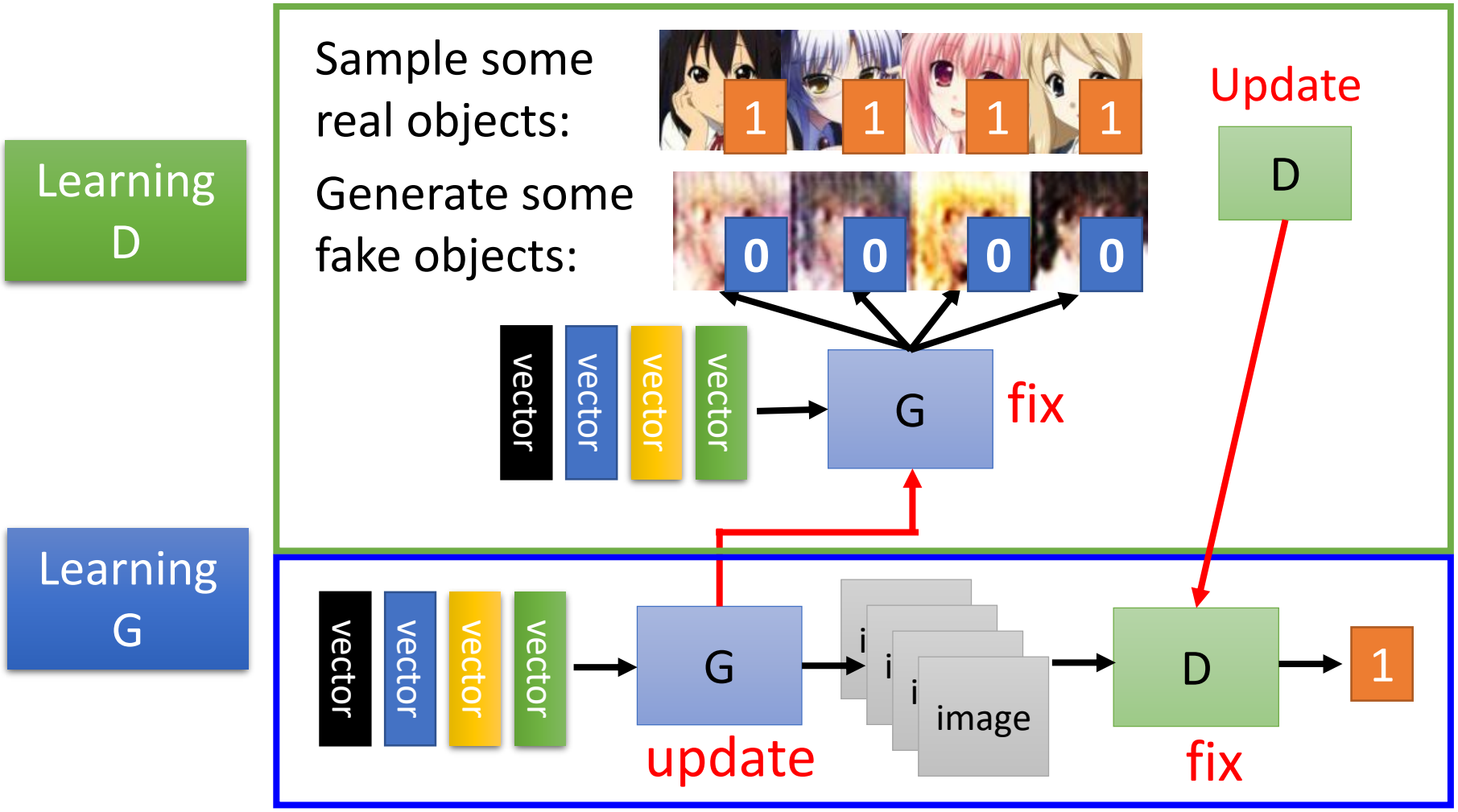



***large network***

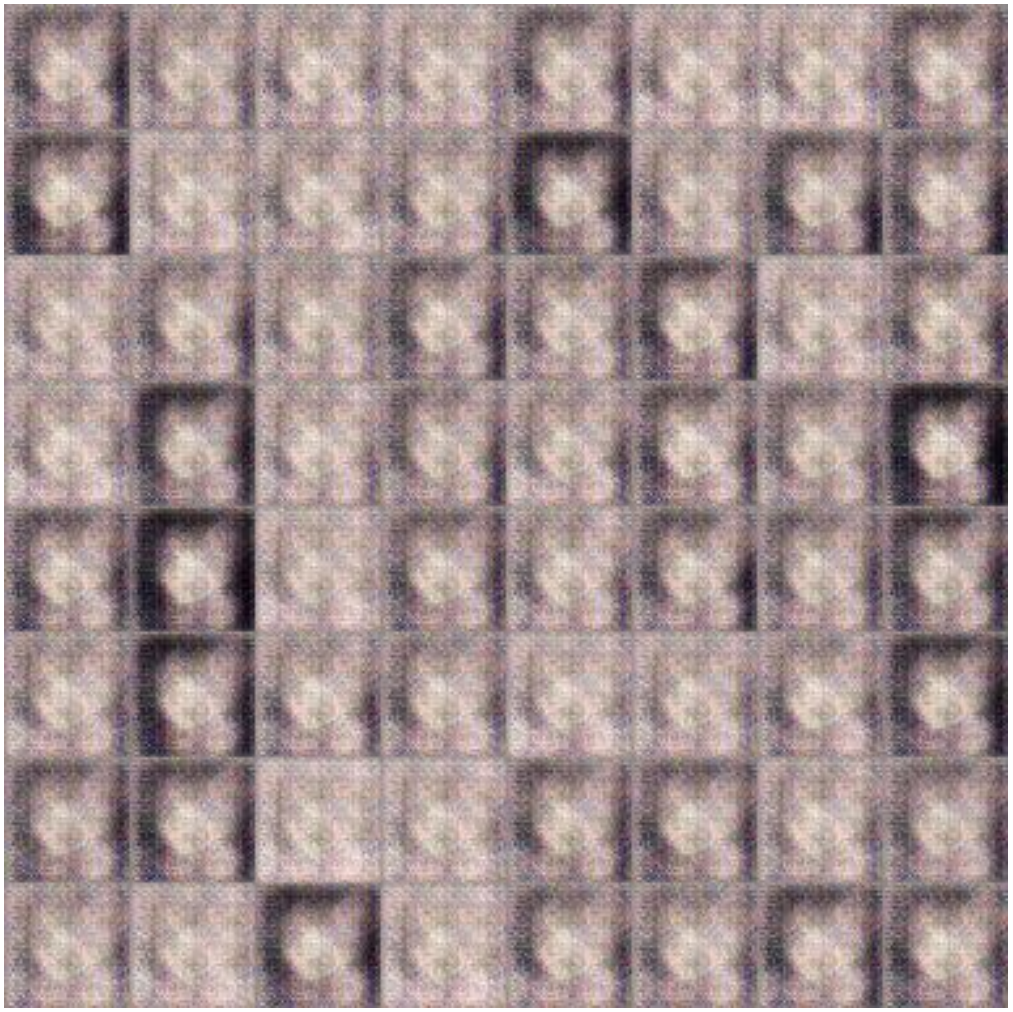
# *Algorithm*

- Initialize generator and discriminator


- In each training iteration:



# Anime Face Generation



100 updates

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

# Anime Face Generation

1000 updates

# Anime Face Generation

2000 updates

# Anime Face Generation

5000 updates

# Anime Face Generation

10,000 updates

# Anime Face Generation

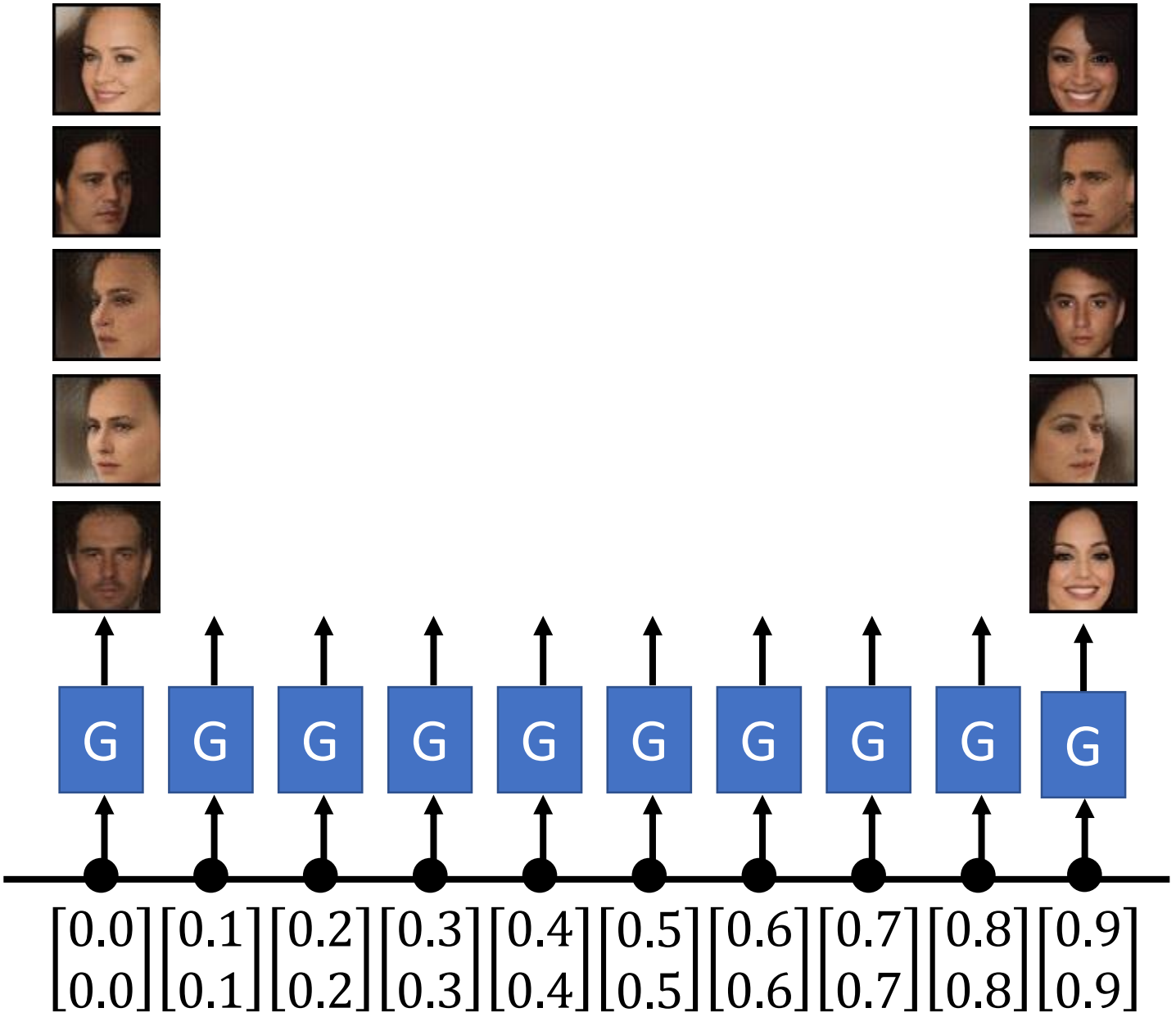
20,000 updates

# Anime Face Generation

50,000 updates

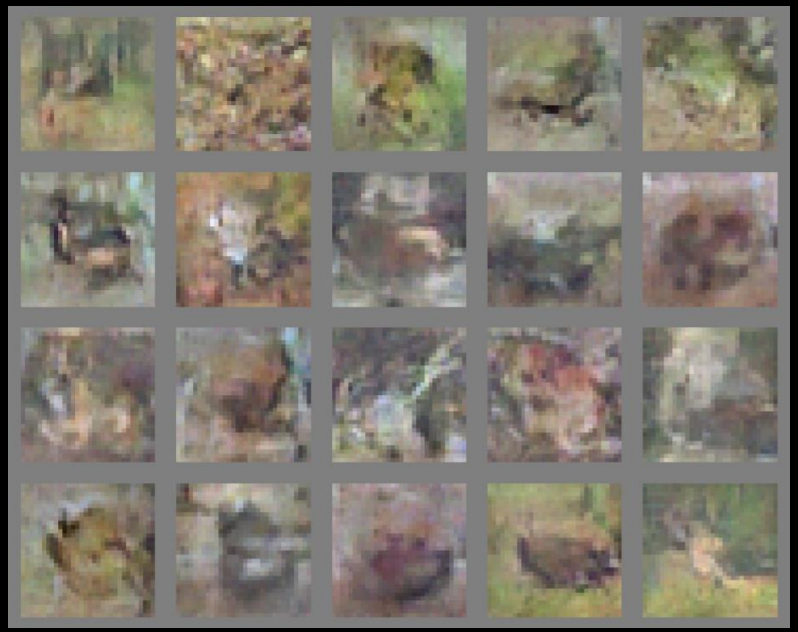
# In 2019, with StyleGAN ……

Source of video:
https://www.gwern.net/Faces


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

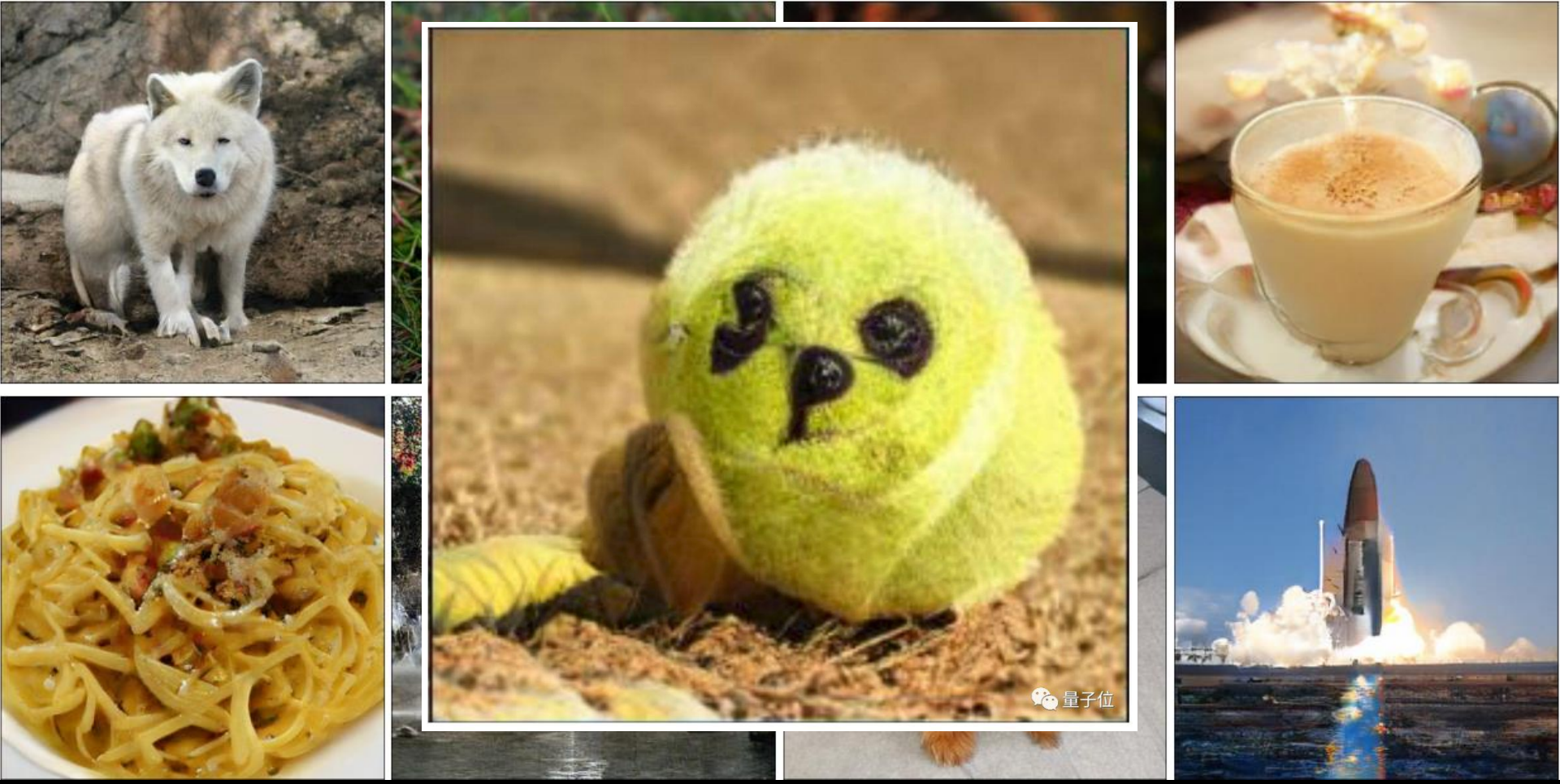
# Progressive GAN

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



# The first GAN

[Ian J. Goodfellow, et al., NIPS, 2014]




# Today ……

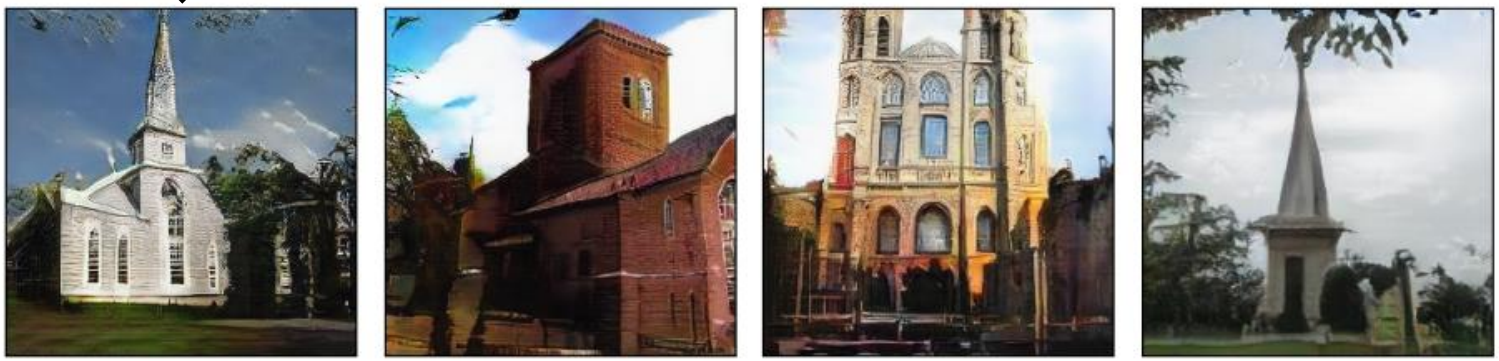
[Andrew Brock, et al., arXiv, 2018]

Does the generator have the concept of objects?

Some neurons correspond to specific objects, for example, tree

[David Bau, et al., ICLR 2019]

Remove the neurons for tree

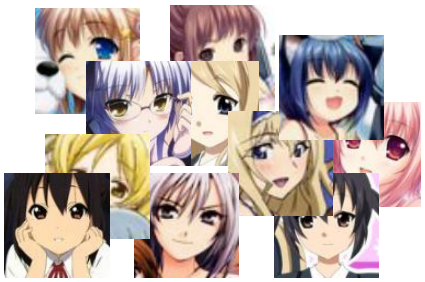


Activate the neurons for tree

[David Bau, et al., ICLR 2019]

# Three Categories of GAN

## 1. Generation



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

random vector

→ Generator →

image

## 2. Conditional Generation

blue eyes,
red hair,
short hair

*paired data*

"Girl with red hair"

text

→ Generator →

image

## 3. Unsupervised Conditional Generation

domain x

domain y

*unpaired data*

x

Photo

→ Generator →

y

Vincent van Gogh's style

# Text-to-Image

a dog is running

a bird is flying

- **Traditional supervised approach**

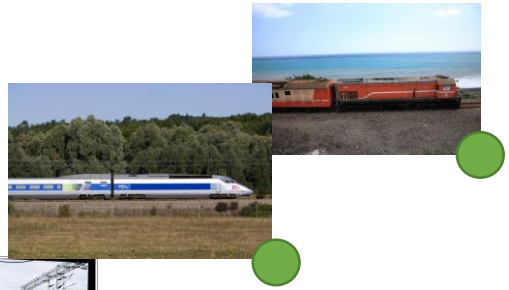
$c^1$: a dog is running → NN → Image ↔ 

as close as possible

Text: “train”

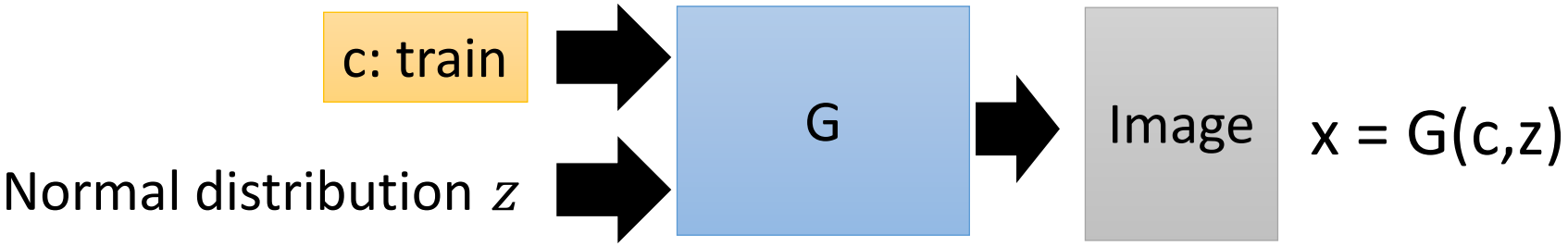
Target of NN output

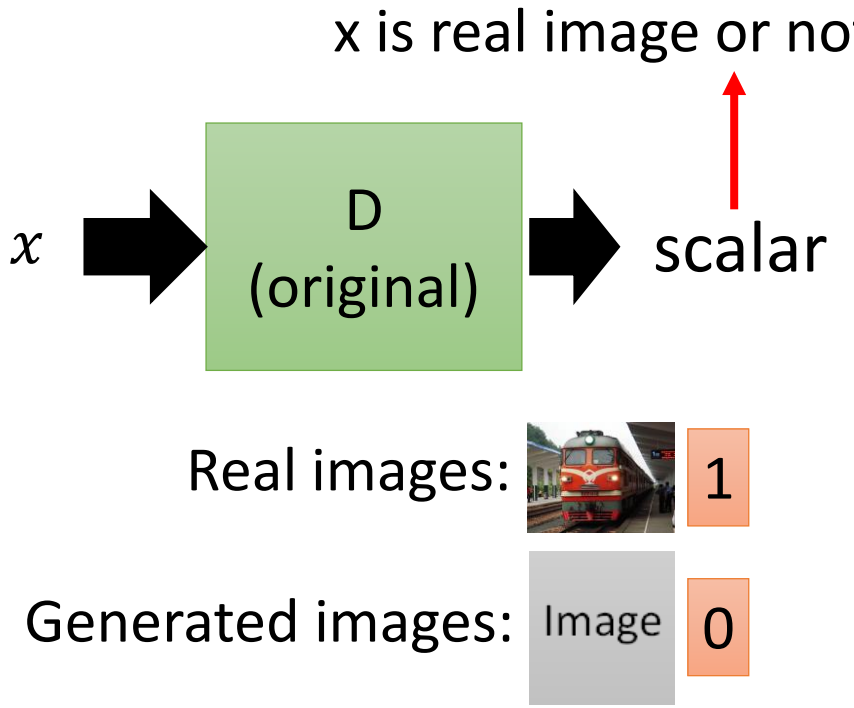
A blurry image!



[Scott Reed, et al, ICML, 2016]

# Conditional GAN



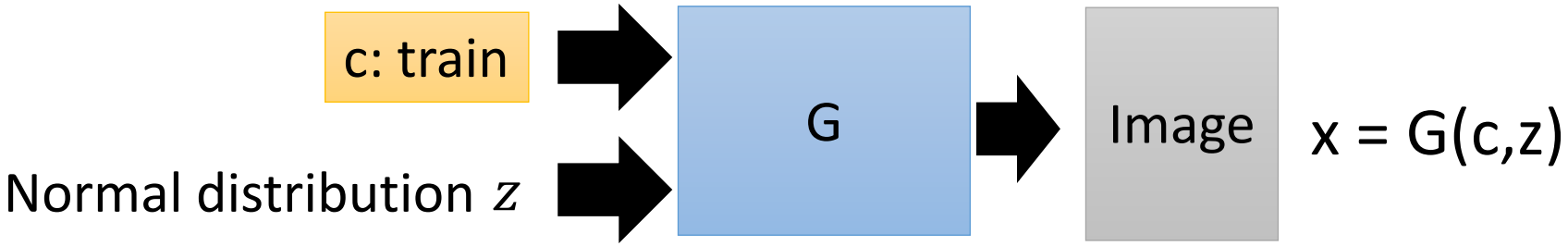



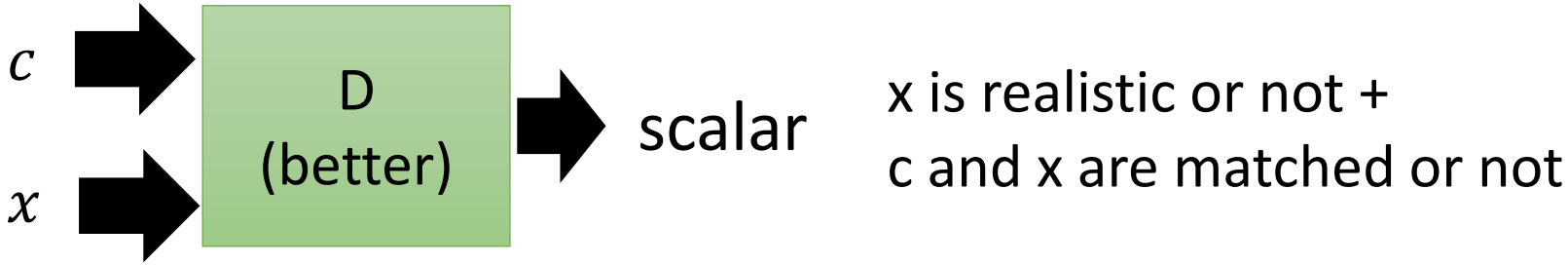

Generator will learn to generate realistic images ….

But completely ignore the input conditions.

[Scott Reed, et al, ICML, 2016]

# Conditional GAN



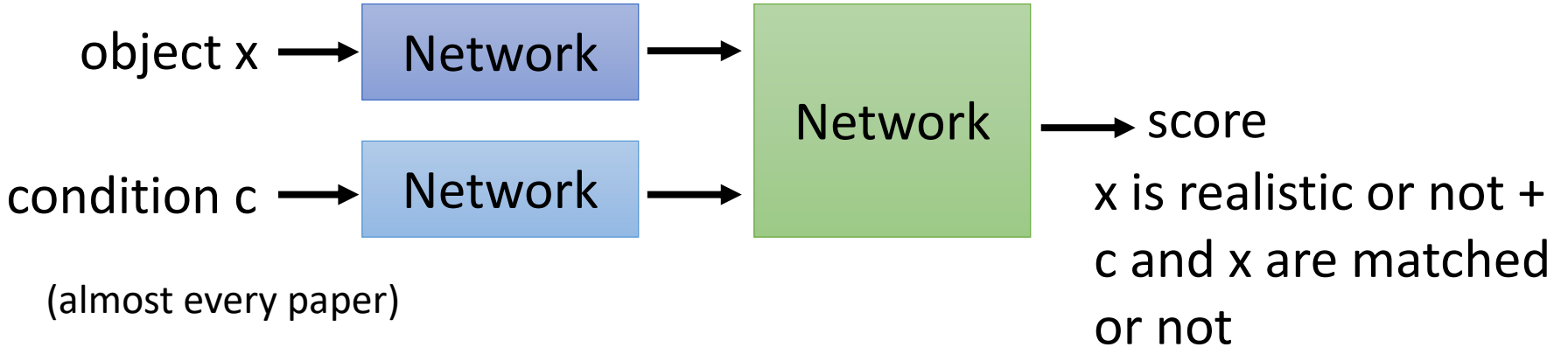




True text-image pairs: (train ,  ) 1

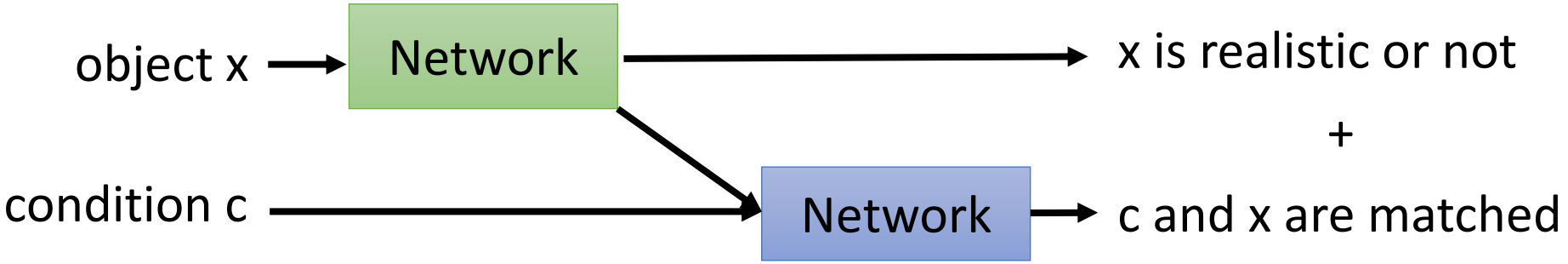
(cat ,  ) 0

(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

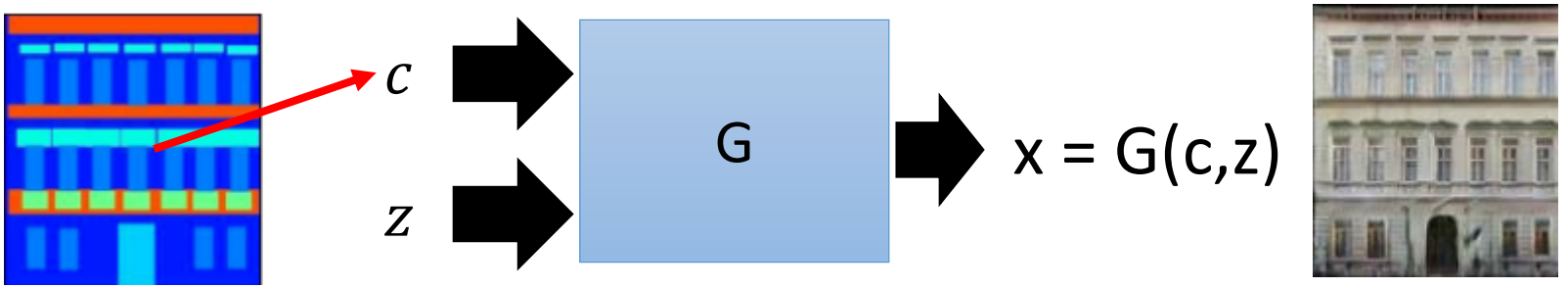
Collecting anime faces and the description of its characteristics

red hair, green eyes

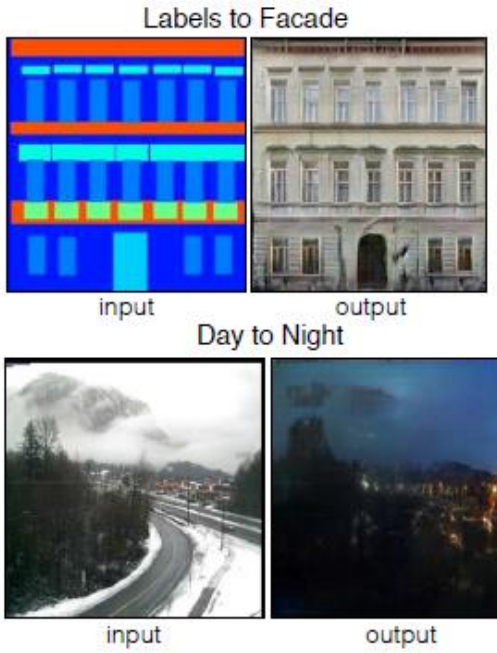
blue hair, red eyes

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

# Conditional GAN - Image-to-image







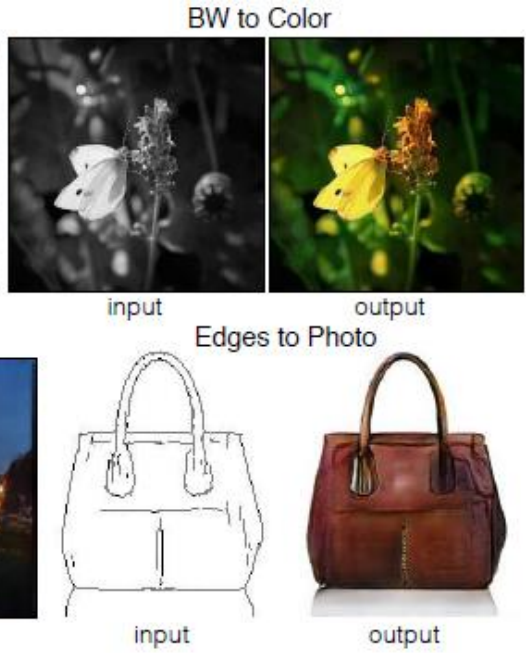
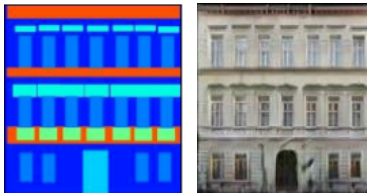



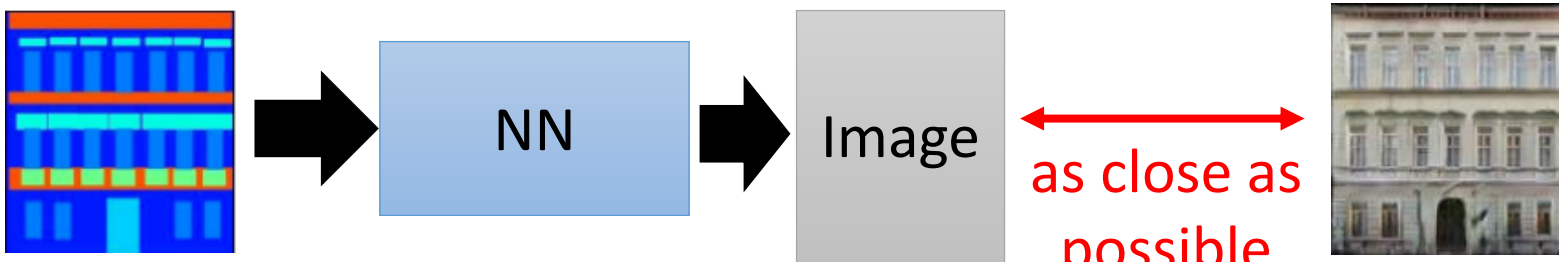

**Image translation, or pix2pix**

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

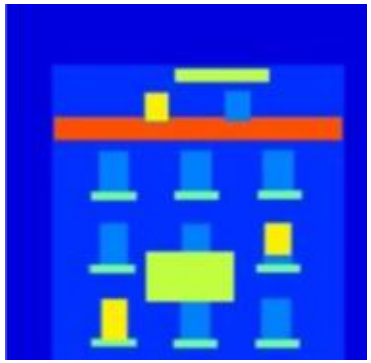
# Conditional GAN - Image-to-image



- **Traditional supervised approach**




Testing:



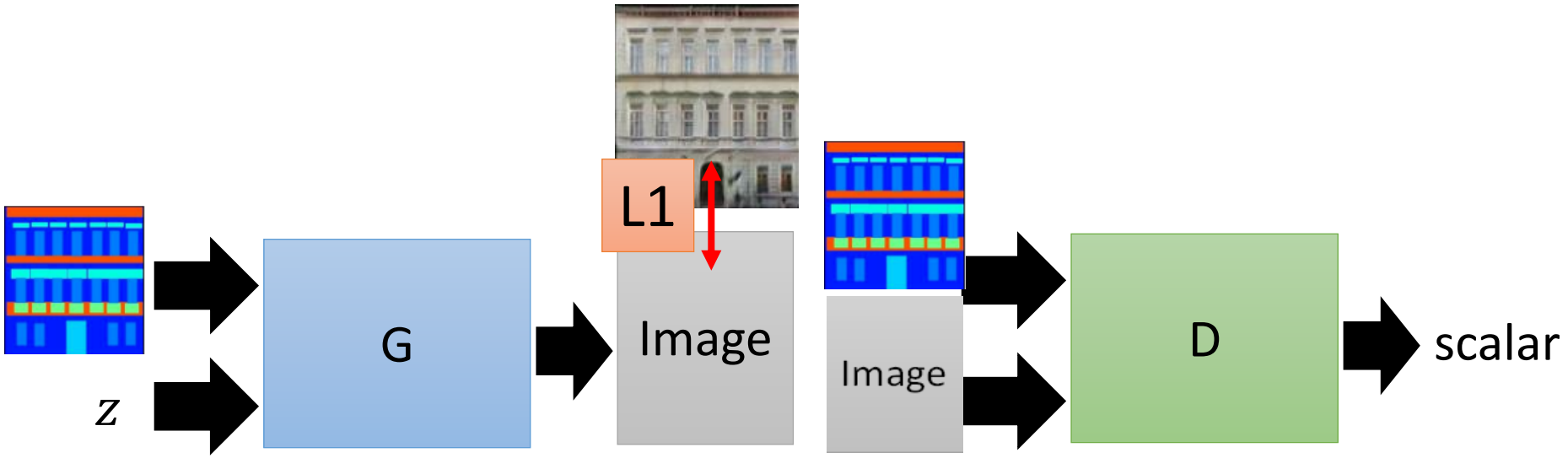
input

L1

It is blurry.

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

# Conditional GAN - Image-to-image

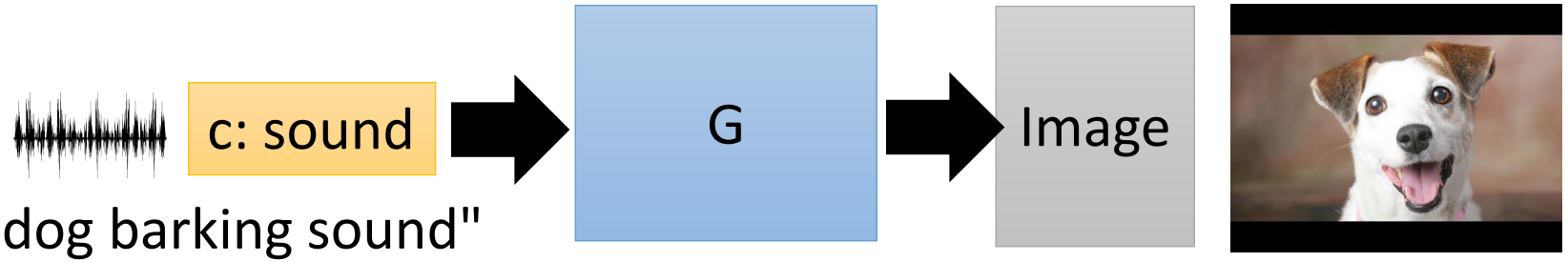



Testing:

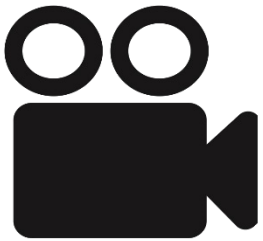
input L1 GAN GAN + L1

# Conditional GAN - Sound-to-image

[Wan, et al., ICASSP 2019]




***Training Data Collection***



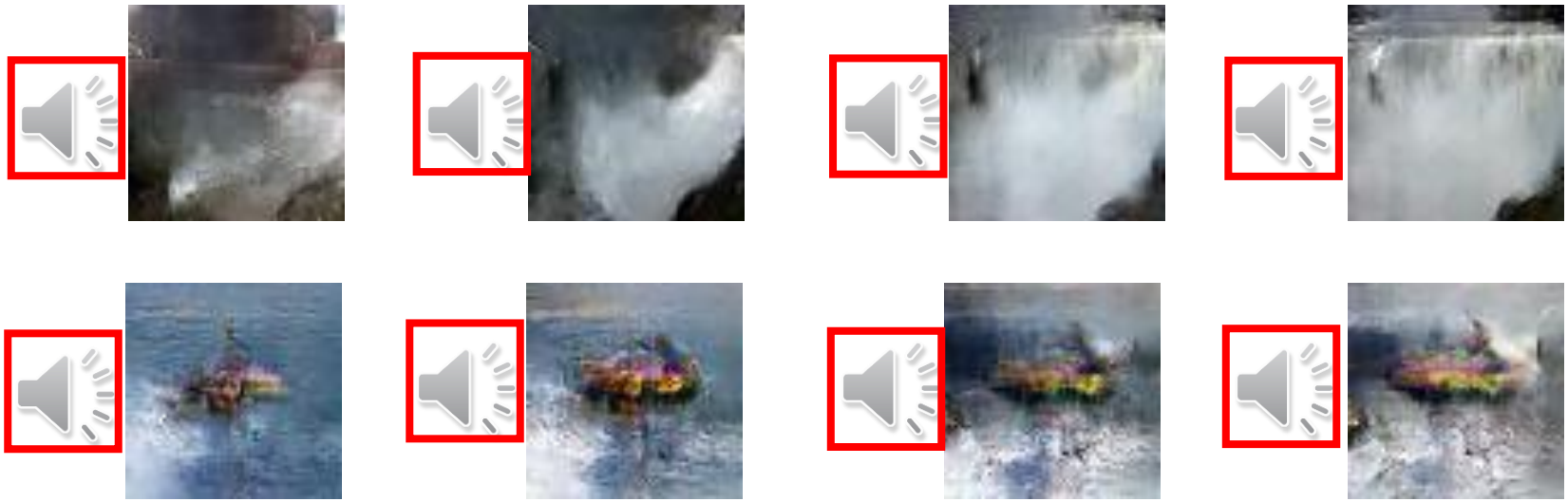
video

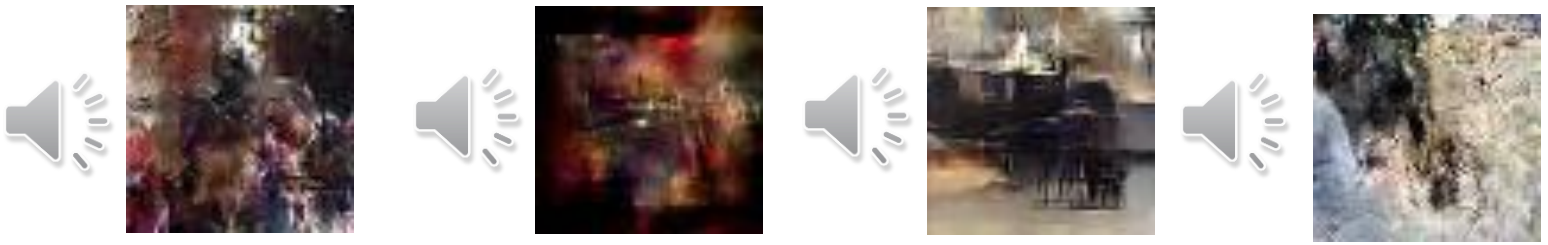
# Conditional GAN - Sound-to-image

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





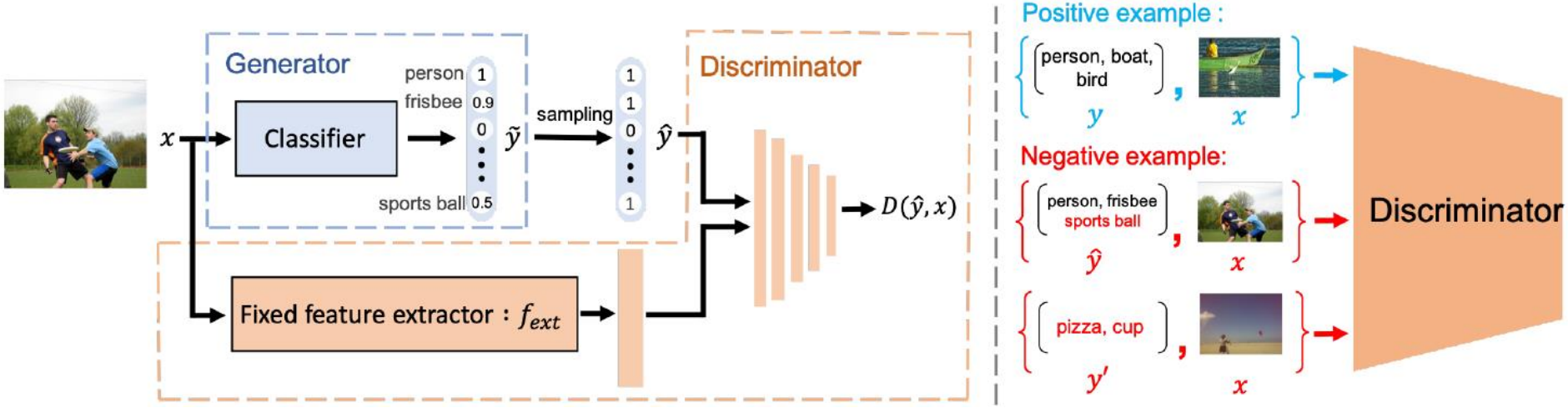
# Conditional GAN - Image-to-label

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., 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.

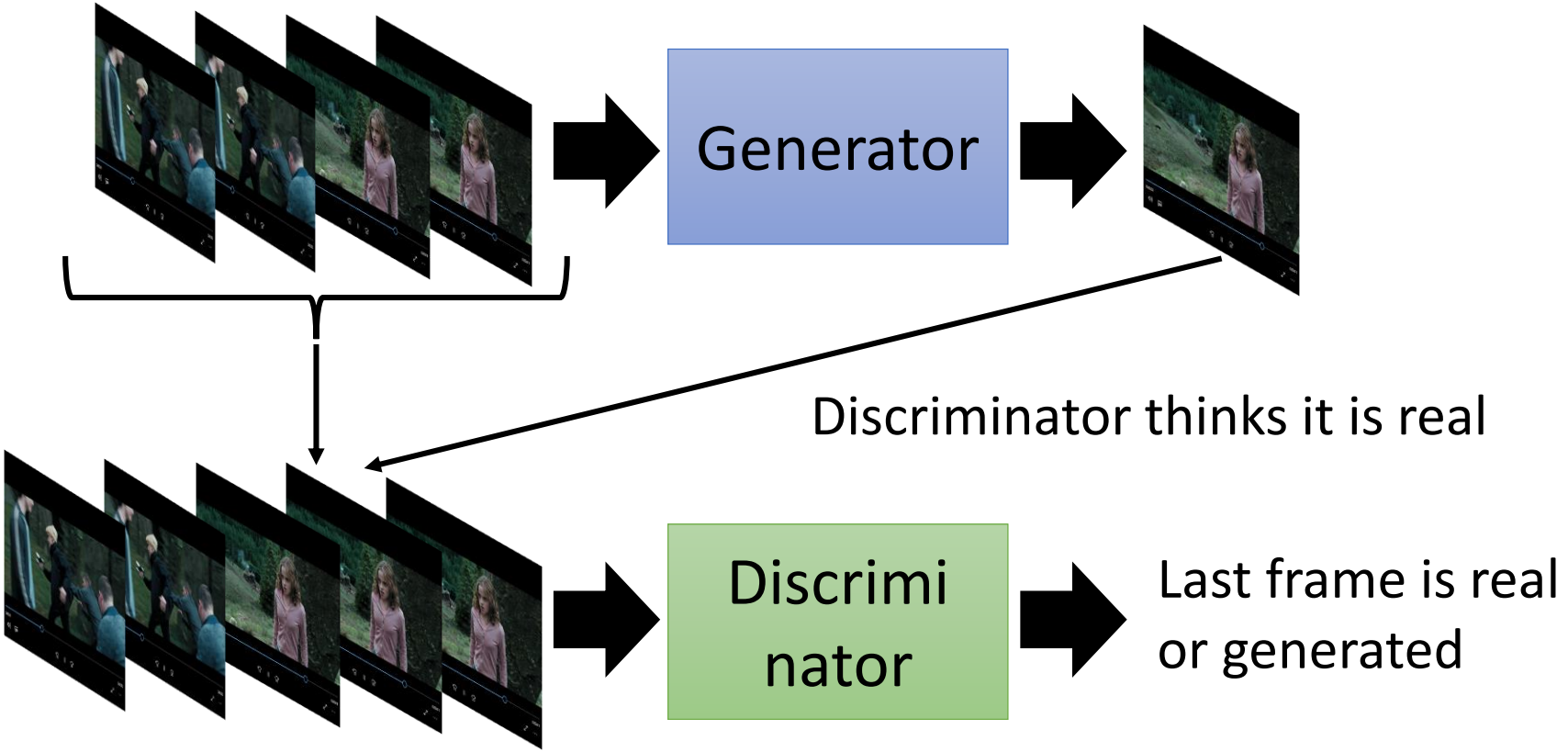
The classifiers are trained as conditional GAN.

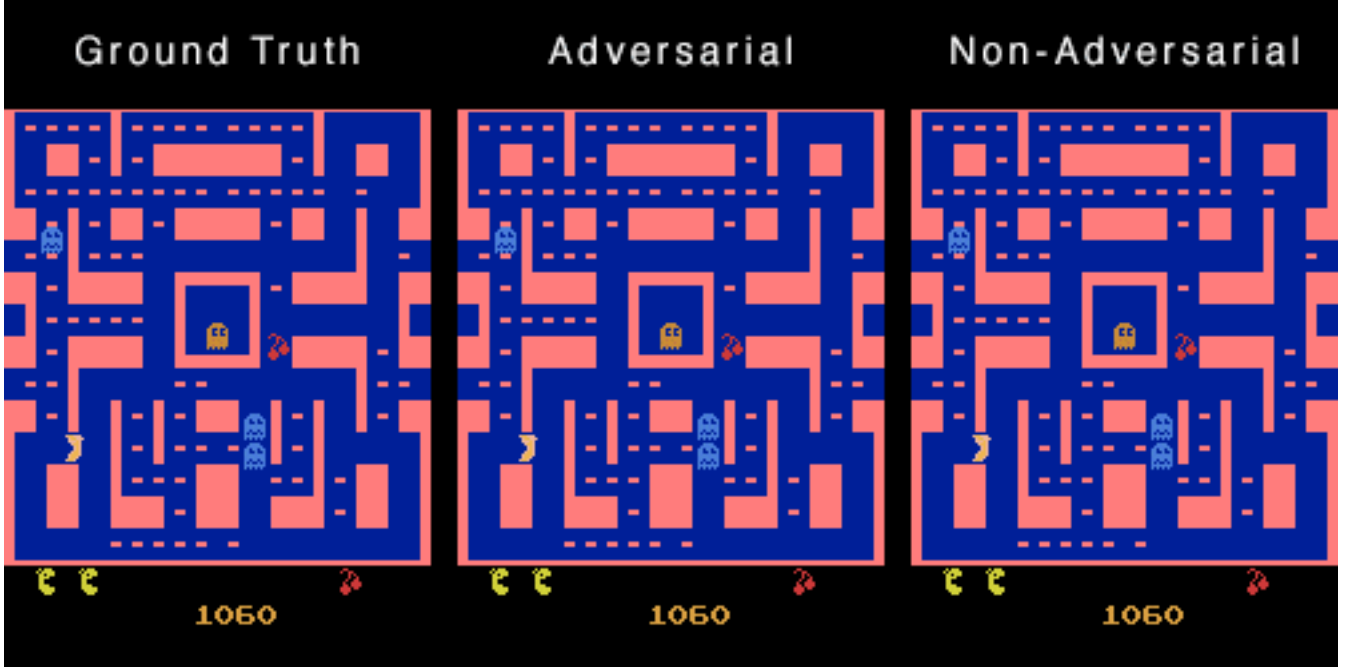
Conditional GAN outperforms other models designed for multi-label.

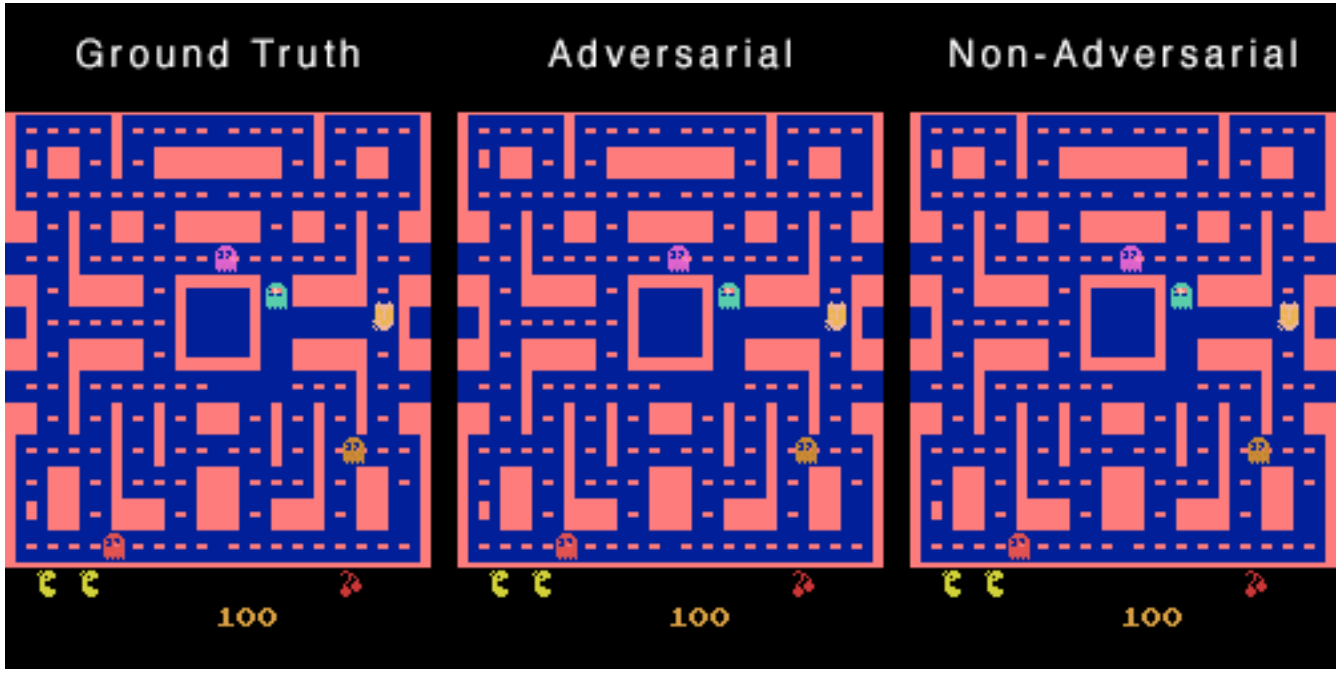
| F1 | MS-COCO | NUS-WIDE |
|---|---|---|
| VGG-16 | 56.0 | 33.9 |
| + GAN | 60.4 | 41.2 |
| Inception | 62.4 | 53.5 |
| +GAN | 63.8 | 55.8 |
| Resnet-101 | 62.8 | 53.1 |
| +GAN | 64.0 | 55.4 |
| Resnet-152 | 63.3 | 52.1 |
| +GAN | 63.9 | 54.1 |
| Att-RNN | 62.1 | 54.7 |
| RLSD | 62.0 | 46.9 |

# Conditional GAN - Video Generation

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






https://github.com/dyelax/Adversarial_Video_Generation

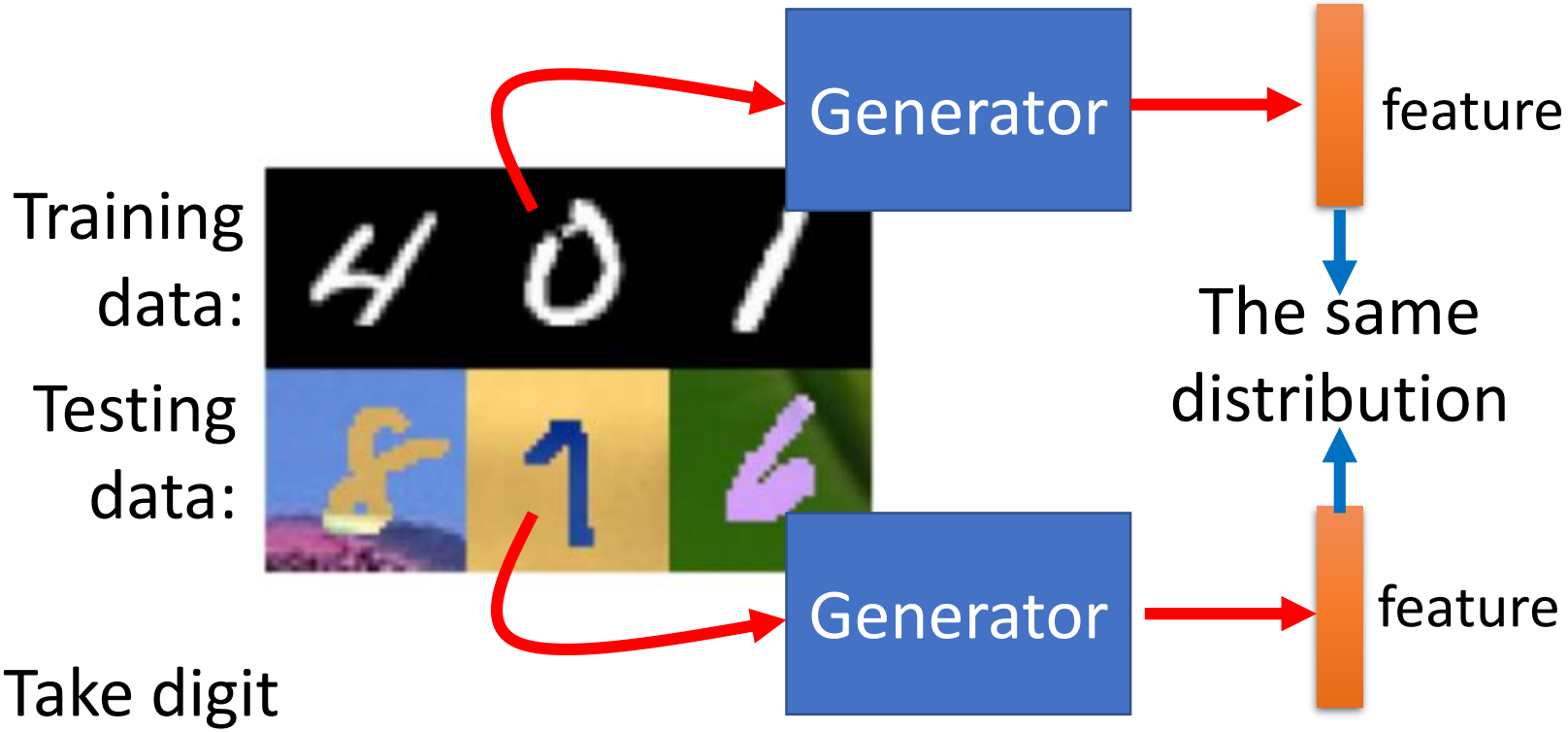
https://arxiv.org/abs/1905.08233

# More about Video Generation

[Egor Zakharov, et al., arXiv, 2019]

# Domain Adversarial Training

- Training and testing data are in different domains




Take digit classification as example

# Domain Adversarial Training

feature extractor (Generator)

Always output zero vectors

input $x$

image

features $f$

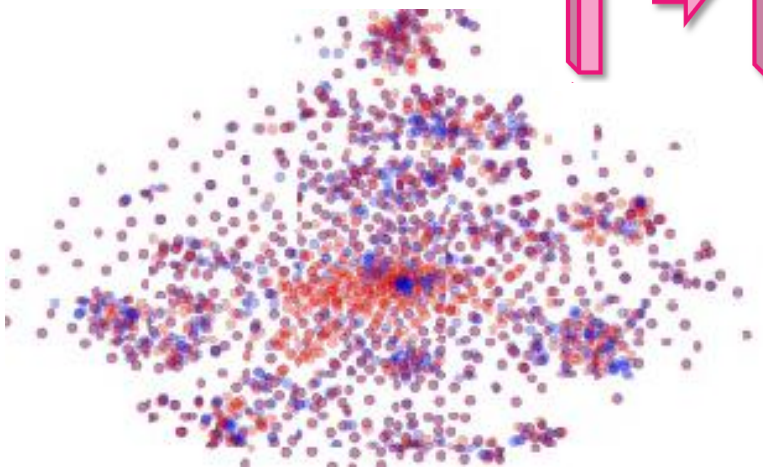
Domain Classifier Fails

Which domain?

domain label $d$

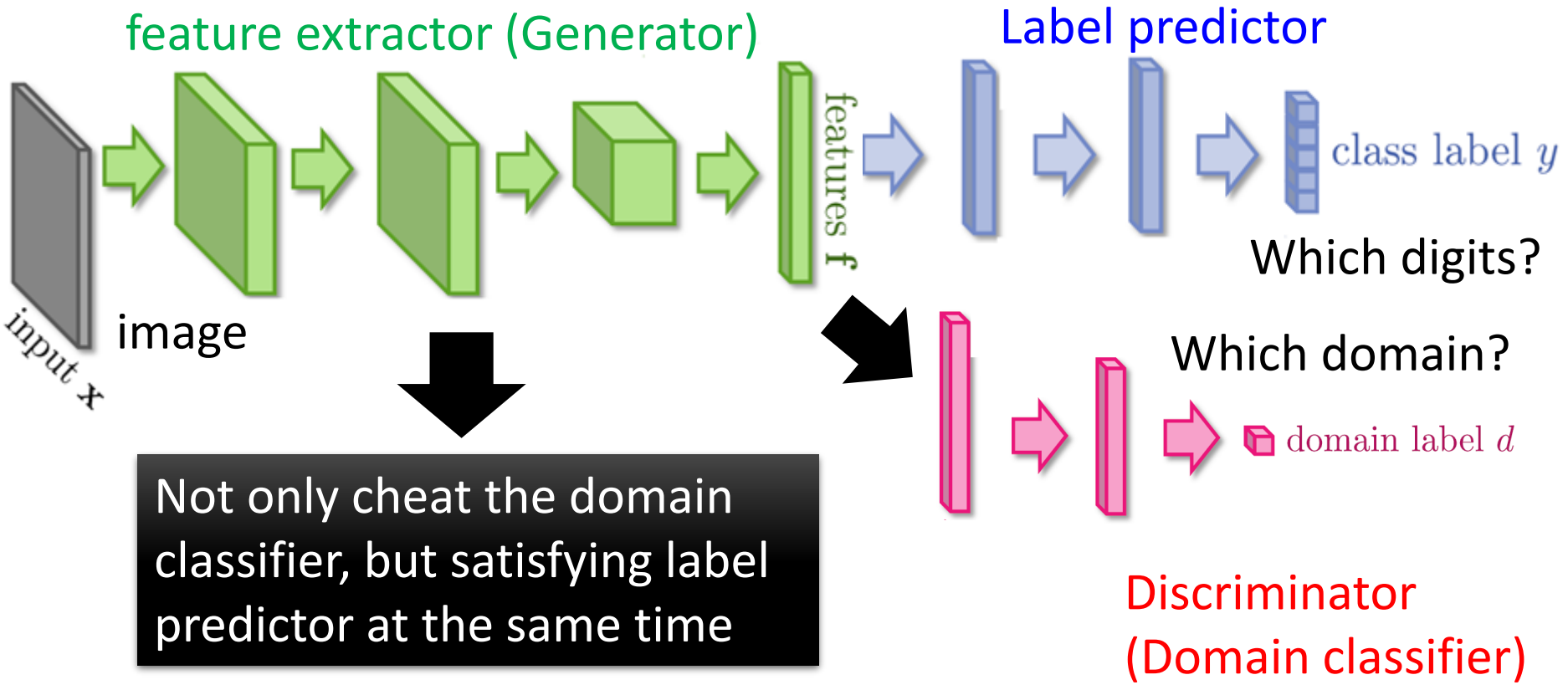
blue points

red points



Discriminator
(Domain classifier)

# Domain Adversarial Training



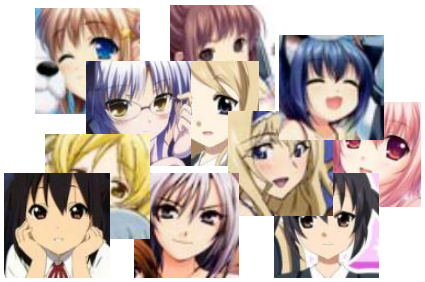

Successfully applied on image classification

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

More speech-related applications in Part III.

# Three Categories of GAN

## 1. Generation



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

random vector

Generator

image

## 2. Conditional Generation

blue eyes,
red hair,
short hair

*paired data*

"Girl with red hair"

text

Generator

image

## 3. Unsupervised Conditional Generation

domain x

domain y

*unpaired data*

x

Photo

Generator

y

Vincent van Gogh's style

# Unsupervised Conditional Generation

Condition
Object in Domain X


Generated Object
Object in Domain Y

Transform an object from one domain to another

***without paired data***

Domain X

photos

Not Paired

Domain Y

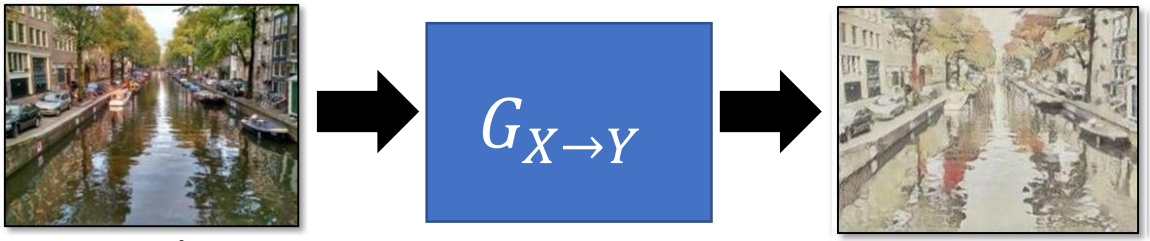
Vincent van Gogh's paintings

Use image style transfer as example here

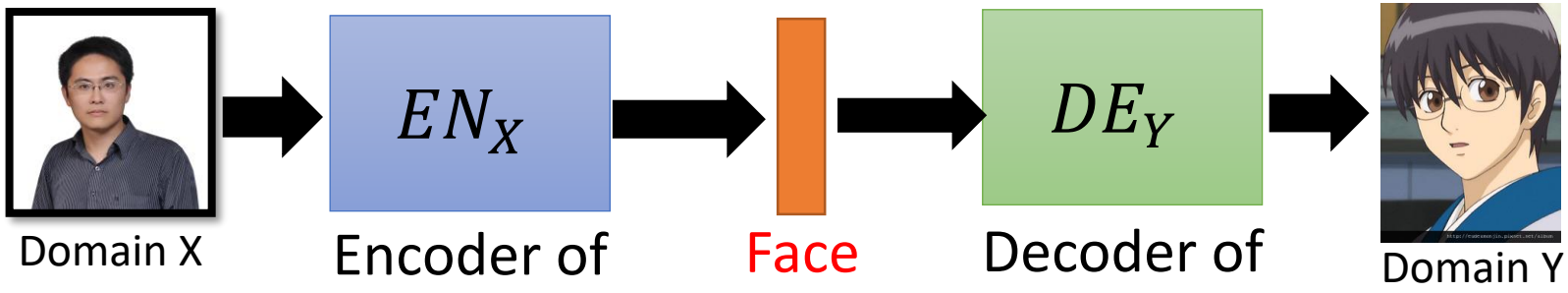
**More Applications in Parts III and IV**

# Unsupervised Conditional Generation

- Approach 1: Cycle-GAN and its variants




- Approach 2: Shared latent space



# Cycle GAN

Domain X

Domain Y

Domain X

$G_{X \to Y}$

Become similar to domain Y

$D_Y$

scalar

Input image belongs to domain Y or not

Domain Y

# Cycle GAN

Domain X

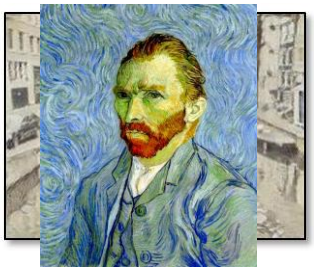
Domain Y

Domain X

ignore input

$G_{X \to Y}$

Become similar to domain Y



Not what we want!

$D_Y$

scalar

Input image belongs to domain Y or not

Domain Y

# Cycle GAN

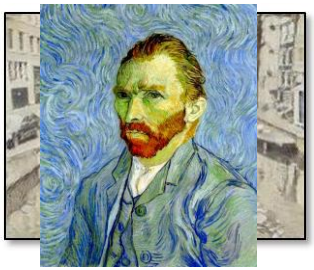
Domain X

Domain Y

Domain X

$G_{X \to Y}$

Become similar to domain Y



Not what we want!

$D_Y$ → scalar

ignore input

The issue can be avoided by network design.

Simpler generator makes the input and output more closely related.

Input image belongs to domain Y or not

[Tomer Galanti, et al. ICLR, 2018]

# Cycle GAN



Domain X

$G_{X \to Y}$

Become similar to domain Y

$D_Y$ → scalar

Input image belongs to domain Y or not

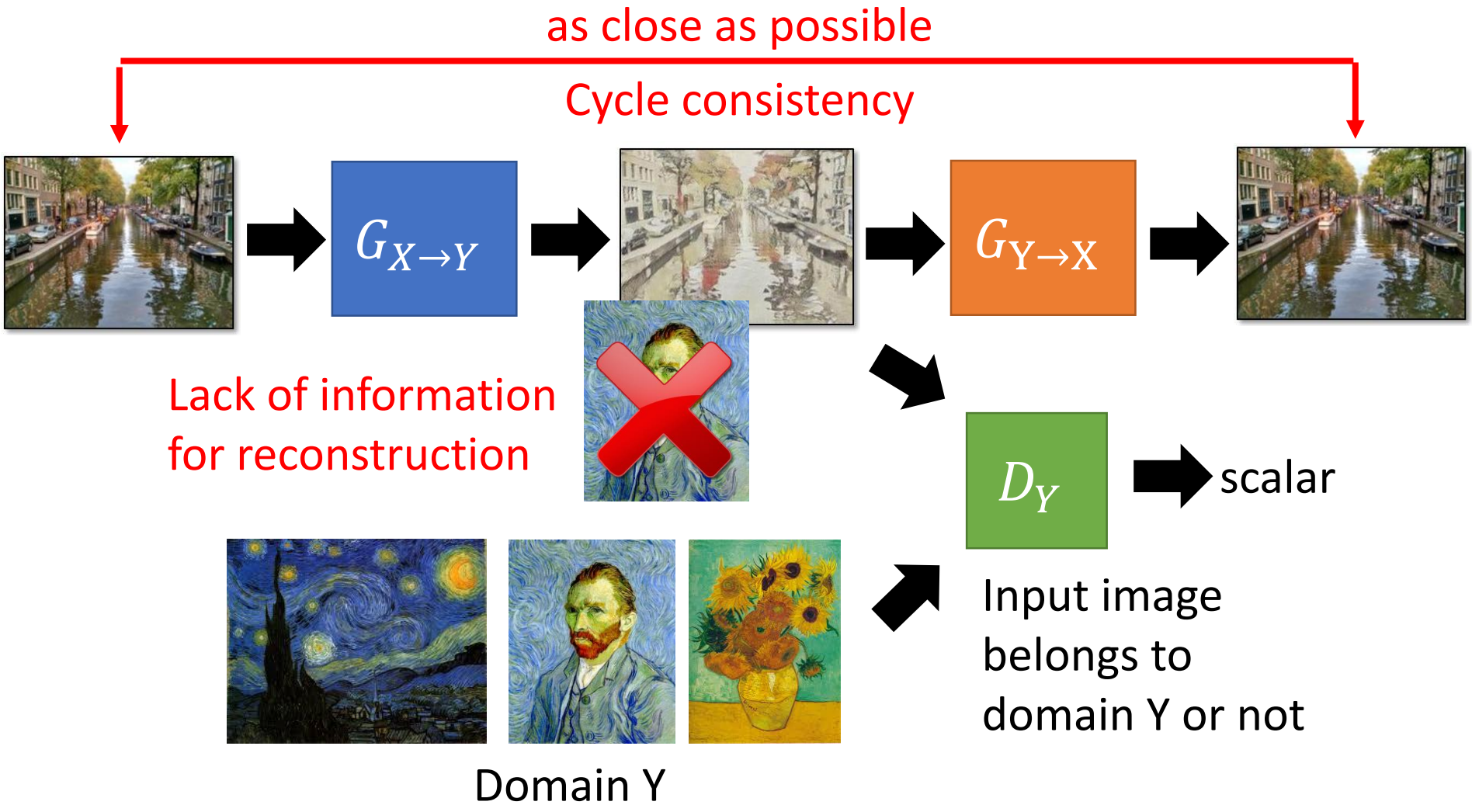
Encoder Network ← pre-trained → Encoder Network

as close as possible

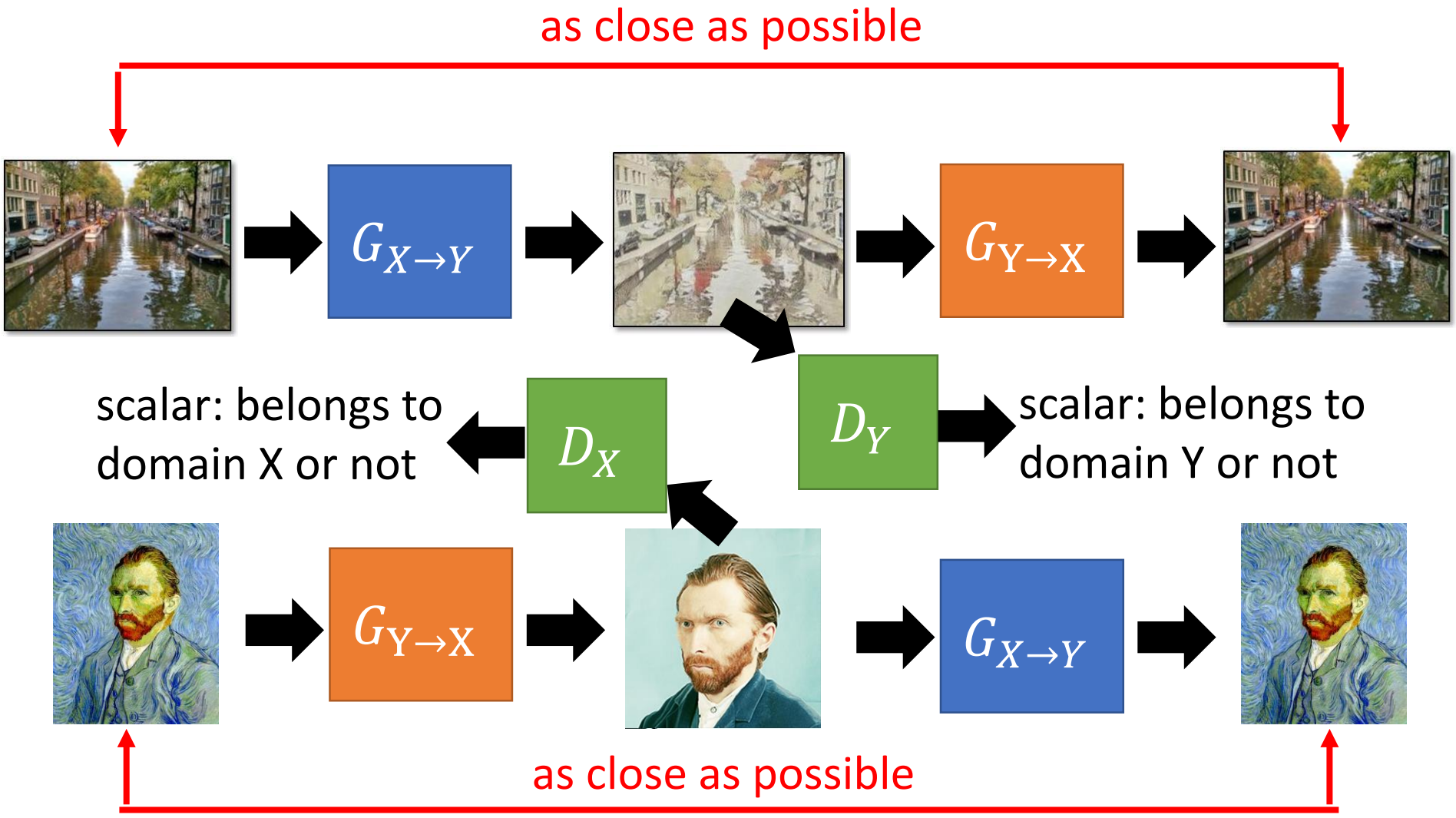
Baseline of DTN [Yaniv Taigman, et al., ICLR, 2017]

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

# Cycle GAN



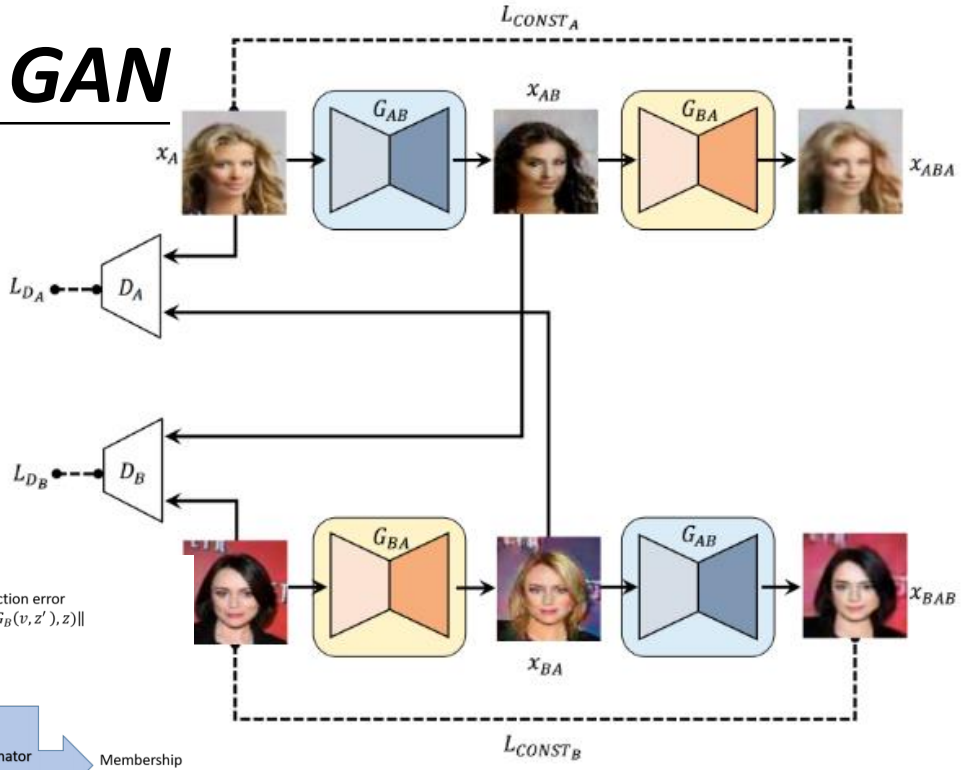
# Cycle GAN


as close as possible
$G_{X\to Y}$
$G_{Y\to X}$
$D_Y$
$D_X$
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

For multiple domains, considering starGAN

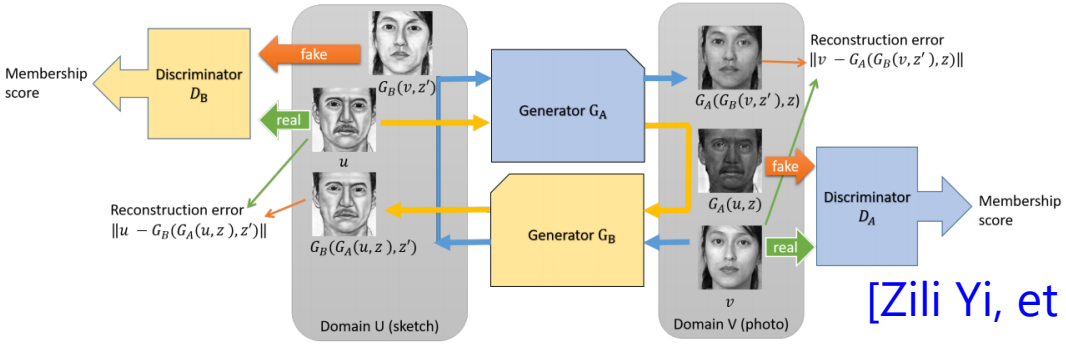
[Yunjey Choi, arXiv, 2017]

## Disco GAN



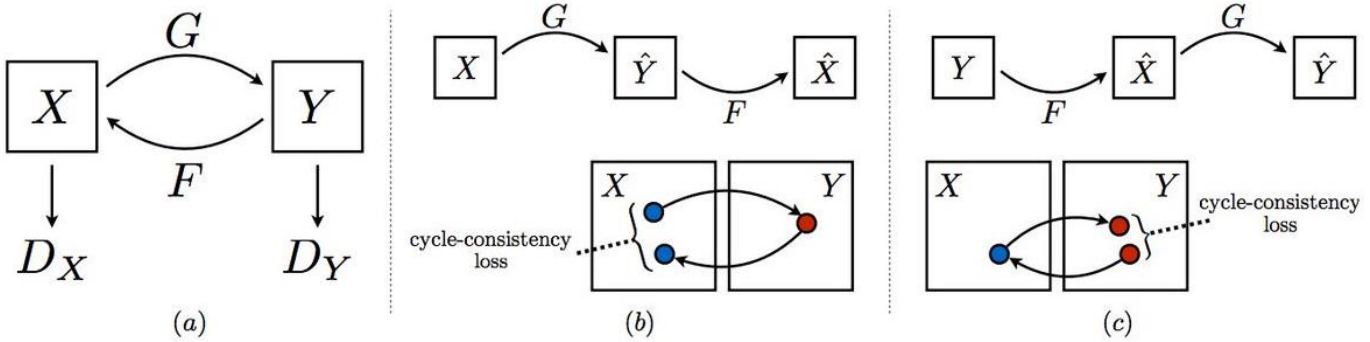

[Taeksoo Kim, et al., ICML, 2017]

## Dual GAN




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

## Cycle GAN

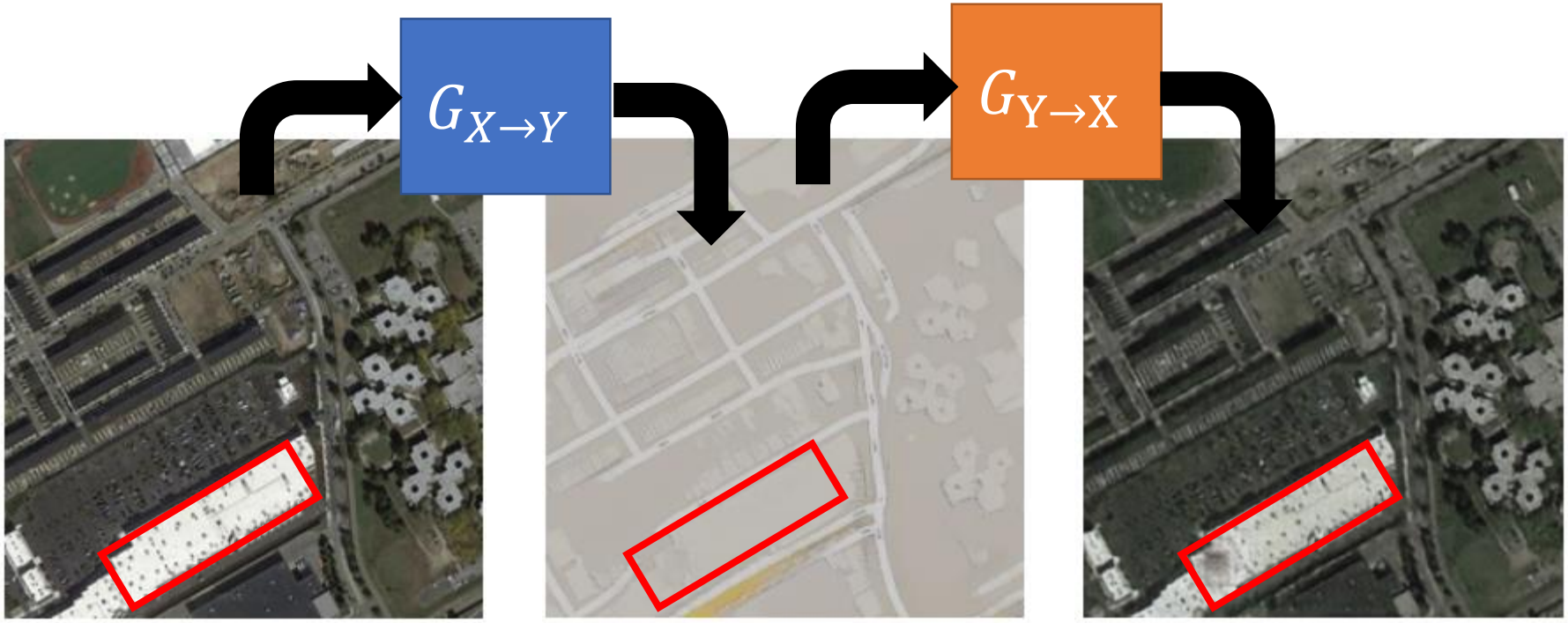



[Jun-Yan Zhu, et al., ICCV, 2017]

# Issue of Cycle Consistency

- **CycleGAN: a Master of Steganography**

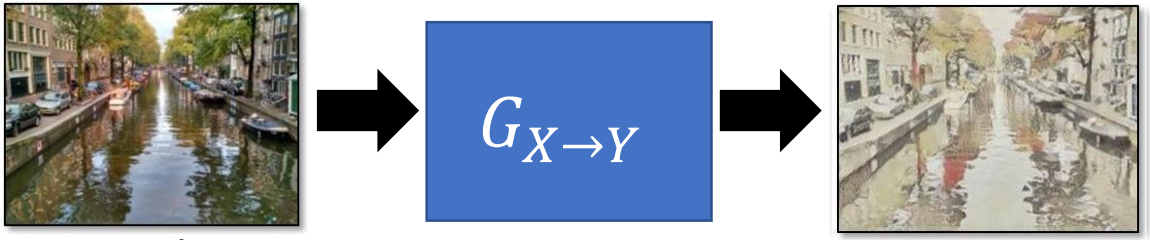
[Casey Chu, et al., NIPS workshop, 2017]



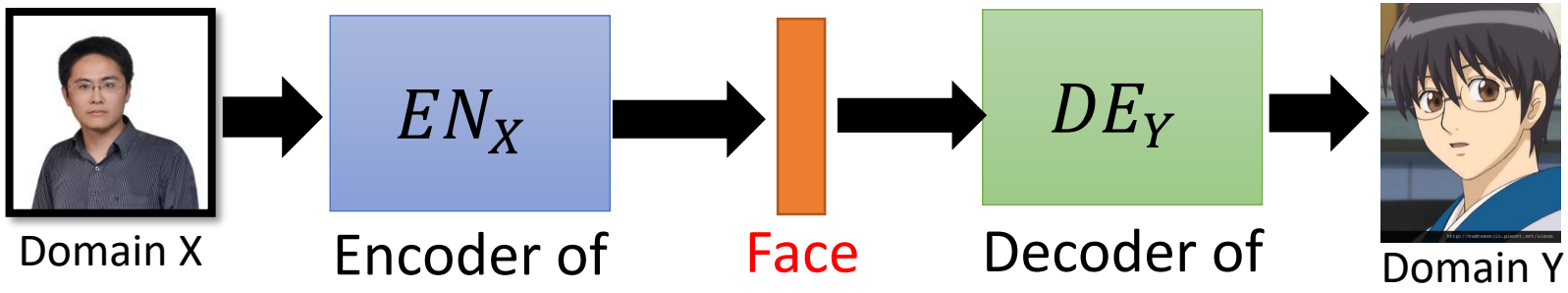

The information is hidden.

# Unsupervised Conditional Generation

- Approach 1: Cycle-GAN and its variants




- Approach 2: Shared latent space



# *Shared latent space*

## *Target*

- domain-x information

$EN_X$

$DE_X$

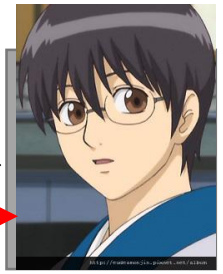
image

image

$EN_Y$

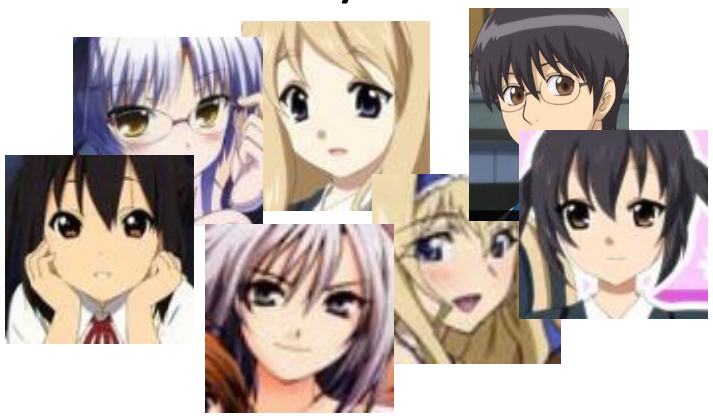
Face Attribute

$DE_Y$



+ domain-y information

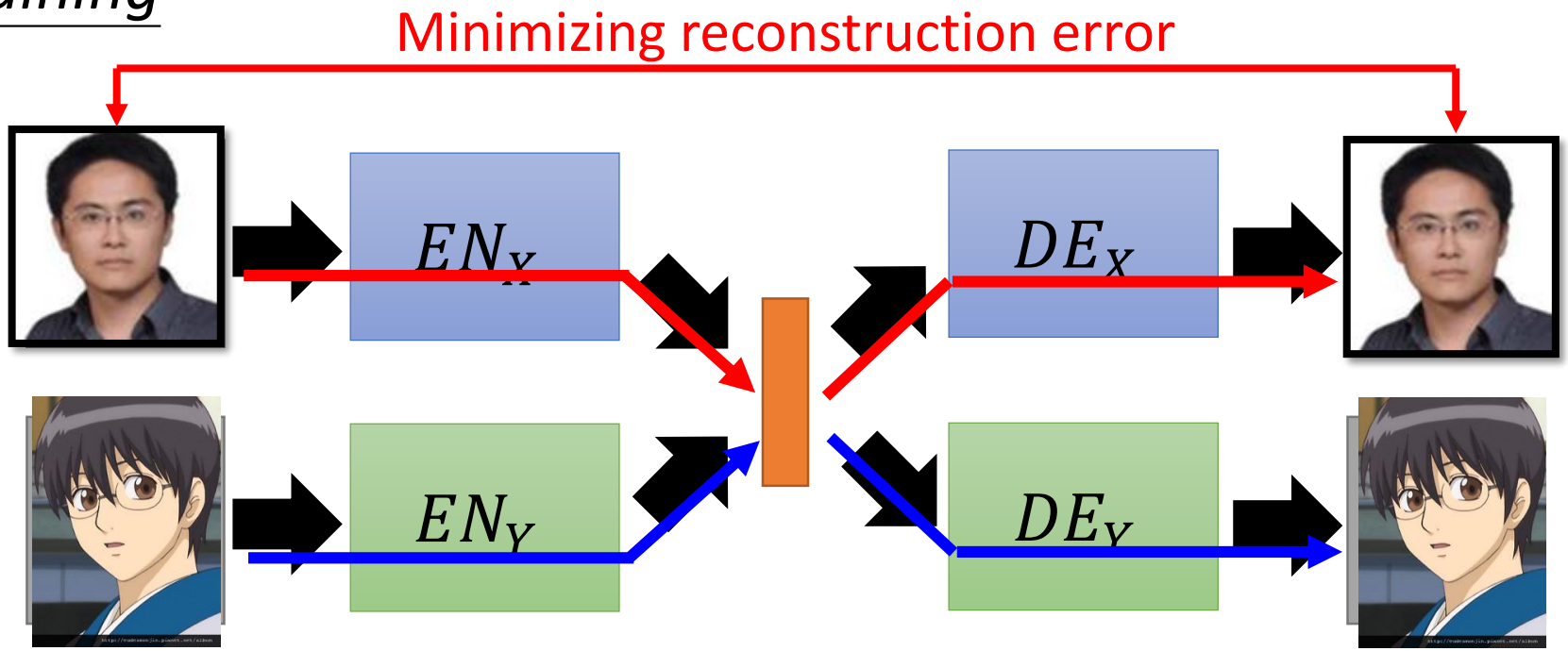
Domain X



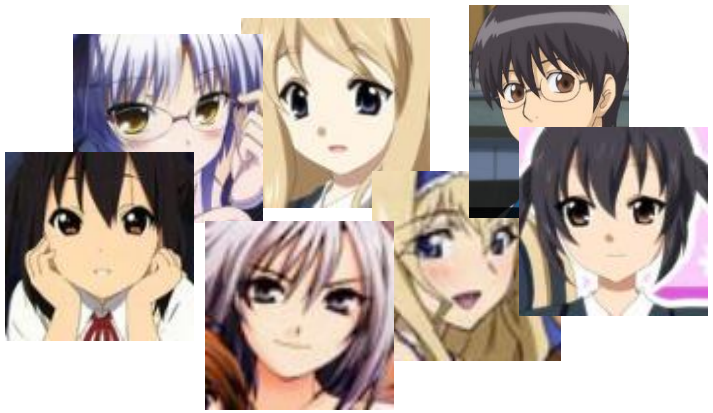
Domain Y

# *Shared latent space*

## *Training*



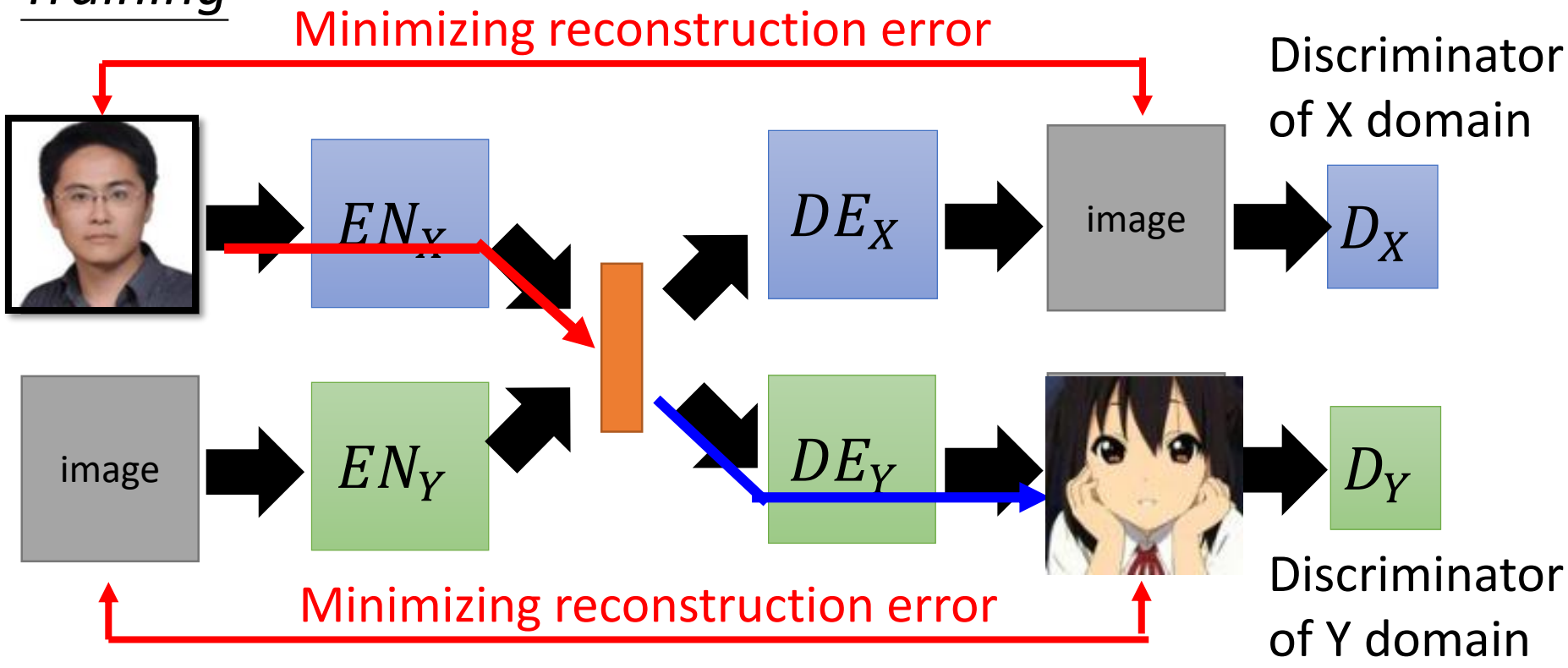

Domain X



Domain Y

# *Shared latent space*

## *Training*

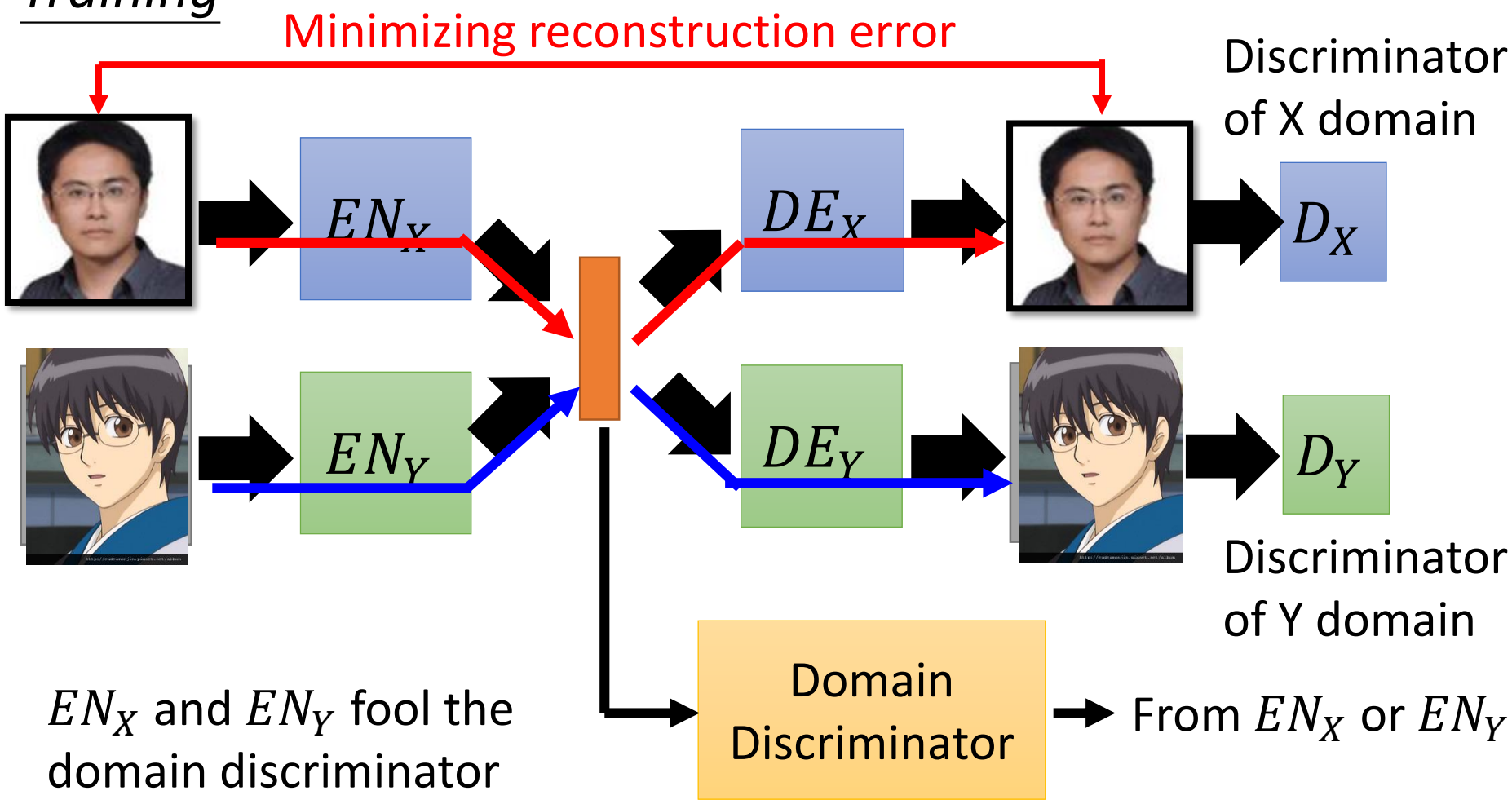



Because we train two auto-encoders separately …

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

# *Shared latent space*

## *Training*



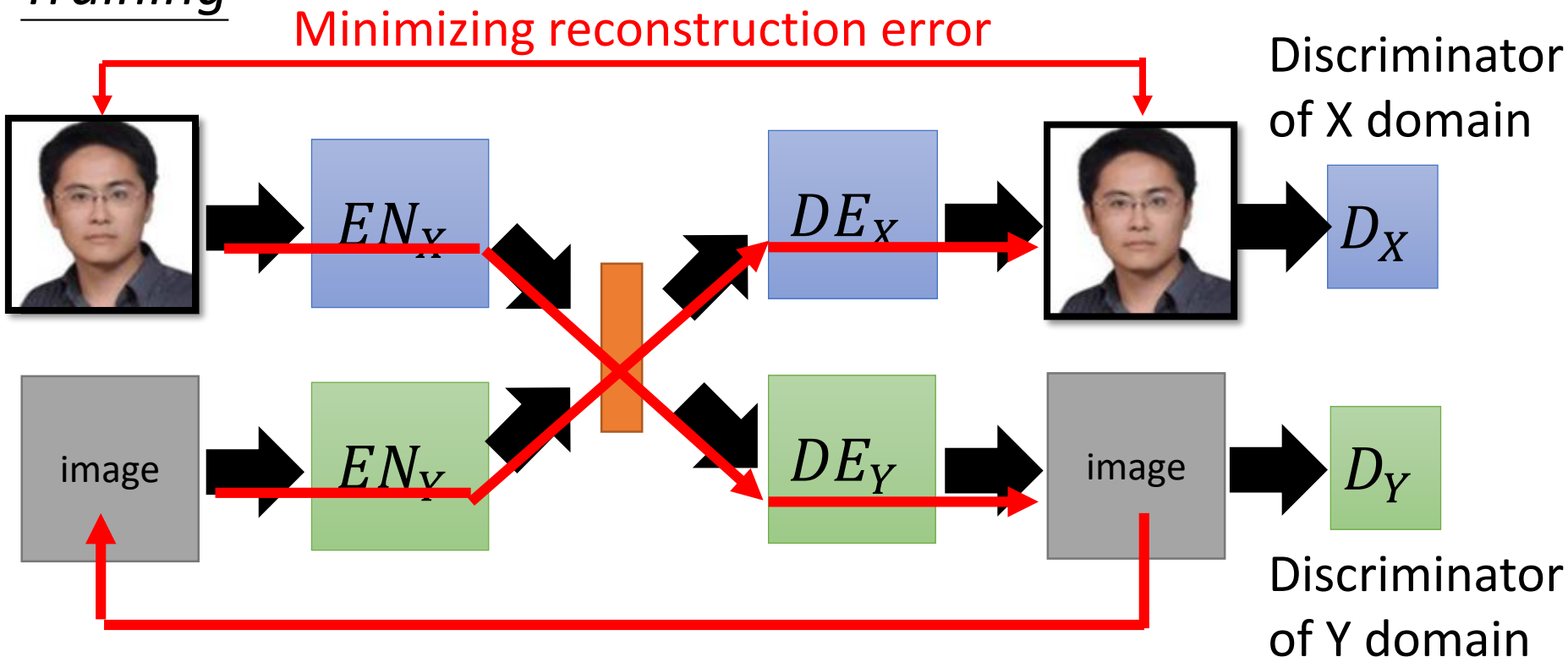

$EN_X$ and $EN_Y$ fool the domain discriminator

The domain discriminator forces the output of $EN_X$ and $EN_Y$ have the same distribution.

[Guillaume Lample, et al., NIPS, 2017]

# *Shared latent space*

*Training*

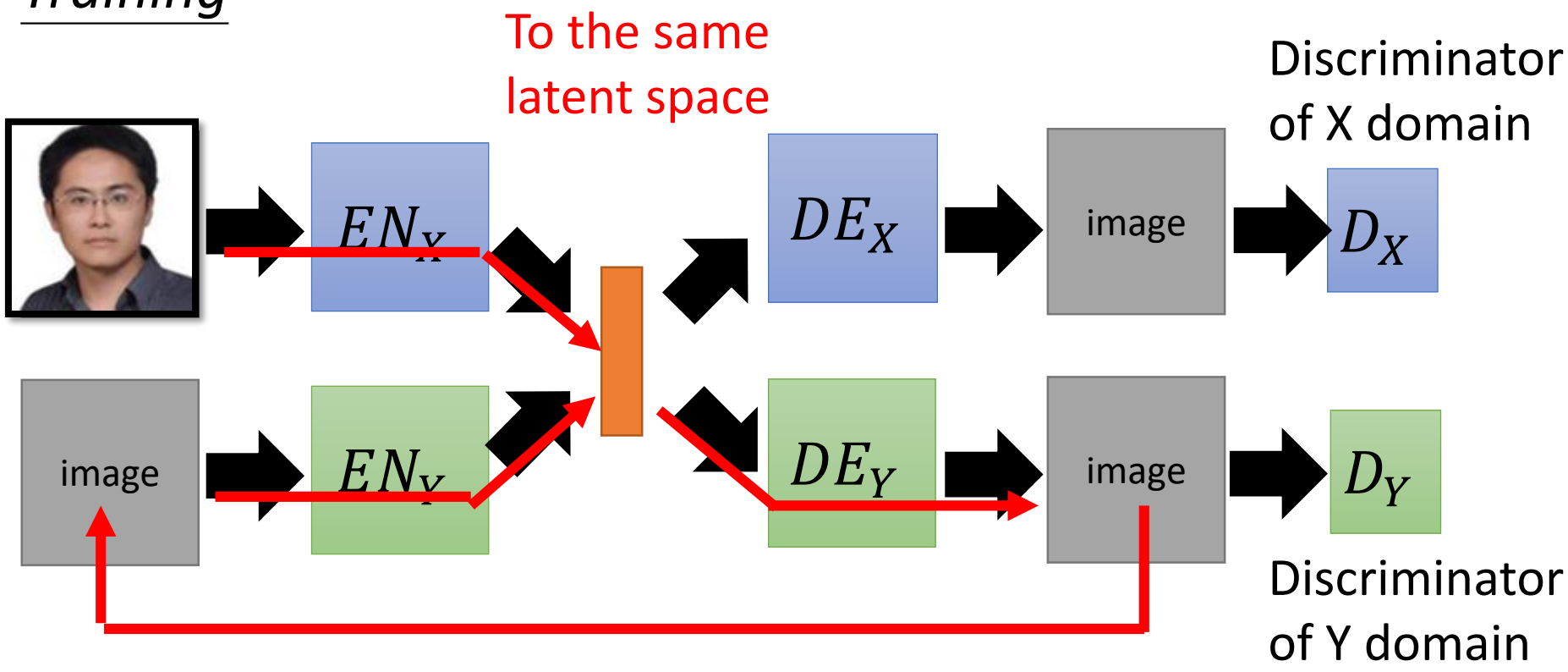



Cycle Consistency:

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

# *Shared latent space*

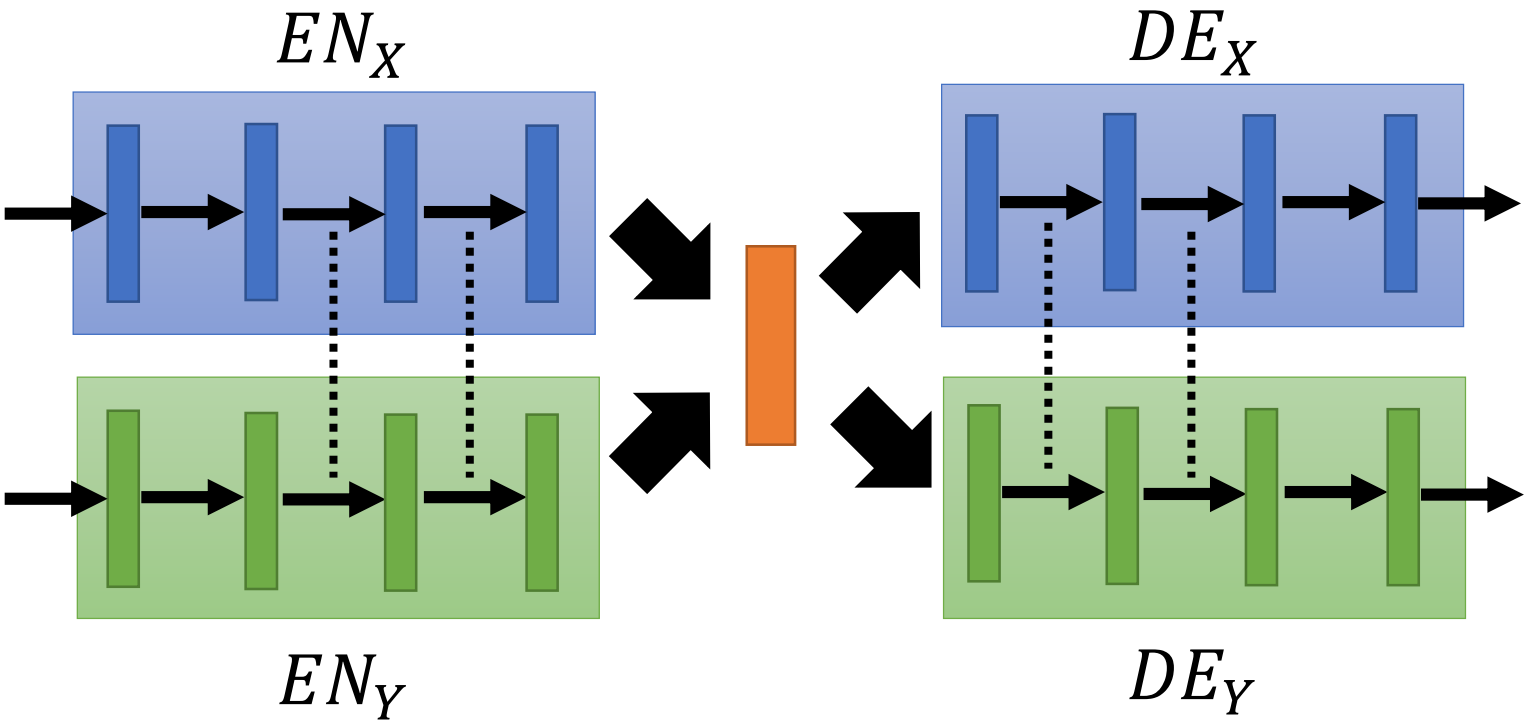
*Training*




Semantic Consistency:

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

# ***Shared latent space***



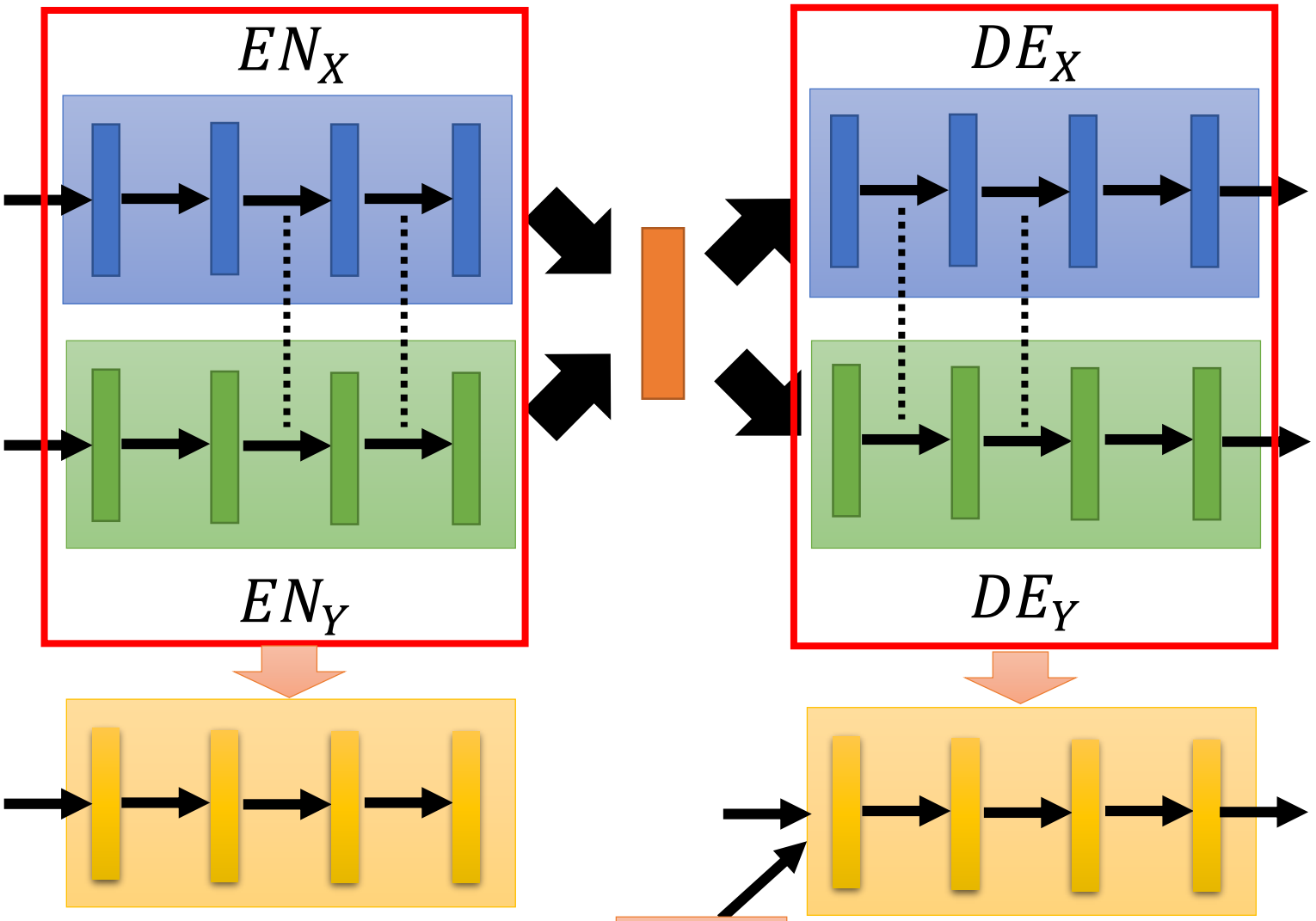

Sharing the parameters of encoders and decoders

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

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

# *Shared latent space*

**Widely used in Voice Conversion (Part III)**




One encoder to extract domain-independent information

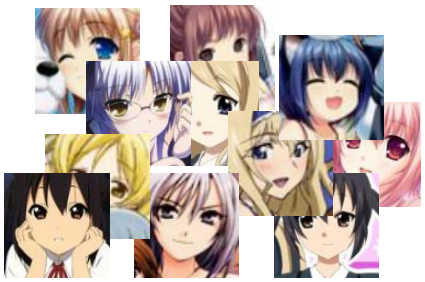
Input an extra indicator to control the decoder

# SELFIE2ANIME

https://selfie2anime.com/

# Three Categories of GAN

## 1. Typical GAN



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

random vector

→ Generator →

image

## 2. Conditional GAN

blue eyes,
red hair,
short hair

*paired data*

"Girl with red hair"

text

→ Generator →

image

## 3. Unsupervised Conditional GAN

domain x

domain y

*unpaired data*

x

Photo

→ Generator →

y

Vincent van Gogh's style

# Reference

- **Generation**
  - Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative Adversarial Nets, NIPS, 2014
  - 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
  - David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, Antonio Torralba, GAN Dissection: Visualizing and Understanding Generative Adversarial Networks, ICLR 2019

# 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
  - Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, Victor Lempitsky, Few-Shot Adversarial Learning of Realistic Neural Talking Head Models, arXiv 2019
  - Chia-Hung Wan, Shun-Po Chuang, Hung-Yi Lee, "Towards Audio to Scene Image Synthesis using Generative Adversarial Network", 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

# Part II: A little bit Theory

# Outline of Part II

Basic Theory of GAN

Helpful Tips

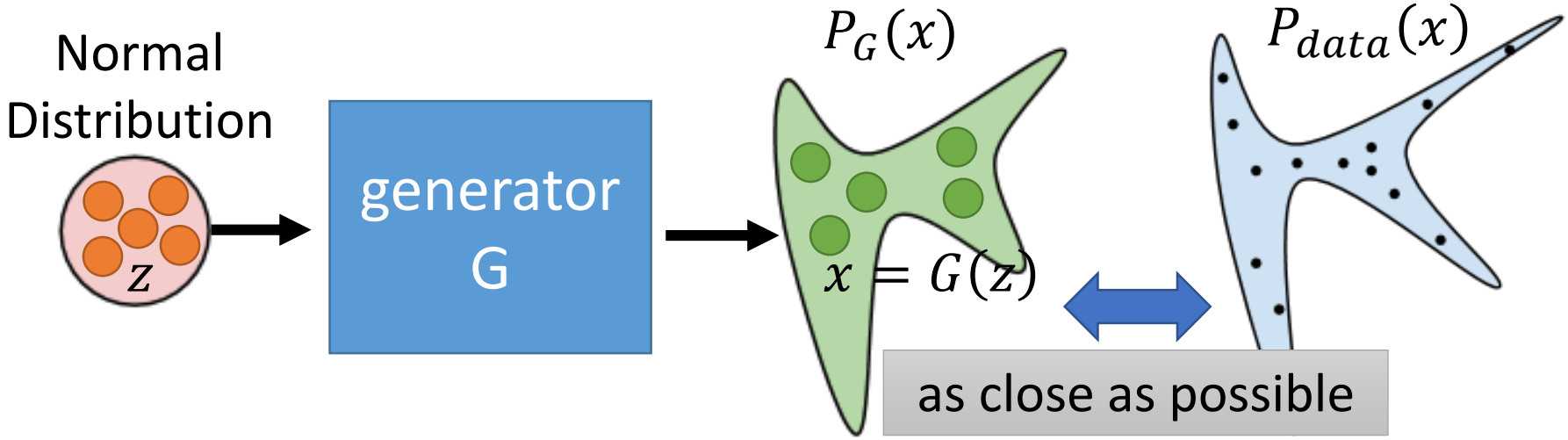
How to evaluate GAN

Relation to Reinforcement Learning

# Generator

$x$: an image (a high-dimensional vector)

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




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

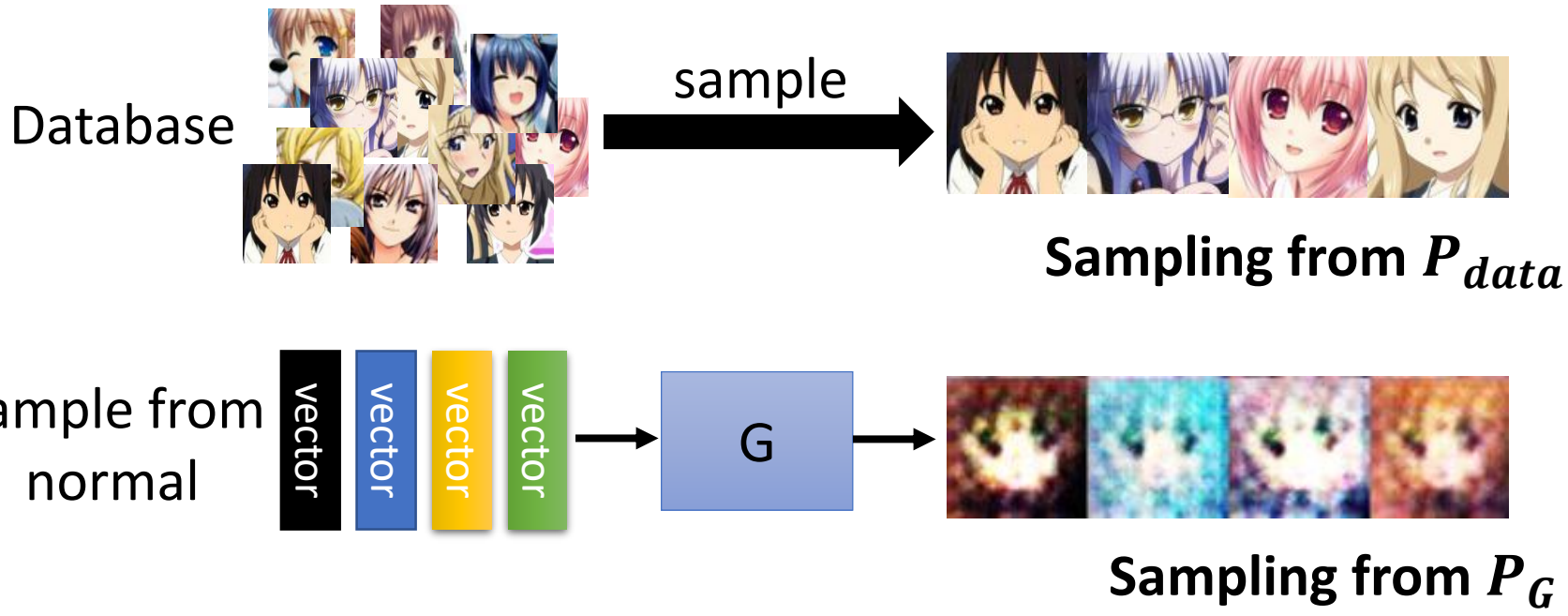
Divergence between distributions $P_G$ and $P_{data}$

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.



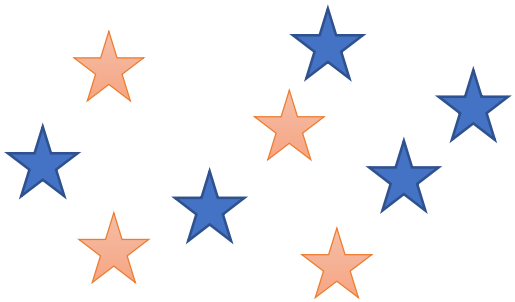
# Discriminator

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

: data sampled from $P_{data}$

: data sampled from $P_G$

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



train

Discriminator

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

The maximum objective value is related to JS divergence.

[Goodfellow, et al., NIPS, 2014]

# Discriminator

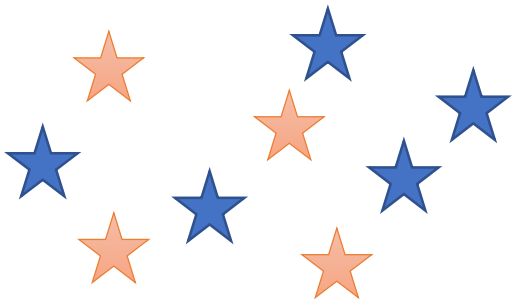
$$G^* = arg \min_G Div(P_G, P_{data})$$

: data sampled from $P_{data}$

: data sampled from $P_G$

**Training:**

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



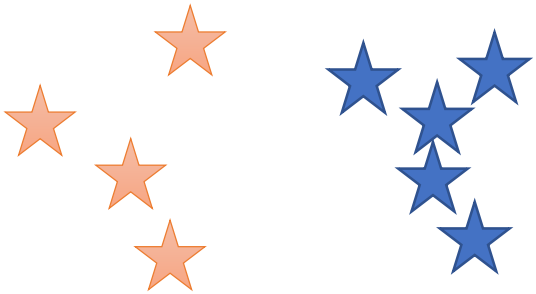
small divergence

train

Discriminator

hard to discriminate

Small $\max_D V(D, G)$



large divergence

train

Discriminator

easy to discriminate

[Goodfellow, et al., NIPS, 2014]

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

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

The maximum objective value is related to JS divergence.

- Initialize generator and discriminator
- In each training iteration:

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

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

# *Can we use other divergence?*

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

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 $\chi^2$ | $\frac{1}{4}t^2+t$ |
| Neyman $\chi^2$ | $2-2\sqrt{1-t}$ |
| Squared Hellinger | $\frac{t}{1-t}$ |
| Jeffrey | $W(e^{1-t})+\frac{1}{W(e^{1-t})}+t-2$ |
| Jensen-Shannon | $-\log(2-\exp(t))$ |
| Jensen-Shannon-weighted | $(1-\pi)\log\frac{1-\pi}{1-\pi e^{t/\pi}}$ |
| GAN | $-\log(1-\exp(t))$ |

# Outline of Part II

Basic Theory of GAN

Helpful Tips

How to evaluate GAN

Relation to Reinforcement Learning

# GAN is difficult to train ……


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

# Too many tips ……

- I do a little survey among 12 students …..

Q: What is the most helpful tip for training GAN?


WGAN (33.3%)

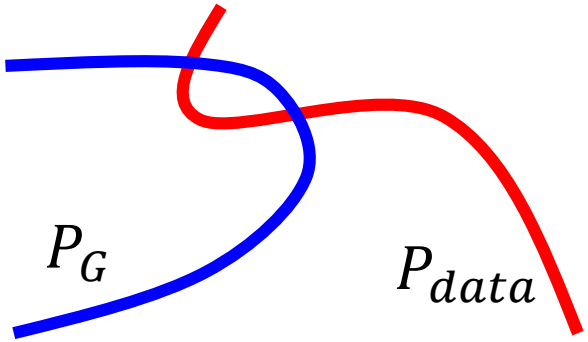

Spectral Norm (16.7%)

# 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.

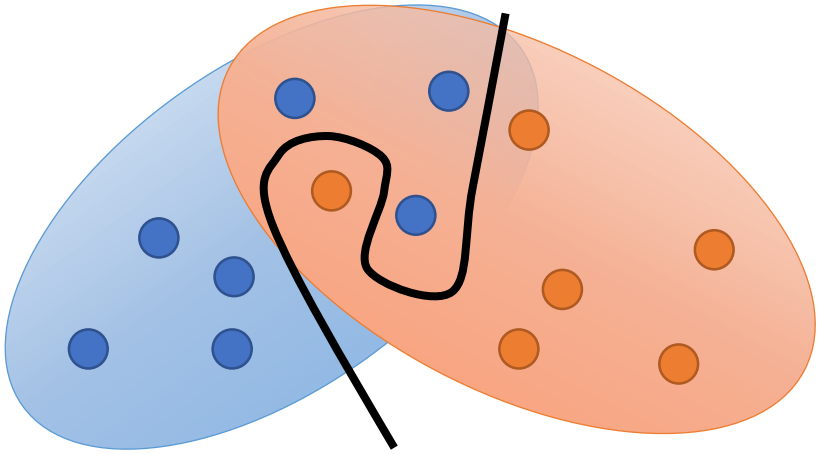
  The overlap can be ignored.



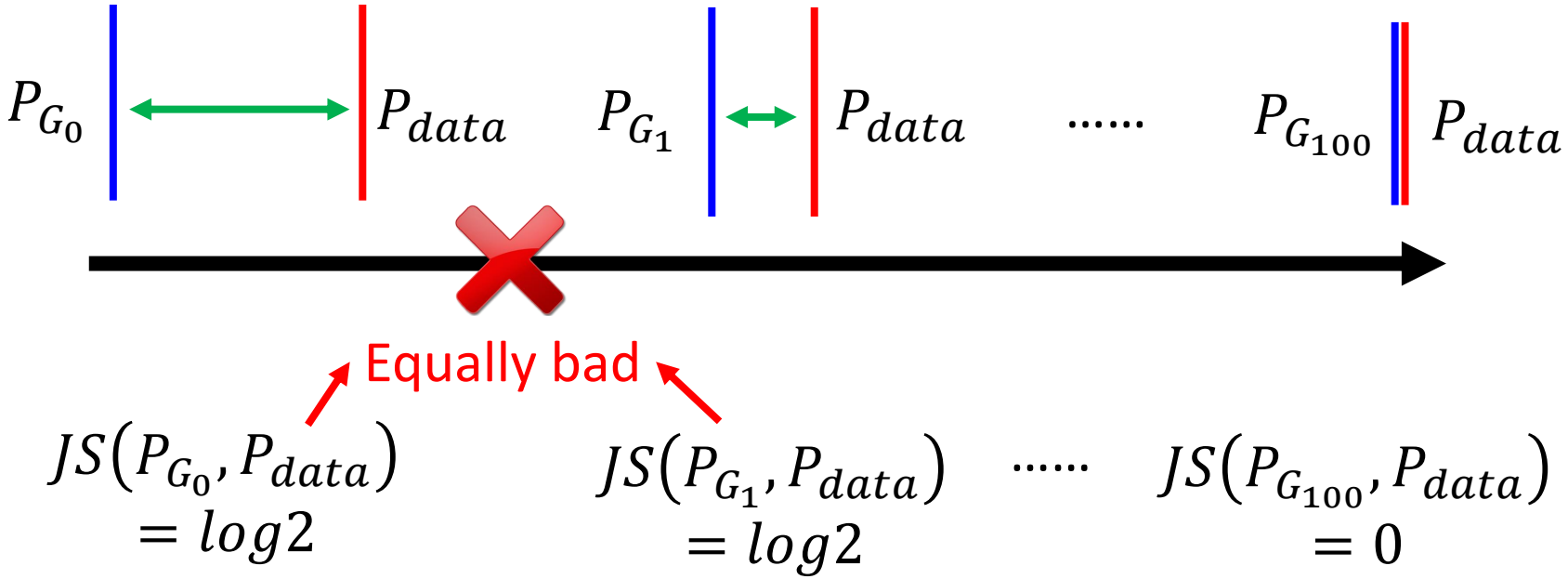

- 2. Sampling

  Even though $P_{data}$ and $P_G$ have overlap.

  If you do not have enough sampling ……



# *What is the problem of JS divergence?*



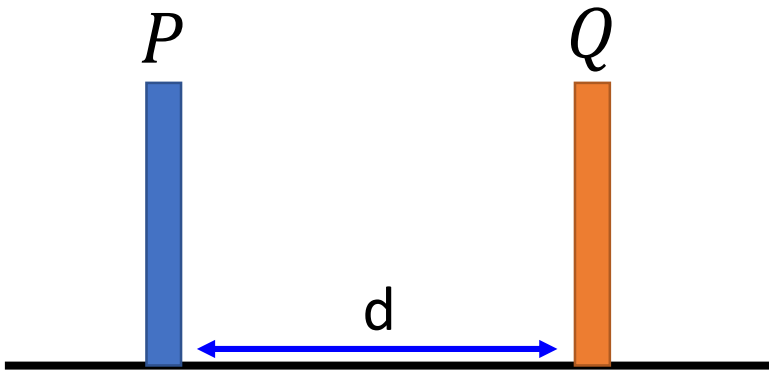

JS divergence is log2 if two distributions do not overlap.

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

➡ The same max objective value is obtained. ➡ Same divergence

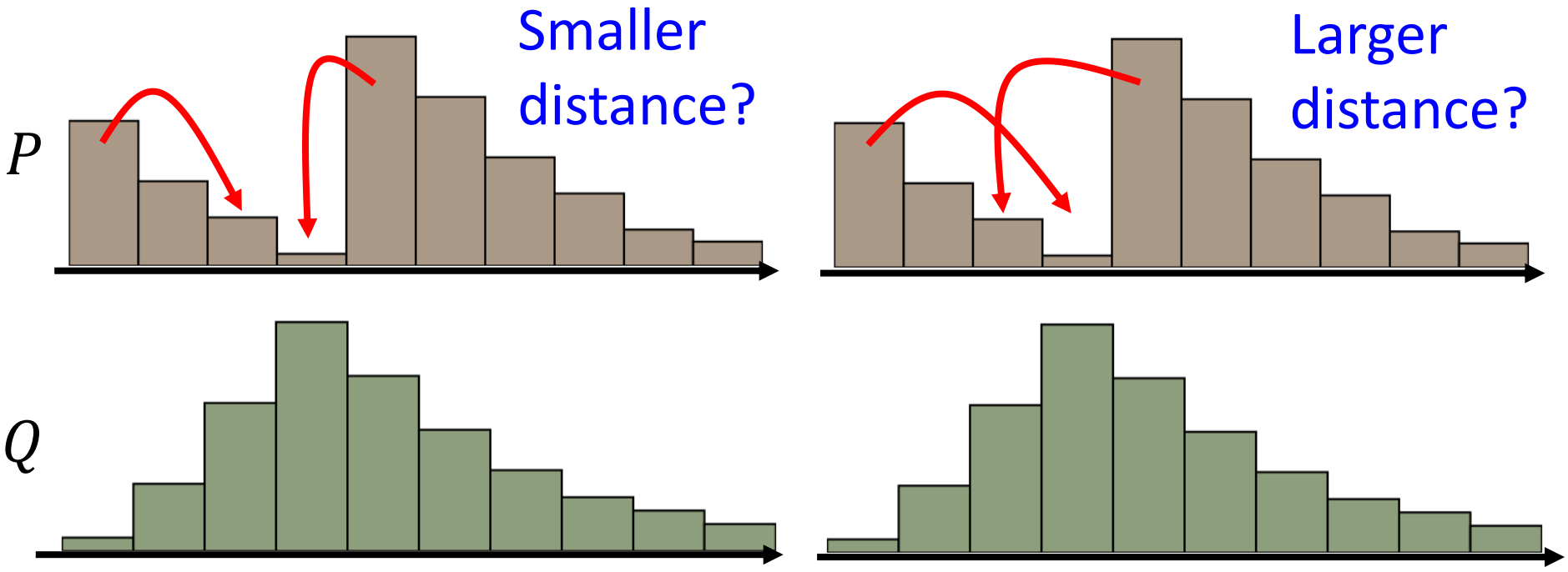
# Wasserstein distance

- 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

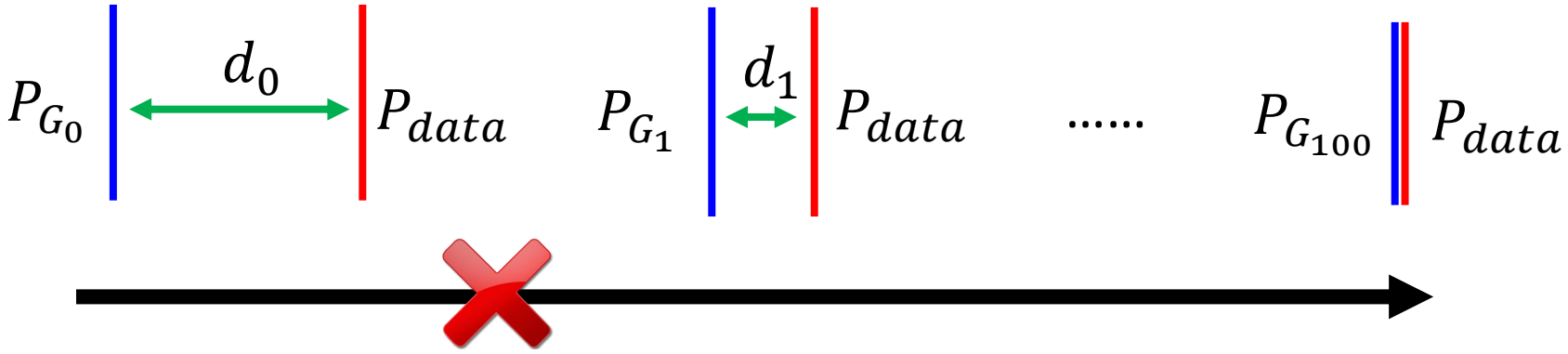



There are many possible “moving plans”.

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?




$JS(P_{G_0}, P_{data}) = log2$

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

..... $JS(P_{G_{100}}, P_{data}) = 0$

$W(P_{G_0}, P_{data}) = d_0$

$W(P_{G_1}, P_{data}) = d_1$

..... $W(P_{G_{100}}, P_{data}) = 0$

Better!

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

# WGAN

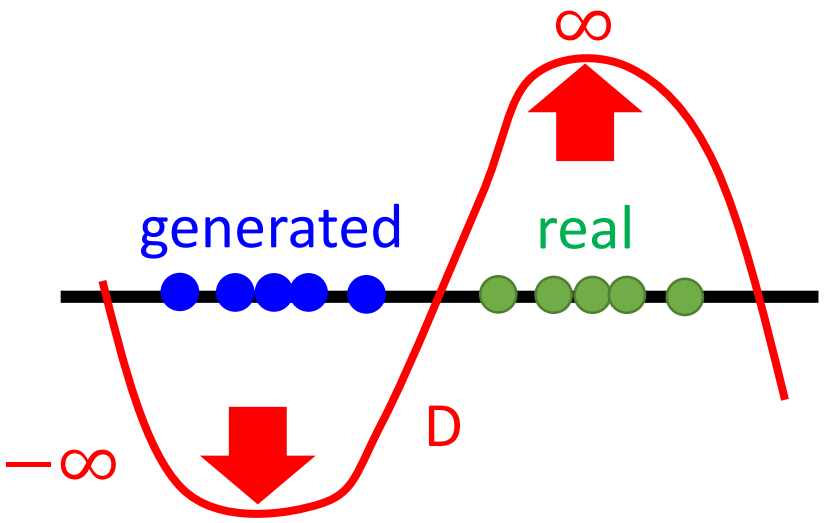
Evaluate Wasserstein distance between $P_{data}$ and $P_G$

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

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 $\infty$ and $-\infty$



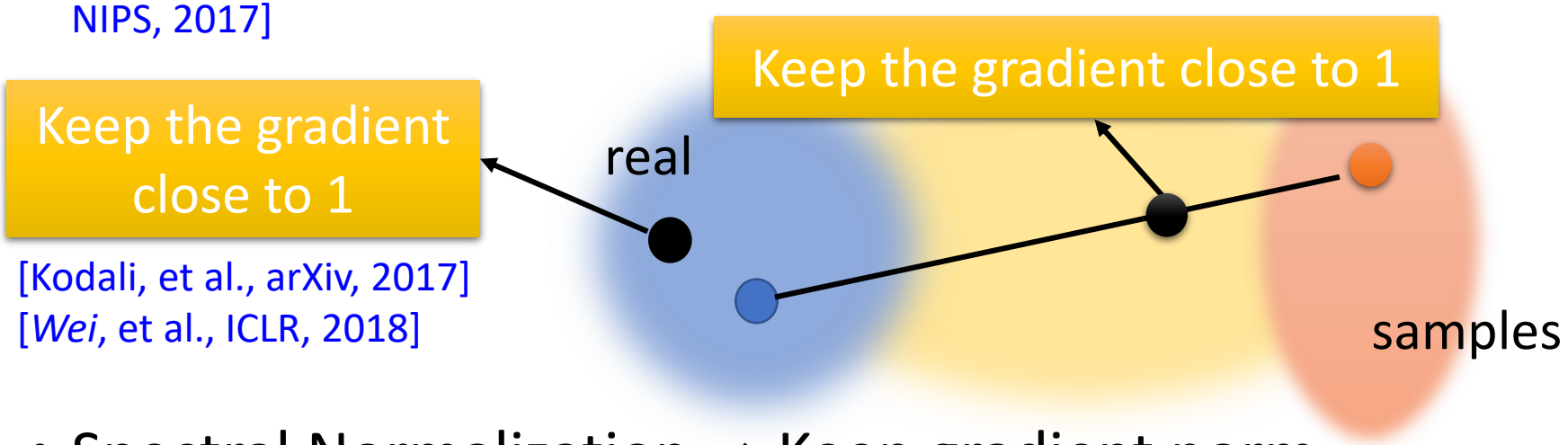
$$\max_{D \in 1-Lipschitz} \left\{ E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)] \right\}$$

- 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]

# More Tips

- Improved techniques for training GANs
  [Tim Salimans, et al., NIPS, 2016]
- Tips in DCGAN [Alec Radford, et al., ICLR 2016]
  - Guideline for network architecture design for image generation
- Tips from Soumith
  - https://github.com/soumith/ganhacks
- Tips from BigGAN [Andrew Brock, et al., arXiv, 2018]

# Outline of Part II

Basic Theory of GAN

Helpful Tips

How to evaluate GAN

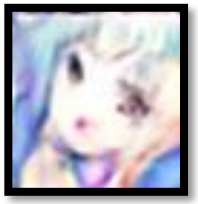
Relation to Reinforcement Learning

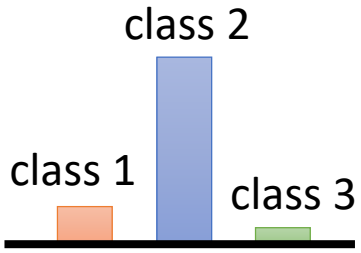
[Tim Salimans, et al., NIPS, 2016]

# Inception Score

$x$: 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 → $P(y^2|x^2)$

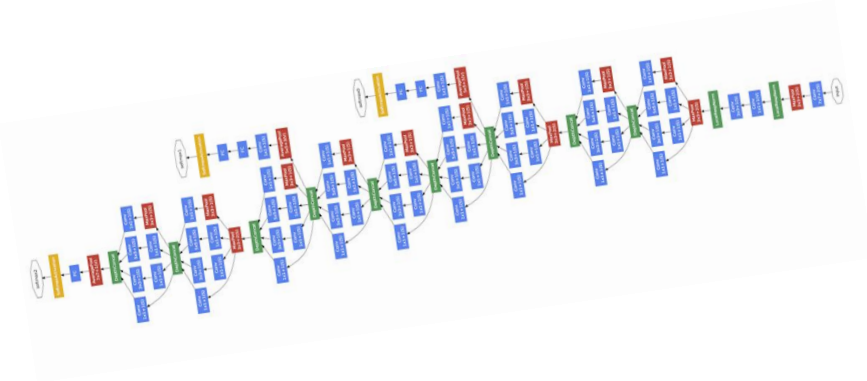
$x^3$ → CNN → $P(y^3|x^3)$

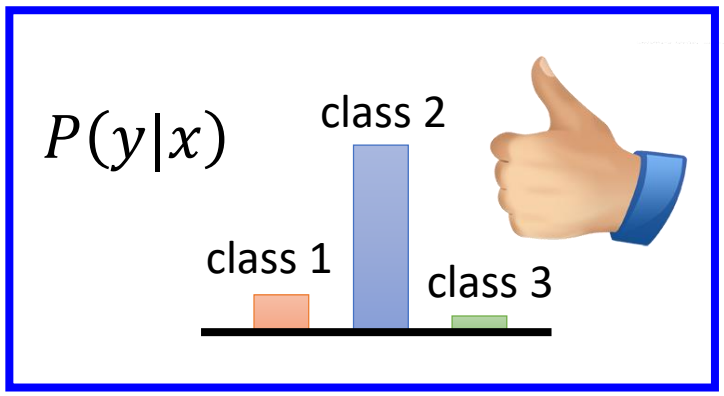
⋮

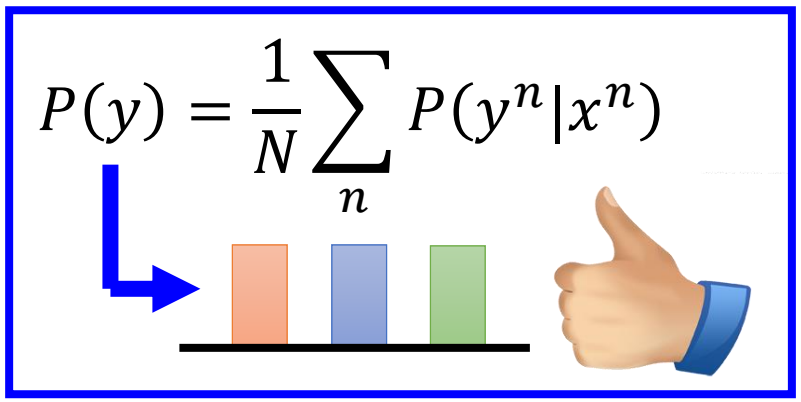
$$P(y) = \frac{1}{N}\sum_n P(y^n|x^n)$$

Uniform distribution means higher variety

# Inception Score





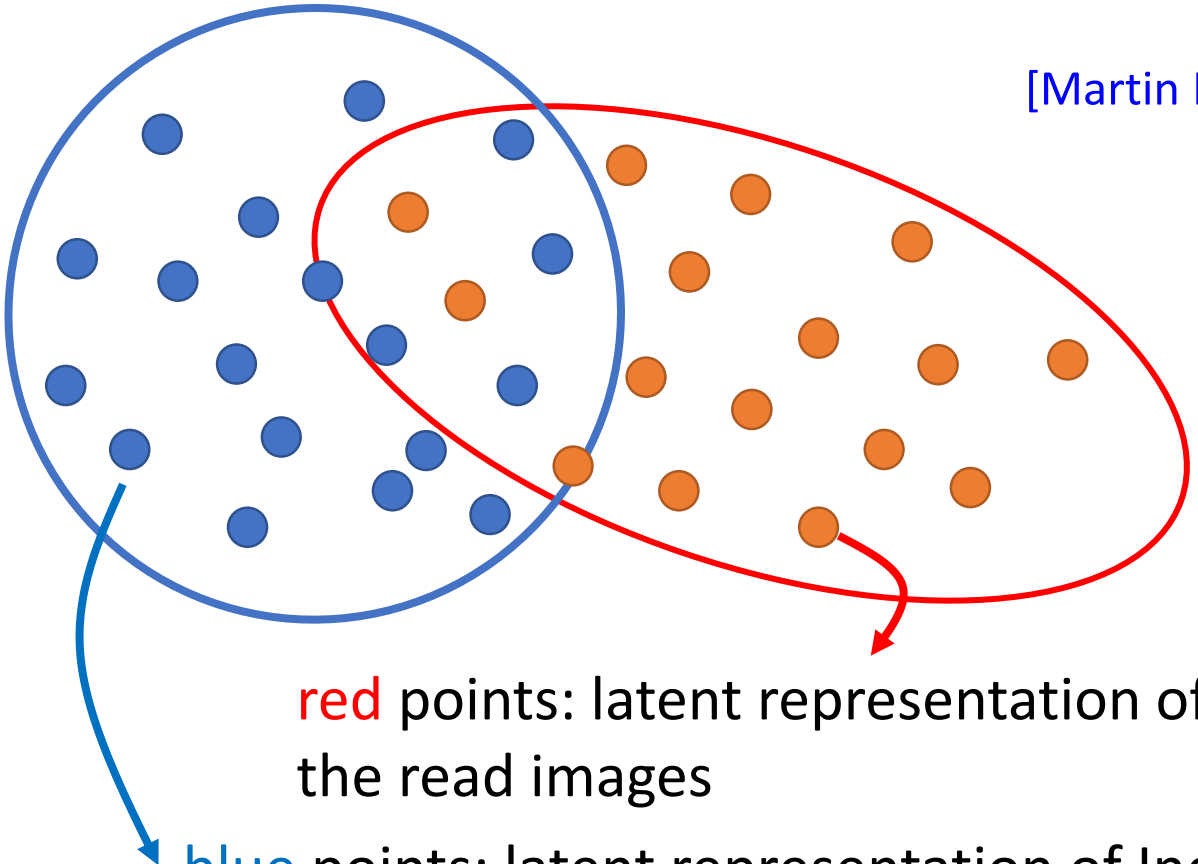




***Inception Score*** [Tim Salimans, et al., NIPS 2016]

$$= \sum_{x}\sum_{y} P(y|x)logP(y|x)$$ Negative entropy of P(y|x)

$$-\sum_{y} P(y)logP(y)$$ Entropy of P(y)

# Fréchet Inception Distance (FID)



[Martin Heusel, et al., NIPS, 2017]

FID =
Fréchet distance between the two Gaussians

red points: latent representation of Inception net for the read images

blue points: latent representation of Inception net for the generated images

# To learn more about evaluation …

https://arxiv.org/abs/1802.03446

| | Measure | Description |
|---|---|---|
| Quantitative | 1. Average Log-likelihood [18, 22] | • Log likelihood of explaining realworld held out/test data using a density estimated from the generated data (*e.g.* using KDE or Parzen window estimation). $L = \frac{1}{N}\sum_i \log P_{model}(\mathbf{x}_i)$ |
| | 2. Coverage Metric [33] | • The probability mass of the true data "covered" by the model distribution $C := P_{data}(dP_{model} > t)$ with $t$ such that $P_{model}(dP_{model} > t) = 0.95$ |
| | 3. Inception Score (IS) [3] | • KLD between conditional and marginal label distributions over generated data. $\exp\left(\mathbb{E}_{\mathbf{x}}\left[\mathbb{KL}\left(p(y \mid \mathbf{x}) \parallel p(y)\right)\right]\right)$ |
| | 4. Modified Inception Score (m-IS) [34] | • Encourages diversity within images sampled from a particular category. $\exp(\mathbb{E}_{\mathbf{x}_i}[\mathbb{E}_{\mathbf{x}_j}[(\mathbb{KL}(P(y\|\mathbf{x}_i)\|P(y\|\mathbf{x}_j))]])$ |
| | 5. Mode Score (MS) [35] | • Similar to IS but also takes into account the prior distribution of the labels over real data. $\exp\left(\mathbb{E}_{\mathbf{x}}\left[\mathbb{KL}\left(p(y \mid \mathbf{x}) \parallel p(y^{train})\right)\right] - \mathbb{KL}\left(p(y) \parallel p(y^{train})\right)\right)$ |
| | 6. AM Score [36] | • Takes into account the KLD between distributions of training labels vs. predicted labels, as well as the entropy of predictions. $\mathbb{KL}(p(y^{train}) \parallel p(y)) + \mathbb{E}_{\mathbf{x}}\left[H(y\|\mathbf{x})\right]$ |
| | 7. Fréchet Inception Distance (FID) [37] | • Wasserstein-2 distance between multi-variate Gaussians fitted to data embedded into a feature space $FID(r,g) = \|\mu_r - \mu_g\|_2^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}})$ |
| | 8. Maximum Mean Discrepancy (MMD) [38] | • Measures the dissimilarity between two probability distributions $P_r$ and $P_g$ using samples drawn independently from each distribution. $M_k(P_r,P_g) = \mathbb{E}_{\mathbf{x},\mathbf{x}'\sim P_r}[k(\mathbf{x},\mathbf{x}')] - 2\mathbb{E}_{\mathbf{x}\sim P_r,\mathbf{y}\sim P_g}[k(\mathbf{x},\mathbf{y})] + \mathbb{E}_{\mathbf{y},\mathbf{y}'\sim P_g}[k(\mathbf{y},\mathbf{y}')]$ |
| | 9. The Wasserstein Critic [39] | • The critic (*e.g.* an NN) is trained to produce high values at real samples and low values at generated samples $\hat{W}(\mathbf{x}_{test},\mathbf{x}_g) = \frac{1}{N}\sum_{i=1}^{N}\hat{f}(\mathbf{x}_{test}[i]) - \frac{1}{N}\sum_{i=1}^{N}\hat{f}(\mathbf{x}_g[i])$ |
| | 10. Birthday Paradox Test [27] | • Measures the support size of a discrete (continuous) distribution by counting the duplicates (near duplicates) |
| | 11. Classifier Two Sample Test (C2ST) [40] | • Answers whether two samples are drawn from the same distribution (*e.g.* by training a binary classifier) |
| | 12. Classification Performance [1, 15] | • An indirect technique for evaluating the quality of unsupervised representations (*e.g.* feature extraction; FCN score). See also the GAN Quality Index (GQI) [41]. |
| | 13. Boundary Distortion [42] | • Measures diversity of generated samples and covariate shift using classification methods. |
| | 14. Number of Statistically-Different Bins (NDB) [43] | • Given two sets of samples from the same distribution, the number of samples that fall into a given bin should be the same up to sampling noise |
| | 15. Image Retrieval Performance [44] | • Measures the distributions of distances to the nearest neighbors of some query images (*i.e.* diversity) |
| | 16. Generative Adversarial Metric (GAM) [31] | • Compares two GANs by having them engaged in a battle against each other by swapping discriminators or generators. $p(\mathbf{x}\|y=1;M_1^*)/p(\mathbf{x}\|y=1;M_2^*) = (p(y=1\|\mathbf{x};D_1)p(\mathbf{x};G_2))/(p(y=1\|\mathbf{x};D_2)p(\mathbf{x};G_1))$ |
| | 17. Tournament Win Rate and Skill Rating [45] | • Implements a tournament in which a player is either a discriminator that attempts to distinguish between real and fake data or a generator that attempts to fool the discriminators into accepting fake data as real. |
| | 18. Normalized Relative Discriminative Score (NRDS) [32] | • Compares $n$ GANs based on the idea that if the generated samples are closer to real ones, more epochs would be needed to distinguish them from real samples. |
| | 19. Adversarial Accuracy and Divergence [46] | • Adversarial Accuracy. Computes the classification accuracies achieved by the two classifiers, one trained on real data and another on generated data, on a labeled validation set to approximate $P_g(y\|\mathbf{x})$ and $P_r(y\|\mathbf{x})$. Adversarial Divergence: Computes $\mathbb{KL}(P_g(y\|\mathbf{x}), P_r(y\|\mathbf{x}))$ |
| | 20. Geometry Score [47] | • Compares geometrical properties of the underlying data manifold between real and generated data. |
| | 21. Reconstruction Error [48] | • Measures the reconstruction error (*e.g.* $L_2$ norm) between a test image and its closest generated image by optimizing for $z$ (*i.e.* $min_{\mathbf{z}}\|\|G(\mathbf{z}) - \mathbf{x}^{(test)}\|\|^2$) |
| | 22. Image Quality Measures [49, 50, 51] | • Evaluates the quality of generated images using measures such as SSIM, PSNR, and sharpness difference |
| | 23. Low-level Image Statistics [52, 53] | • Evaluates how similar low-level statistics of generated images are to those of natural scenes in terms of mean power spectrum, distribution of random filter responses, contrast distribution, etc. |
| | 24. Precision, Recall and $F_1$ score [23] | • These measures are used to quantify the degree of overfitting in GANs, often over toy datasets. |
| Qualitative | 1. Nearest Neighbors | • To detect overfitting, generated samples are shown next to their nearest neighbors in the training set |
| | 2. Rapid Scene Categorization [18] | • In these experiments, participants are asked to distinguish generated samples from real images in a short presentation time (*e.g.* 100 ms); *i.e.* real v.s fake |
| | 3. Preference Judgment [54, 55, 56, 57] | • Participants are asked to rank models in terms of the fidelity of their generated images (*e.g.* pairs, triples) |
| | 4. Mode Drop and Collapse [58, 59] | • Over datasets with known modes (*e.g.* a GMM or a labeled dataset), modes are computed as by measuring the distances of generated data to mode centers |
| | 5. Network Internals [1, 60, 61, 62, 63, 64] | • Regards exploring and illustrating the internal representation and dynamics of models (*e.g.* space continuity) as well as visualizing learned features |

Pros and cons of GAN evaluation measures [Ali Borji, 2019]

# Outline of Part II

Basic Theory of GAN

Helpful Tips

How to evaluate GAN

Relation to Reinforcement Learning

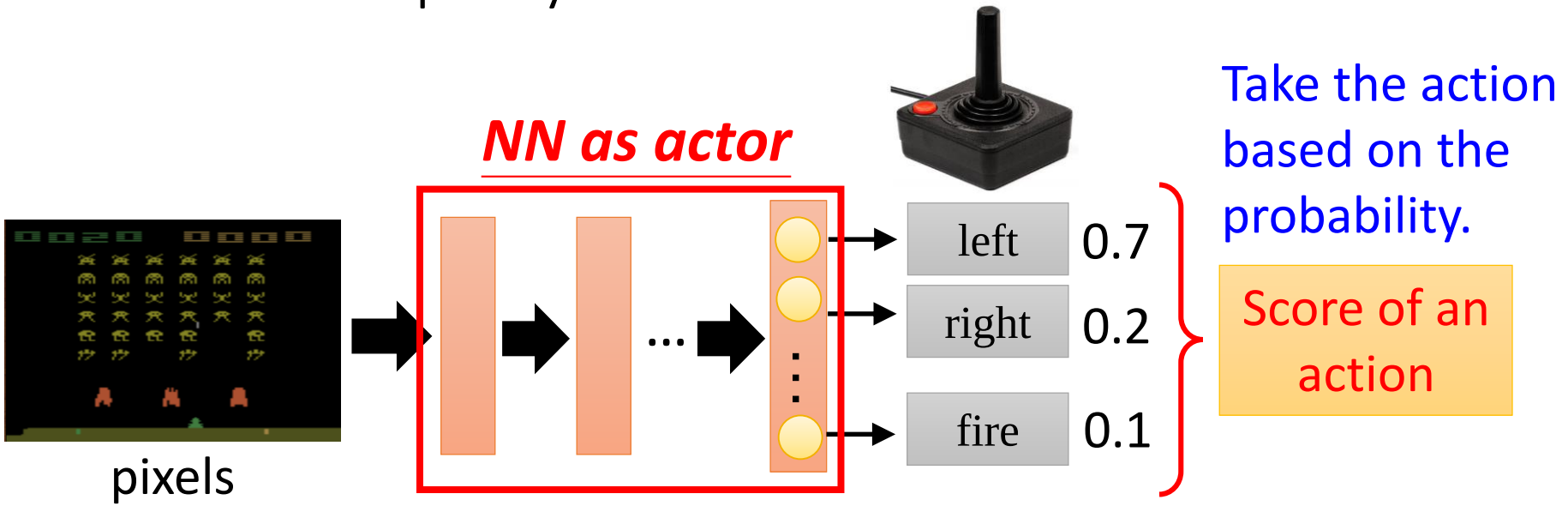
# Basic Components

You cannot control

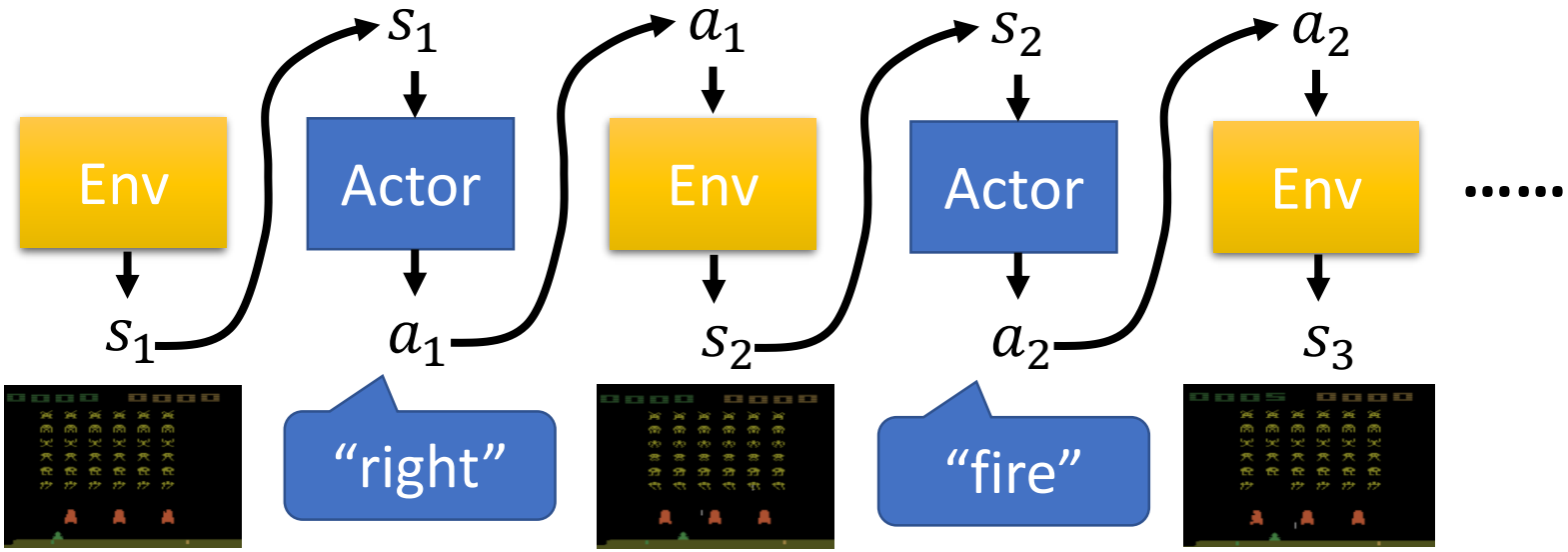
| |  Actor | Env | Reward Function |
|---|---|---|---|
| Video Game |  |  | Get 20 scores when killing a monster |
| Go |   |   | The rule of GO |

# 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



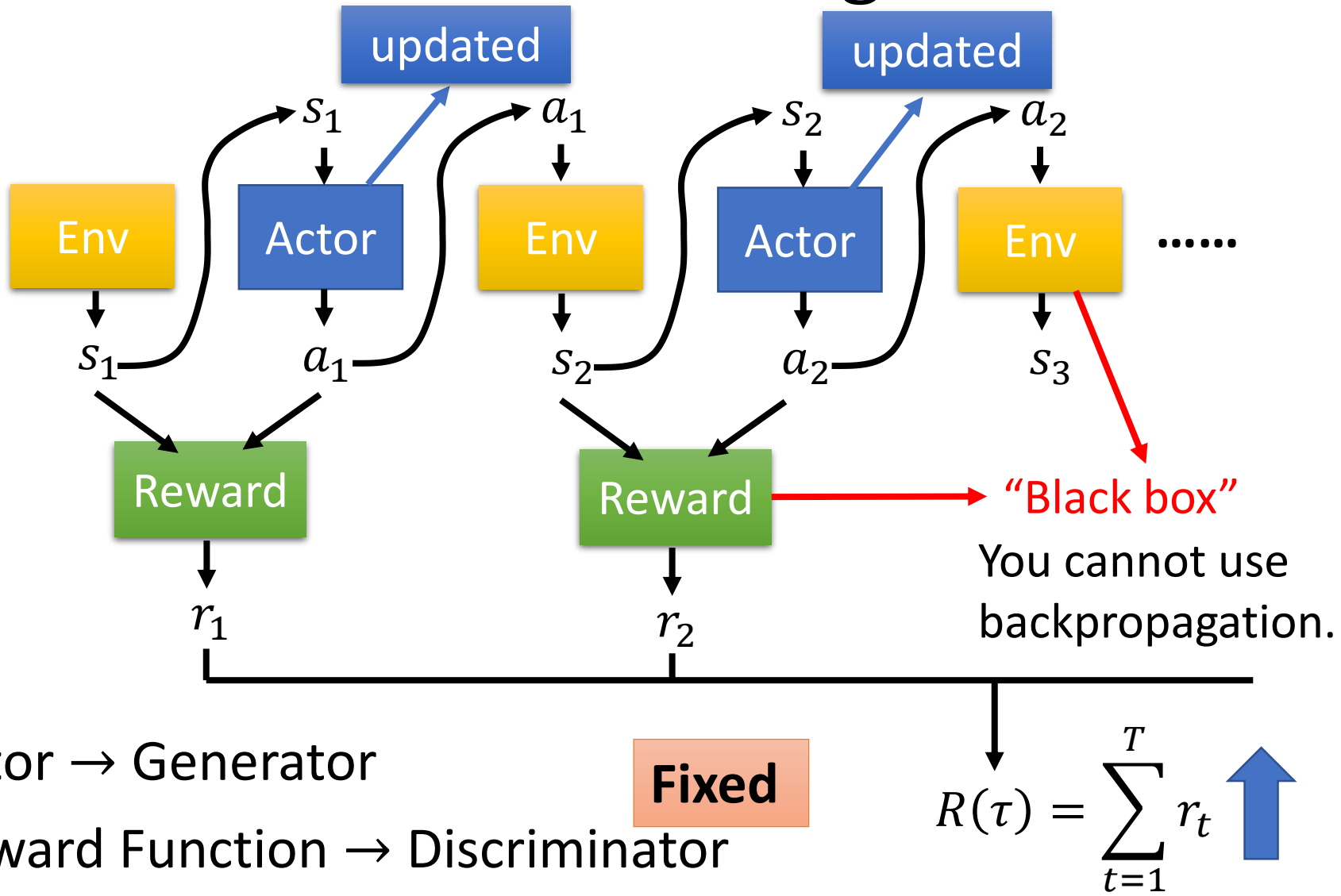
# Actor, Environment, Reward



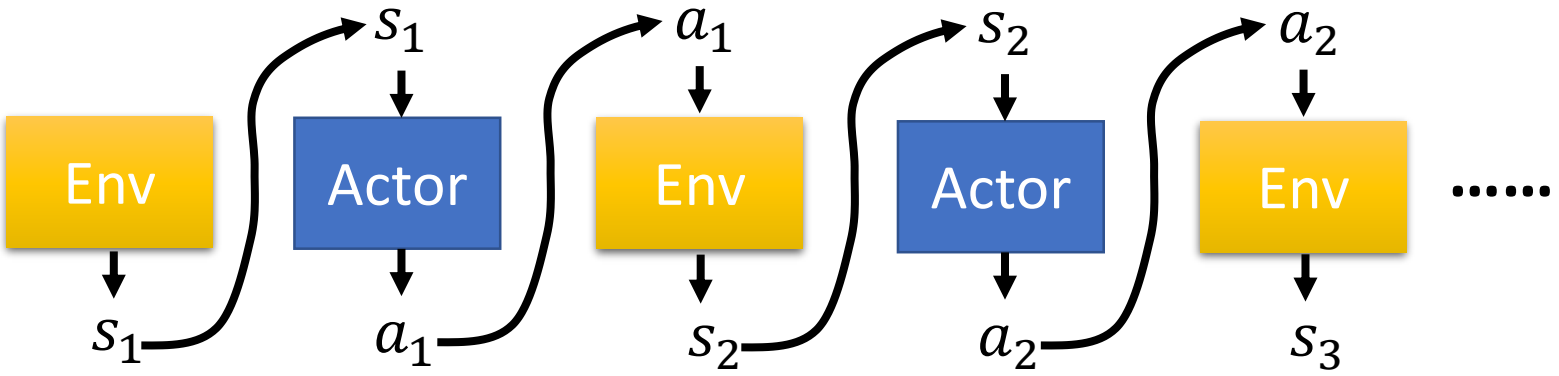

**Trajectory**

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

# Reinforcement Learning v.s. GAN



# Inverse Reinforcement Learning



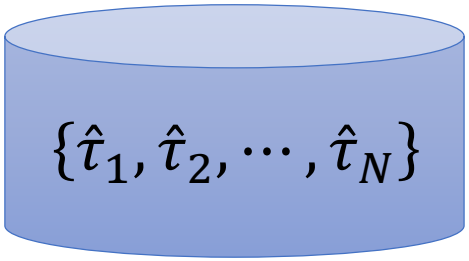

reward function is not available

(in many cases, it is difficult to define reward function)

We have demonstration of the expert.

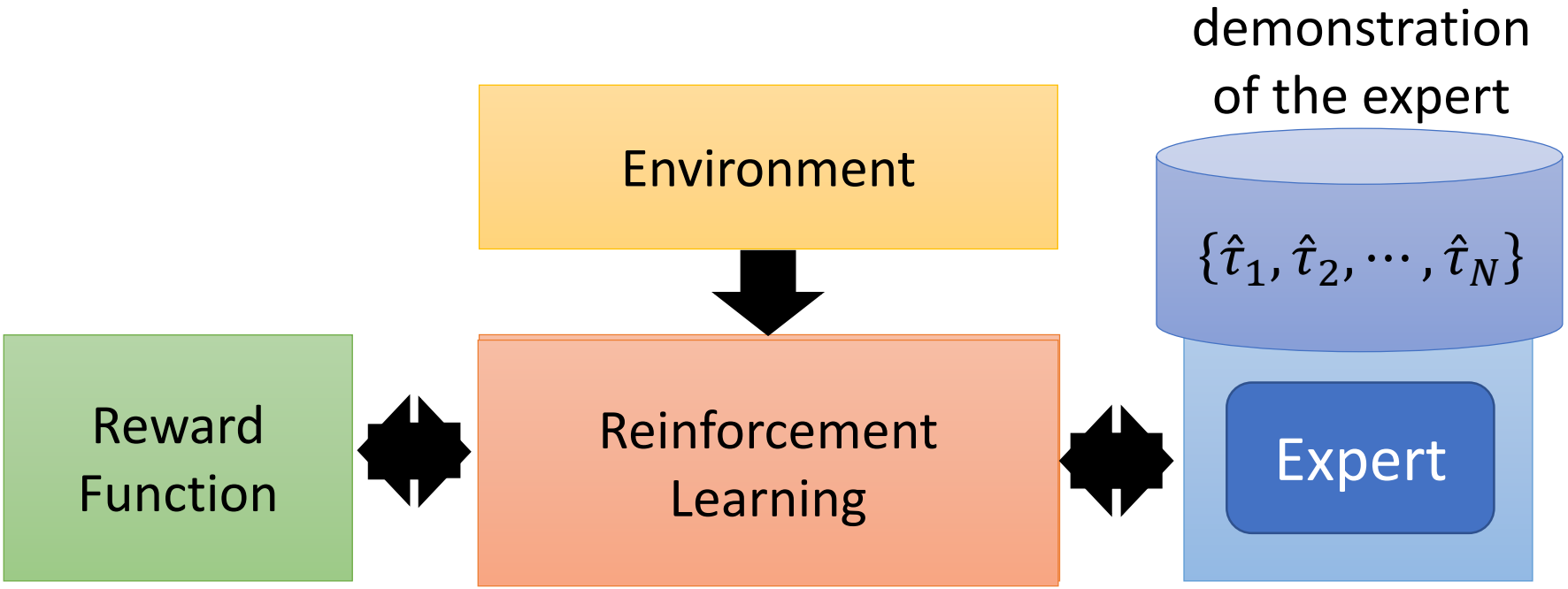
Self driving: record human drivers

Robot: grab the arm of robot




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

# Inverse Reinforcement Learning

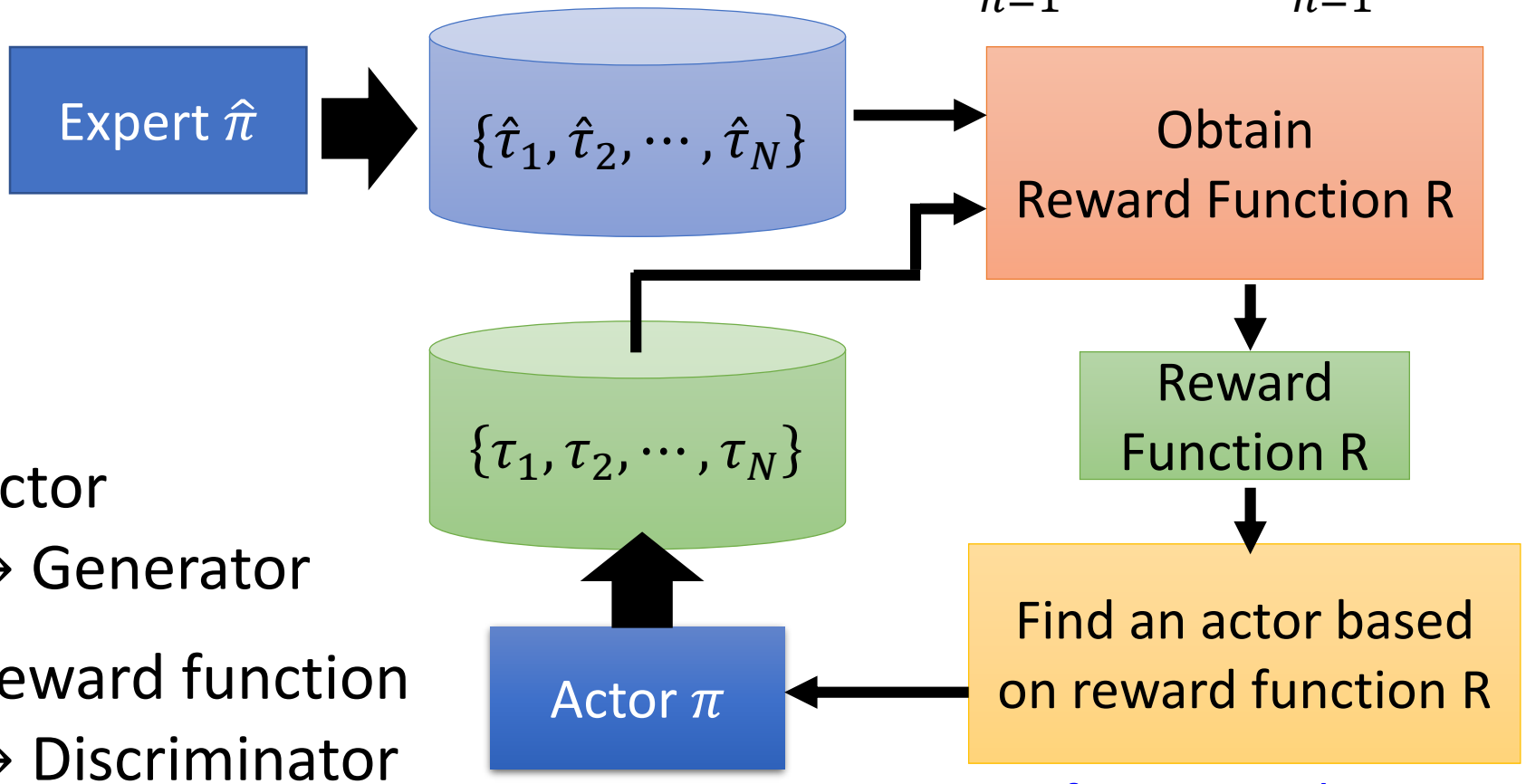



- Using the reward function to find the ***optimal actor***.
- Modeling reward can be easier. Simple reward function can lead to complex policy.

# Framework of IRL

***The expert is always the best.***

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

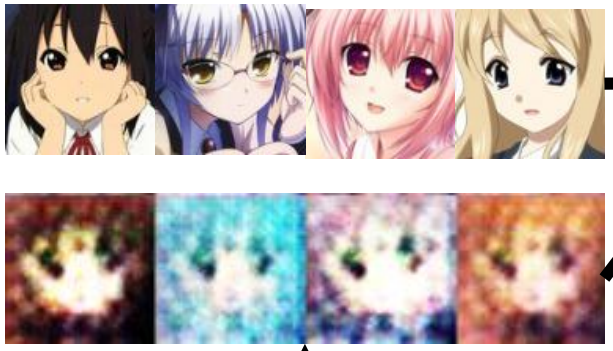



Actor
→ Generator

Reward function
→ Discriminator

By Reinforcement learning

## 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 $\tau$

Reward
Function

Find a Actor obtains
large reward

Actor

# Outline of Part II

Basic Theory of GAN

Helpful Tips

How to evaluate GAN

Relation to Reinforcement Learning

# Reference


- 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

# Generative Adversarial Network

and its Applications to Signal Processing
and Natural Language Processing

# Part III: Speech Signal Processing

Tsao, Yu Ph.D., Academia Sinica
yu.tsao@citi.sinica.edu.tw

# Outline of Part III

## Speech Signal Generation

- Speech enhancement
- Postfilter, speech synthesis, voice conversion

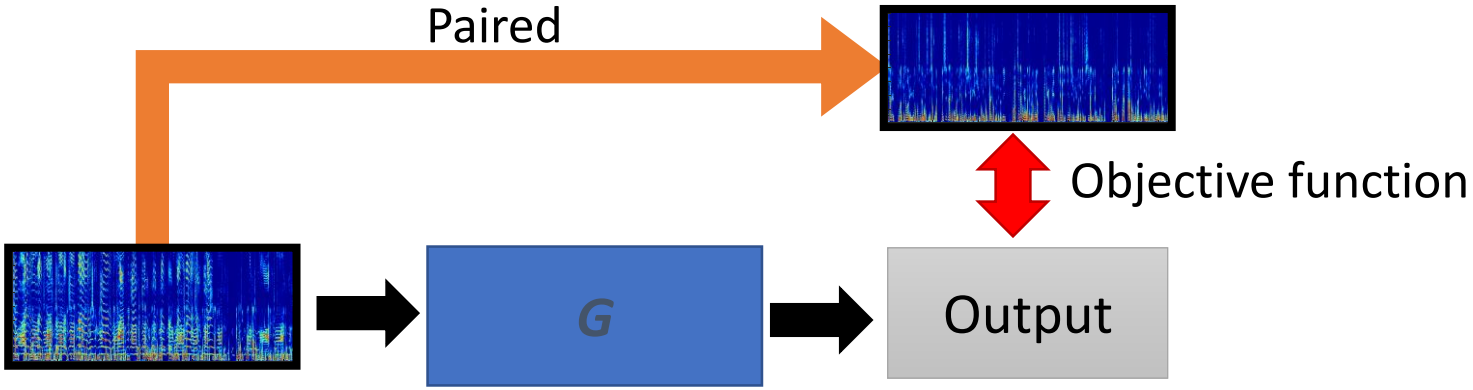
## Speech Signal Recognition

- Speech recognition
- Speaker recognition
- Speech emotion recognition
- Lip reading

## Conclusion

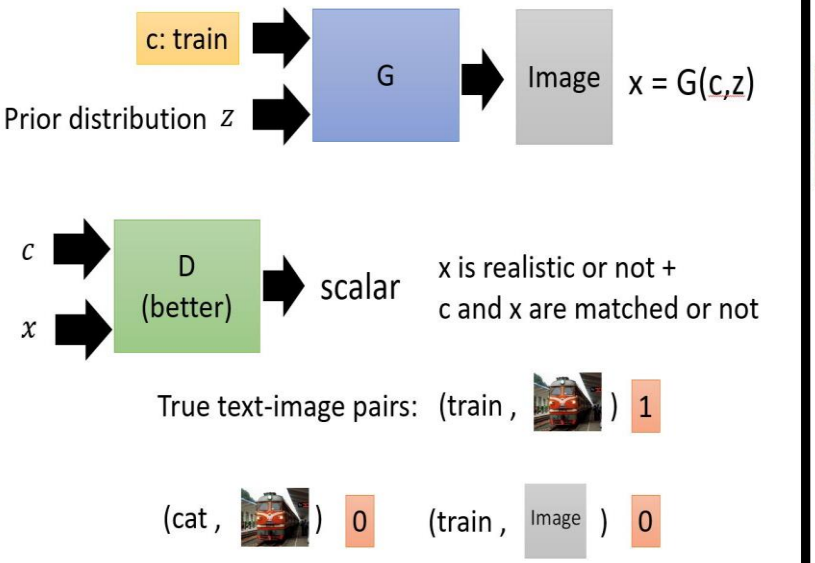
## Our Recent Works

# Speech Signal Generation (Regression Task)

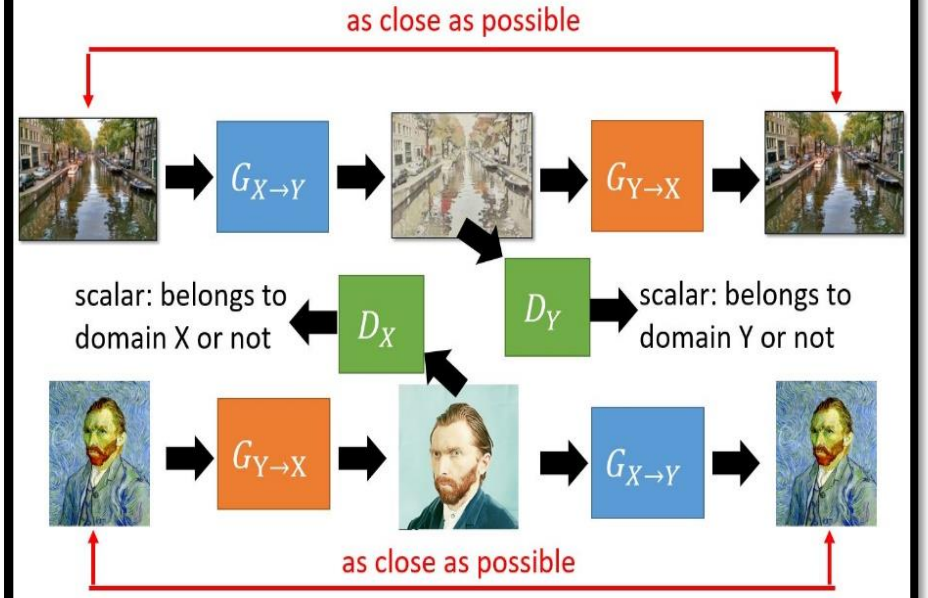



[Scott Reed, et al, ICML, 2016]

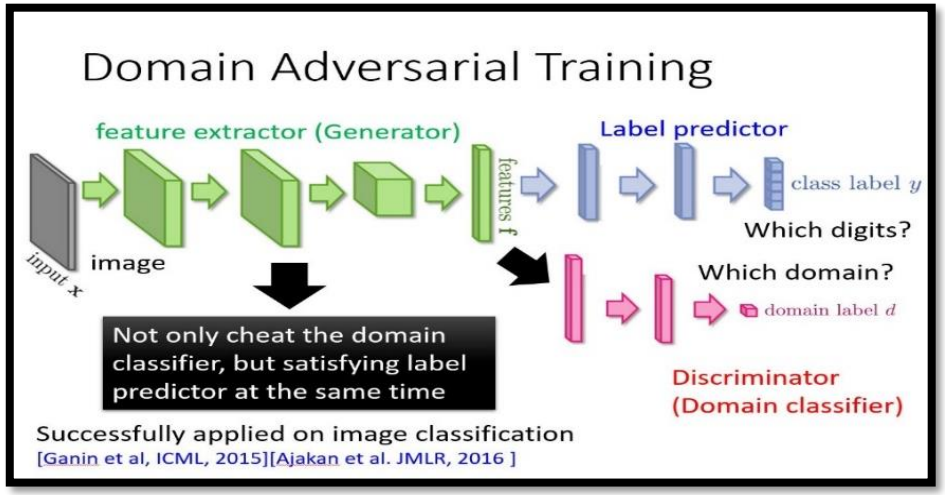
## Conditional GAN



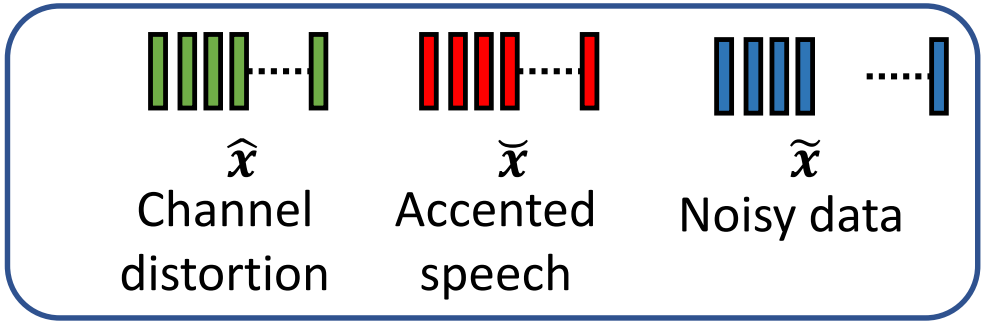

## Cycle-GAN

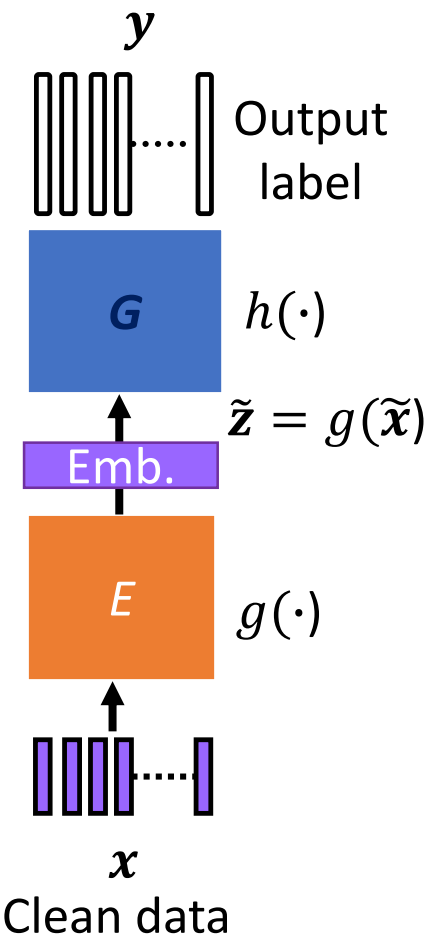


# Speech, Speaker, Emotion Recognition and Lip-reading (Classification Task)




Acoustic Mismatch



# Outline of Part III

## Speech Signal Generation

- Speech enhancement
- Postfilter, speech synthesis, voice conversion

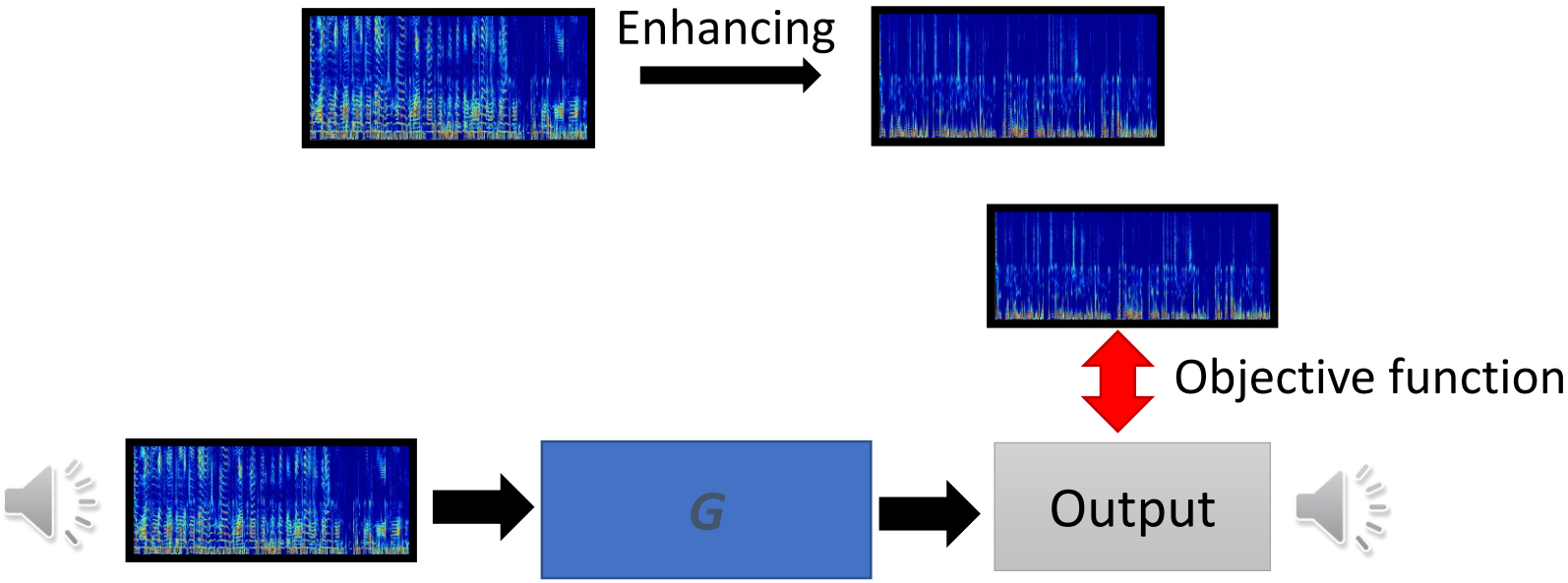
## Speech Signal Recognition

- Speech recognition
- Speaker recognition
- Speech emotion recognition
- Lip reading

## Conclusion

## Our Recent Works

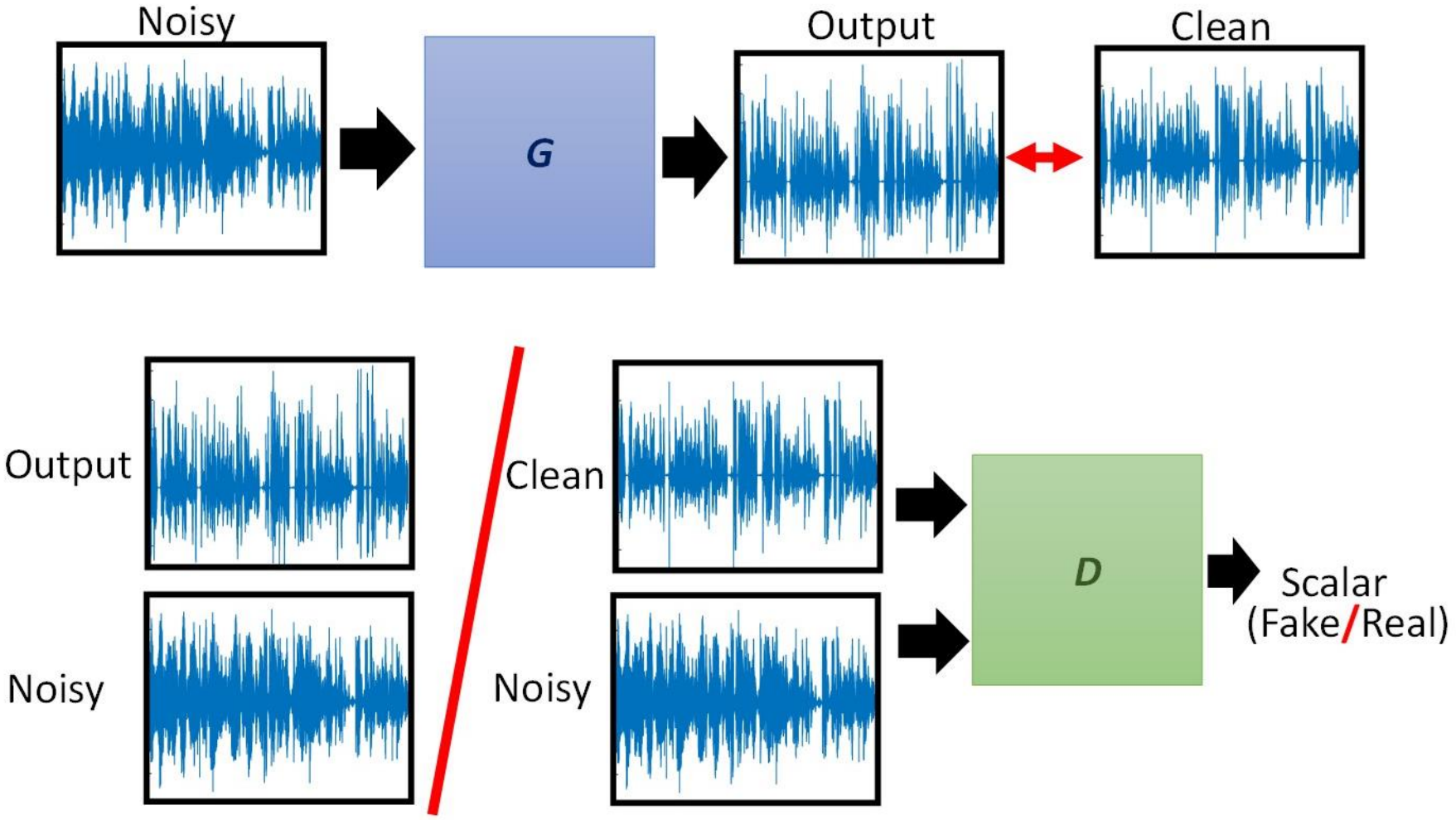
# Speech Enhancement




- Neural network models for spectral mapping
  - Model structures of ***G***: DNN [Wang et al. NIPS 2012; Xu et al., SPL 2014], DDAE [Lu et al., Interspeech 2013], RNN (LSTM) [Chen et al., Interspeech 2015; Weninger et al., LVA/ICA 2015], CNN [Fu et al., Interspeech 2016].
- Typical objective function
  - Mean square error (MSE) [Xu et al., TASLP 2015], L1 [Pascual et al., Interspeech 2017], likelihood [Chai et al., MLSP 2017], STOI [Fu et al., TASLP 2018].
  - GAN is used as a new objective function to estimate the parameters in ***G***.

# Speech Enhancement

- Speech enhancement GAN (SEGAN) [Pascual et al., Interspeech 2017]



# Speech Enhancement (SEGAN)

- Experimental results

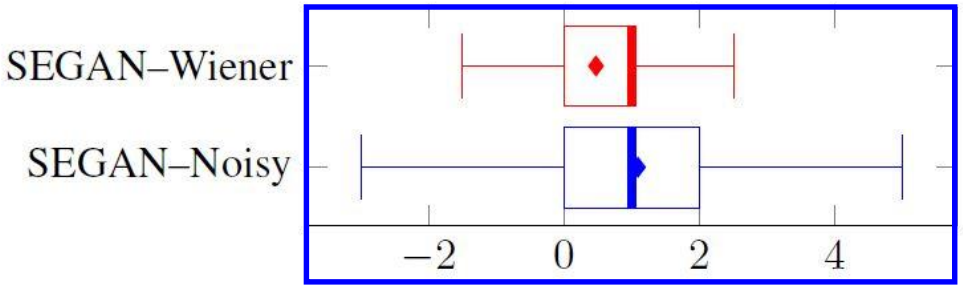
Table 1: Objective evaluation results.

| Metric | Noisy | Wiener | SEGAN |
|---|---|---|---|
| PESQ | 1.97 | 2.22 | 2.16 |
| CSIG | 3.35 | 3.23 | 3.48 |
| CBAK | 2.44 | 2.68 | 2.94 |
| COVL | 2.63 | 2.67 | 2.80 |
| SSNR | 1.68 | 5.07 | 7.73 |

Table 2: Subjective evaluation results.

| Metric | Noisy | Wiener | SEGAN |
|---|---|---|---|
| MOS | 2.09 | 2.70 | 3.18 |

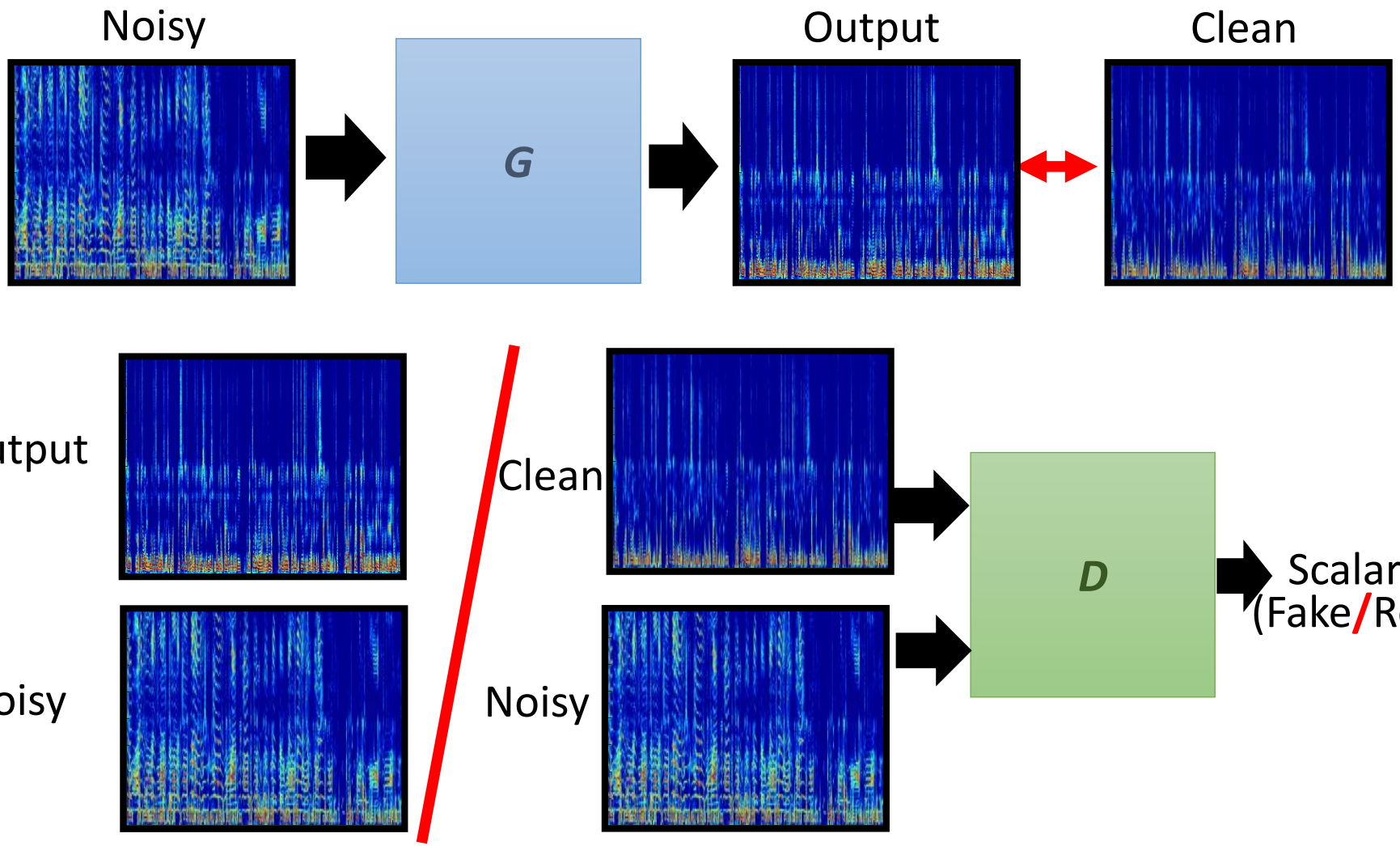
Fig. 1: Preference test results.




SEGAN yields better speech enhancement results than Noisy and Wiener.

# Speech Enhancement

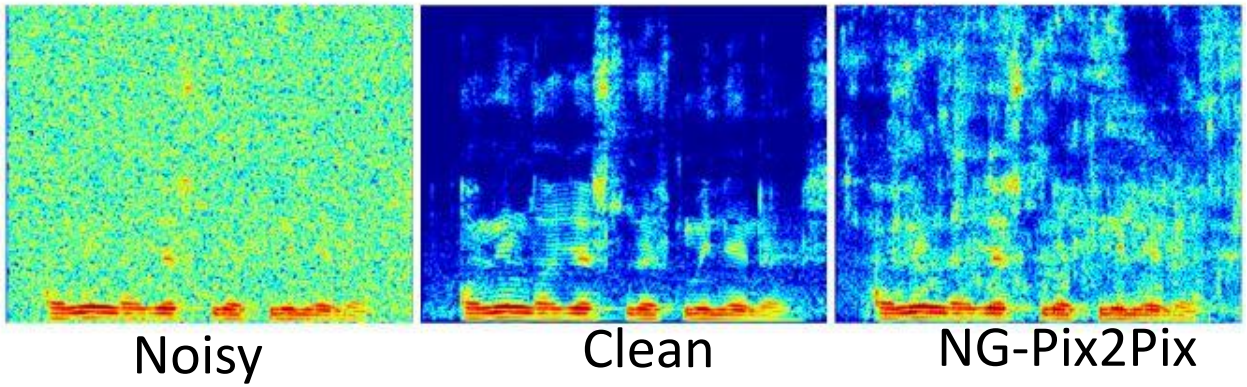
- Pix2Pix [Michelsanti et al., Interpsech 2017]



# Speech Enhancement (Pix2Pix)

- Spectrogram analysis

Fig. 2: Spectrogram comparison of Pix2Pix with baseline methods.




Pix2Pix outperforms STAT-MMSE and is competitive to DNN SE.

# Speech Enhancement (Pix2Pix)

- Objective evaluation and speaker verification test

Table 3: Objective evaluation results.

| | | PESQ | | | | | |
|---|---|---|---|---|---|---|---|
| | SNR | 0 | 5 | 10 | 15 | 20 | mean |
| Babble | (a) | 1.20 | 1.42 | 1.79 | 2.40 | **3.13** | 1.99 |
| | (b) | 1.14 | 1.31 | 1.61 | 2.07 | 2.65 | 1.76 |
| | (c) | **1.25** | 1.51 | 1.87 | 2.31 | 2.78 | 1.95 |
| | **(d)** | 1.20 | 1.48 | 1.98 | 2.52 | 2.93 | 2.02 |
| | (e) | 1.24 | **1.52** | 1.88 | 2.31 | 2.78 | 1.95 |
| | **(f)** | 1.20 | 1.49 | **2.00** | **2.53** | 2.93 | **2.03** |

| | | STOI | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0 | 5 | 10 | 15 | 20 | mean |
| Babble | (a) | 0.44 | 0.56 | 0.67 | 0.77 | 0.85 | 0.66 |
| | (b) | 0.43 | 0.56 | 0.66 | 0.74 | 0.81 | 0.64 |
| | (c) | **0.50** | **0.63** | **0.72** | **0.79** | **0.86** | **0.70** |
| | **(d)** | 0.46 | 0.59 | 0.71 | 0.78 | 0.83 | 0.67 |
| | (e) | 0.49 | 0.62 | **0.72** | **0.79** | 0.85 | **0.70** |
| | **(f)** | 0.46 | 0.60 | 0.71 | 0.77 | 0.82 | 0.67 |

Table 4: Speaker verification results.

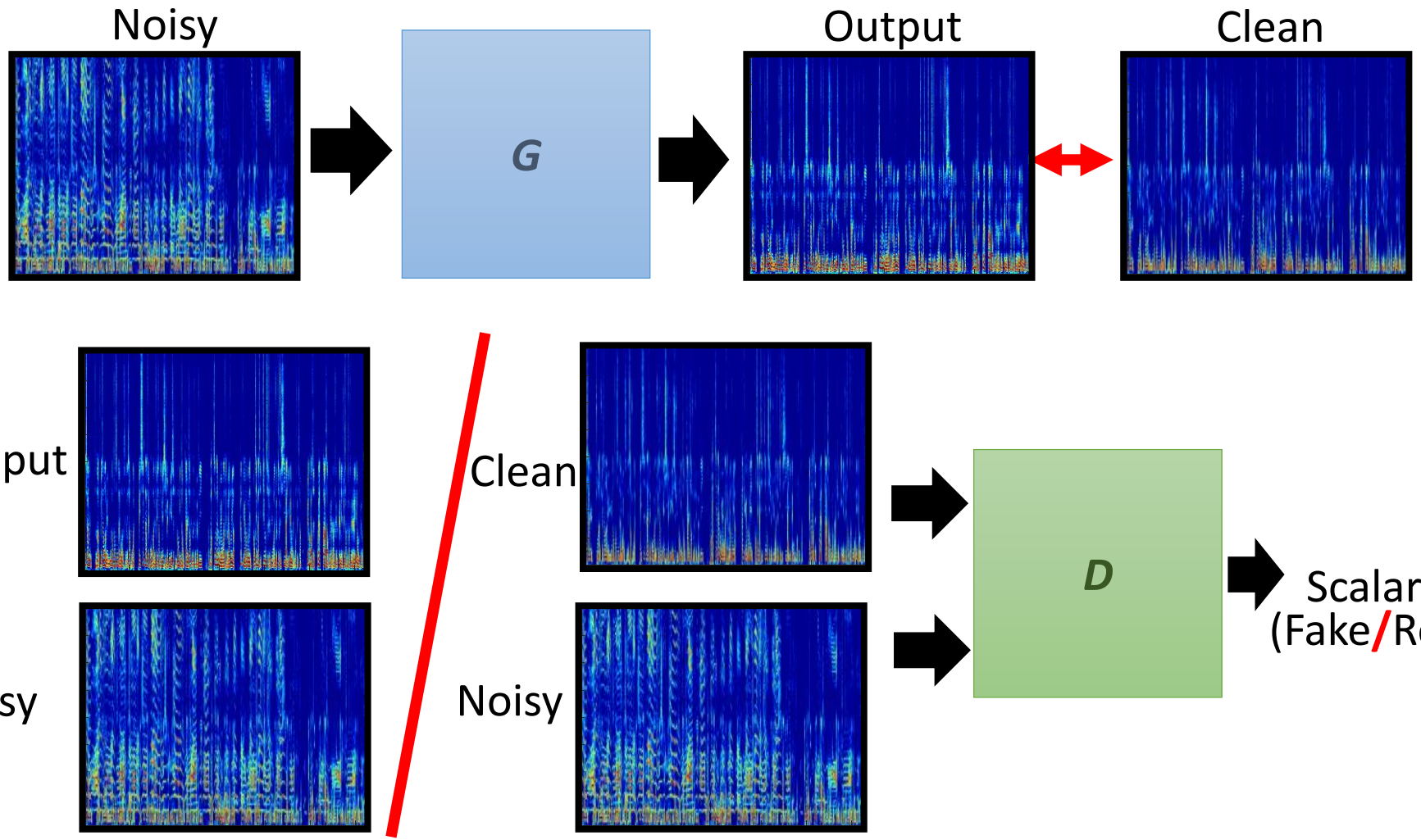
| | SNR | 0 | 5 | 10 | 15 | 20 | clean | mean |
|---|---|---|---|---|---|---|---|---|
| Airplane | (a) | 21.09 | 15.99 | 13.61 | 11.66 | 9.18 | 6.99 | 13.08 |
| | (b) | 17.69 | 12.58 | 8.17 | 6.53 | 6.27 | 5.80 | 9.51 |
| | (c) | 16.99 | 10.55 | 7.48 | 6.99 | 6.15 | 6.12 | 9.05 |
| | **(d)** | 17.19 | 8.84 | **5.44** | 5.05 | **4.63** | **3.74** | 7.48 |
| | (e) | 15.99 | 8.99 | 6.12 | 6.12 | 5.58 | 5.67 | 8.08 |
| | **(f)** | **15.31** | **7.89** | 5.58 | **4.77** | 4.76 | 5.44 | **7.29** |

| | |
|---|---|
| (a) | No enhancement |
| (b) | STSA-MMSE |
| (c) | NS-DNN |
| **(d)** | **NS-Pix2Pix** |
| (e) | NG-DNN |
| **(f)** | **NG-Pix2Pix** |

1. From the PESQ and STOI evaluations, Pix2Pix outperforms Noisy and MMSE and is competitive to DNN SE.
2. From the speaker verification results, Pix2Pix outperforms the baseline models when the clean training data is used.

# Speech Enhancement

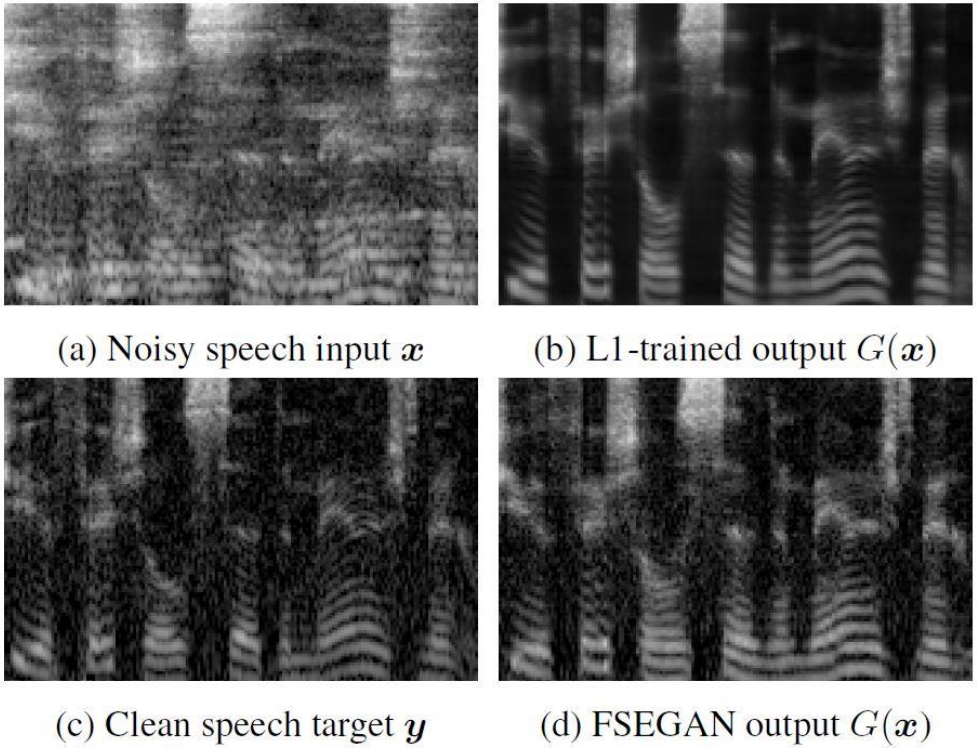
- Frequency-domain SEGAN (FSEGAN) [Donahue et al., ICASSP 2018]



# Speech Enhancement (FSEGAN)

- Spectrogram analysis

Fig. 3: Spectrogram comparison of FSEGAN with L1-trained method.



(a) Noisy speech input $\boldsymbol{x}$ (b) L1-trained output $G(\boldsymbol{x})$

(c) Clean speech target $\boldsymbol{y}$ (d) FSEGAN output $G(\boldsymbol{x})$

FSEGAN reduces both additive noise and reverberant smearing.

# Speech Enhancement (FSEGAN)

- ASR results

Table 5: WER (%) of SEGAN and FSEGAN.

| Test Set | Enhancer | ASR-Clean WER | ASR-MTR WER |
|---|---|---|---|
| Clean | None | 11.9 | 14.3 |
| MTR | None | 72.2 | 20.3 |
| | SEGAN | 80.7 | 52.8 |
| | FSEGAN | 33.3 | 25.4 |

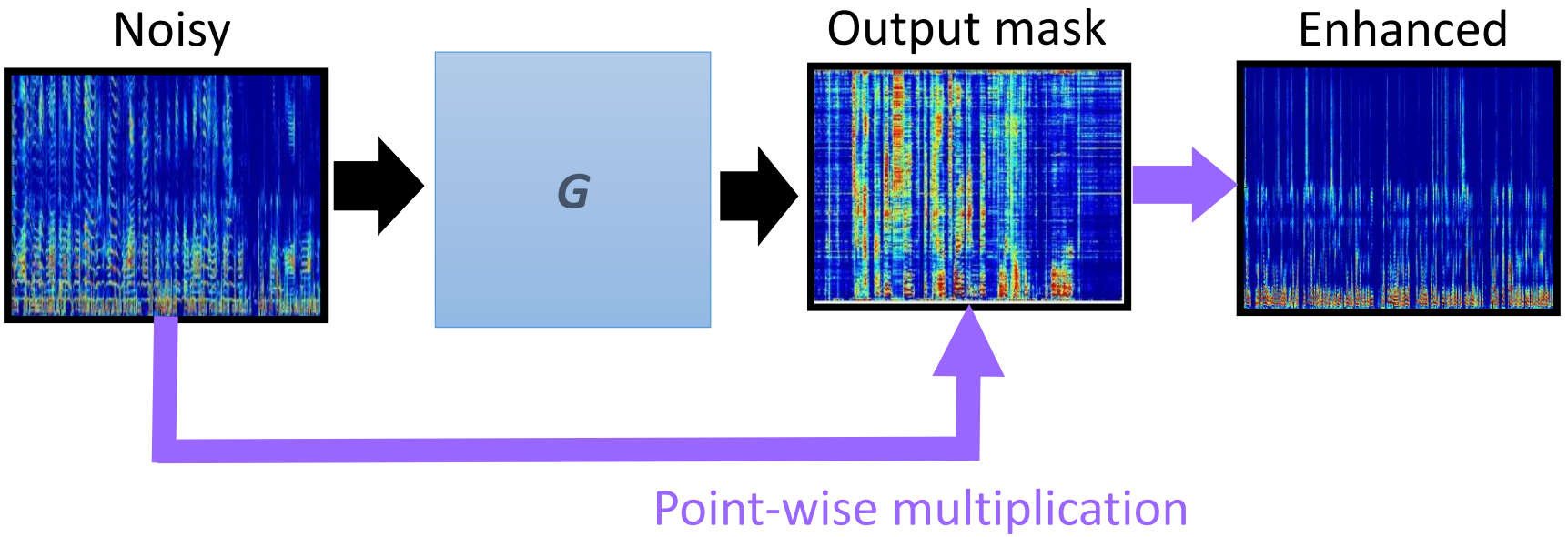
Table 6: WER (%) of FSEGAN with retrain.

| Model | WER (%) |
|---|---|
| MTR Baseline * | 20.3 |
| + Stereo | 19.0 |
| MTR + FSEGAN Enhancer * | 25.4 |
| + Retraining | 21.0 |
| + Hybrid Retraining | 17.6 |
| MTR + L1-trained Enhancer * | 21.4 |
| + Retraining | 18.0 |
| + Hybrid Retraining | 17.1 |

1. From Table 5, (1) FSEGAN improves recognition results for ASR-Clean.
   (2) FSEGAN outperforms SEGAN as front-ends.
2. From Table 6, (1) Hybrid Retraining with FSEGAN outperforms Baseline;
   (2) FSEGAN retraining slightly underperforms L1–based retraining.

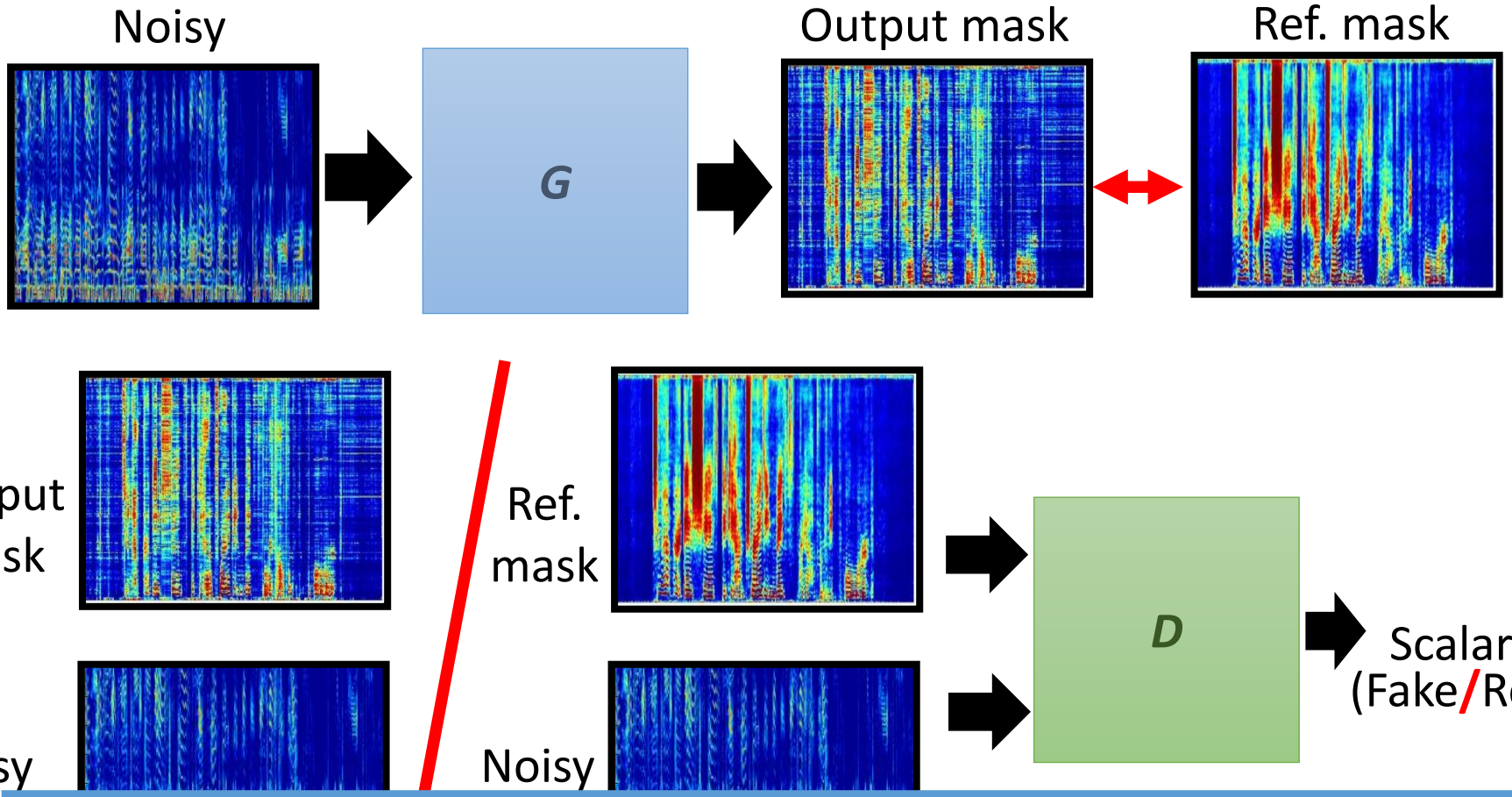
# Speech Enhancement

- Speech enhancement through a mask function



# Speech Enhancement

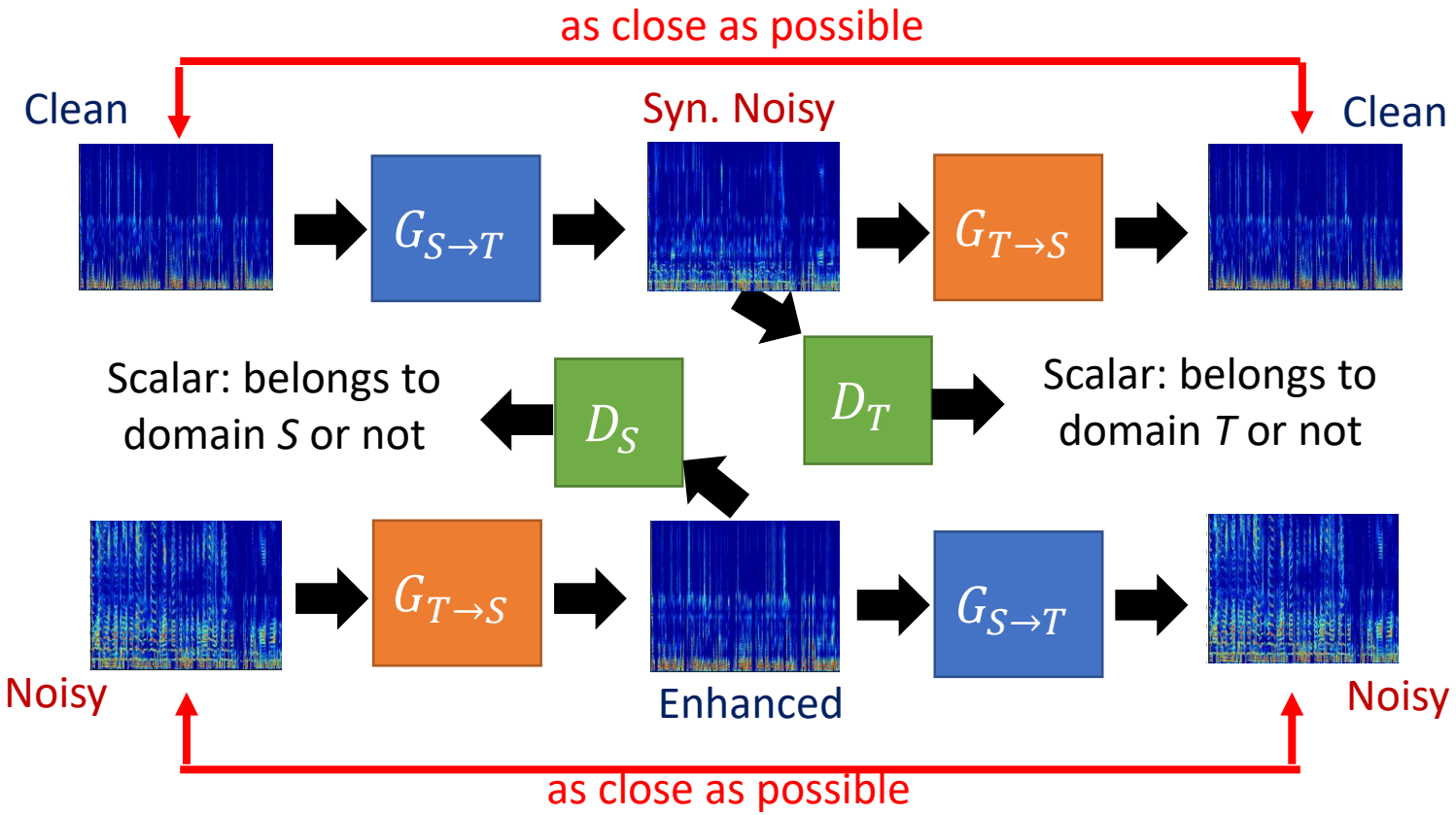
- GAN for spectral magnitude mask estimation (MMS-GAN)
  [Ashutosh Pandey and Deliang Wang, ICASSP 2018]




We don't know exactly what D functions.
Our ICML 2019 paper shed some lights on a potential future direction.

# Speech Enhancement (AFT)

- Cycle-GAN-based acoustic feature transformation (AFT)
  [Mimura et al., ASRU 2017]




$$V_{Full} = V_{GAN}(G_{X\to Y}, D_Y) + V_{GAN}(G_{X\to Y}, D_Y) + \lambda\, V_{Cyc}(G_{X\to Y}, G_{Y\to X})$$

# Speech Enhancement (AFT)

- ASR results on noise robustness and style adaptation

Table 7: Noise robust ASR.

| acoustic model | feature | cycle loss | $\lambda$ and $\mu$ | WER | ID |
|---|---|---|---|---|---|
| no adapt. | no adapt. | - | - | 41.08 | (1) |
| no adapt. | adapt. with $G_{T\to S}$ | no | 1, 1 | 55.45 | (2) |
| | | yes | 1, 1 | 37.34 | (3) |
| | | yes | trained | 36.56 | (4) |
| adapt. with $G_{S\to T}$ | no adapt. | yes | 1, 1 | 35.98 | (5) |
| | | yes | trained | 34.31 | (6) |

$S$: Clean; $T$: Noisy

Table 8: Speaker style adaptation.

| source | target | feature | WER |
|---|---|---|---|
| JNAS | CSJ-SPS | no adapt. | 26.47 |
| | | adapt. with $G_{T\to S}$ | 25.93 |
| CSJ-APS | CSJ-SPS | no adapt. | 17.15 |
| | | adapt. with $G_{T\to S}$ | 16.60 |

JNAS: Read; CSJ-SPS: Spontaneous (relax);
CSJ-APS: Spontaneous (formal);

1. $G_{T\to S}$ can transform acoustic features and effectively improve ASR results for both noisy and accented speech.
2. $G_{S\to T}$ can be used for model adaptation and effectively improve ASR results for noisy speech.

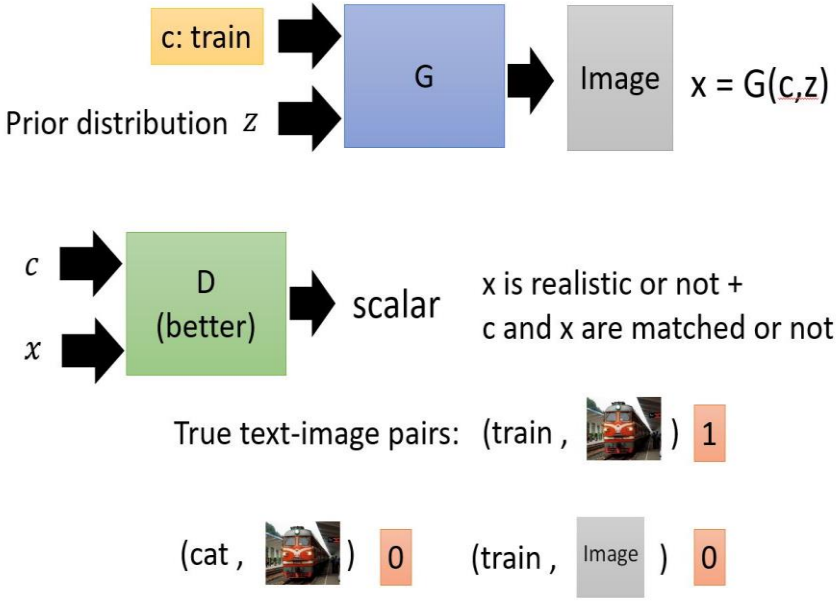
# Outline of Part III

## Speech Signal Generation

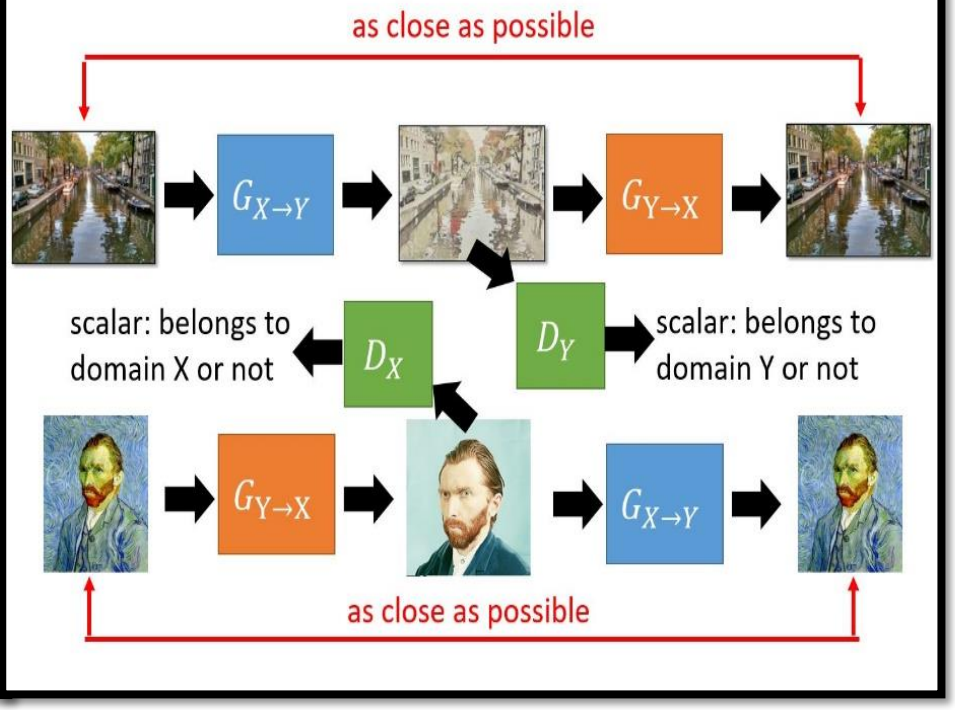
- Speech enhancement
- Postfilter, speech synthesis, voice conversion

[Scott Reed, et al, ICML, 2016]

### Conditional GAN




### Cycle-GAN



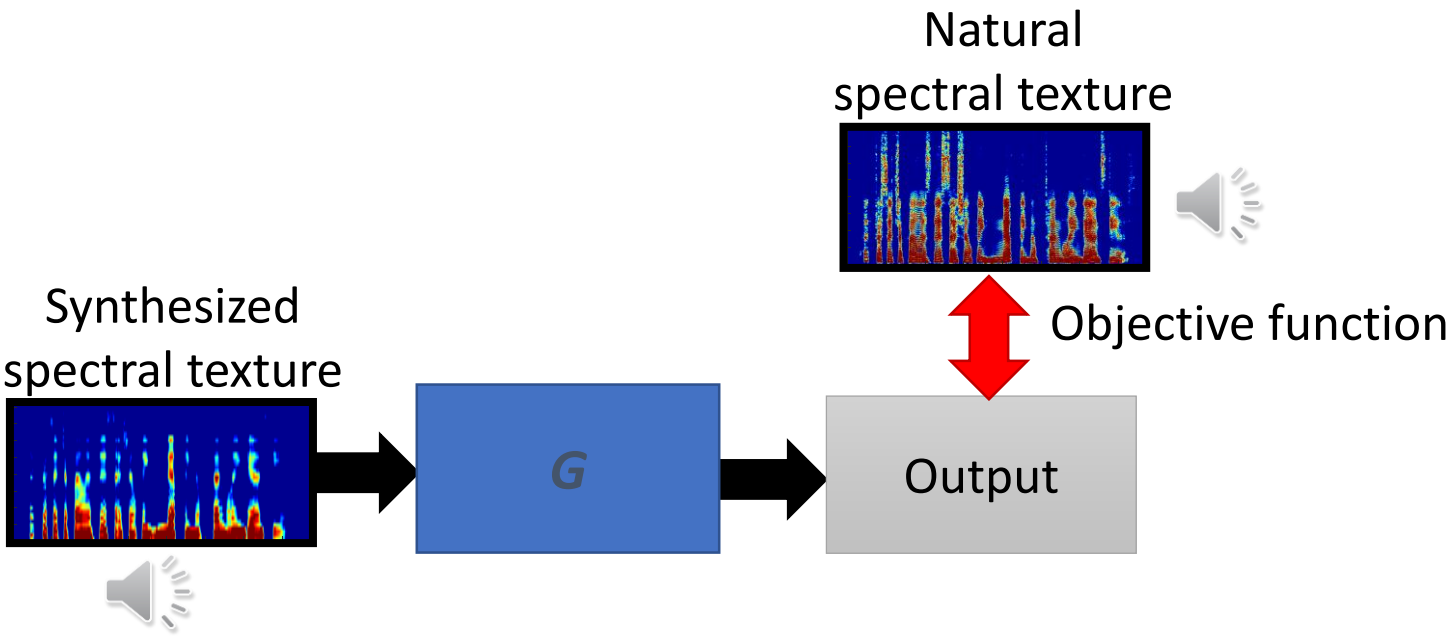
# Postfilter

- Postfilter for synthesized or transformed speech

Speech synthesizer

Voice conversion

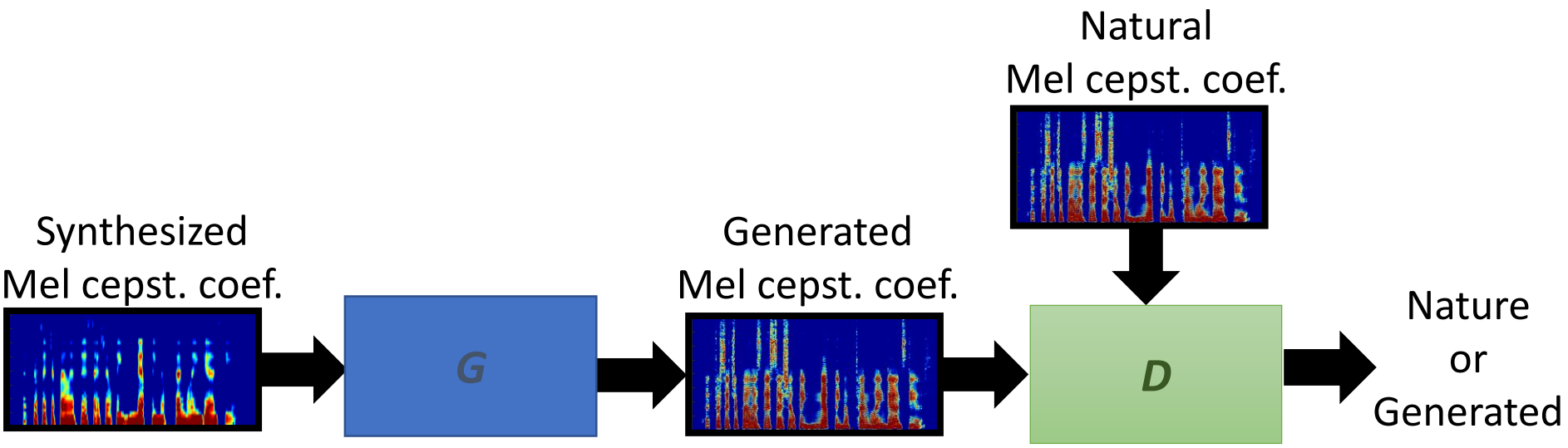
Speech enhancement




- Conventional postfilter approaches for ***G*** estimation include global variance (GV) [Toda et al., IEICE 2007], variance scaling (VS) [Sil'en et al., Interpseech 2012], modulation spectrum (MS) [Takamichi et al., ICASSP 2014],DNN with MSE criterion [Chen et al., Interspeech 2014; Chen et al., TASLP 2015].
- GAN is used a new objective function to estimate the parameters in ***G***.

# Postfilter

- GAN postfilter [Kaneko et al., ICASSP 2017]

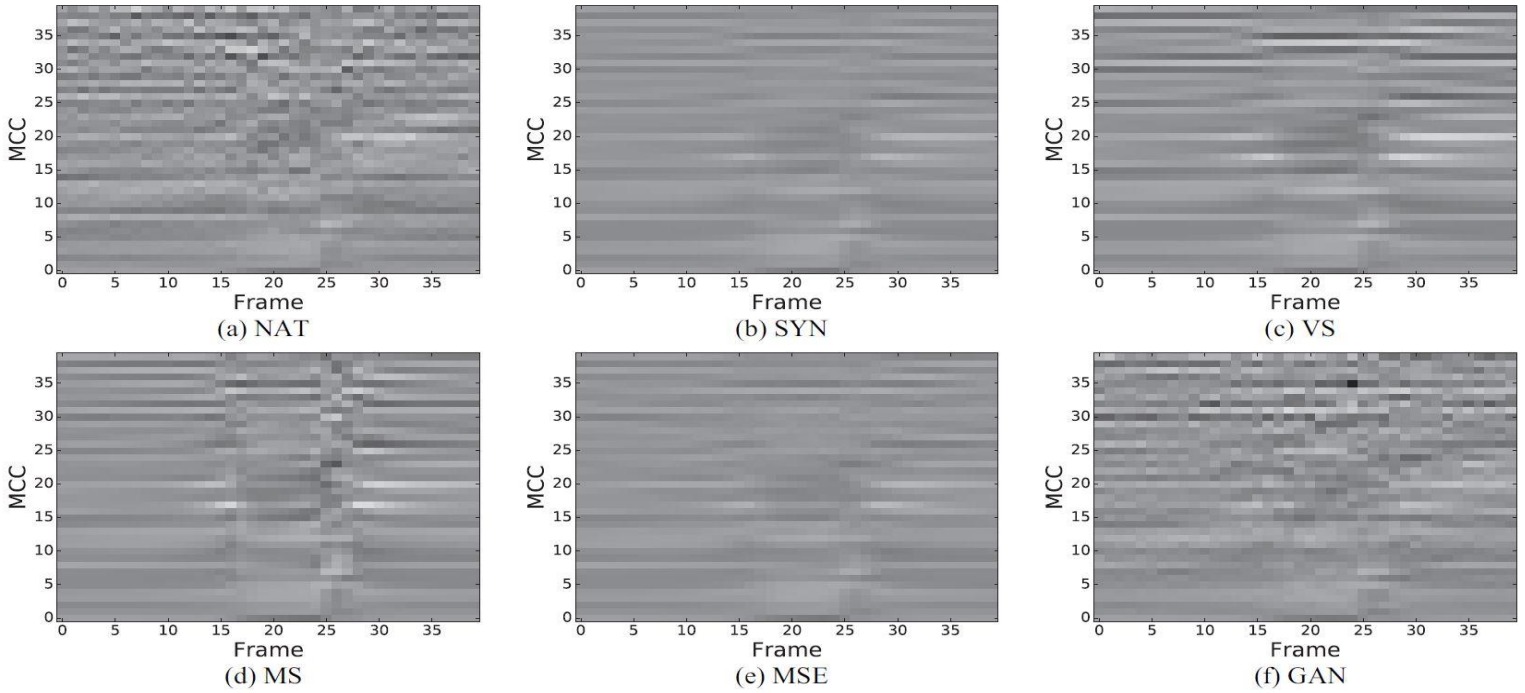



- Traditional MMSE criterion results in statistical averaging.
- GAN is used as a new objective function to estimate the parameters in ***G***.
- The proposed work intends to further improve the naturalness of synthesized speech or parameters from a synthesizer.

# Postfilter (GAN-based Postfilter)

- Spectrogram analysis

Fig. 4: Spectrograms of: (a) NAT (nature); (b) SYN (synthesized); (c) VS (variance scaling); (d) MS (modulation spectrum); (e) MSE; (f) GAN postfilters.




GAN postfilter reconstructs spectral texture similar to the natural one.

# Postfilter (GAN-based Postfilter)

- Objective evaluations

Fig. 5: Mel-cepstral trajectories (GANv: GAN was applied in voiced part).

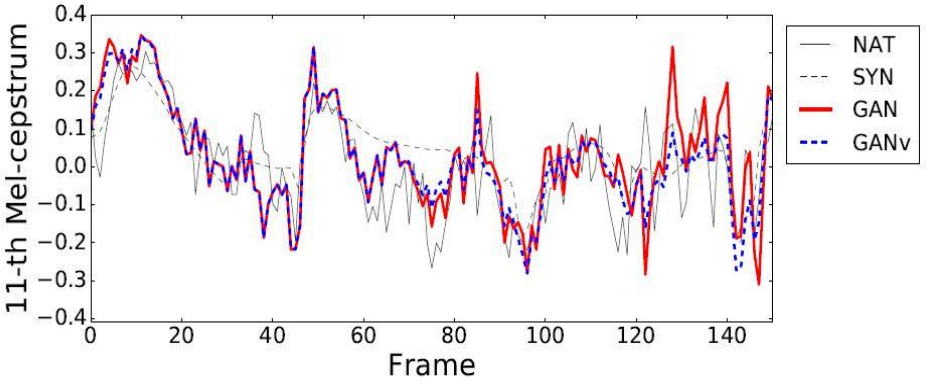
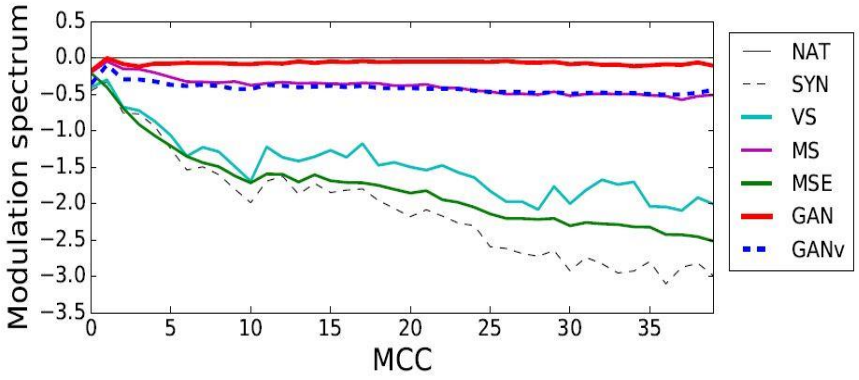



Fig. 6: Averaging difference in modulation spectrum per Mel-cepstral coefficient.




GAN postfilter reconstructs spectral texture similar to the natural one.

# Postfilter (GAN-based Postfilter)

- Subjective evaluations

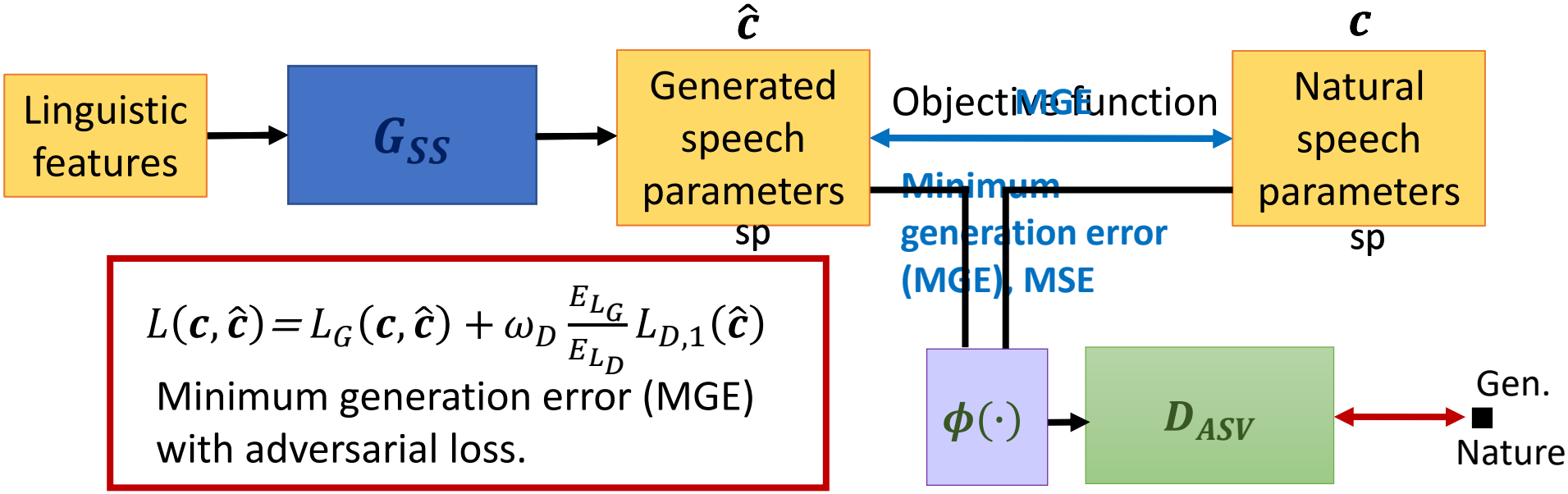
Table 9: Preference score (%). Bold font indicates the numbers over 30%.

| | Former | Latter | Neutral |
|---|---|---|---|
| GAN vs. SYN | **56.5** $\pm$ 4.9 | 22.0 $\pm$ 4.1 | 21.5 $\pm$ 4.0 |
| GAN vs. GANv | 11.3 $\pm$ 3.1 | **37.3** $\pm$ 4.8 | **51.5** $\pm$ 4.9 |
| GAN vs. NAT | 16.8 $\pm$ 3.7 | **53.5** $\pm$ 4.9 | 29.8 $\pm$ 4.5 |
| GANv vs. NAT | **30.3** $\pm$ 4.5 | **34.5** $\pm$ 4.7 | **35.3** $\pm$ 4.7 |

1. GAN postfilter significantly improves the synthesized speech.
2. GAN postfilter is effective particularly in voiced segments.
3. GANv outperforms GAN and is comparable to NAT.

# Speech Synthesis

- Speech synthesis with an anti-spoofing verification module (ASV)
  [Saito et al., ICASSP 2017]




$$L(\boldsymbol{c}, \hat{\boldsymbol{c}}) = L_G(\boldsymbol{c}, \hat{\boldsymbol{c}}) + \omega_D \frac{E_{L_G}}{E_{L_D}} L_{D,1}(\hat{\boldsymbol{c}})$$

Minimum generation error (MGE) with adversarial loss.

$$L_D(\boldsymbol{c}, \hat{\boldsymbol{c}}) = L_{D,1}(\boldsymbol{c}) + L_{D,0}(\hat{\boldsymbol{c}})$$

$$L_{D,1}(\boldsymbol{c}) = -\frac{1}{T}\sum_{t=1}^{T} \log(D(\boldsymbol{c}_t)) \text{...NAT}$$

$$L_{D,0}(\hat{\boldsymbol{c}}) = -\frac{1}{T}\sum_{t=1}^{T} \log(1 - D(\hat{\boldsymbol{c}}_t)) \text{...SYN}$$

# Speech Synthesis (ASV)

- Objective and subjective evaluations

Fig. 7: Averaged GVs of MCCs.

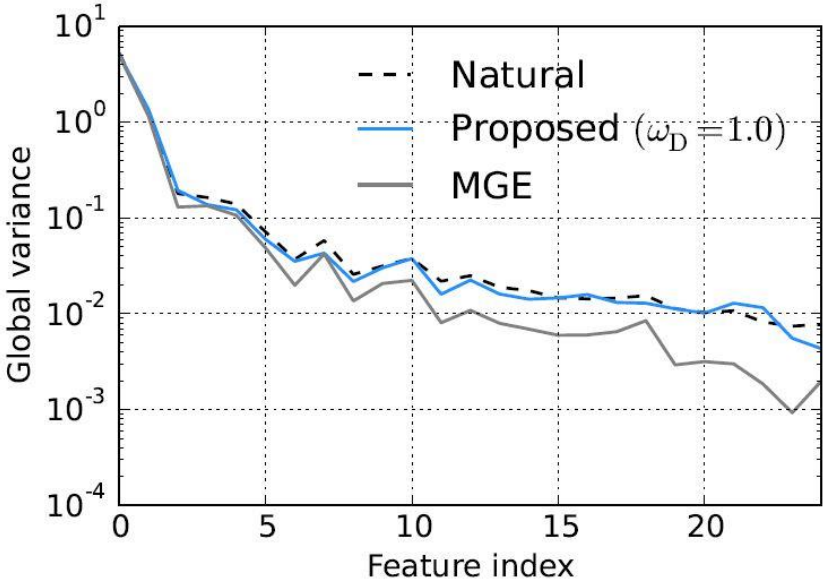
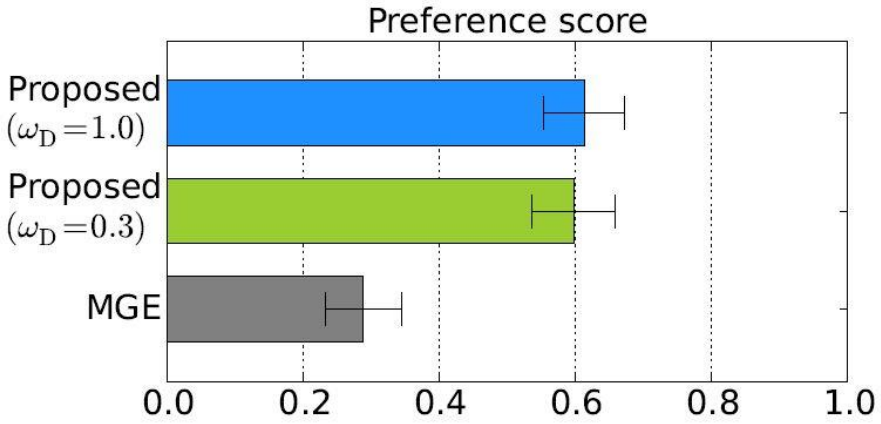


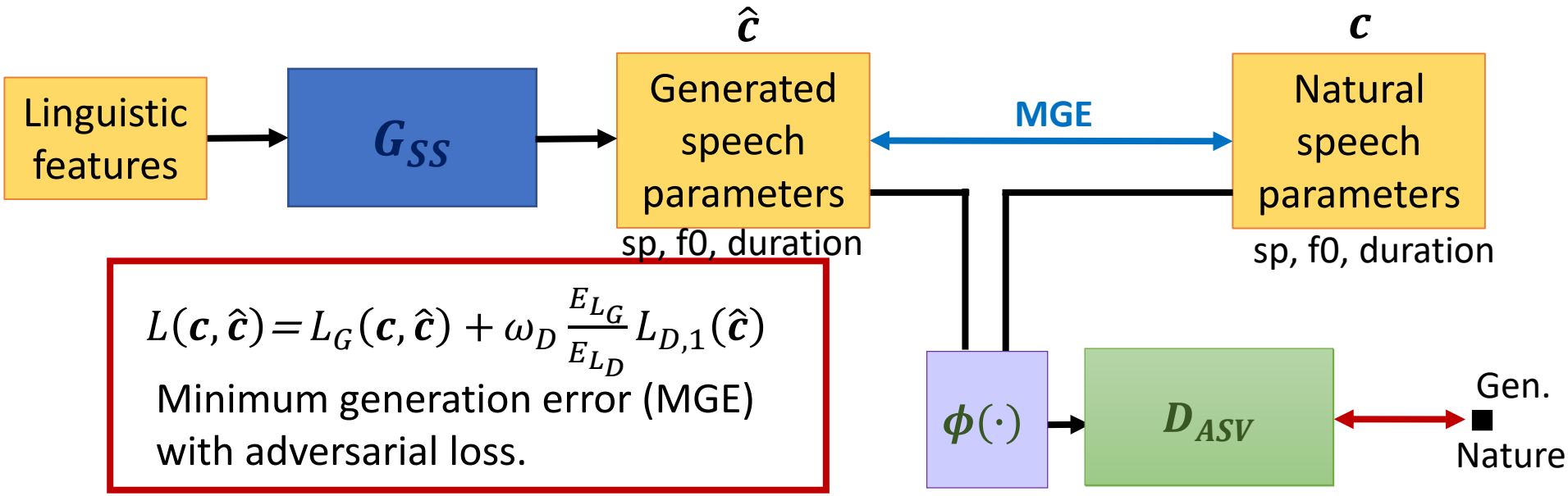

Fig. 8: Scores of speech quality.




1. The proposed algorithm generates MCCs similar to the natural ones.
2. The proposed algorithm outperforms conventional MGE training.

# Speech Synthesis

- Speech synthesis with GAN (SS-GAN) [Saito et al., TASLP 2018]



# Speech Synthesis (SS-GAN)

- Subjective evaluations

Fig. 9: Scores of speech quality (sp).

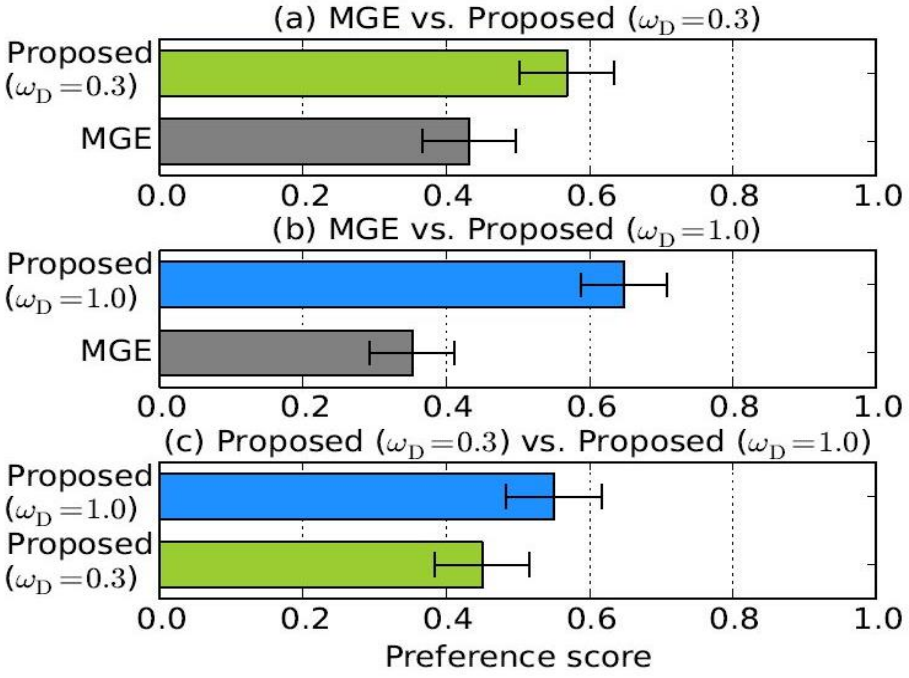
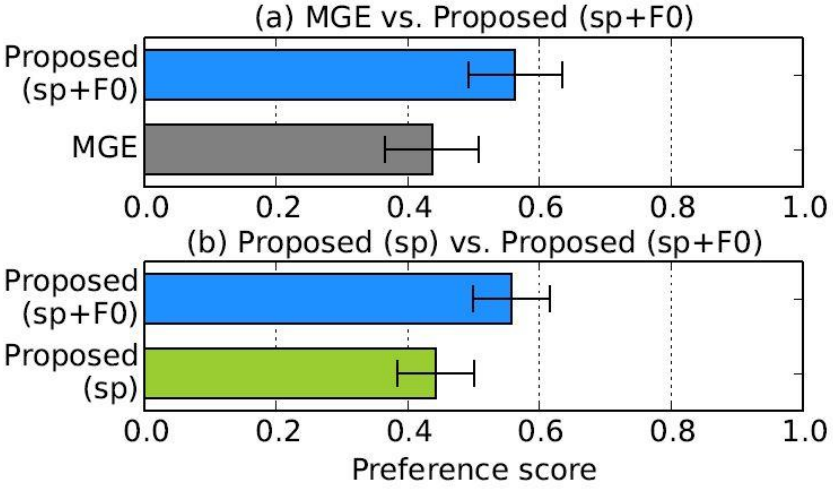


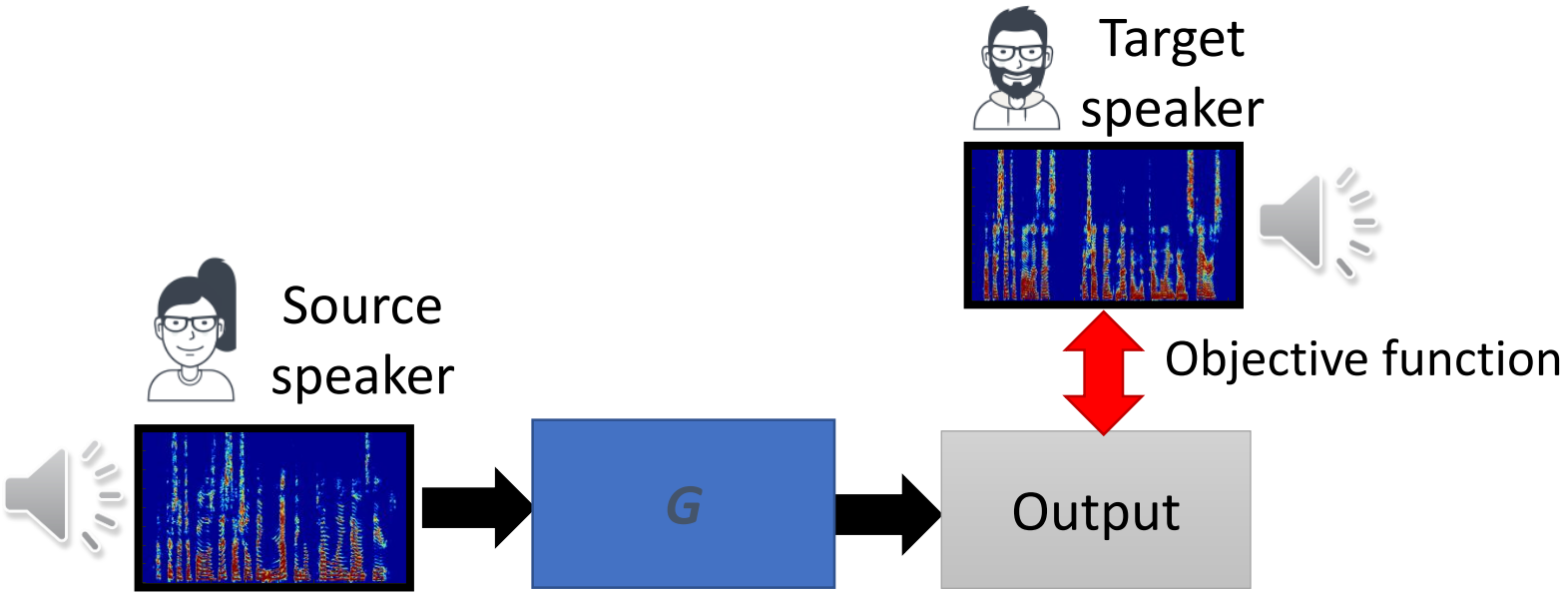

Fig. 10: Scores of speech quality (sp and F0).




The proposed algorithm works for both spectral parameters and F0.

# Voice Conversion

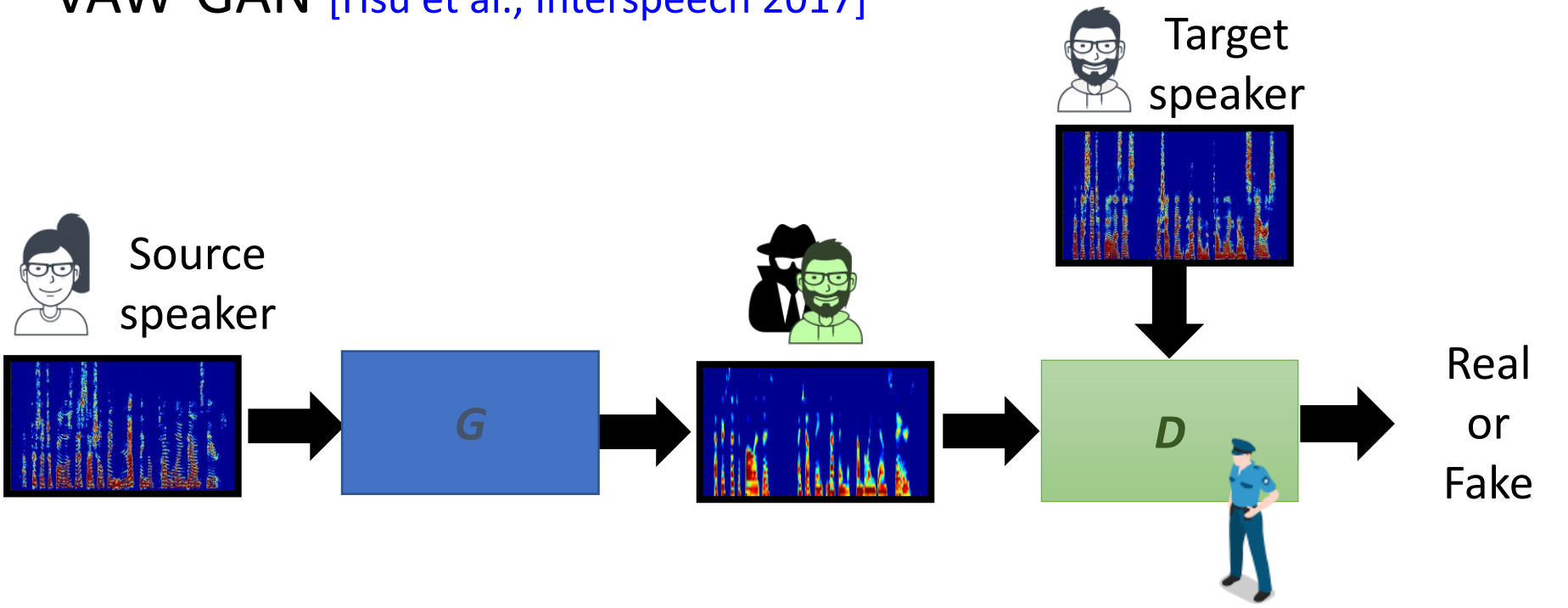
- Convert (transform) speech from source to target




- Conventional VC approaches include Gaussian mixture model (GMM) [Toda et al., TASLP 2007], non-negative matrix factorization (NMF) [Wu et al., TASLP 2014; Fu et al., TBME 2017], locally linear embedding (LLE) [Wu et al., Interspeech 2016], variational autoencoder (VAE) [Hsu et al., APSIPA 2016], restricted Boltzmann machine (RBM) [Chen et al., TASLP 2014], feed forward NN [Desai et al., TASLP 2010], recurrent NN (RNN) [Nakashika et al., Interspeech 2014].

# Voice Conversion

- VAW-GAN [Hsu et al., Interspeech 2017]




- Conventional MMSE approaches often encounter the "over-smoothing" issue.
- GAN is used a new objective function to estimate ***G***.
- The goal is to increase the naturalness, clarity, similarity of converted speech.

$$V(G,D) = V_{GAN}(G,D) + \lambda V_{VAE}(\boldsymbol{x}|\boldsymbol{y})$$

# Voice Conversion (VAW-GAN)

- Objective and subjective evaluations

Fig. 11: The spectral envelopes.

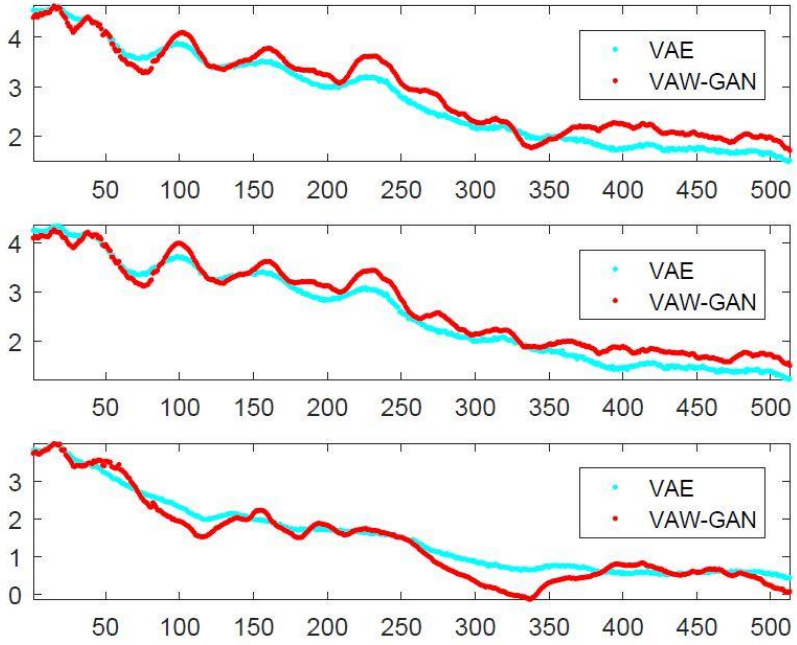
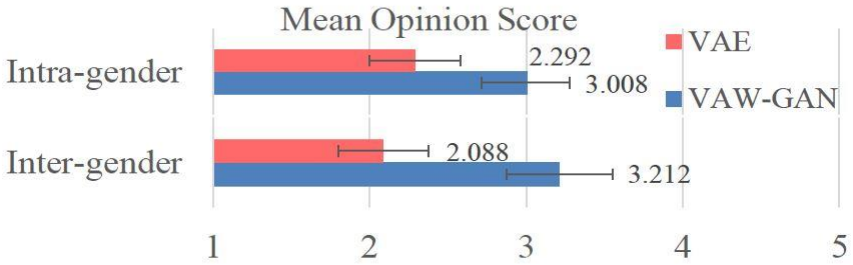


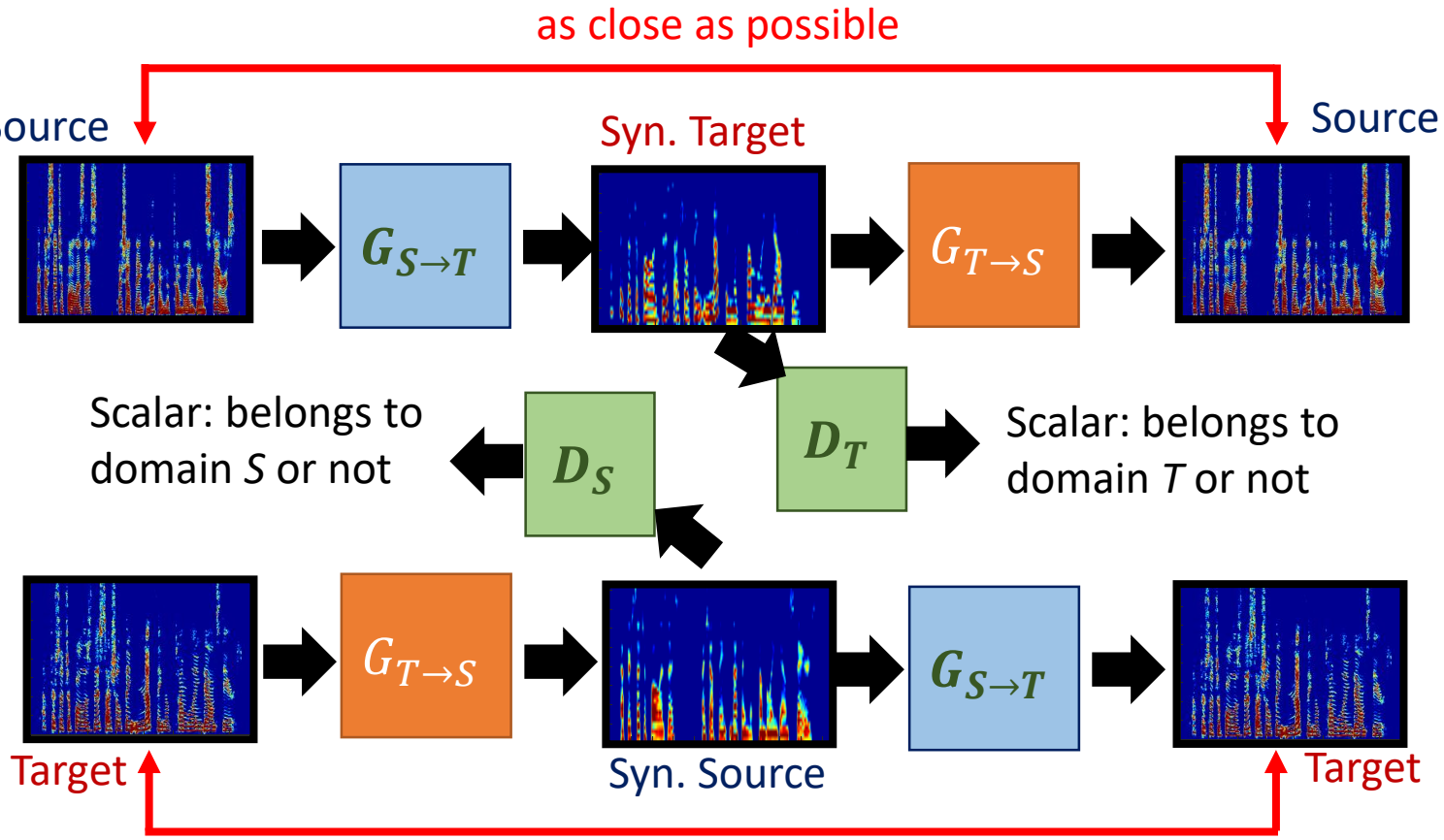

Fig. 12: MOS on naturalness.




VAW-GAN outperforms VAE in terms of objective and subjective evaluations with generating more structured speech.

# Voice Conversion

- CycleGAN-VC [Kaneko et al., Eusipco 2018]




$$V_{Full} = V_{GAN}(G_{X\to Y}, D_Y) + V_{GAN}(G_{X\to Y}, D_Y) + \lambda V_{Cyc}(G_{X\to Y}, G_{Y\to X})$$

# Voice Conversion (CycleGAN-VC)

- Subjective evaluations

Fig. 13: MOS for naturalness.

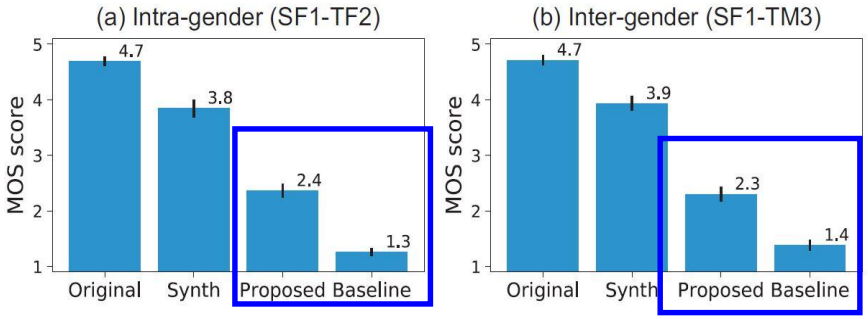
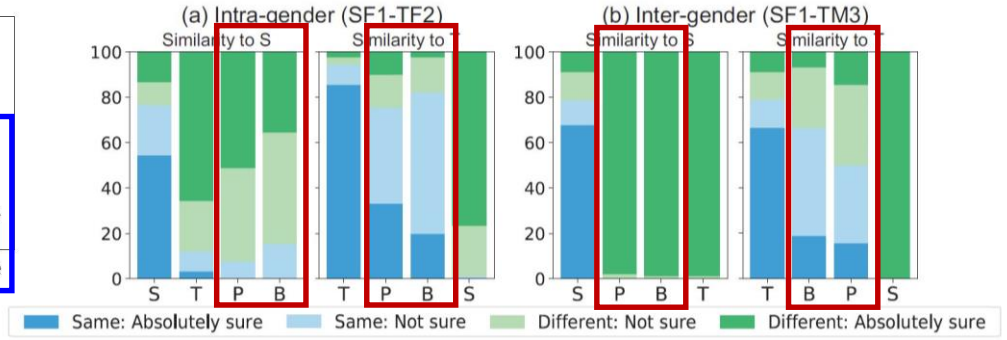



Fig. 14: Similarity of to source and to target speakers. S: Source; T:Target; P: Proposed; B:Baseline





1. The proposed method uses **non-parallel** data.
2. For naturalness, the proposed method outperforms baseline.
3. For similarity, the proposed method is comparable to the baseline.

# Outline of Part III

## Speech Signal Generation

- Speech enhancement
- Postfilter, speech synthesis, voice conversion

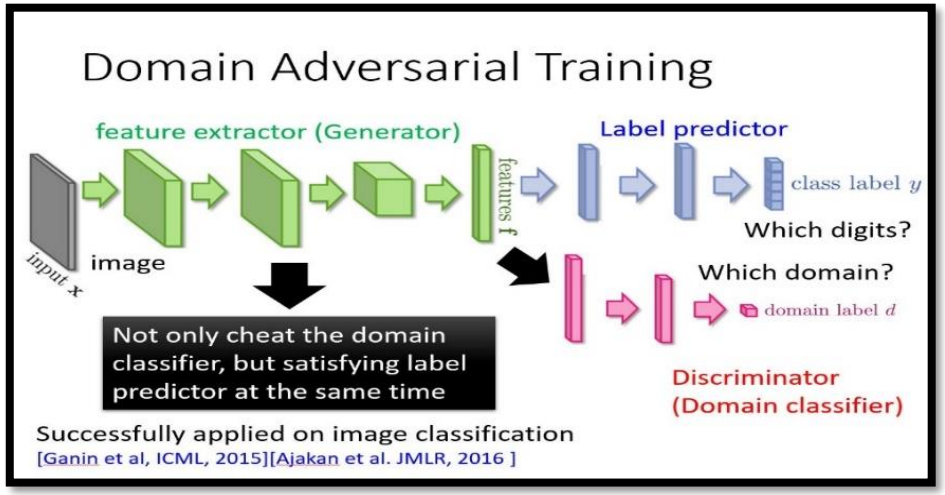
## Speech Signal Recognition

- Speech recognition
- Speaker recognition
- Speech emotion recognition
- Lip reading

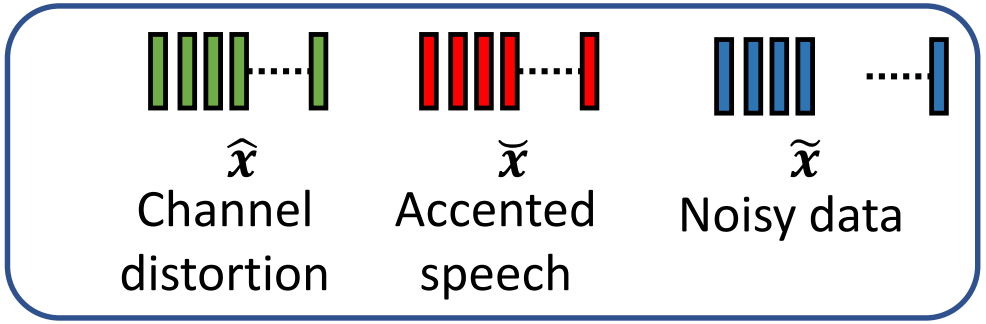
## Conclusion

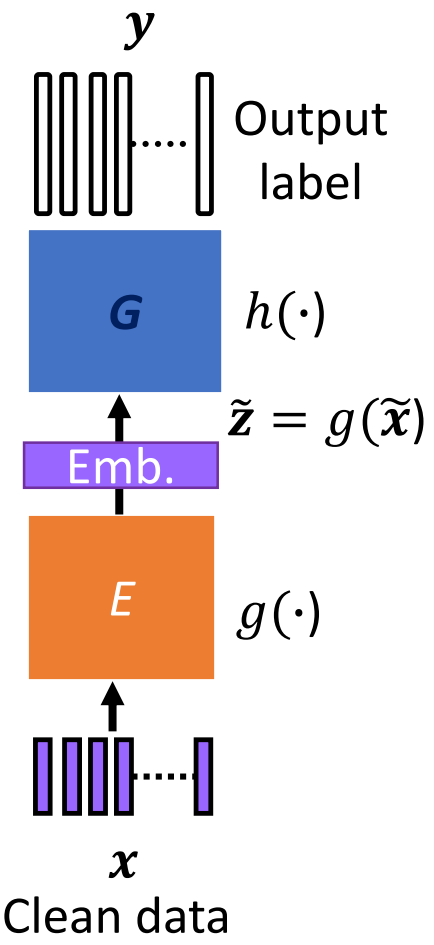
## Our Recent Works

# Speech, Speaker, Emotion Recognition and Lip-reading (Classification Task)



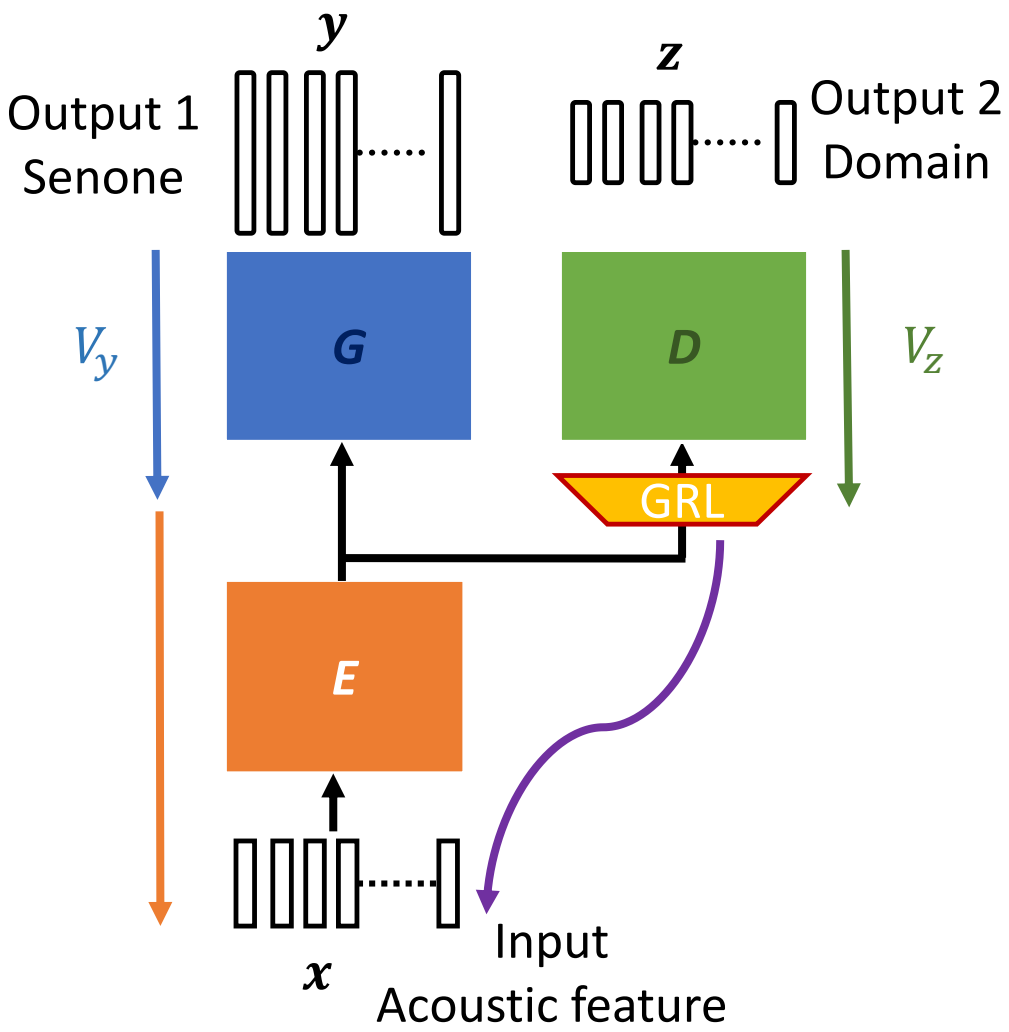

Acoustic Mismatch



# Speech Recognition

- Adversarial multi-task learning (AMT)
  [Shinohara Interspeech 2016]




Objective function

$$V_y = -\sum_i \log P(y_i|x_i;\theta_E,\theta_G)$$

$$V_z = -\sum_i \log P(z_i|x_i;\theta_E,\theta_D)$$

Model update

$$\theta_G \leftarrow \theta_G - \epsilon\frac{\partial V_y}{\partial \theta_G}$$

Max classification accuracy

$$\theta_D \leftarrow \theta_D - \epsilon\frac{\partial V_z}{\partial \theta_D}$$

Max domain accuracy

$$\theta_E \leftarrow \theta_E - \epsilon\left(\frac{\partial V_y}{\partial \theta_E}\right) + \alpha\frac{\partial V_z}{\partial \theta_E}$$

Max classification accuracy and Min domain accuracy

# Speech Recognition (AMT)

- ASR results in known (k) and unknown (unk) noisy conditions

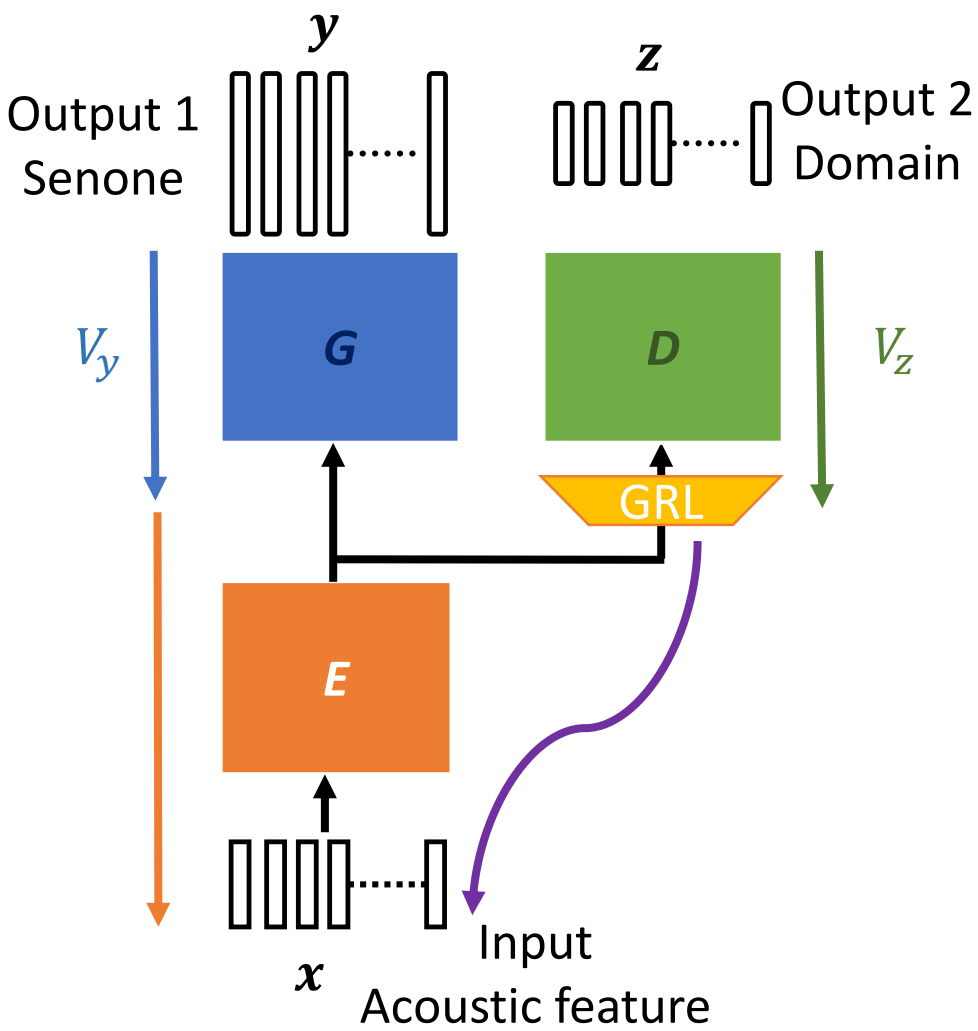
Table 10: WER of DNNs with single-task learning (ST) and AMT.

| | noise | ST | AMT | RERR |
|---|---|---|---|---|
| k | car 2000cc | 5.83 | 5.56 | 4.63 |
| k | exhib. booth | 6.80 | 6.66 | 2.06 |
| k | station | 7.89 | 7.76 | 1.65 |
| k | crossing | 6.96 | 6.65 | 4.45 |
| unk | car 1500cc | 5.58 | 5.46 | 2.15 |
| unk | exhib. aisle | 7.71 | 6.93 | 10.12 |
| unk | factory | 12.17 | 12.92 | -6.16 |
| unk | highway | 9.73 | 9.52 | 2.16 |
| unk | crowd | 6.72 | 6.40 | 4.76 |
| unk | server room | 8.54 | 7.76 | 9.13 |
| unk | air cond. | 6.96 | 6.98 | -0.29 |
| unk | elev. hall | 9.23 | 9.60 | -4.01 |
| - | average | 7.84 | 7.68 | 2.04 |

The AMT-DNN outperforms ST-DNN with yielding lower WERs.

# Speech Recognition

- Domain adversarial training for accented ASR (DAT)
  [Sun et al., ICASSP2018]




Objective function

$$V_y = -\sum_i \log P(y_i|x_i;\theta_E,\theta_G)$$

$$V_z = -\sum_i \log P(z_i|x_i;\theta_E,\theta_D)$$

Model update

$$\theta_G \leftarrow \theta_G - \epsilon\frac{\partial V_y}{\partial \theta_G}$$

Max classification accuracy

$$\theta_D \leftarrow \theta_D - \epsilon\frac{\partial V_z}{\partial \theta_D}$$

Max domain accuracy

$$\theta_E \leftarrow \theta_E - \epsilon\left(\frac{\partial V_y}{\partial \theta_E}\right) + \alpha\frac{\partial V_z}{\partial \theta_E}$$

Max classification accuracy and Min domain accuracy

# Speech Recognition (DAT)

- ASR results on accented speech

Table 11: WER of the baseline and adapted model.

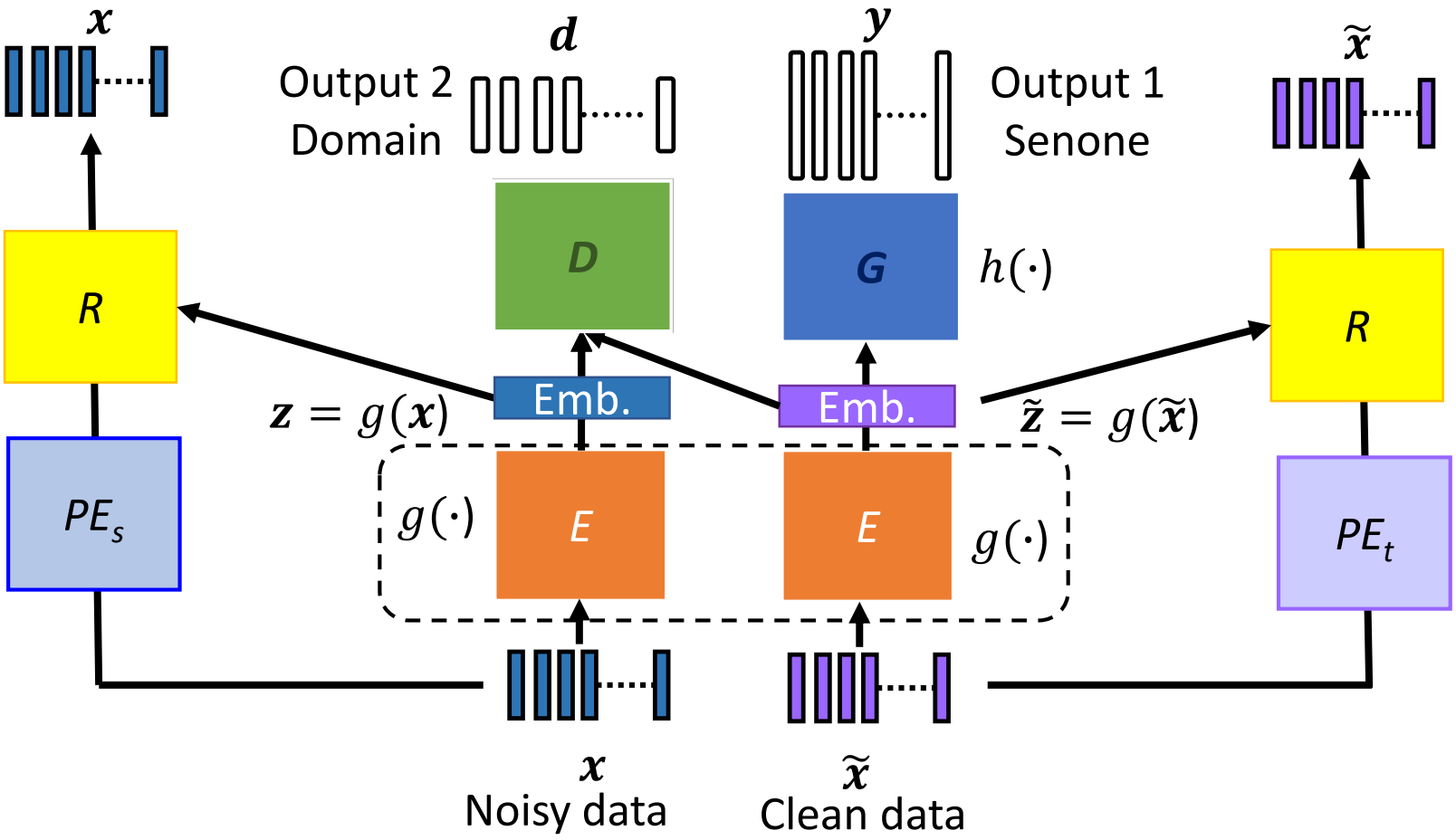
| training data | $\lambda$ | test | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | STD | FJ | JS | JX | SC | GD | HN | Avg. |
| STD | - | 15.55 | 23.58 | 15.75 | 14.08 | 15.62 | 15.32 | 19.34 | 17.28 |
| STD + (600hrs with trans) | - | 14.22 | 14.84 | 9.41 | 8.68 | 9.13 | 9.62 | 11.89 | 10.60 |
| STD + (600hrs no trans) | 0.03 | 15.37 | 22.96 | 14.48 | 13.79 | 15.35 | 14.86 | 18.24 | 16.61 |

STD: standard speech

1. With labeled transcriptions, ASR performance notably improves.
2. DAT is effective in learning features invariant to domain differences with and without labeled transcriptions.

# Speech Recognition

- Unsupervised Adaptation with Domain Separation Networks (DSN) [Meng et al., ASRU 2017]



# Speech Recognition (DSN)

- Results on ASR in noise (CHiME3):

Table 12: WER (in %) of Robust ASR on the CHiME3 task.

| System | Data | BUS | CAF | PED | STR | Avg. |
|---|---|---|---|---|---|---|
| Clean | Real | 36.25 | 31.78 | 22.76 | 27.18 | 29.44 |
| | Simu | 26.89 | 37.74 | 24.38 | 26.76 | 28.94 |
| GRL | Real | 35.93 | 28.24 | 19.58 | 25.16 | 27.16 |
| | Simu | 26.14 | 34.68 | 22.01 | 25.83 | 27.16 |
| DSN | Real | 32.62 | 23.48 | 17.29 | 23.46 | **24.15** |
| | Simu | 23.38 | 30.39 | 19.51 | 22.01 | **23.82** |

1. DSN outperforms GRL consistently over different noise types.
2. The results confirmed the additional gains provided by private component extractors.

# Outline of Part III

## Speech Signal Generation

- Speech enhancement
- Postfilter, speech synthesis, voice conversion

## Speech Signal Recognition

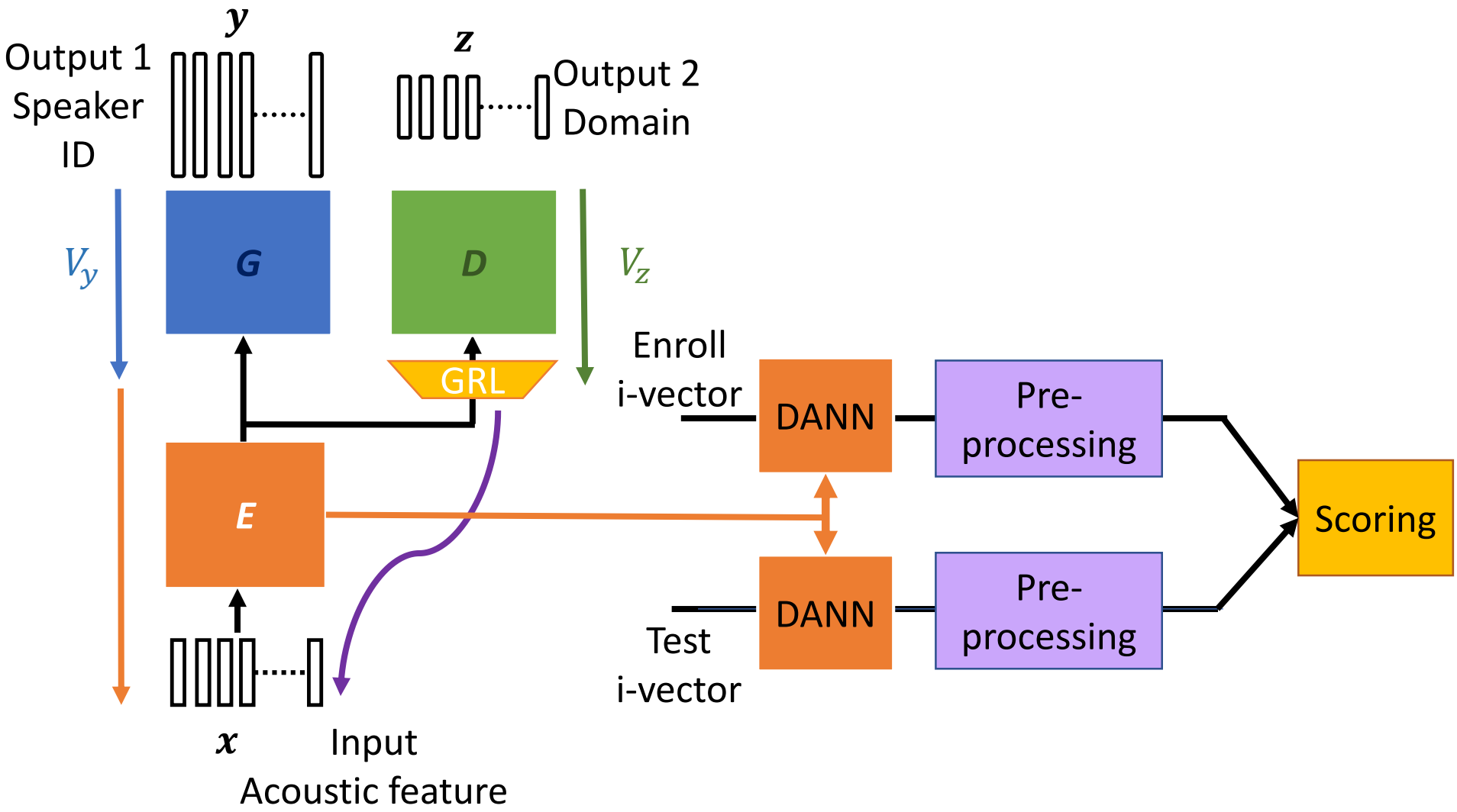
- Speech recognition
- Speaker recognition
- Speech emotion recognition
- Lip reading

## Conclusion

## Our Recent Works

# Speaker Recognition

- Domain adversarial neural network (DANN)
  [Wang et al., ICASSP 2018]



# Speaker Recognition (DANN)

- Recognition results of domain mismatched conditions

Table 13: Performance of DAT and the state-of-the-art methods.

| Systems# | Adaptation Methods | EER% | DCF10 [21] | DCF08 |
|---|---|---|---|---|
| 1 | – | 9.35 | 0.724 | 0.520 |
| 2 | – | 5.66 | 0.633 | 0.427 |
| 3 | Interpolated [6] [12] | 6.55 | 0.652 | 0.454 |
| 4 | IDV [9] [12] | 6.15 | 0.676 | 0.476 |
| 5 | DICN [11] [12] | 4.99 | 0.623 | 0.416 |
| 6 | DAE [22] [12] | 4.81 | 0.610 | 0.398 |
| 7 | AEDA [12] | 4.50 | 0.589 | 0.362 |
| **8** | **DAT** | **3.73** | **0.541** | **0.335** |

The DAT approach outperforms other methods with achieving lowest EER and DCF scores.

# Outline of Part III

## Speech Signal Generation

- Speech enhancement
- Postfilter, speech synthesis, voice conversion

## Speech Signal Recognition

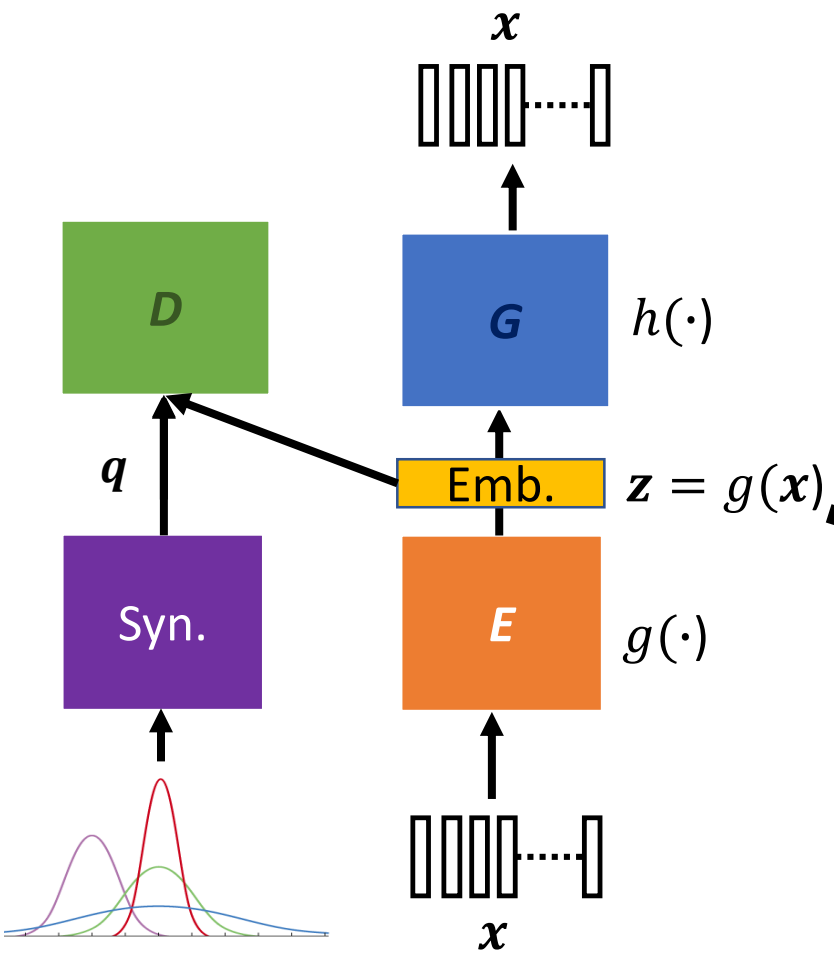
- Speech recognition
- Speaker recognition
- Speech emotion recognition
- Lip reading

## Conclusion

## Our Recent Works

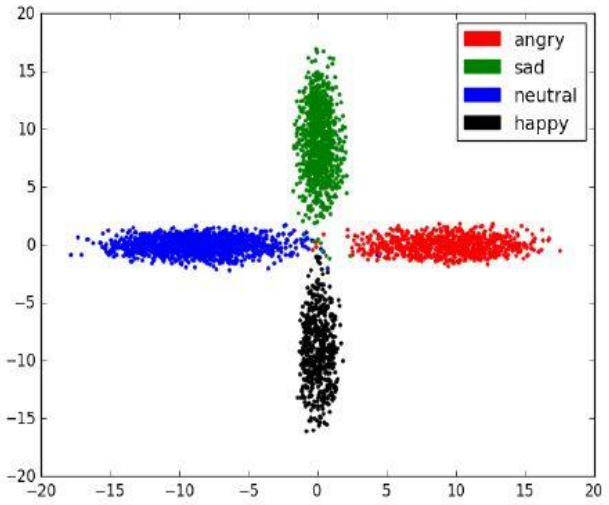
# Emotion Recognition

- Adversarial AE for emotion recognition (AAE-ER)
  [Sahu et al., Interspeech 2017]




AE with GAN :

$$H(h(\boldsymbol{z}), \boldsymbol{x}) + \lambda V_{GAN}(\boldsymbol{q}, g(\boldsymbol{x}))$$




The distribution of code vectors

# Emotion Recognition (AAE-ER)

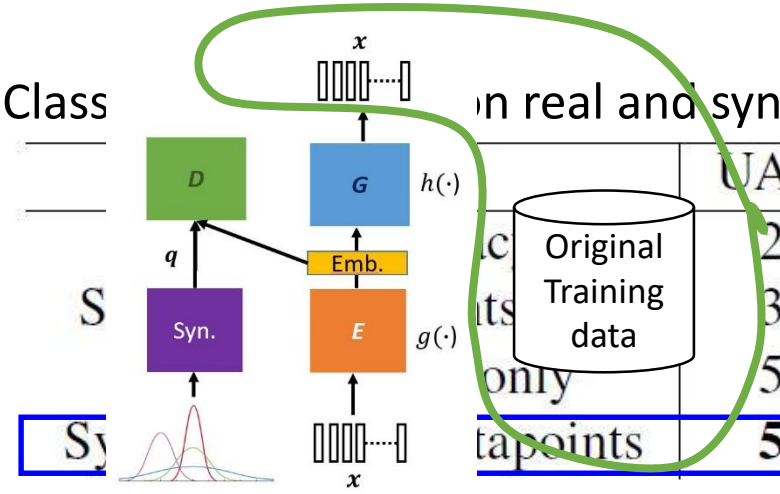
- Recognition results of domain mismatched conditions:

Table 14: Classification results on different systems.

| | OpenSmile features (1582-D) | Code vectors (2-D) | Auto-encoder (100-D) | LDA (2-D) | PCA (2-D) |
|---|---|---|---|---|---|
| UAR (%) | 57.88 | 56.38 | 53.92 | 48.67 | 43.12 |

Table 15: Class … n real and synthesized features.

| | UAR (%) |
|---|---|
| … | 25.00 |
| S… | 33.75 |
| …only | 57.88 |
| Sy… tapoints | **58.38** |




1. AAE alone could not yield performance improvements.
2. Using synthetic data from AAE can yield higher UAR.

# Outline of Part III

## Speech Signal Generation

- Speech enhancement
- Postfilter, speech synthesis, voice conversion

## Speech Signal Recognition

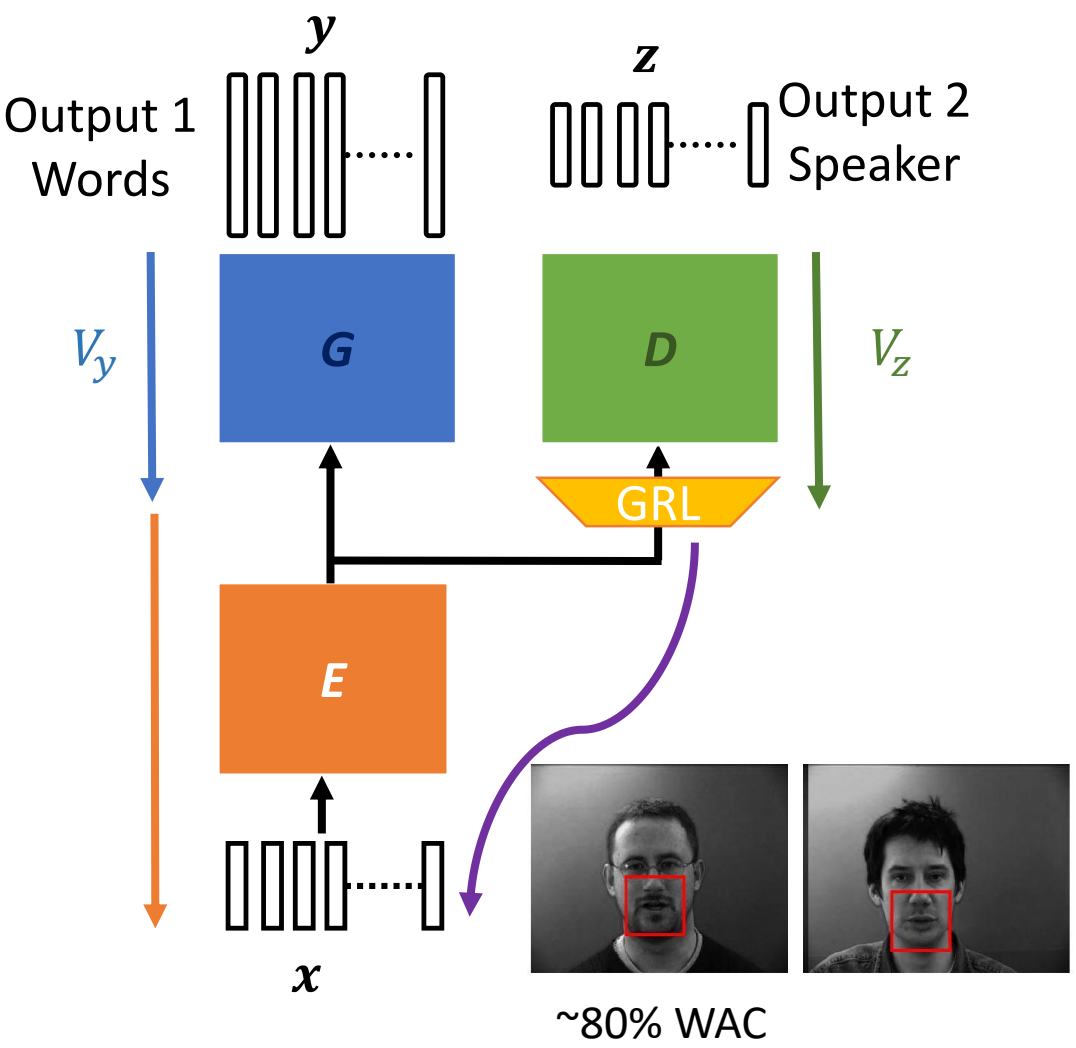
- Speech recognition
- Speaker recognition
- Speech emotion recognition
- Lip reading

## Conclusion

## Our Recent Works

# Lip-reading

- Domain adversarial training for lip-reading (DAT-LR)
  [Wand et al., Interspeech 2017]




~80% WAC

Objective function

$$V_y = -\sum_i \log P(y_i|x_i;\theta_E,\theta_G)$$

$$V_z = -\sum_i \log P(z_i|x_i;\theta_E,\theta_D)$$

Model update

$$\theta_G \leftarrow \theta_G - \epsilon\frac{\partial V_y}{\partial \theta_G}$$

Max classification accuracy

$$\theta_D \leftarrow \theta_D - \epsilon\frac{\partial V_z}{\partial \theta_D}$$

Max domain accuracy

$$\theta_E \leftarrow \theta_E - \epsilon\left(\frac{\partial V_y}{\partial \theta_E}\right) + \alpha\frac{\partial V_z}{\partial \theta_E}$$

Max classification accuracy and Min domain accuracy

# Lip-reading (DAT-LR)

- Recognition results of speaker mismatched conditions

Table 16: Performance of DAT and the baseline.

| Adversarial Training on | Number of training spk | Target Test acc. | Relative Improvement | p-value |
|---|---|---|---|---|
| None | 1 | 18.7% | - | - |
| | 4 | 39.4% | - | - |
| | 8 | 46.5% | - | - |
| All Target Sequences | 1 | 25.4% | 35.8% | 0.0030* |
| | 4 | 43.6% | 10.7% | 0.0261* |
| | 8 | 49.3% | 6.0% | 0.0266* |
| 50 Target Sequences | 1 | 24.1% | 28.9% | 0.0045* |
| | 4 | 41.5% | 5.3% | 0.1367 |
| | 8 | 47.0% | 1.1% | 0.3555 |

The DAT approach notably enhances the recognition accuracies in different conditions.

# Outline of Part III

## Speech Signal Generation

- Speech enhancement
- Postfilter, speech synthesis, voice conversion

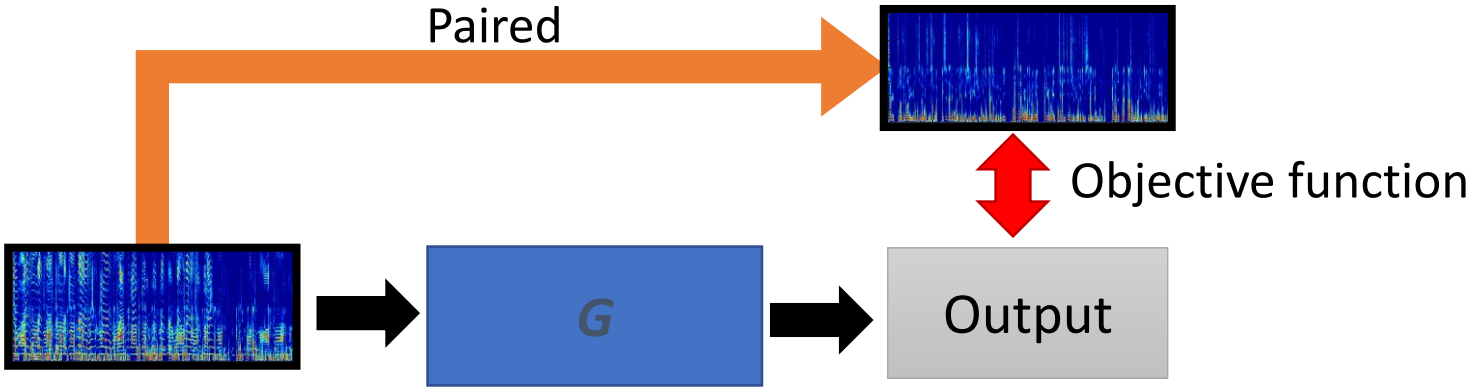
## Speech Signal Recognition

- Speech recognition
- Speaker recognition
- Speech emotion recognition
- Lip reading

## Conclusion

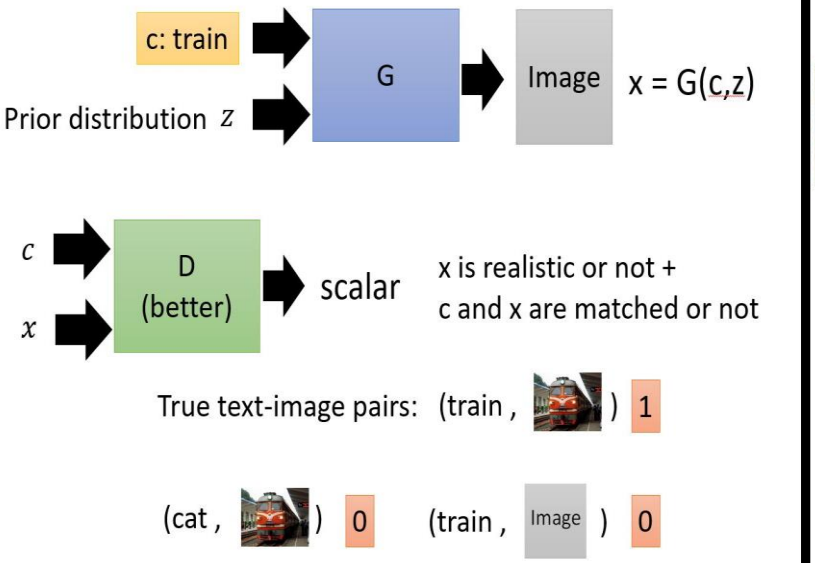
## Our Recent Works

# Speech Signal Generation (Regression Task)

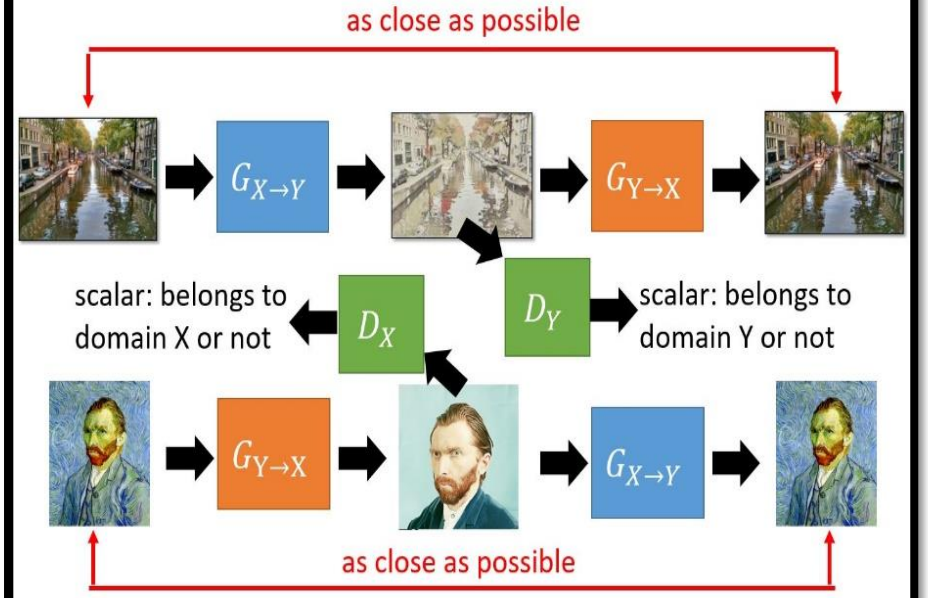



[Scott Reed, et al, ICML, 2016]

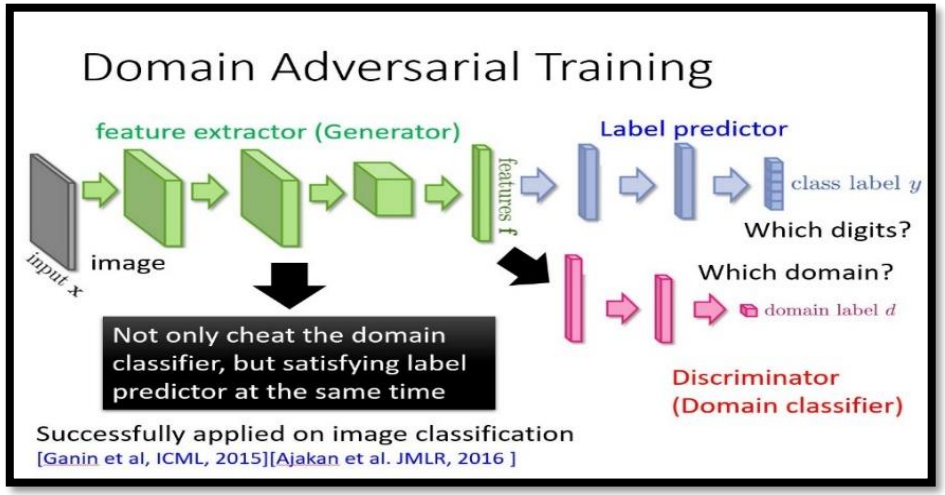
## Conditional GAN



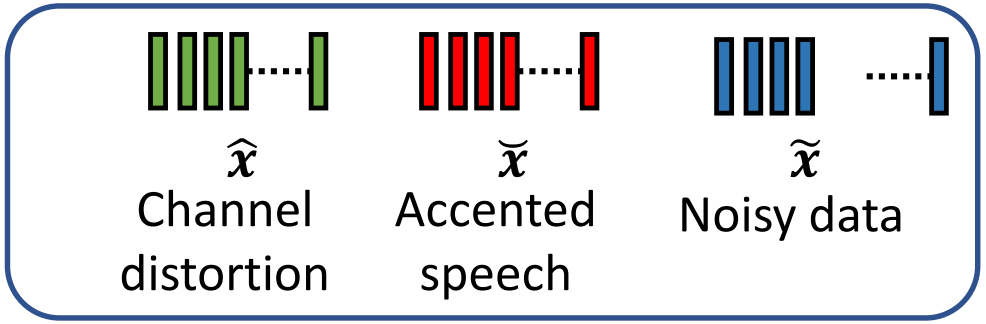

## Cycle-GAN

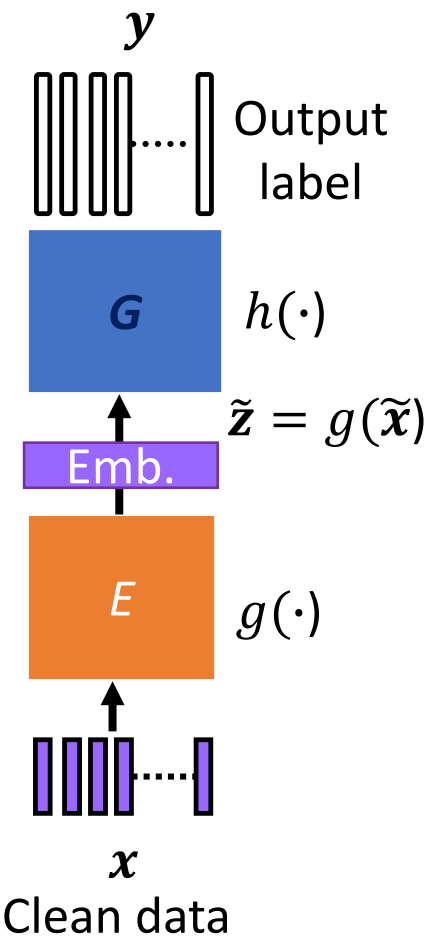


# Speech, Speaker, Emotion Recognition and Lip-reading (Classification Task)




Acoustic Mismatch



# References


**Speech enhancement (conventional methods)**

- Y.-X. Wang and D.-L. Wang, Cocktail party processing via structured prediction, NIPS 2012.
- Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, An experimental study on speech enhancement based on deep neural networks, IEEE SPL, 2014.
- Y. Xu, J. Du, L.-R. Dai, and Chin-Hui Lee, A regression approach to speech enhancement based on deep neural networks, IEEE/ACM TASLP, 2015.
- X. Lu, Y. Tsao, S. Matsuda, H. Chiroi, Speech enhancement based on deep denoising autoencoder, Interspeech 2012.
- Z. Chen, S. Watanabe, H. Erdogan, J. R. Hershey, Integration of speech enhancement and recognition using long-short term memory recurrent neural network, Interspeech 2015.
- F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. L. Roux, J. R. Hershey, and B. Schuller, Speech enhancement with LSTM recurrent neural networks and Its application to noise-robust ASR, LVA/ICA, 2015.
- S.-W. Fu, Y. Tsao, and X.-G. Lu, SNR-aware convolutional neural network modeling for speech enhancement, Interspeech, 2016.
- S.-W. Fu, Y. Tsao, X.-G. Lu, and Hisashi Kawai, End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks, IEEE/ACM TASLP, 2018.

**Speech enhancement (GAN-based methods)**

- P. Santiago, B. Antonio, and S. Joan, SEGAN: Speech enhancement generative adversarial network, Interspeech, 2017.
- D. Michelsanti, and Z.-H. Tan, Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification, Interspeech, 2017.
- C. Donahue, B. Li, and P. Rohit, Exploring speech enhancement with generative adversarial networks for robust speech recognition, ICASSP, 2018.
- T. Higuchi Takuya, K. Kinoshita, D. Marc, and T. Nakatani. Adversarial training for data-driven speech enhancement without parallel Corpus, ASRU, 2017.
- S. Pascual, M. Park, J. Serrà, A. Bonafonte, K.-H. Ahn, Language and noise transfer in speech enhancement generative adversarial network, ICASSP 2018.

# References

**Speech enhancement (GAN-based methods)**


- A. Pandey and D. Wang, On adversarial training and loss functions for speech enhancement, ICASSP 2018.
- M. H. Soni, Neil Shah, and H. A. Patil, Time-frequency masking-based speech enhancement using generative adversarial network, ICASSP 2018.
- Z. Meng, J.-Y. Li, Y.-G. Gong, B.-H. Juang, Adversarial feature-mapping for speech enhancemen, Interspeech, 2018.
- L.-W. Chen, M.Yu, Y.-M. Qian, D. Su, D. Yu, Permutation invariant training of generative adversarial network for monaural speech separation, Interspeech 2018.
- D. Baby and S. Verhulst, Sergan: Speech enhancement using relativistic generative adversarial networks with gradient penalty, ICASSP 2019.

# References


**Postfilter (conventional methods)**

- T. Tod, and K. Tokuda, A speech parameter generation algorithm considering global variance for HMM-based speech synthesis, IEICE Trans. Inf. Syst., 2007.
- H. Sil'en, E. Helander, J. Nurminen, and M. Gabbouj, Ways to implement global variance in statistical speech synthesis, Interspeech, 2012.
- S. Takamichi, T. Toda, N. Graham, S. Sakriani, and S. Nakamura, A postfilter to modify the modulation spectrum in HMM-based speech synthesis, ICASSP, 2014.
- L.-H. Chen, T. Raitio, C. V. Botinhao, J. Yamagishi, and Z.-H. Ling, DNN-based stochastic postfilter for HMM-based speech synthesis, Interspeech, 2014.
- L.-H. Chen, T. Raitio, C. V. Botinhao, Z.-H. Ling, and J. Yamagishi, A deep generative architecture for postfiltering in statistical parametric speech synthesis, IEEE/ACM TASLP, 2015.

**Postfilter (GAN-based methods)**

- K. Takuhiro, K. Hirokazu, H. Nobukatsu, Y. Ijima, K. Hiramatsu, and K. Kashino, Generative adversarial network-based postfilter for statistical parametric speech synthesis, ICASSP, 2017.
- K. Takuhiro, T. Shinji, K. Hirokazu, and J. Yamagishi, Generative adversarial network-based postfilter for STFT spectrograms, Interspeech, 2017.
- Y. Saito, S. Takamichi, and H. Saruwatari, Training algorithm to deceive anti-spoofing verification for DNN-based speech synthesis, ICASSP, 2017.
- Y. Saito, S. Takamichi, H. Saruwatari, Statistical parametric speech synthesis incorporating generative adversarial networks, IEEE/ACM TASLP, 2018.
- B. Bollepalli, L. Juvela, and A. Paavo, Generative adversarial network-based glottal waveform model for statistical parametric speech synthesis, Interspeech, 2017.
- S. Yang, L. Xie, X. Chen, X.-Y. Lou, X. Zhu, D.-Y. Huang, and H.-Z. Li, Statistical parametric speech synthesis using generative adversarial networks under a multi-task learning framework, ASRU, 2017.

# References


**VC (conventional methods)**

- T. Toda, A. W. Black, and K. Tokuda, Voice conversion based on maximum likelihood estimation of spectral parameter trajectory, IEEE/ACM TASLP, 2007.
- L.-H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, Voice conversion using deep neural networks with layer-wise generative training, IEEE/ACM TASLP, 2014.
- S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, Spectral mapping using artificial neural networks for voice conversion, IEEE/ACM TASLP, 2010.
- T. Nakashika, T. Takiguchi, Y. Ariki, High-order sequence modeling using speaker-dependent recurrent temporal restricted boltzmann machines for voice conversion, Interspeech, 2014.
- K. Takuhiro, K. Hirokazu, H. Kaoru, and K. Kunio, Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks, Interspeech, 2017.
- Z.-Z. Wu, T. Virtanen, E.-S. Chng, and H.-Z. Li, Exemplar-based sparse representation with residual compensation for voice conversion, IEEE/ACM TASLP, 2014.
- S.-. Fu, P.-C. Li, Y.-H. Lai, C.-C. Yang, L.-C. Hsieh, and Y. Tsao, Joint dictionary learning-based non-negative matrix factorization for voice conversion to improve speech intelligibility after oral surgery, IEEE TBME, 2017.
- Y.-C. Wu, H.-T. Hwang, C.-C. Hsu, Y. Tsao, and H.-M. Wang, Locally linear embedding for exemplar-based spectral conversion, Interspeech, 2016.
- C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, Y., and H.-M. Wang, Voice conversion from non-parallel corpora using variational auto-encoder. APSIPA 2016.

**VC (GAN-based methods)**

- C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks, Interspeech 2017.
- K. Takuhiro, K. Hirokazu, H. Kaoru, and K. Kunio, Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks, Interspeech, 2017.

# References


**VC (GAN-based methods)**

- K. Takuhiro, and K. Hirokazu. Parallel-data-free voice conversion using cycle-consistent adversarial networks, arXiv, 2017.
- N. Shah, N. J. Shah, and H. A. Patil, Effectiveness of generative adversarial network for non-audible murmur-to-whisper speech conversion, Interspeech, 2018.
- J.-C. Chou, C.-C. Yeh, H.-Y. Lee, and L.-S. Lee, Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations, Interspeech, 2018.
- G. Degottex, and M. Gales, A spectrally weighted mixture of least square error and wasserstein discriminator loss for generative SPSS, SLT, 2018.
- B. Sisman, M. Zhang, S. Sakti, H. Li, and S. Nakamura, Adaptive wavenet vocoder for residual compensation in GAN-based voice conversion, SLT, 2018.
- C.-C. Yeh, P.-C. Hsu, J.-C. Chou, H.-Y. Lee, and L.-S. Lee, Rhythm-flexible voice conversion without parallel data using cycle-GAN over phoneme posteriorgram sequences, SLT, 2018.
- H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, STARGAN-VC: Non-parallel many-to-many voice conversion with star generative adversarial networks, SLT, 2018.
- K. Tanaka, T. Kaneko, N. Hojo, and H. Kameoka, Synthetic-to-natural speech waveform conversion using cycle-consistent adversarial networks, SLT, 2018.
- O. Ocal, O. H. Elibol, G. Keskin, C. Stephenson, A. Thomas, and K. Ramchandran, Adversarially trained autoencoders for parallel-data-free voice conversion, ICASSP, 2019.
- F. Fang, X. Wang, J. Yamagishi, and I. Echizen, Audiovisual speaker conversion: Jointly and simultaneously transforming facial expression and acoustic characteristics, ICASSP, 2019.
- S. Seshadri, L. Juvela, J. Yamagishi, Okko Räsänen, and P. Alku, Cycle-consistent adversarial networks for non-parallel vocal effort based speaking style conversion, ICASSP, 2019.
- T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, CYCLEGAN-VC2: Improved cyclegan-based non-parallel voice conversion, ICASSP, 2019.
- L. Juvela, B. Bollepalli, J. Yamagishi, and P. Alku, Waveform generation for text-to-speech synthesis using pitch-synchronous multi-scale generative adversarial networks, ICASSP, 2019.

# References


**Speaker recognition**

- Q. Wang, W. Rao, S.-I. Sun, L. Xie, E.-S. Chng, and H.-Z. Li, Unsupervised domain adaptation via domain adversarial training for speaker recognition, ICASSP, 2018.
- H. Yu, Z.-H. Tan, Z.-Y. Ma, and J. Guo, Adversarial network bottleneck features for noise robust speaker verification, arXiv, 2017.
- G. Bhattacharya, J. Alam, & P. Kenny, Adapting end-to-end neural speaker verification to new languages and recording conditions with adversarial training, ICASSP, 2019.
- Z. Peng, S. Feng, & T. Lee, Adversarial multi-task deep features and unsupervised back-end adaptation for language recognition, ICASSP, 2019.
- Z. Meng, Y. Zhao, J. Li, & Y. Gong, Adversarial speaker verification, ICASSP, 2019.
- X. Fang, L. Zou, J. Li, L. Sun, & Z.-H. Ling, Channel adversarial training for cross-channel text-independent speaker recognition, ICASSP, 2019.
- W. Xia, J. Huang, & J. H. Hansen, Cross-lingual text-independent speaker verification using unsupervised adversarial discriminative domain adaptation, ICASSP, 2019.
- P. S. Nidadavolu, J. Villalba, & N. Dehak, Cycle-GANs for domain adaptation of acoustic features for speaker recognition, ICASSP, 2019.
- G. Bhattacharya, J. Monteiro, J. Alam, & P. Kenny, Generative adversarial speaker embedding networks for domain robust end-to-end speaker verification, ICASSP, 2019.
- J. Rohdin, T. Stafylakis, A. Silnova, H. Zeinali, L. Burget, & O. Plchot, Speaker verification using end-to-end adversarial language adaptation, ICASSP, 2019.
- Zhou, J., Jiang, T., Li, L., Hong, Q., Wang, Z., & Xia, B., Training multi-task adversarial network for extracting noise-robust speaker embedding, ICASSP, 2019.
- J. Zhang, N. Inoue, & K. Shinoda, I-vector transformation using conditional generative adversarial networks for short utterance speaker verification, arXiv, 2018.
- W. Ding, & L. He, Mtgan: Speaker verification through multitasking triplet generative adversarial networks, arXiv, 2018.
- X. Miao, I. McLoughlin, S. Yao, & Y. Yan, Improved conditional generative adversarial net classification for spoken language recognition, SLT, 2018.

# References


**Automatic Speech Recognition**

- Yusuke Shinohara, Adversarial multi-task learning of deep neural networks for robust speech recognition, Interspeech, 2016.
- D. Serdyuk, K. Audhkhasi, P. Brakel, B. Ramabhadran, S. Thomas, and Y. Bengio, Invariant Representations for Noisy Speech Recognition, arXiv, 2016.
- Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara, Cross-domain speech recognition using nonparallel corpora with cycle-consistent adversarial networks, ASRU, 2017.
- A. Sriram, H.-W Jun, Y. Gaur, and S. Satheesh, Robust speech recognition using generative adversarial networks, arXiv, 2017.
- Z. Meng, Z. Chen, V. Mazalov, J. Li, J., and Y. Gong, Unsupervised adaptation with domain separation networks for robust speech recognition, ASRU, 2017.
- Z. Meng, J. Li, Z. Chen, Y. Zhao, V. Mazalov, Y. Gong, and B.-H. Juang, Speaker-invariant training via adversarial learning, ICASSP, 2018.
- Z. Meng, J. Li, Y. Gong, and B.-H. Juang, Adversarial teacher-student learning for unsupervised domain adaptation, ICASSP, 2018.
- Y. Zhang, P. Zhang, and Y. Yan, Improving language modeling with an adversarial critic for automatic speech recognition, Interspeech, 2018.
- S. Sun, C. Yeh, M. Ostendorf, M. Hwang, and L. Xie, Training augmentation with adversarial examples for robust speech recognition, Interspeech, 2018.
- Z. Meng, J. Li, Y. Gong, and B.-H. Juang, Adversarial feature-mapping for speech enhancement, Interspeech 2018.
- K. Wang, J. Zhang, S. Sun, Y. Wang, F. Xiang, and L. Xie, Investigating generative adversarial networks based speech dereverberation for robust speech recognition, Interspeech 2018.
- Z. Meng, J. Li, Y. Gong, B.-H. Juang, Cycle-consistent speech enhancement, Interspeech 2018.
- J. Drexler and J. Glass, Combining end-to-end and adversarial training for low-resource speech recognition, SLT, 2018.
- A. H. Liu, H. Lee and L. Lee, Adversarial training of end-to-end speech recognition using a criticizing language model, ICASSP, 2019.

# References


**Automatic Speech Recognition**

- J. Yi, J. Tao and Y. Bai, Language-invariant bottleneck features from adversarial end-to-end acoustic models for
- low resource speech recognition, ICASSP, 2019.
- D. Haws and X. Cui, Cyclegan bandwidth extension acoustic modeling for automatic speech recognition, ICASSP, 2019.
- Z. Meng, J. Li, J. and Y. Gong, Attentive adversarial learning for domain-Invariant training, ICASSP, 2019.
- Z. Meng, Y. Zhao, J. Li, and Y. Gong, Adversarial speaker verification, ICASSP, 2019.
- Z. Meng, Y. Zhao, J. Li, and Y. Gong., Adversarial speaker adaptation, ICASSP, 2019.

**Emotion recognition**

- J. Chang, and S. Scherer, Learning representations of emotional speech with deep convolutional generative adversarial networks, ICASSP, 2017.
- S. Sahu, R. Gupta, G. Sivaraman, W. AbdAlmageed, and C. Espy-Wilson, Adversarial auto-encoders for speech based emotion recognition. Interspeech, 2017.
- S. Sahu, R. Gupta, and C. E.-Wilson, On enhancing speech emotion recognition using generative adversarial networks, Interspeech 2018.
- C.-M. Chang, and C.-C. Lee, Adversarially-enriched acoustic code vector learned from out-of-context affective corpus for robust emotion recognition, ICASSP 2019.
- J. Liang, S. Chen, J. Zhao, Q. Jin, H. Liu, and L. Lu, Cross-culture multimodal emotion recognition with adversarial learning, ICASSP 2019.

**Lipreading**

- M. Wand, and J. Schmidhuber, Improving speaker-independent lipreading with domain-adversarial training, arXiv, 2017.

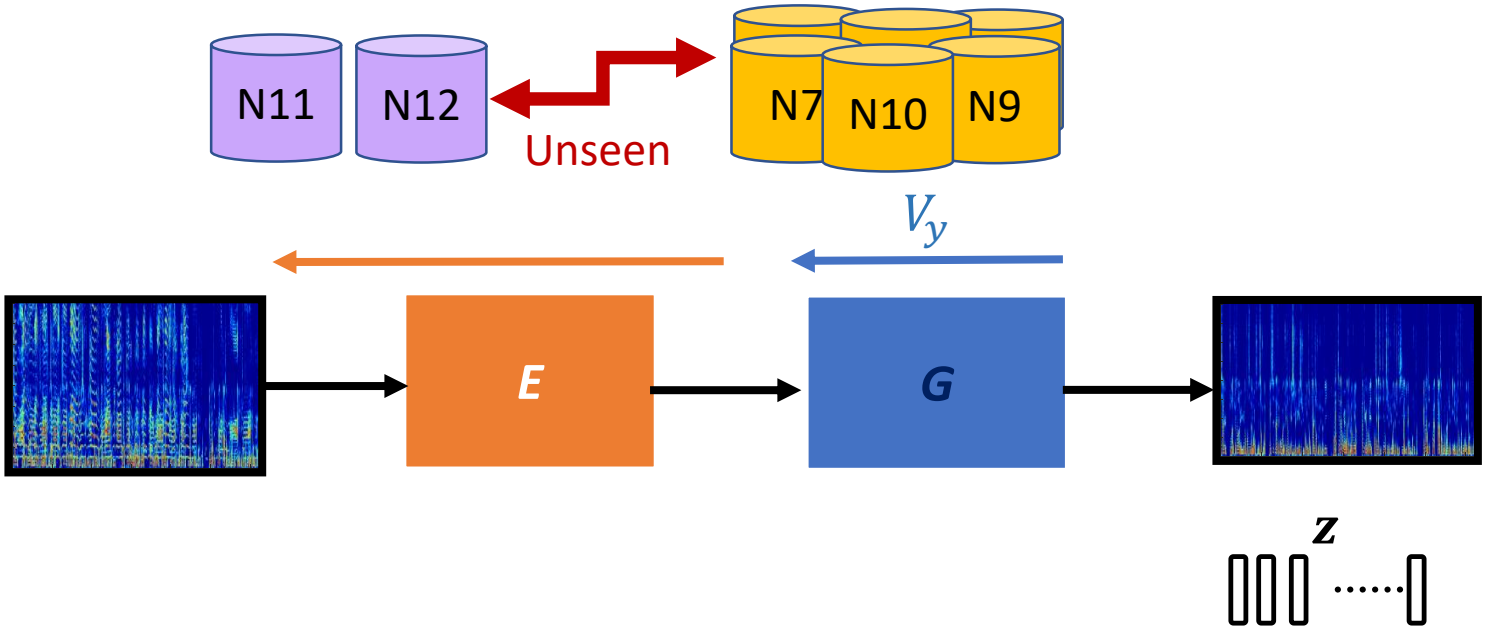
# Outline of Part III

## Our Recent Works

- Noise adaptive speech enhancement [Interspeech 2019] [Wed-P-6-E]
- MetricGAN for speech enhancement [ICML 2019]
- Multi-Target voice conversion [Interspeech 2018]
- Impaired speech conversion [Interspeech 2019] [Mon-P-2-A]
- Pathological voice detection [NeurIPS workshop 2018]

# Speech Enhancement

- Noise Adaptive Speech Enhancement (NA-SE)
  [Liao et al., Interspeech 2019] [Wed-P-6-E]




$$\theta_G \leftarrow \theta_G - \epsilon \frac{\partial V_y}{\partial \theta_G}$$

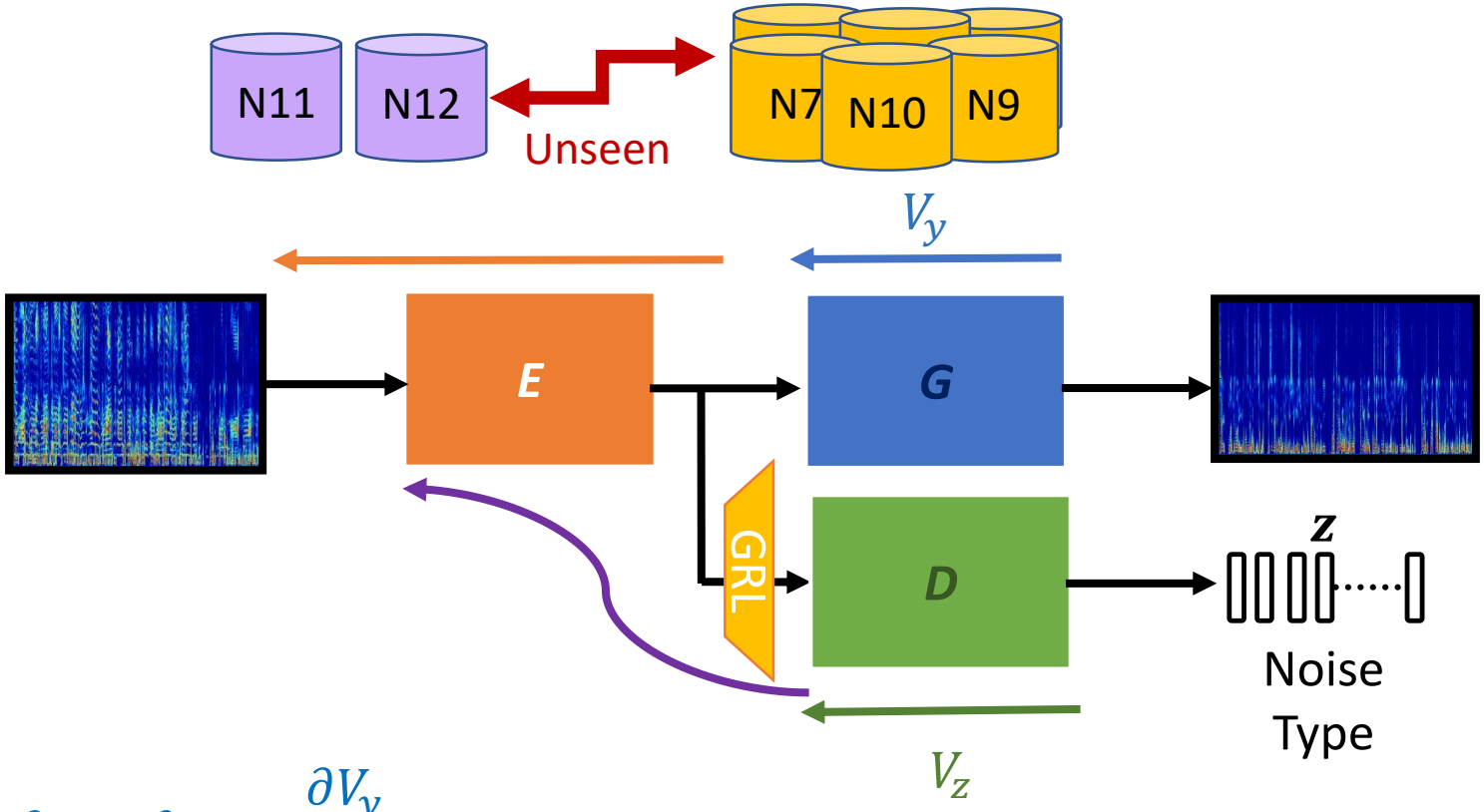
Min reconstruction error

$$\theta_E \leftarrow \theta_E - \epsilon \left( \frac{\partial V_y}{\partial \theta_E} \right)$$

Min reconstruction error

# Speech Enhancement (NA-SE)

- Domain adversarial training for NA-SE




$\theta_G \leftarrow \theta_G - \epsilon \frac{\partial V_y}{\partial \theta_G}$ Min reconstruction error

$\theta_D \leftarrow \theta_D - \epsilon \frac{\partial V_z}{\partial \theta_D}$ Max domain accuracy

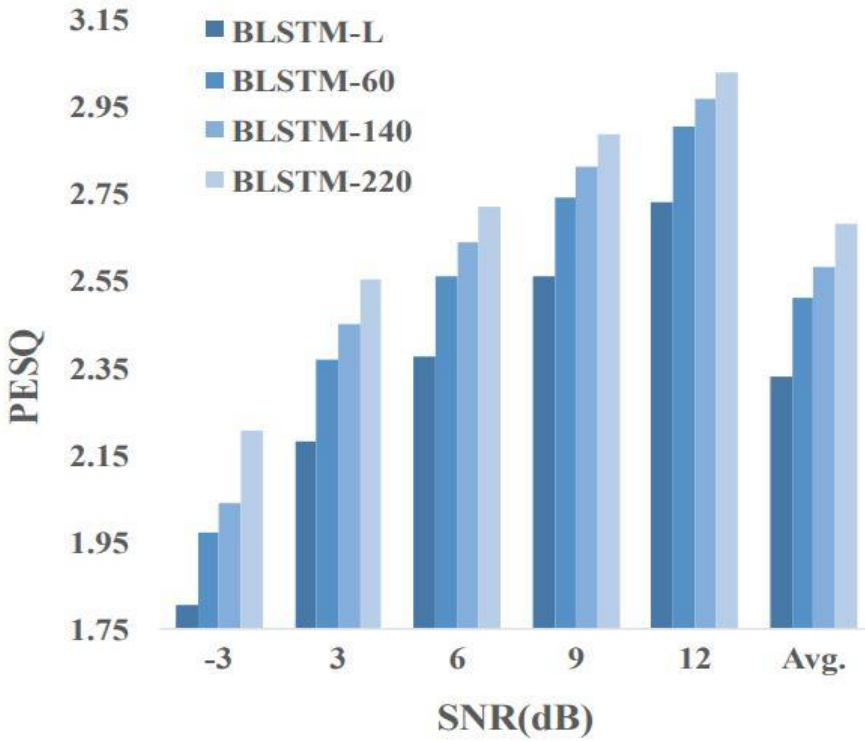
$\theta_E \leftarrow \theta_E - \epsilon \left( \frac{\partial V_y}{\partial \theta_E} \right) + \alpha \frac{\partial V_z}{\partial \theta_E}$

Min reconstruction error and Min domain accuracy

# Speech Enhancement (NA-SE)

- Objective evaluations

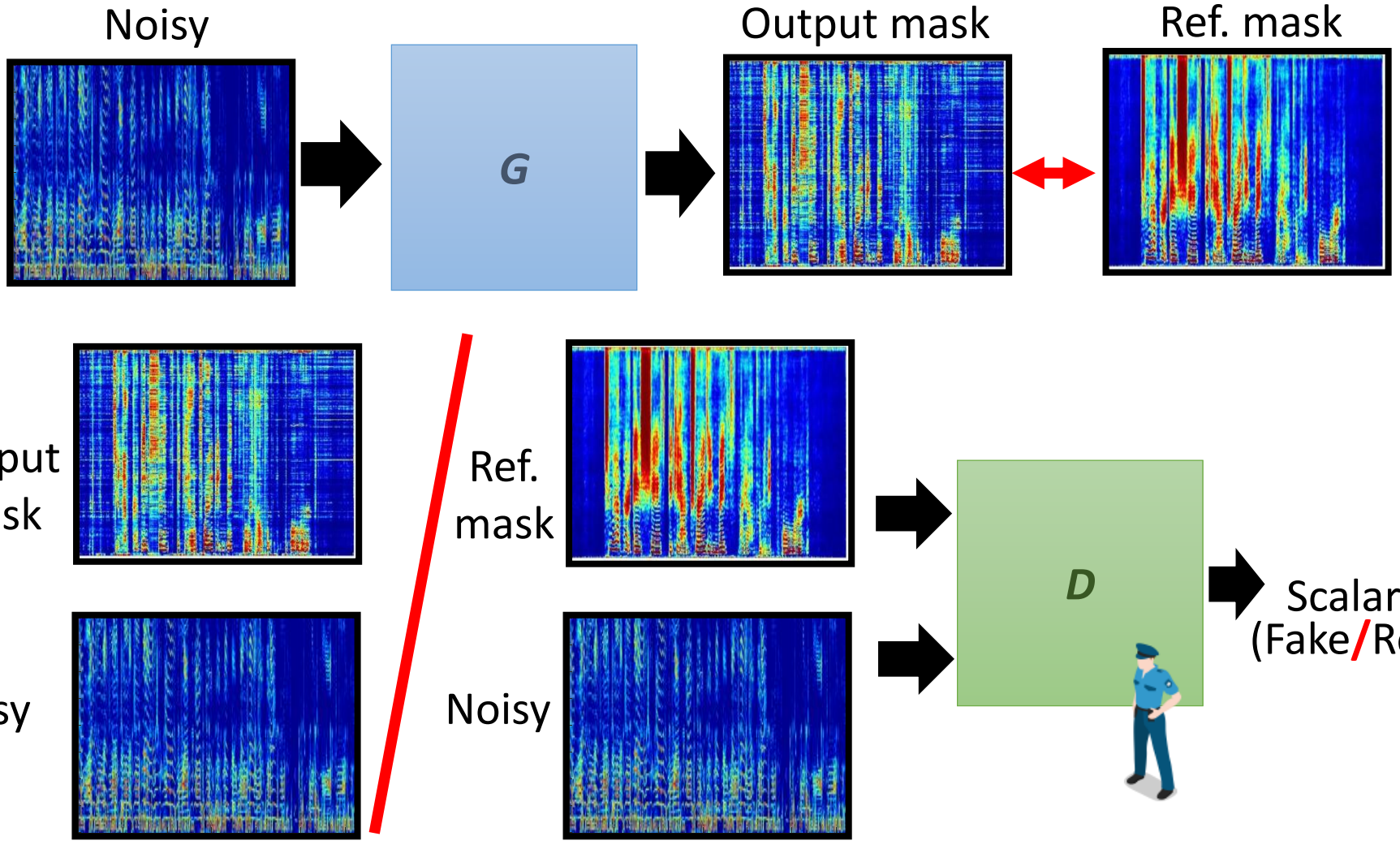
Fig. 15: PESQ at different SNR levels.




The DAT-based unsupervised adaptation can notably overcome the mismatch issue of training and testing noise types.

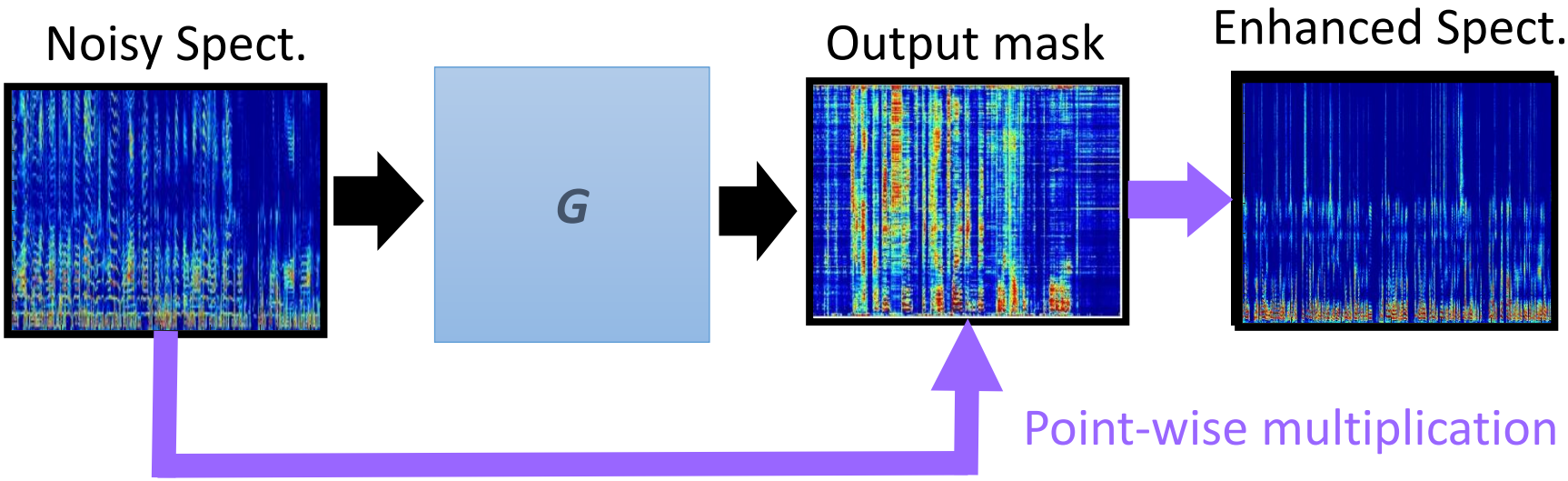
# Speech Enhancement

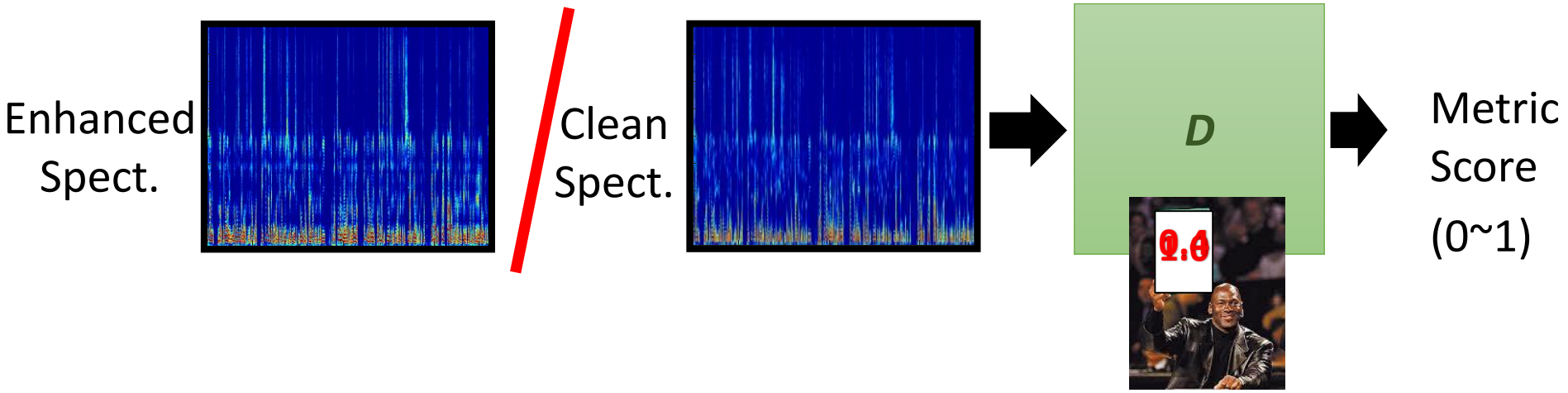
- GAN for spectral magnitude mask estimation (MMS-GAN)
  [Pandey et al., ICASSP 2018]



# Speech Enhancement

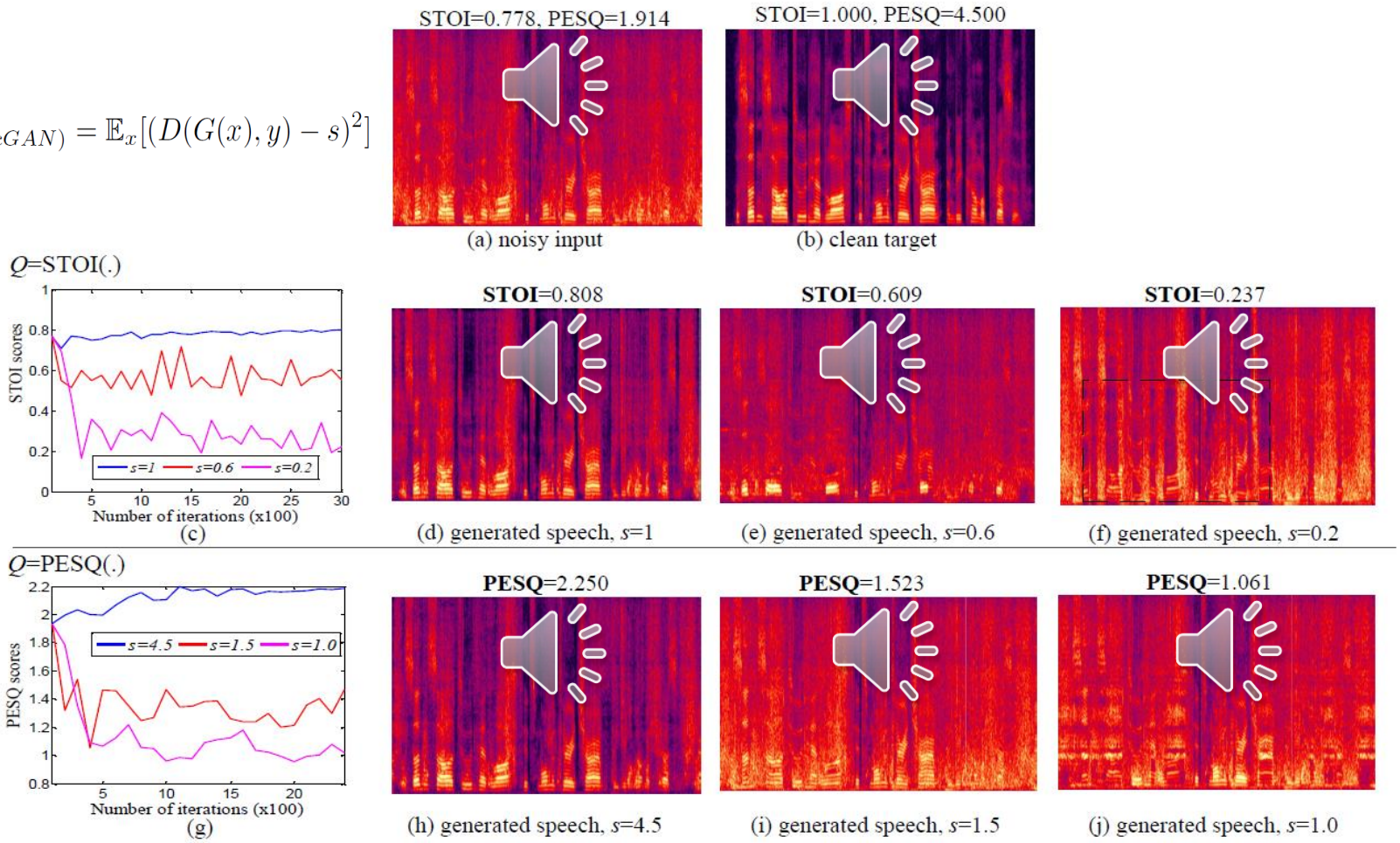
- MetricGAN for Speech Enhancement [Fu et al., ICML 2019]



# Speech Enhancement (MetricGAN)

$$L_{G(MetricGAN)} = \mathbb{E}_x[(D(G(x), y) - s)^2]$$

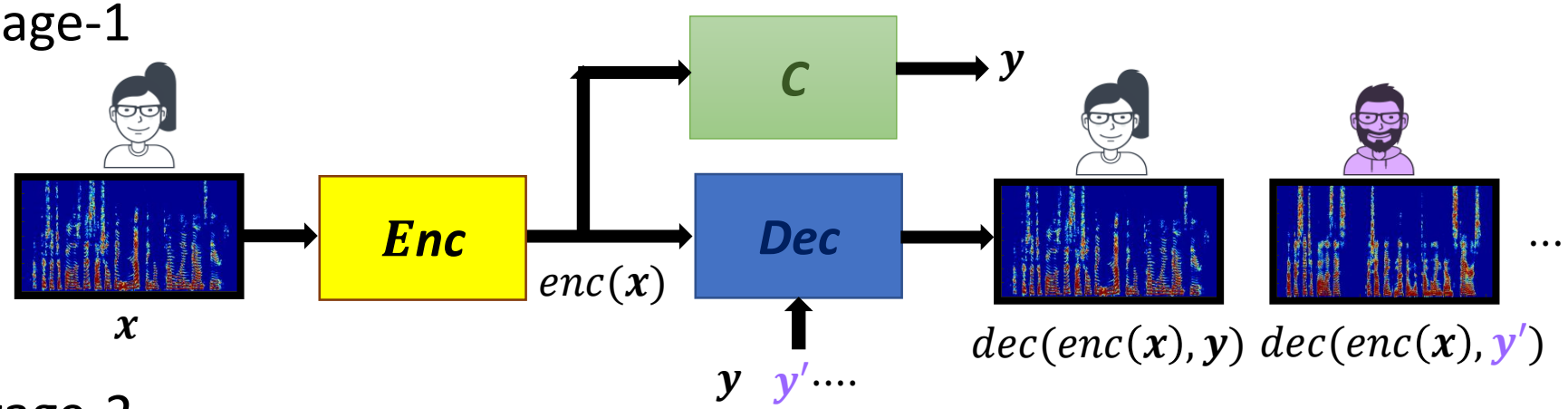



With MetricGAN, we have freedom to specify the target metric scores (PESQ or STOI) to generated speech.

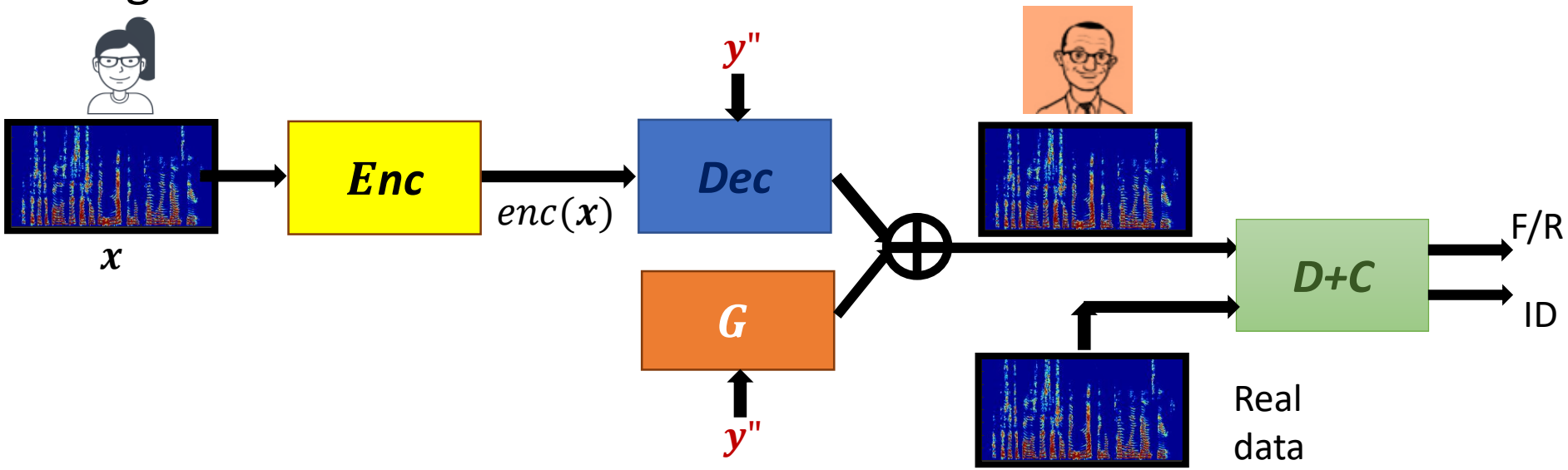
# Voice Conversion

- Multi-target VC [Chou et al., Interspeech 2018]

➢ Stage-1



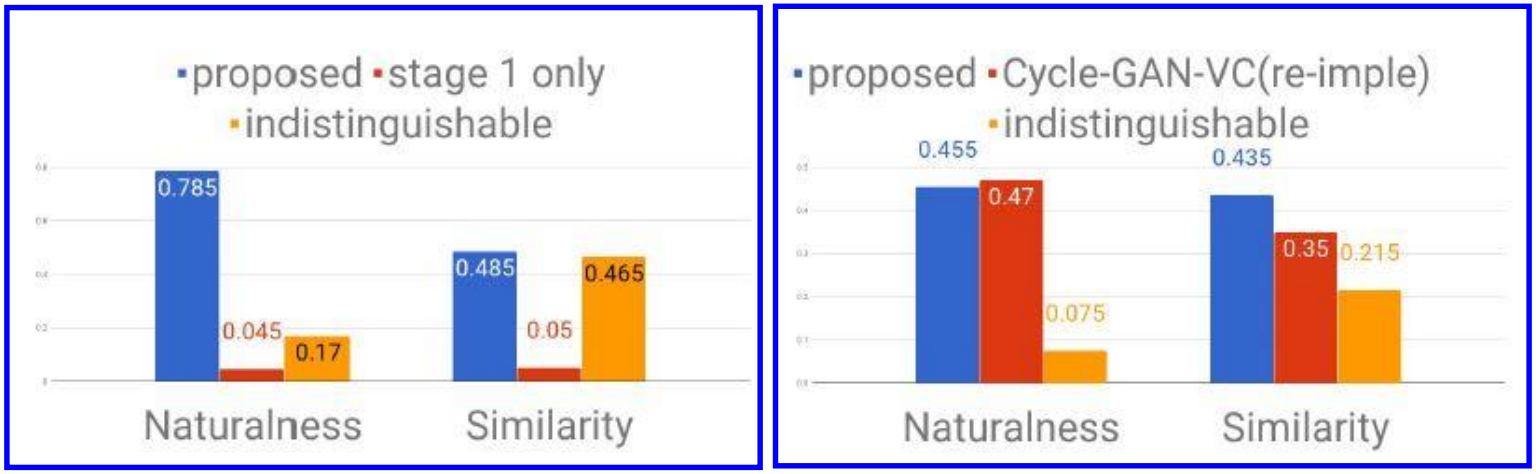

➢ Stage-2



# Voice Conversion (Multi-target VC)

- Subjective evaluations

Fig. 16: Preference test results




1. The proposed method uses **non-parallel** data.
2. The multi-target VC approach outperforms one-stage only.
3. The multi-target VC approach is comparable to Cycle-GAN-VC in terms of the naturalness and the similarity.

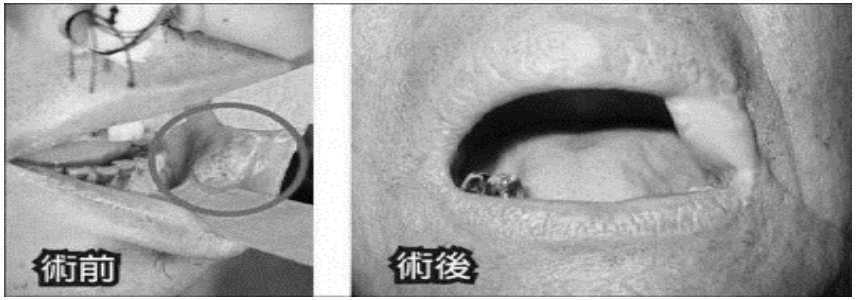
# Voice Conversion

- Controller-generator-discriminator VC on Impaired Speech [Chen et al., Interspeech 2019] [Mon-P-2-A]

**Previous applications:** hearing aids; murmur to normal speech; bone-conductive microphone to air-conductive microphone.

**Proposed:** improving the speech intelligibility of surgical patients.

**Target:** oral cancer (top five cancer for male in Taiwan).

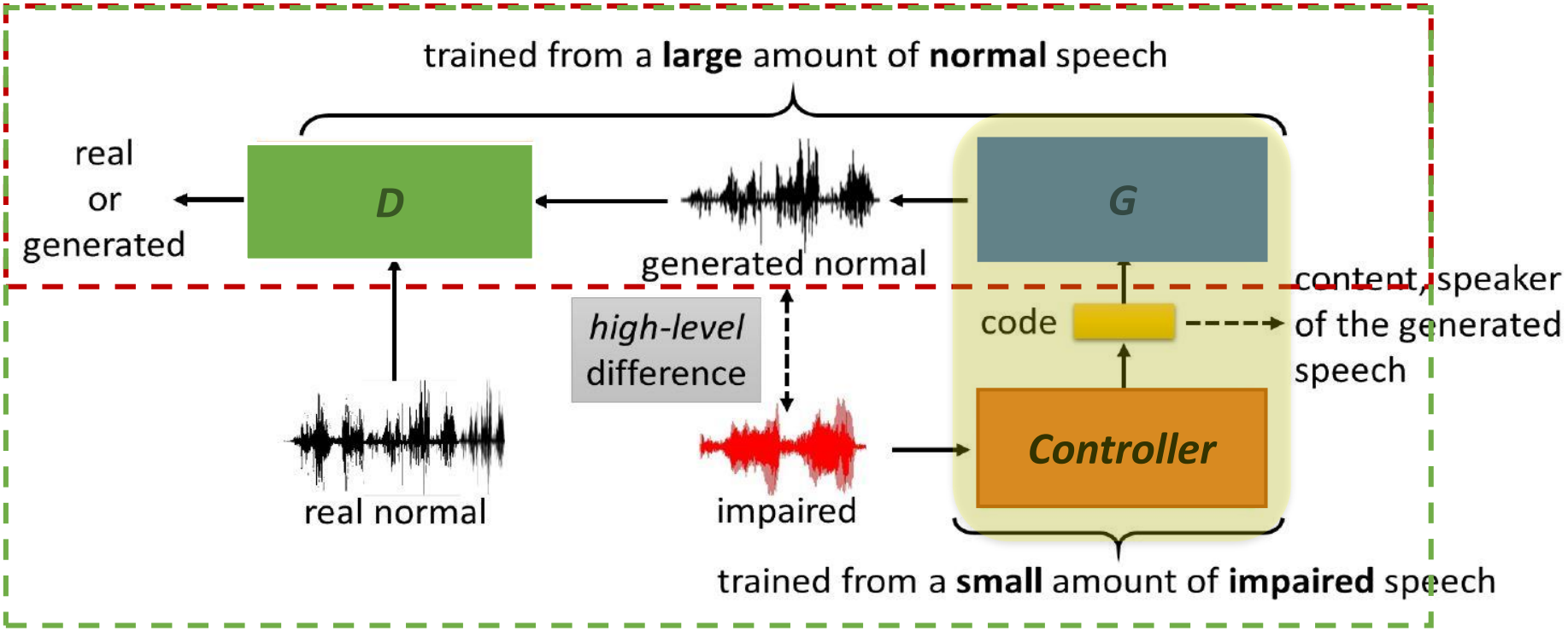



Before After

Before After

# Voice Conversion

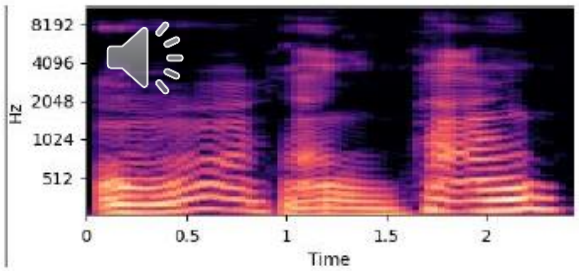
- Controller-generator-discriminator VC (CGD VC) on impaired speech [Chen et al., Interspeech 2019]



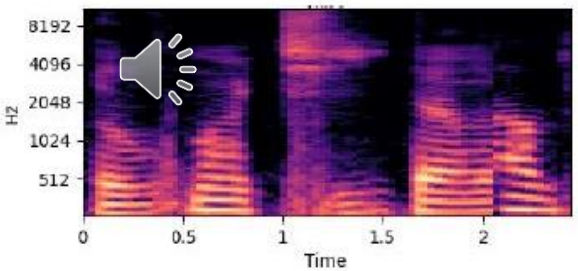
# Voice Conversion (CGD VC)

- Spectrogram analysis

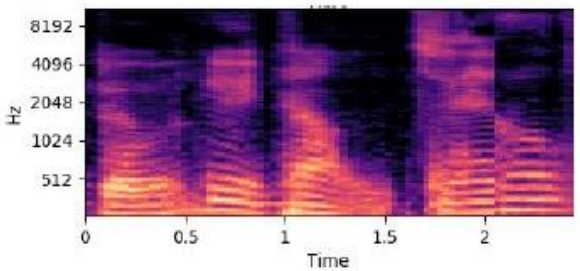
Fig. 17: Spectrogram comparison of CGD with CycleGAN.



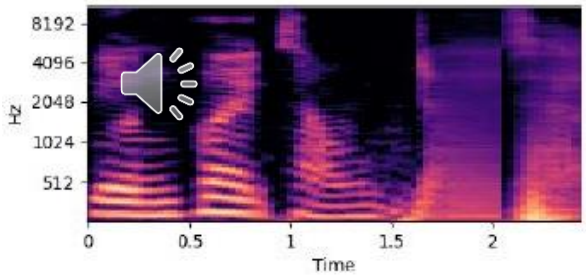

(a) Impaired speech



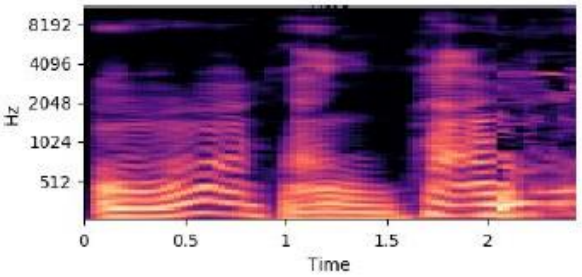

(b) Proposed model




(c) NoSD model




(d) CycleGAN transformed

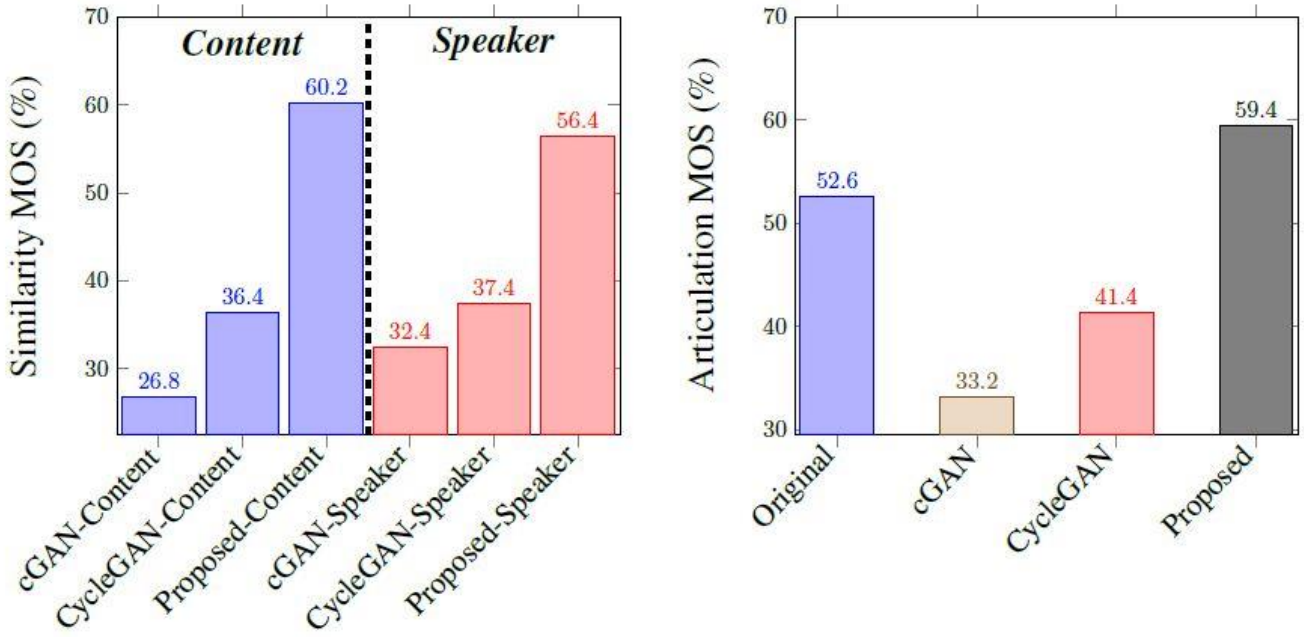



(e) CycleGAN reconstructed

# Voice Conversion (CGD VC)

- Subjective evaluations

Fig. 18: MOS for content similarity, speaker similarity, and articulation.




The proposed method outperforms conditional GAN and CycleGAN in terms of content similarity, speaker similarity, and articulation.

# Pathological Voice Detection

- Detection of Pathological Voice Using Cepstrum Vectors: A Deep Learning Approach [Fang et al., Journal of Voice 2018]

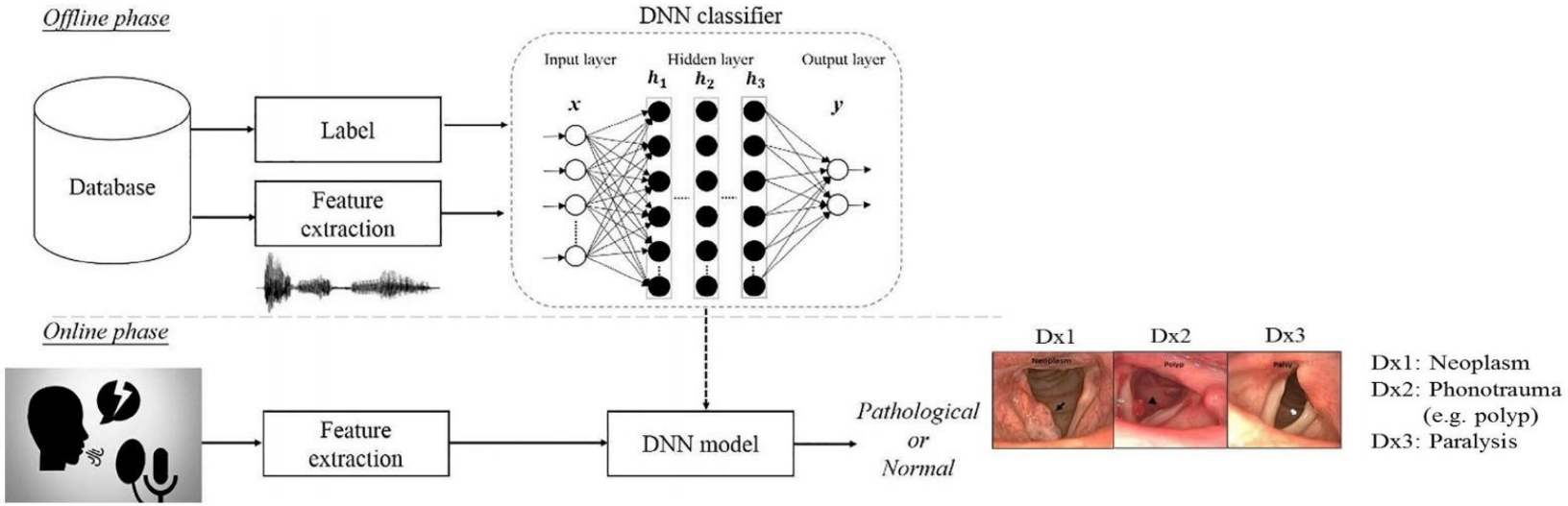



Table 17: Detection performance based on voice.

| | GMM | SVM | DNN |
|---|---|---|---|
| MEEI | 98.28 | 98.26 | 99.14 |
| FEMH (M) | 90.24 | 93.04 | 94.26 |
| FEMH (F) | 90.20 | 87.40 | 90.52 |

# Pathological Voice Detection

- Robustness Against Channel [Hsu et al., NeurIPS Workshop 2018]

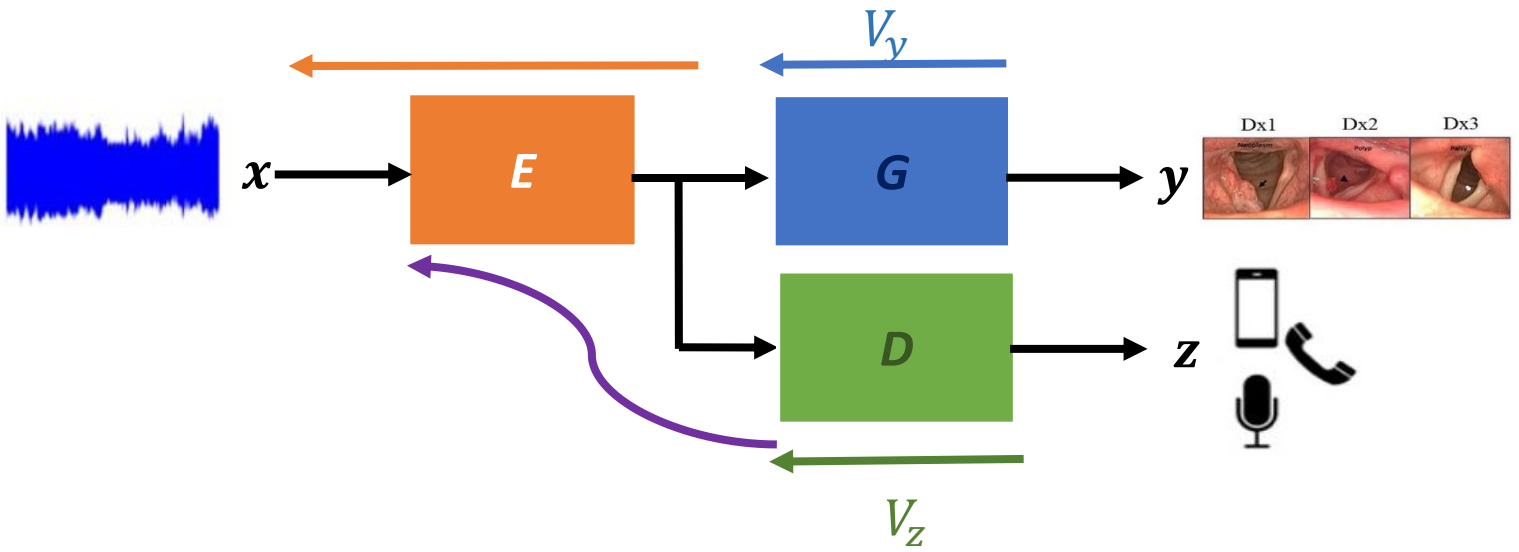



Table 18: Detection results of sup. and unsup. DAT under channel mismatches.

| | DNN (S) | DNN (T) | DNN (FT) | Unsup. DAT | Sup. DAT |
|---|---|---|---|---|---|
| PR-AUC | 0.8848 | 0.8509 | 0.9021 | 0.9455 | 0.9522 |

The unsupervised DAT notably increased the performance robustness against channel effects and generated comparable results as compared to supervised DAT.

# References


- C.-F. Liao, Y. Tsao, H.-Y. Lee and H.-M. Wang, Noise adaptive speech enhancement using domain adversarial training, Interspeech 2019.
- J.-C. Chou, C.-C. Yeh, H.-Y. Lee, and L.-S. Lee. "Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations. Interspeech 2018.
- L.-W. Chen, H.-Y. Lee, and Y. Tsao, Generative adversarial networks for unpaired voice transformation on impaired speech, Interspeech 2019.
- S.-W. Fu, C.-F. Liao, Y. Tsao, S.-D. Lin, MetricGAN: Generative adversarial networks based black-box metric scores optimization for speech enhancement, ICML, 2019.
- C.-T. Wang, F.-C. Lin, J.-Y. Chen, M.-J. Hsiao, S.-H. Fang, Y.-H. Lai, Y. Tsao, Detection of pathological voice using cepstrum vectors: a deep learning approach, Journal of Voice, 2018.
- S.-Y. Tsui, Y. Tsao, C.-W. Lin, S.-H. Fang, and C.-T. Wang, Demographic and symptomatic features of voice disorders and their potential application in classification using machine learning algorithms, Folia Phoniatrica et Logopaedica, 2018.
- S.-H. Fang, C.-T. Wang, J.-Y. Chen, Y. Tsao and F.-C. Lin, Combining acoustic signals and medical records to improve pathological voice classification, APSIPA, 2019.
- Y.-T. Hsu, Z. Zhu, C.-T. Wang, S.-H. Fang, F. Rudzicz, and Y. Tsao, Robustness against the channel effect in pathological voice detection, NeurIPS 2018 Machine Learning for Health (ML4H) Workshop, 2018.

# Generative Adversarial Network

## and its Applications to Signal Processing and Natural Language Processing

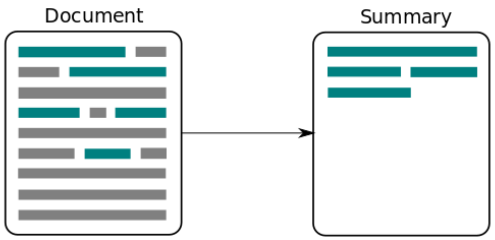
### Part III: Speech Signal Processing

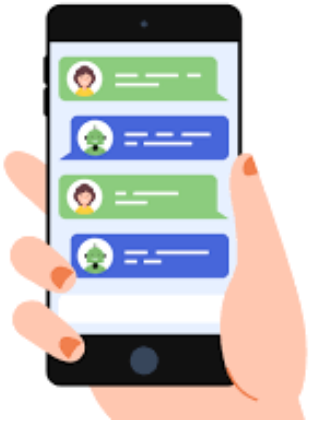
## Thank You Very Much

Tsao, Yu Ph.D., Academia Sinica

yu.tsao@citi.sinica.edu.tw

# Part IV: Natural Language Processing

NLP tasks usually involve **Sequence Generation**

How to use GAN to improve sequence generation?

# Outline of Part IV

Sequence Generation by GAN

Unsupervised Conditional Sequence Generation

- Text Style Transfer
- Unsupervised Abstractive Summarization
- Unsupervised Translation
- Unsupervised Speech Recognition

# Why we need GAN?

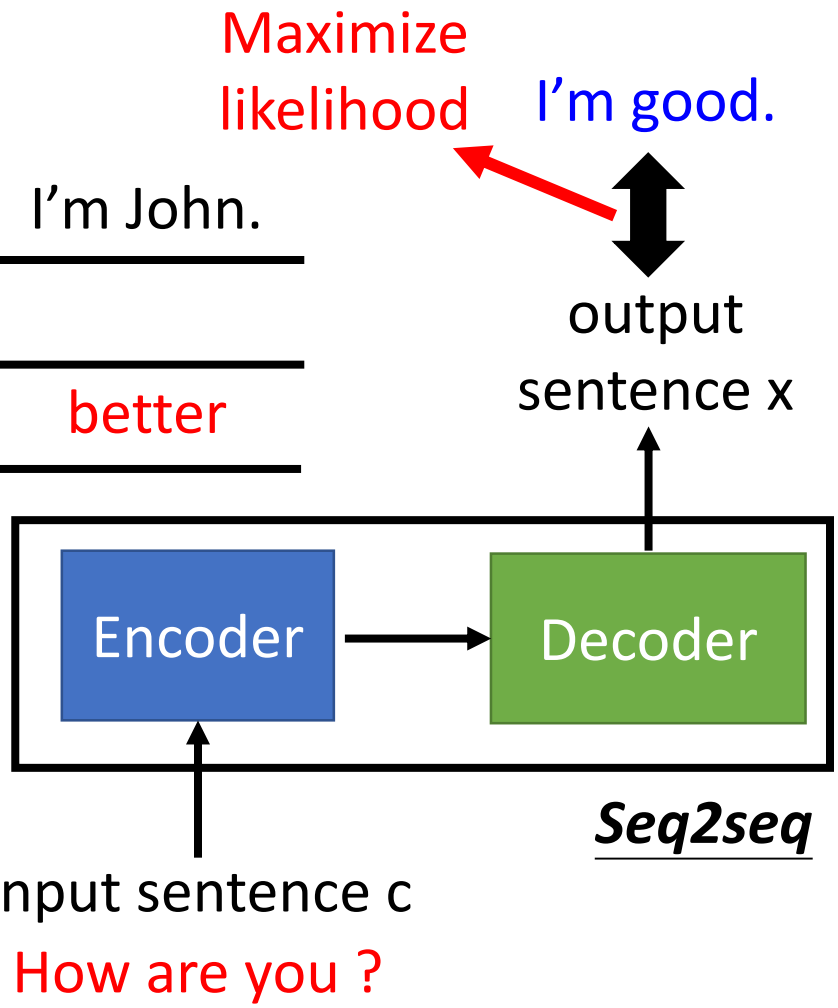
- Chat-bot as example

| Output: | Not bad | I'm John. |
|---|---|---|
| Human | better | |
| Training Criterion | | better |

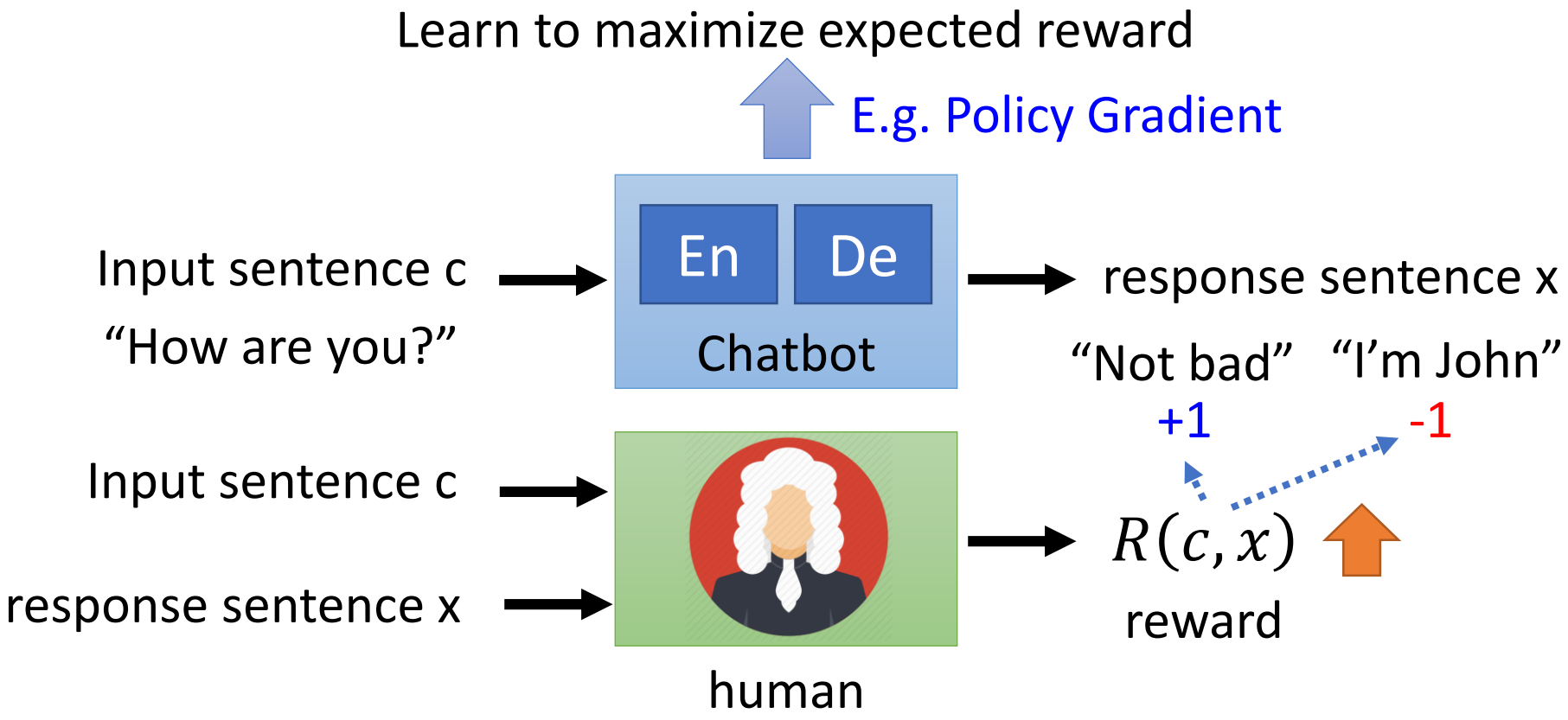
Training data:

A: How are you ?

B: I'm good.



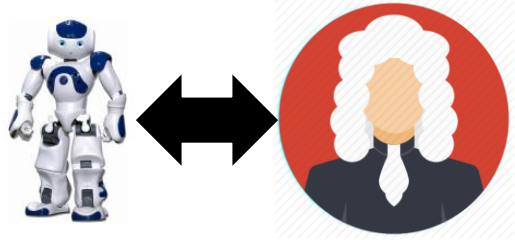
# Reinforcement Learning




[Li, et al., EMNLP, 2016]

# Policy Gradient

$\theta^t$



$(c^1, x^1)$ $R(c^1, x^1)$

$(c^2, x^2)$ $R(c^2, x^2)$

$(c^N, x^N)$ $R(c^N, x^N)$

$$\theta^{t+1} \leftarrow \theta^t + \eta \nabla \bar{R}_{\theta^t}$$

$$\frac{1}{N}\sum_{i=1}^{N} R(c^i, x^i) \nabla log P_{\theta^t}(x^i|c^i)$$

$R(c^i, x^i)$ is positive

Updating $\theta$ to increase $P_\theta(x^i|c^i)$

$R(c^i, x^i)$ is negative

Updating $\theta$ to decrease $P_\theta(x^i|c^i)$

# Policy Gradient

| | Maximum Likelihood | Reinforcement Learning - Policy Gradient |
|---|---|---|
| Objective Function | $\frac{1}{N}\sum_{i=1}^{N} logP_\theta(\hat{x}^i \mid c^i)$ | $\frac{1}{N}\sum_{i=1}^{N} R(c^i, x^i) logP_\theta(x^i \mid c^i)$ |
| Gradient | $\frac{1}{N}\sum_{i=1}^{N} \nabla logP_\theta(\hat{x}^i \mid c^i)$ | $\frac{1}{N}\sum_{i=1}^{N} R(c^i, x^i) \nabla logP_\theta(x^i \mid c^i)$ |
| Training Data | $\{(c^1, \hat{x}^1), \dots, (c^N, \hat{x}^N)\}$ $R(c^i, \hat{x}^i) = 1$ | $\{(c^1, x^1), \dots, (c^N, x^N)\}$ obtained from interaction weighted by $R(c^i, x^i)$ |

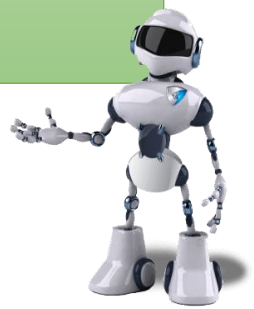
# Conditional GAN


However, there is an issue when you train your generator.

Input sentence c → En De → Chatbot → response sentence x

Input sentence c → Discriminator → $R(c, x)$ → reward

response sentence x → Discriminator

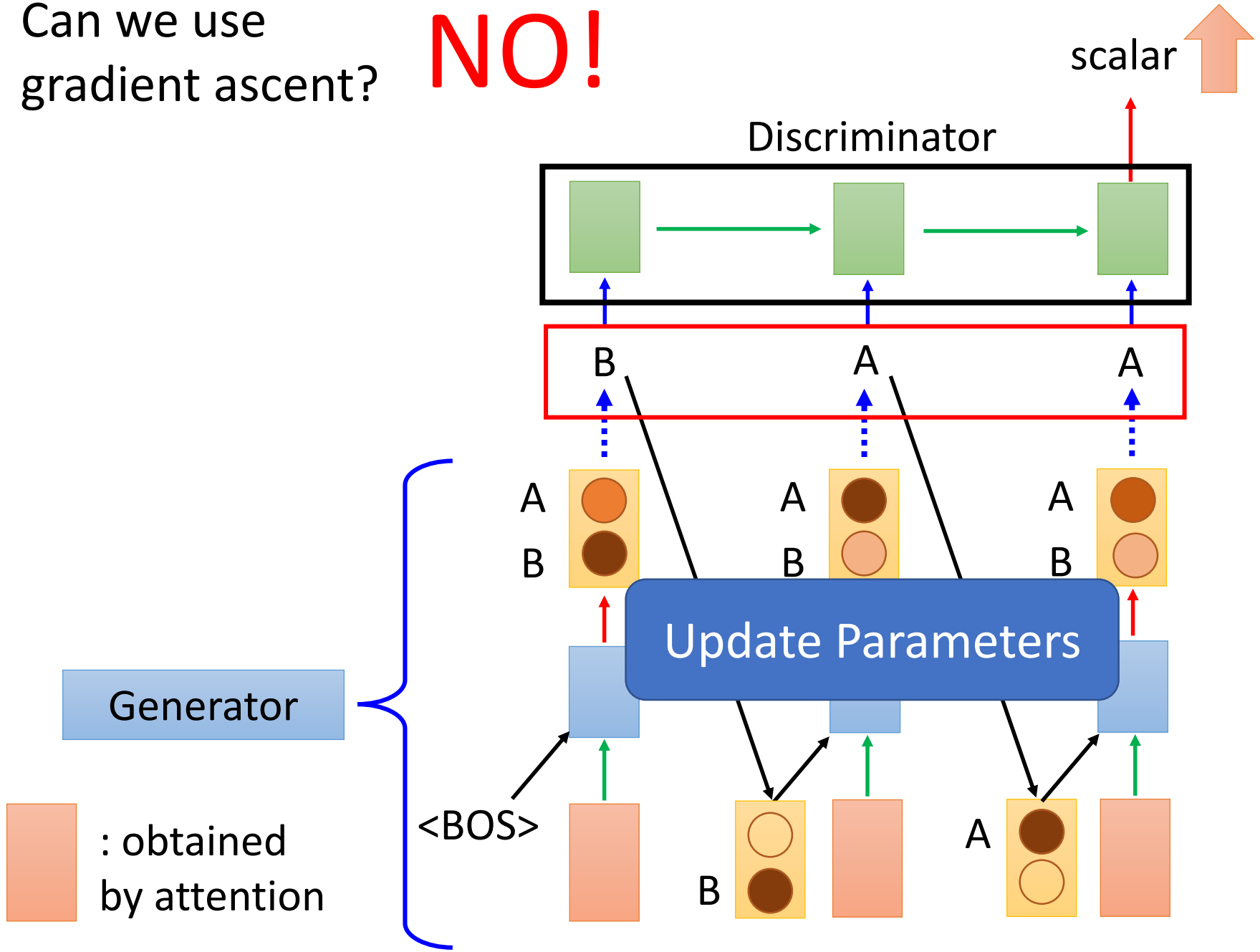


Replace human evaluation with **machine evaluation**

[Li, et al., EMNLP, 2017]

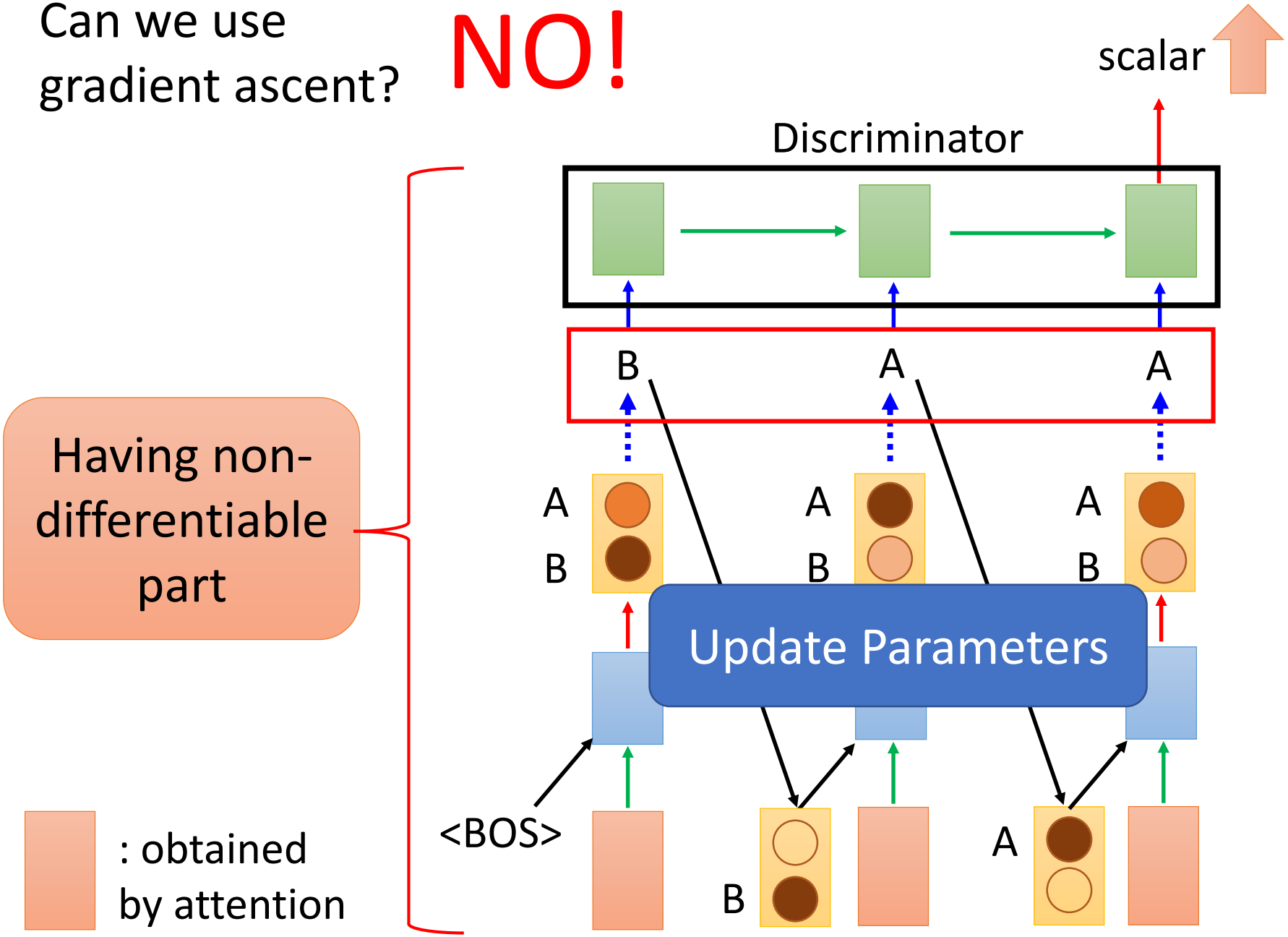
Can we use gradient ascent?

NO!



Can we use gradient ascent?

NO!



# Three Categories of Solutions

## Gumbel-softmax

- [Matt J. Kusner, et al., arXiv, 2016][Weili Nie, et al. ICLR, 2019]

## 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

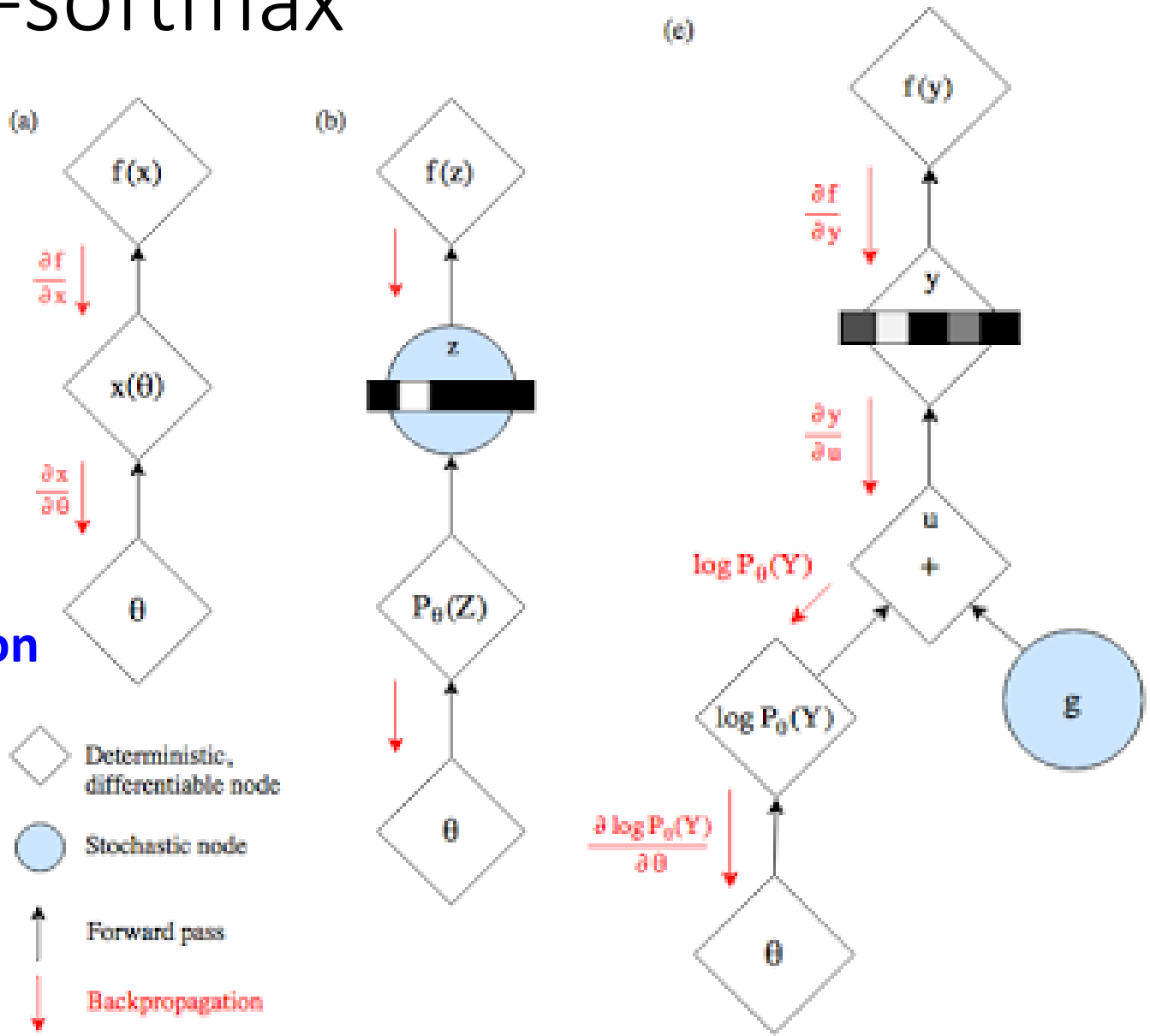
- [Yu, et al., AAAI, 2017][Li, et al., EMNLP, 2017][Tong Che, et al, arXiv, 2017][Jiaxian Guo, et al., AAAI, 2018][Kevin Lin, et al, NIPS, 2017][William Fedus, et al., ICLR, 2018]

Source of image: https://blog.evjang.com/2016/11/tutorial-categorical-variational.html

# Gumbel-softmax

Using the **reparameterization** trick

As what people do for training **VAE**



# Three Categories of Solutions

## Gumbel-softmax

- [Matt J. Kusner, et al., arXiv, 2016][Weili Nie, et al. ICLR, 2019]

## 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]

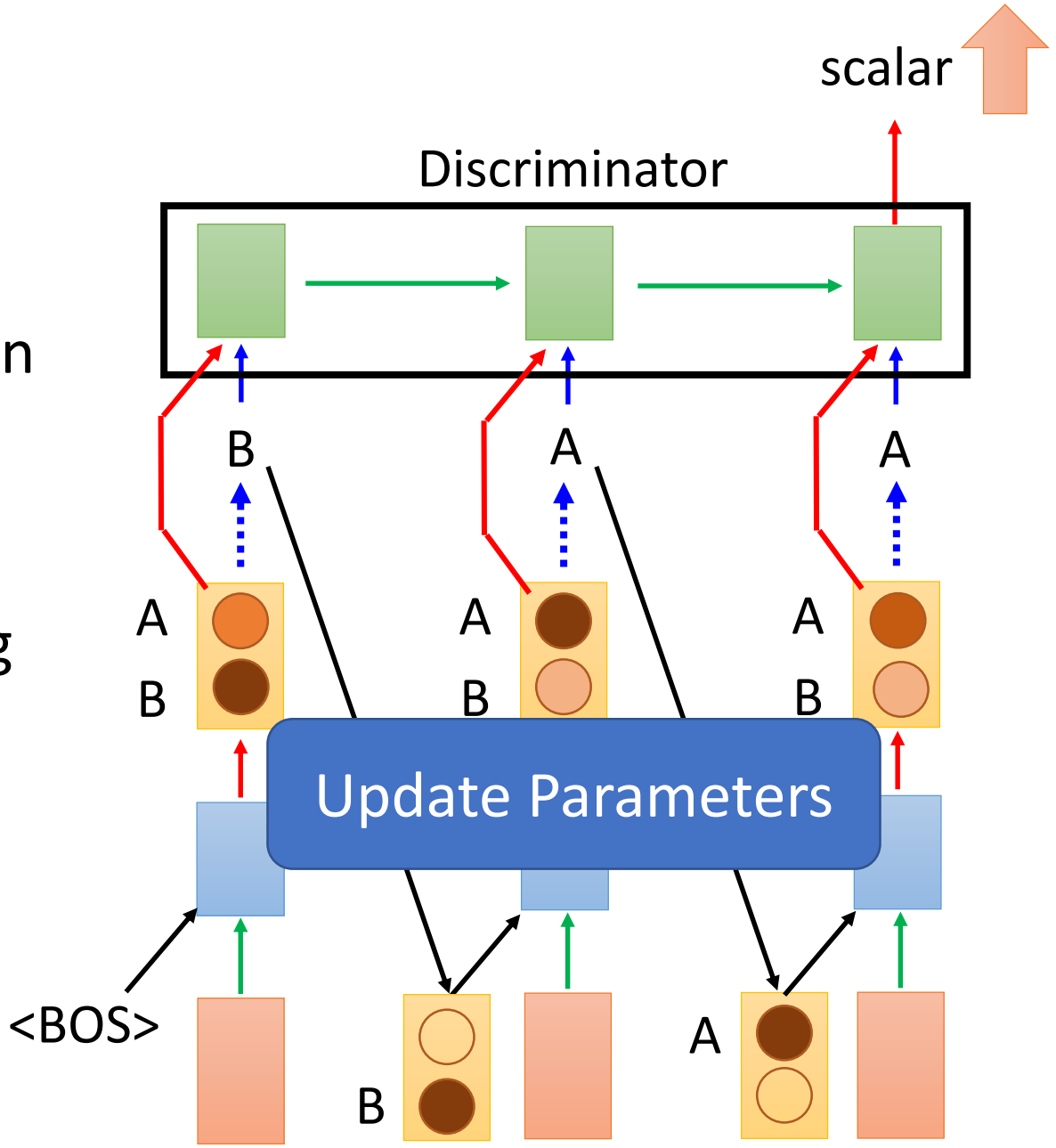
## Reinforcement Learning

- [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]

Use the distribution as the input of discriminator

Avoid the sampling process

We can do backpropagation now.



# What is the problem?

Discriminator with constraint (e.g. WGAN) can be helpful.

- 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-hot

Discriminator can immediately find the difference.

# Three Categories of Solutions

## Gumbel-softmax

- [Matt J. Kusner, et al., arXiv, 2016][Weili Nie, et al. ICLR, 2019]

## Continuous Input for Discriminator

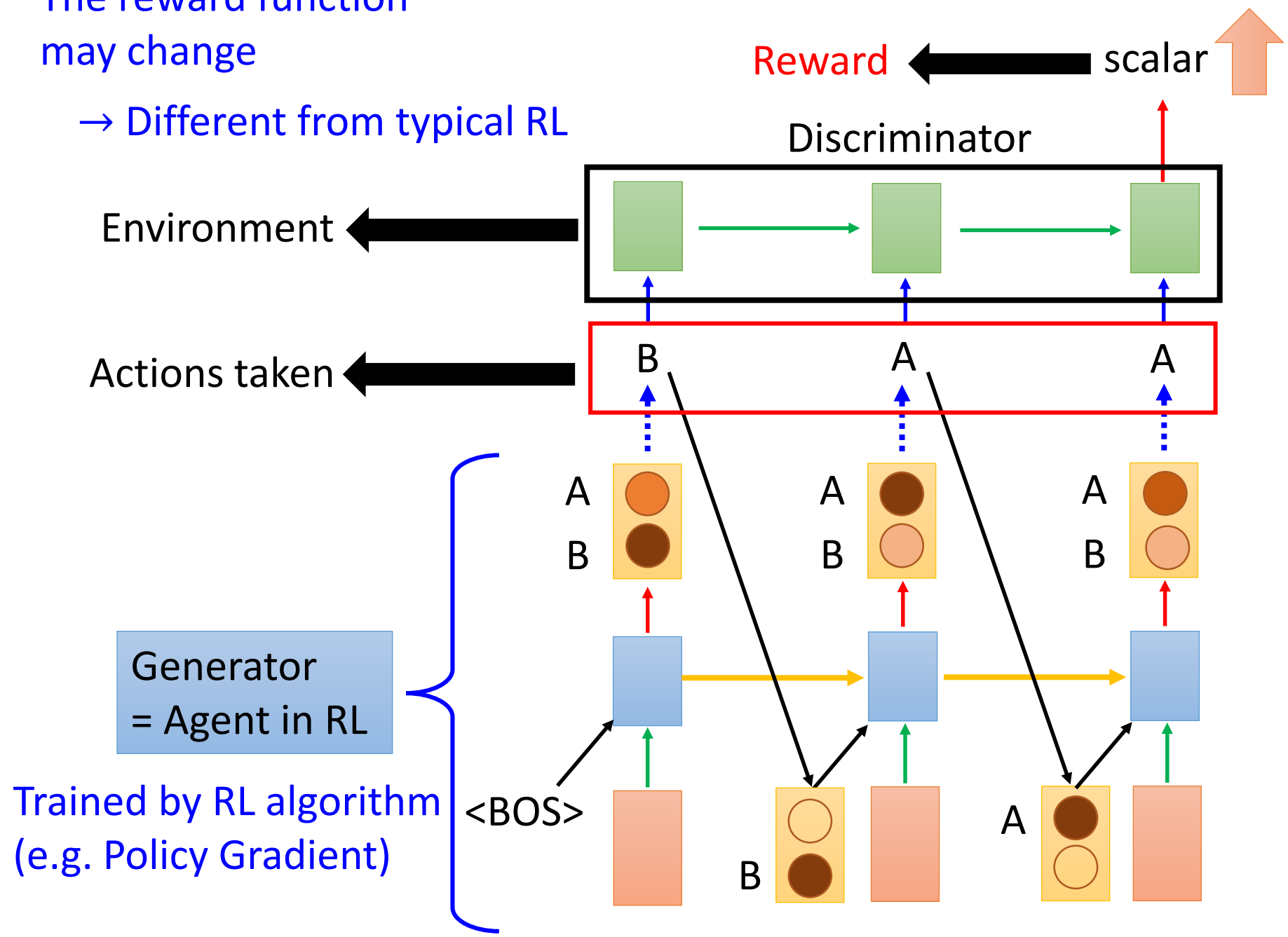
- [Sai Rajeswar, et al., arXiv, 2017][Ofir Press, et al., ICML workshop, 2017][Zhen Xu, et al., EMNLP, 2017][Alex Lamb, et al., NIPS, 2016][Yizhe Zhang, et al., ICML, 2017]

## 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]

The reward function may change

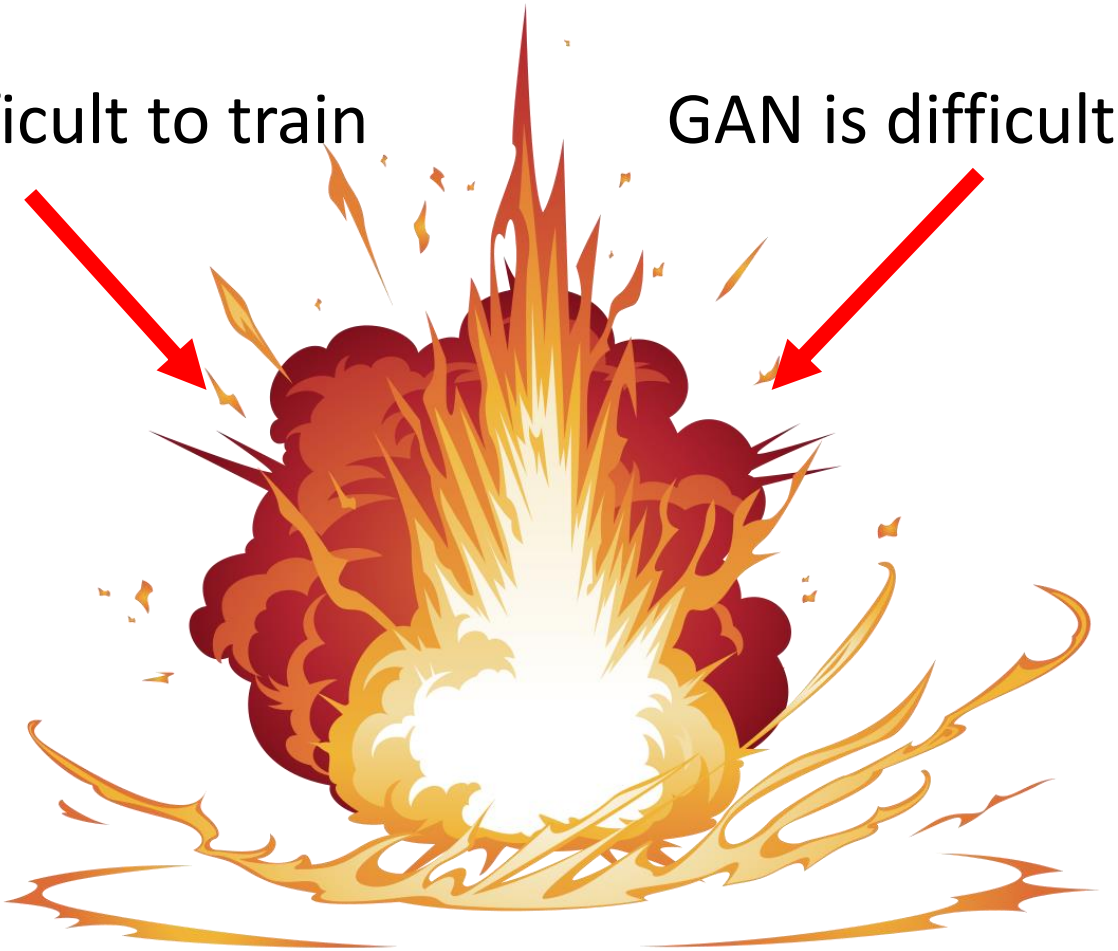
→ Different from typical RL



# Tips for Sequence Generation GAN

RL is difficult to train

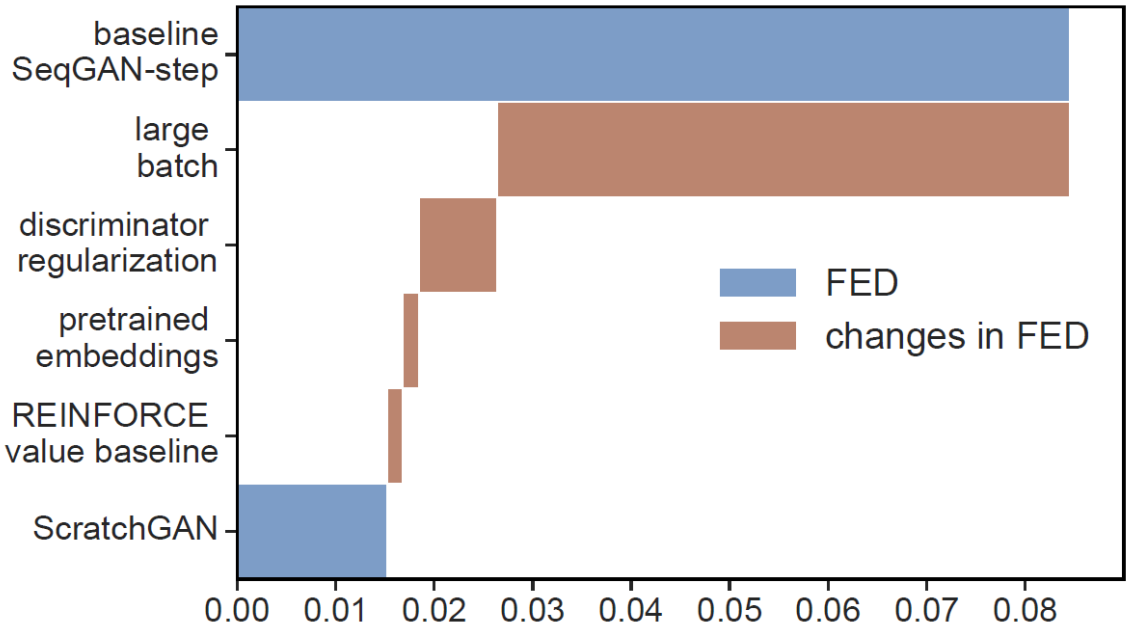
GAN is difficult to train



Sequence Generation GAN (RL+GAN)

# Tips for Sequence Generation GAN

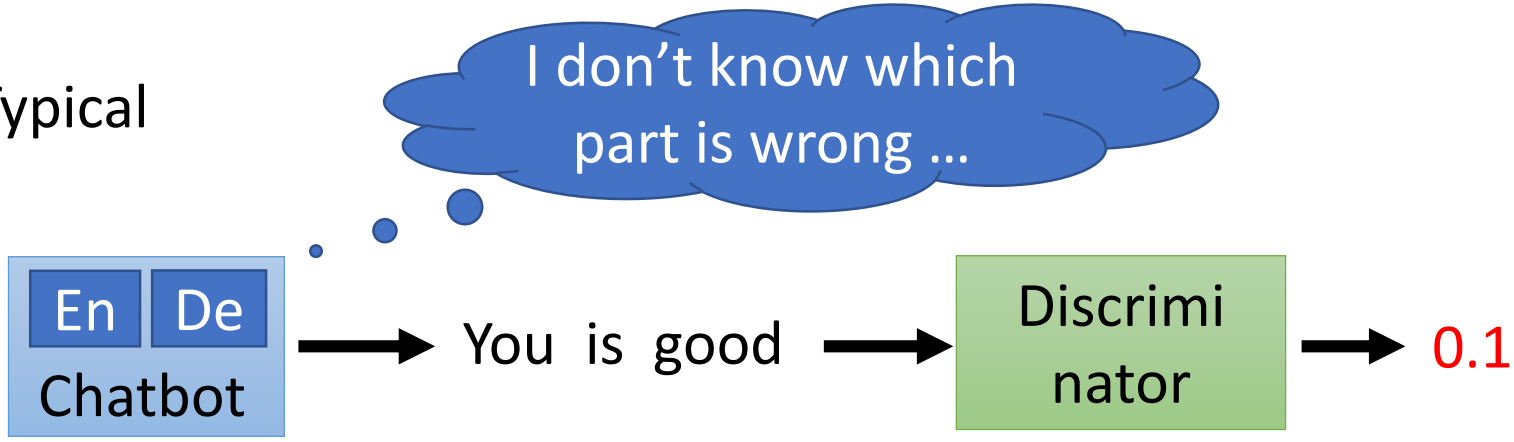
- Usually the generator are fine-tuned from a model learned by maximum-likelihood.
- However, with enough hyperparameter-tuning and tips, ScarchGAN can train from scratch.



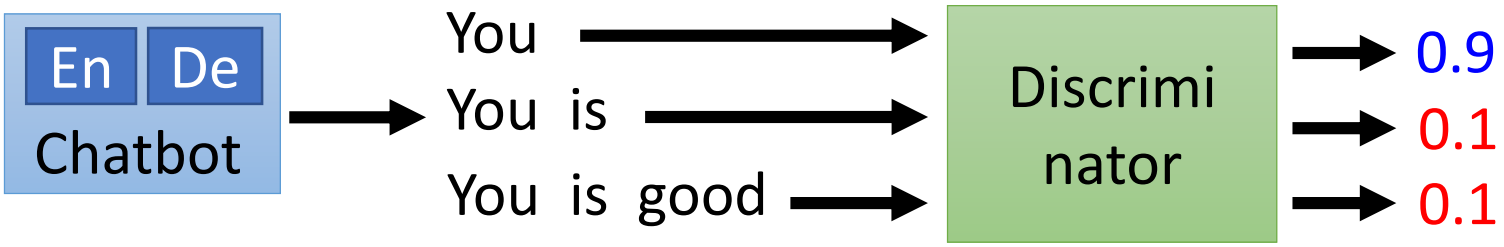

[Cyprien de Masson d'Autume, et al., arXiv 2019]

# Tips for Sequence Generation GAN

- Typical

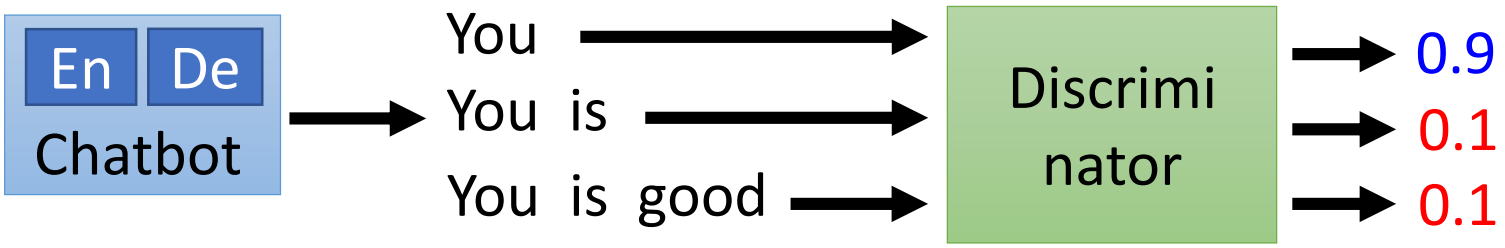



- Reward for Every Generation Step



# Tips for Sequence Generation GAN

- Reward for Every Generation Step




Method 1. Monte Carlo (MC) Search [Yu, et al., AAAI, 2017]

Method 2. Discriminator For Partially Decoded Sequences [Li, et al., EMNLP, 2017]

Method 3. Step-wise evaluation [Tual, Lee, TASLP, 2019][Xu, et al., EMNLP, 2018][William Fedus, et al., ICLR, 2018]

# Empirical Performance

- MLE frequently generates “I’m sorry”, “I don’t know”, etc. (corresponding to fuzzy images?)
- GAN generates longer and more complex responses.
- Find more comparison in the survey papers.
  - [Lu, et al., arXiv, 2018][Zhu, et al., arXiv, 2018]
- However, no strong evidence shows that GANs are better than MLE.
  - [Stanislau Semeniuta, et al., arXiv, 2018] [Guy Tevet, et al., arXiv, 2018] [Massimo Caccia, et al., arXiv, 2018]

# More Applications

- Supervised machine translation [Wu, et al., arXiv 2017][Yang, et al., arXiv 2017]
- Supervised abstractive summarization [Liu, et al., AAAI 2018]
- Image/video caption generation [Rakshith Shetty, et al., ICCV 2017][Liang, et al., arXiv 2017]
- Data augmentation for code-switching ASR [Mon-P-1-D] [Chang, et al., INTERSPEECH 2019]

If you are trying to generate some sequences, you can consider GAN.

# Outline of Part IV

Sequence Generation by GAN

Unsupervised Conditional Sequence Generation

- Text Style Transfer
- Unsupervised Abstractive Summarization
- Unsupervised Translation
- Unsupervised Speech Recognition

**Part I**

**Part III**

male

female


positive sentences


negative sentences

***Text Style Transfer***

document

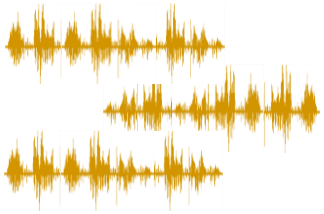
summary

***Unsupervised Abstractive Summarization***


Language 1

Language 2

***Unsupervised Translation***

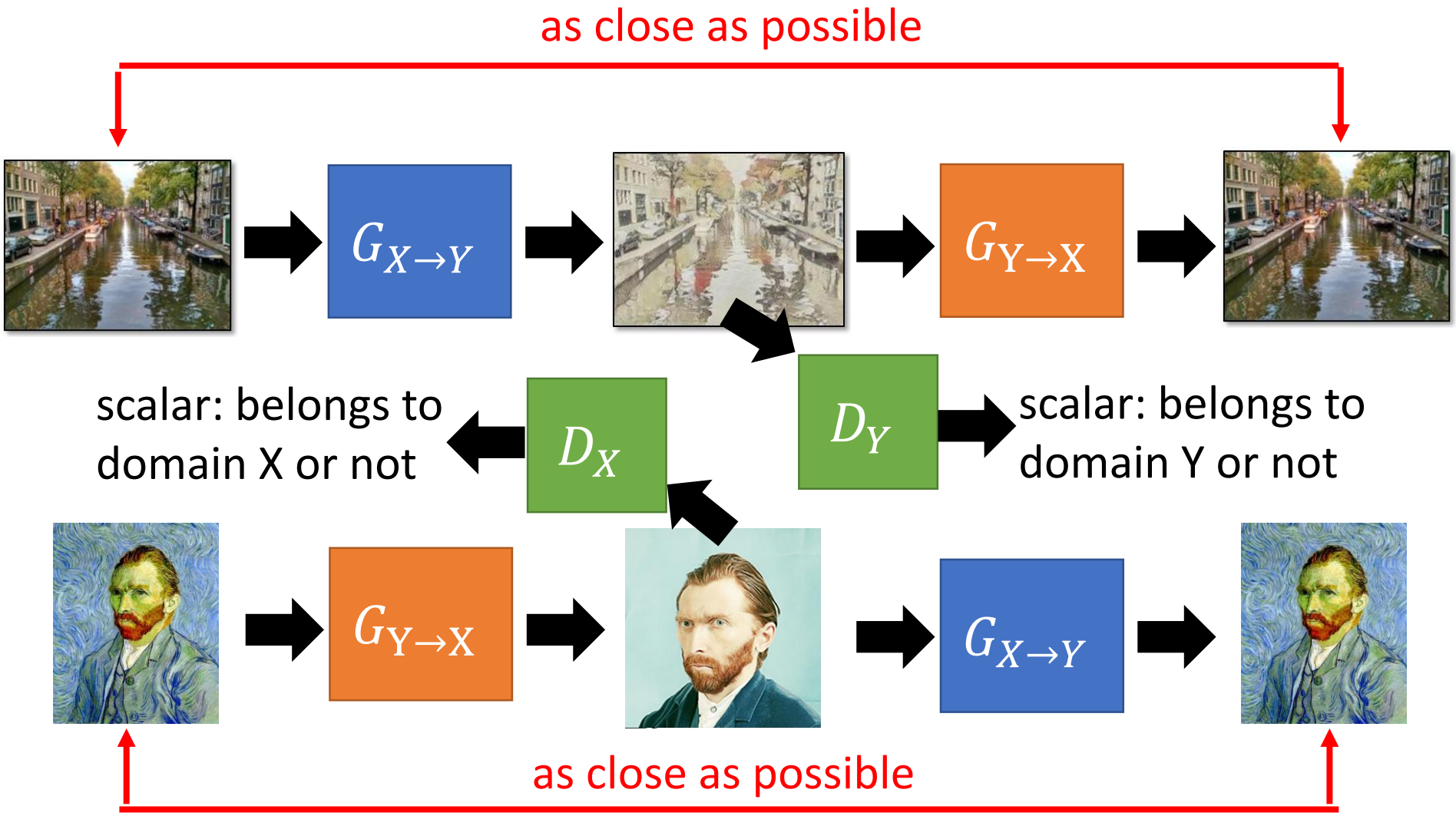


Audio

Text

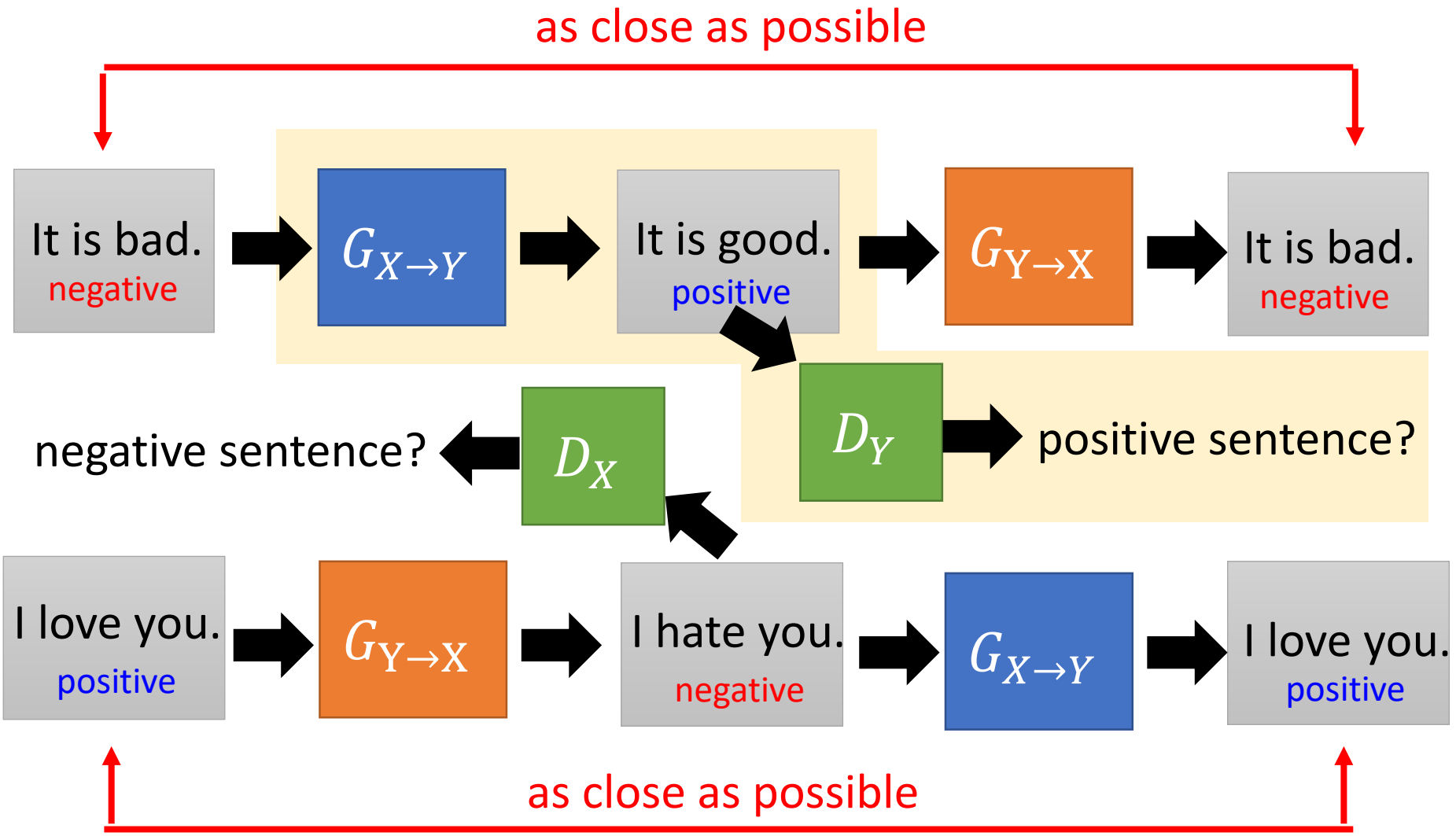
***Unsupervised ASR***

# 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

Non-differentiable Issue?
You already know how to deal with it.



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 dummy day -> it's a great day

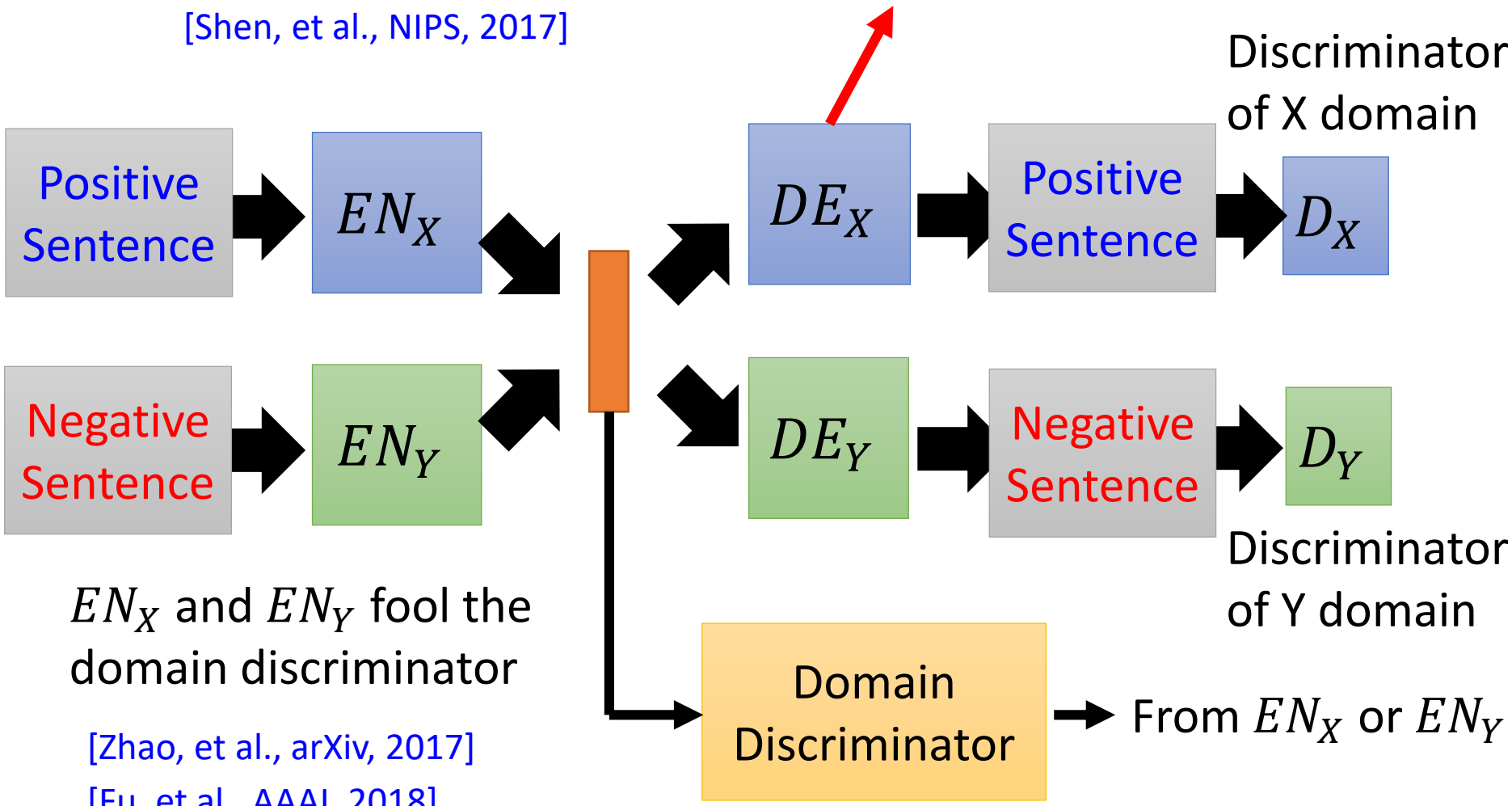
sorry for doing such a horrible thing -> thanks for doing a great thing

[Lee, et al., ICASSP, 2018]

# Shared Latent Space

Decoder hidden layer as discriminator input

[Shen, et al., NIPS, 2017]




$EN_X$ and $EN_Y$ fool the domain discriminator

[Zhao, et al., arXiv, 2017]

[Fu, et al., AAAI, 2018]

**Part I**

**Part III**

male

female

positive sentences

negative sentences

***Text Style Transfer***

document

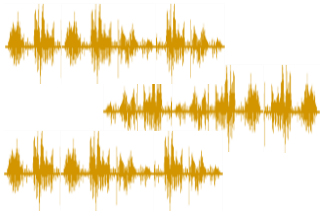
summary

***Unsupervised Abstractive Summarization***

Language 1

Language 2

***Unsupervised Translation***



Audio

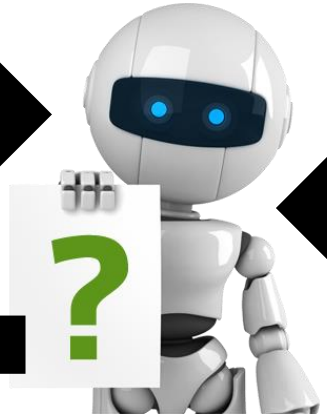
Text

***Unsupervised ASR***

# Abstractive Summarization

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


summary
(in its own words)




seq2seq


summary 1


summary 2


summary 3

Training Data

**Supervised: We need lots of labelled training data.**

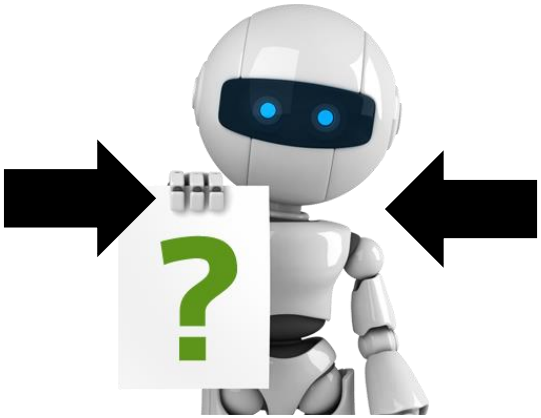
# Unsupervised Abstractive Summarization

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

summary 1

summary 2

summary 3




seq2seq

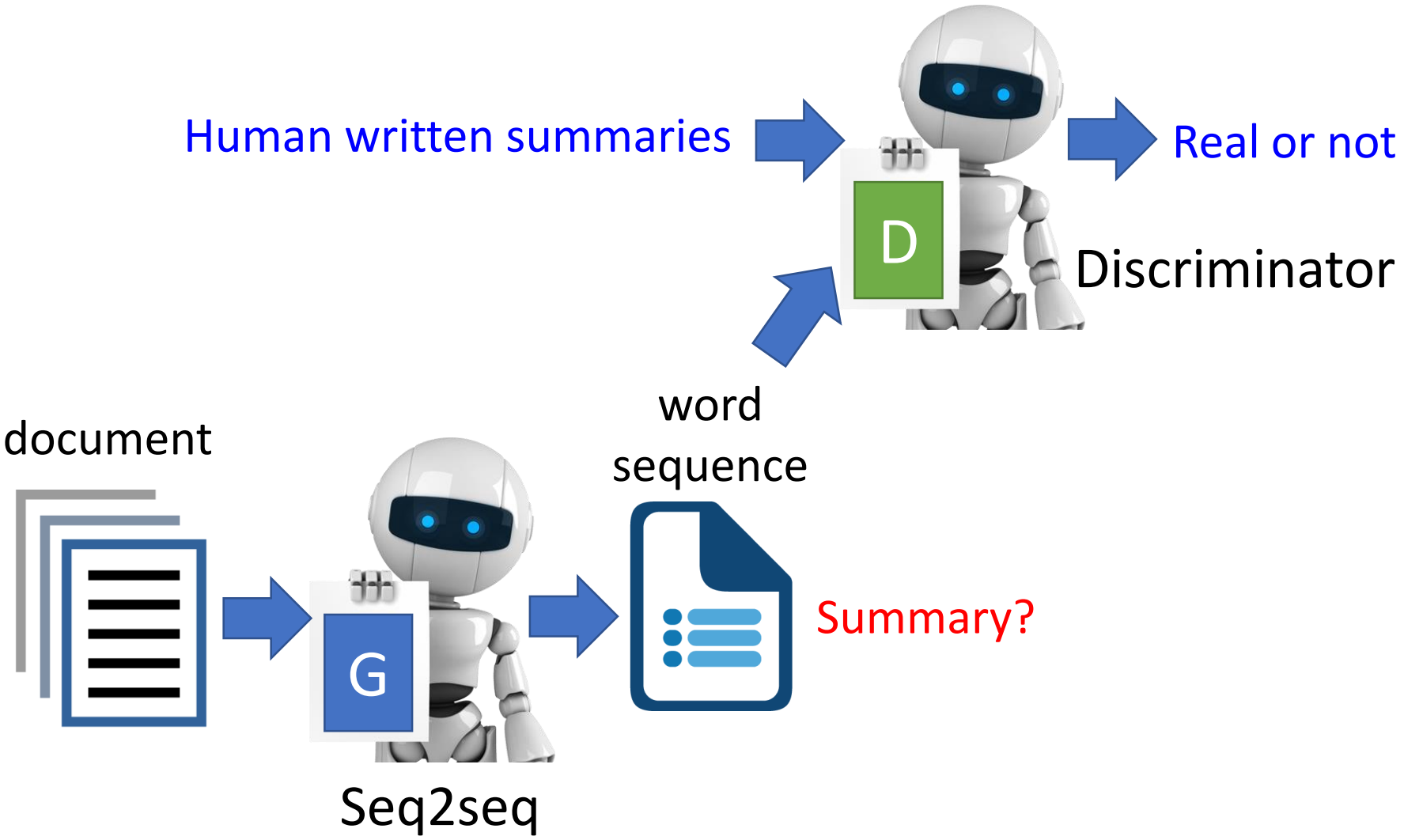

document

***Domain Y***

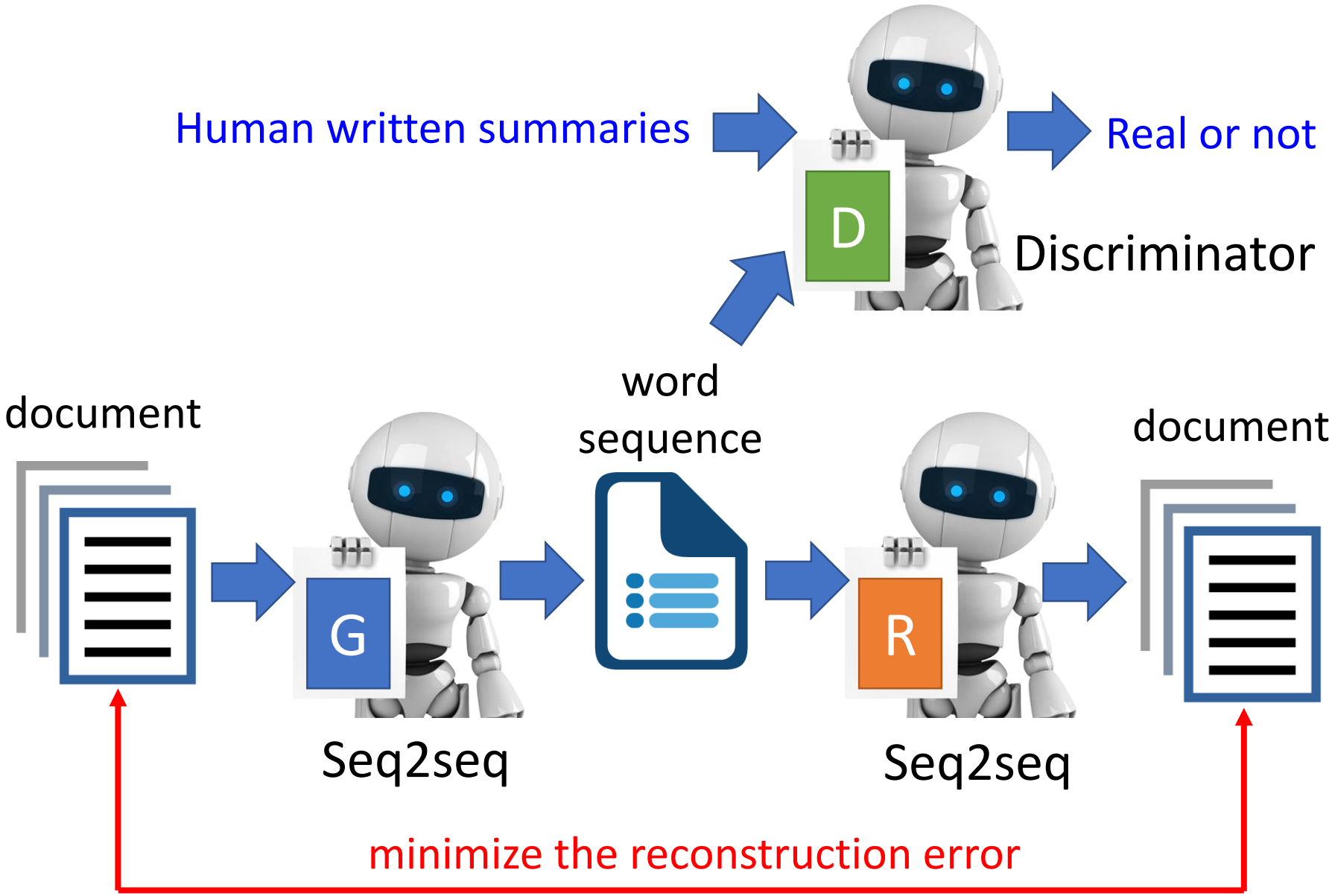
[Wang, et al., EMNLP, 2018]

***Domain X***

# *Unsupervised Abstractive Summarization*



# *Unsupervised Abstractive Summarization*

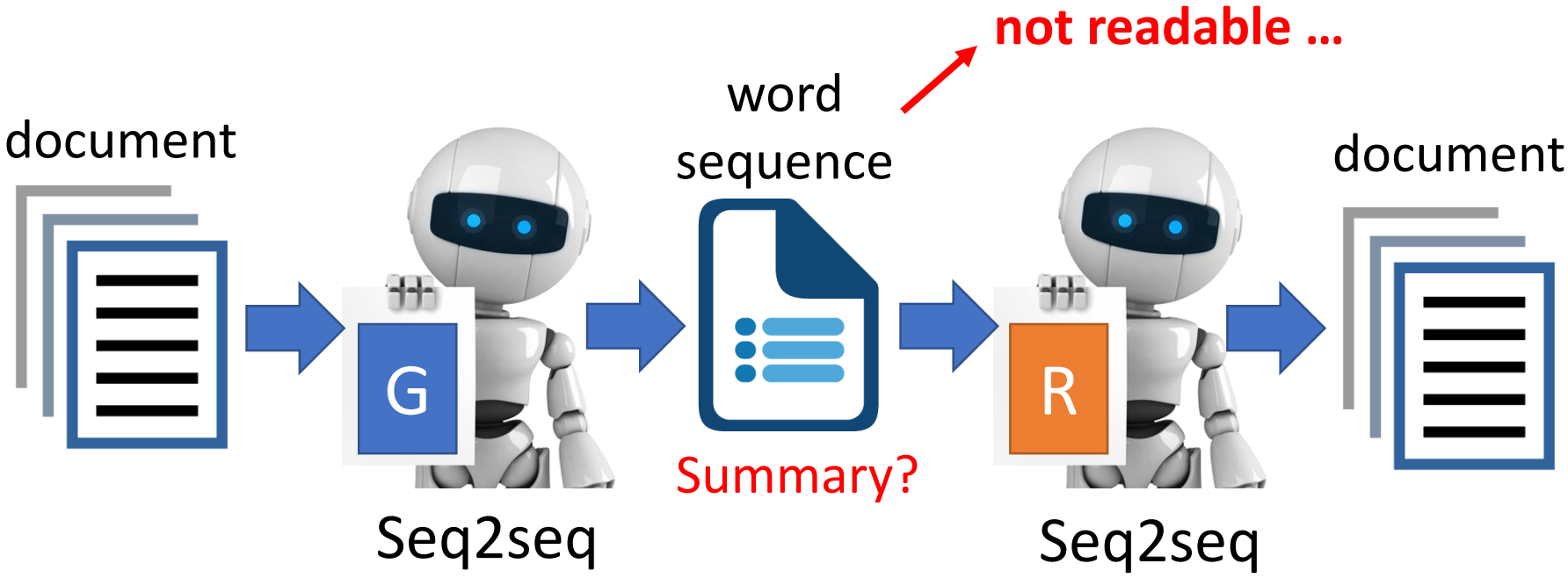


# Unsupervised Abstractive Summarization

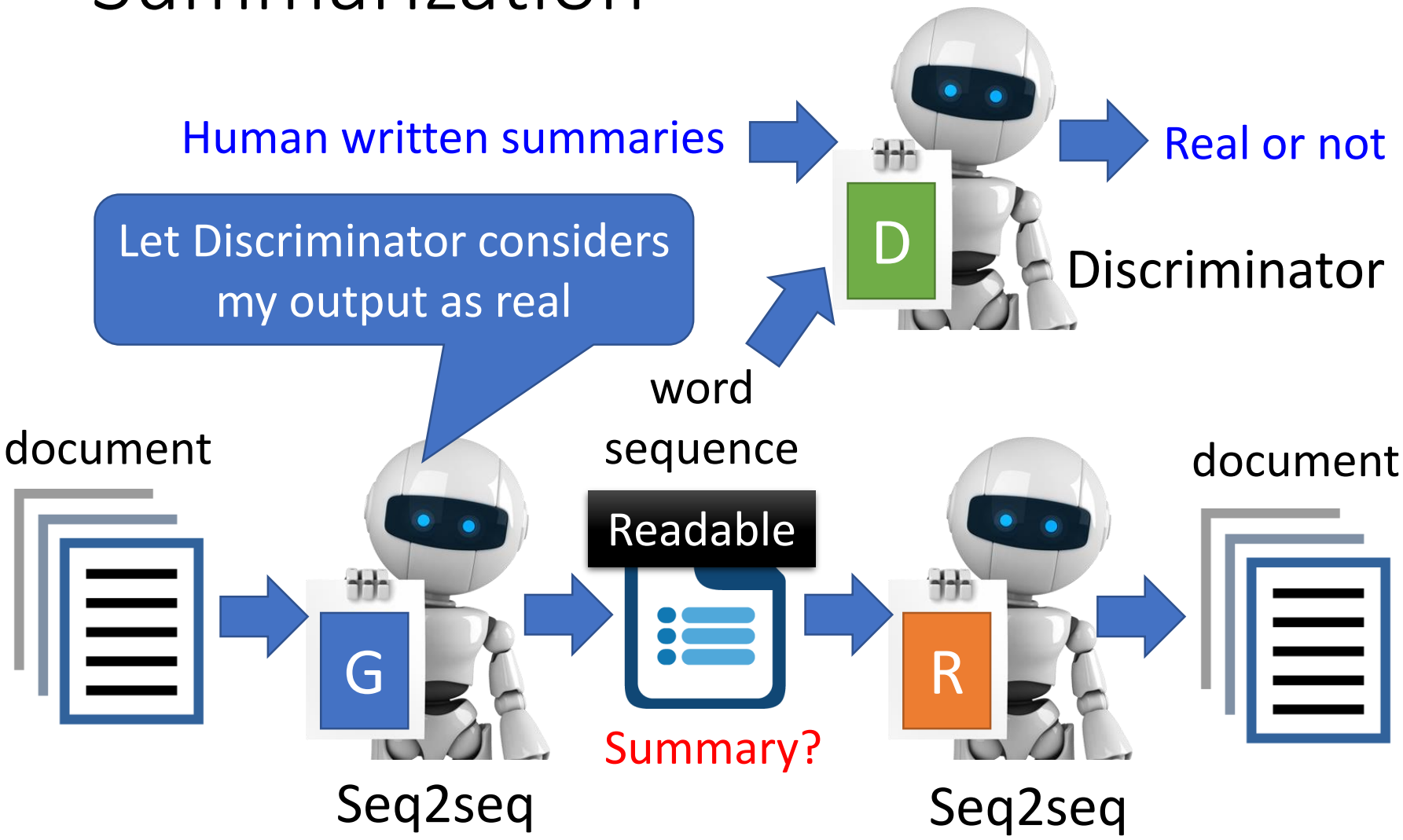
Only need a lot of documents to train the model

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

Using a sequence of words as latent representation.



# Unsupervised Abstractive Summarization



[Wang, Lee, EMNLP 2018]

# Experimental results

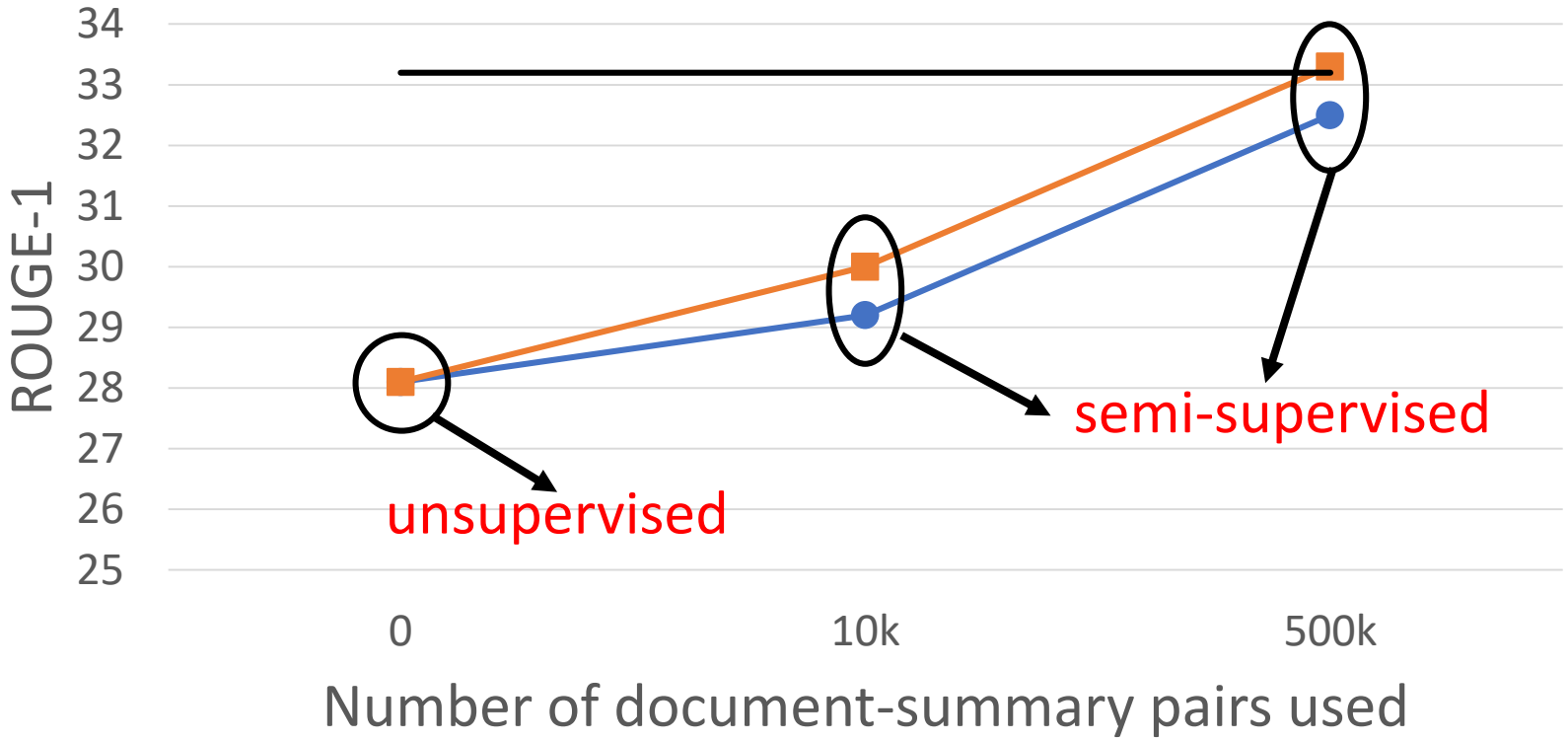
English Gigaword (Document title as summary)

| | 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

[Wang, Lee, EMNLP 2018]



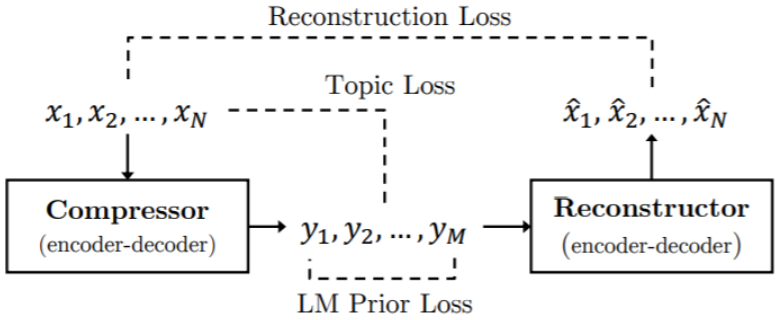

Approaches to deal with the discrete issue.

3.8M pairs are used.

# More Unsupervised Summarization

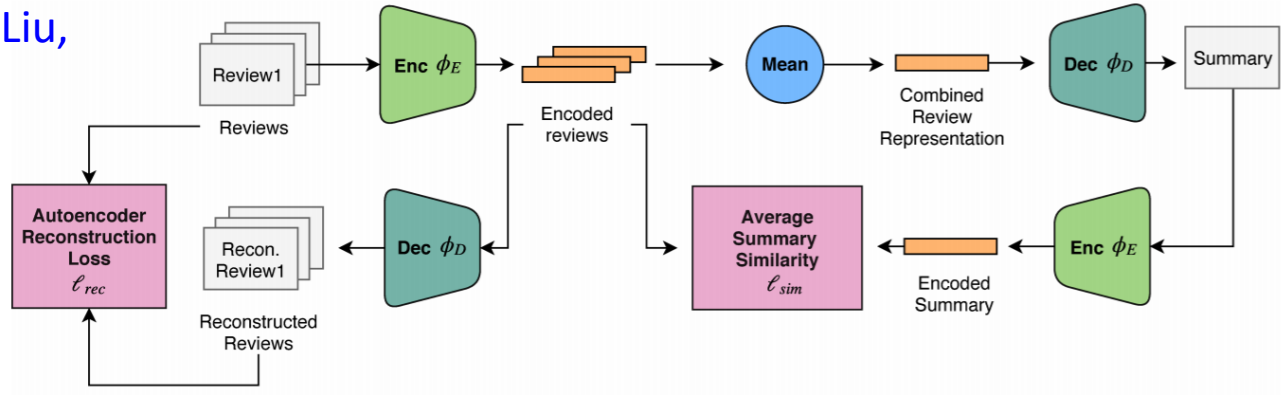
- Unsupervised summarization with language prior

[Christos Baziotis, etc al., NAACL 2019]



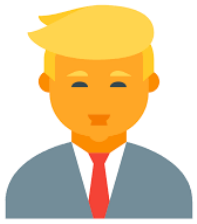

- Unsupervised multi-document summarization

[Eric Chu, Peter Liu, ICML 2019]



# *Dialogue Response Generation*

[Su, et al., INTERSPEECH, 2019]
(Thu-P-9-C)

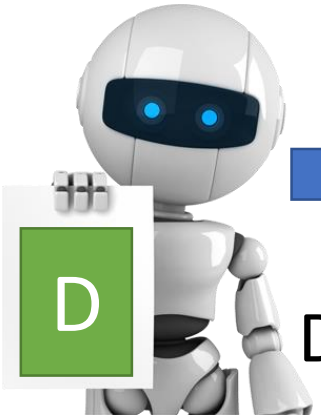


What Trump has said

Make the US great again

I would build a great wall

you are fired



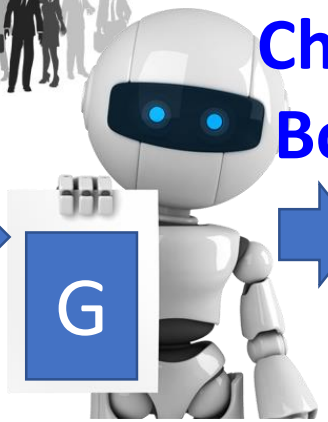

Said by Trump?

Discriminator

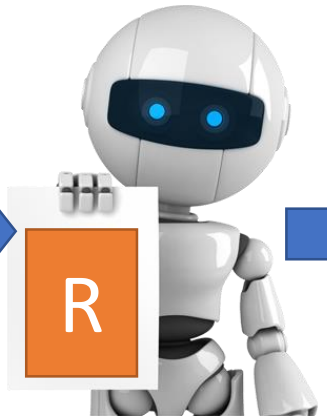
General Dialogues

**Chat Bot**

Input Sentence




Generated Response




Input Sentence (Reconstruct)

minimize the reconstruction error

**Part I**

**Part III**

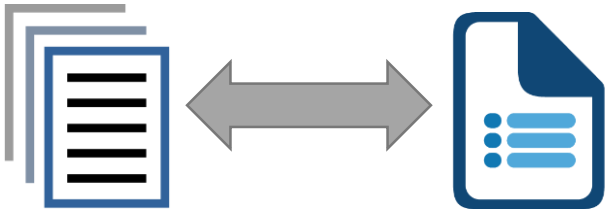
male

female


positive sentences


negative sentences

**Text Style Transfer**

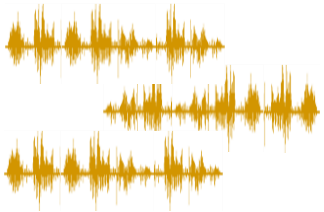


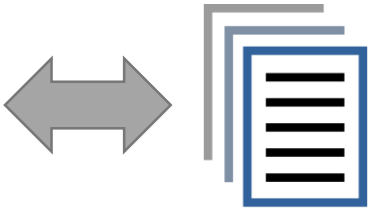
document summary

**Unsupervised Abstractive Summarization**


Language 1 Language 2

**Unsupervised Translation**

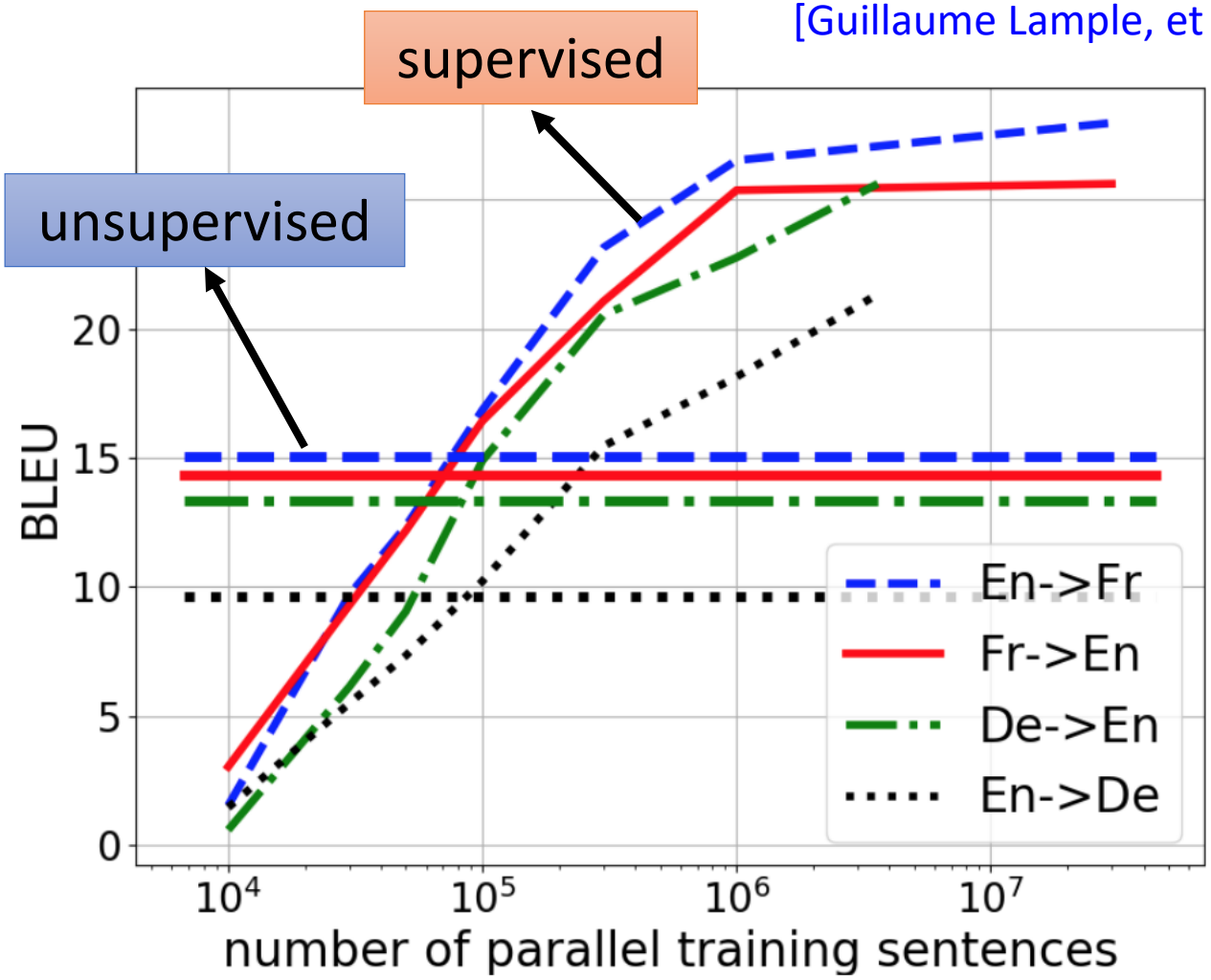




Audio Text

**Unsupervised ASR**

[Alexis Conneau, et al., ICLR, 2018]
[Guillaume Lample, et al., ICLR, 2018]




**Unsupervised** learning with 10M sentences

=

**Supervised** learning with 100K sentence pairs

**Part I**

**Part III**

male

female

positive sentences


negative sentences


**Text Style Transfer**

document

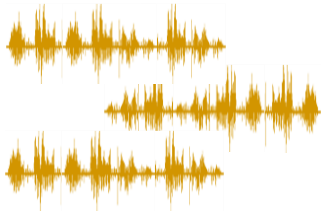
summary

***Unsupervised Abstractive Summarization***


Language 1

Language 2

***Unsupervised Translation***



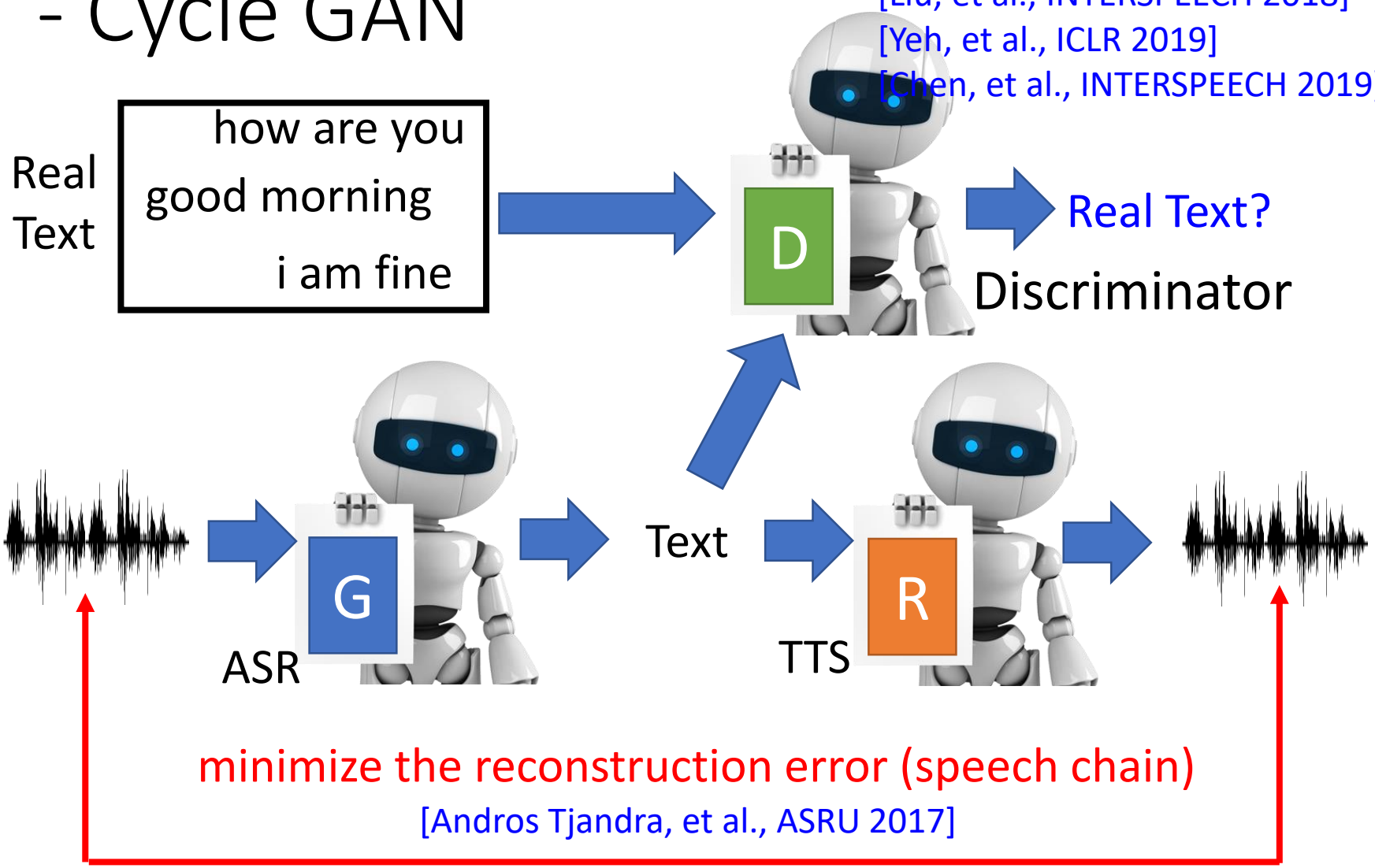
Audio

Text

***Unsupervised ASR***

# Towards Unsupervised ASR - Cycle GAN

[Liu, et al., INTERSPEECH 2018]
[Yeh, et al., ICLR 2019]
[Chen, et al., INTERSPEECH 2019]




[Andros Tjandra, et al., ASRU 2017]

# Towards Unsupervised ASR - Cycle GAN

- Unsupervised setting on TIMIT (text and audio are unpair, text is not the transcription of audio)
  - 63.6% PER (oracle boundaries) [Liu, et al., INTERSPEECH 2018]
  - 41.6% PER (automatic segmentation) [Yeh, et al., ICLR 2019]
  - 33.1% PER (automatic segmentation)

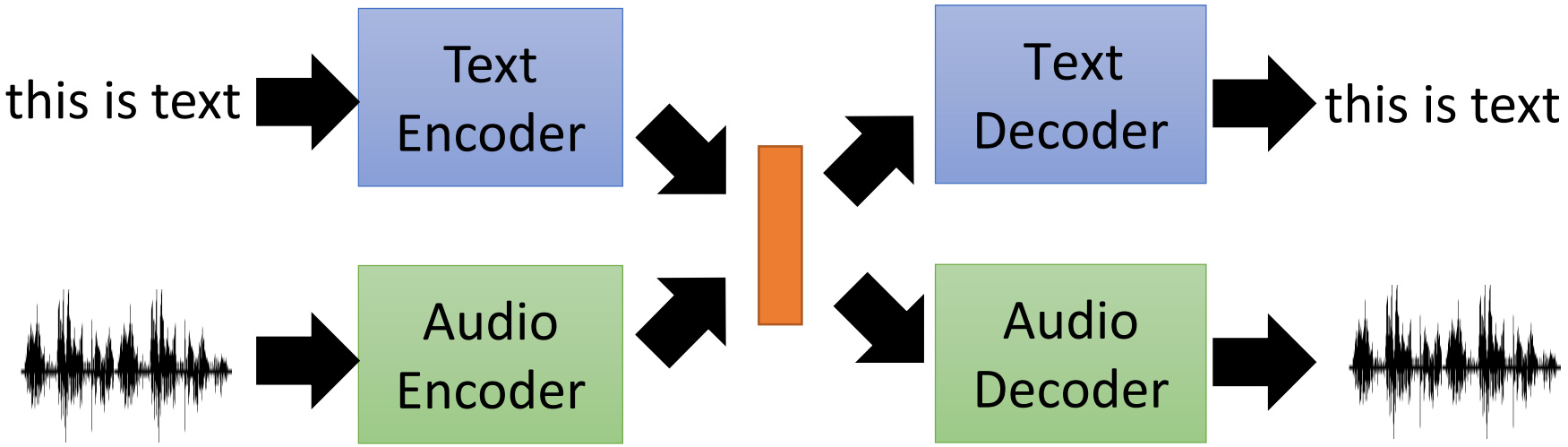
    (Tue-P-4-B)[Chen, et al., INTERSPEECH 2019]

- Semi-supervised setting on Librispeech

  [Liu, et al., ICASSP 2019]
  [Tomoki Hayashi, et al., SLT 2018]
  [Takaaki Hori, et al., ICASSP 2019]
  [Murali Karthick Baskar, et al., INTERSPEECH 2019]

# Towards Unsupervised ASR - Shared Latent Space

[Chen, et al., SLT 2018]




Unsupervised setting on Librispeech: 76.3% WER

[Chung, et al., NIPS 2018]

Unsupervised speech translation is also possible!

[Chung, et al., ICASSP 2019]

WSJ with 2.5 hours paired data: 64.6% WER

[Jennifer Drexler, et al., SLT 2018]

LJ speech with 20 mins paired data: 11.7% PER [Ren, et al., ICML 2019]

# Outline of Part IV

Sequence Generation by GAN

Unsupervised Conditional Sequence Generation

- Text Style Transfer
- Unsupervised Abstractive Summarization
- Unsupervised Translation
- Unsupervised Speech Recognition

# To Learn More …

You can learn more from the YouTube Channel

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

(in Mandarin)

# Reference

- **Sequence Generation**
  - Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, Dan Jurafsky, Deep Reinforcement Learning for Dialogue Generation, EMNLP, 2016
  - Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, Dan Jurafsky, Adversarial Learning for Neural Dialogue Generation, EMNLP, 2017
  - Matt J. Kusner, José Miguel Hernández-Lobato, GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution, arXiv 2016
  - Tong Che, Yanran Li, Ruixiang Zhang, R Devon Hjelm, Wenjie Li, Yangqiu Song, Yoshua Bengio, Maximum-Likelihood Augmented Discrete Generative Adversarial Networks, arXiv 2017
  - Lantao Yu, Weinan Zhang, Jun Wang, Yong Yu, SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient, AAAI 2017
  - Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Pal, Aaron Courville, Adversarial Generation of Natural Language, arXiv, 2017
  - Ofir Press, Amir Bar, Ben Bogin, Jonathan Berant, Lior Wolf, Language Generation with Recurrent Generative Adversarial Networks without Pre-training, ICML workshop, 2017

# Reference


- **Sequence Generation**
  - Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, Xiaolong Wang, Zhuoran Wang, Chao Qi , Neural Response Generation via GAN with an Approximate Embedding Layer, EMNLP, 2017
  - Alex Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron Courville, Yoshua Bengio, Professor Forcing: A New Algorithm for Training Recurrent Networks, NIPS, 2016
  - Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, Lawrence Carin, Adversarial Feature Matching for Text Generation, ICML, 2017
  - Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, Jun Wang, Long Text Generation via Adversarial Training with Leaked Information, AAAI, 2018
  - Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, Ming-Ting Sun, Adversarial Ranking for Language Generation, NIPS, 2017
  - William Fedus, Ian Goodfellow, Andrew M. Dai, MaskGAN: Better Text Generation via Filling in the_____, ICLR, 2018

# Reference


- **Sequence Generation**
  - Yi-Lin Tuan, Hung-Yi Lee, Improving Conditional Sequence Generative Adversarial Networks by Stepwise Evaluation, TASLP, 2019
  - Jingjing Xu, Xuancheng Ren, Junyang Lin, Xu Sun, Diversity-Promoting GAN: A Cross-Entropy Based Generative Adversarial Network for Diversified Text Generation, EMNLP, 2018
  - Sidi Lu, Yaoming Zhu, Weinan Zhang, Jun Wang, Yong Yu, Neural Text Generation: Past, Present and Beyond, arXiv, 2018
  - Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, Yong Yu, Texygen: A Benchmarking Platform for Text Generation Models, arXiv, 2018
  - Stanislau Semeniuta, Aliaksei Severyn, Sylvain Gelly, On Accurate Evaluation of GANs for Language Generation, arXiv, 2018
  - Guy Tevet, Gavriel Habib, Vered Shwartz, Jonathan Berant, Evaluating Text GANs as Language Models, arXiv, 2018
  - Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, Laurent Charlin, Language GANs Falling Short, arXiv, 2018

# Reference


- **Sequence Generation**
  - Zhen Yang, Wei Chen, Feng Wang, Bo Xu, Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets, NAACL, 2018
  - Lijun Wu, Yingce Xia, Li Zhao, Fei Tian, Tao Qin, Jianhuang Lai, Tie-Yan Liu, Adversarial Neural Machine Translation, arXiv 2017
  - Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, Hongyan Li, Generative Adversarial Network for Abstractive Text Summarization, AAAI 2018
  - Rakshith Shetty, Marcus Rohrbach, Lisa Anne Hendricks, Mario Fritz, Bernt Schiele, Speaking the Same Language: Matching Machine to Human Captions by Adversarial Training, ICCV 2017
  - Xiaodan Liang, Zhiting Hu, Hao Zhang, Chuang Gan, Eric P. Xing, Recurrent Topic-Transition GAN for Visual Paragraph Generation, arXiv 2017
  - Weili Nie, Nina Narodytska, Ankit Patel, RelGAN: Relational Generative Adversarial Networks for Text Generation, ICLR 2019

# Reference

- **Sequence Generation**
  - Ching-Ting Chang, Shun-Po Chuang, Hung-Yi Lee, "Code-switching Sentence Generation by Generative Adversarial Networks and its Application to Data Augmentation", INTERSPEECH 2019
  - Cyprien de Masson d'Autume, Mihaela Rosca, Jack Rae, Shakir Mohamed, Training language GANs from Scratch, arXiv 2019

# Reference


- **Text Style Transfer**
  - Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, Rui Yan, Style Transfer in Text: Exploration and Evaluation, AAAI, 2018
  - Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola, Style Transfer from Non-Parallel Text by Cross-Alignment, NIPS 2017
  - Chih-Wei Lee, Yau-Shian Wang, Tsung-Yuan Hsu, Kuan-Yu Chen, Hung-Yi Lee, Lin-shan Lee, Scalable Sentiment for Sequence-to-sequence Chatbot Response with Performance Analysis, ICASSP, 2018
  - Junbo (Jake) Zhao, Yoon Kim, Kelly Zhang, Alexander M. Rush, Yann LeCun, Adversarially Regularized Autoencoders, arxiv, 2017
  - Feng-Guang Su, Aliyah Hsu, Yi-Lin Tuan and Hung-yi Lee, "Personalized Dialogue Response Generation Learned from Monologues", INTERSPEECH, 2019

# Reference

- **Unsupervised Abstractive Summarization**
  - Yau-Shian Wang, Hung-Yi Lee, "Learning to Encode Text as Human-Readable Summaries using Generative Adversarial Networks", EMNLP, 2018
  - Eric Chu, Peter Liu, “MeanSum: A Neural Model for Unsupervised Multi-Document Abstractive Summarization”, ICML, 2019
  - Christos Baziotis, Ion Androutsopoulos, Ioannis Konstas, Alexandros Potamianos, “SEQ^3: Differentiable Sequence-to-Sequence-to-Sequence Autoencoder for Unsupervised Abstractive Sentence Compression”, NAACL 2019

# Reference

- **Unsupervised Machine Translation**
  - Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, Hervé Jégou, Word Translation Without Parallel Data, ICRL 2018
  - Guillaume Lample, Ludovic Denoyer, Marc'Aurelio Ranzato, Unsupervised Machine Translation Using Monolingual Corpora Only, ICRL 2018

# Reference


- **Unsupervised Speech Recognition**
  - Alexander H. Liu, Hung-yi Lee, Lin-shan Lee, Adversarial Training of End-to-end Speech Recognition Using a Criticizing Language Model, ICASSP 2018
  - Da-Rong Liu, Kuan-Yu Chen, Hung-Yi Lee, Lin-shan Lee, Completely Unsupervised Phoneme Recognition by Adversarially Learning Mapping Relationships from Audio Embeddings, INTERSPEECH, 2018
  - Kuan-yu Chen, Che-ping Tsai, Da-Rong Liu, Hung-yi Lee and Lin-shan Lee, "Completely Unsupervised Phoneme Recognition By A Generative Adversarial Network Harmonized With Iteratively Refined Hidden Markov Models", INTERSPEECH, 2019
  - Yi-Chen Chen, Sung-Feng Huang, Chia-Hao Shen, Hung-yi Lee, Lin-shan Lee, "Phonetic-and-Semantic Embedding of Spoken Words with Applications in Spoken Content Retrieval", SLT, 2018
  - Chih-Kuan Yeh, Jianshu Chen, Chengzhu Yu, Dong Yu, Unsupervised Speech Recognition via Segmental Empirical Output Distribution Matching, ICLR, 2019

# Reference


- **Unsupervised Speech Recognition**
  - Takaaki Hori, Ramon Astudillo, Tomoki Hayashi, Yu Zhang, Shinji Watanabe, Jonathan Le Roux, Cycle-consistency training for end-to-end speech recognition, ICASSP 2019
  - Murali Karthick Baskar, Shinji Watanabe, Ramon Astudillo, Takaaki Hori, Lukáš Burget, Jan Černocký, Semi-supervised Sequence-to-sequence ASR using Unpaired Speech and Text, INTERSPEECH 2019
  - Andros Tjandra, Sakriani Sakti, Satoshi Nakamura, Listening while Speaking: Speech Chain by Deep Learning, ASRU 2017
  - Yu-An Chung, Wei-Hung Weng, Schrasing Tong, James Glass, Unsupervised Cross-Modal Alignment of Speech and Text Embedding Spaces, NIPS, 2018
  - Yu-An Chung, Wei-Hung Weng, Schrasing Tong, James Glass, Towards Unsupervised Speech-to-Text Translation, ICASSP 2019
  - Yi Ren, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, Tie-Yan Liu, Almost Unsupervised Text to Speech and Automatic Speech Recognition, ICML 2019

# Reference

- **Unsupervised Speech Recognition**
  - Shigeki Karita , Shinji Watanabe, Tomoharu Iwata, Atsunori Ogawa, Marc Delcroix, Semi-Supervised End-to-End Speech Recognition, INTERSPEECH, 2018
  - Jennifer Drexler, James R. Glass, “Combining End-to-End and Adversarial Training for Low-Resource Speech Recognition”, SLT 2018
  - Tomoki Hayashi, Shinji Watanabe, Yu Zhang, Tomoki Toda, Takaaki Hori, Ramon Astudillo, Kazuya Takeda, Back-Translation-Style Data Augmentation for End-to-End ASR, SLT, 2018

Please download the latest slides here:
http://speech.ee.ntu.edu.tw/~tlkagk/GAN_3hour.pdf