Generative Adversarial Network

Weizi Li

Department of Computer Science
University of Memphis

Outline

- Generative Adversarial Network
- Conditional GAN
- Example Code

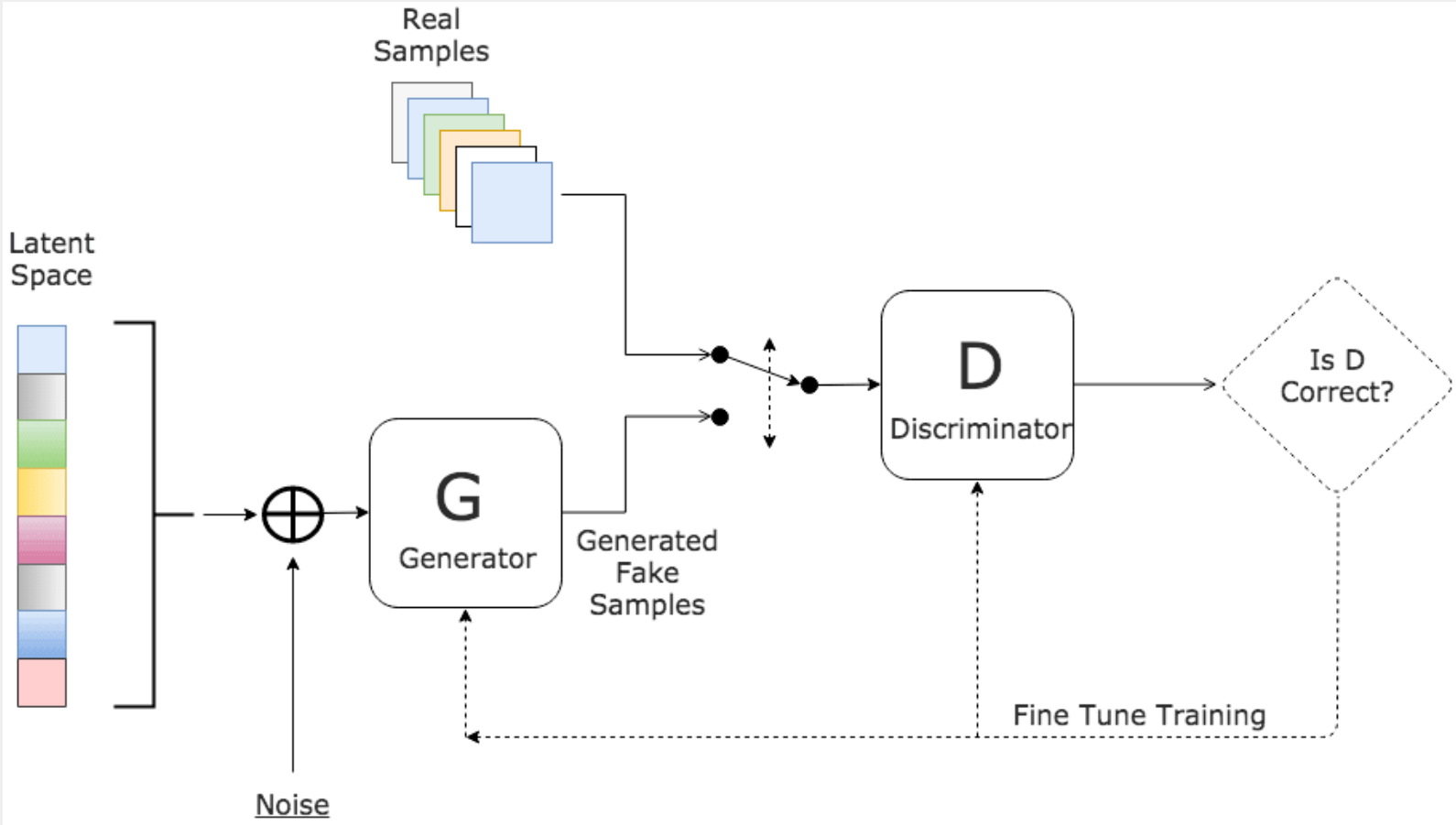
Generative Adversarial Network

- Motivation: if deep neural network is so expressive, how about put two in competition?

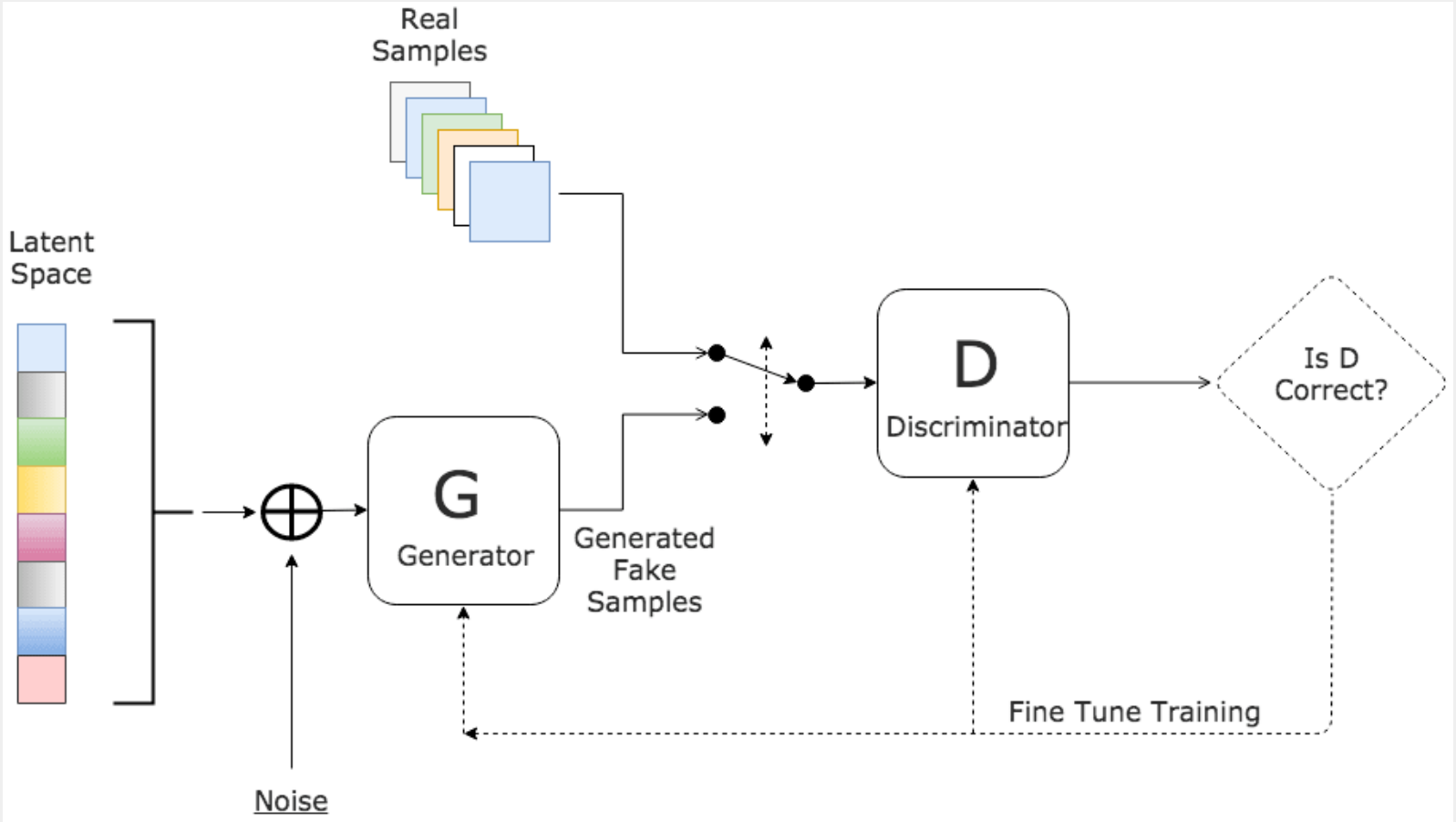
- Goodfellow et al., Generative Adversarial Nets, 2014
- Generate synthetic data: images, sentences, ...
- “the most interesting idea in the last 10 years in Machine Learning” – Yann LeCun, Turing Award winner (2018)

- Karras et al., A Style-Based Generator Architecture for Generative Adversarial Networks, 2019

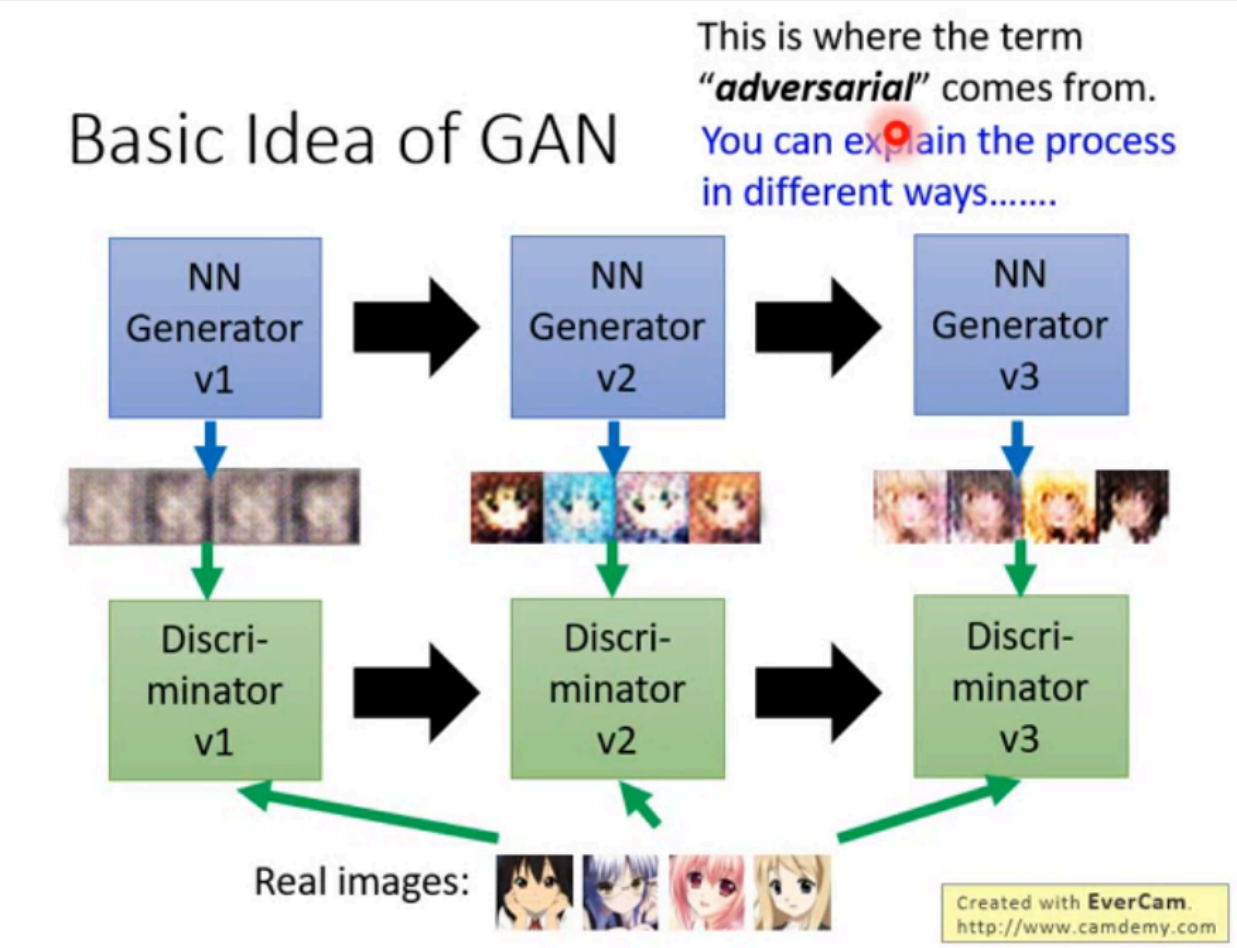
- Generator (G): latent space $\rightarrow$ synthetic data
- Each element in the latent space can represent a characteristic of the synthetic data



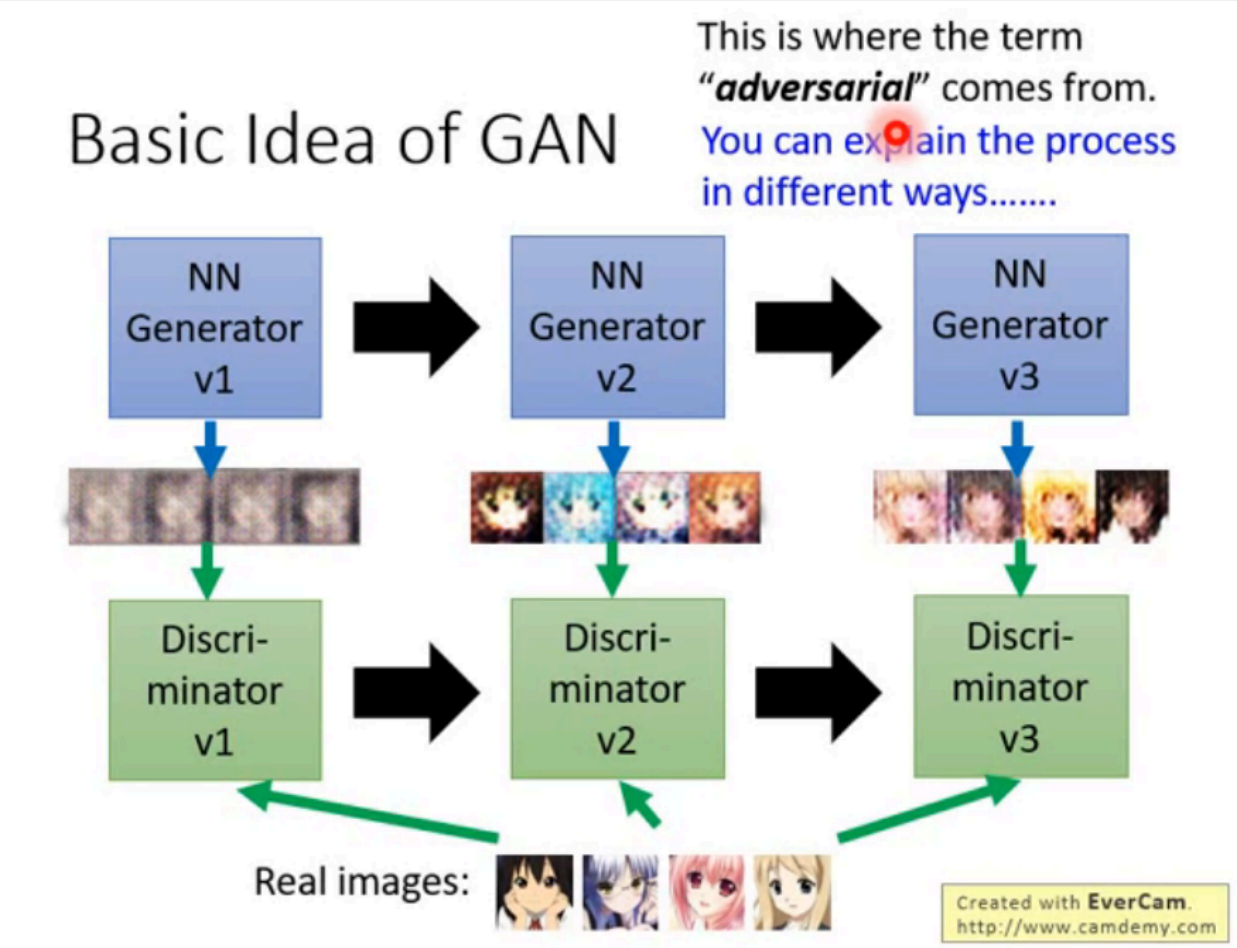
- Discriminator (D): synthetic data $\rightarrow$ score (scalar)
- The score is used to judge whether the synthetic data is "realistic" or not



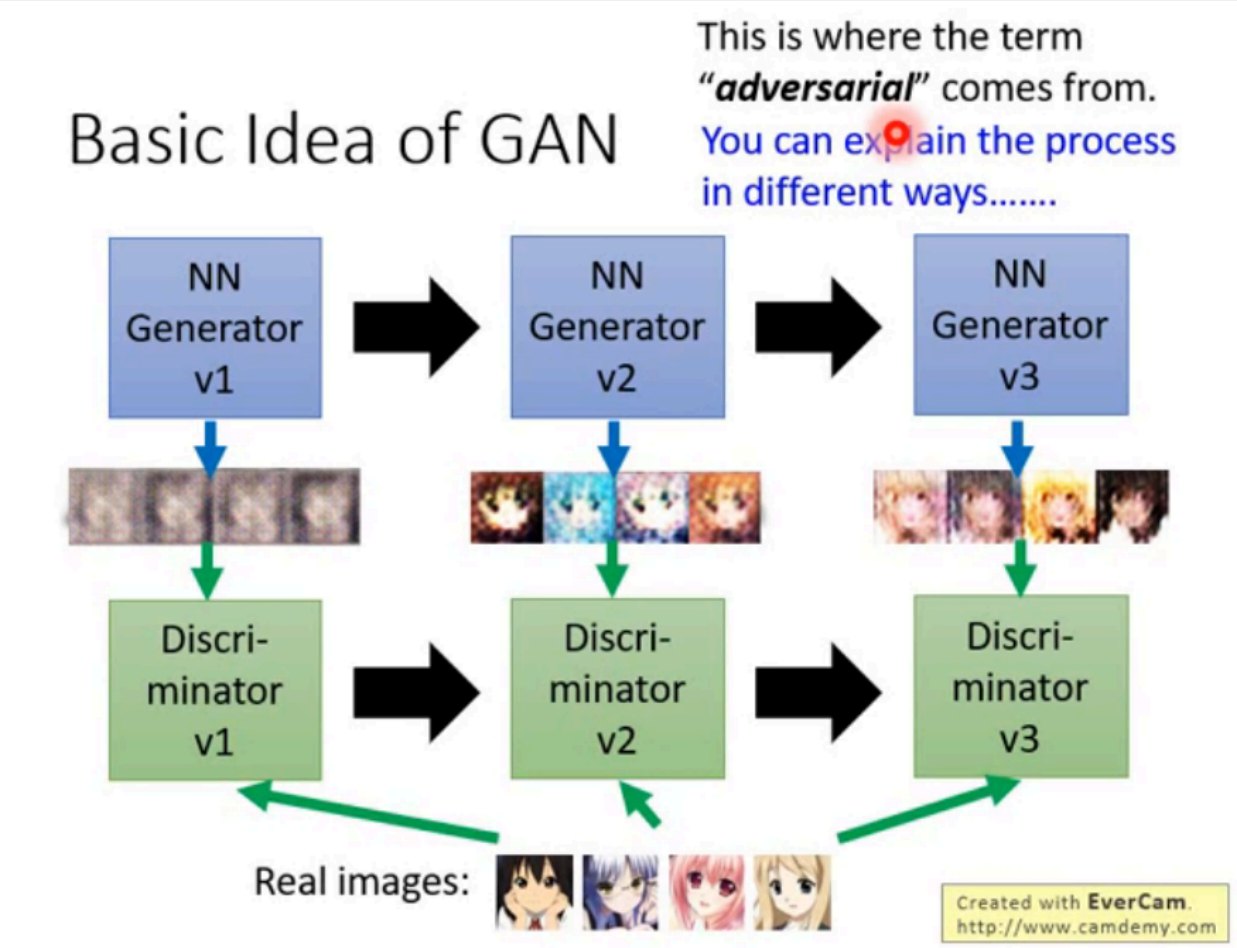
- To start: Generator v1 generates some synthetic data and Discriminator v1 tries to differentiate real and synthetic



- Generator v2's goal is to fool Discriminator v1 by improving synthetic data quality

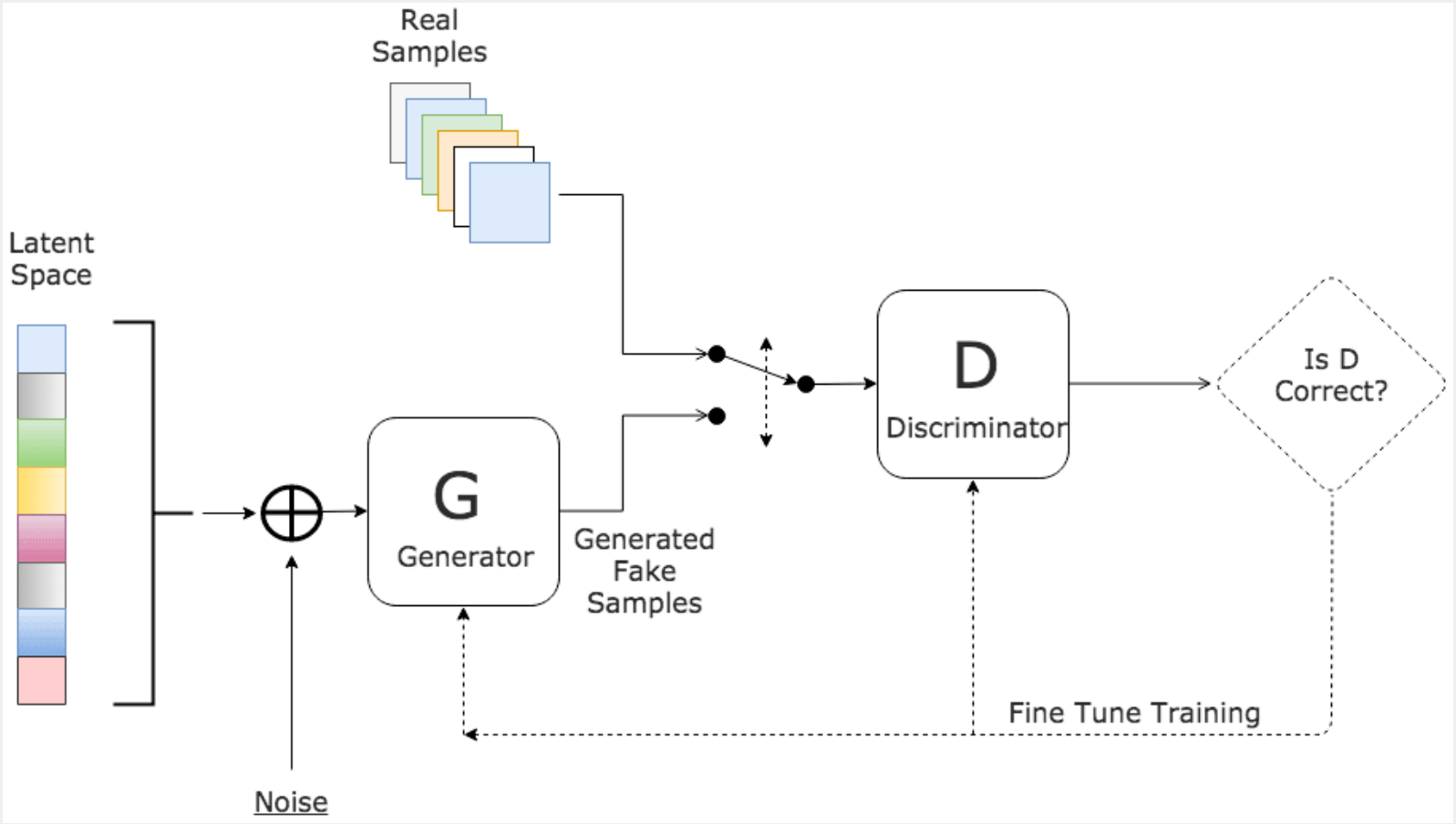


- Discriminator v2's goal is to defeat Generator v2 by identifying what is real and what is synthetic



- Step 1: initialize the parameters of generator and discriminator
- Step 2: fix generator, use it to generate synthetic data
- Step 3: sample real data from training set
- Step 4: **update** discriminator so that it can differentiate real and synthetic
- Step 5: fix discriminator, **update** generator till the current discriminator can no longer differentiate real and synthetic
- Step 6: go to Step 2

- During training, generator and discriminator are combined to form a large network with one hidden layer in the middle representing the generated data



Algorithm Initialize θ_d for D and θ_g for G

- In each training iteration:

Learning D

- Sample m examples $\{x^1, x^2, \dots, x^m\}$ from database
- Sample m noise samples $\{z^1, z^2, \dots, z^m\}$ from a distribution
- Obtaining generated data $\{\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^m\}$, $\tilde{x}^i = G(z^i)$
- Update discriminator parameters θ_d to maximize
  - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} log D(x^i) + \frac{1}{m}\sum_{i=1}^{m} log\left(1 - D(\tilde{x}^i)\right)$
  - $\theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d)$

Learning G

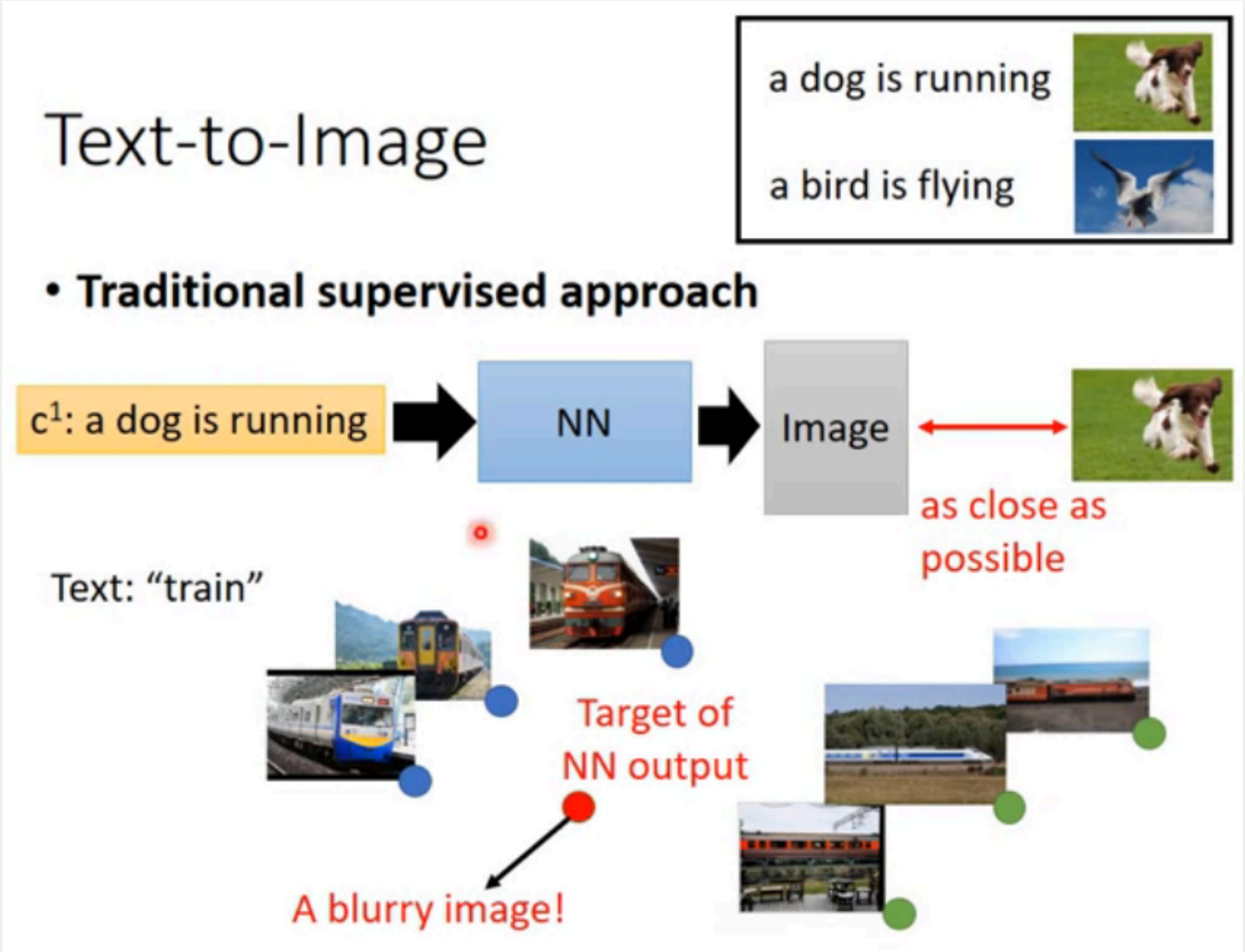
- Sample m noise samples $\{z^1, z^2, \dots, z^m\}$ from a distribution
- Update generator parameters θ_g to maximize
  - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} log\left(D\left(G(z^i)\right)\right)$
  - $\theta_g \leftarrow \theta_g + \eta \nabla \tilde{V}(\theta_g)$

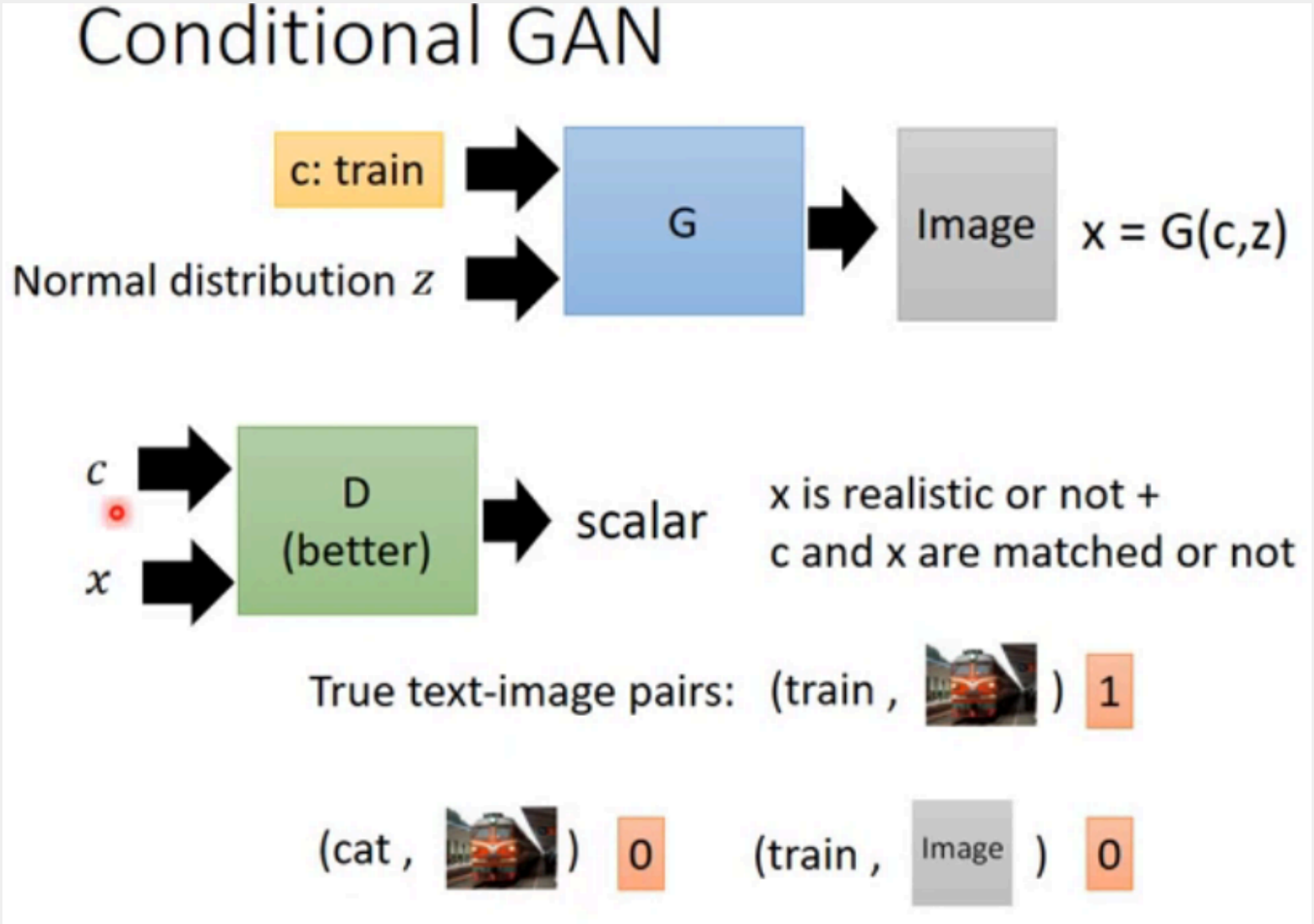
Created with EverCam

- link

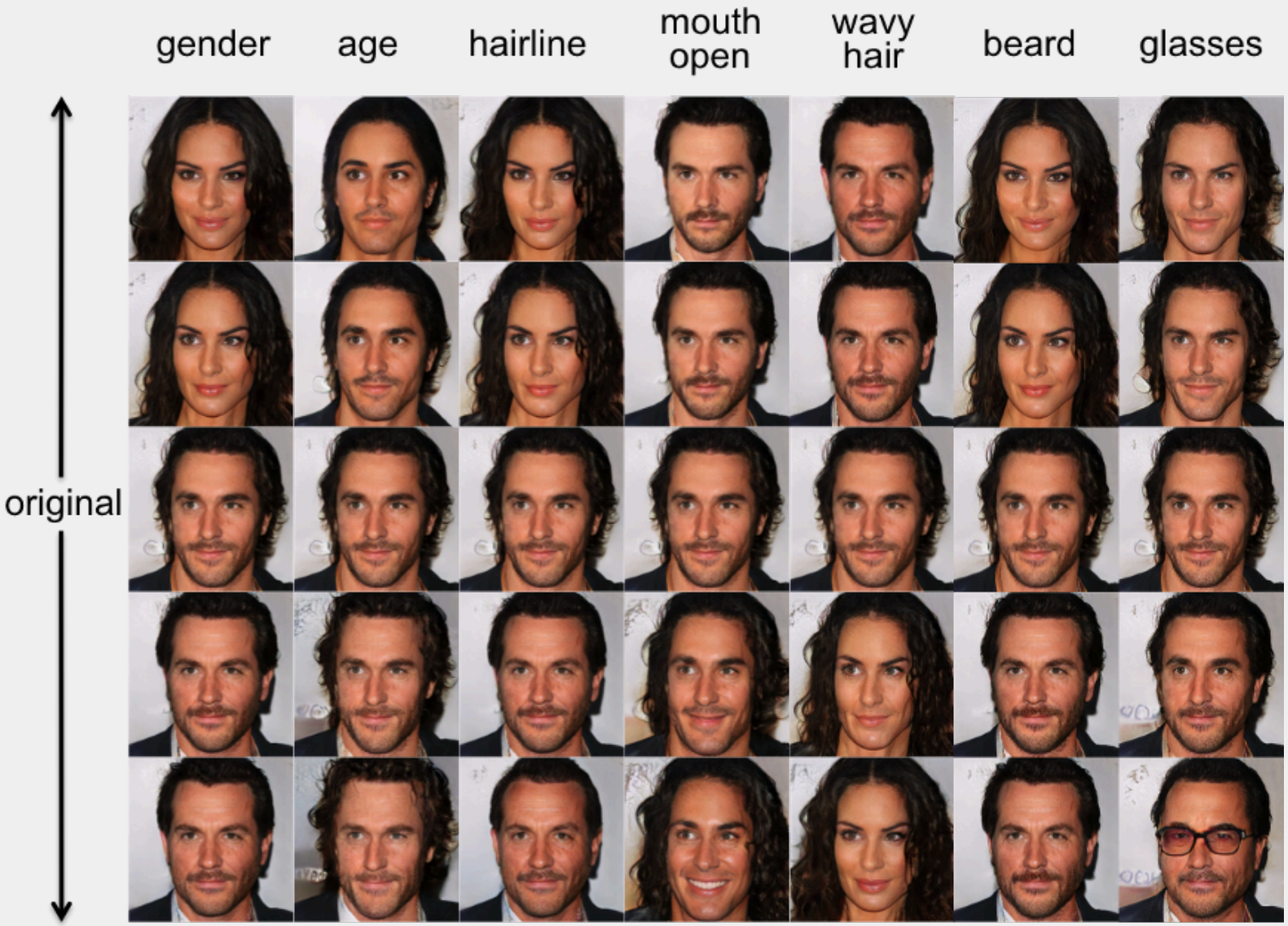
Conditional GAN

- Conditional GAN represents a class of useful GAN variations: supervised, unsupervised


Text-to-Image
a dog is running
a bird is flying
• Traditional supervised approach
c1: a dog is running
NN
Image
as close as possible
Text: "train"
Target of NN output
A blurry image!


Conditional GAN
c: train
Normal distribution z
G
Image
x = G(c,z)
c
x
D
(better)
scalar
x is realistic or not +
c and x are matched or not
True text-image pairs: (train ,) 1
(cat ,) 0
(train , Image) 0

Supervised Conditional GAN: text-to-image



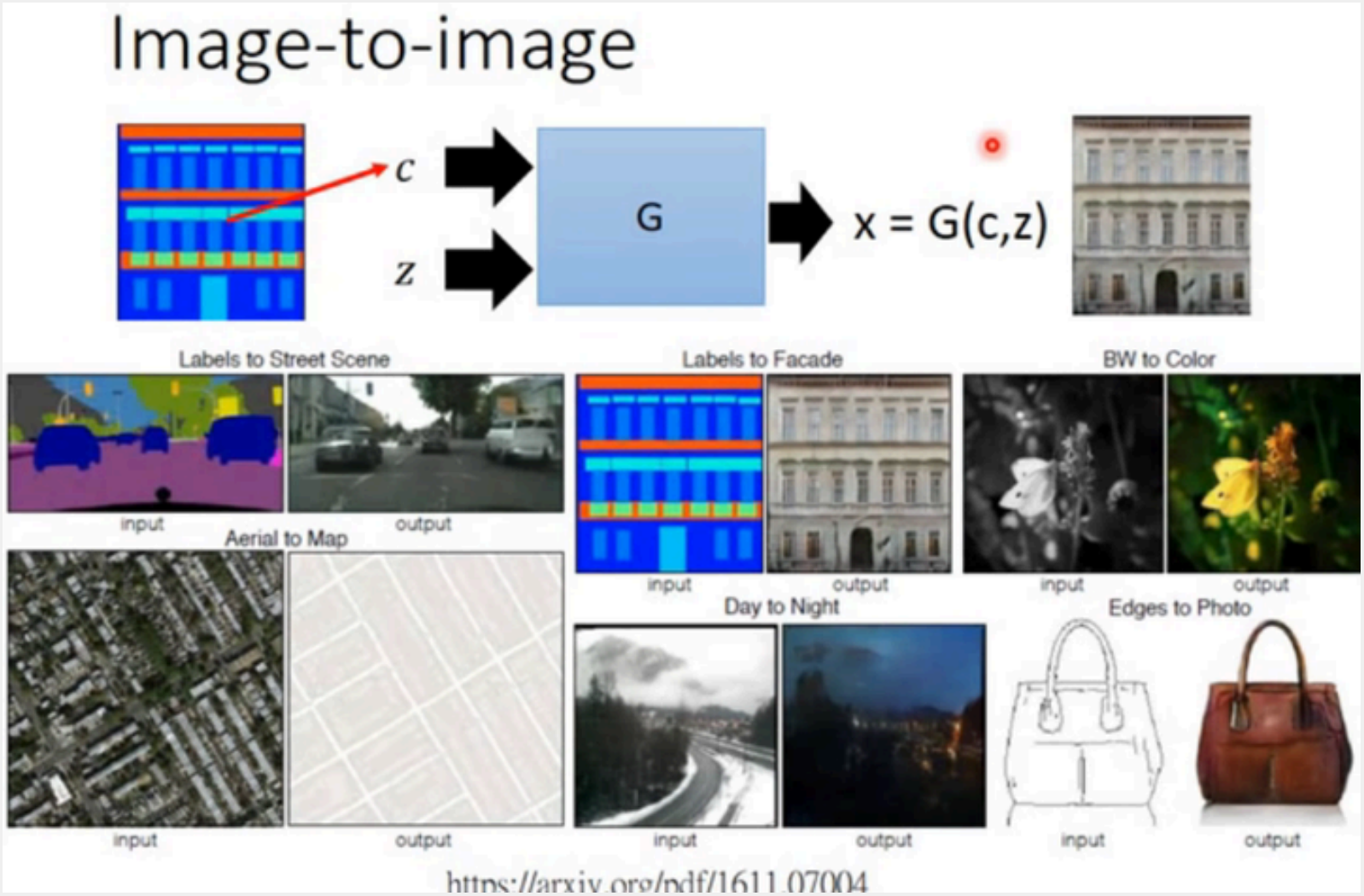
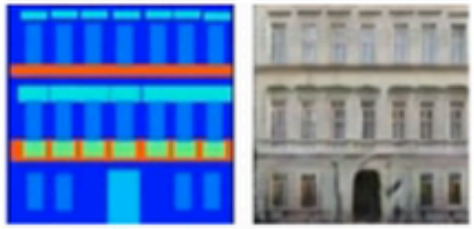
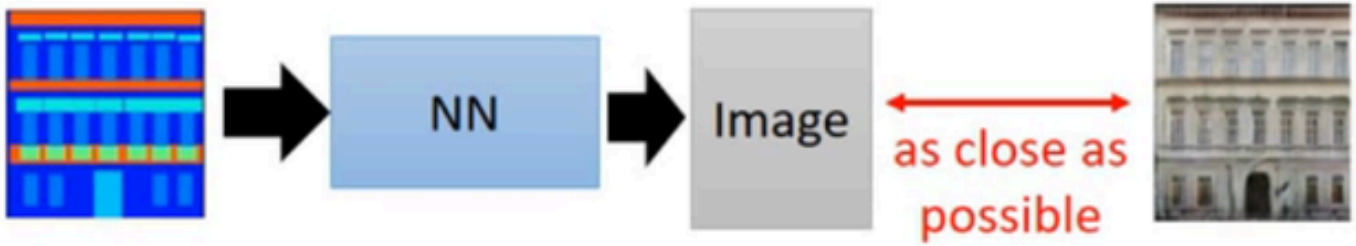


Image-to-image
c
z
G
x = G(c,z)
Labels to Street Scene
input
output
Labels to Facade
input
output
BW to Color
input
output
Aerial to Map
input
output
Day to Night
input
output
Edges to Photo
input
output
https://arxiv.org/pdf/1611.07004

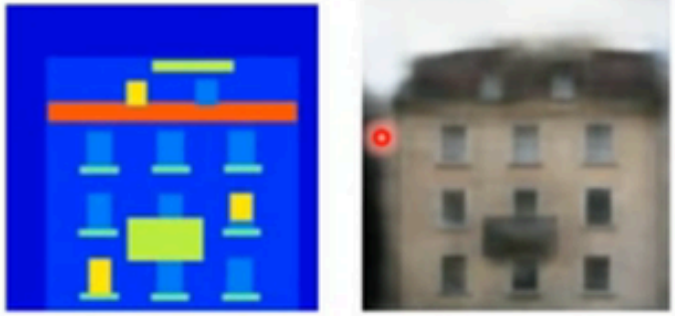
Image-to-image



- Traditional supervised approach



Testing:



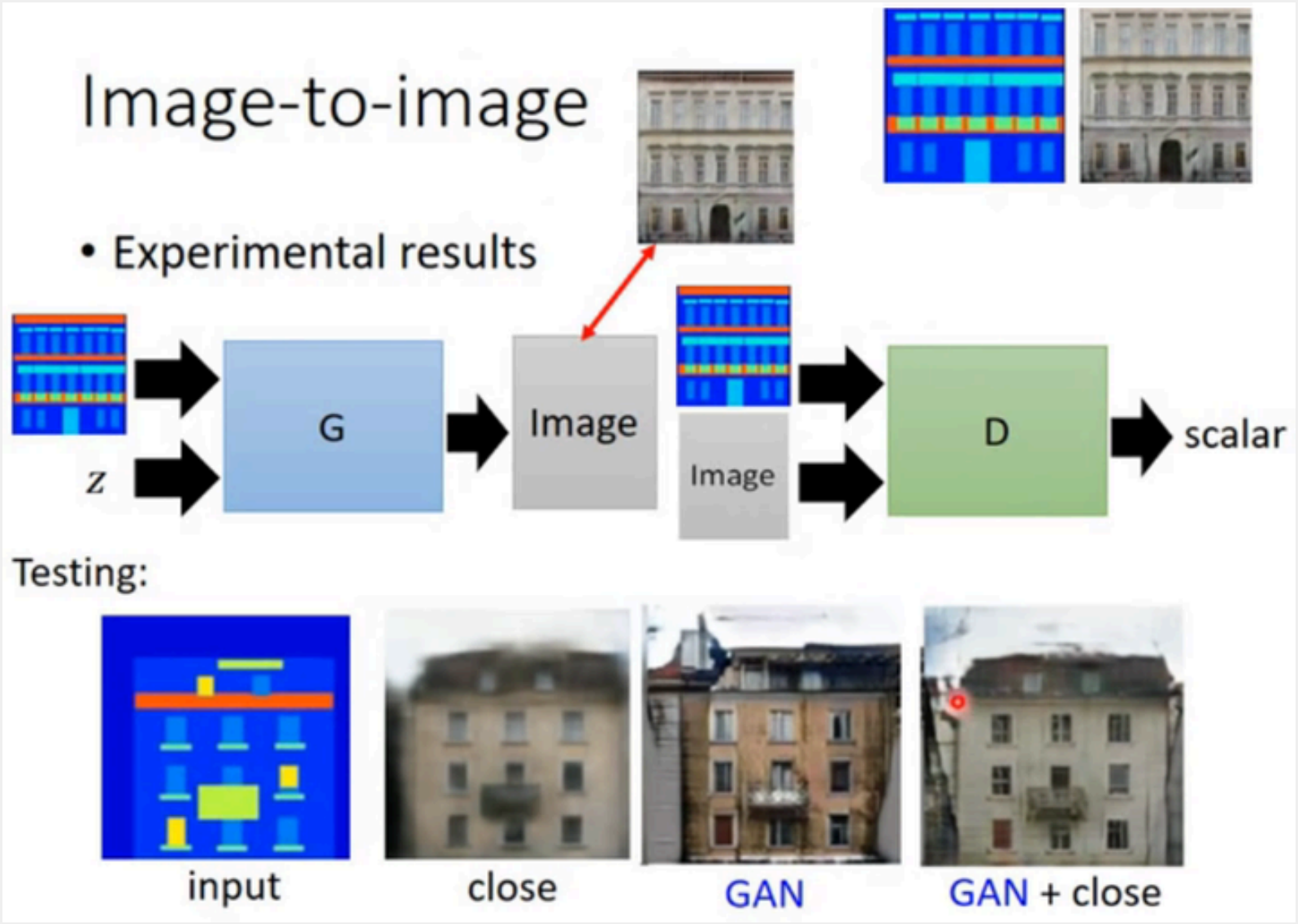
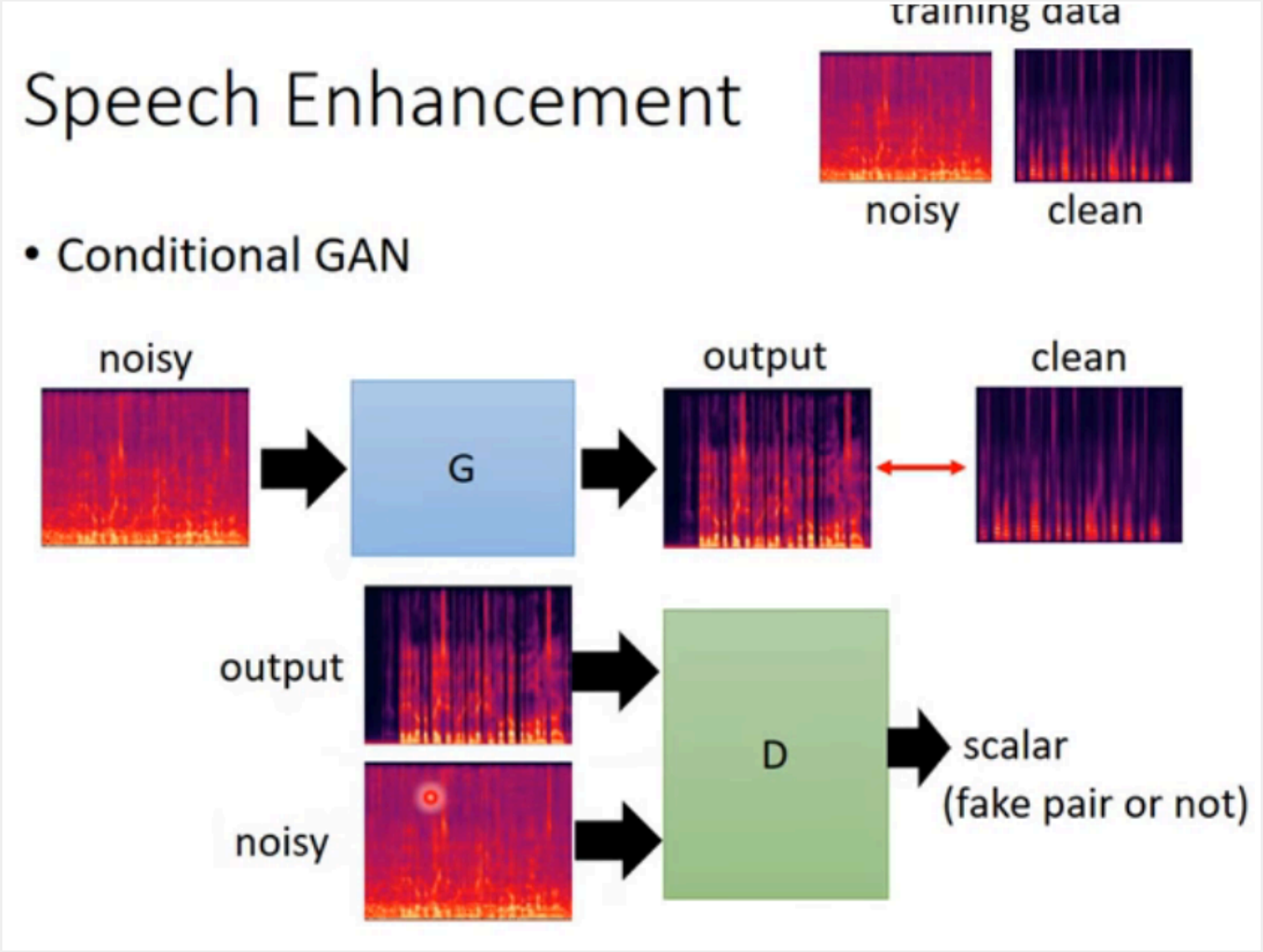


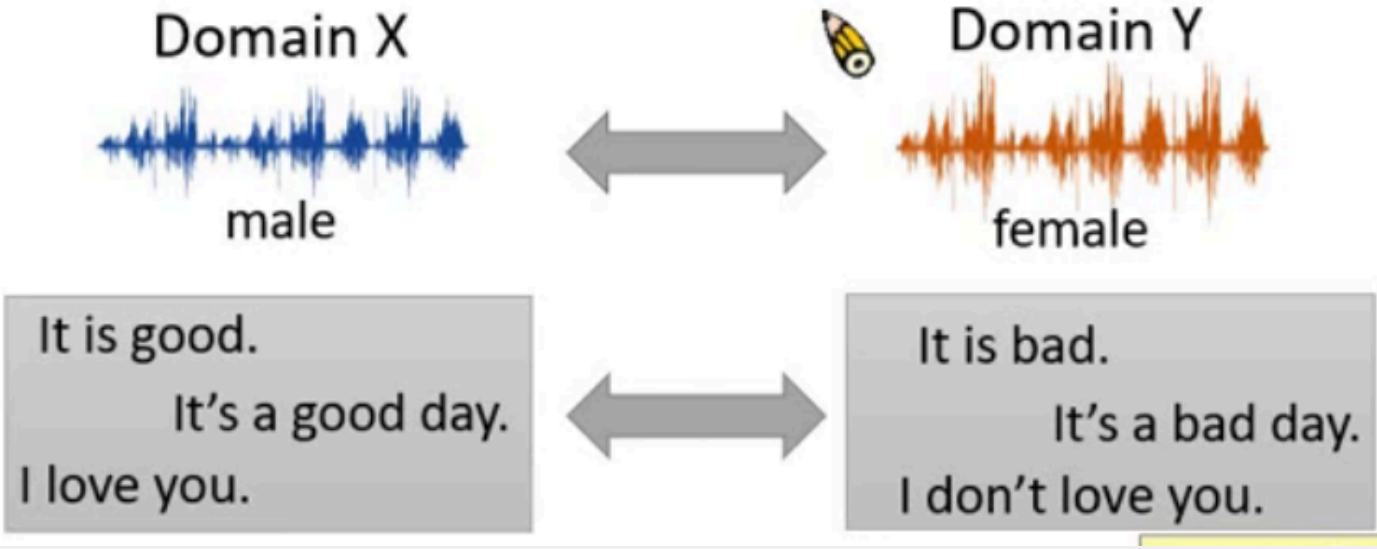
Image-to-image
• Experimental results
G
Image
Image
D
scalar
z
Testing:
input
close
GAN
GAN + close


training data
Speech Enhancement
noisy
clean
• Conditional GAN
noisy
G
output
clean
output
noisy
D
scalar
(fake pair or not)

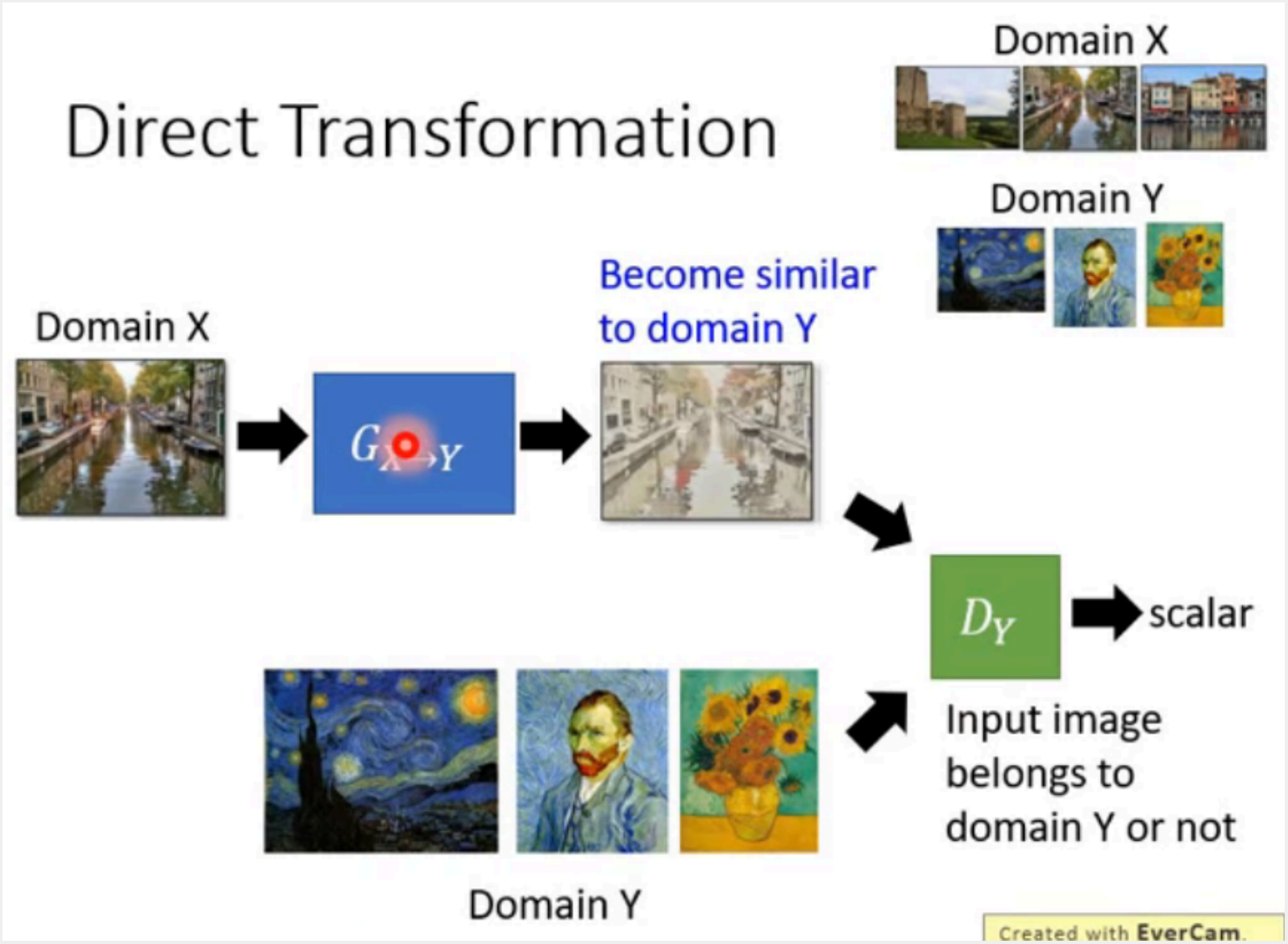
Unsupervised Conditional Generation

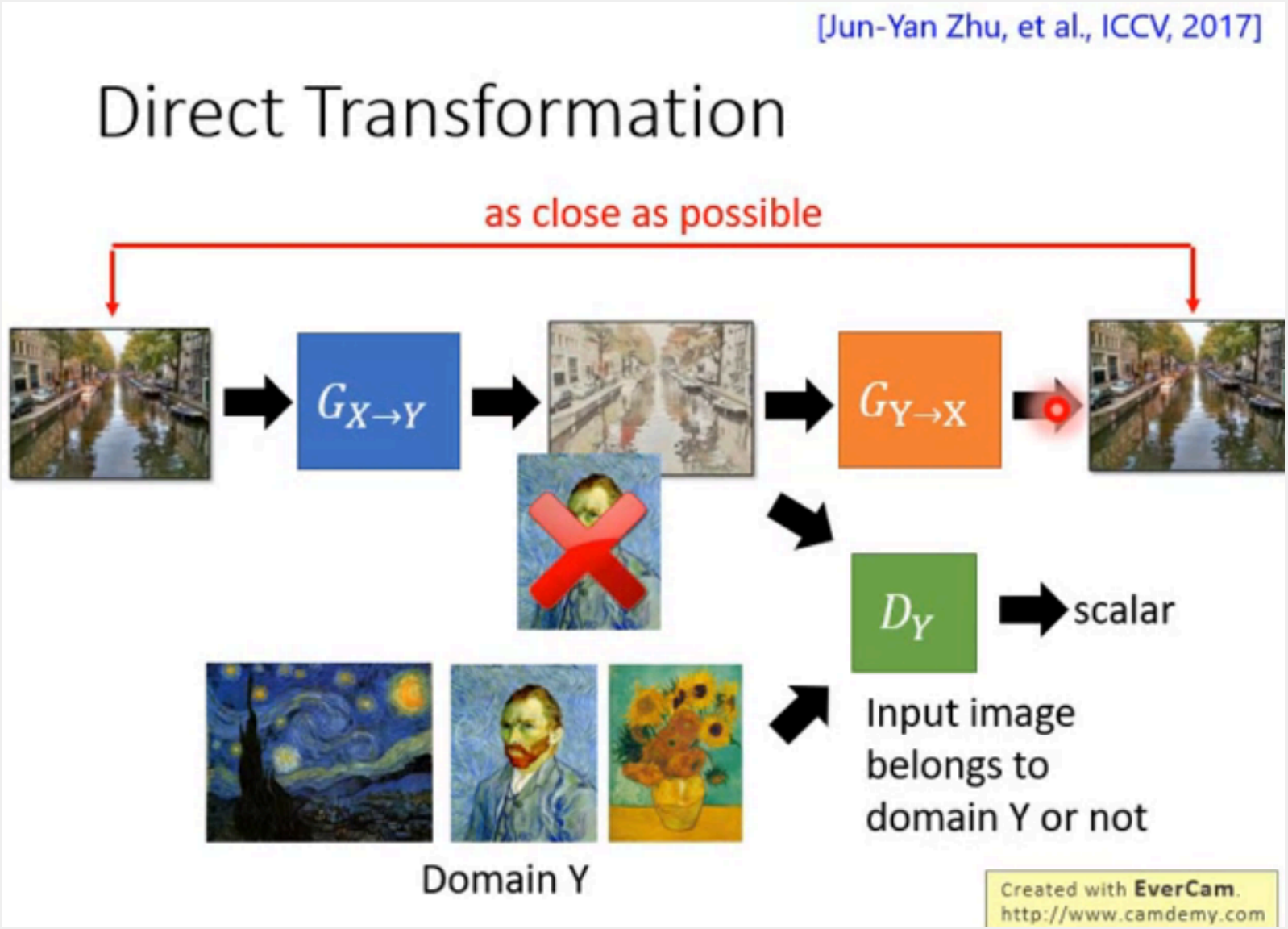
Transform an object from one domain to another

without paired data (e.g. style transfer)



- Application: voice style transfer
- Past: need speech of the *same content* from two people
- Now: only need speech from two people


Direct Transformation
Domain X
Domain Y
Domain X
Become similar to domain Y
$G_{X \to Y}$
D_Y
scalar
Input image belongs to domain Y or not
Domain Y
Created with EverCam.


[Jun-Yan Zhu, et al., ICCV, 2017]
Direct Transformation
as close as possible
$G_{X\to Y}$
$G_{Y\to X}$
D_Y
scalar
Input image belongs to domain Y or not
Domain Y
Created with EverCam.
http://www.camdemy.com

Example Code

- Conditional GAN: link