SWINBURNE UNIVERSITY OF TECHNOLOGY

COS30082
Applied Machine Learning

Week 9
Generative Deep Learning

What is Generative models?

SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS

- In supervised learning, we have data x and target output (label) y, and the goal is to learn a function to map x to y e.g. regression, classification, object detection.
- While in unsupervised learning, there are no labels and the goal is to find some underlying hidden structure of the data e.g. clustering, dimensionality reduction, feature learning.
- A Generative Model is a powerful way of **learning any kind of data distribution using unsupervised learning**.
- All types of generative models aim at learning the true data distribution of the training set so as to **generate new data points** with some variations.

Generative and Discriminative model

SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS

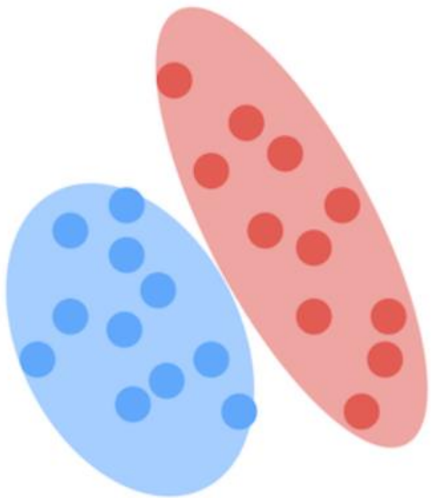
A story of lion and elephant

- A father has two kids, Kid A and Kid B. Kid A has a special character whereas he can learn everything in depth. Kid B have a special character whereas he can only learn the differences between what he saw.
- One fine day, The father takes two of his kids (Kid A and Kid B) to a zoo. This zoo is a very small one and has only two kinds of animals say a lion and an elephant. After they came out of the zoo, the father showed them an animal and asked both of them "is this animal a lion or an elephant?"
- The Kid A, the kid suddenly draw the image of lion and elephant in a piece of paper based on what he saw inside the zoo. He compared both the images with the animal standing before and answered based on the closest match of image & animal, he answered: "The animal is Lion".
- The Kid B knows only the differences, based on different properties learned, he answered: "The animal is a Lion".
- Here, we can see both of them is finding the kind of animal, but the way of learning and the way of finding answer is entirely different. In Machine Learning, We generally call Kid A as a Generative Model & Kid B as a Discriminative Model.

Generative and Discriminative model

SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS

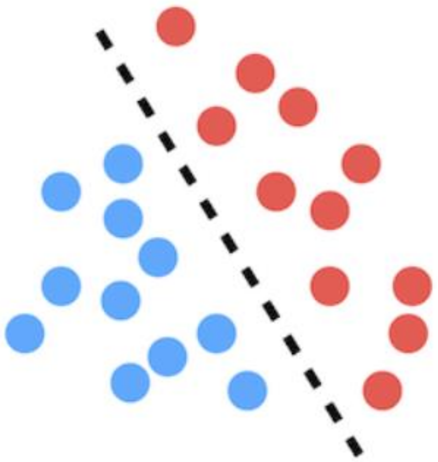
Generative model



Informally:
Generative models can generate new data instances.

Formally:
Generative models capture the joint probability p(X, Y), or just p(X) if there are no labels.

Discriminative model



Informally:
Discriminative models discriminate between different kinds of data instances.

Formally:
Discriminative models capture the conditional probability p(Y | X).

Introduction of
Generative Adversarial network (GAN)

Basic Idea of GAN - Generator

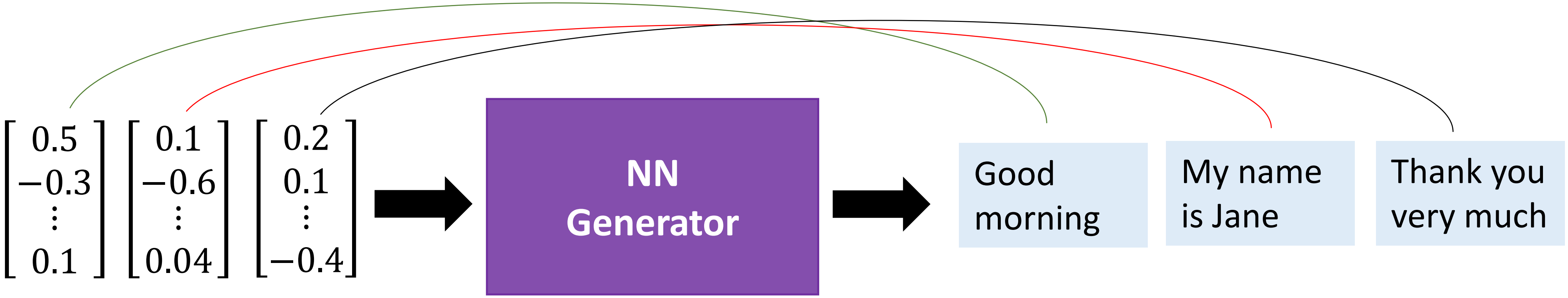
- Image generation

$$\begin{bmatrix} 0.5 \\ -0.3 \\ \vdots \\ 0.1 \end{bmatrix} \begin{bmatrix} 0.1 \\ -0.6 \\ \vdots \\ 0.04 \end{bmatrix} \begin{bmatrix} 0.2 \\ 0.1 \\ \vdots \\ -0.4 \end{bmatrix} \rightarrow \boxed{\textbf{NN Generator}} \rightarrow$$

In a specific range

Basic Idea of GAN - Generator

- Sentence generation



Basic Idea of GAN - Generator

It is a neural network (NN), or function

vector → **Generator** → image

Each dimension of input vector represent some characteristics

$$\begin{bmatrix} 0.5 \\ -0.3 \\ \vdots \\ 0.1 \\ 0.4 \end{bmatrix} \rightarrow \textbf{Generator} \rightarrow$$

Hair texture

$$\begin{bmatrix} 0.5 \\ -0.3 \\ \vdots \\ \boxed{0.2} \\ 0.4 \end{bmatrix} \rightarrow \textbf{Generator} \rightarrow$$

Open mouth

$$\begin{bmatrix} 0.5 \\ -0.3 \\ \vdots \\ 0.2 \\ \boxed{0.8} \end{bmatrix} \rightarrow \textbf{Generator} \rightarrow$$

Basic Idea of GAN - Discriminator

It is a neural network (NN), or function

image → **Discriminator** → scalar

Larger value means real, smaller value means fake

 → **Discriminator** → 1.0

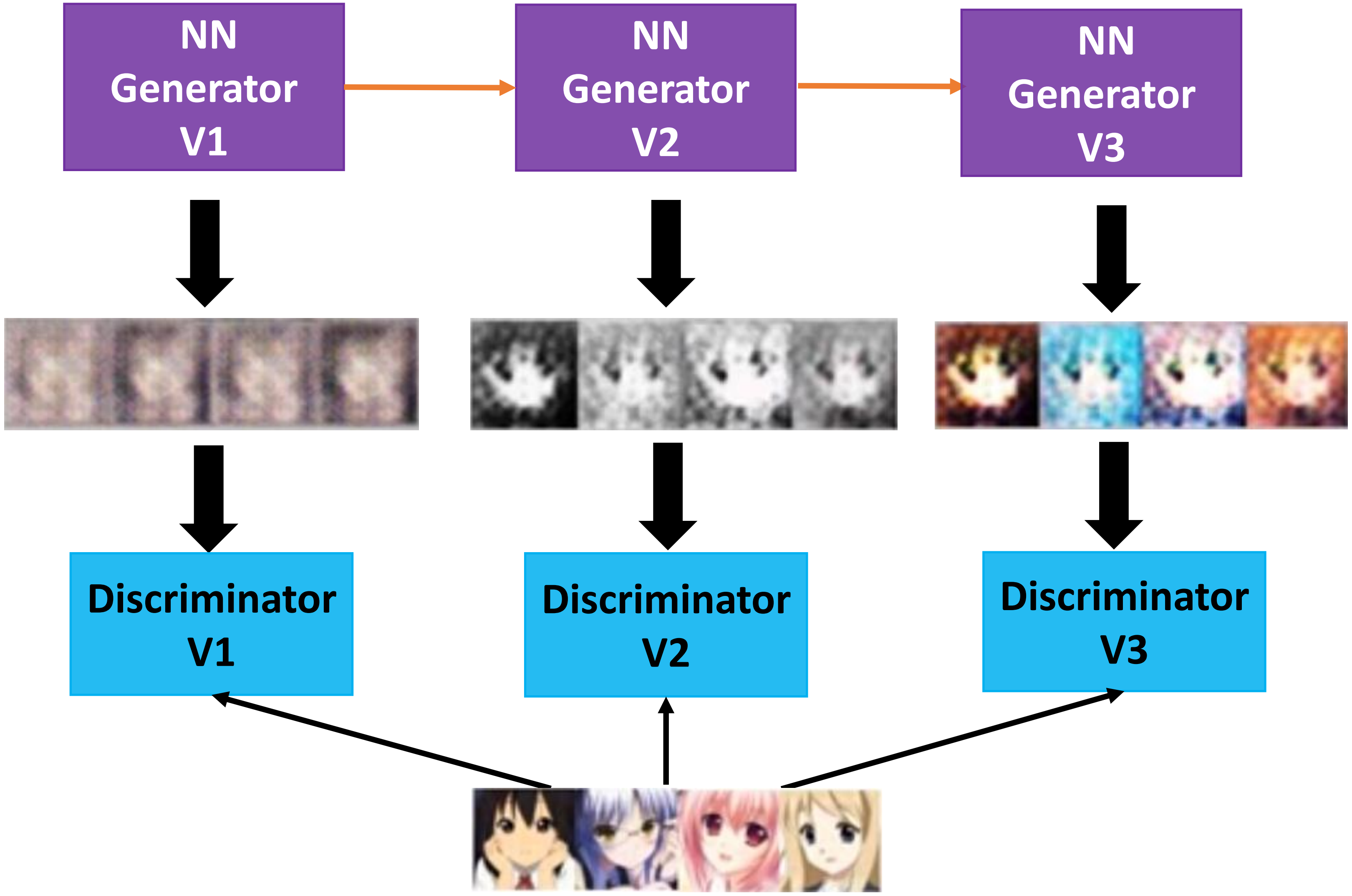
 → **Discriminator** → 1.0

 → **Discriminator** → 0.1

 → **Discriminator** → 0.1

Generator + Discriminator

Example: Manga generation using GAN



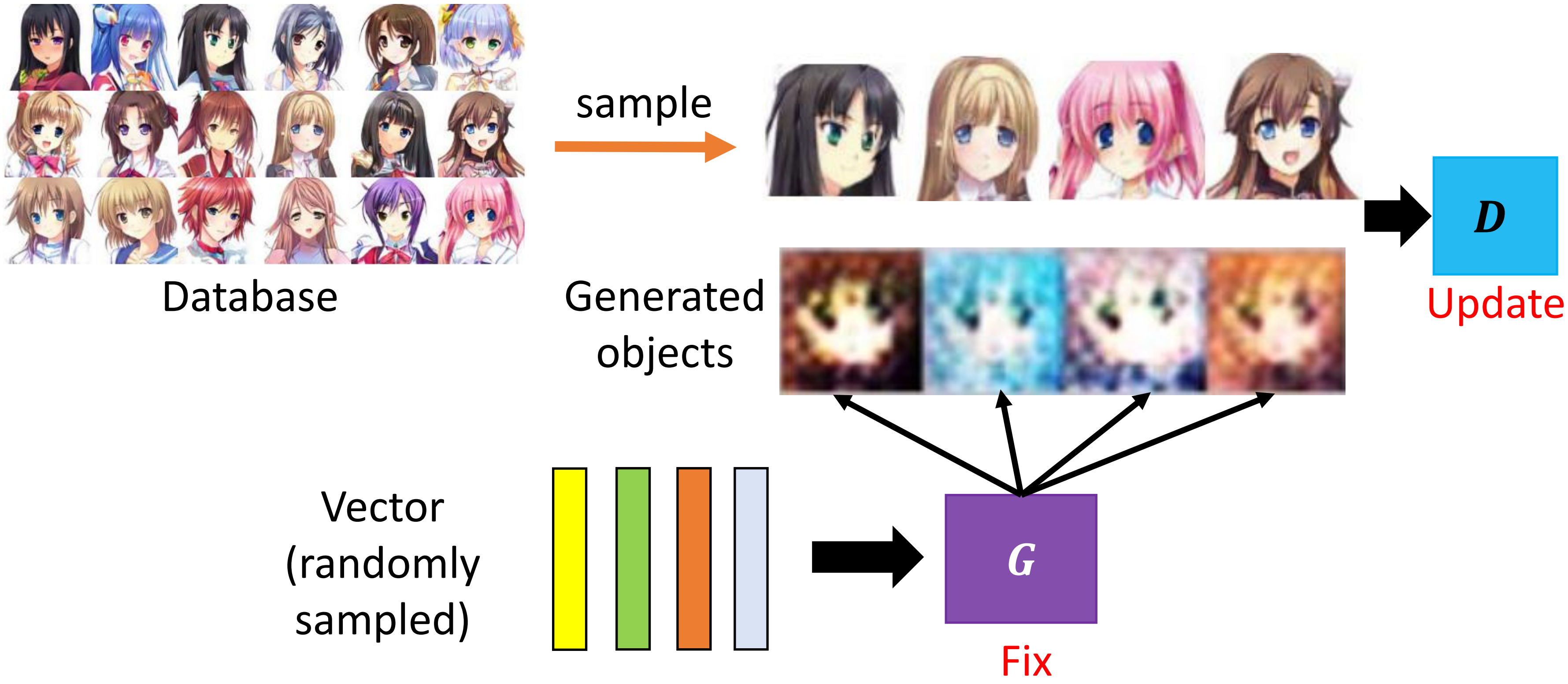
Training algorithm overview

- Initialize generator and discriminator

Generator Discriminator

- In each training iteration:

Step 1: Fix generator G and update discriminator D.



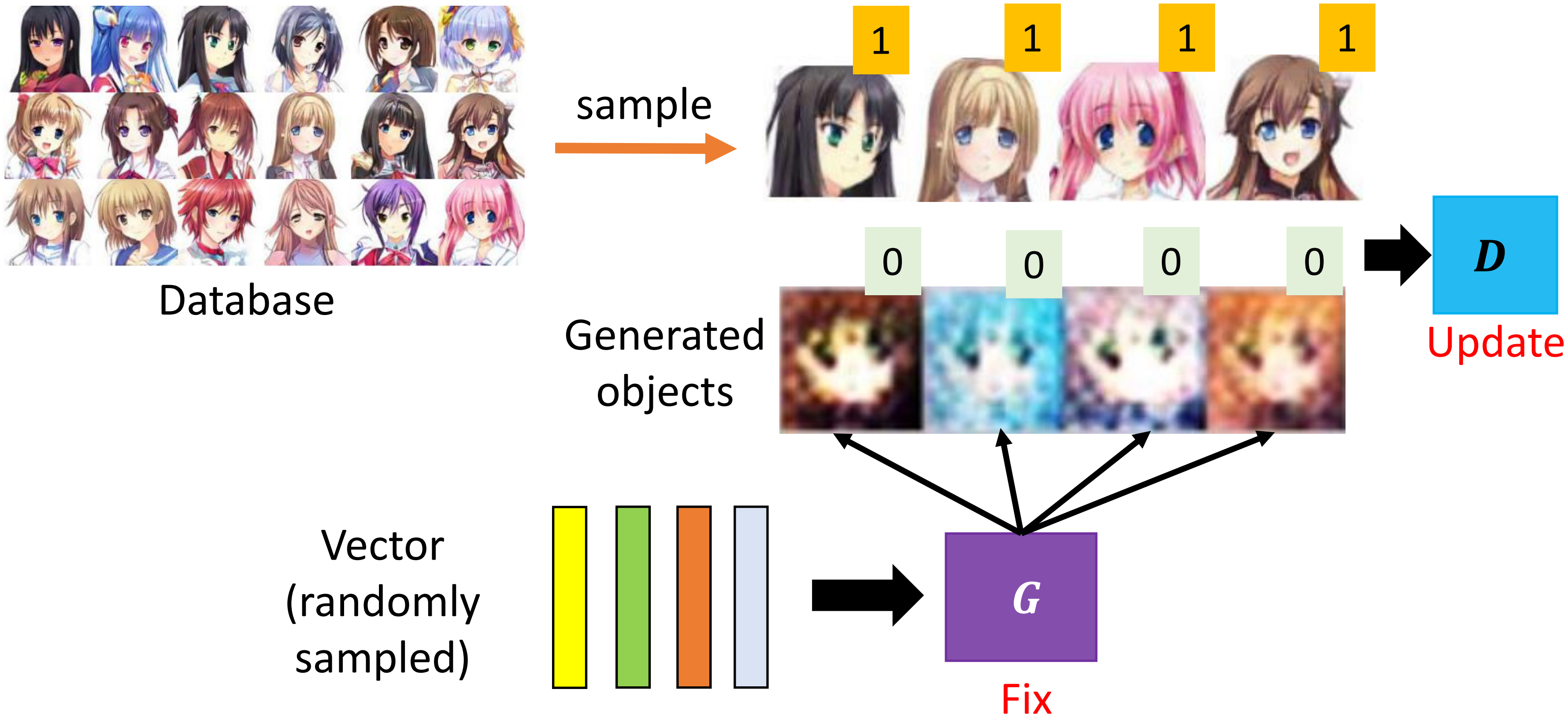
Training algorithm overview

- Initialize generator and discriminator

Generator   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

Training algorithm overview

SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS

- Initialize generator and discriminator
- In each training iteration:

Generator

Discriminator

Step 2: Fix discriminator D and update generator G

Generator learns to “fool” the discriminator

The goal is to increase the confidence score of the discriminator on the prediction of the real image.

Training algorithm overview

SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS

- Initialize generator and discriminator

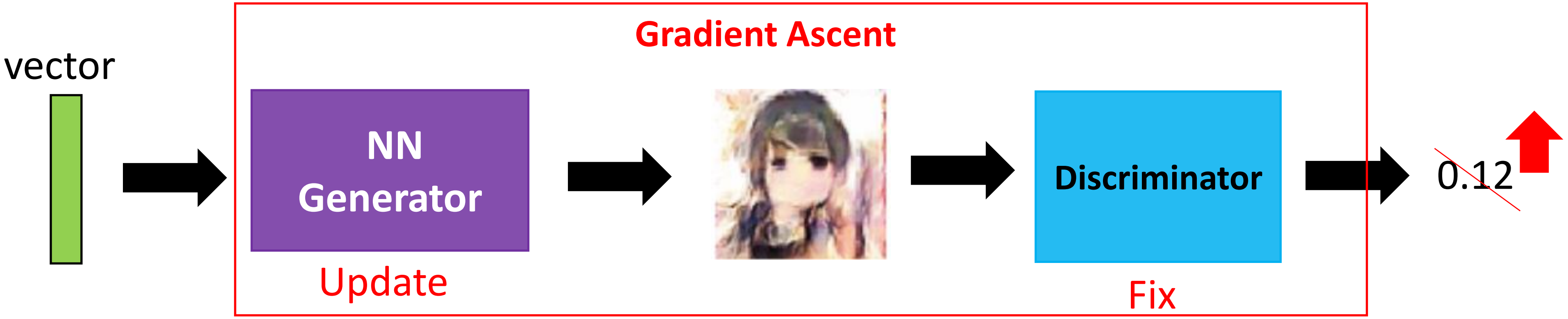
Generator

Discriminator

- In each training iteration:

Step 2: Fix discriminator D and update generator G

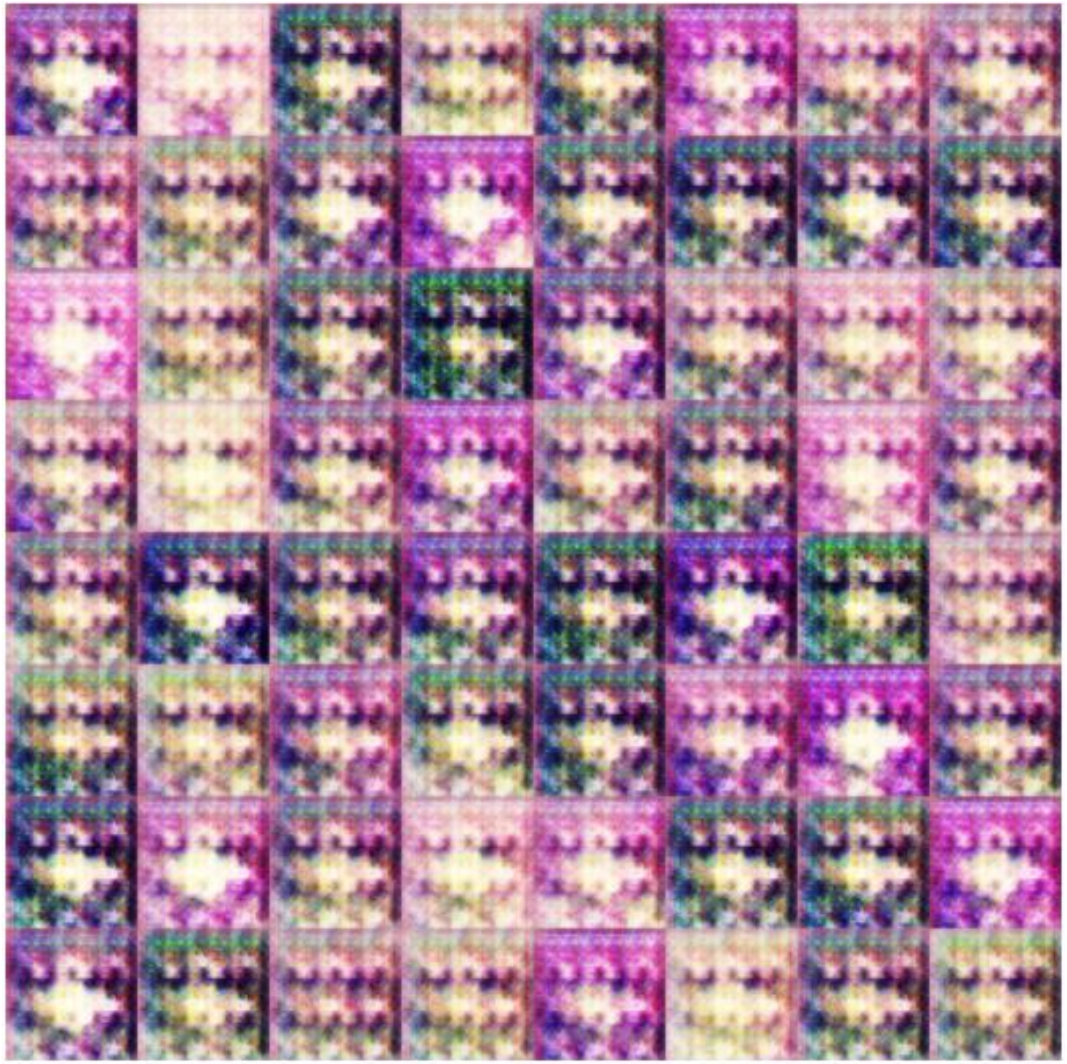
Generator learns to "fool" the discriminator



The goal is to increase the confidence score of the discriminator on the prediction of the real image.

Both modules are actually connected forming a large network

Manga generation results



1st epoch

10th epoch

https://zhuanlan.zhihu.com/p/24767059

Manga generation results

200th epoch

300th epoch

https://zhuanlan.zhihu.com/p/24767059

GAN as Structured Learning

Structure Learning

- Machine learning is to find a function f

$$f: x \rightarrow y$$

- Regression: output a scalar
- Classification: output a “class” (one-hot vector)
- **Structured Learning/Prediction**: output a sequence, a matrix, a graph, a tree..
  - Output is composed of component with dependencies

Examples of structured learning

SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS

Output Sequence: $\boldsymbol{f}: \boldsymbol{x} \rightarrow \boldsymbol{y}$

- *Machine Translation*

$x =$ "She very loves cooking" (English sentence) $y =$ "彼女は料理が大好きです" (Japanese sentence)

- *Speech recognition*

$x =$  $y =$ “The generator learns to generate plausible data”

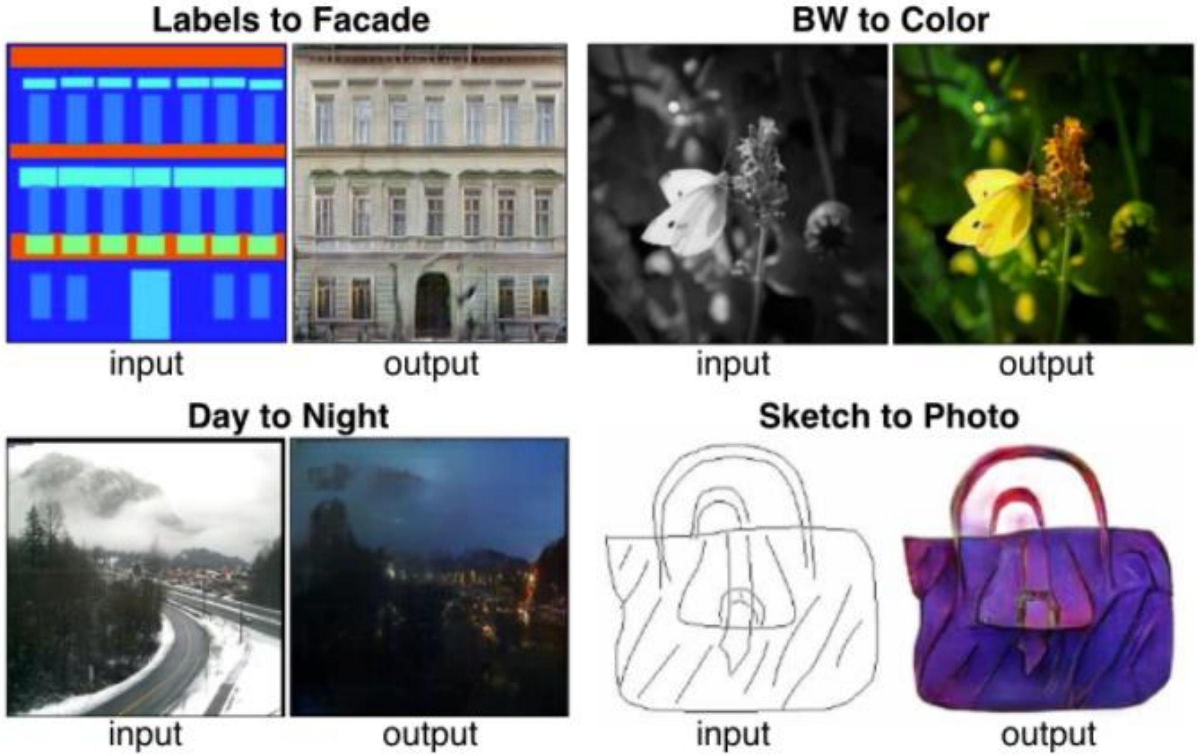
- *Chat-box*

$x =$ "Hello, how are you?" (user says) $y =$ “Thank you, I am fine.” (machine response)

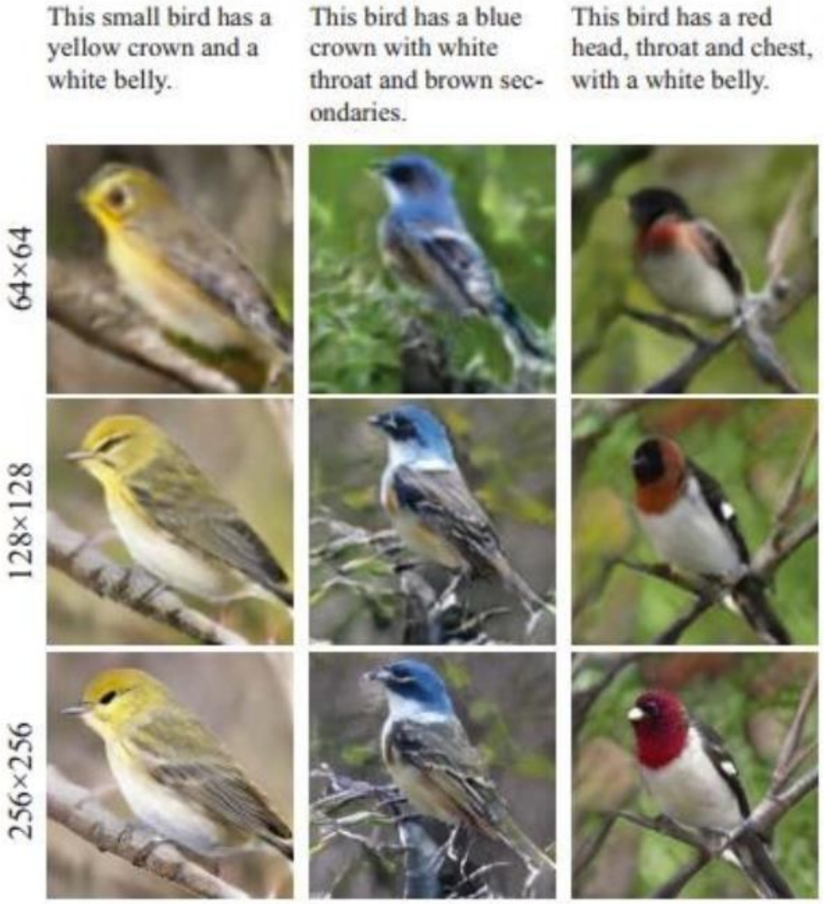
Examples of structured learning

Output Matrix: $f: x \rightarrow y$

- Image to Image translation



- Text to Image translation



Source: https://phillipi.github.io/pix2pix/
https://zjuvag.org/publications/dmgan/

Why structured learning challenging?

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

Why structured learning challenging?

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

  - For example, a pixel of image might depend on its neighboring pixels, the next word in a sentence is dependent on the sequence of the previous words.

GAN as structured learning solution

SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS

Generator
Learn to generate the object at the component level

Bottom Up

+

Discriminator
Evaluating the whole object, and find the best one

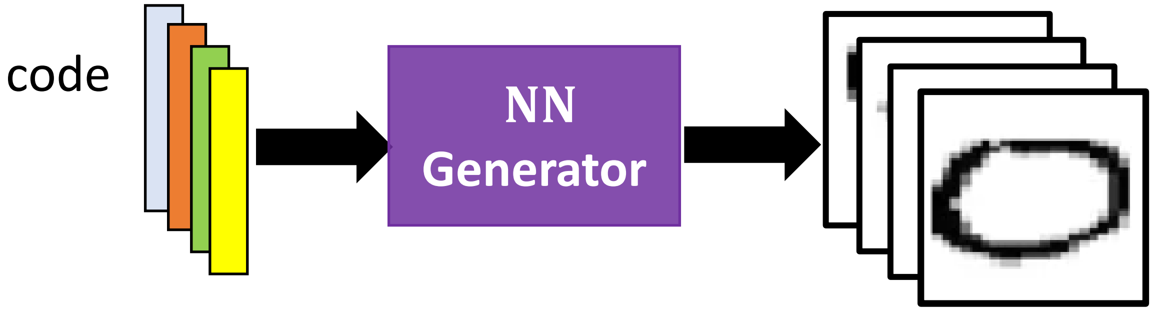
Top down

→

Generative Adversarial Network (GAN)

Can Generator learn itself without depending on Discriminator?

Generator

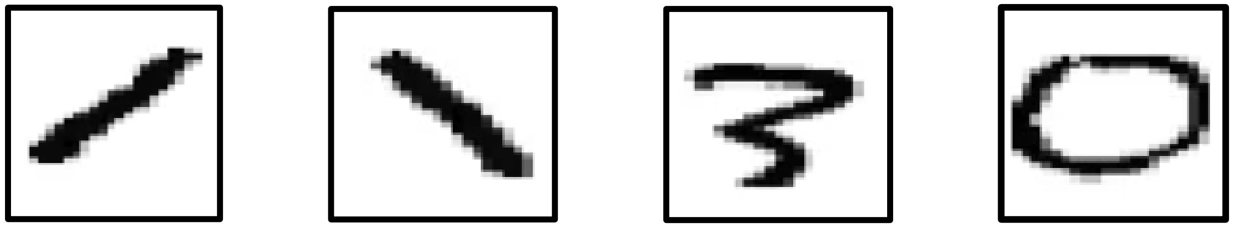


How to generate these codes?

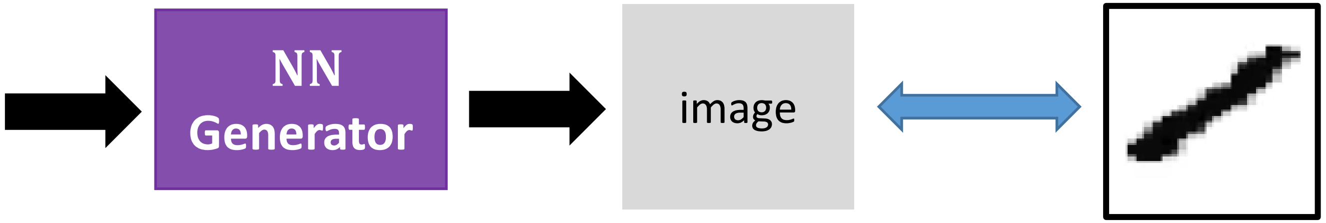
Code:

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

Image:

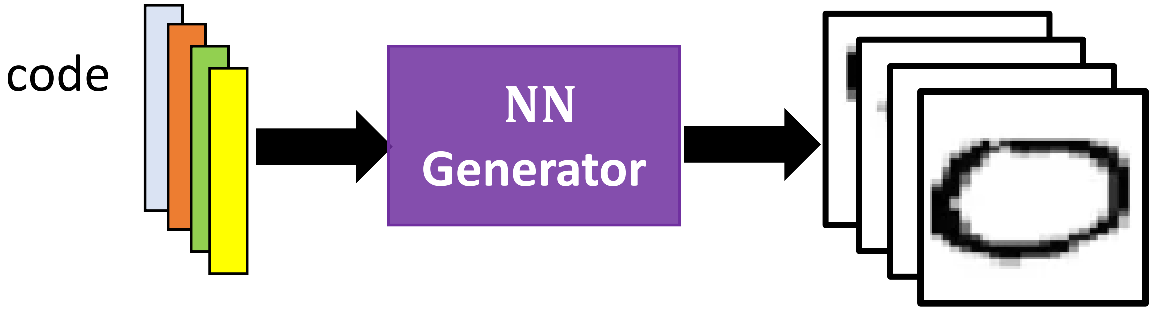


$$\begin{bmatrix} 0.5 \\ -0.3 \\ \vdots \\ 0.1 \end{bmatrix}$$



As close as possible

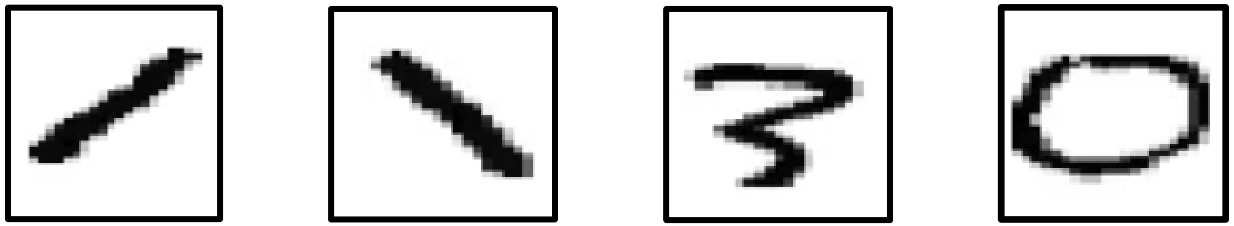
Generator



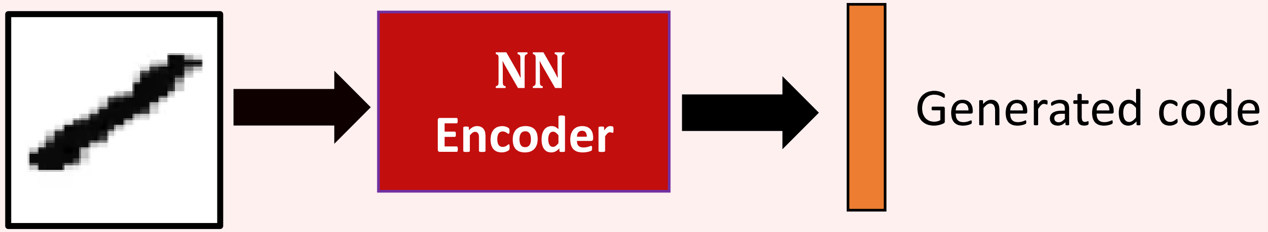
How to generate these codes?

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

Image:

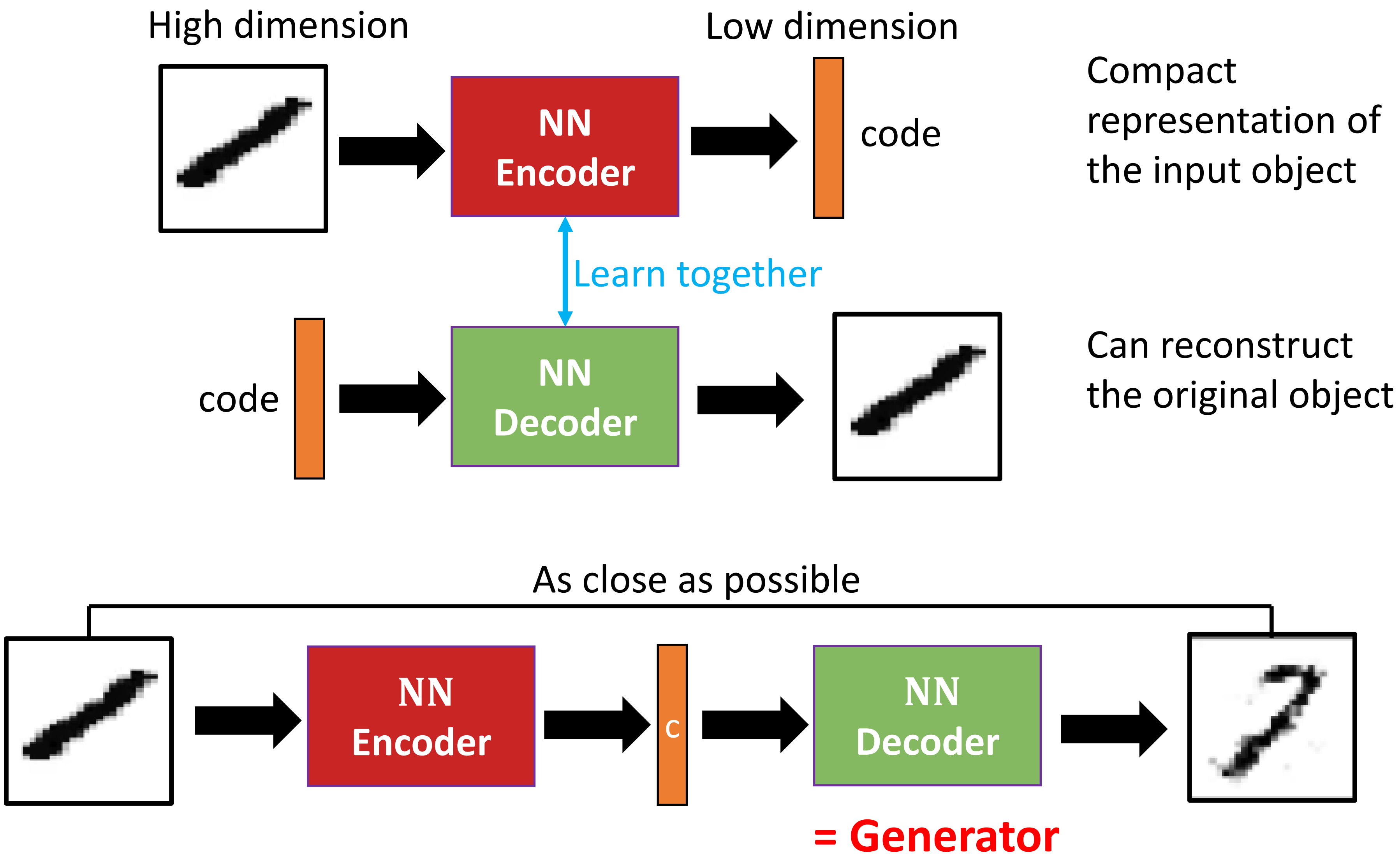


You can use encoder

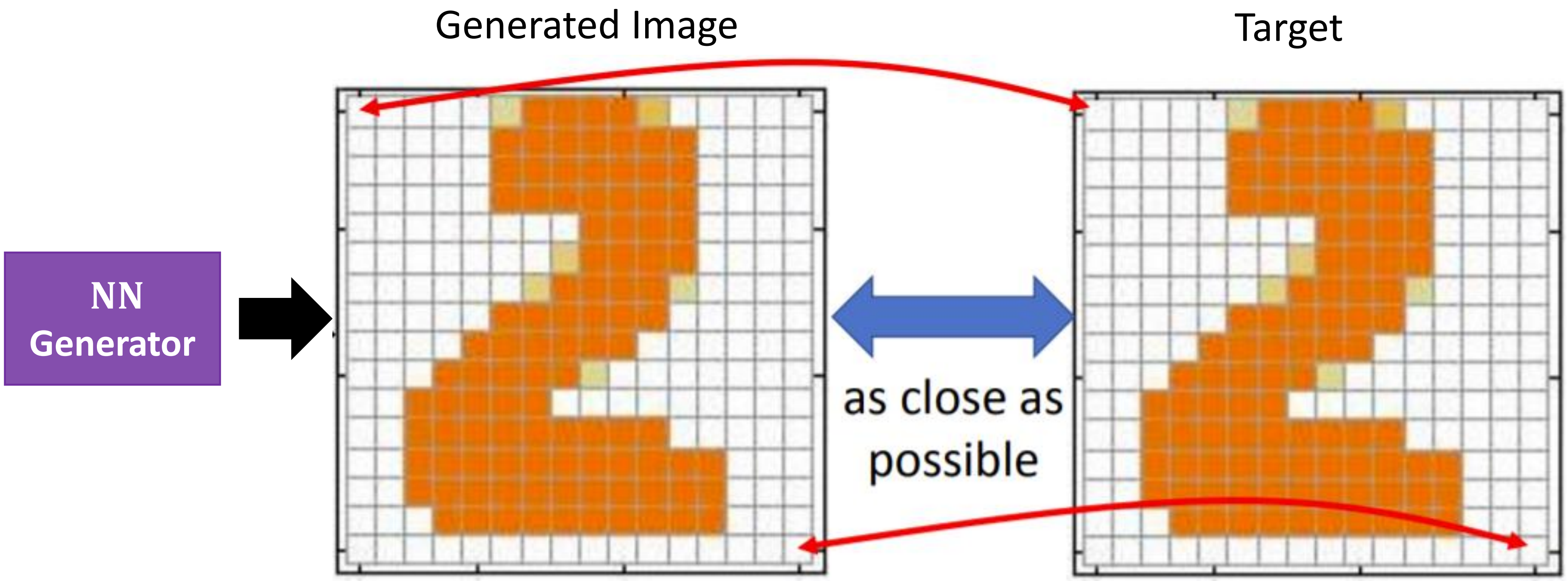


So, how we train an encoder?

Auto-encoder



Problem of Auto-encoder in structured learning



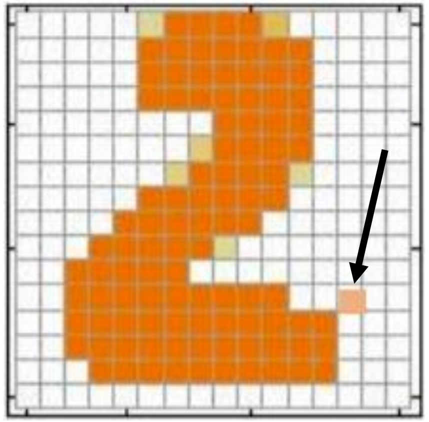
- It will be fine of the generator can truly copy the target image.
- However, this will not always be the case, and if the generator makes errors when generating an image ...

http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2018/Lecture/GAN%20(v2).pdf

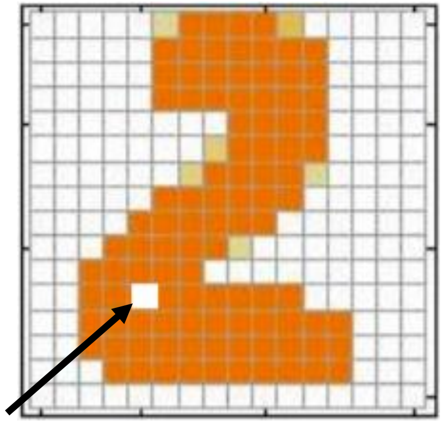
Problem of Auto-encoder in structured learning

NN Generator

Generated Images

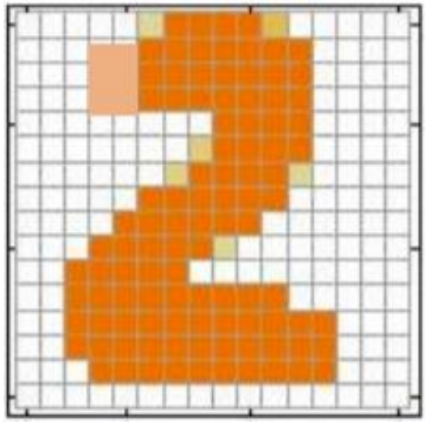


1 pixel error



1 pixel error

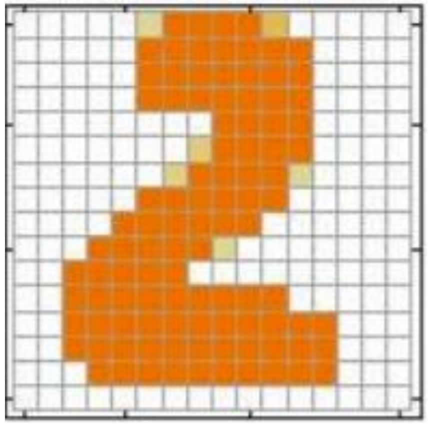
Consider
major error



6 pixels error

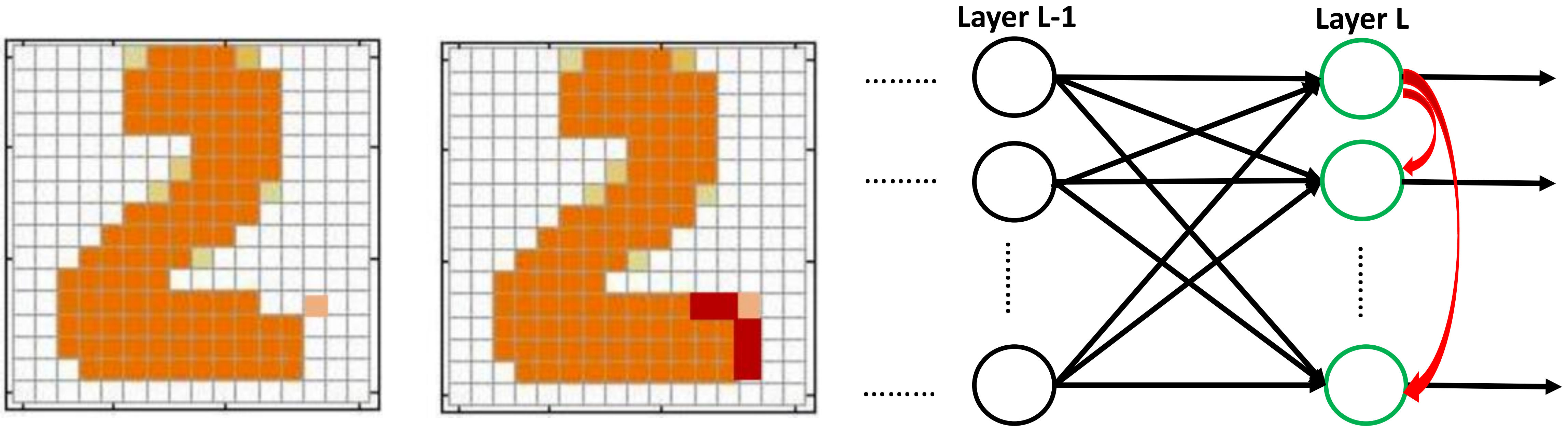
Consider
minor error

Target



Problem of Auto-encoder in structured learning

SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS



- The relation between components which is critical will be missed out.
- The highly correlated components are not able to influence each other.
- Hence, we need a deep structure to catch the relation between components.

Can Discriminator generates image on its own?

Discriminator

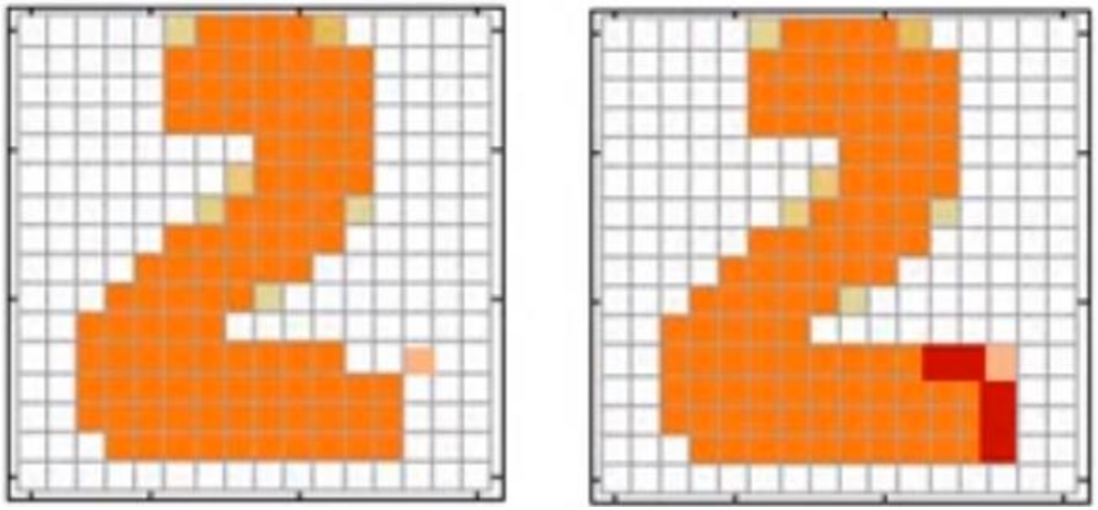
- Discriminator is a function D (network, can be deep)

$$\boldsymbol{D}: \boldsymbol{x} \rightarrow \boldsymbol{R}$$

- Input $\boldsymbol{x}$: an object $\boldsymbol{x}$ (e.g. an image)
- Output $\mathrm{D}(\boldsymbol{x})$: scalar which represents how "good" an object $\boldsymbol{x}$ is.

Discriminator

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

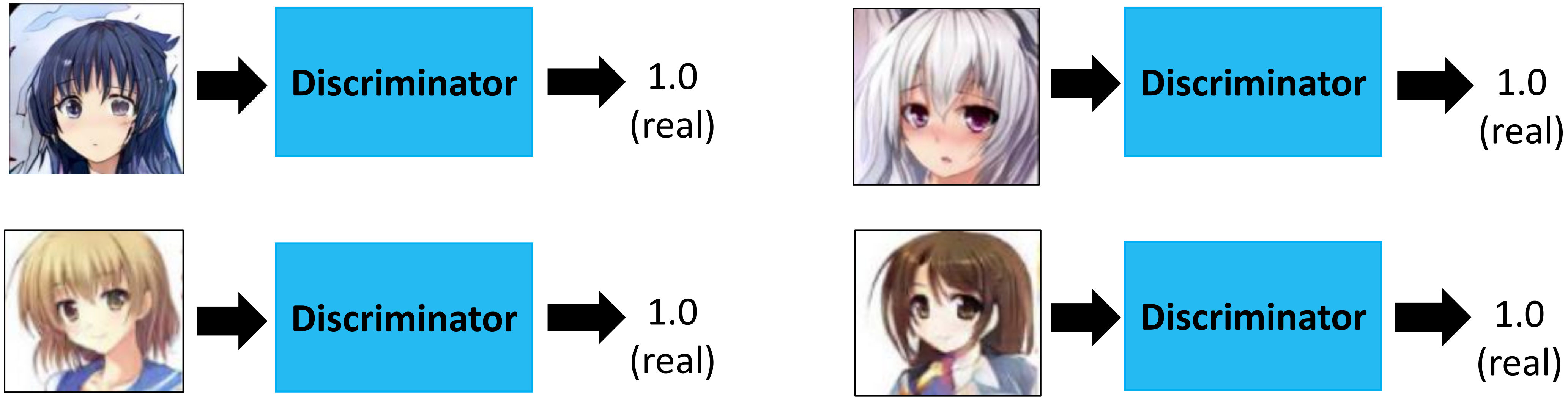


Through learnable CNN filters to check for isolated pixels.

But how to learn a discriminator?

Discriminator-Training

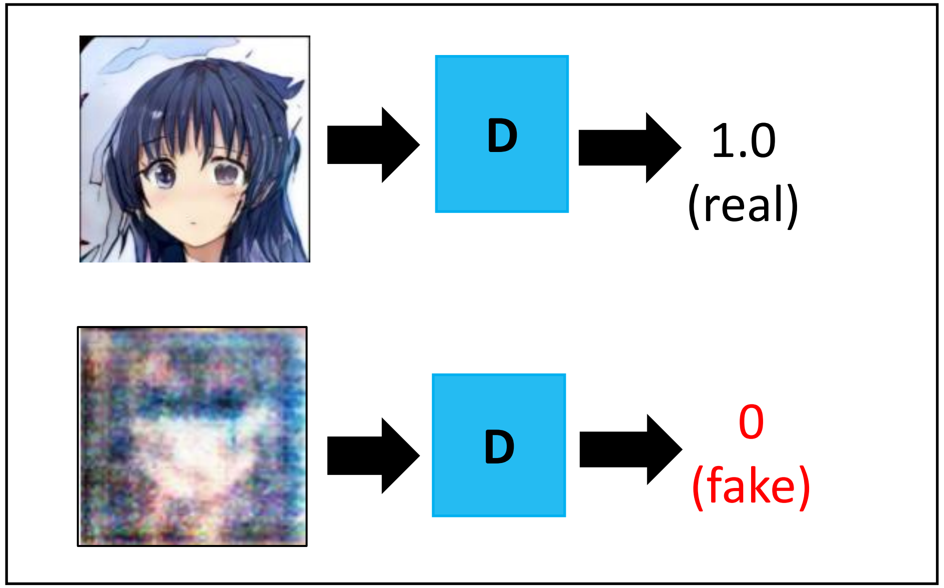
- Discriminator only manage to learn to output “1” (real) as we only have real images

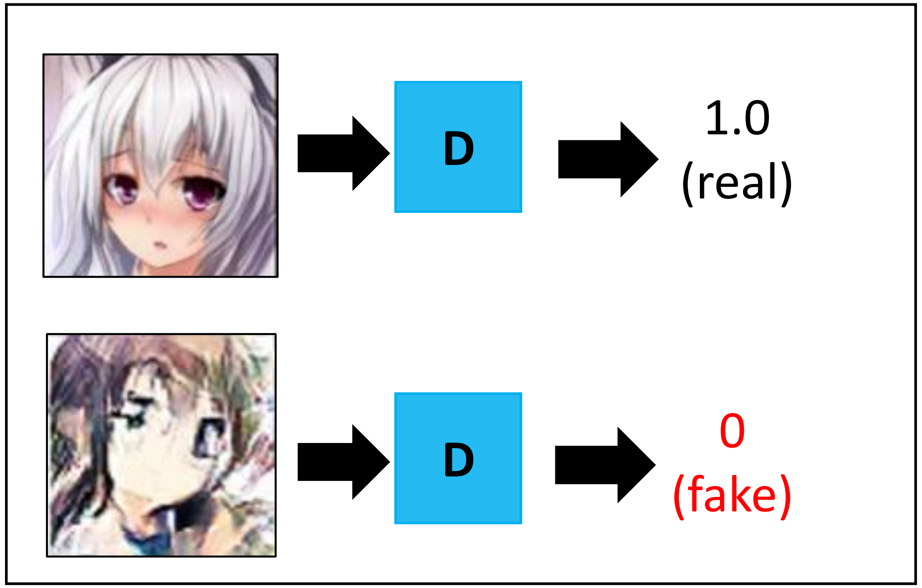


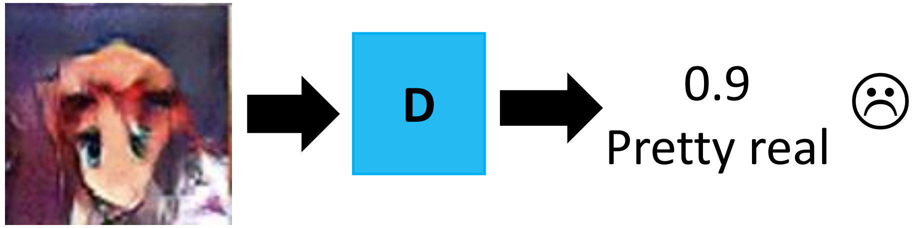
Discriminator training need some negative samples

Discriminator-Training

- Negative examples are critical. But, how to produce/decide negative samples?







How to generate realistic negative samples?

Discriminator-Training

- General Algorithm
  - Given a set of **positive example**, and randomly generate a set of **negative examples**.
  - In each iteration
    - Learn a discriminator D that can discriminate positive and negative examples.
    - Generate negative example by discriminator by solving $\tilde{x} = \boldsymbol{argmax}_{x \in X}\, \boldsymbol{D}(x)$, which aims to maximize the score of classifying real and generated images.

However, this function turns out to be **too complicated** to address if we do not define any limitation. Since this limitation will also restrict the model's capacity, people find it easier to replace solving the $argmax\, D(x)$ function with a separate generator.

Pros and Cons

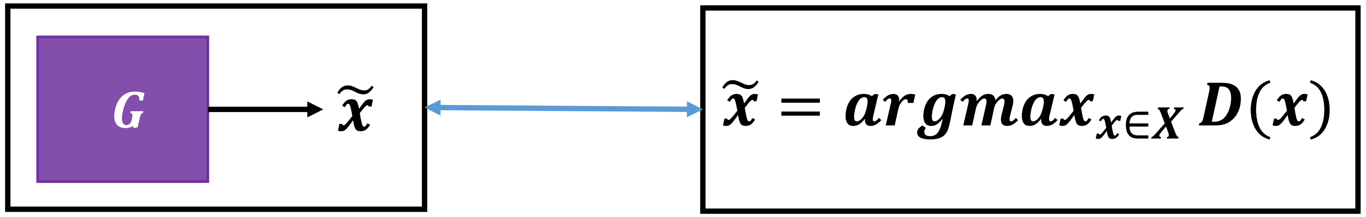
Generative model

- Pros
  - Easy to generate especially with deep model
- Cons
  - Imitate the appearance
  - Hard to learn the correlation between components

Discriminative model

- Pros
  - Considering the big picture
- Cons
  - Generation is not always feasible
  - How to do negative sampling?

Benefit of GAN



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

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

Conditional Generation by GAN

Limitation of GAN

- A limitation of a GAN model is that it may generate a random image from the domain. There is a relationship between points in the latent space to the generated images, but this relationship is complex and hard to map.
- Some datasets have additional information, such as a class label, and it is desirable to make use of this information.
- There are two motivations for making use of the class label information in a GAN model.
    - Improve the GAN.
    - Targeted Image Generation.

Text-to-Image

- Traditional supervised learning

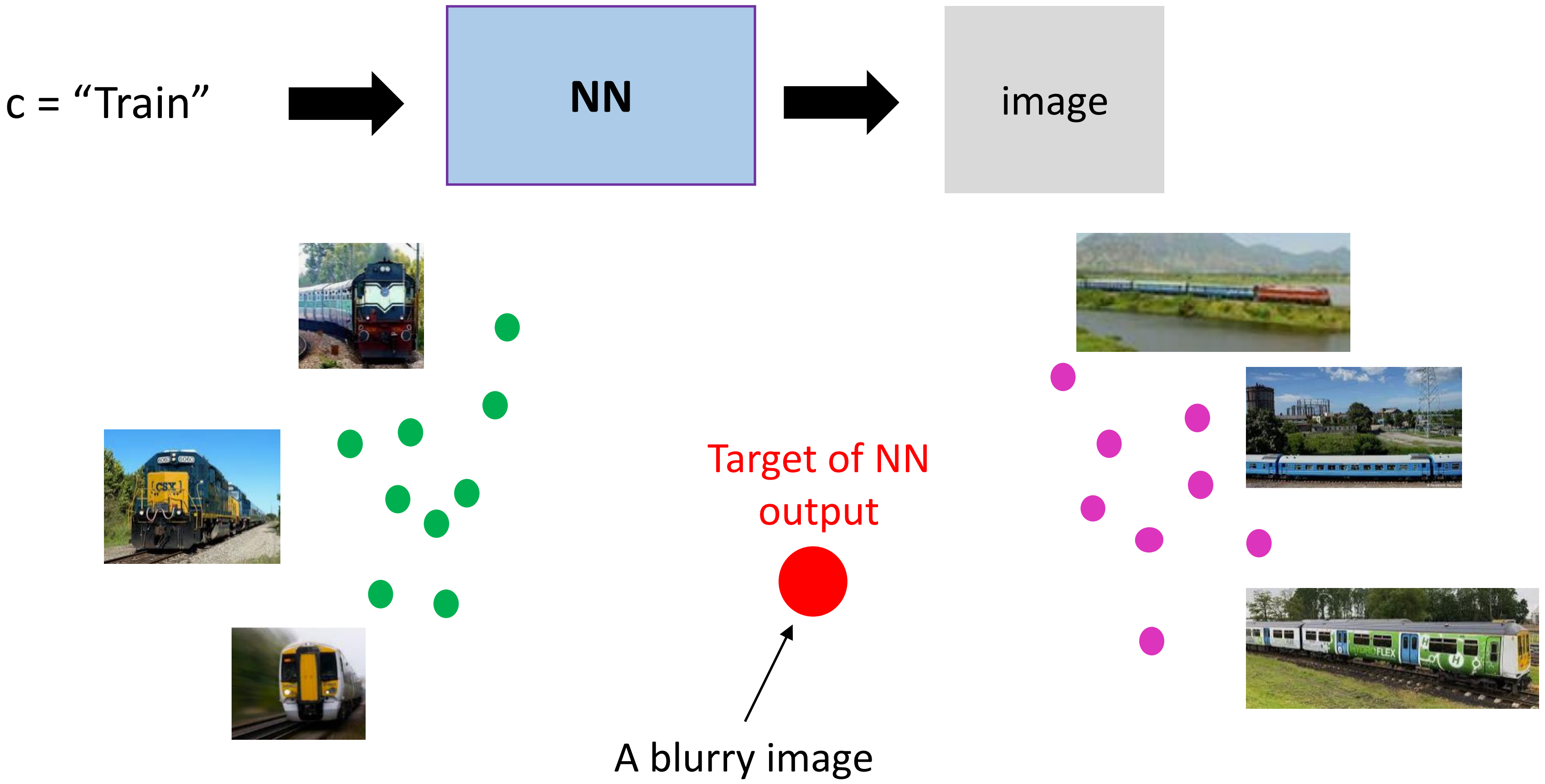
A bird is flying

A dog is running

c = “a bird is flying” ➡ **NN** ➡ image ↔ As close as possible

Text-to-Image

- Traditional supervised learning’s problem
  - A greater possibility to produce blurred images. Blurred images may be due to the average of several images.



Conditional GAN

c = 'Train' → **G** → image → $\tilde{x} = G(c, z)$

Normal distribution z → **G**

If discriminator model trained in such way:

x → **D** → $scalar$ (x is real image or not)

x
- Generated image — image — 0
- Real image —  — 1

Generator will learn to generate realistic images.

But completely ignore the input conditions

Conditional GAN

c = 'Train' → **G** ← Normal distribution z

G → image → $\tilde{x} = G(c, z)$

If discriminator model trained in such way:

Discriminator takes in two inputs which are the input and output of generator.

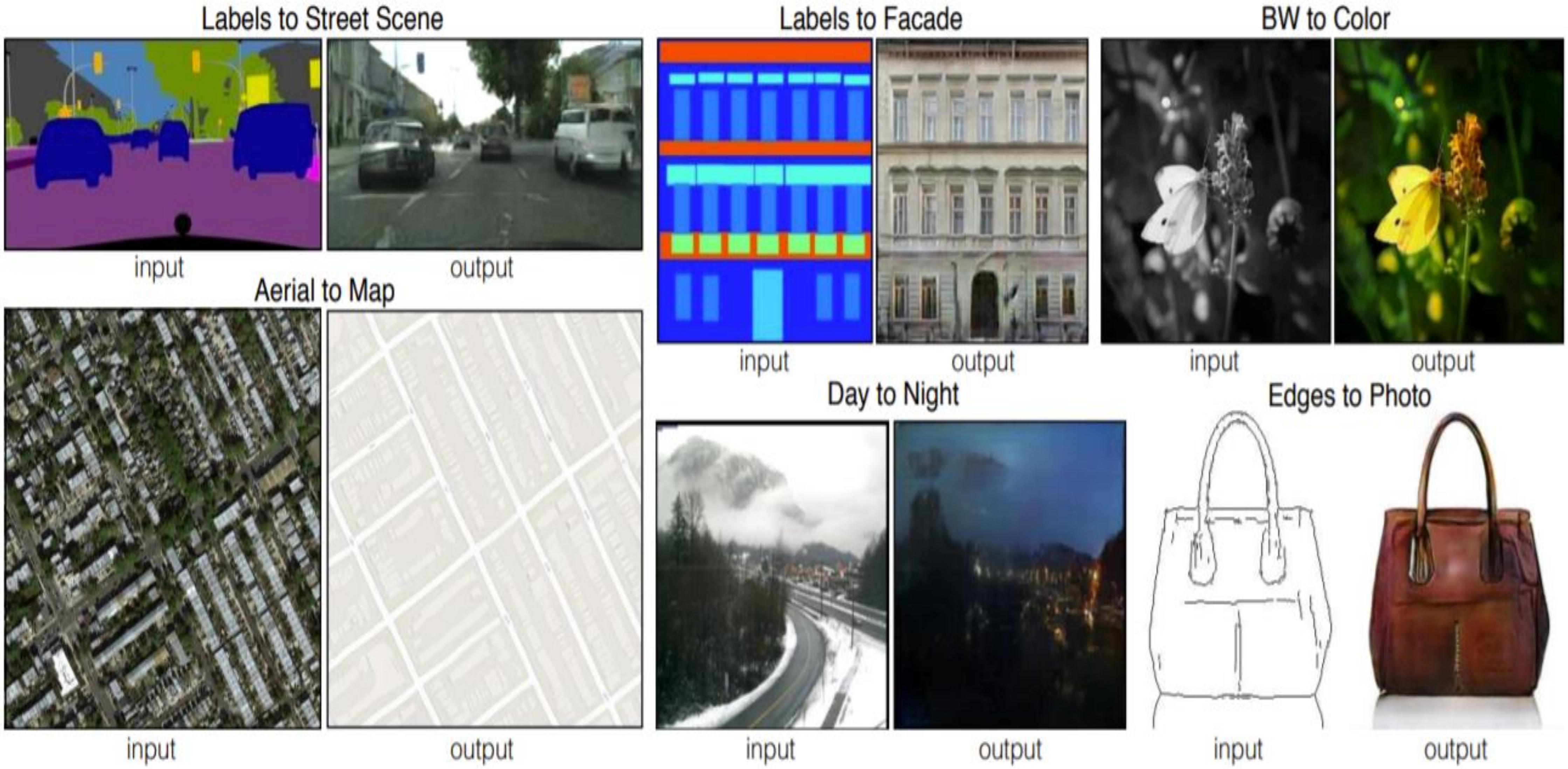
c, x → **D** → $scalar$ (x is real image or not + c and x are matched or not)

$(c\,, x)$

False pairs: (cat, ) 0 (train, image) 0

True pairs: (train, ) 1

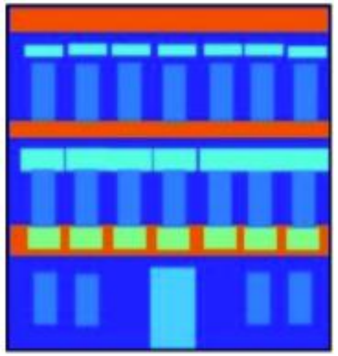
Image-to-Image



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

Image-to-Image

- Traditional supervised approach can be used but generated image will be blurry.

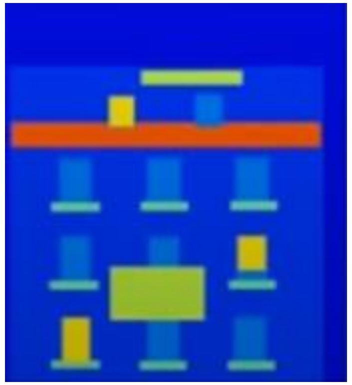


NN

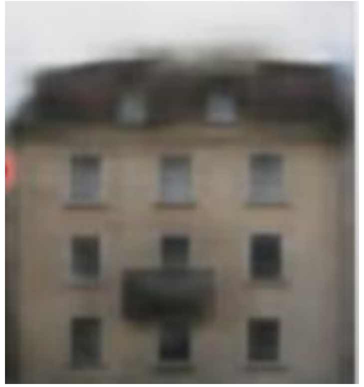
image

As close as possible

Testing:



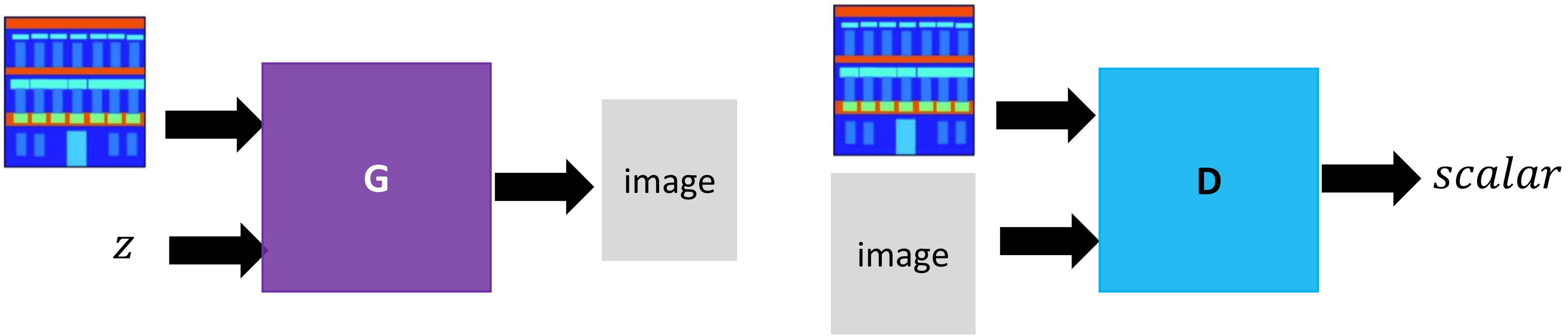
Input



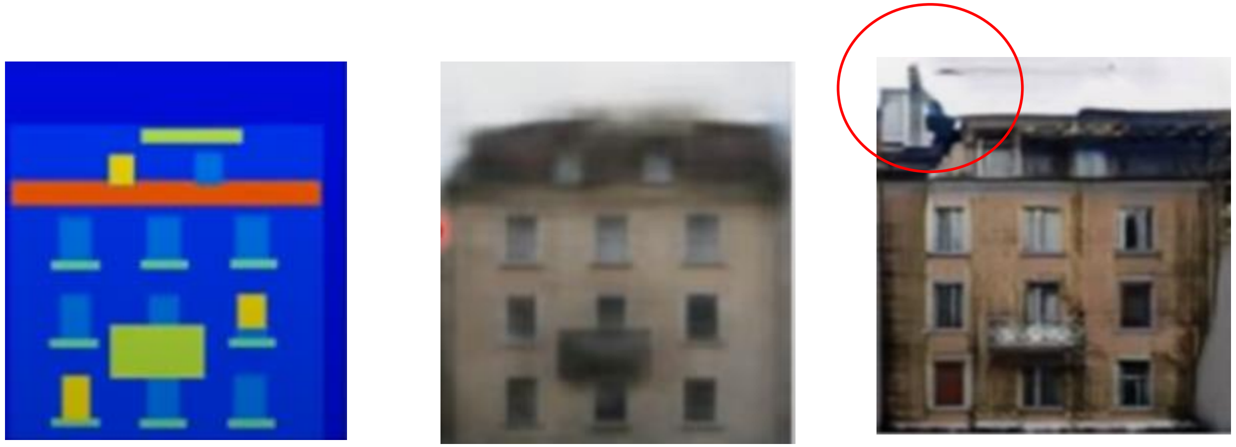
Close

It is blurred because it may be due to the average of several images.

Image-to-Image using GAN



Testing:

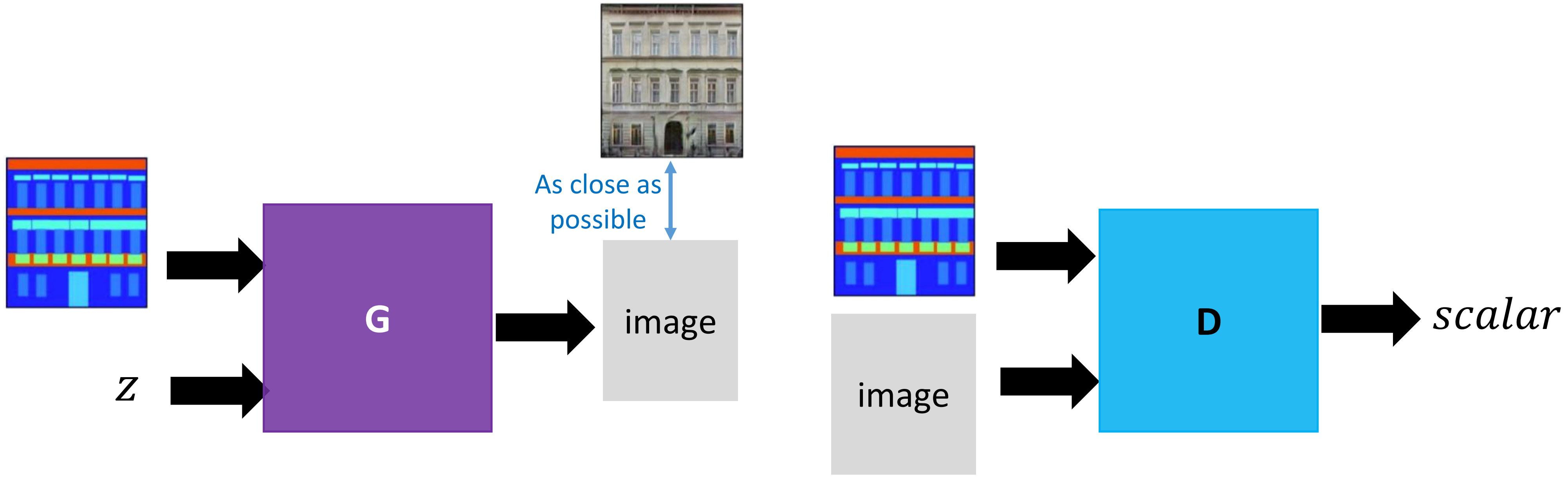


Input

Close

GAN

Image-to-Image using GAN

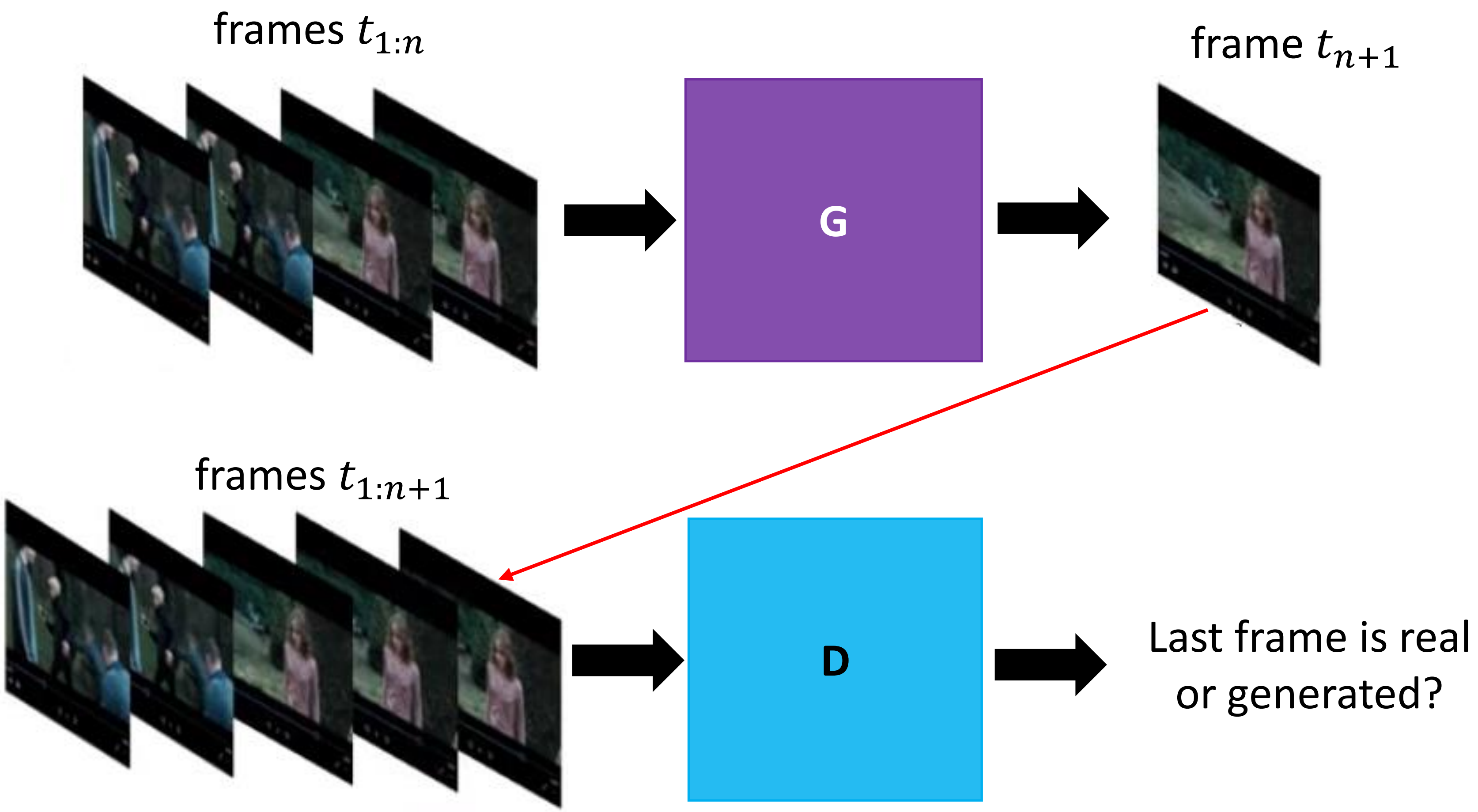


Testing:

Input | Traditional | GAN | GAN + close

Other Applications using conditional GAN

- Video Generation



Unsupervised
Conditional Generation

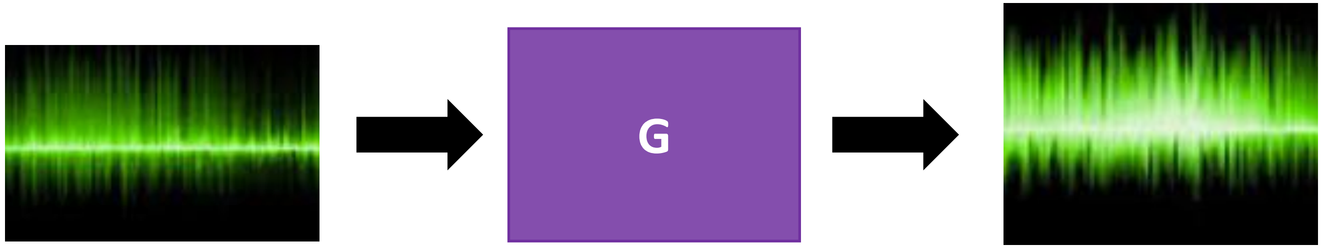
Unsupervised conditional generation

- Transform an object from one domain to another **without paired data** (e.g.: style transfer)

Domain X

Domain Y

Domain X

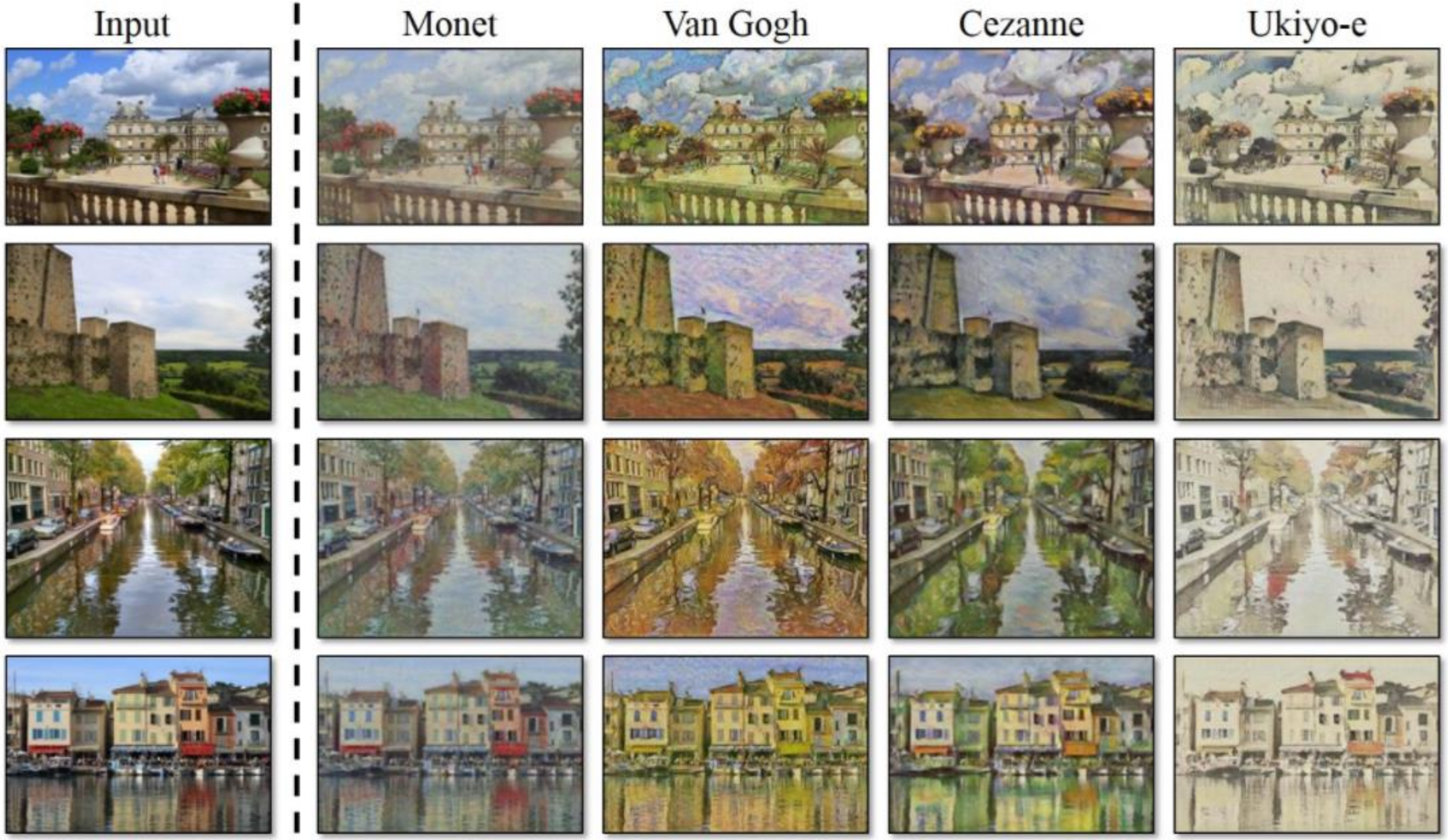


Domain Y

male speech

Female speech

Style transfer experimental results



https://arxiv.org/pdf/1703.10593.pdf

Approaches

- Approach 1: Direct transformation

Domain X

→ $G_{X \to Y}$ →

Domain Y

For texture or colour change

- Approach 2: Projection to Common Space

Domain X

→ **Encoder of domain *X*** → Face attribute → **Decoder of domain *Y*** →

Domain Y

For larger change, only keep the semantics

https://syncedreview.com/2019/08/08/south-korean-game-developers-ai-turns-your-selfie-into-an-anime-face/

Approaches

- Approach 1: Direct transformation

Domain X

$G_{X \to Y}$

Domain Y

For texture or colour change

- Approach 2: Projection to Common Space

Domain X

Encoder of domain X

Face attribute

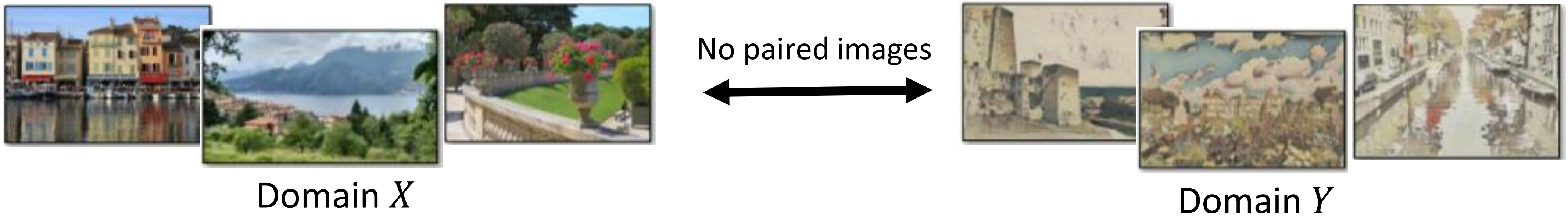
Decoder of domain Y

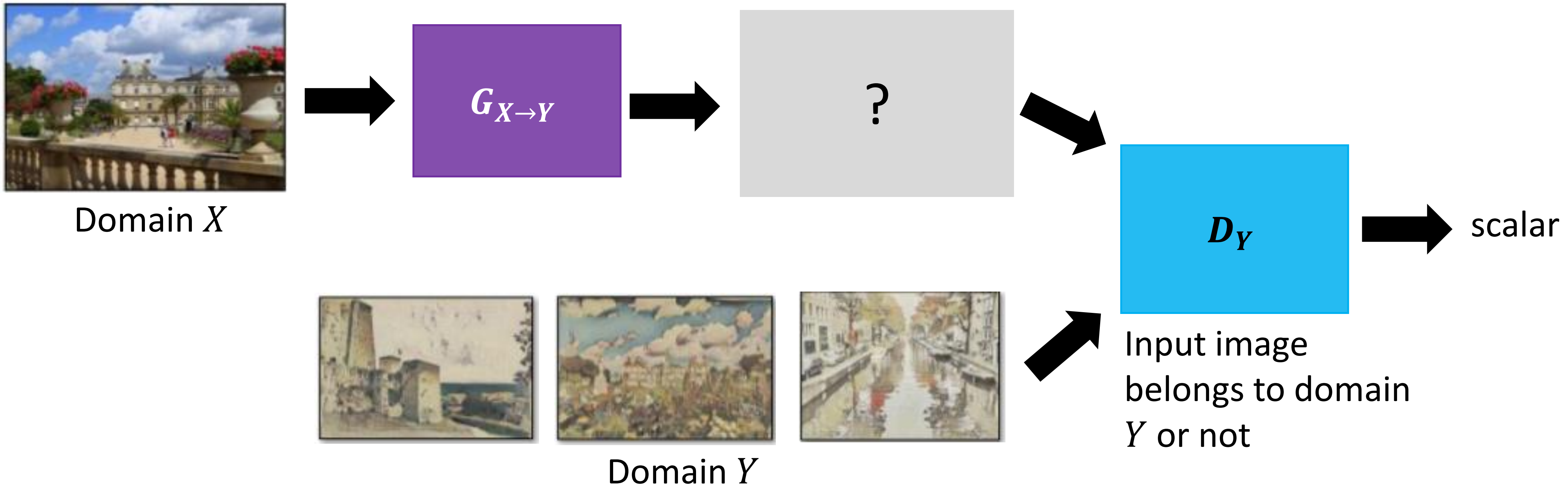
Domain Y

For larger change, only keep the semantics

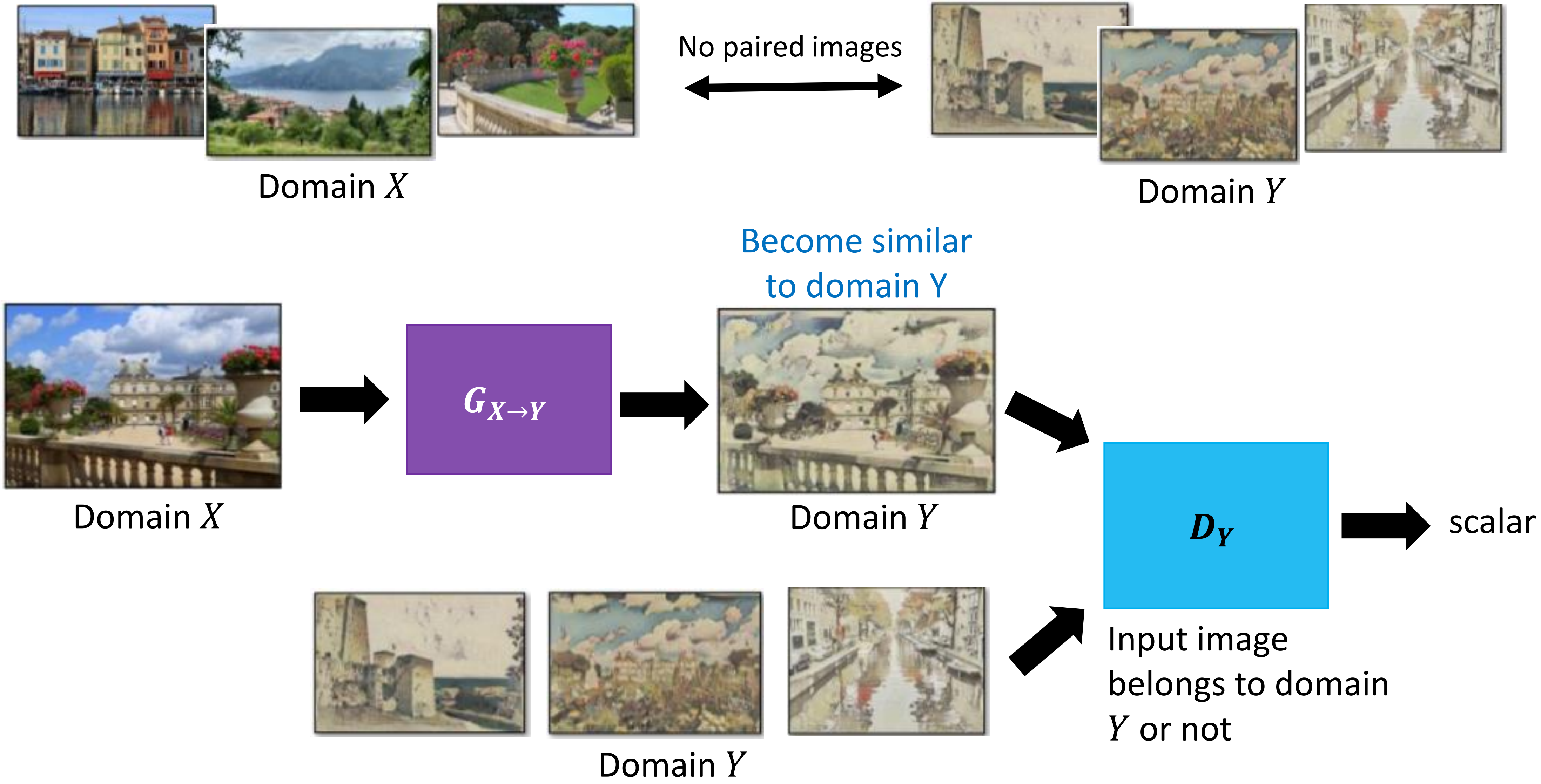
https://syncedreview.com/2019/08/08/south-korean-game-developers-ai-turns-your-selfie-into-an-anime-face/

Direct transformation

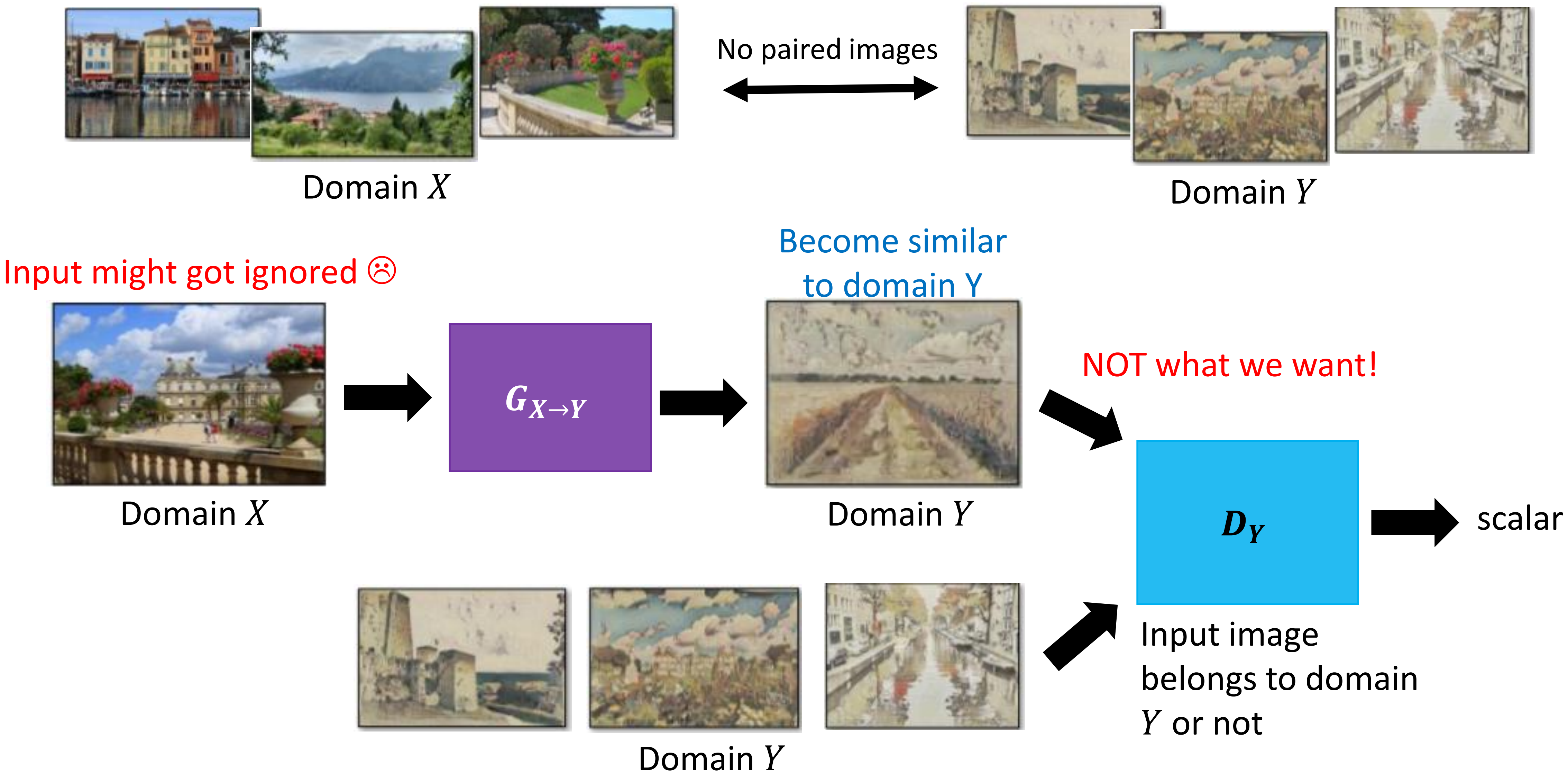




Direct transformation



Direct transformation



Direct transformation

Domain X

No paired images

Domain Y

Input might got ignored ☹

Domain X

$G_{X \to Y}$

Become similar to domain Y

Domain Y

NOT what we want!

D_Y

scalar

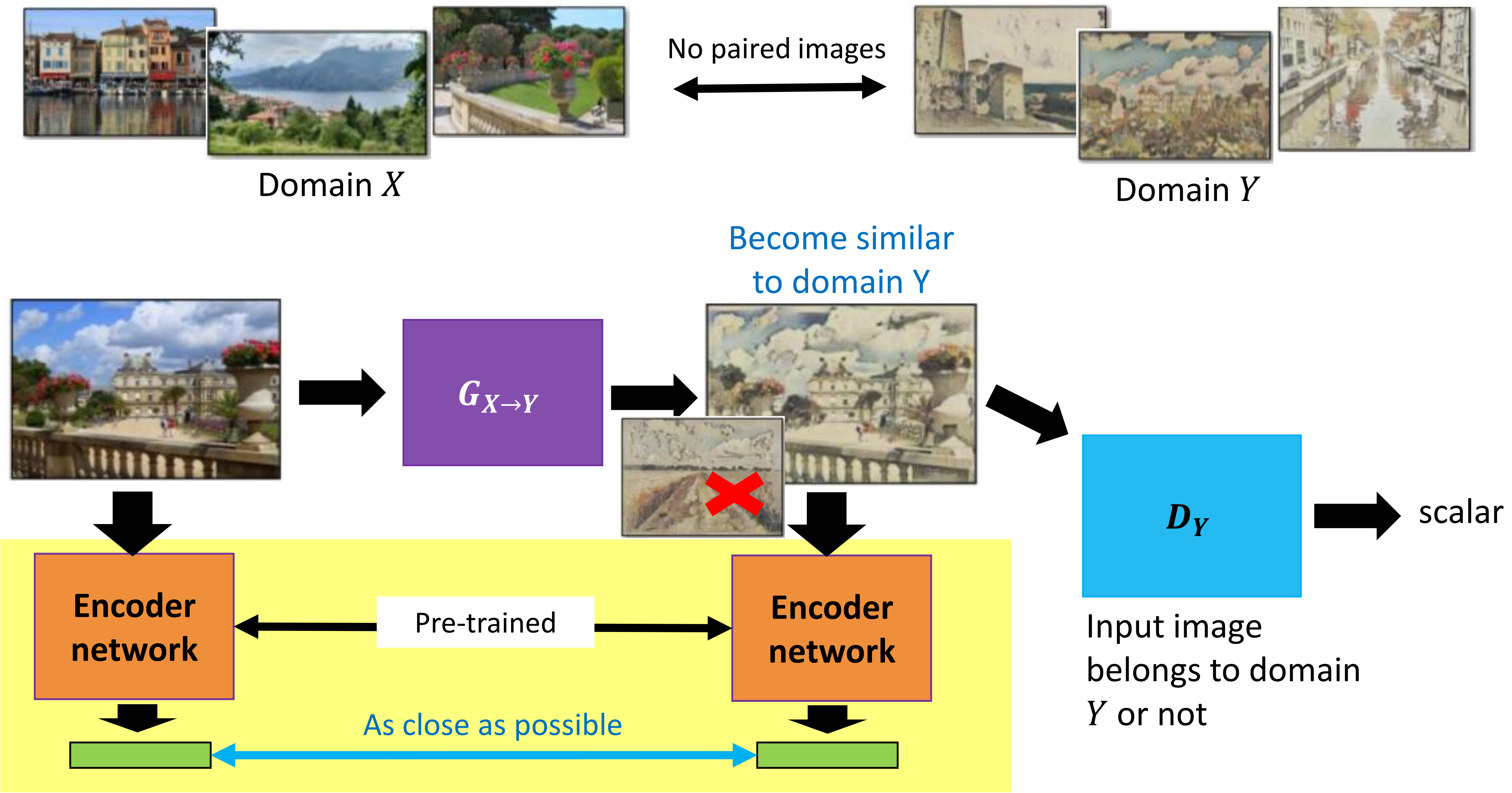
Input image belongs to domain Y or not

- The issue can be avoided by network design.
- Simpler generator makes input and output more closely related.

Domain Y

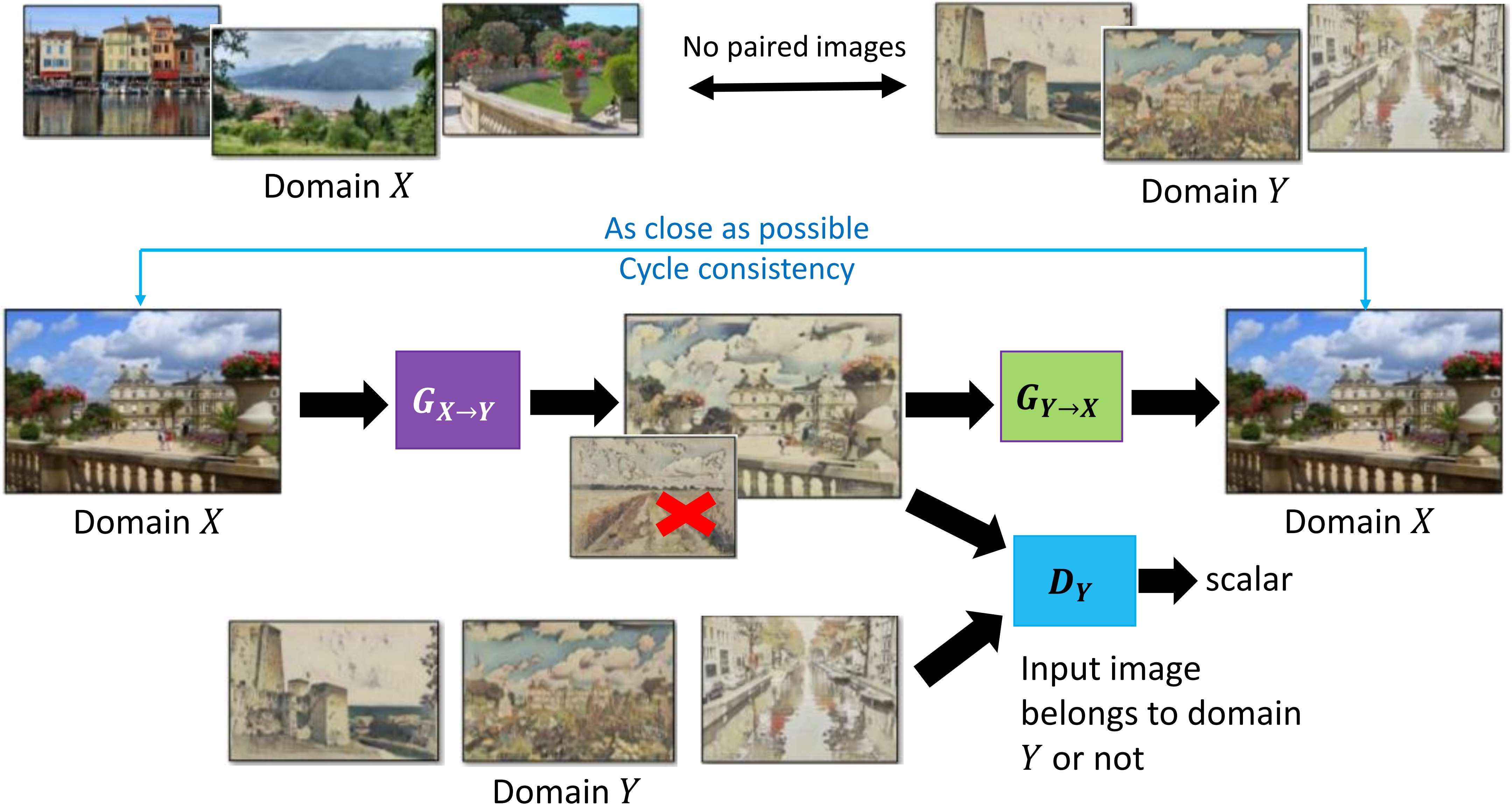
https://arxiv.org/abs/1709.00074

Direct transformation



[Unsupervised Cross-Domain Image Generation] https://research.fb.com/publications/unsupervised-cross-domain-image-generation/

Direct transformation - Cycle GAN



https://arxiv.org/abs/1703.10593

Approaches

- Approach 1: Direct transformation

Domain X

$G_{X \to Y}$

Domain Y

For texture or colour change

- Approach 2: Projection to Common Space

Domain X

Encoder of domain *X*

Face attribute

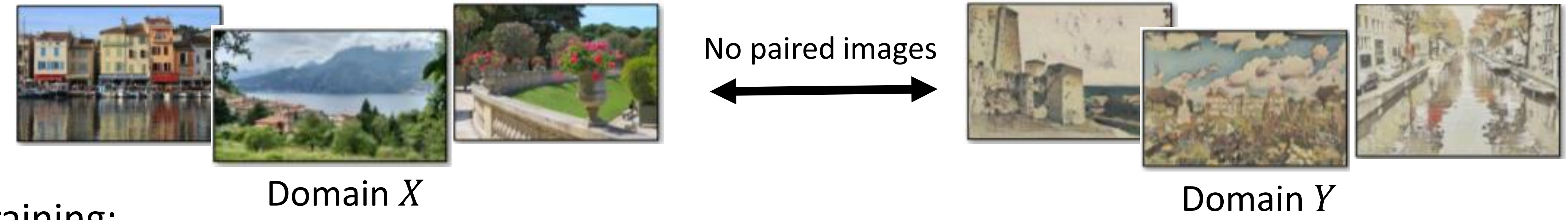
Decoder of domain *Y*

Domain Y

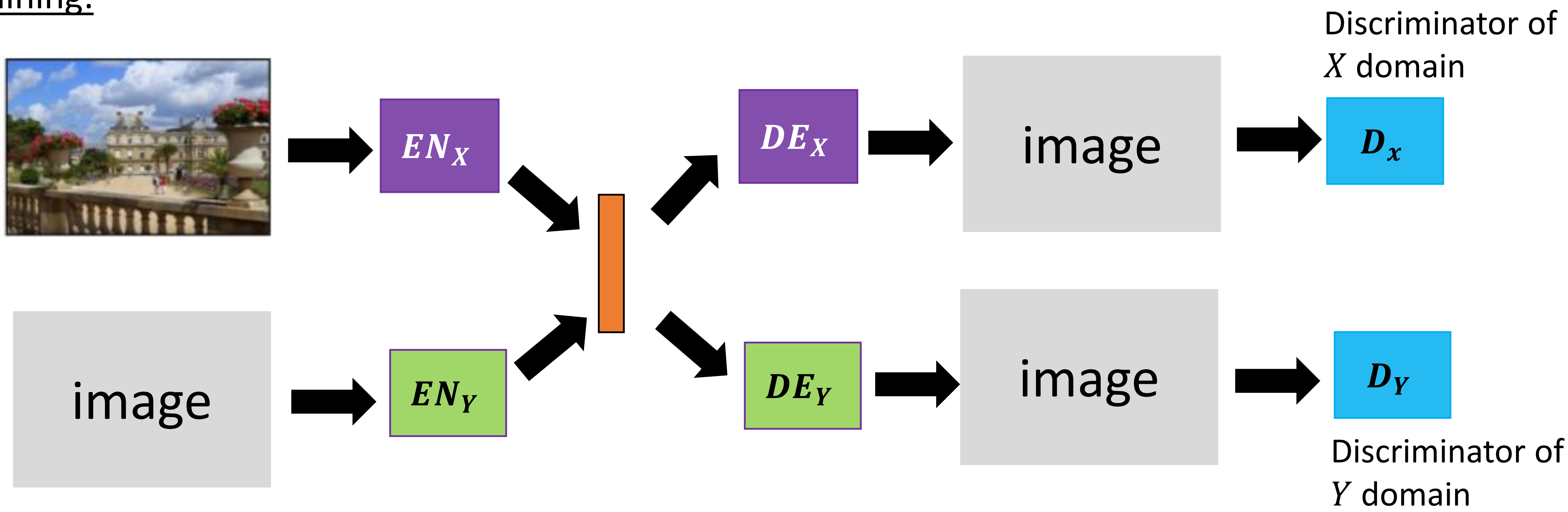
For larger change, only keep the semantics

https://syncedreview.com/2019/08/08/south-korean-game-developers-ai-turns-your-selfie-into-an-anime-face/

Projection to common Space – ComboGAN

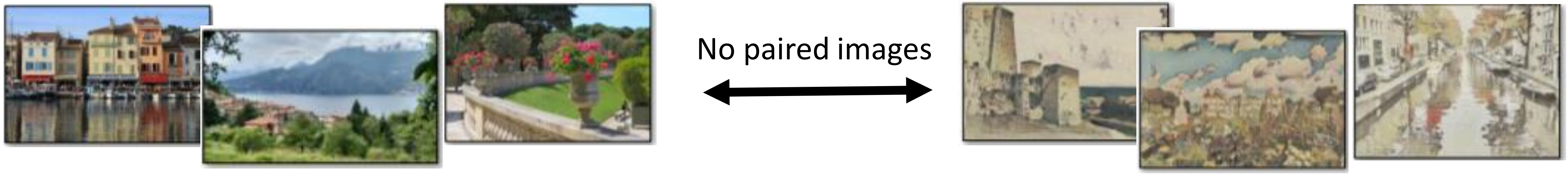


Training:



https://arxiv.org/pdf/1712.06909.pdf

Projection to common Space – ComboGAN

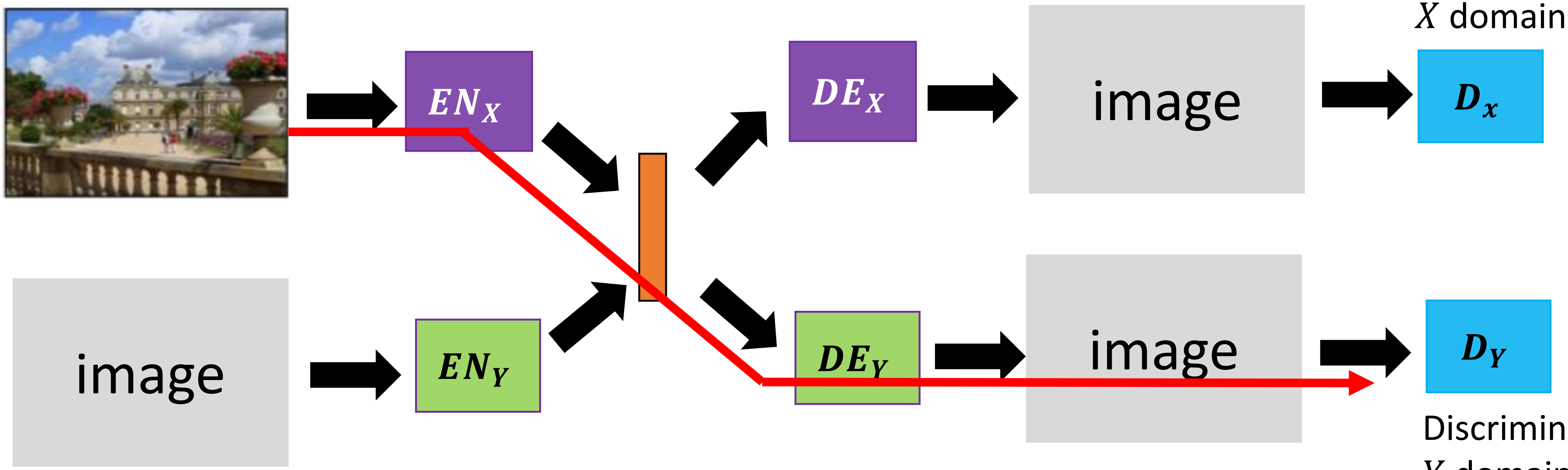


Domain X

Domain Y

<u>Training:</u>

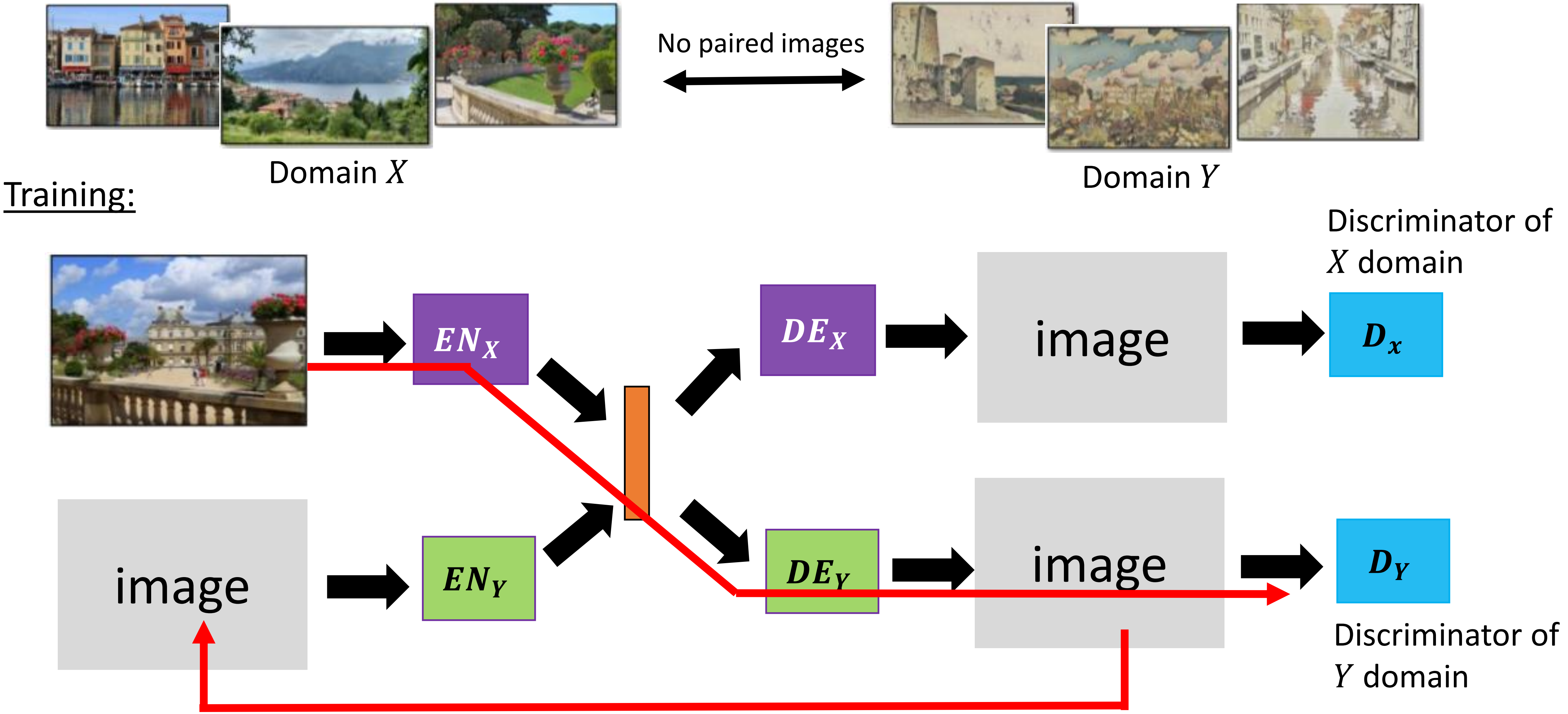
Discriminator of X domain



Discriminator of Y domain

https://arxiv.org/pdf/1712.06909.pdf

Projection to common Space – ComboGAN



https://arxiv.org/pdf/1712.06909.pdf

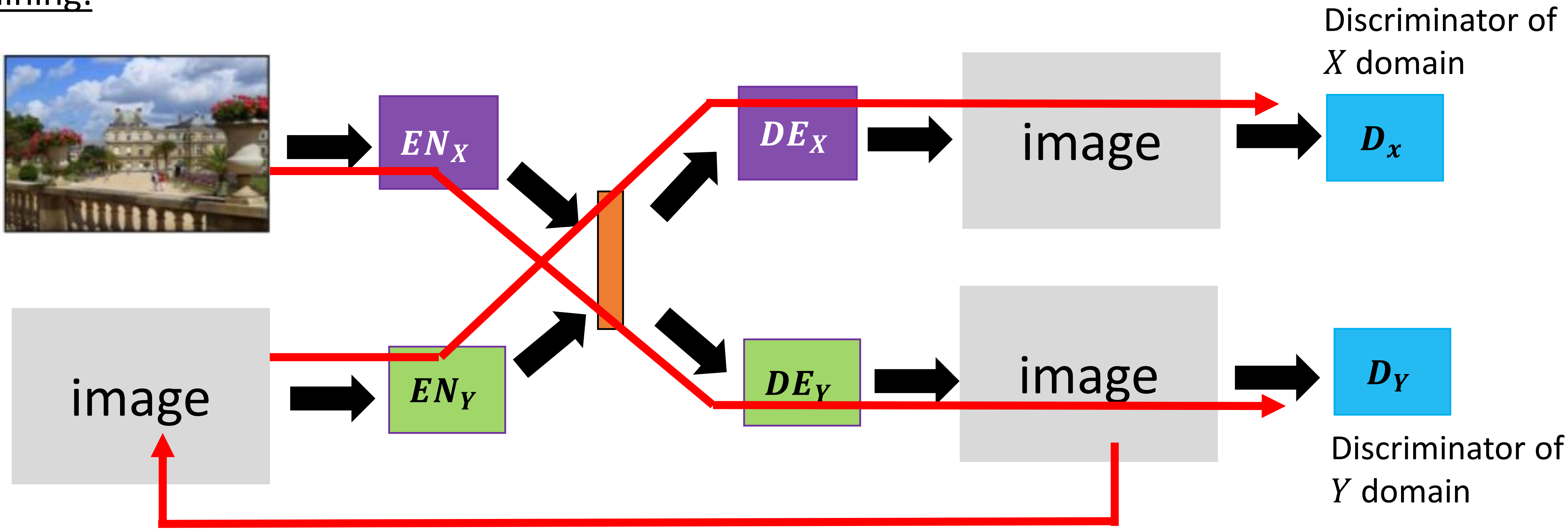
Projection to common Space – ComboGAN

Domain X

No paired images

Domain Y

Training:

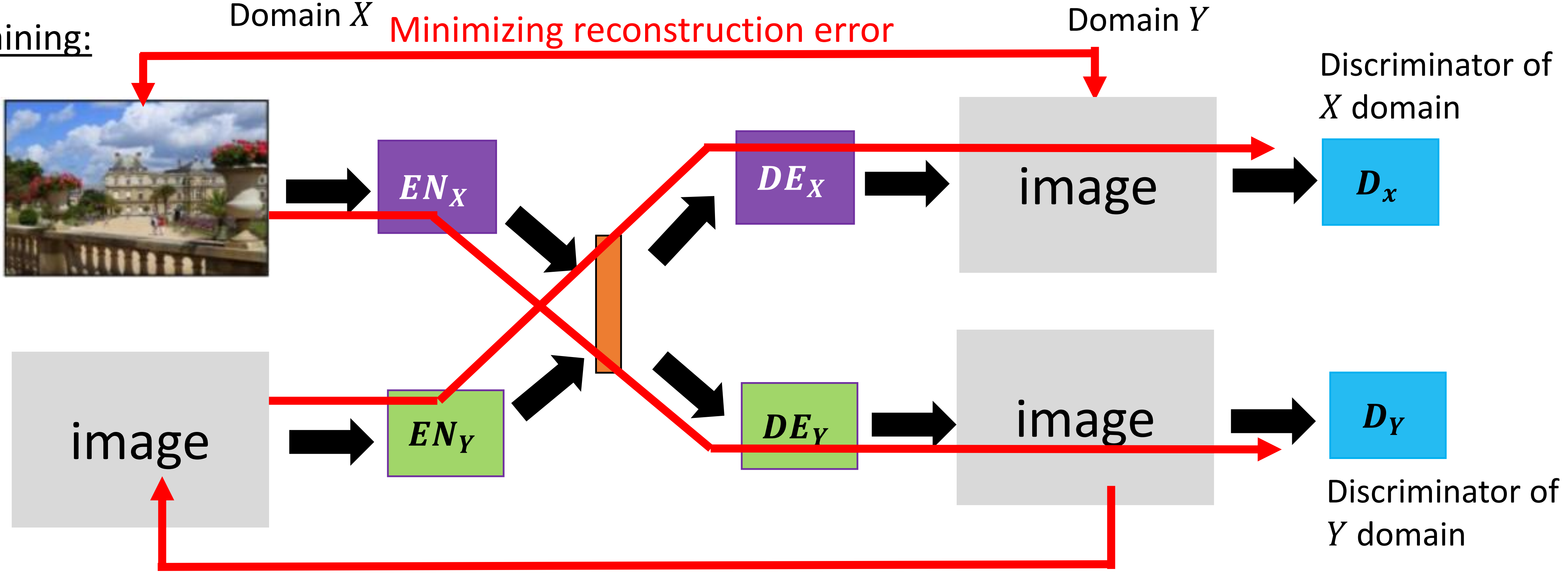


https://arxiv.org/pdf/1712.06909.pdf

Projection to common Space – ComboGAN

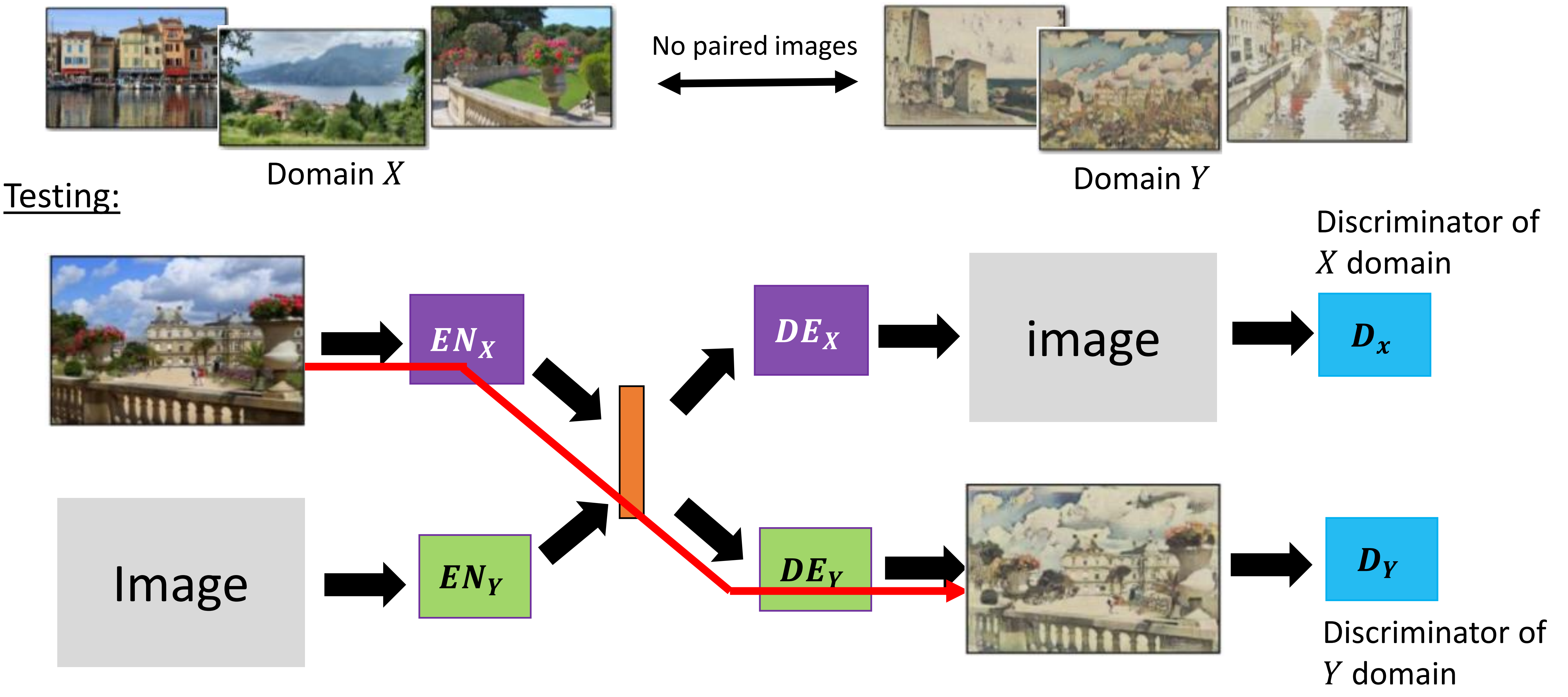
No paired images

Training:



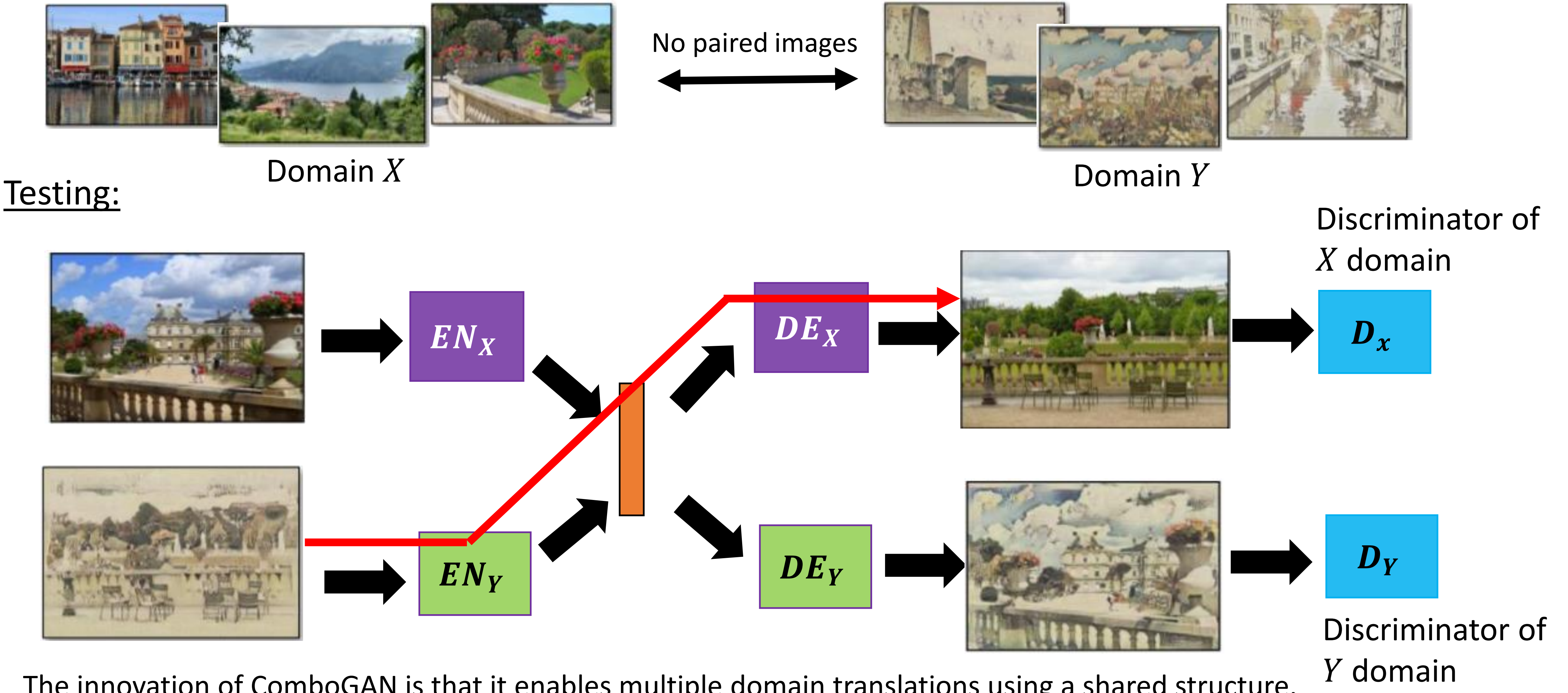
Cycle consistency

Projection to common Space – ComboGAN



Projection to common Space – ComboGAN

SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS



The innovation of ComboGAN is that it enables multiple domain translations using a shared structure.

https://arxiv.org/pdf/1712.06909.pdf

References

- The-GAN-Zoo
  - https://github.com/hindupuravinash/the-gan-zoo
- Keras-GAN
  - https://github.com/eriklindernoren/Keras-GAN