國立交通大學
National Chiao Tung University

Introduction to Deep Generative Models

Wei-Chen Chiu

邱維辰

April 25th, 2018

Enriched Vision Applications Laboratory

BSc in EECS
MSc in CS

Doctor of Engineering

Viscovery

Research Scientist

Postdoctoral Researcher

Assistant Professor, CS Dept.

Outlines

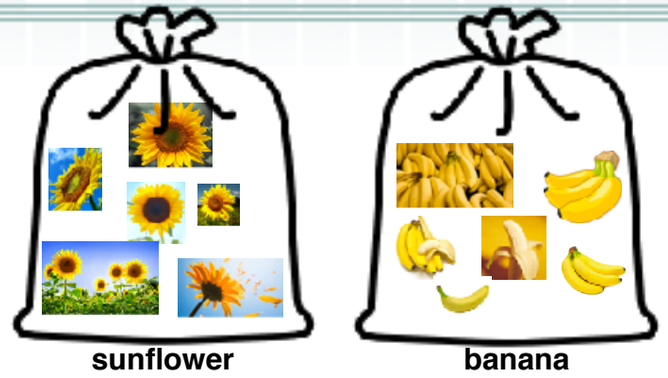
- **Discriminative versus Generative Models**
- Going Into Deep Generative Models
- From Autoencoder to Variational Autoencoder (VAE)
- From VAE to Generative Adversarial Network (GAN)
- Various Applications
- Understanding the latent space: disentanglement

資訊工程學系
Department of Computer Science

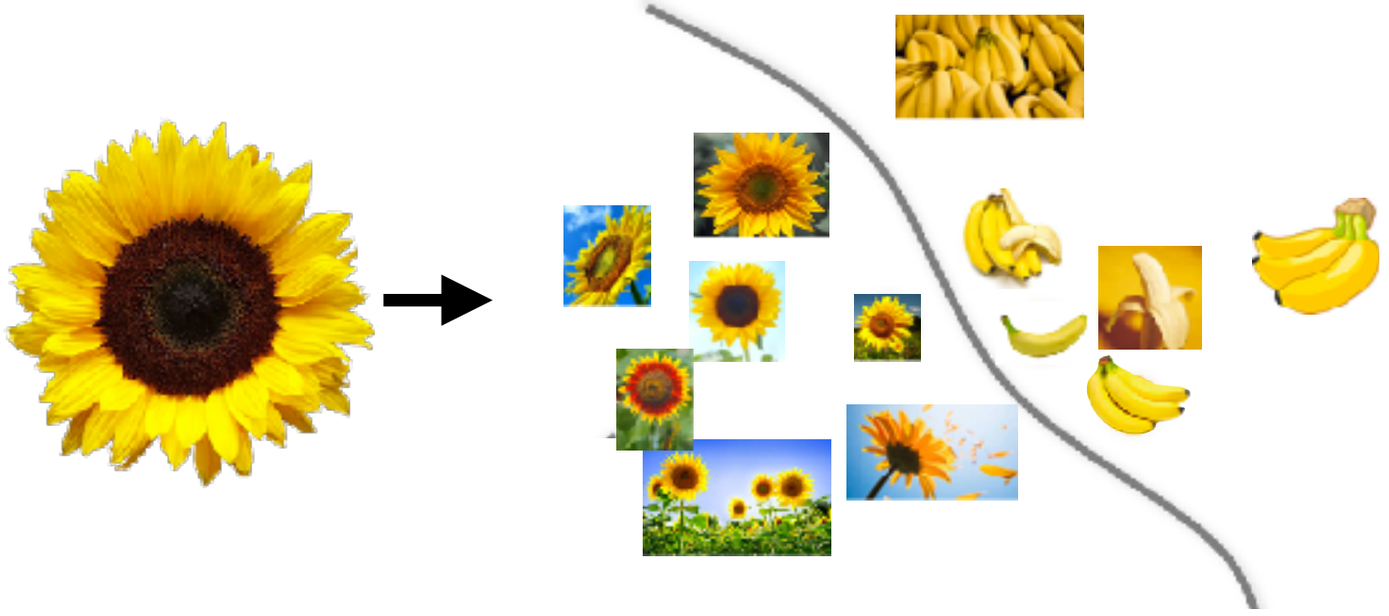
Discriminative versus Generative Models

- **Discriminative**
  - model **posterior**

$$p(z|x)$$



learn a decision boundary in a specific feature space



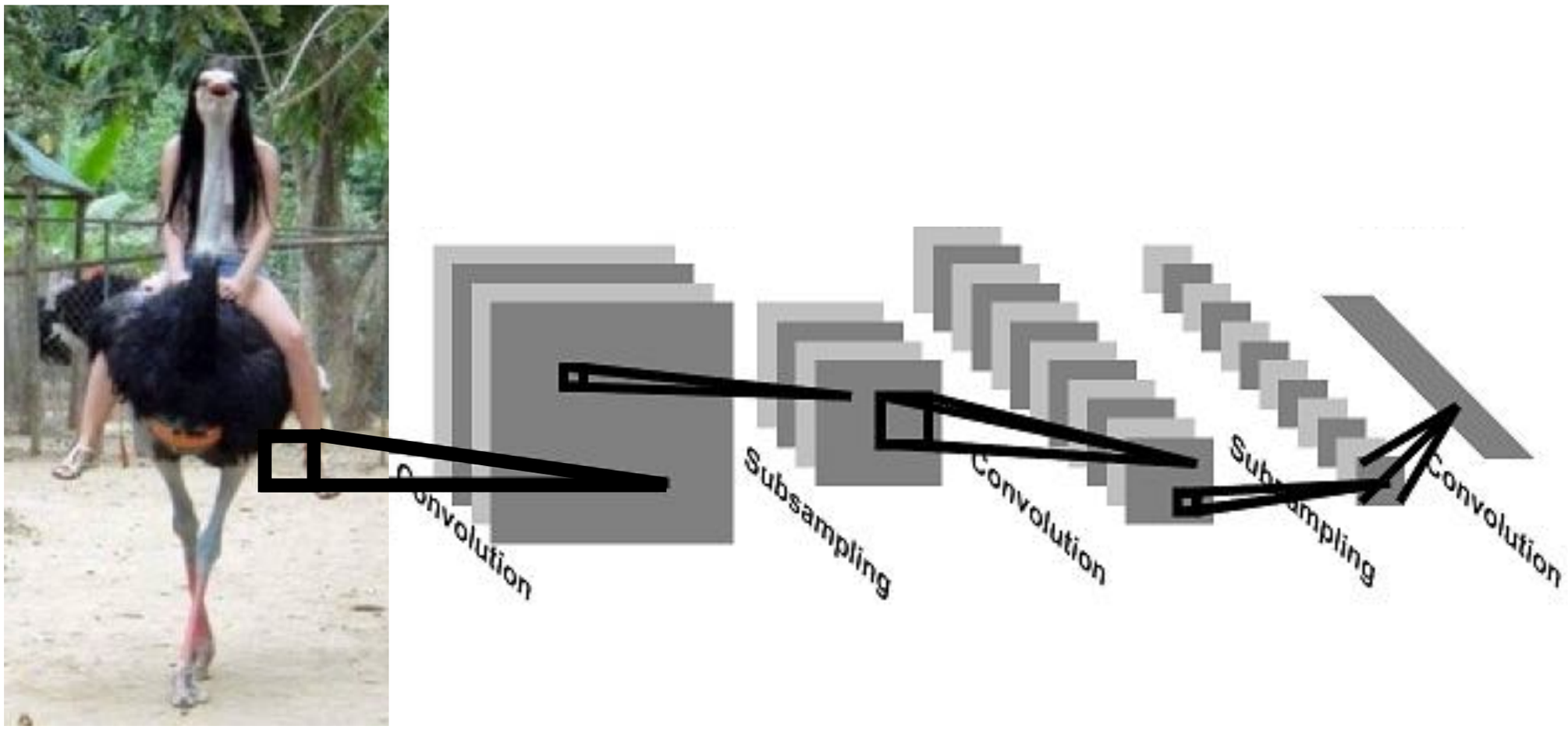
x z

Question: Given an input, how likely its class is sunflower or banana?

Discriminative versus Generative Models

- **Discriminative**
  - model **posterior**

$$p(z|x)$$



x z

Question: Given an input, how likely its class is girl or ostrich?

Discriminative versus Generative Models

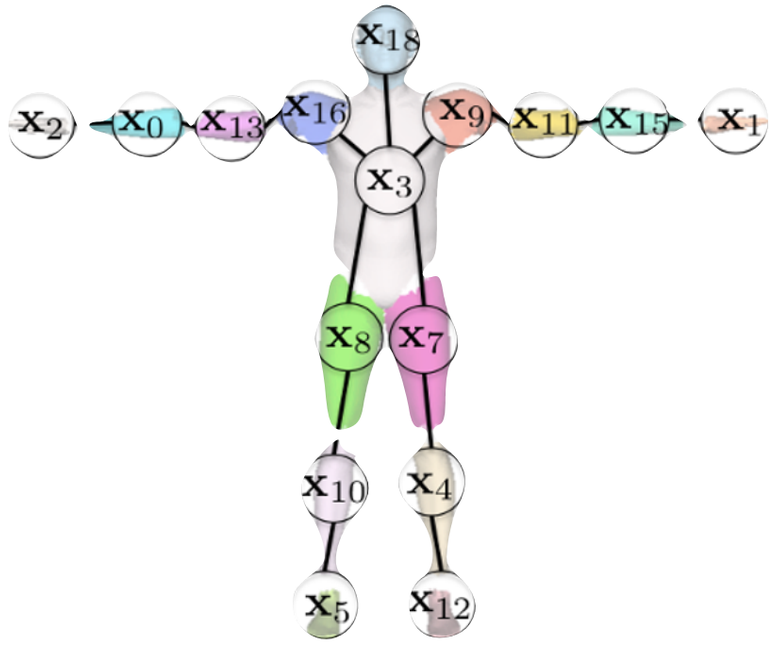
- **Discriminative**
  - model **posterior**

$$p(z|x)$$

- **Generative**
  - model **likelihood w/ prior**

$$p(x|z) \cdot p(z)$$

$x|z$



z

Question: I know how human pose can be,
given a pose, how well it fit the body of Mayor Ko?

Why Generative Models?

- **Generative**
  - model **likelihood w/ prior**

$$p(x|z) \cdot p(z)$$

☞ Given data, train a model to generate samples like it

Why Generative Models?

- **Generative**
  - model **likelihood w/ prior**

$$p(x|z) \cdot p(z)$$

data: images of dog

model

samples

☞ Given data, train a model to generate samples like it

$p(z)$ ***latent variables: attributes of dogs e.g. color, species***

$p(x|z)$ ***given a setting of attributes, how the dog will look like?***

☞ process of data formation

Why Generative Models?

- **Generative**
  - model **likelihood w/ prior**

$$p(x|z) \cdot p(z)$$

☞ Given data, train a model to generate samples like it

data: images of dog

$p(z)$ *latent variables: attributes of dogs e.g. color, species*

$p(x|z)$ *given a setting of attributes, how the dog will look like?*

model ☞ process of data formation

HOW?

Richard Feynman: "**What I cannot create, I do not understand**"

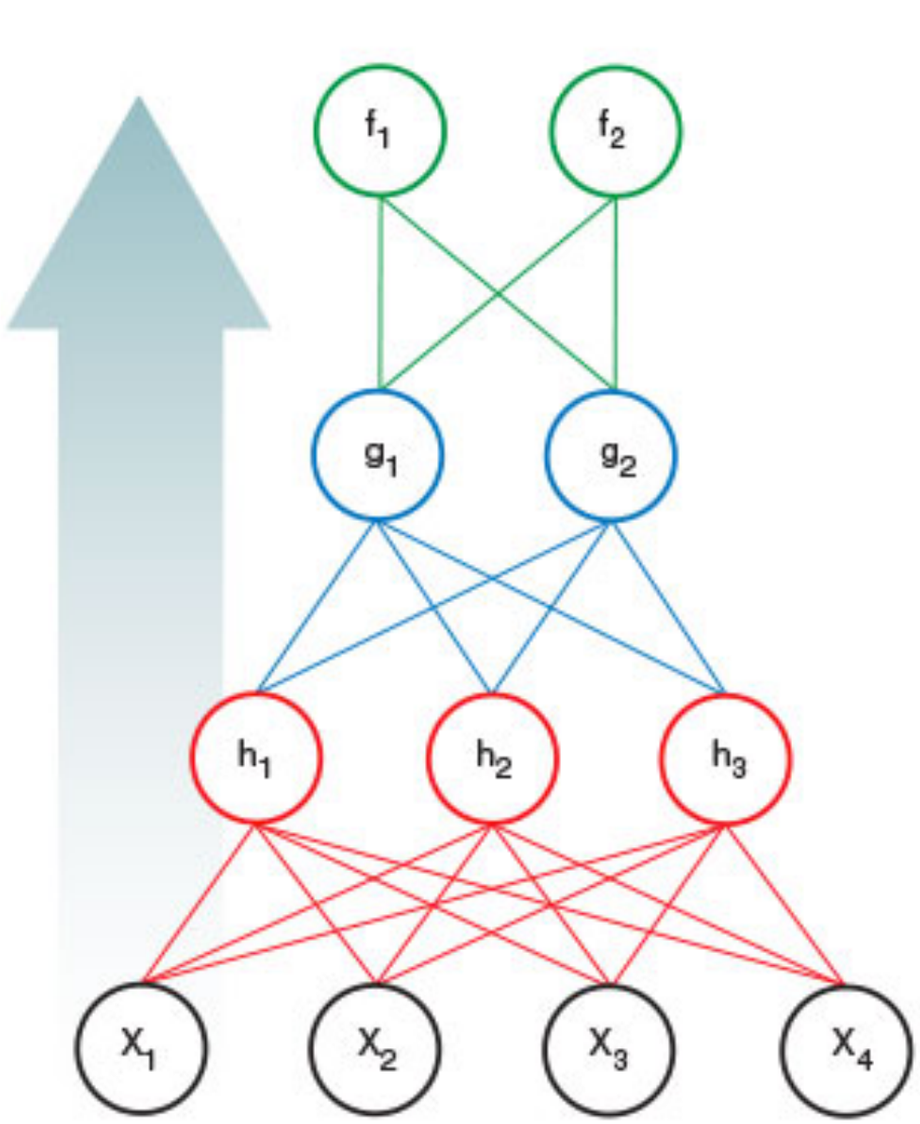
[Ref: https://openai.com/blog/generative-models/]

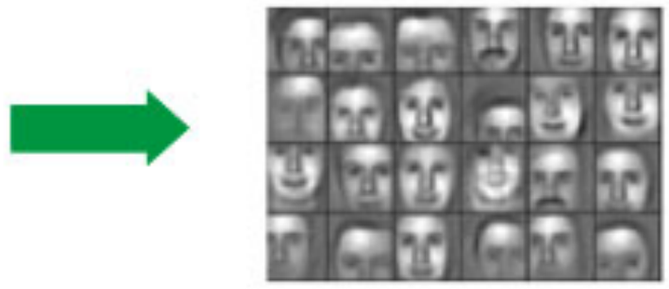
Outlines

- Discriminative versus Generative Models
- **Going Into Deep Generative Models**
- From Autoencoder to Variational Autoencoder (VAE)
- From VAE to Generative Adversarial Network (GAN)
- Various Applications
- Understanding the latent space: disentanglement

All We Want to Know is Latent Variables

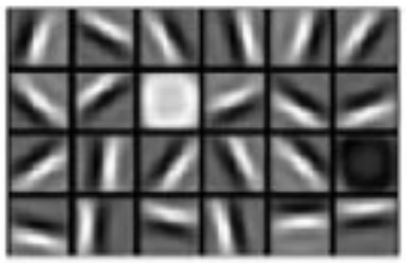
- More "**semantic**" representation in the **latent space *z***





Object models

Object parts (combination) of edges

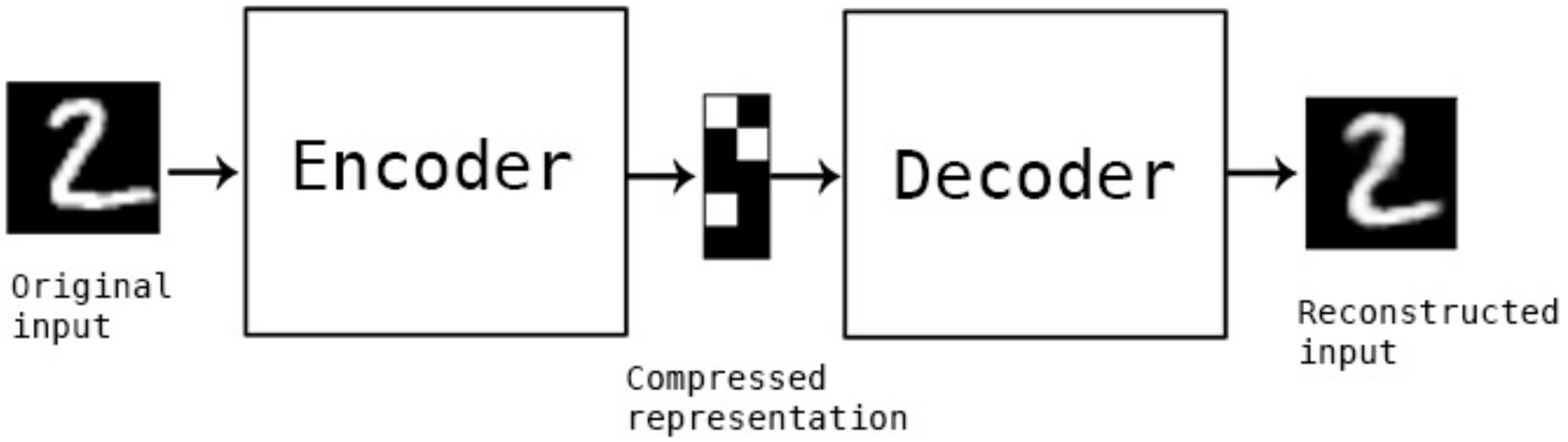


Edges at various orientations

Input pixels

latent variables: color, species

A Novel Way To Discover Latent Variables

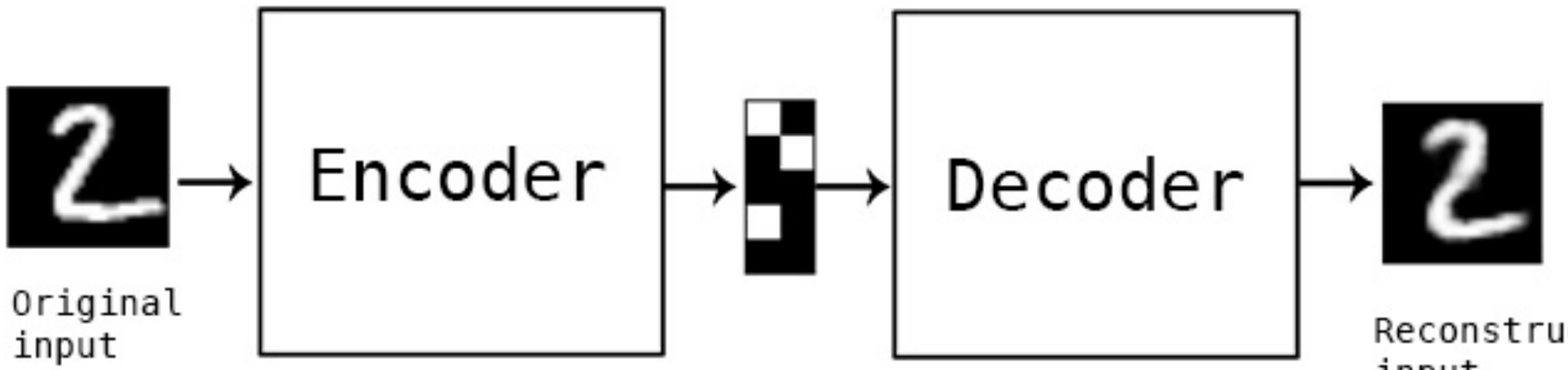


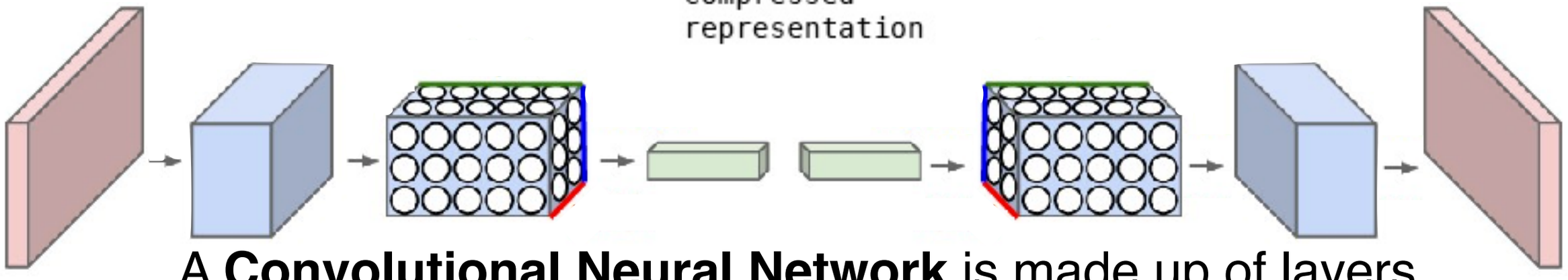
most important information
of input image ☞ latent space

named as **auto-encoder**, once superstar few years ago…

Now Include Our Favourite Deep Models

: images formation includes almost infinite parameters, use deep models with known largest capacities

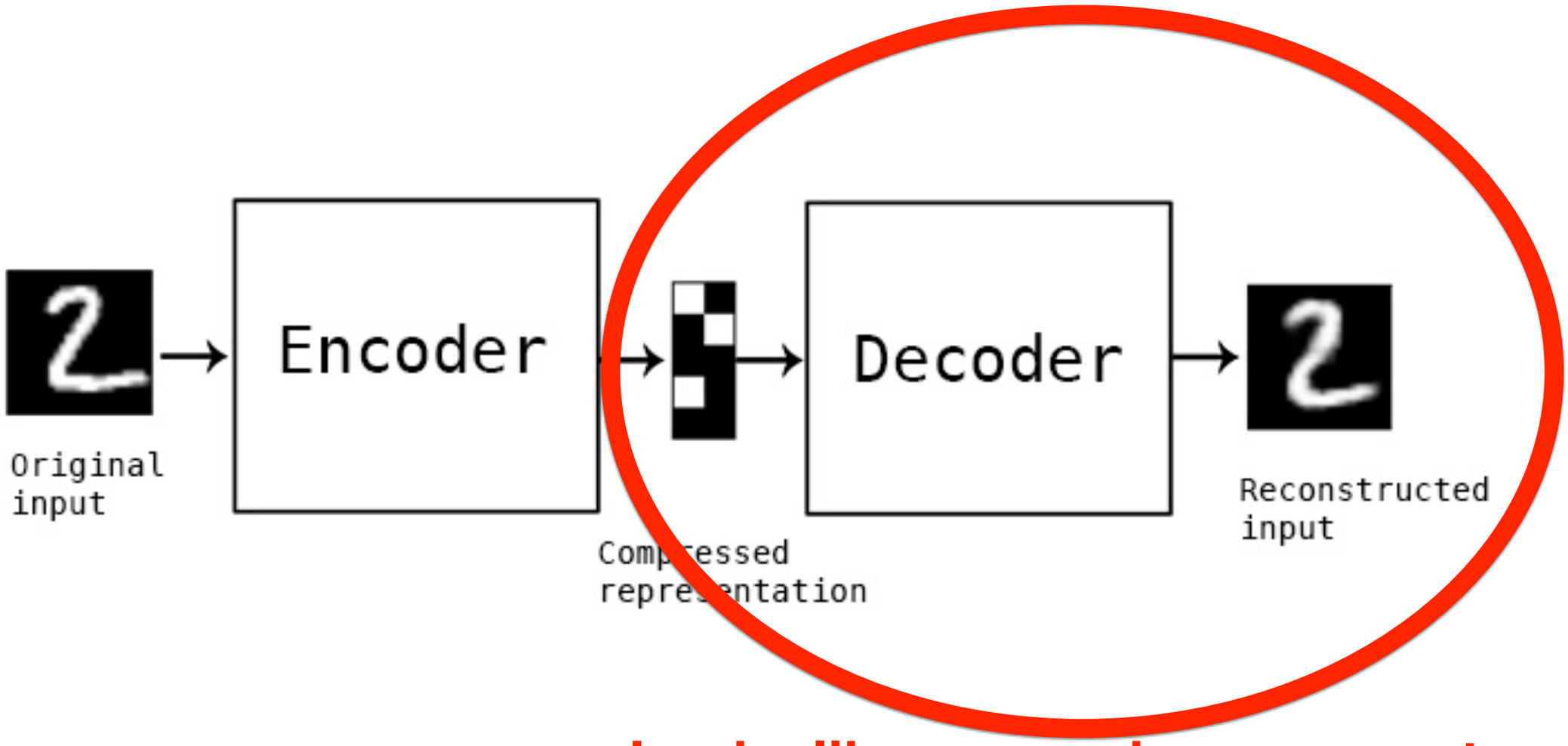




A **Convolutional Neural Network** is made up of layers.
Every layer transforms input to output with some differentiable functions.

☞ simply treat as a super powerful function

Okay, Back to Our Autoencoder Story First



Looks like we can have generator for free here?

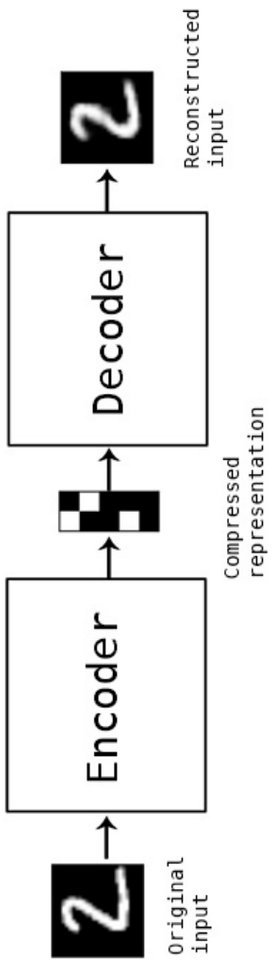
like the drinks in the hotel
<u>not</u> for free!

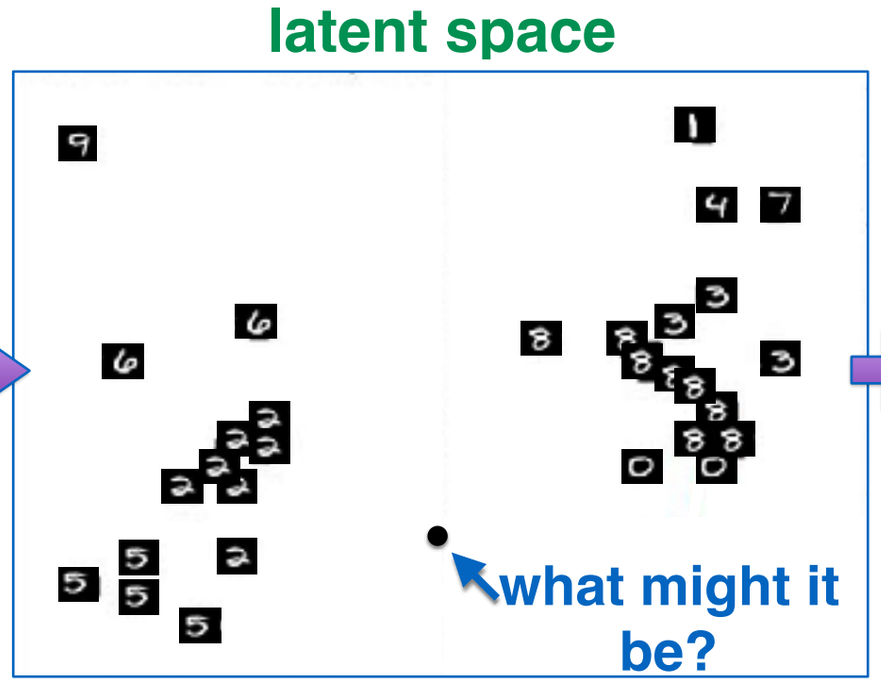
Outlines

- Discriminative versus Generative Models
- Going Into Deep Generative Models
- **From Autoencoder to Variational Autoencoder (VAE)**
- From VAE to Generative Adversarial Network (GAN)
- Various Applications
- Understanding the latent space: disentanglement

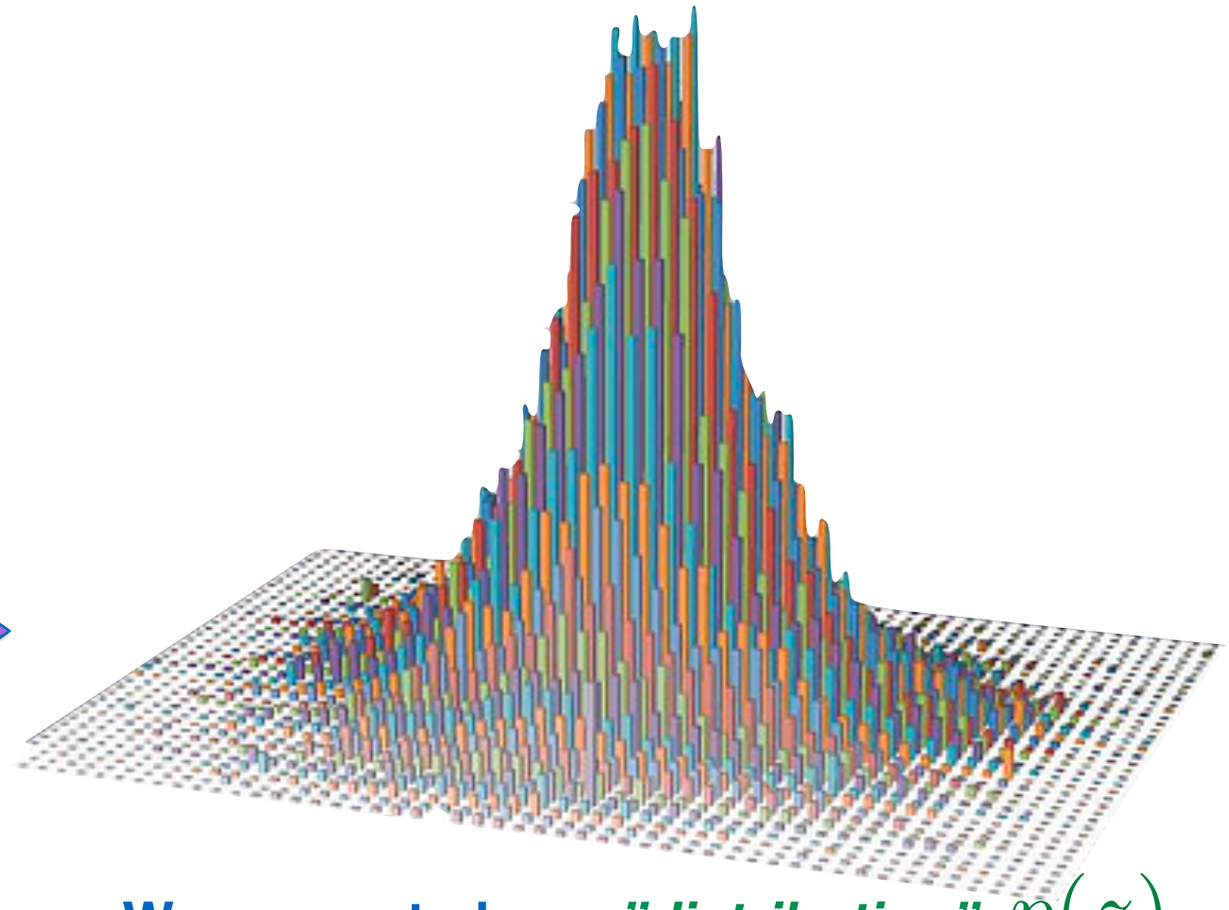
What Are Problems of Free Generator from Autoencoder?

資訊工程學系
Department of Computer Science





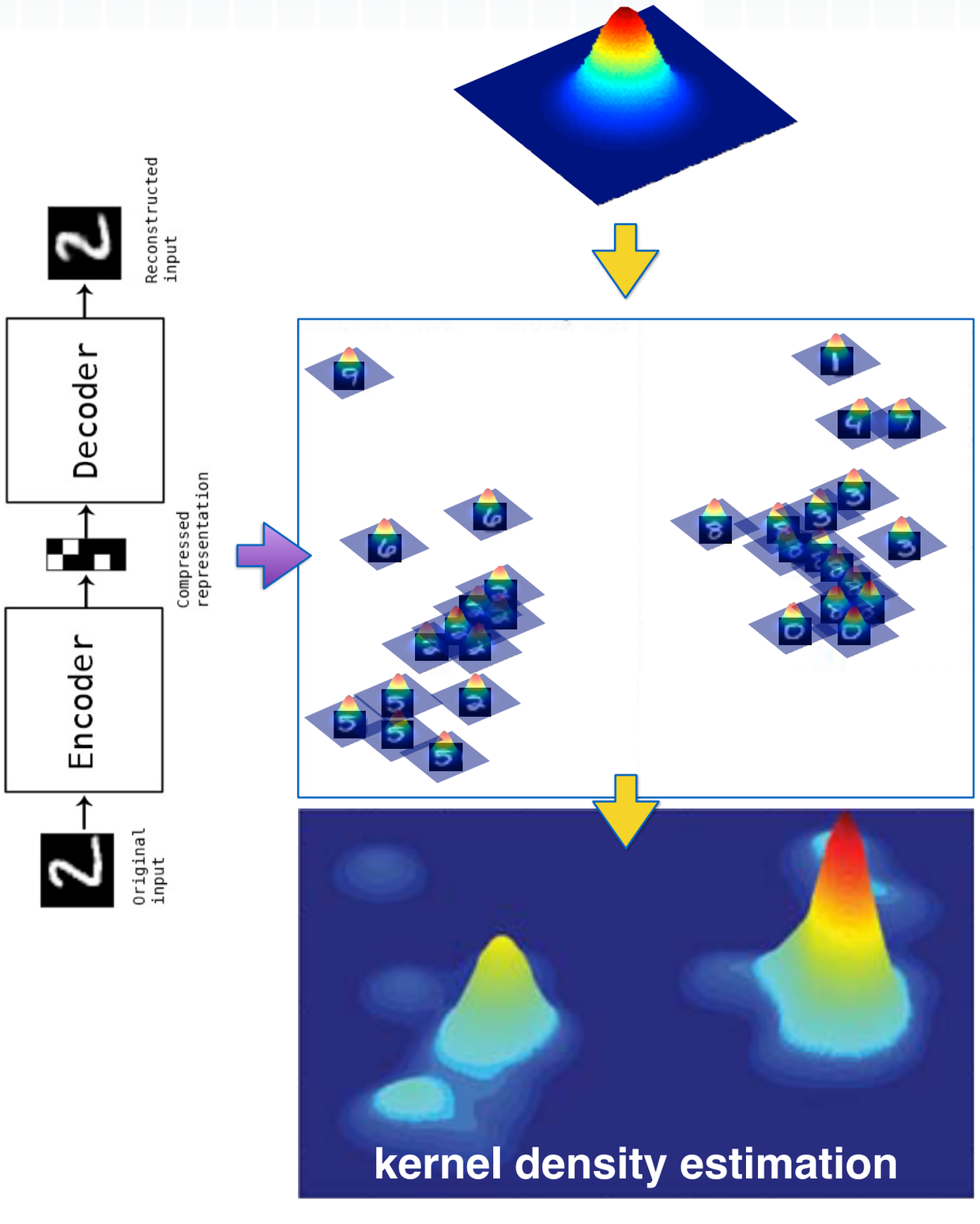
Only samples, how about other regions not covered?

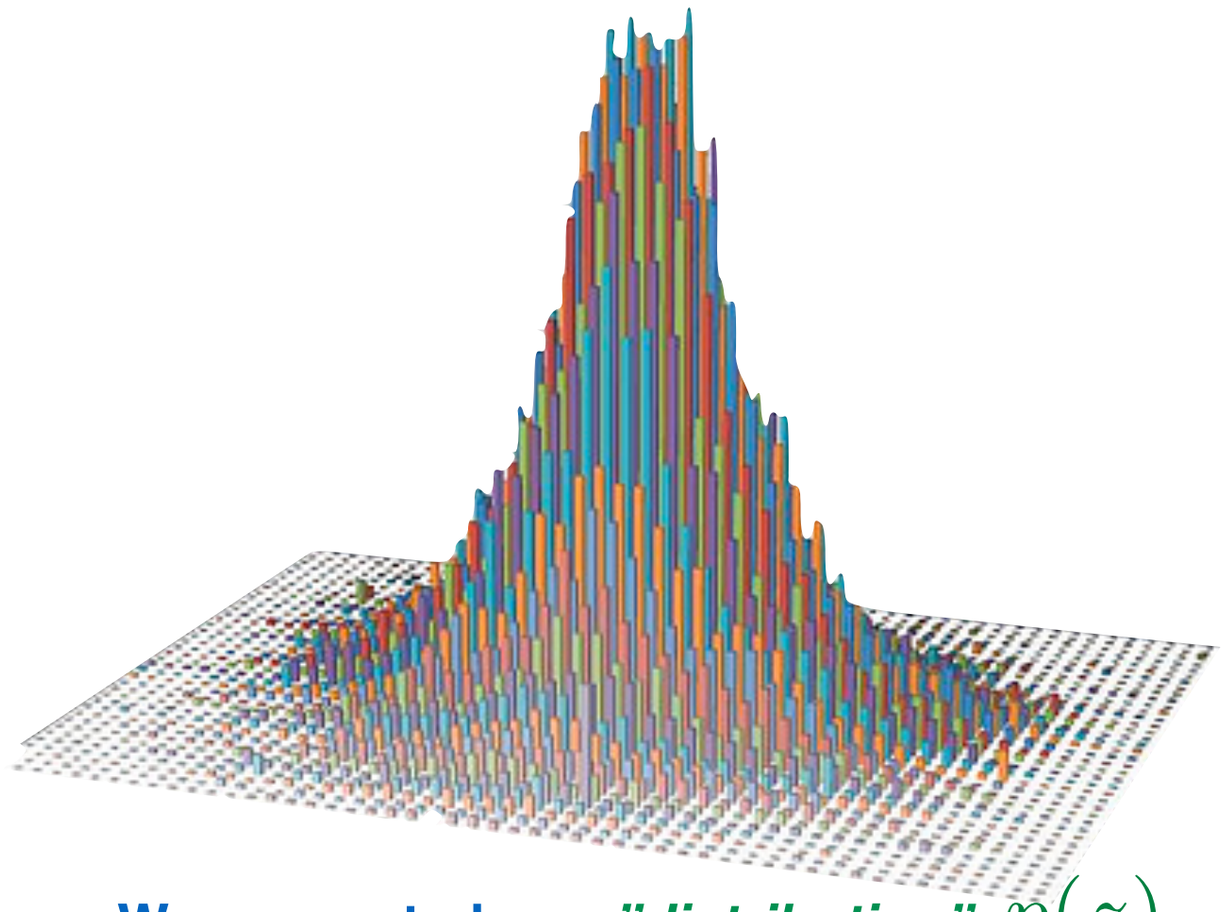


We wanna to have ***"distribution"*** $p(z)$ where we can sample from any location

What Are Problems of Free Generator from Autoencoder?

資訊工程學系
Department of Computer Science





We wanna to have *"distribution"* $p(z)$
where we can sample from any location

get them closer, so we can sample easily

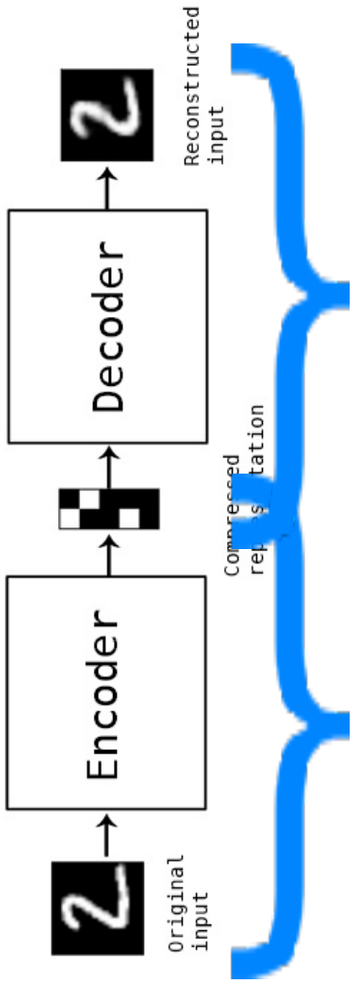
資訊工程學系
Department of Computer Science

Here Comes Variational Autoencoder (VAE)

D. Kingma

$$p(x) = \int p(x|z;\theta)p(z)dz$$

Original intuition: maximize probability of each x in training set under the entire generative process



☞ sample z that are likely to have produced x, compute $p(x)$

$$E_{z\sim Q}P(x|z) \Leftrightarrow p(x)$$

maximize

☞ new function $Q(z|x)$ takes x and gives distribution over z

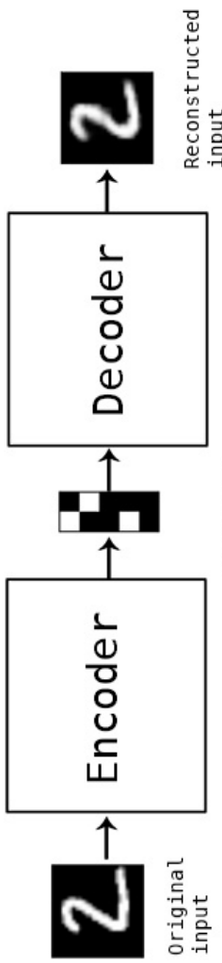
$Q(z|x) \Leftrightarrow p(z)$ **variational! minimize their KL divergence**

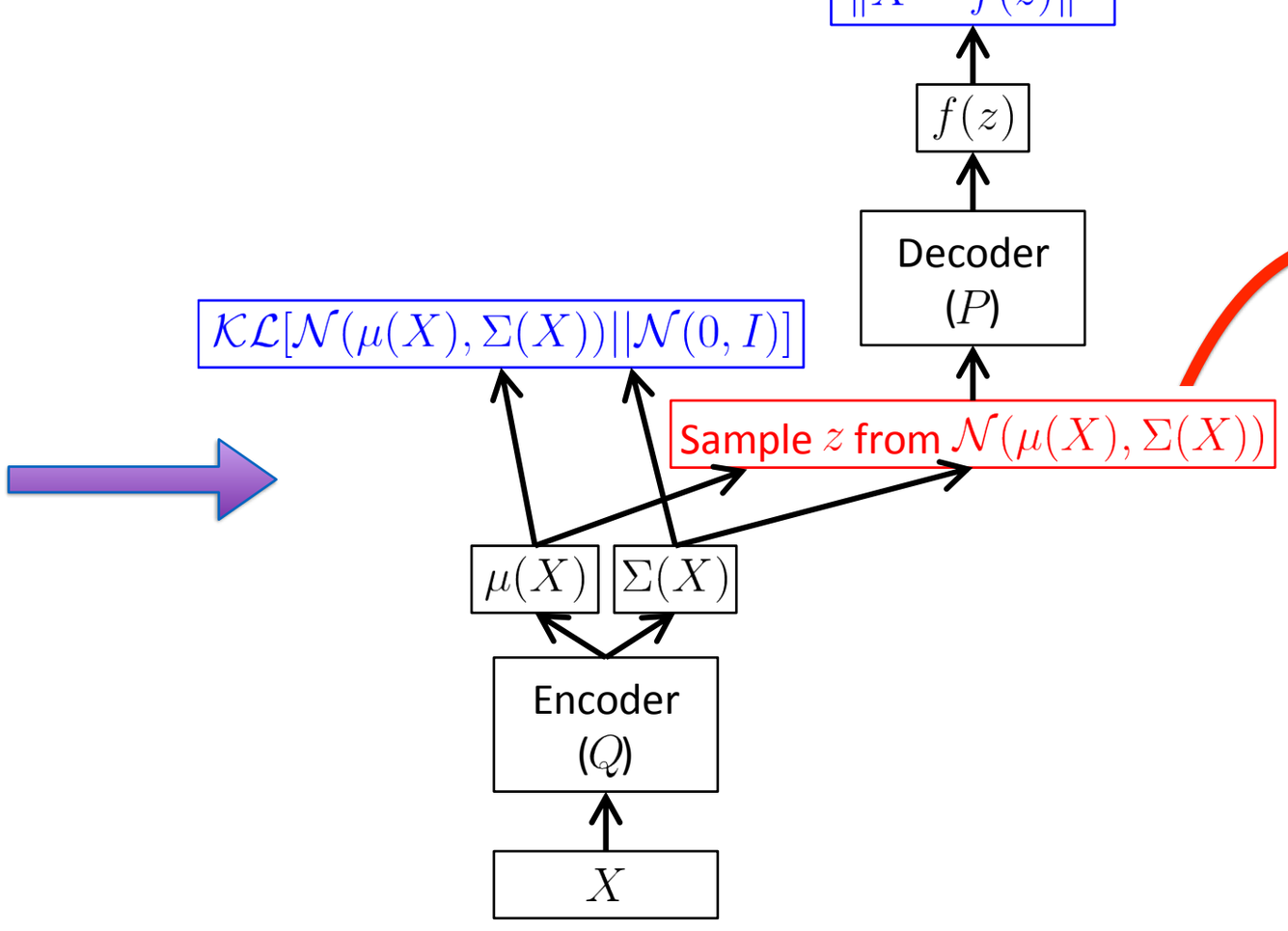
Now is a "*distribution*", we can assume it to be a distribution easy to sample from, e.g. Gaussian

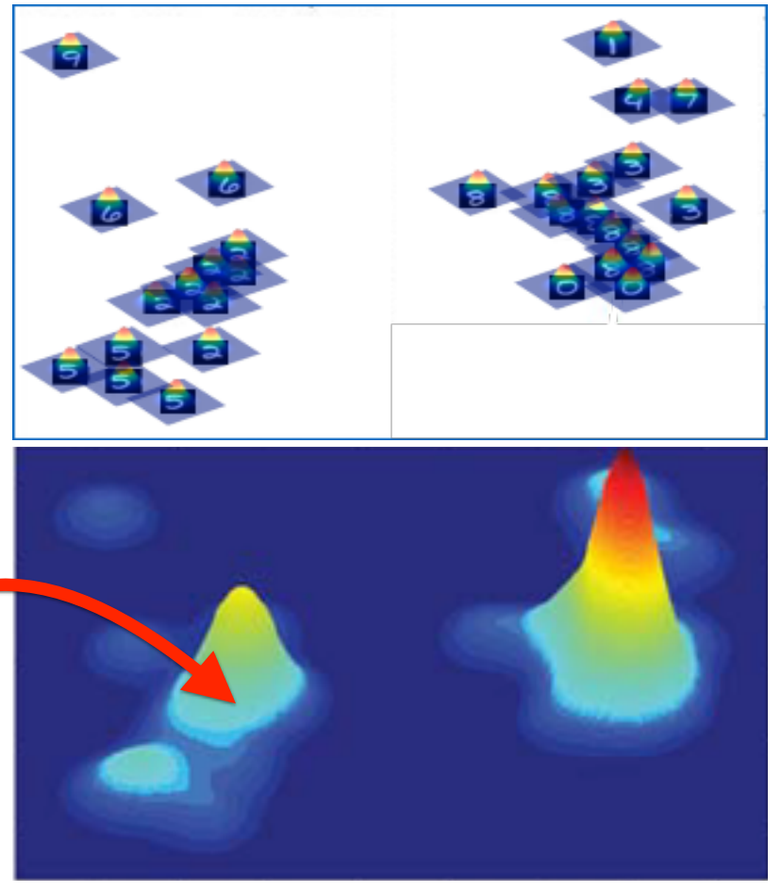
資訊工程學系
Department of Computer Science

Here Comes Variational Autoencoder (VAE)

D. Kingma





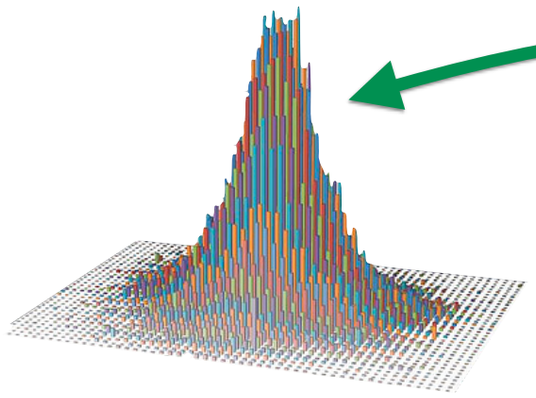


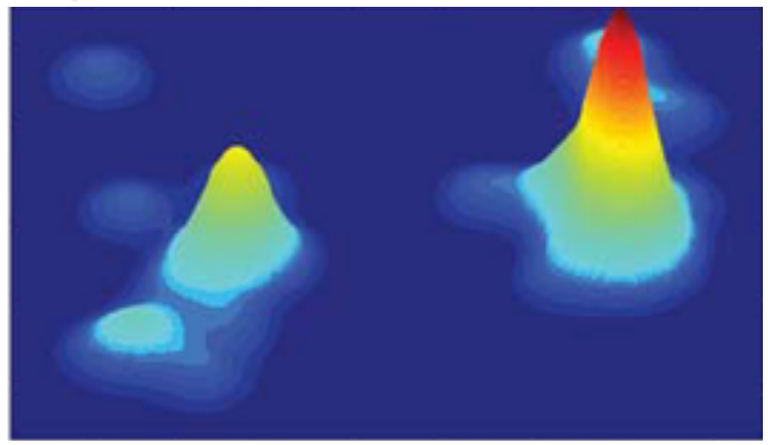
資訊工程學系
Department of Computer Science

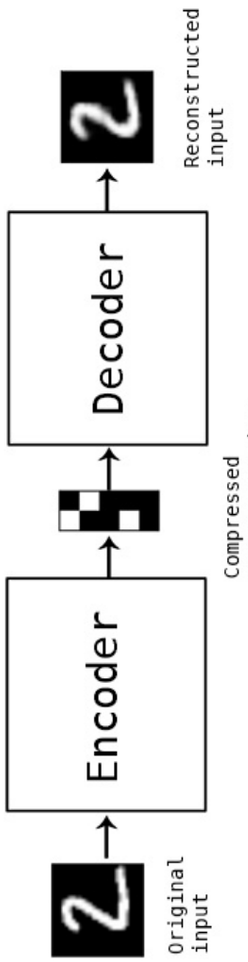
Here Comes Variational Autoencoder (VAE)

D. Kingma

Now is a "*distribution*", we can assume it to be a distribution easy to sample from, e.g. Gaussian



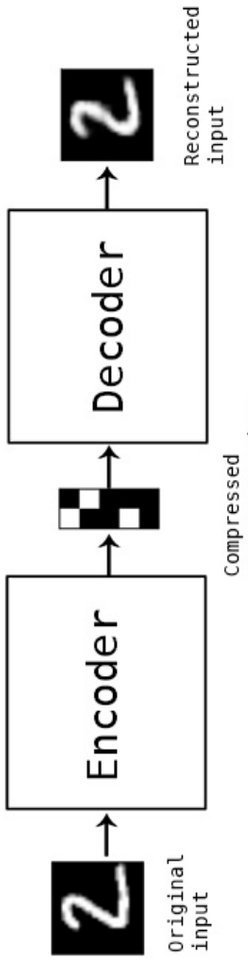


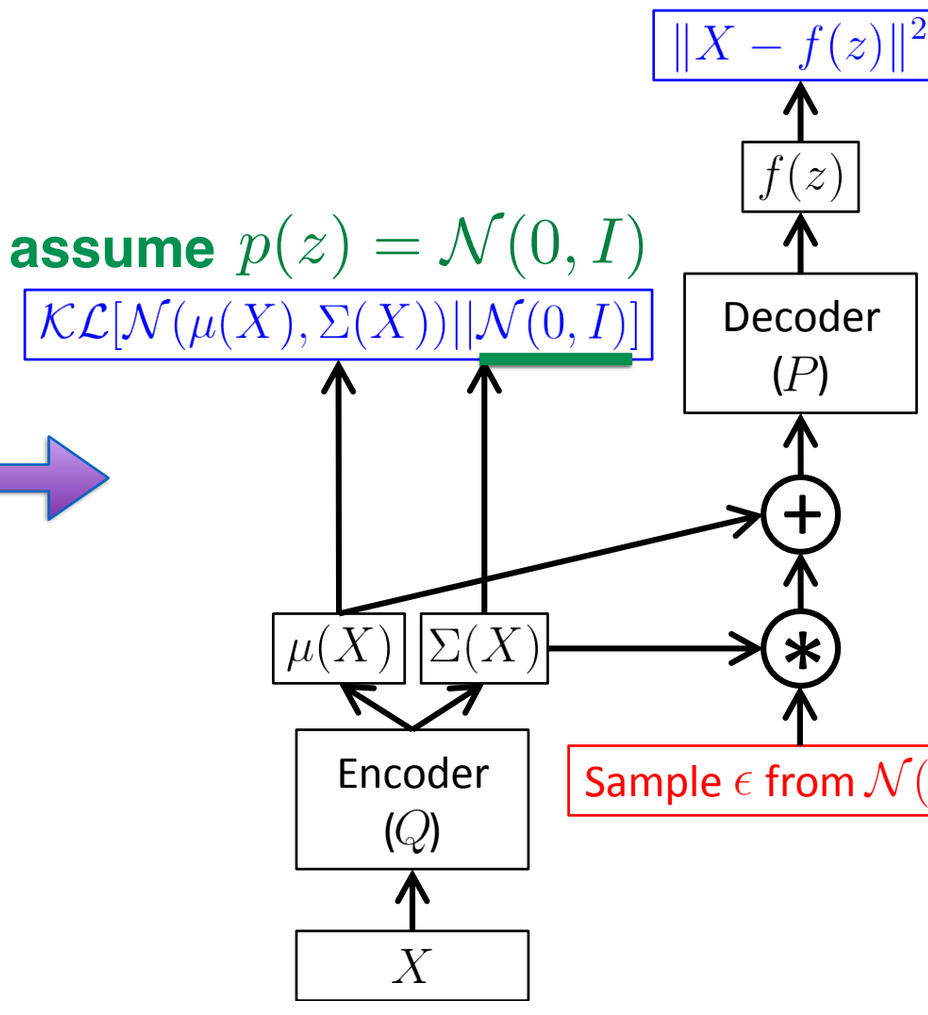


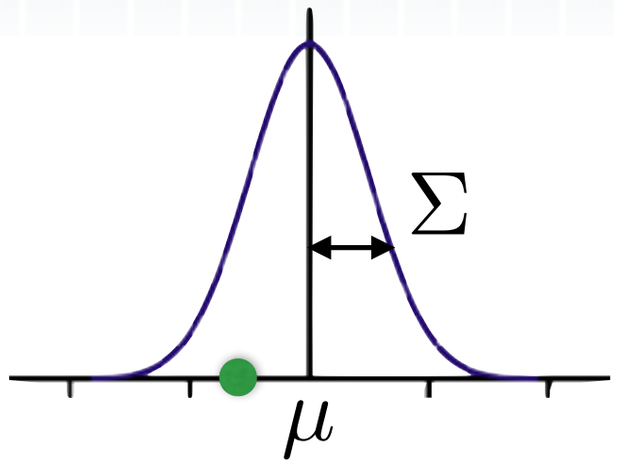
資訊工程學系
Department of Computer Science

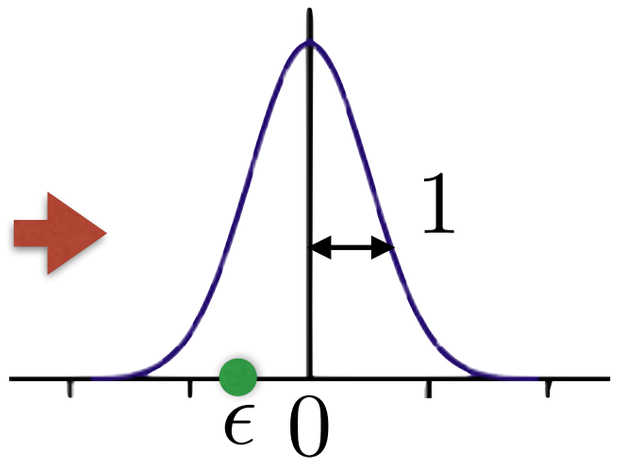
Here Comes Variational Autoencoder (VAE)

D. Kingma









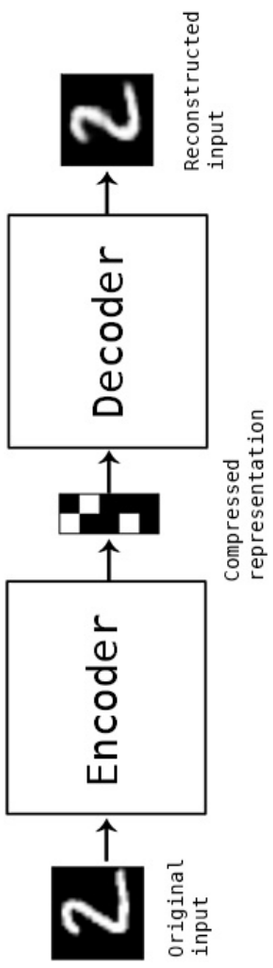
$$\mu + \epsilon \cdot \Sigma$$

reparameterization trick to enable end-to-end optimization

[Ref: C. Doersch, Tutorial on Variational Autoencoder, 2016]

Here Comes Variational Autoencoder (VAE)

D. Kingma

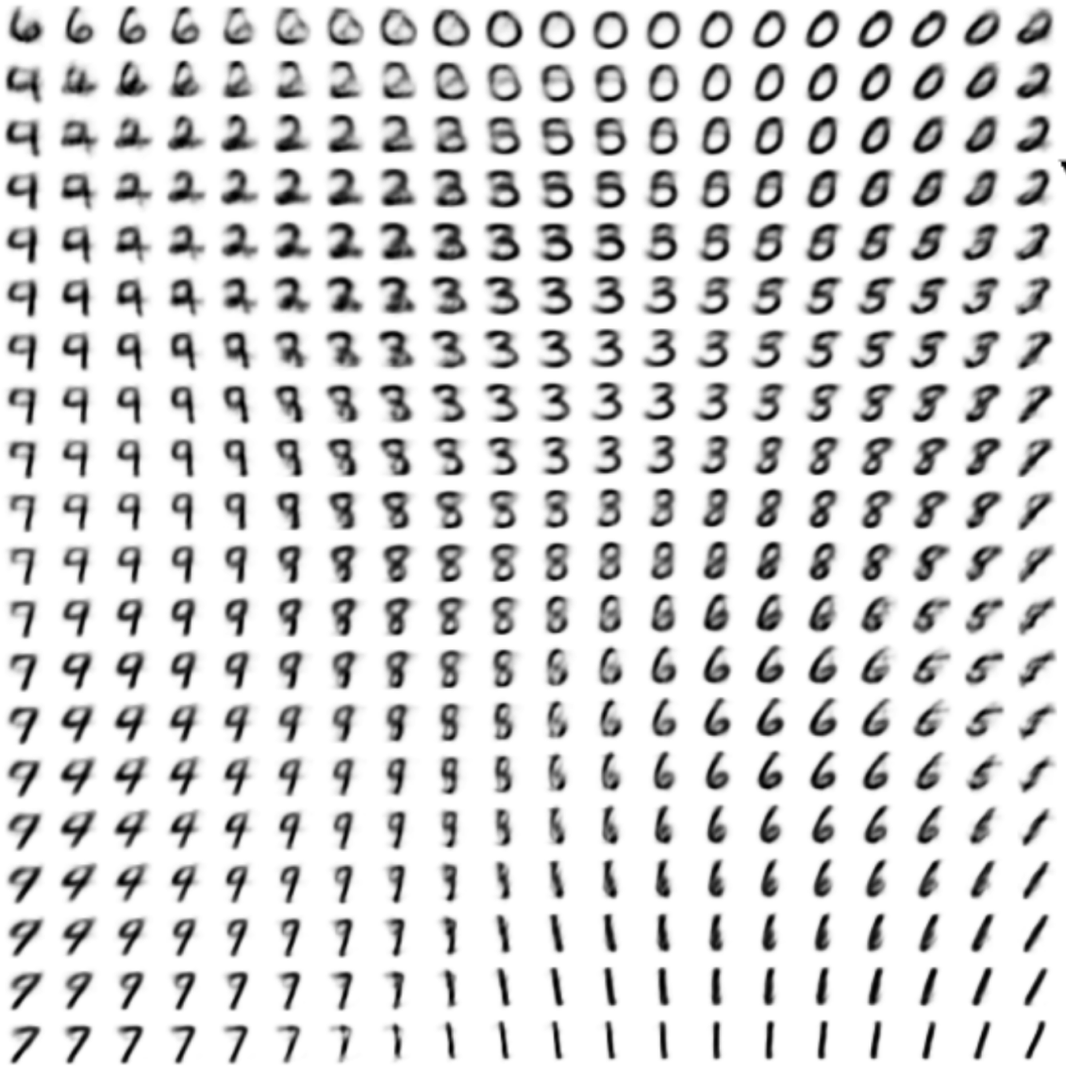


compactly capture the underlying factors
that explain most of the variation in the data

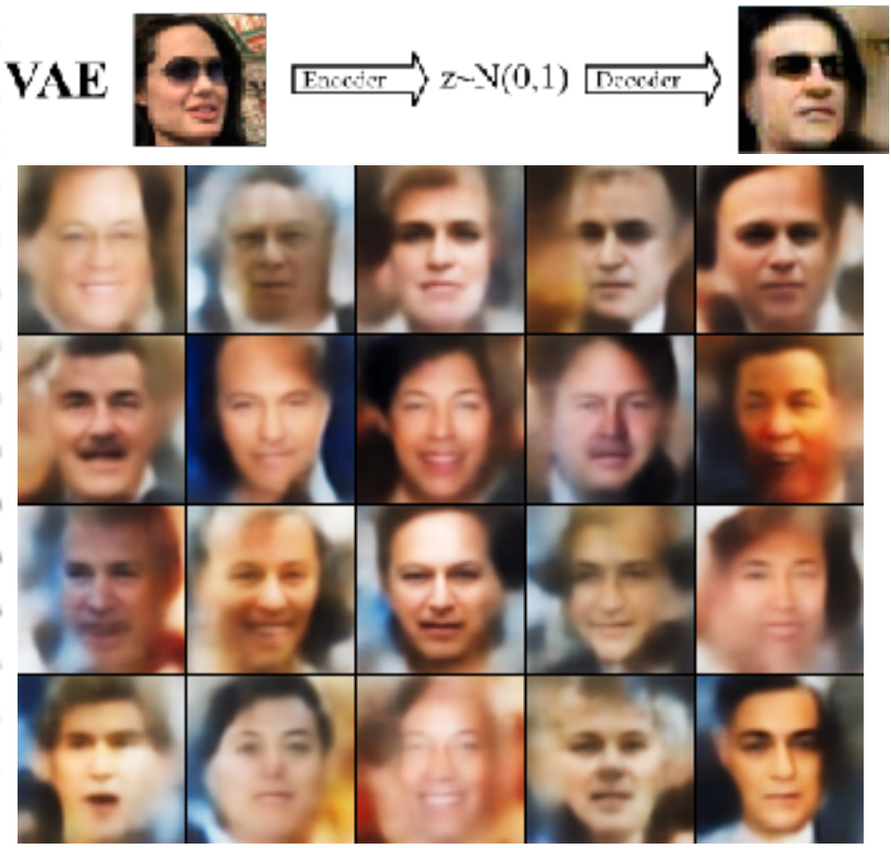
Correspondingly, the actual task to generate content can be considered as a *measure* of the power of the internal representation

Example Results for Variational Autoencoder (VAE)

(a) Learned Frey Face manifold



(b) Learned MNIST manifold



[Ref: Kingma et al., Auto-Encoding Variational Bayes, 2013]
[Ref: http://torch.ch/blog/2015/11/13/gan.html]

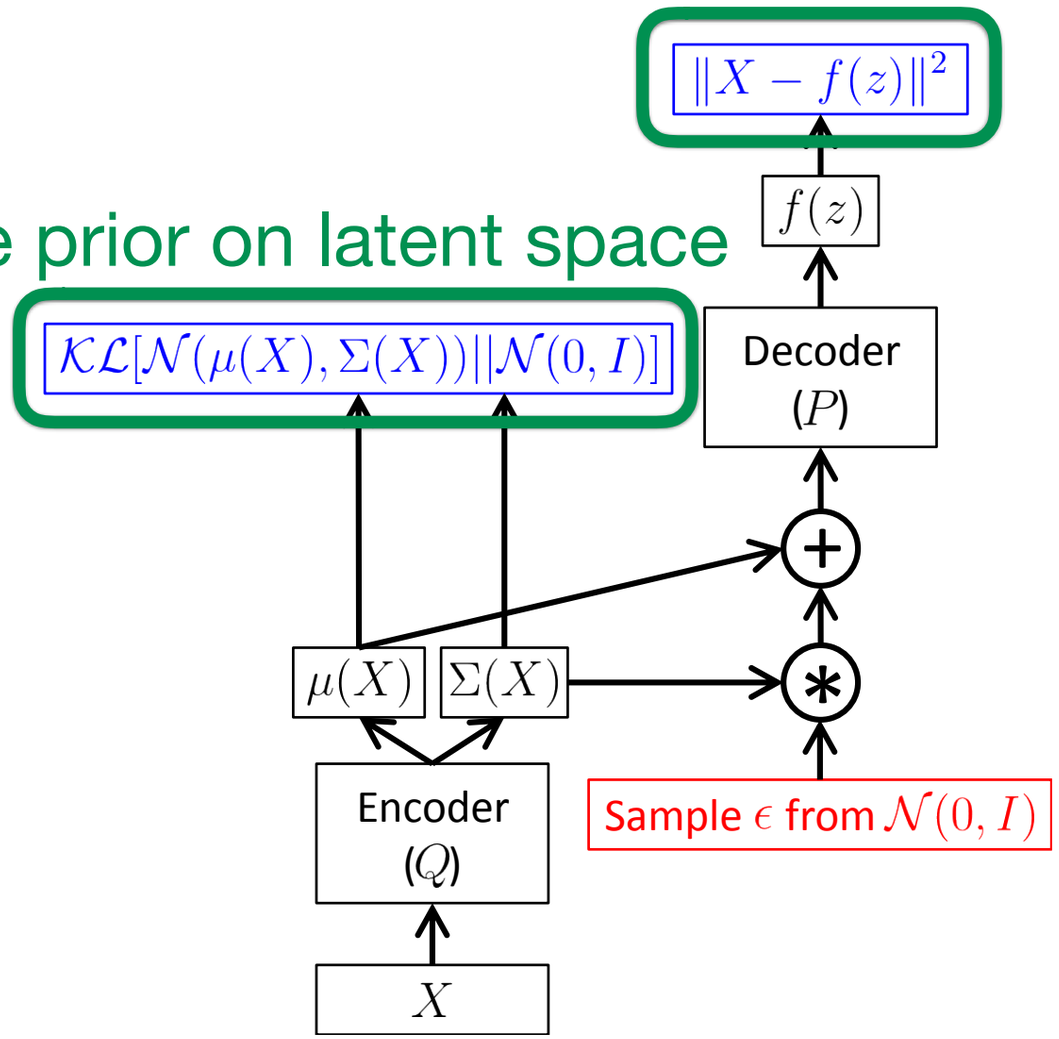
Outlines

- Discriminative versus Generative Models
- Going Into Deep Generative Models
- From Autoencoder to Variational Autoencoder (VAE)
- **From VAE to Generative Adversarial Network (GAN)**
- Various Applications
- Understanding the latent space: disentanglement

Different Modelling Perspective v.s. VAE

☞ reconstruction error

☞ impose prior on latent space



Yoshua Bengio [Quora]

- injected noise
- imperfect reconstruction

☞ more blurred output

Different Modelling Perspective v.s. VAE

☞ reconstruction error

☞ impose prior on latent space

☞impose adversarial loss on **data distribution**

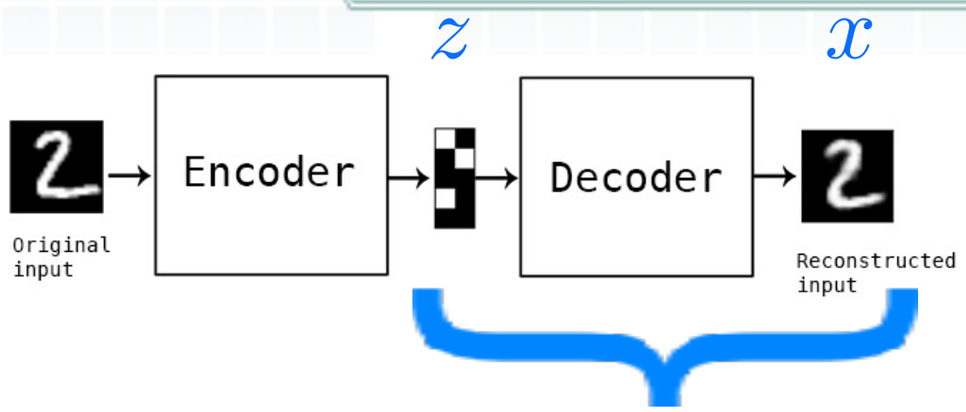
☞generative adversarial network [Goodfellow et al., 2014]

bossing

I. Goodfellow

Generative Adversarial Network (GAN)

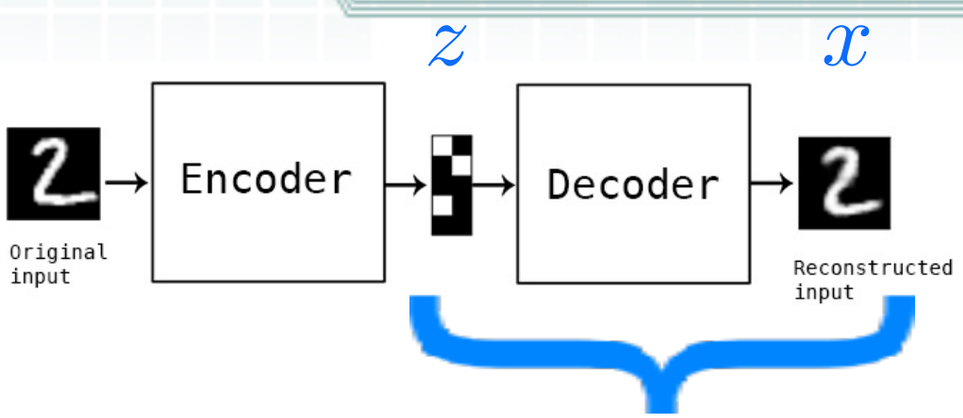
I. Goodfellow



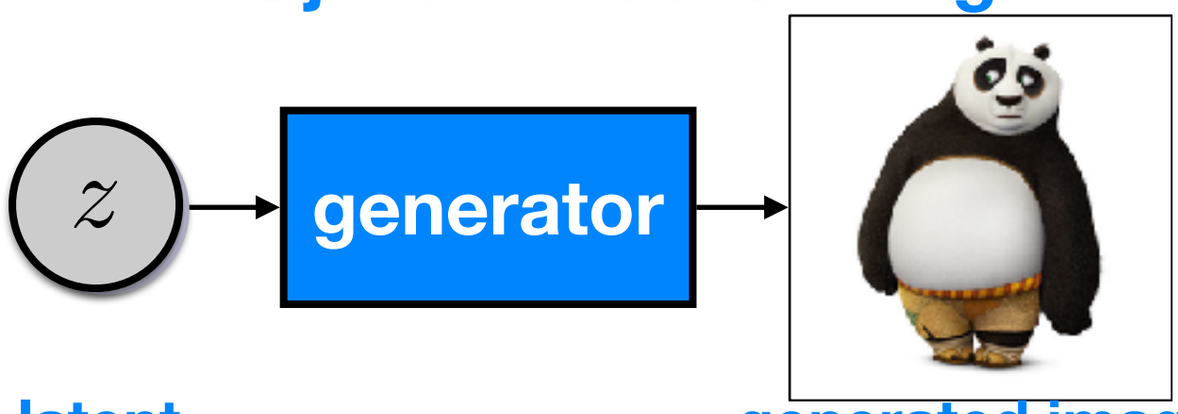
I just want to learn generator!

Generative Adversarial Network (GAN)

I. Goodfellow



I just want to learn generator!



latent code

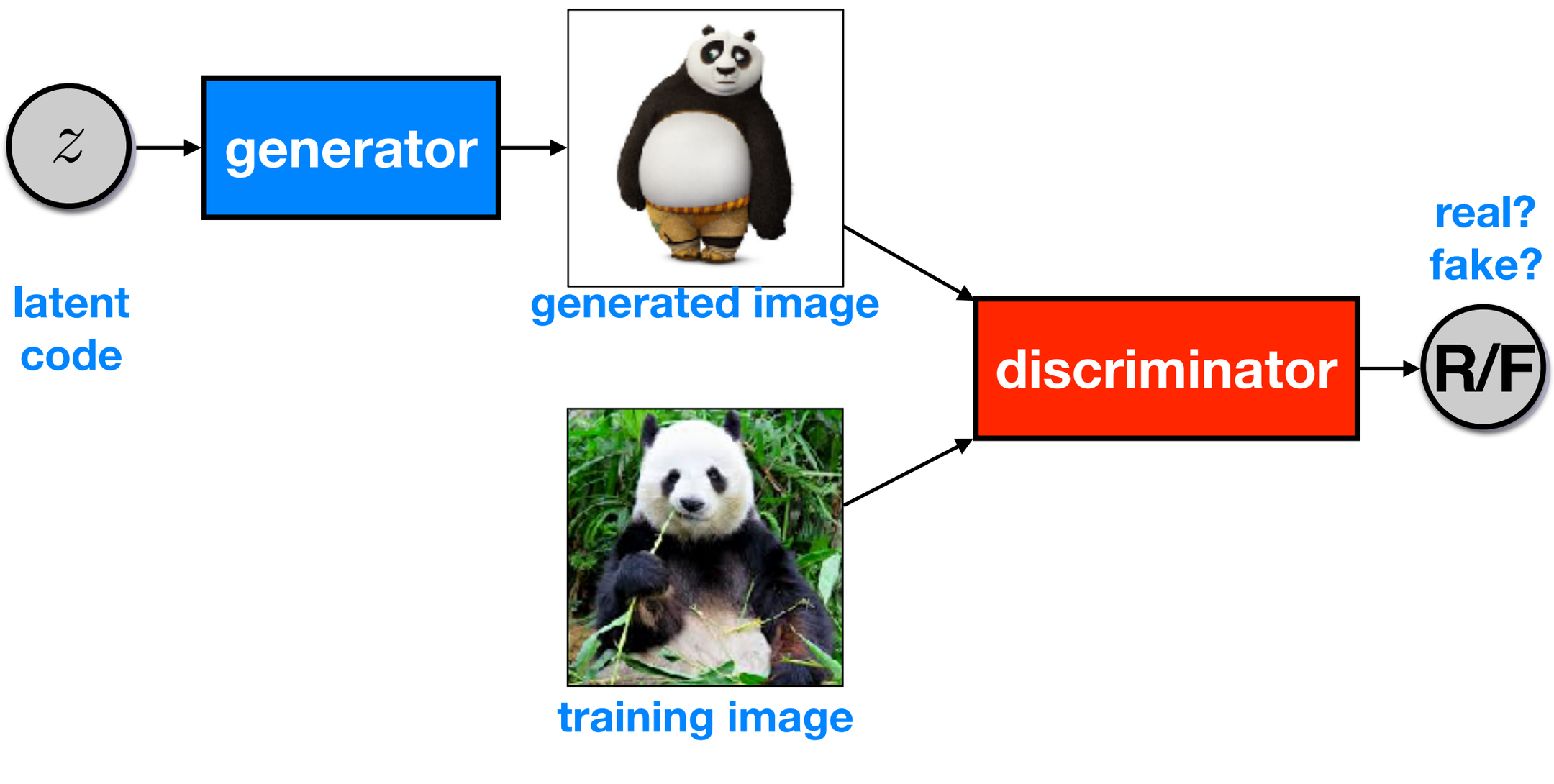
generated image

might look like a fake image,
how to get it more realistic?

I. Goodfellow

Generative Adversarial Network (GAN)

☞impose **adversarial loss** on **data distribution**



資訊工程學系
Department of Computer Science

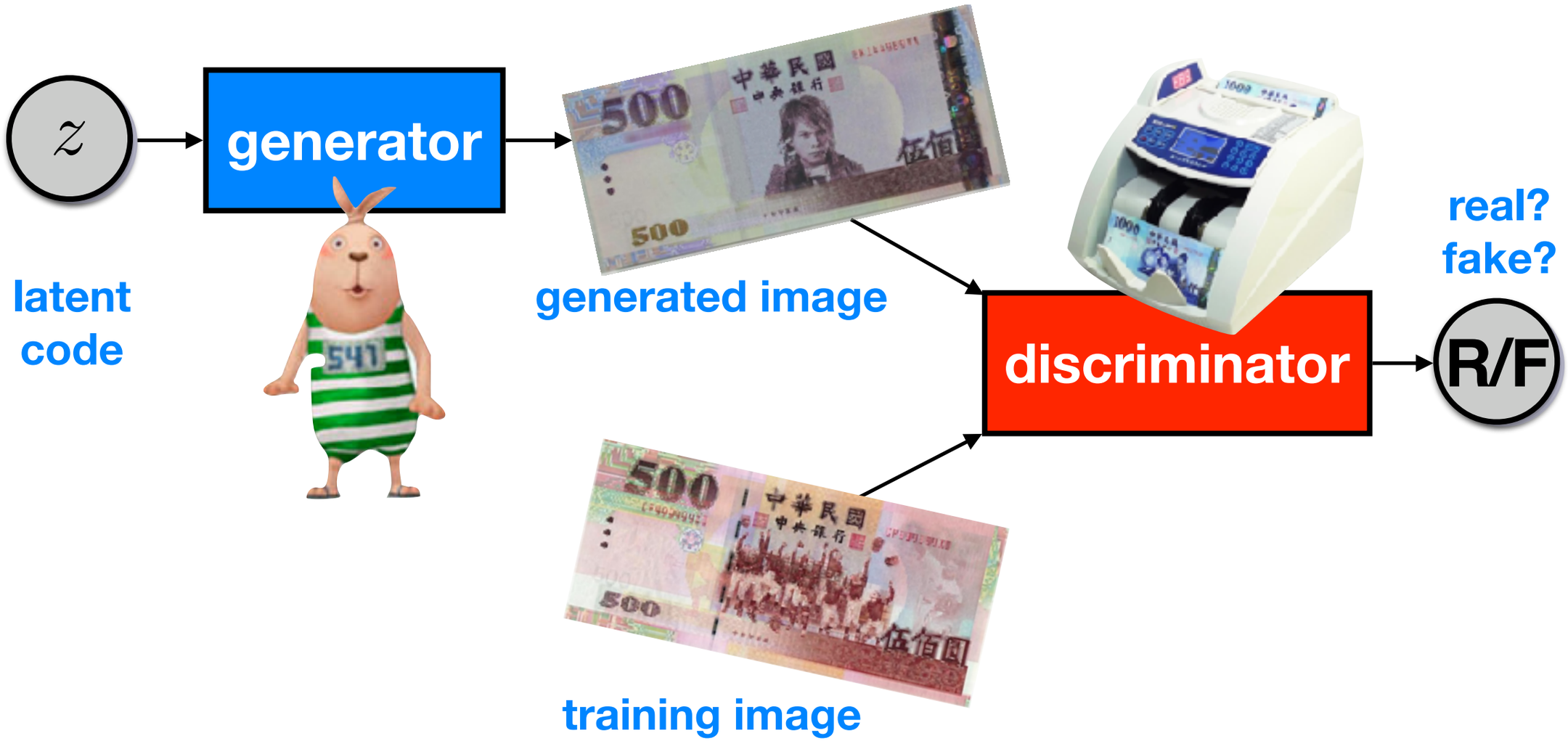
Generative Adversarial Network (GAN)

I. Goodfellow

☞impose **adversarial loss** on **data distribution**

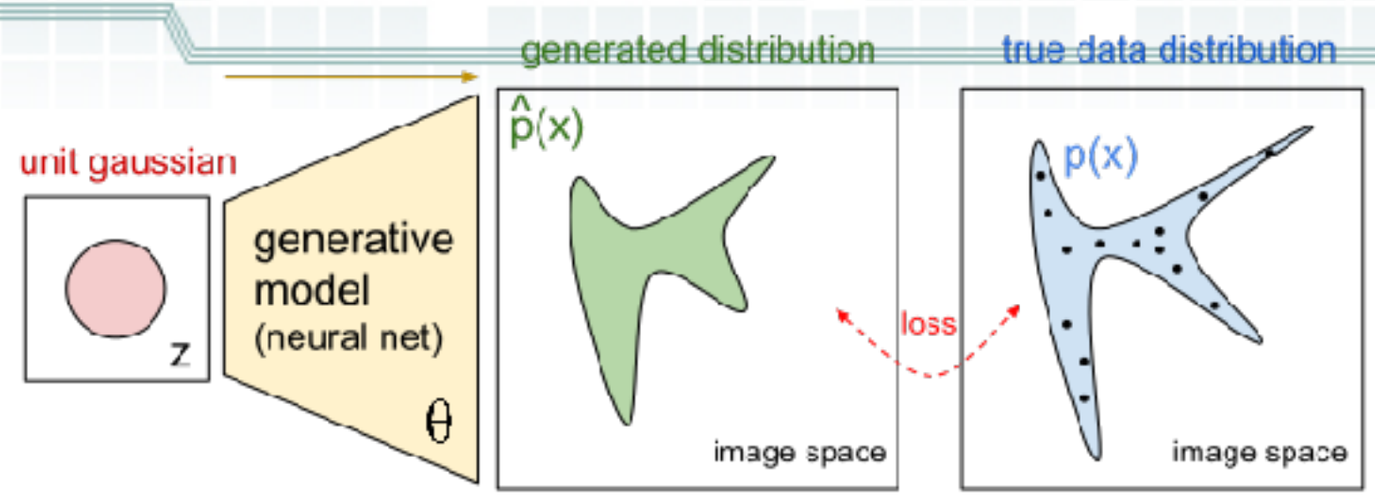
generator: try to generate more realistic images to cheat discriminator

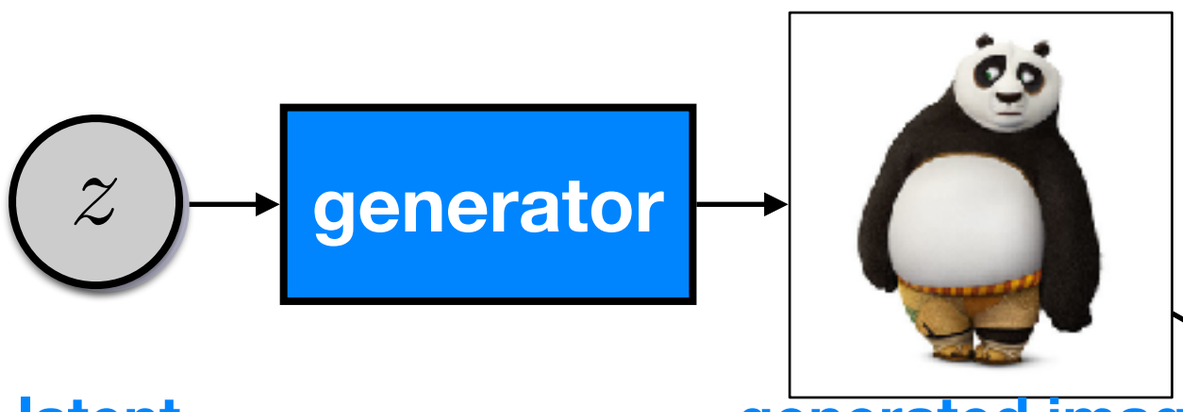
discriminator: try to distinguish whether the image is generated or real

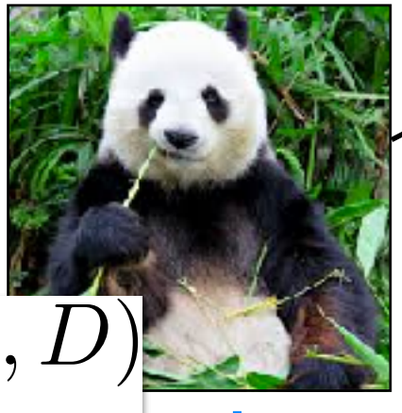


Generative Adversarial Network (GAN)

I. Goodfellow







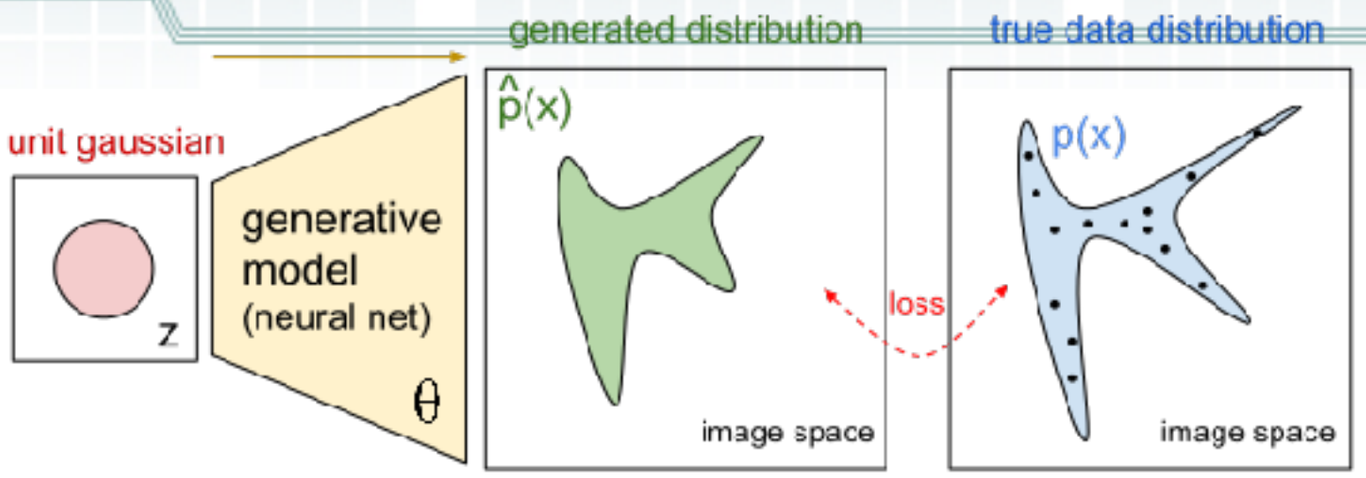
min-max game on a function *V(G, D)*

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

資訊工程學系
Department of Computer Science

Generative Adversarial Network (GAN)

I. Goodfellow



min-max game on a function *V(G, D)*

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

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

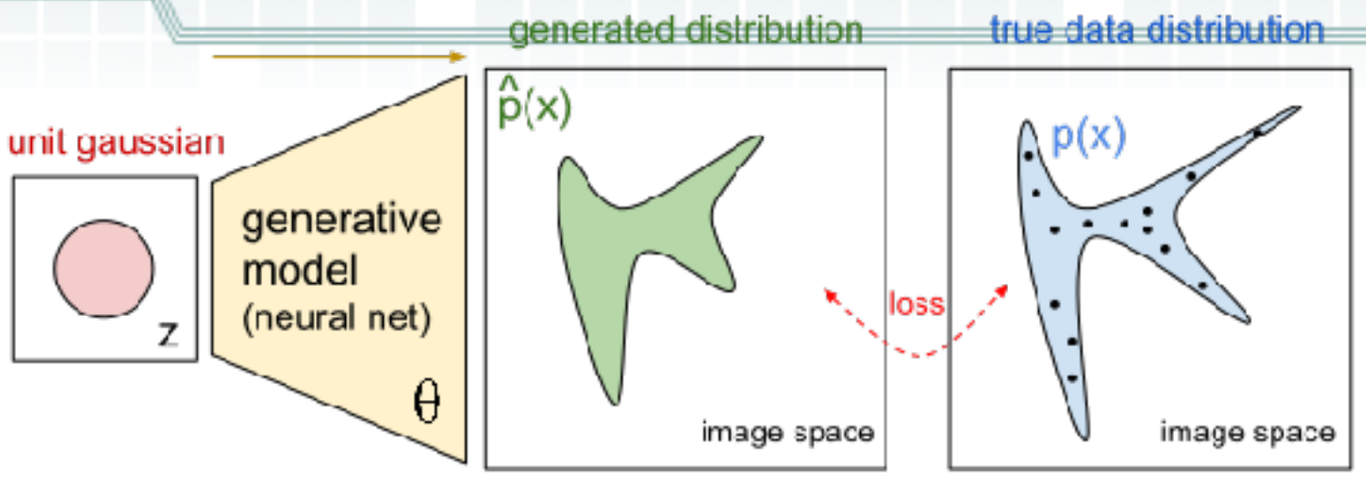
1. Given G, maximise V to find D: if x is from real data, maximise logD(x) ☞ D(x) = 1
(update parameters for D) if x is synthetic, maximise log(1-D(x)) ☞ D(x) = 0

2. Given D, minimise V to find G: x is synthetic, minimise log(1-D(x)) ☞ D(x) = 1
(update parameters for G)

資訊工程學系
Department of Computer Science

Generative Adversarial Network (GAN)

I. Goodfellow



min-max game on a function *V(G, D)*

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

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

optimal D: $D^*(x) = \frac{P_{data}(x)}{P_{data}(x) + P_G(x)}$

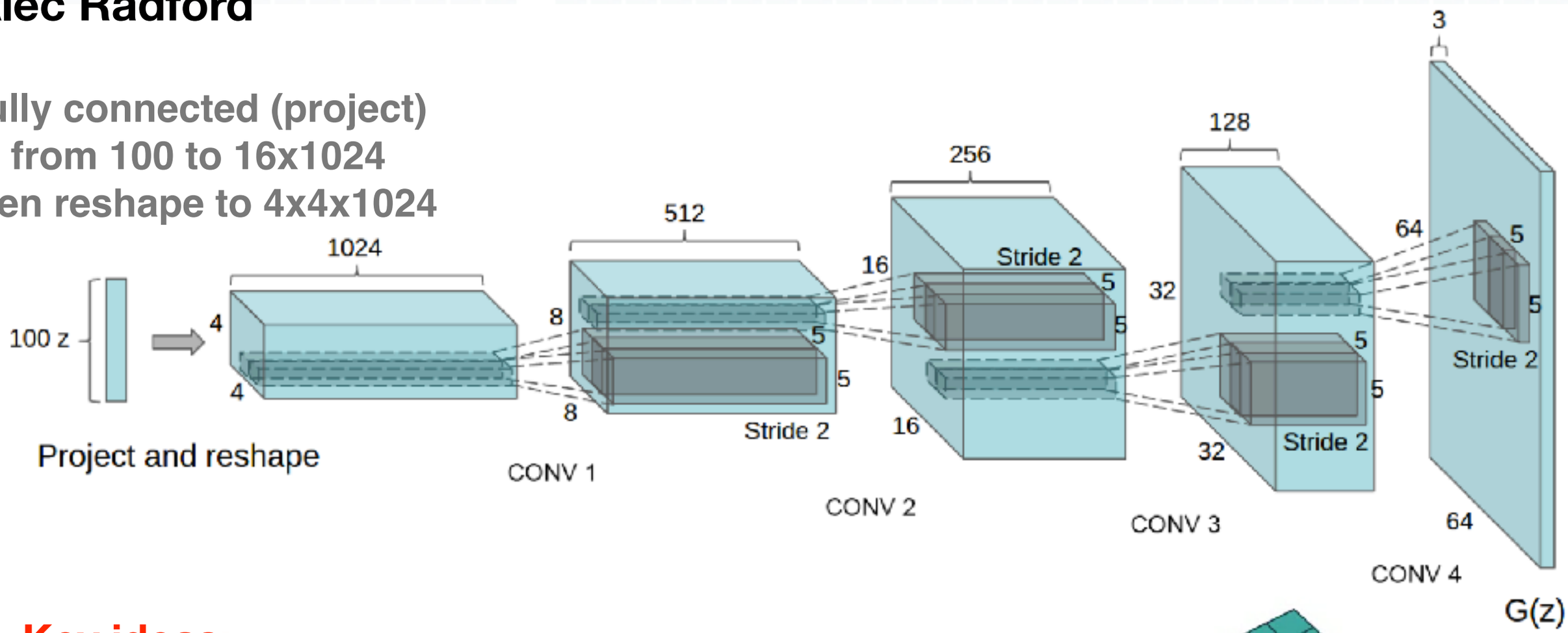
optimal G: $P_G(x) = P_{data}(x)$

資訊工程學系
Department of Computer Science

DC-GAN (Deep Convolutional GAN)

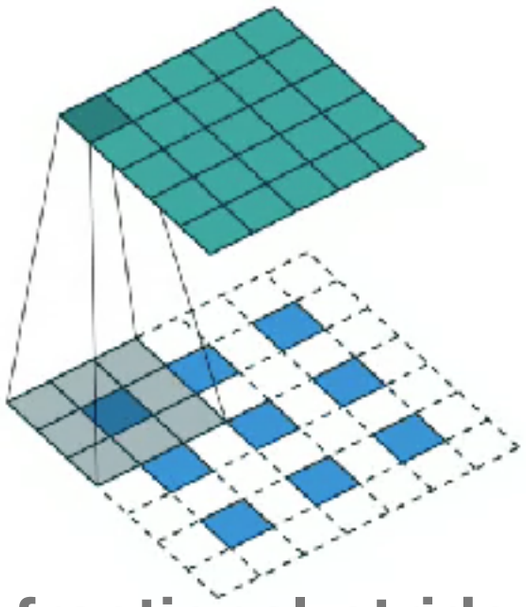
Alec Radford

fully connected (project)
from 100 to 16x1024
then reshape to 4x4x1024



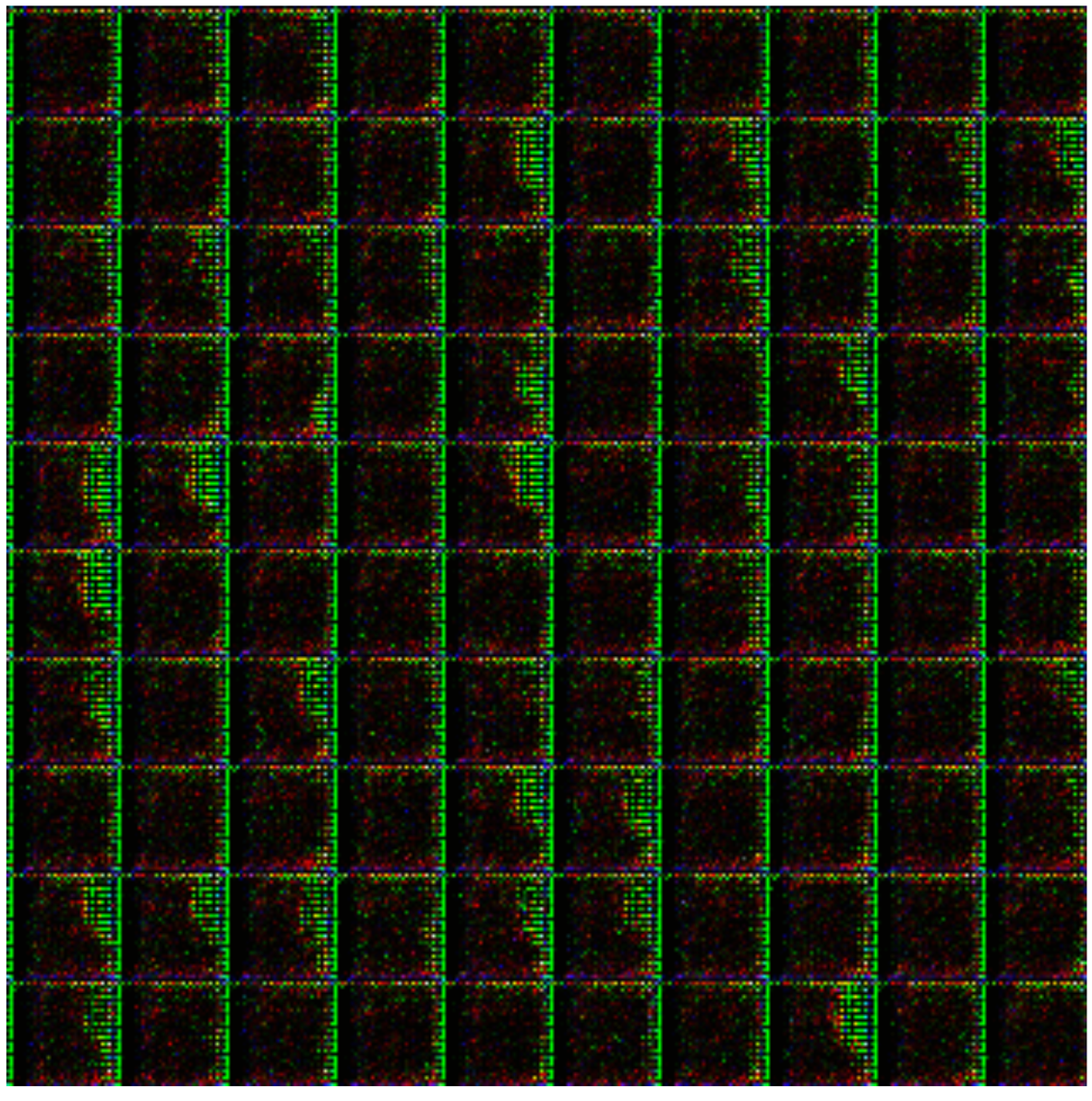
Key ideas:

- Replace FC hidden layers with convolutions
  - Generator: fractional-strided convolutions
- Use batch-normalization after each layer
- Insider Generator
  - Use ReLU for hidden layers
  - Use Tanh for the output layer



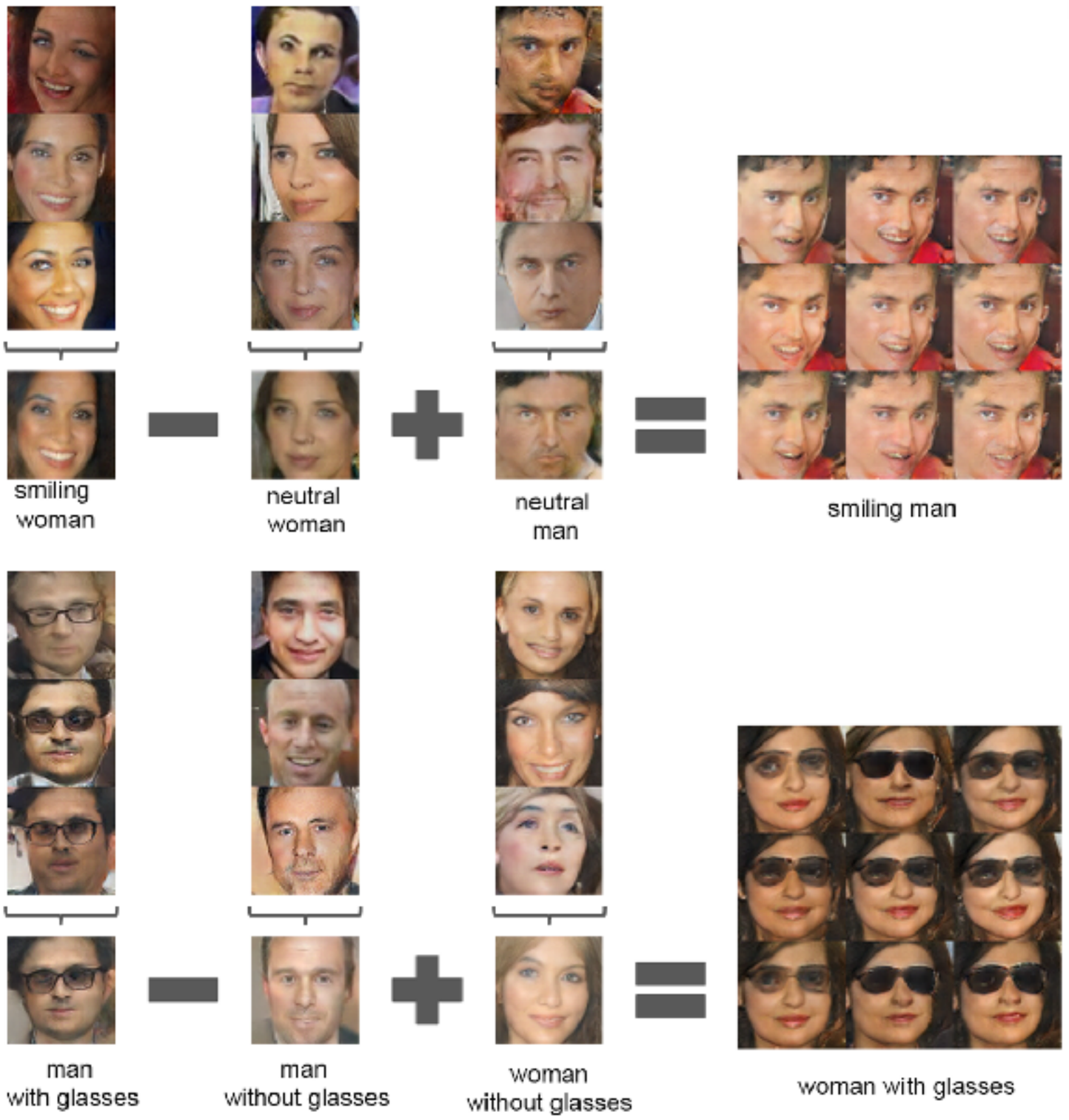
fractional-strided

Example Results for Adversarial Generative Model



[Ref:Radford et al., DC-GAN, 2015]
[Ref:https://github.com/carpedm20/DCGAN-tensorflow/]

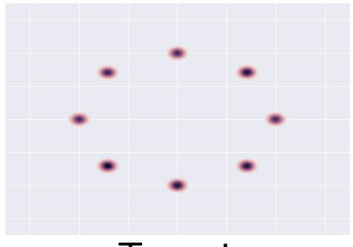
Example Results for Adversarial Generative Model



[Ref:Radford et al., DC-GAN, 2015]

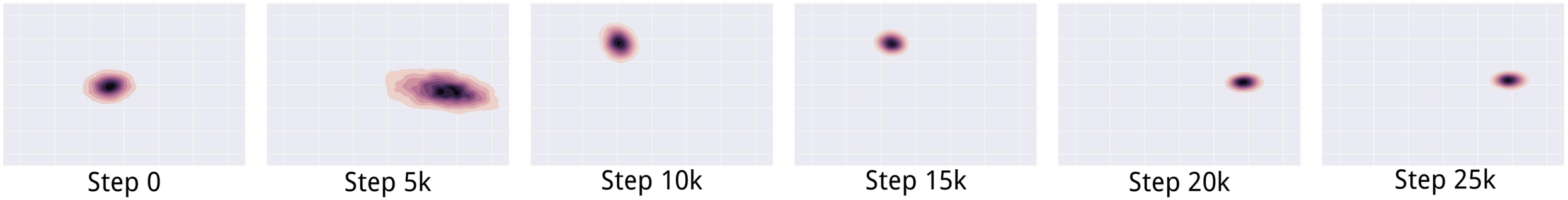
Mode Collapse in GANs

real data distribution



Target

data distribution learnt by generator

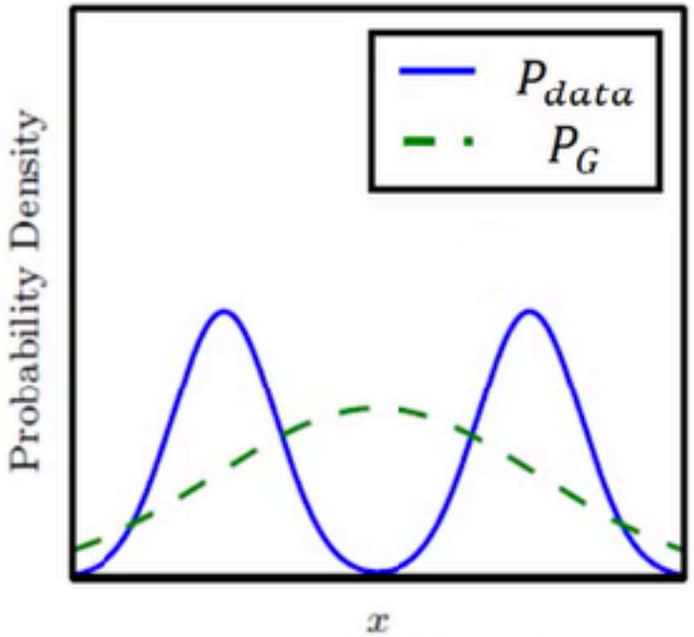


Generator fails to output diverse samples!

[Ref: Metz et al., Unrolled GAN, 2017]

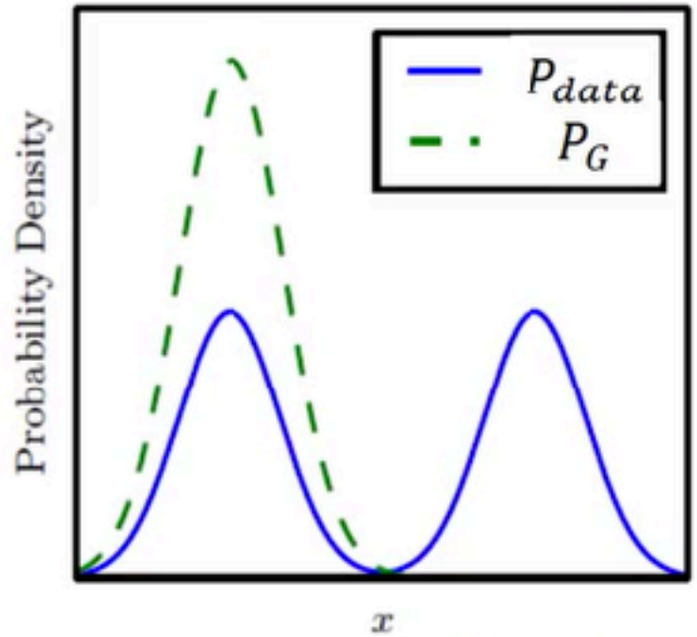
Mode Collapse in GANs

$$KL = \int P_{data} log \frac{P_{data}}{P_G} dx$$



Maximum likelihood
(minimize KL($P_{data}||P_G$))

Reverse $KL = \int P_G log \frac{P_G}{P_{data}} dx$



Minimize KL($P_G||P_{data}$)
(reverse KL)

The objective used in GAN produces distance measure of Jensen-Shannon divergence

$$\mathcal{JSD}(P||Q) = \frac{1}{2}\mathcal{KL}(P||M) + \frac{1}{2}\mathcal{KL}(Q||M)$$

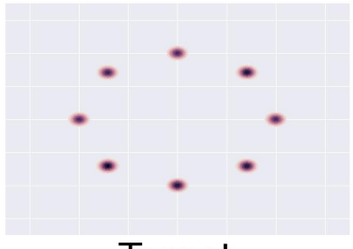
$$\text{where } M = \frac{1}{2}(P+Q)$$

whose property is close to reverse KL, thus mode collapse

credit: Ian Goodfellow

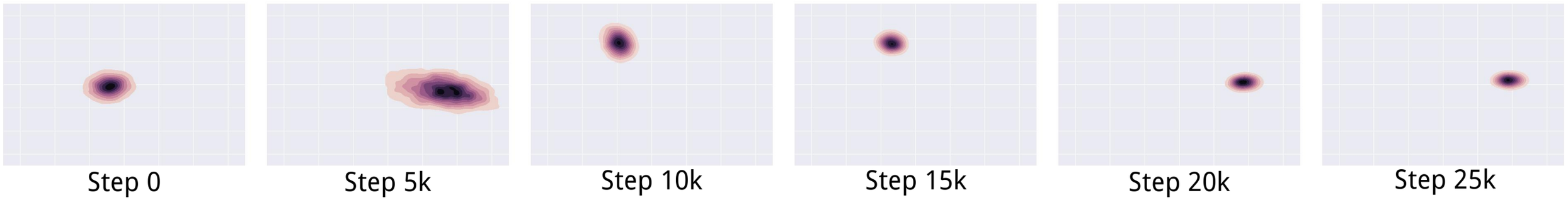
Mode Collapse in GANs

real data distribution



Target

data distribution learnt by generator



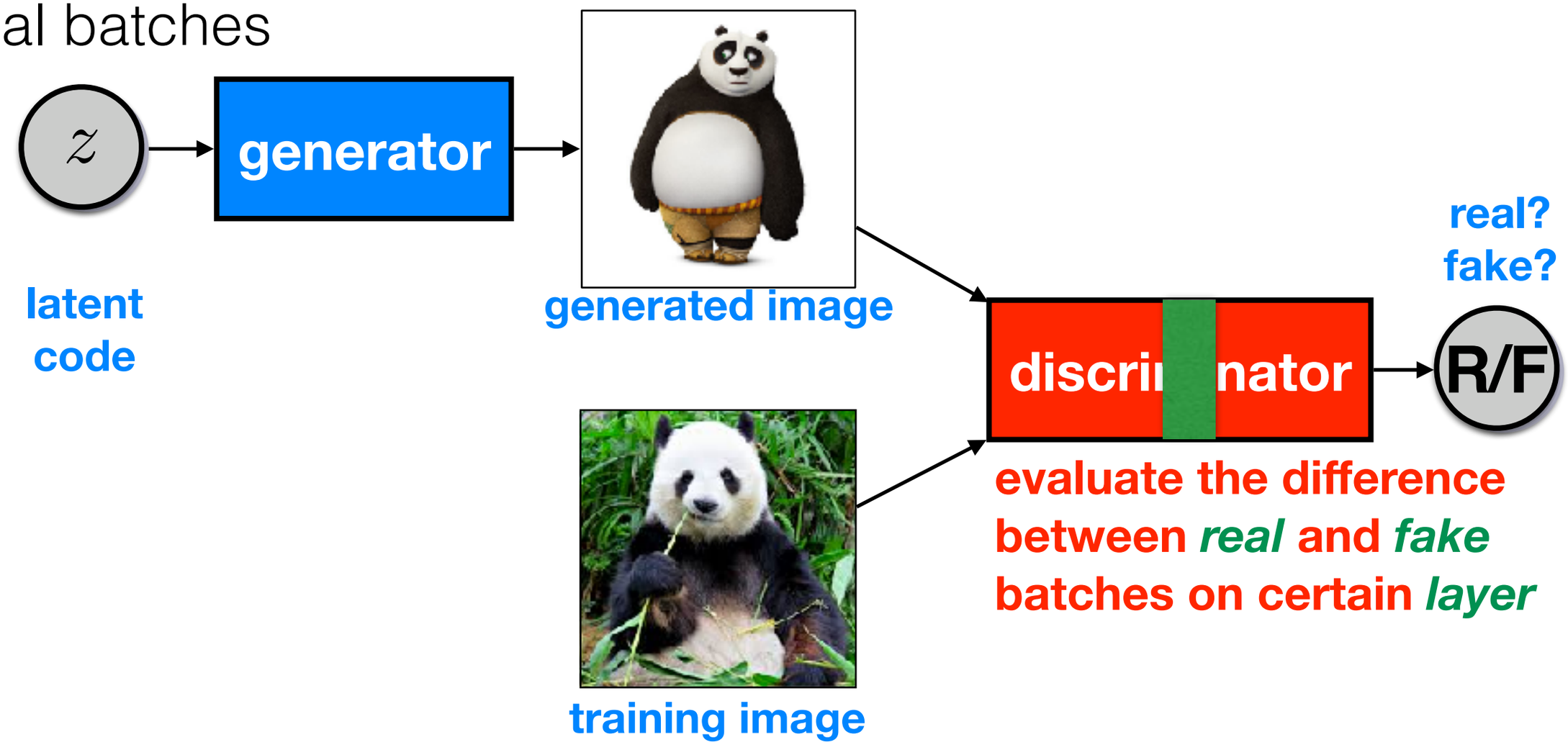
Few solutions:

1. **feature matching**
2. **minbatch discrimination**
3. **unrolled GAN**
4. **noisy labels**

[Ref: Salimans et al., Improved Techniques for training GANS, 2016]

Mode Collapse in GANs - Feature Matching

- Modify the cost function of generator, to include **diversity in real batches** into the generated batches
- Instead of fooling discriminator by the fake batches, now matching the statistic (e.g. L2 norm) of discriminator features (responses on a layer of discriminator) for fake batches to those of real batches



Mode Collapse in GANs - Minibatch Discrimination

- Let the discriminator **look at the entire batch** instead of single examples
  - If there is lack of diversity, it will mark the examples as fake

1. Extract **features** that capture **diversity** in the mini-batch
   ➡ for instance, L2 norm of the difference between all pairs from the batch

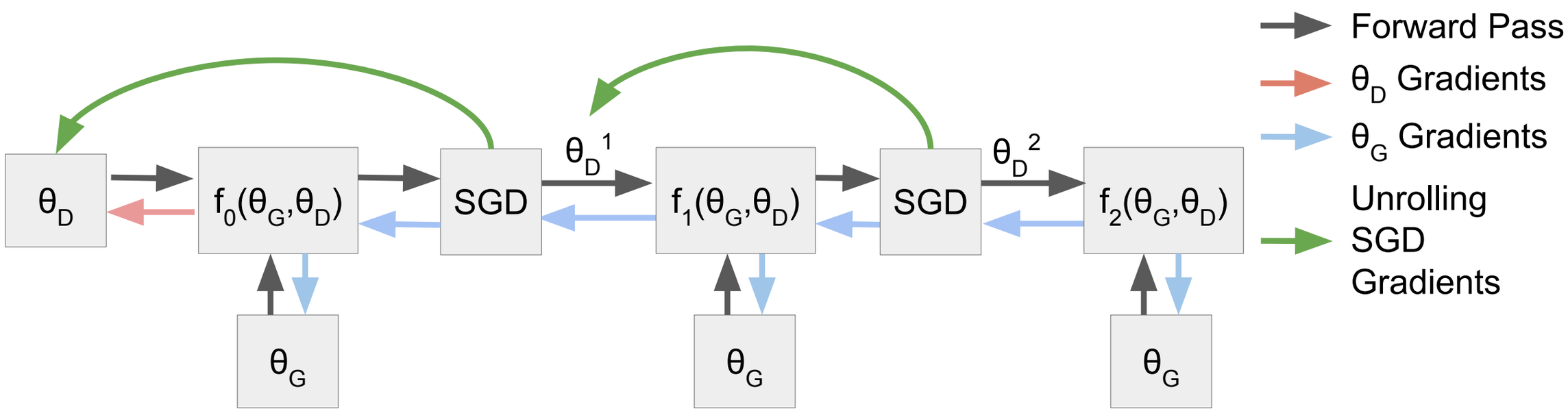
2. Feed those features to the discriminator along with the image

3. Feature values will **differ b/w diverse and non-diverse batches**
   ➡ thus, discriminator will rely on those features for classification

4. Force the generator to **match those features of diversity**, thus generate diverse batches

just like an additional feature attached to a batch

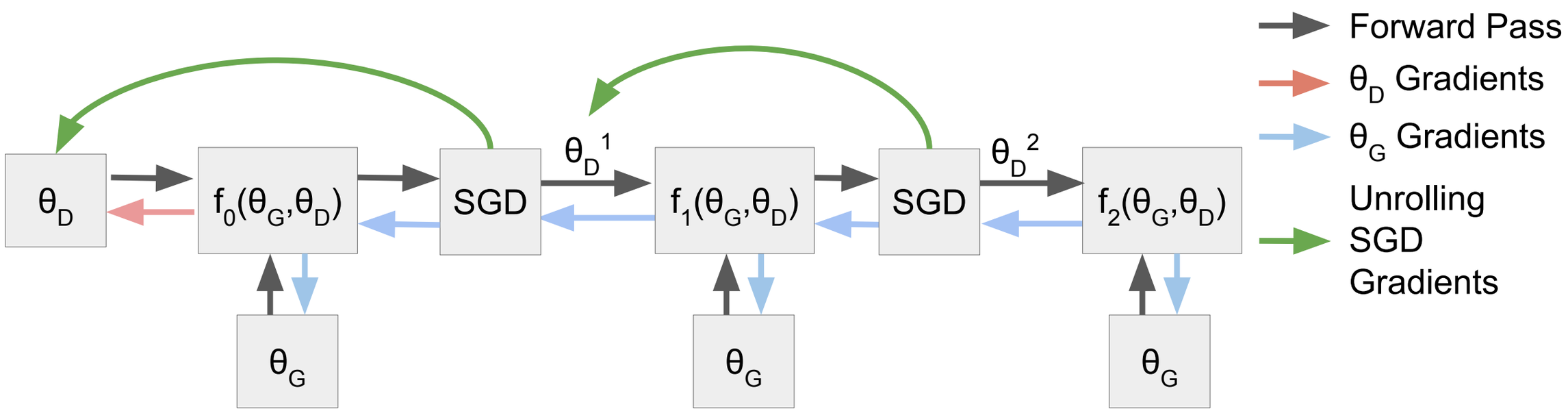
Mode Collapse in GANs - Unrolled GAN

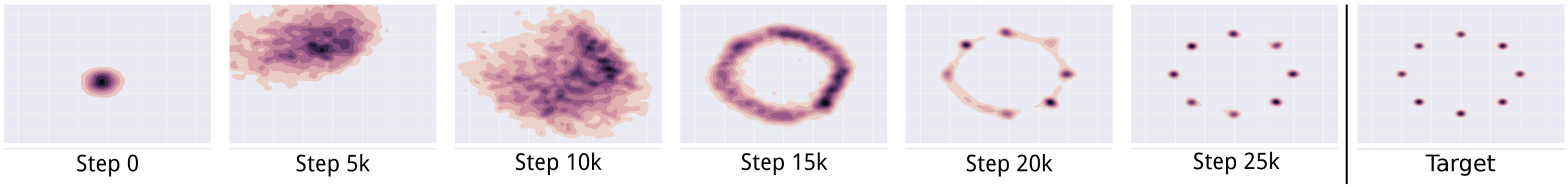


- Instead of learning generator to fool the current discriminator, it learns to maximally fool discriminator after peeking into the future
  - let discriminator updated few more runs and make generator fool the stronger discriminator

make generator great again! don't wanna have cat-mouse-game every time!

Mode Collapse in GANs - Unrolled GAN





Mode Collapse in GANs - Noisy Labels

- Label smoothing:
  - replace the label of real data with a random number b/w 0.7 and 1.2, and replace the label of generated data with 0.0 and 0.3 (for example), or keep label of generated data to be 0 (one-sided)
- Occasionally inverse the labels
  - inversely label real data by 0, and generated data by 1
  - fool the discriminator a little bit more

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

$$\nabla_{\theta_G} V(G,D) = \nabla_{\theta_G} E_{z\sim Q(z)}\left[log(1-D(G(z)))\right]$$

$$\nabla_a log(1-\sigma(a)) = \frac{-\nabla_a \sigma(a)}{1-\sigma(a)} = \frac{-\sigma(a)(1-\sigma(a))}{1-\sigma(a)} = -\sigma(a) \Rightarrow -D(G(z))$$

which means, the gradient goes to 0 if D is confident, i.e. $D(G(z)) \to 0$

discriminator too strong, try to slow it down.

just like sometime we will tune the ratio between times of updating *D* and *G*

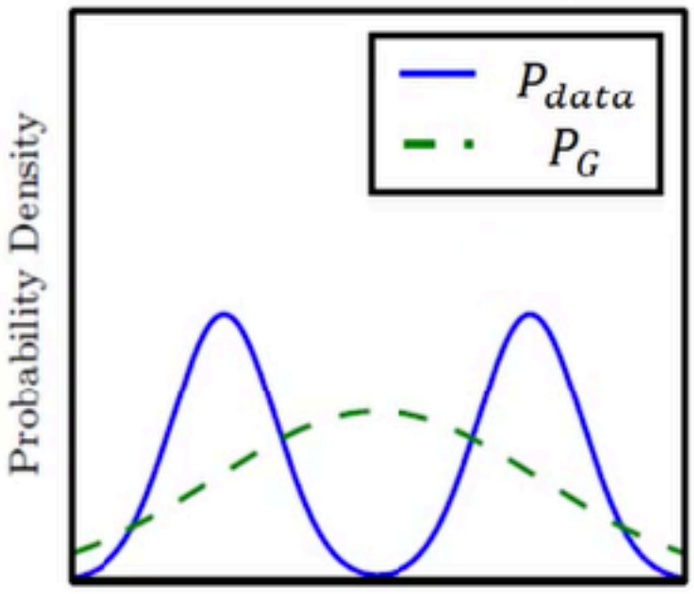
Different Variants of GAN with Different Objectives

- f-GAN
- Energy-Based GAN (EBGAN)
- Boundary Equilibrium GAN (BEGAN)
- Wasserstein GAN (WGAN)
- Least-Square GAN (LSGAN)
- Maximum Mean Discrepancy GAN (MMD-GAN)

How to measure the distance/difference between distributions?

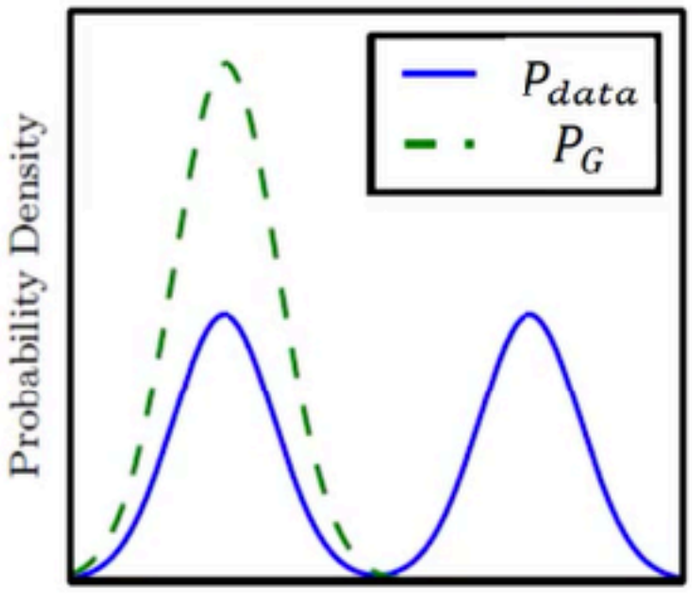
Different Variants of GAN with Different Objectives

$$KL = \int P_{data} log \frac{P_{data}}{P_G} dx$$



Maximum likelihood
(minimize KL($P_{data}||P_G$))

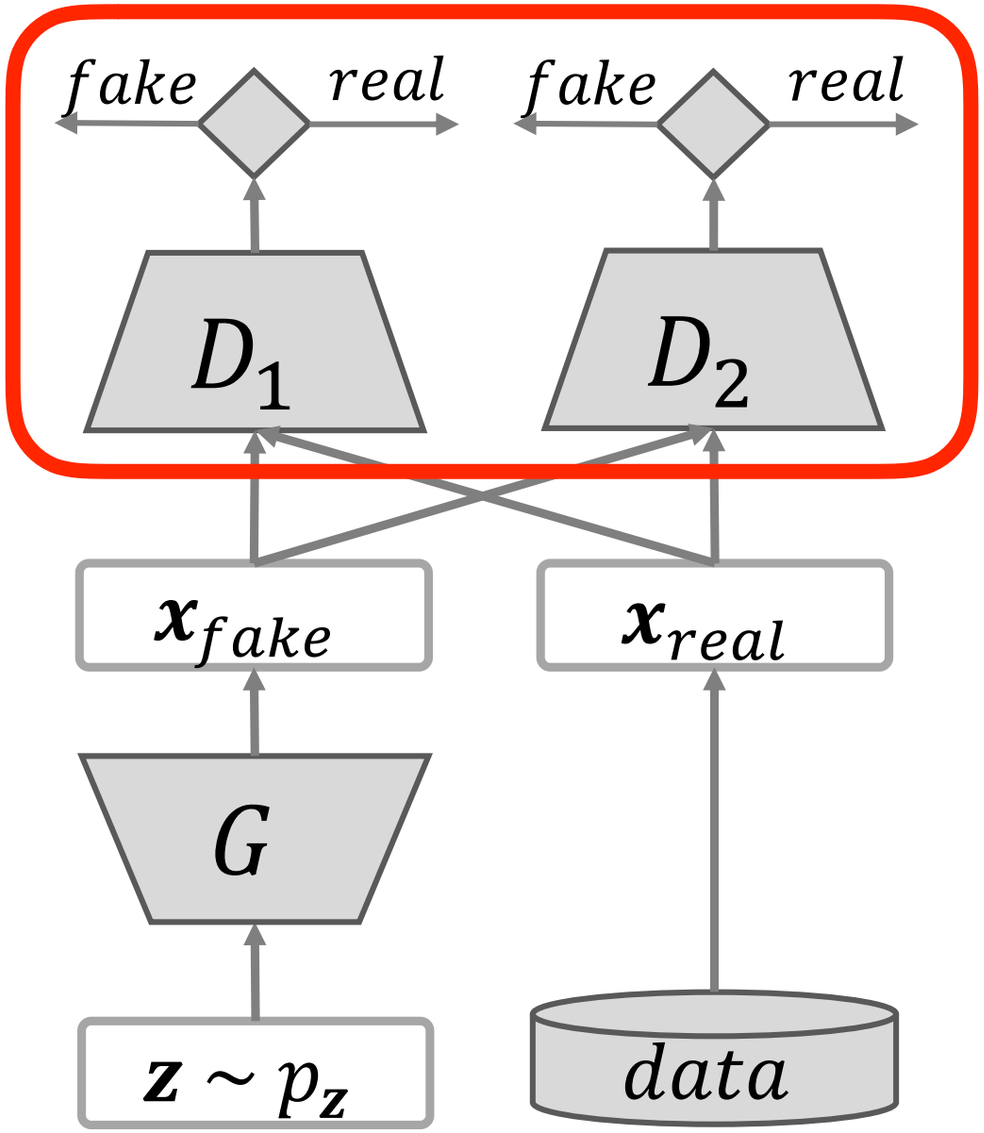
$$\text{Reverse } KL = \int P_G log \frac{P_G}{P_{data}} dx$$



Minimize KL($P_G||P_{data}$)
(reverse KL)

remember complement b/w *KL* and reverse *KL*?

has KL & reverse KL jointly!



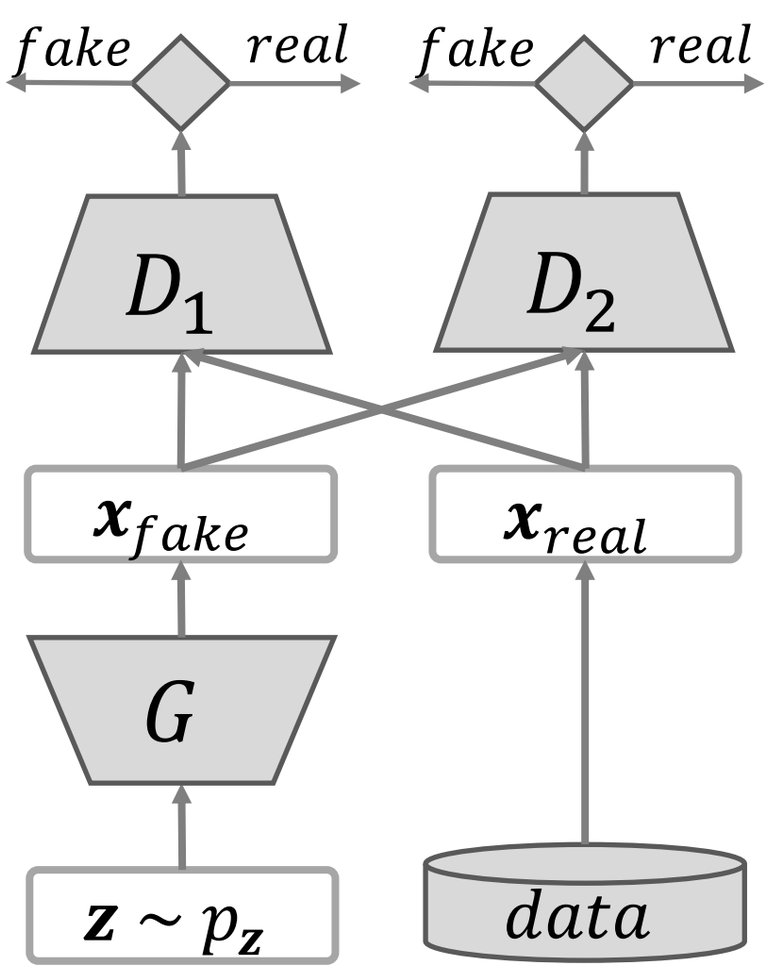
[Ref: Nguyen et al., D2GAN, **Sept. 2017**]

D2GAN

- Two discriminators (no weights sharing)
  - $\boldsymbol{D_1}$: $\boldsymbol{D_1}(\mathbf{x})$ rewards high score if $\mathbf{x}$ is from $\boldsymbol{P}_{\text{data}}$ (real), and gives low score if $\mathbf{x}$ is from $\boldsymbol{P}_G$ (generated)
  - $\boldsymbol{D_2}$: $\boldsymbol{D_2}(\mathbf{x})$ rewards low score if $\mathbf{x}$ is from $\boldsymbol{P}_{\text{data}}$ (real), and gives high score if $\mathbf{x}$ is from $\boldsymbol{P}_G$ (generated)
- Three-player minimax optimization game now:)

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

$$\min_G \max_{D_1,D_2} \mathcal{J}\left(G,D_1,D_2\right) = \alpha \times \mathbb{E}_{\mathbf{x}\sim P_{\text{data}}}\left[\log D_1\left(\mathbf{x}\right)\right] + \mathbb{E}_{\mathbf{z}\sim P_{\mathbf{z}}}\left[-D_1\left(G\left(\mathbf{z}\right)\right)\right] + \mathbb{E}_{\mathbf{x}\sim P_{\text{data}}}\left[-D_2\left(\mathbf{x}\right)\right] + \beta \times \mathbb{E}_{\mathbf{z}\sim P_{\mathbf{z}}}\left[\log D_2\left(G\left(\mathbf{z}\right)\right)\right]$$



D2GAN

$$\min_{G} \max_{D_1, D_2} \mathcal{J}(G, D_1, D_2) = \alpha \times \mathbb{E}_{\mathbf{x} \sim P_{\text{data}}} [\log D_1(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim P_{\mathbf{z}}} [-D_1(G(\mathbf{z}))]$$
$$+ \mathbb{E}_{\mathbf{x} \sim P_{\text{data}}} [-D_2(\mathbf{x})] + \beta \times \mathbb{E}_{\mathbf{z} \sim P_{\mathbf{z}}} [\log D_2(G(\mathbf{z}))]$$

$$\mathcal{J}(G, D_1, D_2) = \alpha \times \mathbb{E}_{\mathbf{x} \sim P_{\text{data}}} [\log D_1(\mathbf{x})] + \mathbb{E}_{\mathbf{x} \sim P_G} [-D_1(\mathbf{x})]$$
$$+ \mathbb{E}_{\mathbf{x} \sim P_{\text{data}}} [-D_2(\mathbf{x})] + \beta \times \mathbb{E}_{\mathbf{x} \sim P_G} [\log D_2(\mathbf{x})]$$

$$= \int_{\mathbf{x}} [\alpha p_{\text{data}}(\mathbf{x}) \log D_1(\mathbf{x}) - p_G D_1(\mathbf{x}) - p_{\text{data}}(\mathbf{x}) D_2(\mathbf{x}) + \beta p_G \log D_2(\mathbf{x})] \, d\mathbf{x}$$

partial derivatives

$$\frac{\alpha p_{\text{data}}(\mathbf{x})}{D_1} - p_G(\mathbf{x}) = 0 \quad \text{and} \quad \frac{\beta p_G(\mathbf{x})}{D_2} - p_{\text{data}}(\mathbf{x}) = 0$$

Proposition 1. *Given a fixed G, maximizing $\mathcal{J}(G, D_1, D_2)$ yields to the following closed-form optimal discriminators $D_1^\star, D_2^\star$:*

$$D_1^\star(\mathbf{x}) = \frac{\alpha p_{\text{data}}(\mathbf{x})}{p_G(\mathbf{x})} \quad \text{and} \quad D_2^\star(\mathbf{x}) = \frac{\beta p_G(\mathbf{x})}{p_{\text{data}}(\mathbf{x})}$$

D2GAN

$$\min_{G} \max_{D_1, D_2} \mathcal{J}(G, D_1, D_2) = \alpha \times \mathbb{E}_{\mathbf{x} \sim P_{\text{data}}} [\log D_1(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim P_{\mathbf{z}}} [-D_1(G(\mathbf{z}))]$$
$$+ \mathbb{E}_{\mathbf{x} \sim P_{\text{data}}} [-D_2(\mathbf{x})] + \beta \times \mathbb{E}_{\mathbf{z} \sim P_{\mathbf{z}}} [\log D_2(G(\mathbf{z}))]$$

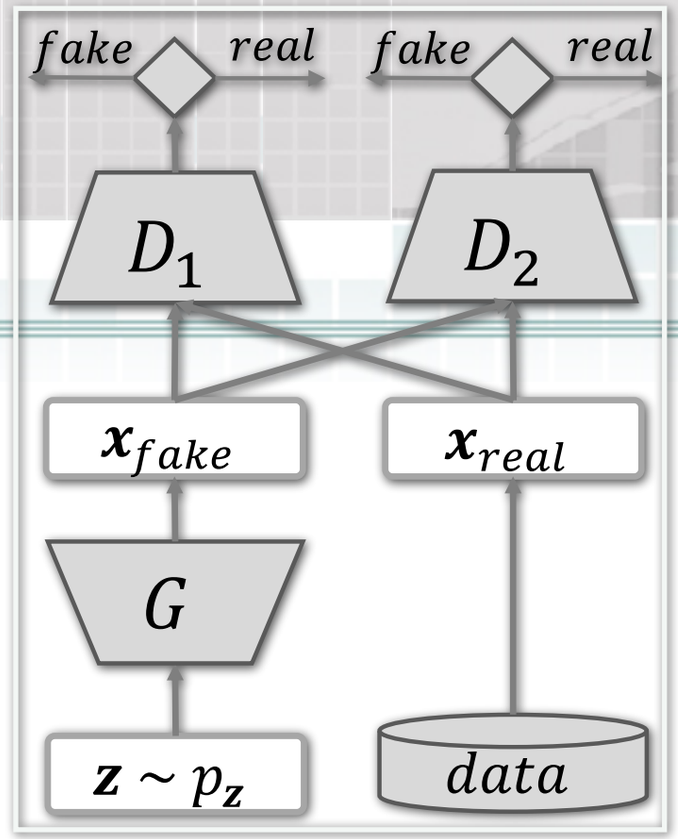
$$\mathcal{J}(G, D_1^\star, D_2^\star) = \alpha \times \mathbb{E}_{\mathbf{x} \sim P_{\text{data}}} \left[\log \alpha + \log \frac{p_{\text{data}}(\mathbf{x})}{p_G(\mathbf{x})} \right] - \alpha \int_{\mathbf{x}} p_G(\mathbf{x}) \frac{p_{\text{data}}(\mathbf{x})}{p_G(\mathbf{x})} \mathrm{d}\mathbf{x}$$
$$- \beta \int_{\mathbf{x}} p_{\text{data}} \frac{p_G(\mathbf{x})}{p_{\text{data}}(\mathbf{x})} \mathrm{d}\mathbf{x} + \beta \times \mathbb{E}_{\mathbf{x} \sim P_G} \left[\log \beta + \log \frac{p_G(\mathbf{x})}{p_{\text{data}}(\mathbf{x})} \right]$$
$$= \alpha(\log \alpha - 1) + \beta(\log \beta - 1) + \alpha D_{\text{KL}}(P_{\text{data}} \| P_G) + \beta D_{\text{KL}}(P_G \| P_{\text{data}})$$

complement b/w *KL* and reverse *KL*!

Proposition 1. *Given a fixed G, maximizing $\mathcal{J}(G, D_1, D_2)$ yields to the following closed-form optimal discriminators $D_1^\star, D_2^\star$:*

$$D_1^\star(\mathbf{x}) = \frac{\alpha p_{\text{data}}(\mathbf{x})}{p_G(\mathbf{x})} \quad \text{and} \quad D_2^\star(\mathbf{x}) = \frac{\beta p_G(\mathbf{x})}{p_{\text{data}}(\mathbf{x})}$$

D2GAN



simplest structure to deal w/ mode collapse in my perspective

$$\mathcal{J}(G, D_1^\star, D_2^\star) = \alpha \times \mathbb{E}_{\mathbf{x}\sim P_{\text{data}}}\left[\log\alpha + \log\frac{p_{\text{data}}(\mathbf{x})}{p_G(\mathbf{x})}\right] - \alpha\int_{\mathbf{x}} p_G(\mathbf{x})\frac{p_{\text{data}}(\mathbf{x})}{p_G(\mathbf{x})}\mathrm{d}\mathbf{x}$$
$$- \beta\int_{\mathbf{x}} p_{\text{data}}\frac{p_G(\mathbf{x})}{p_{\text{data}}(\mathbf{x})}\mathrm{d}\mathbf{x} + \beta \times \mathbb{E}_{\mathbf{x}\sim P_G}\left[\log\beta + \log\frac{p_G(\mathbf{x})}{p_{\text{data}}(\mathbf{x})}\right]$$
$$= \alpha(\log\alpha - 1) + \beta(\log\beta - 1) + \alpha D_{\text{KL}}(P_{\text{data}}\|P_G) + \beta D_{\text{KL}}(P_G\|P_{\text{data}})$$

complement b/w *KL* and reverse *KL*!

optimal $P_G(x) = P_{data}(x)$

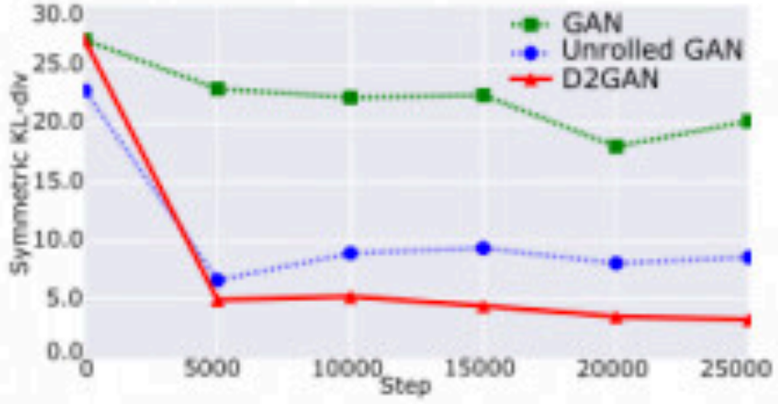
Proposition 1. *Given a fixed G, maximizing $\mathcal{J}(G, D_1, D_2)$ yields to the following closed-form optimal discriminators $D_1^\star, D_2^\star$:*

$$D_1^\star(\mathbf{x}) = \frac{\alpha p_{\text{data}}(\mathbf{x})}{p_G(\mathbf{x})} \quad \text{and} \quad D_2^\star(\mathbf{x}) = \frac{\beta p_G(\mathbf{x})}{p_{\text{data}}(\mathbf{x})}$$
$$\mathcal{J}(G^\star, D_1^\star, D_2^\star) = \alpha(\log\alpha - 1) + \beta(\log\beta - 1)$$
$$D_1^\star(\mathbf{x}) = \alpha \text{ and } D_2^\star(\mathbf{x}) = \beta, \forall\mathbf{x} \text{ at } p_{G^\star} = p_{data}$$

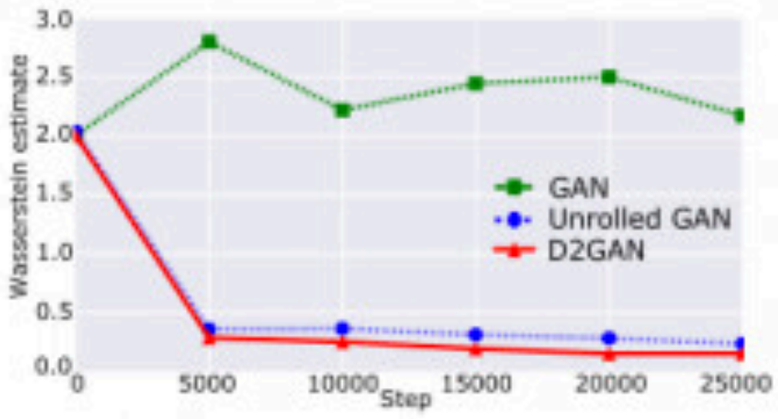
(a) CIFAR-10.

(b) STL-10.

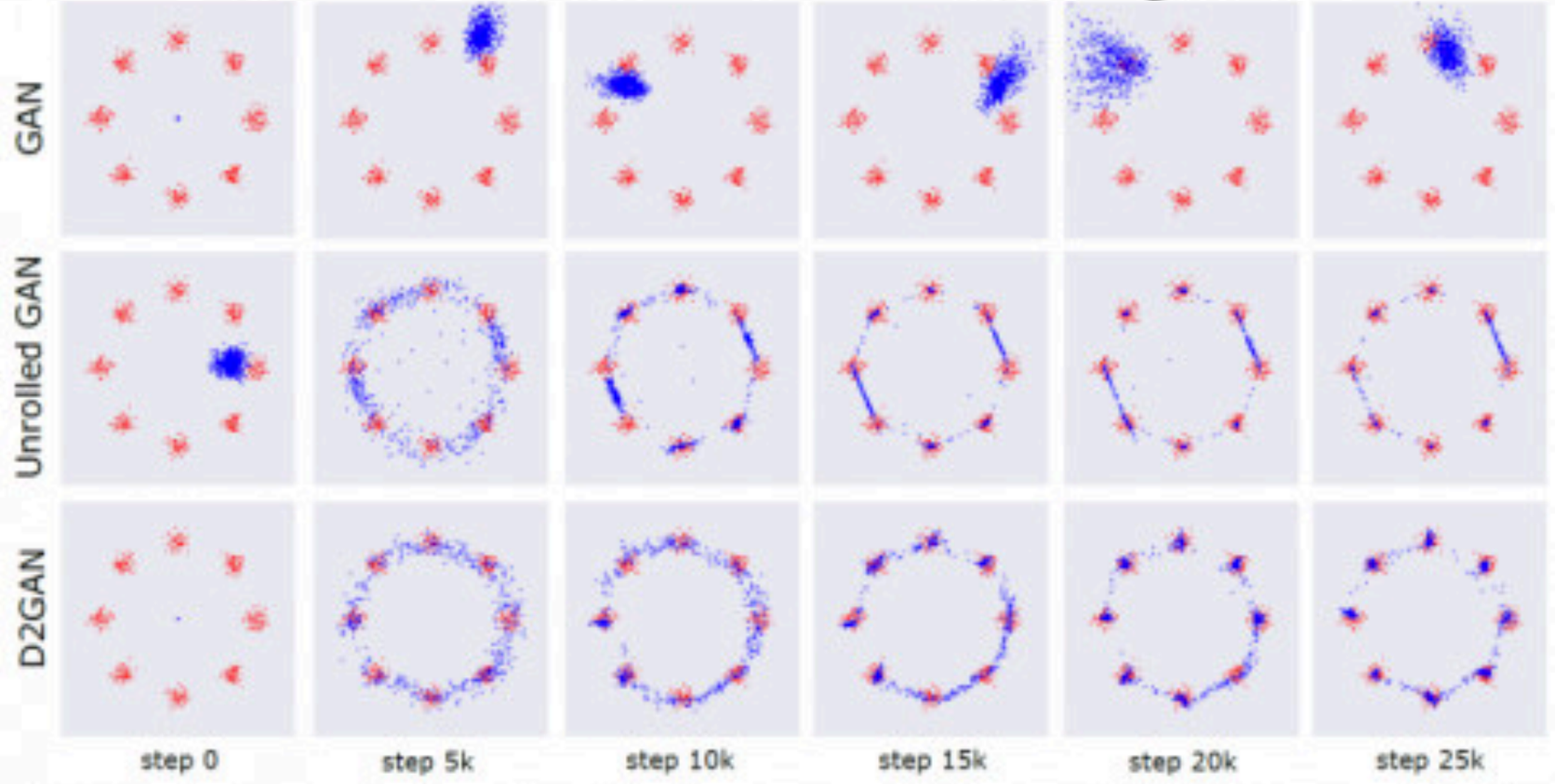
(c) ImageNet.



(a) Symmetric KL divergence.



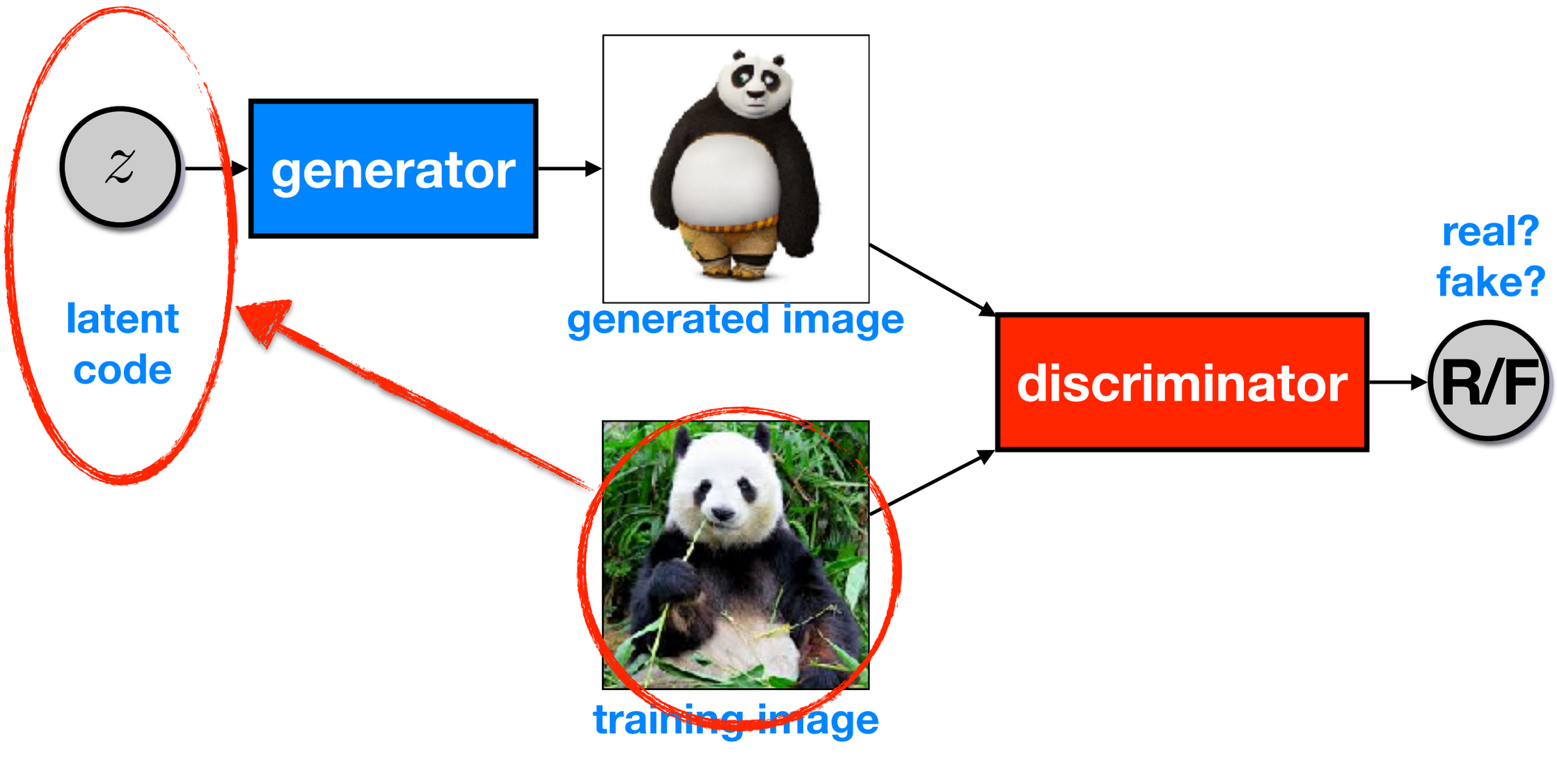
(b) Wasserstein distance.



(c) Evolution of data (in blue) generated from GAN (top row), UnrolledGAN (middle row) and our D2GAN (bottom row) on 2D data of 8 Gaussians. Data sampled from the true mixture are red.

GAN: Works Fine, But We Do Care Latent Space

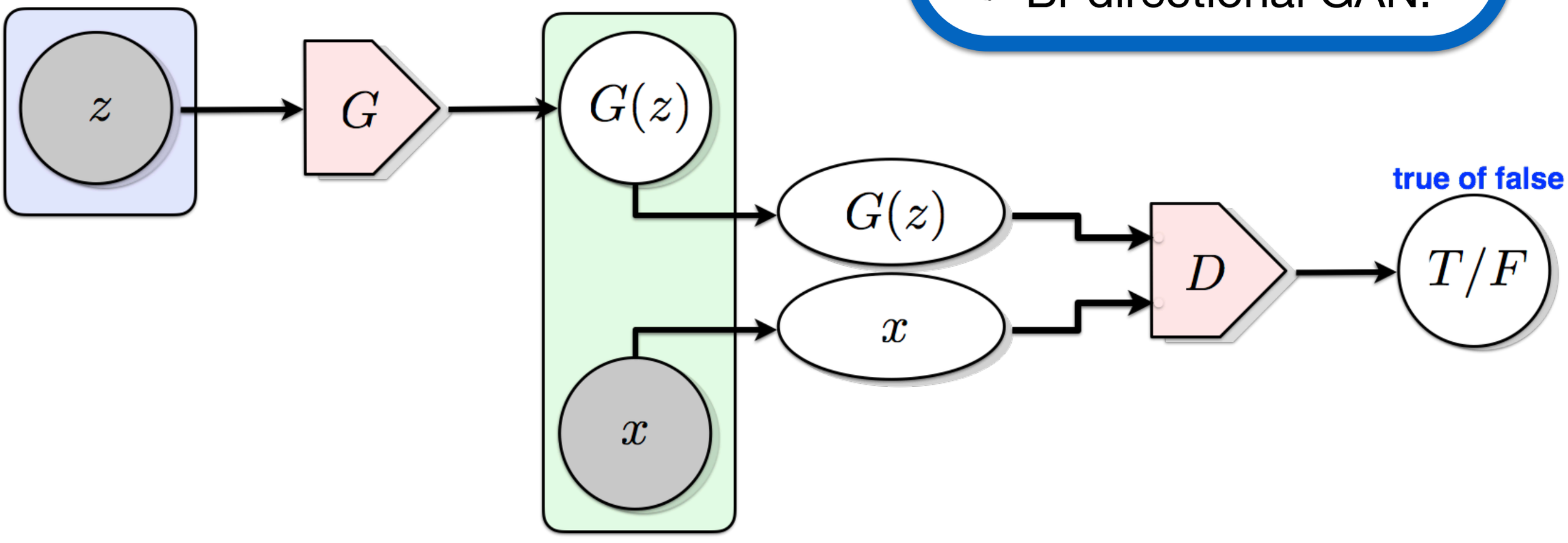
- Do you get any any feature representation?
- Given a x, can you find the most appropriate z ?



GAN: Works Fine, But We Do Care Latent Space

- Do you get any any feature representation?
- Given a x, can you find the most appropriate z ?

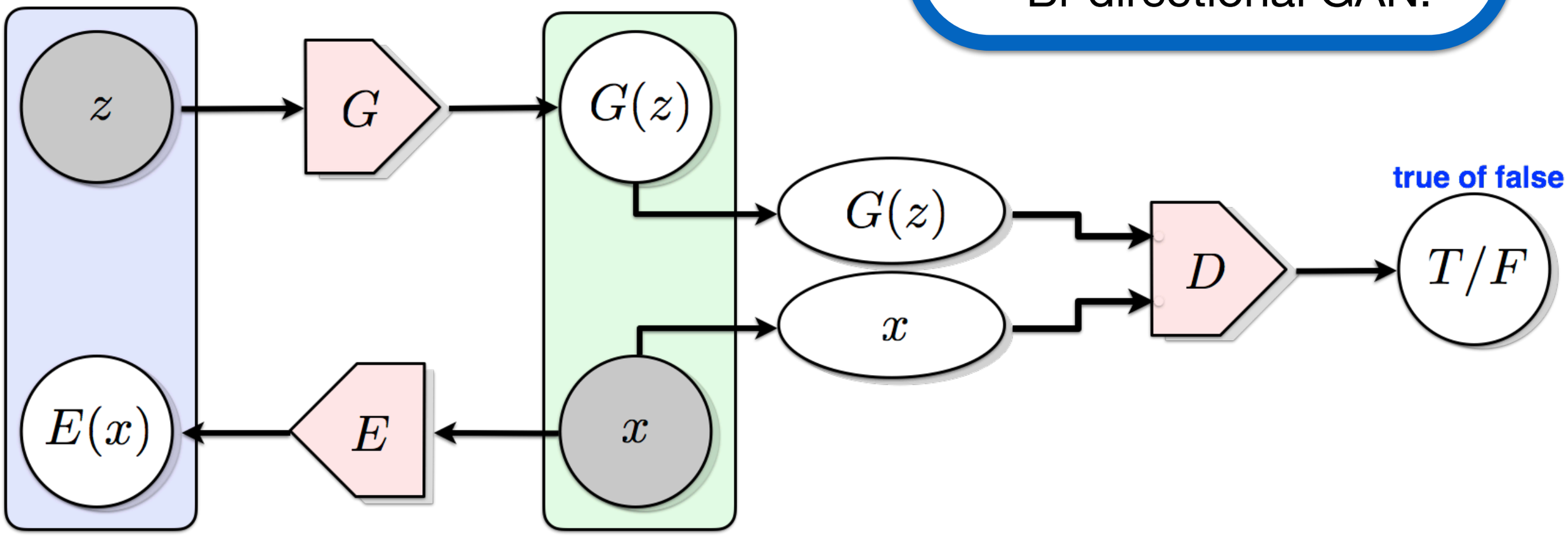
Trevor Darrell @UCB
Bi-directional GAN!



GAN: Works Fine, But We Do Care Latent Space

- ☞ Do you get any any feature representation?
- ☞ Given a x, can you find the most appropriate z ?

Trevor Darrell @UCB
☞ Bi-directional GAN!

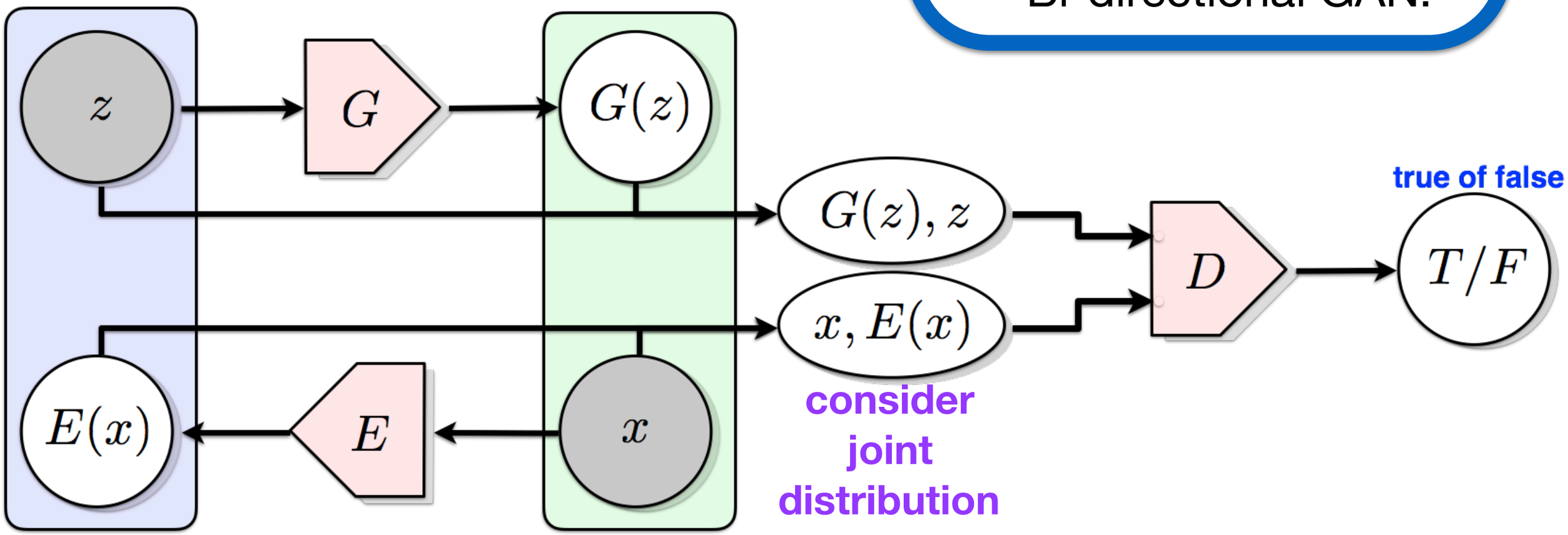


GAN: Works Fine, But We Do Care Latent Space

- Do you get any any feature representation?
- Given a x, can you find the most appropriate z ?

Trevor Darrell @UCB

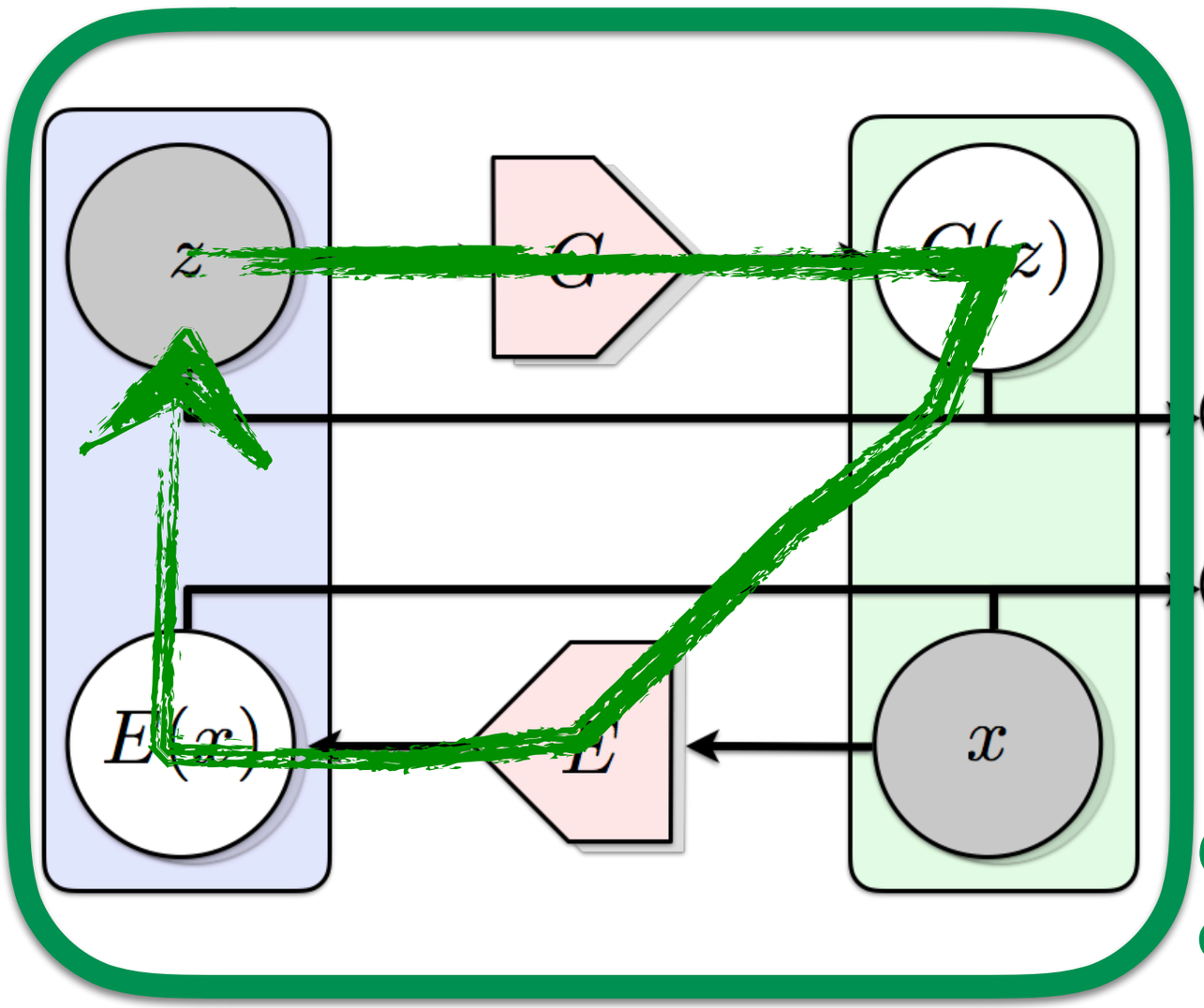
Bi-directional GAN!

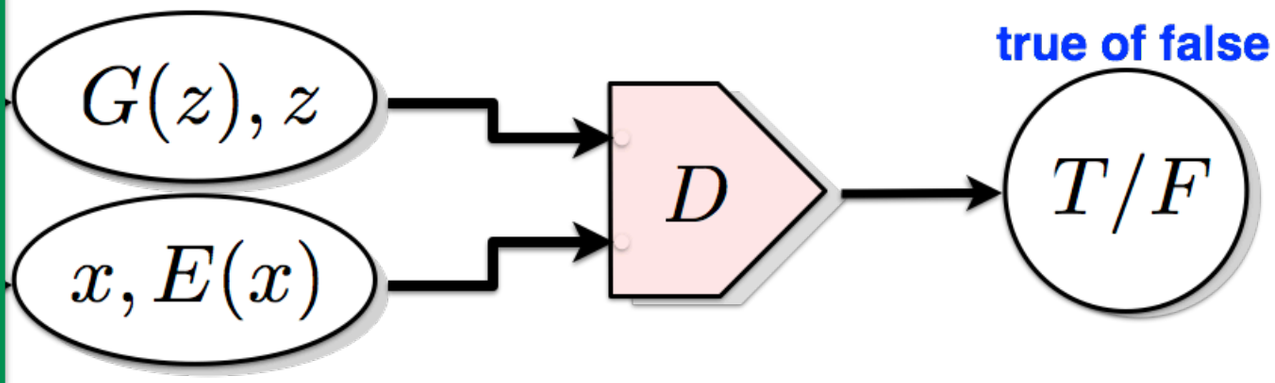


GAN: Works Fine, But We Do Care Latent Space

- Do you get any any feature representation?
- Given a x, can you find the most appropriate z ?

Trevor Darrell @UCB
Bi-directional GAN!

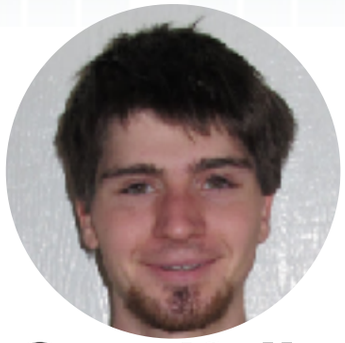




Given optimal *D*, *G* and *E*'s objective equivalent to autoencoder w/ ℓ_0 loss

BiGAN, Similar to VAE? Ahhh!

A. Makhzani

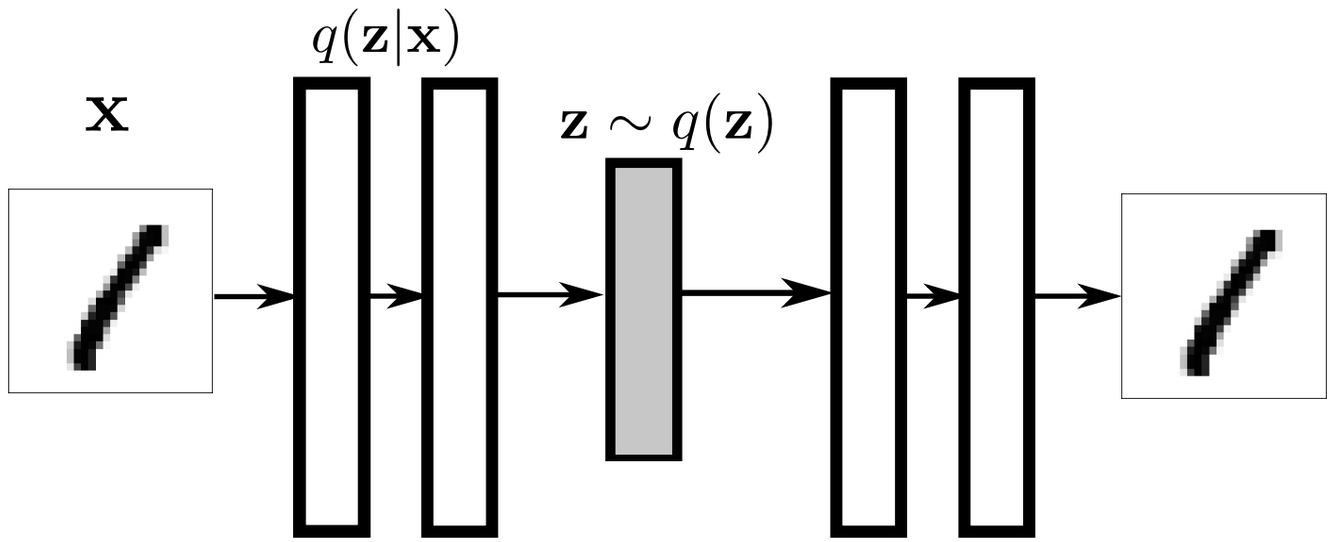


I. Goodfellow

others

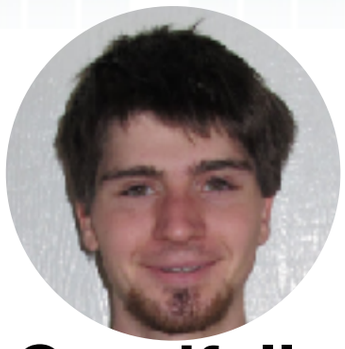
Still remember VAE?

- ☞ **reconstruction error**
- ☞ **impose prior on latent space**



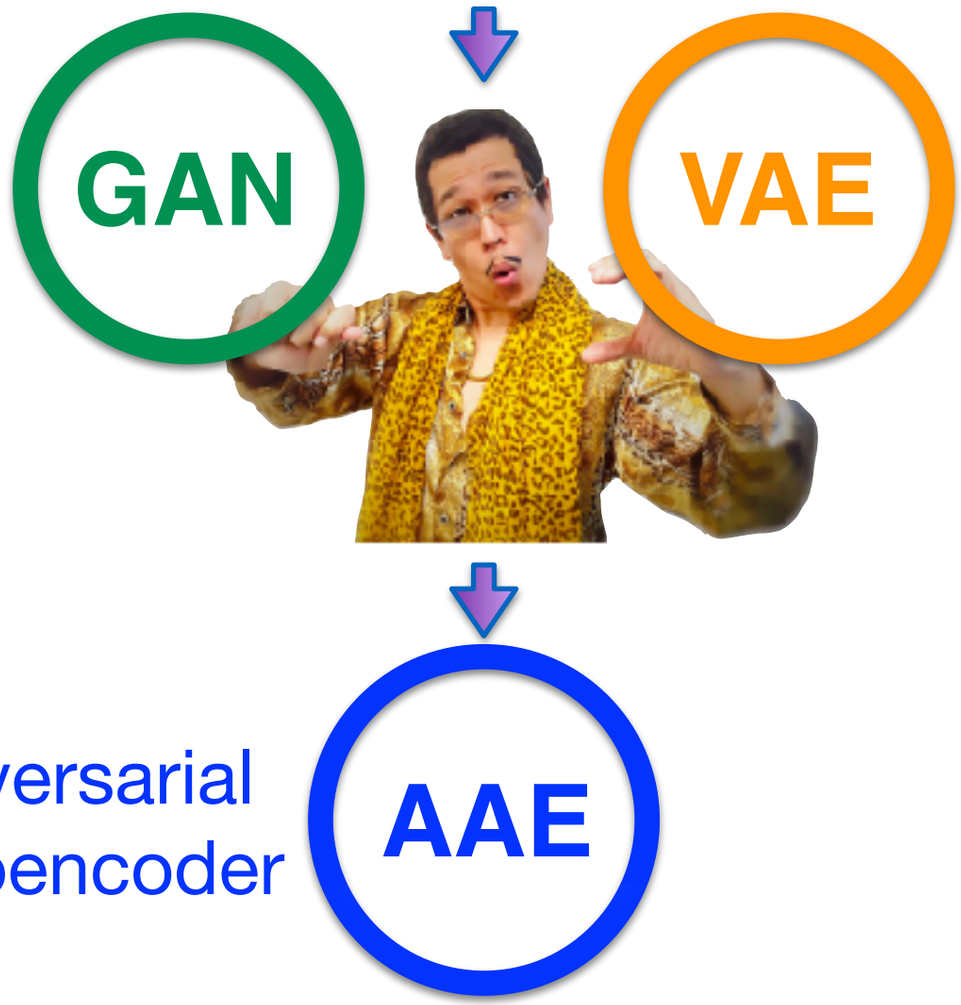
BiGAN, Similar to VAE? Ahhh!

A. Makhzani



I. Goodfellow

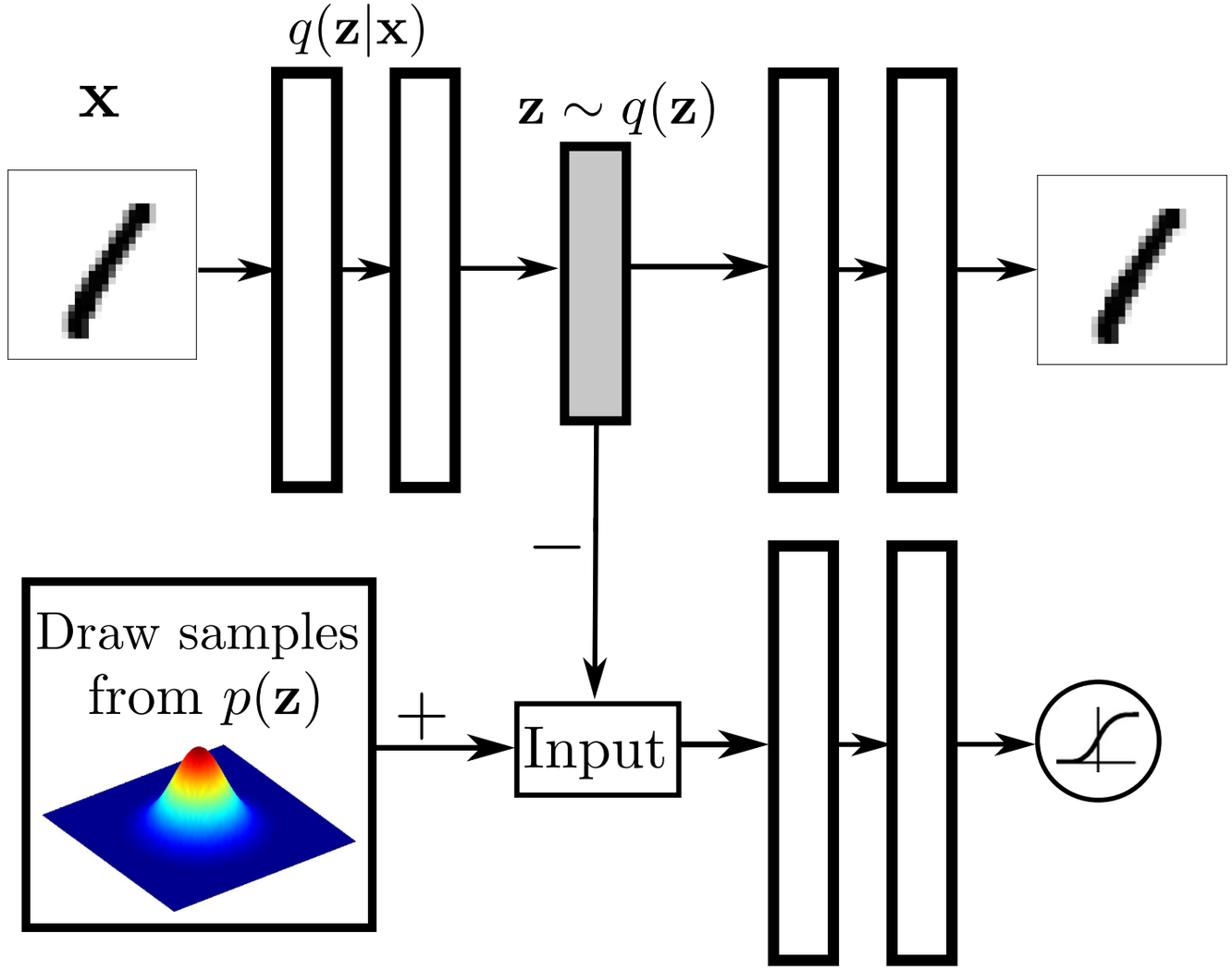
others …



Adversarial Autoencoder

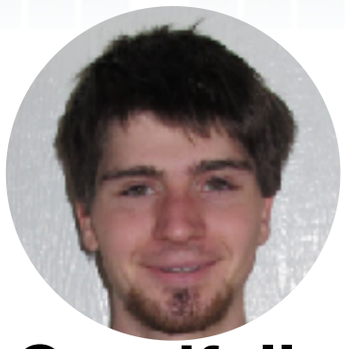
Still remember VAE?

- **reconstruction error**
- **impose prior on latent space**
- **impose ANY prior on latent space by adversarial loss!**



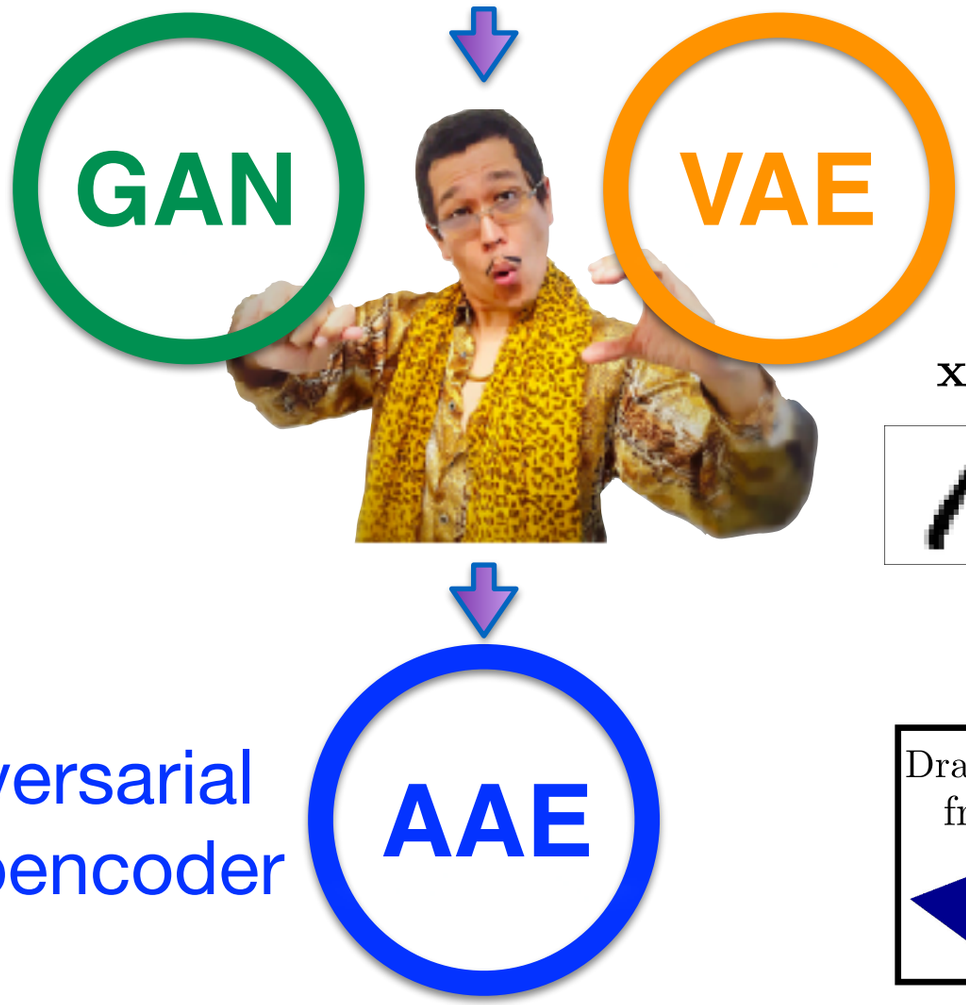
BiGAN, Similar to VAE? Ahhh!

A. Makhzani



I. Goodfellow

others •••

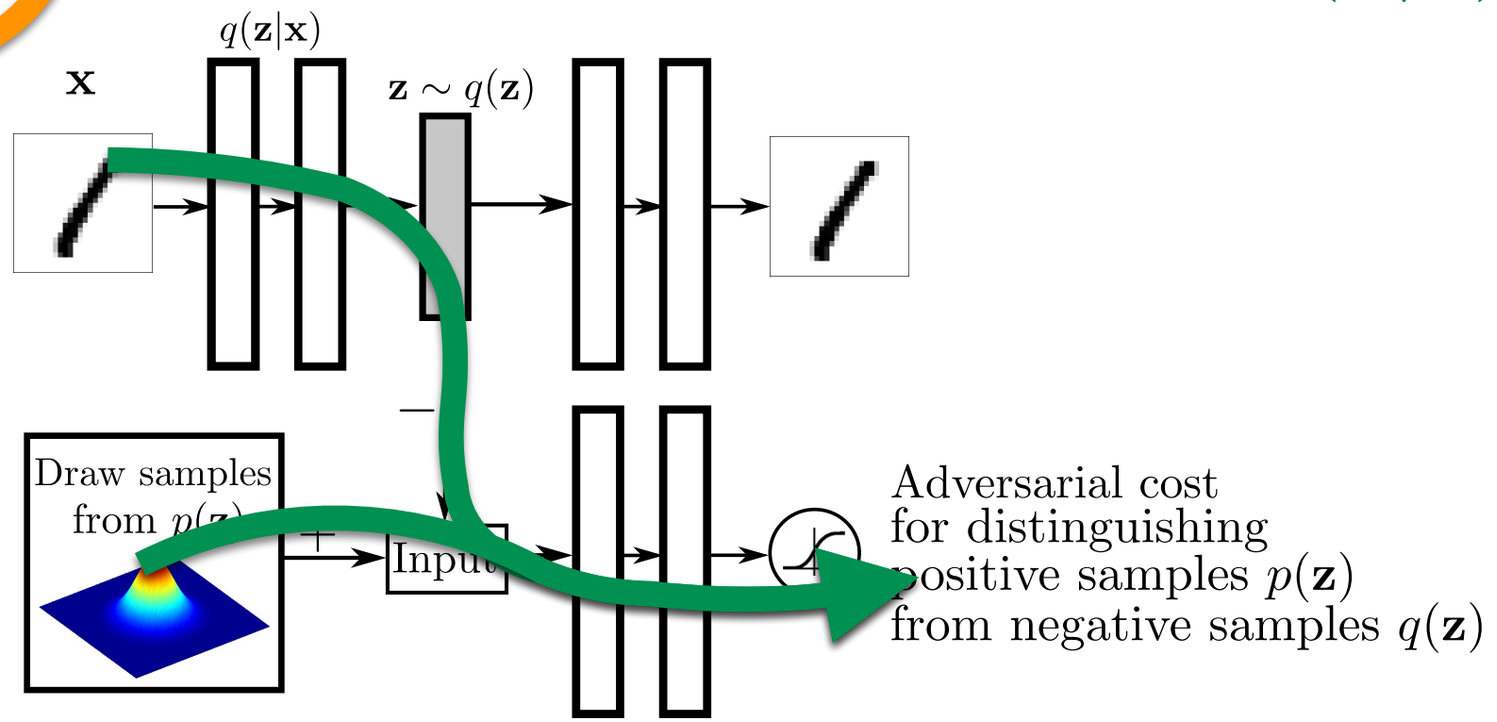


Adversarial Autoencoder

Still remember VAE?

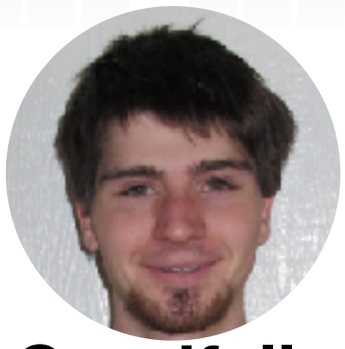
- reconstruction error
- impose prior on latent space
- impose ANY prior on latent space by adversarial loss!

The generator of adversarial network is also the encoder of autoencoder $q(z|x)$



BiGAN, Similar to VAE? Ahhh!

A. Makhzani



I. Goodfellow

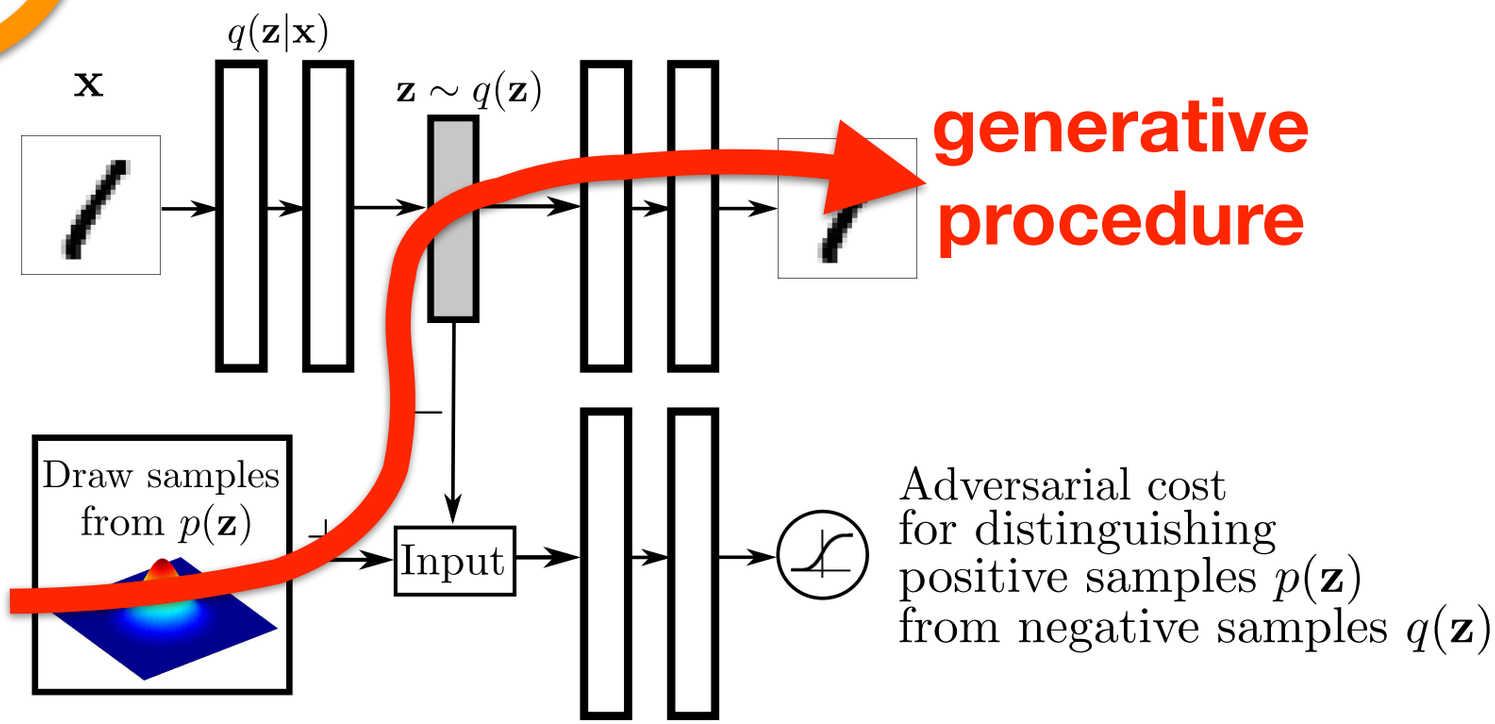
others

•••

Adversarial Autoencoder

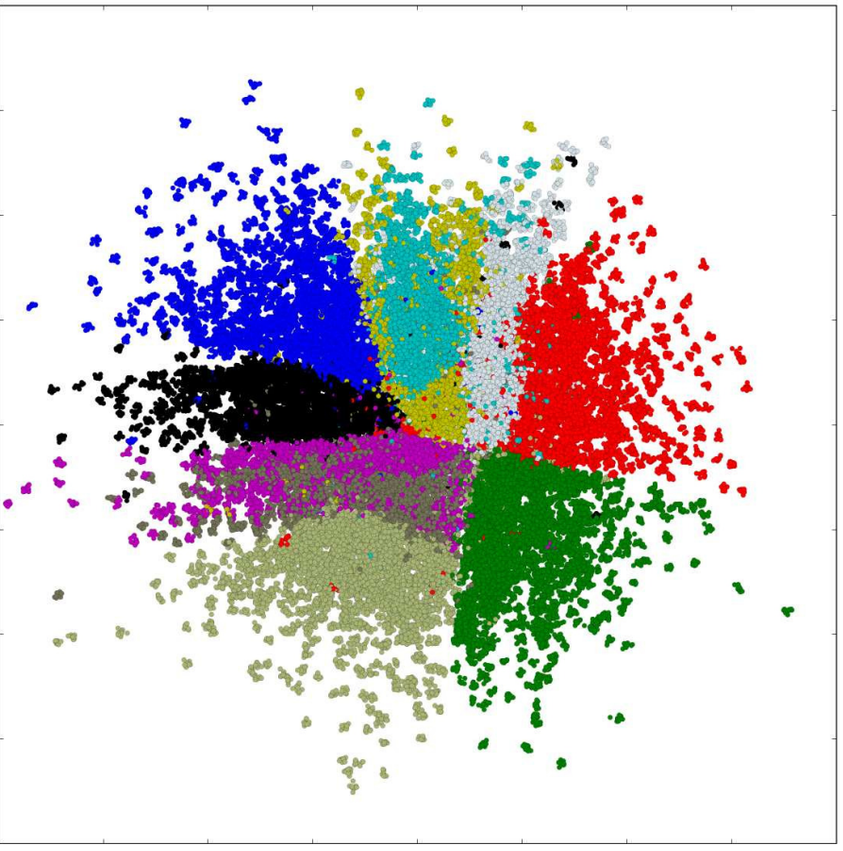
Still remember VAE?

- ☞ **reconstruction error**
- ☞ **impose prior on latent space**
- ☞ **impose ANY prior on latent space by adversarial loss!**

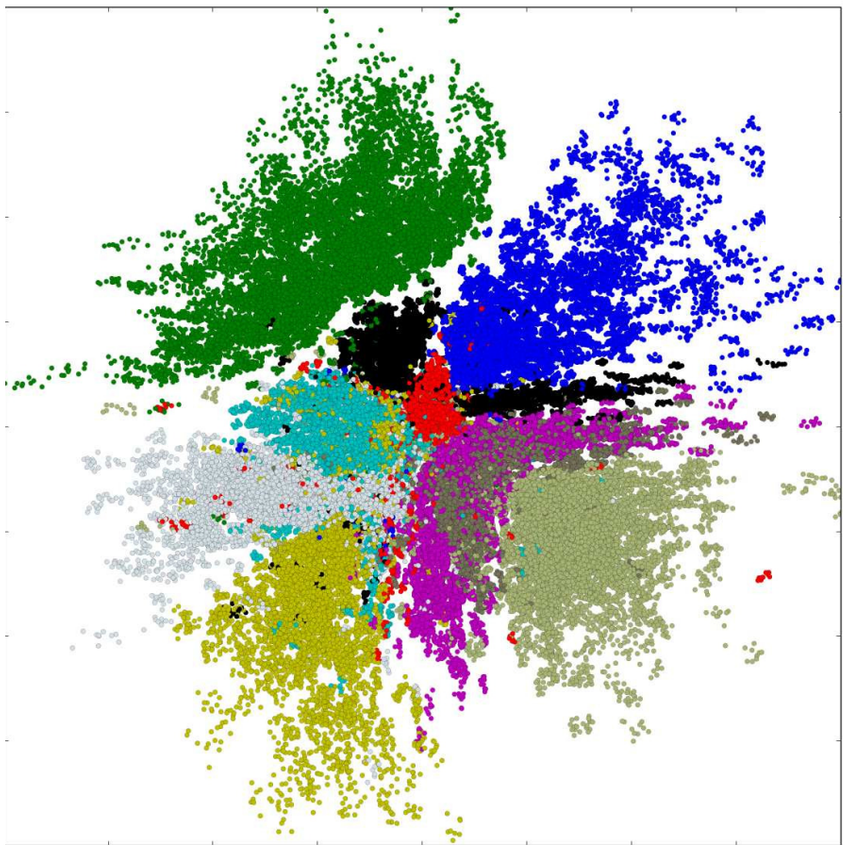


Example Results of Adversarial Autoencoer

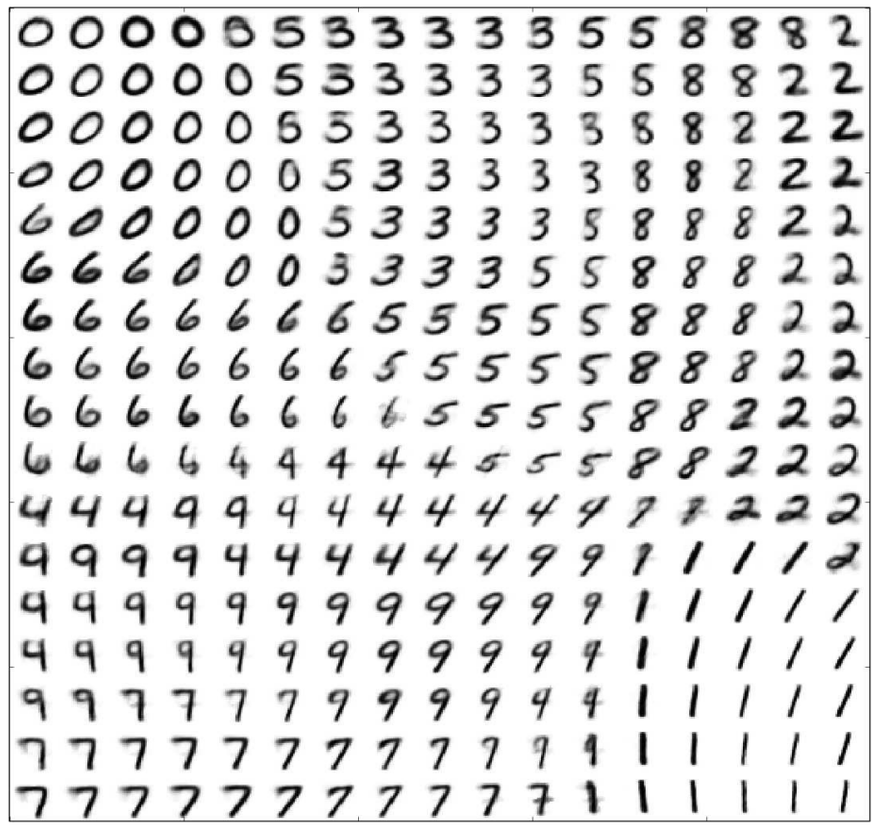
Adversarial Autoencoder



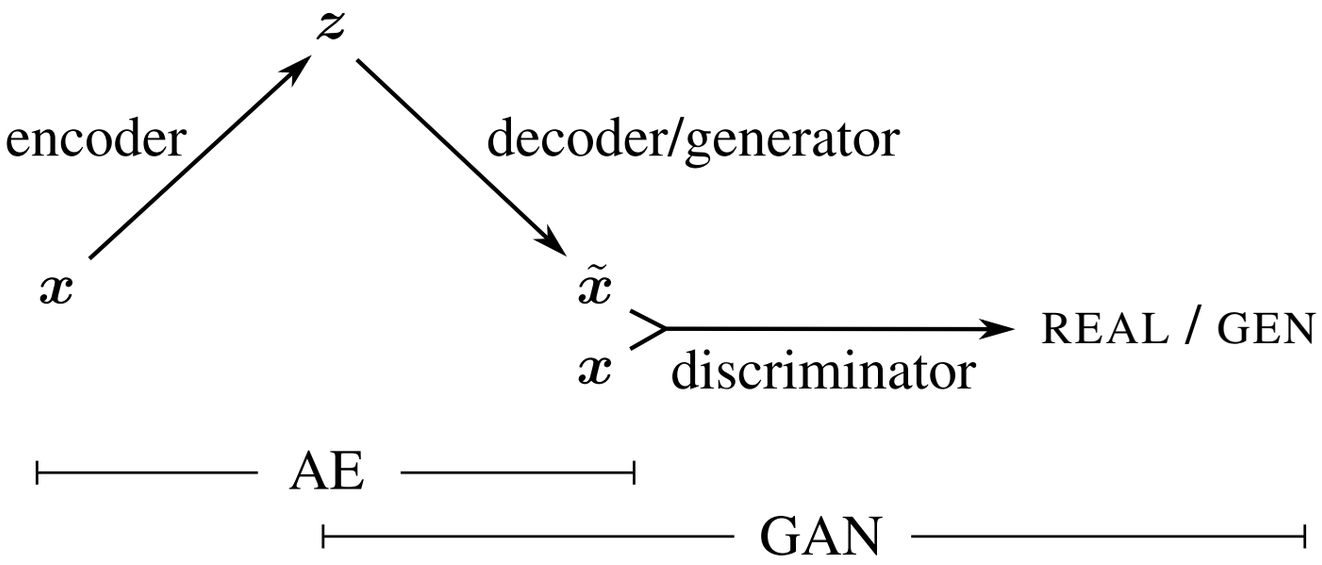
Variational Autoencoder

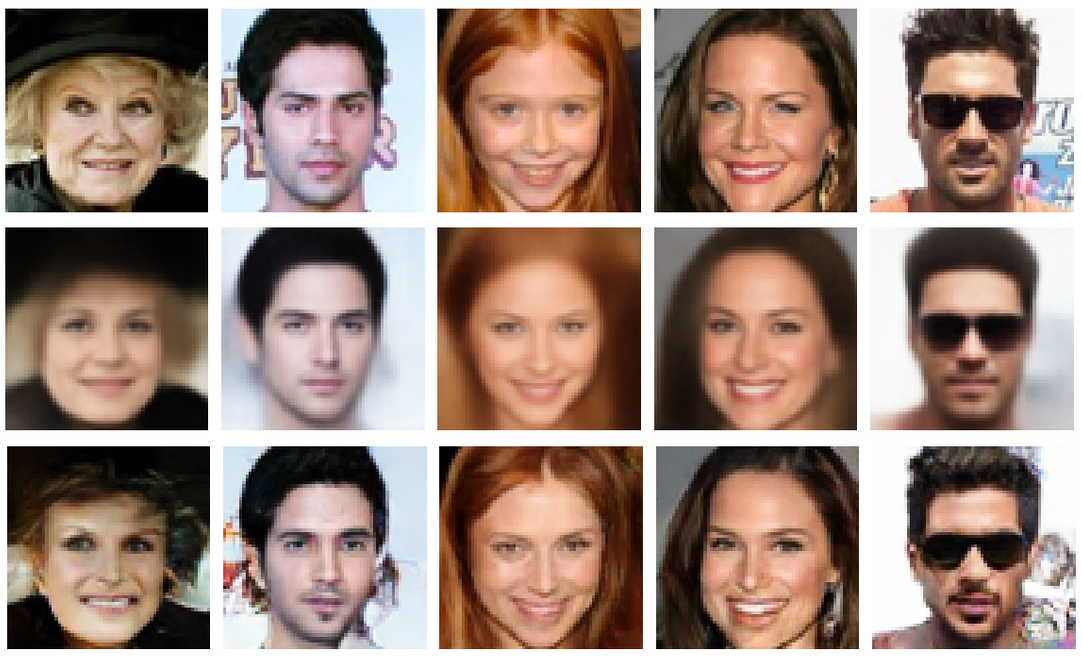


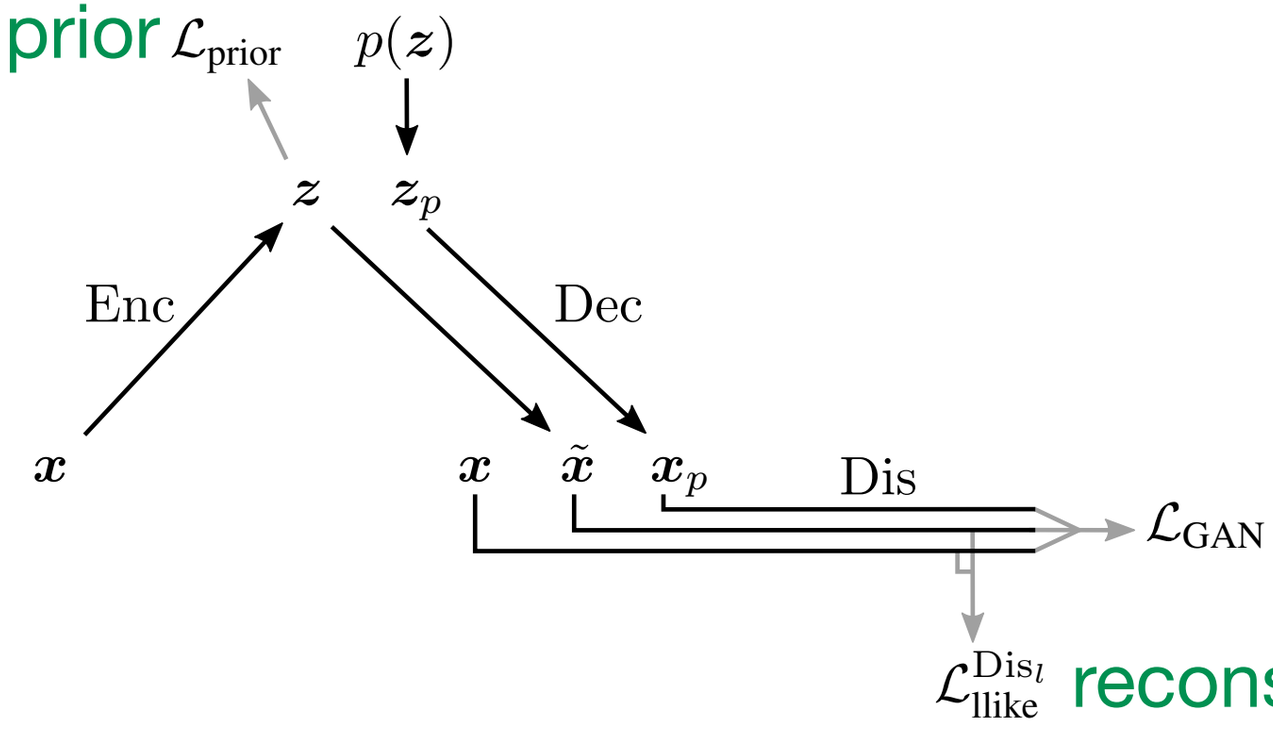
Manifold of
Adversarial Autoencoder



Other Way To Combine VAE and GAN







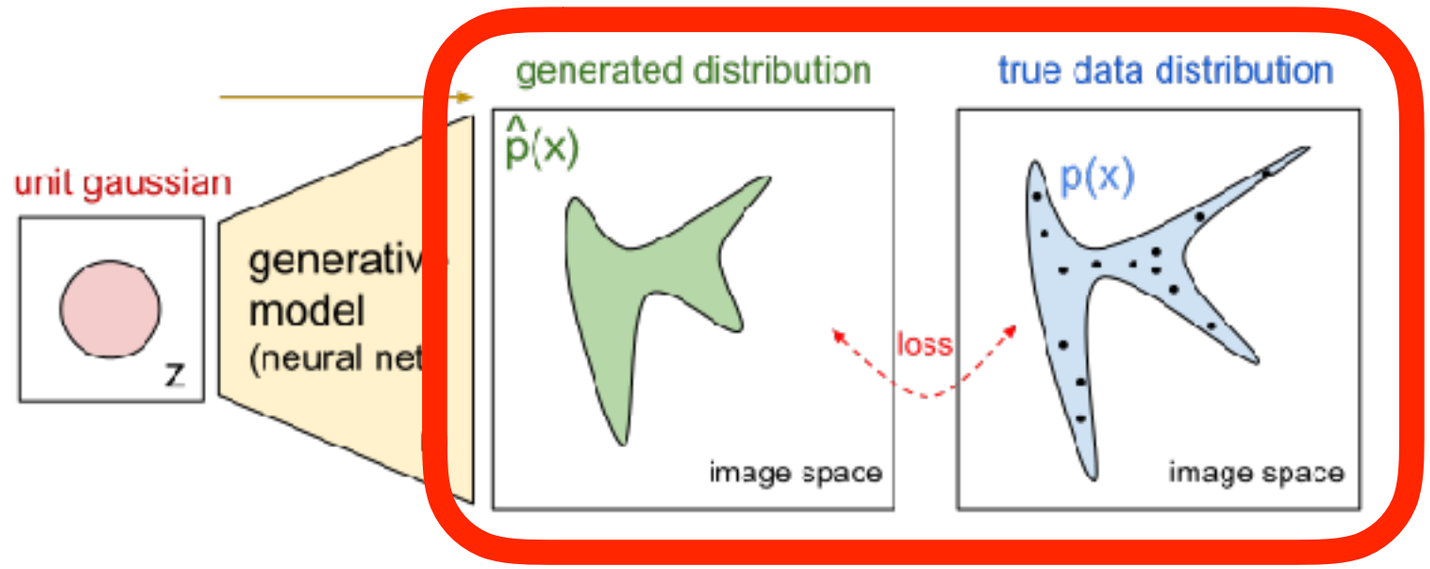
distinguish b/w real and synthesized images (adversarial loss treated most as improving image quality)

reconstruction

[Ref: Larsen et al. Autoencoding beyond pixels using a learned similarity metric. ICML'16]

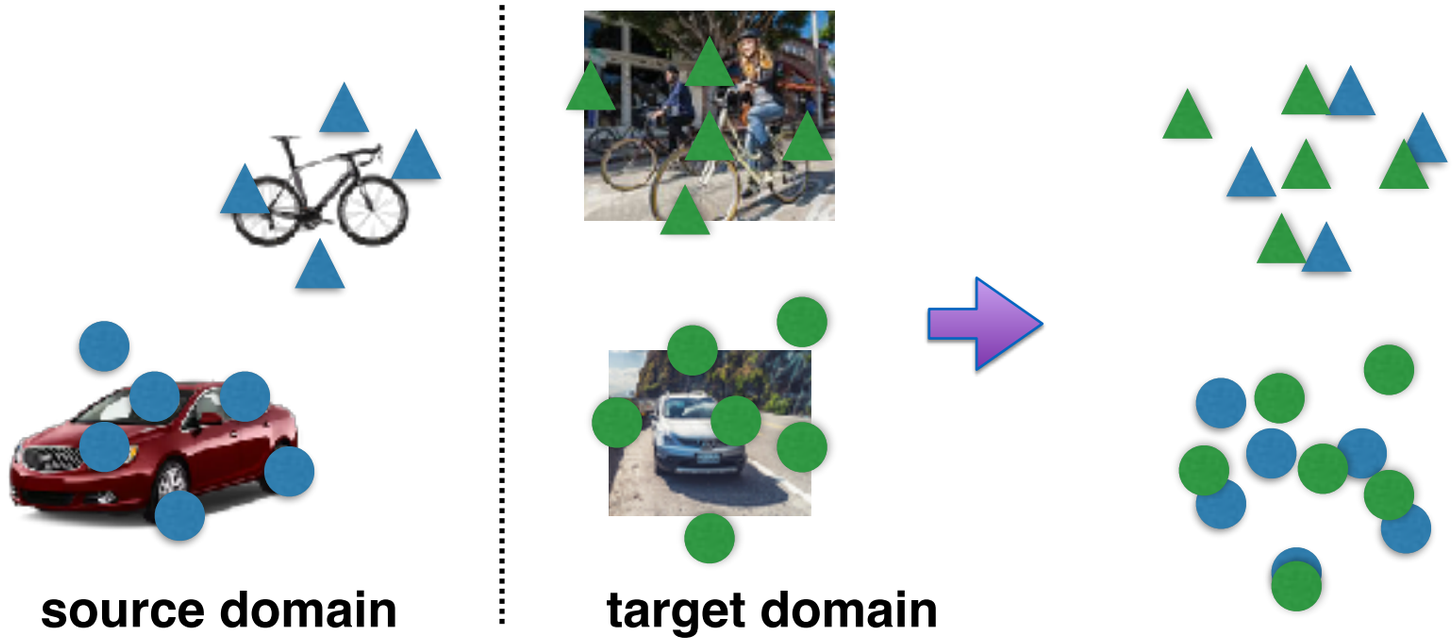
Adversarial Loss

GAN ☞impose **adversarial loss** on **data distribution**

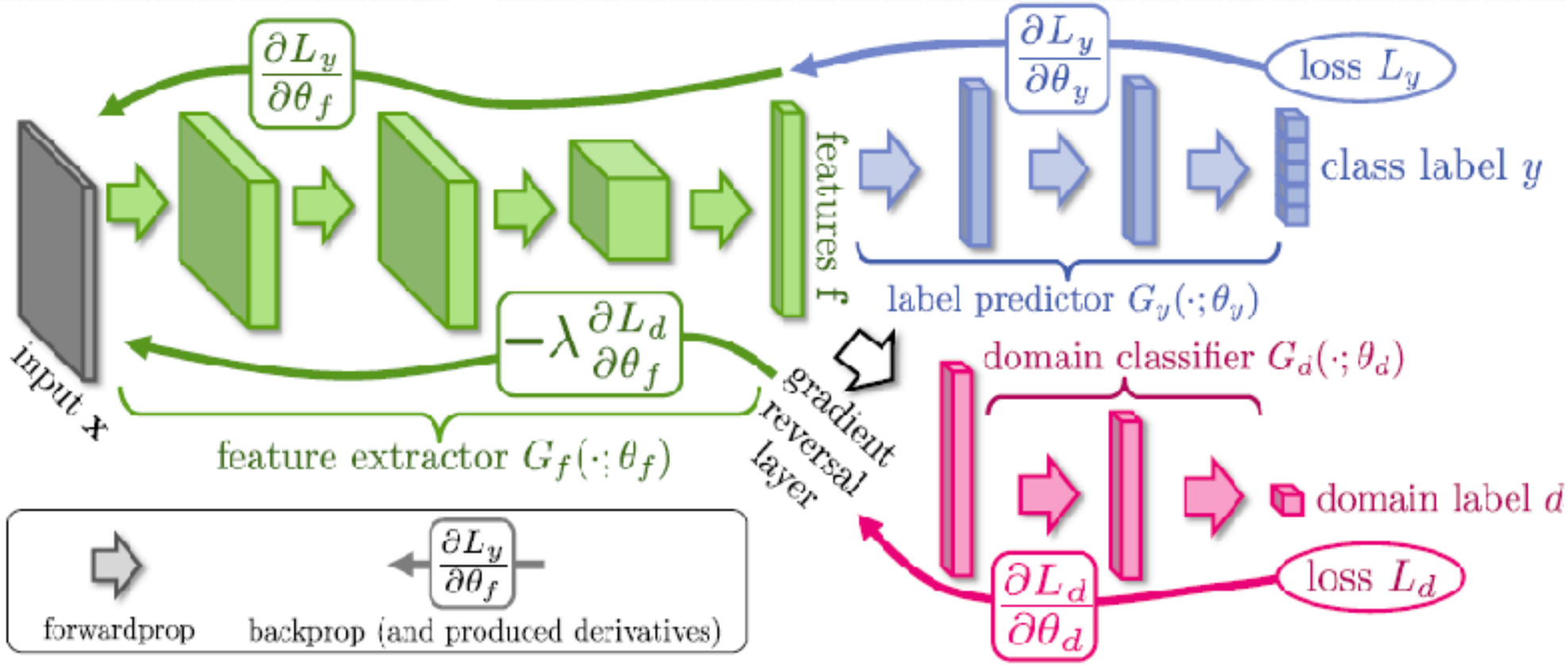


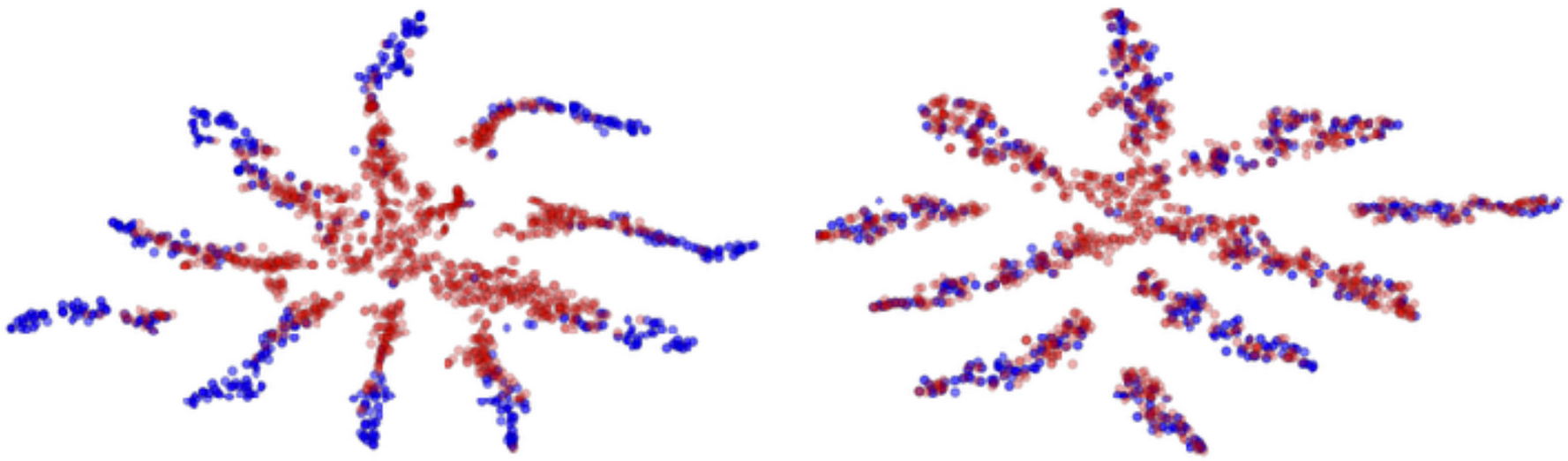
adversarial loss functions to match two distributions!

what we like to have in “Domain Adaptation”



Adversarial Loss for Domain Adaptation

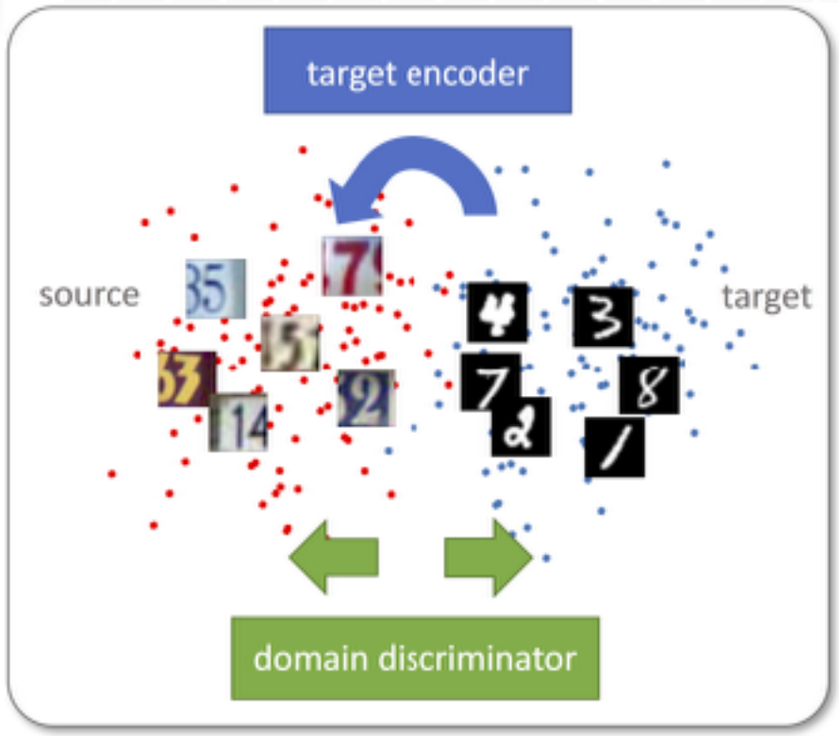


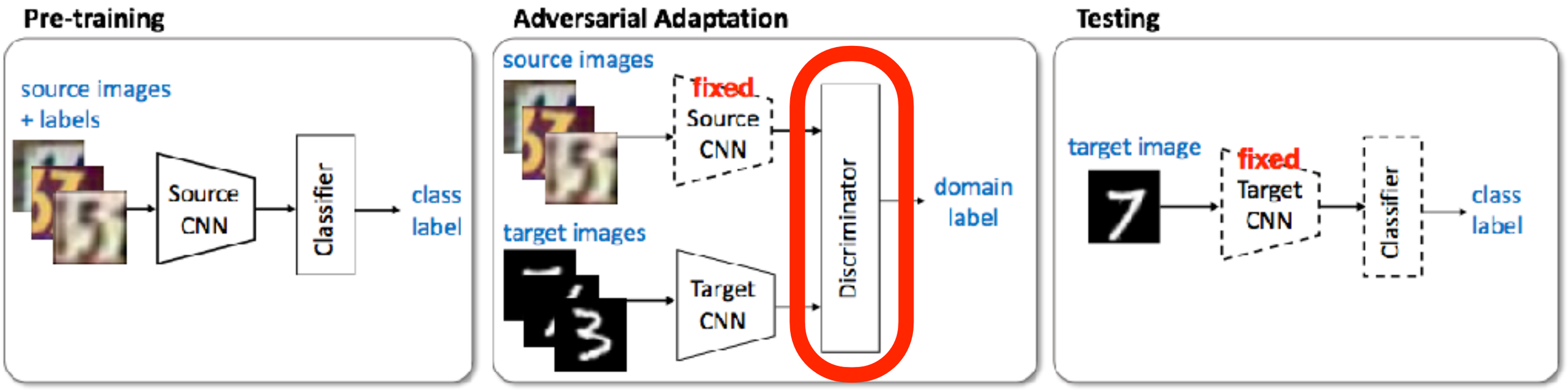


(a) Non-adapted (b) Adapted

[Ref: Ganin et al., Unsupervised Domain Adaptation by Backpropagation, ICML'15]

Adversarial Loss for Domain Adaptation

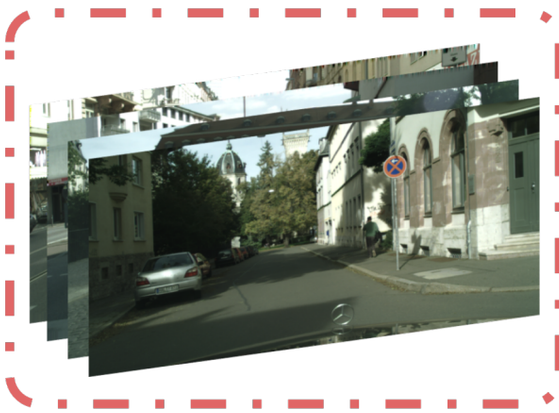




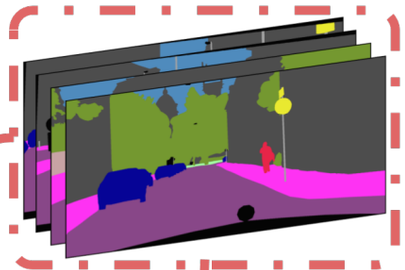
[Ref: Tzeng et al., Adversarial Discriminative Domain Adaptation, 2017]

Adversarial Loss for Domain Adaptation

Source domain: **labeled** data



Source domain: Ground Truth



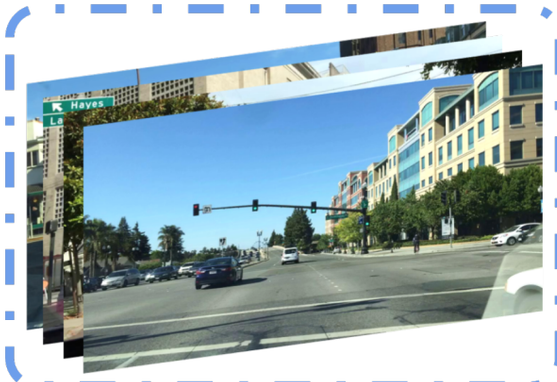
Class Size Distribution

Transfer

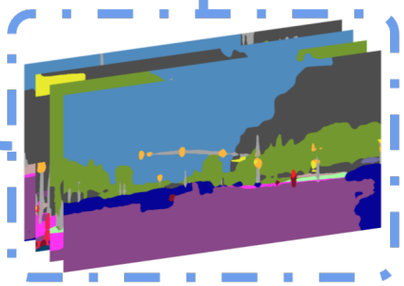
Constrained MI Loss

Shared Weight

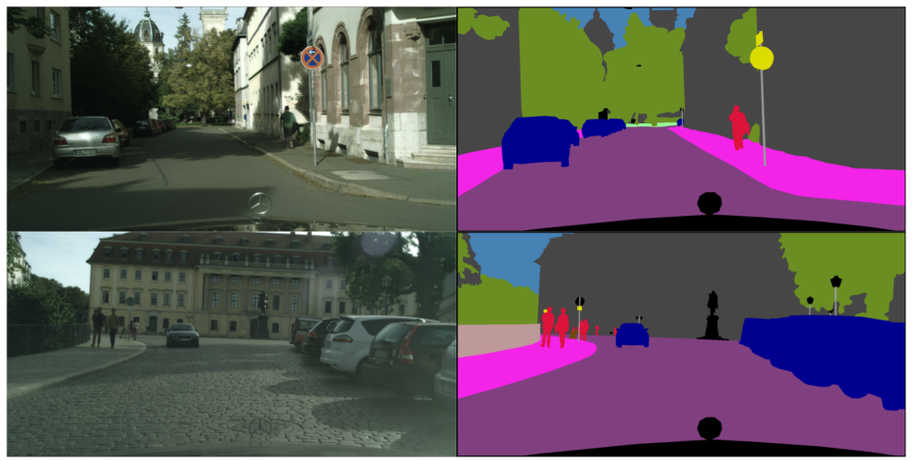
Domain Adversarial Training



Target domain: **unlabeled** data



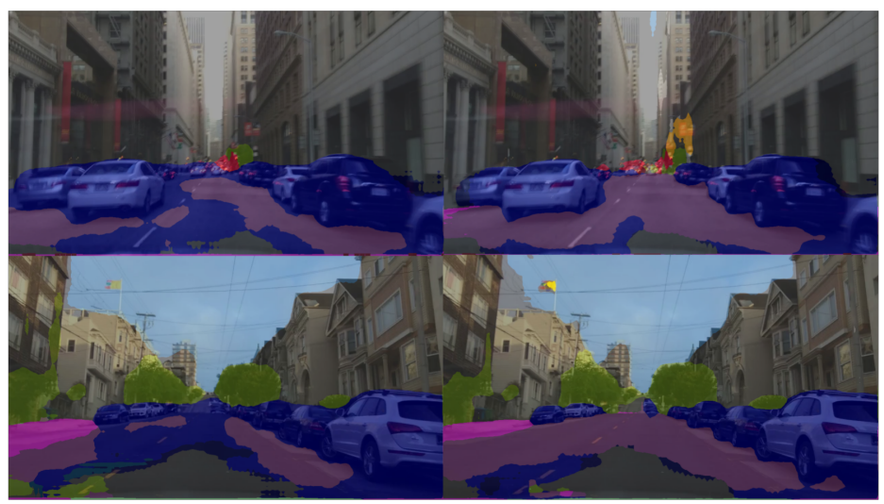
Target domain: Network Output



Source domain: lots of **labeled** data

Pixel-level Adaptation

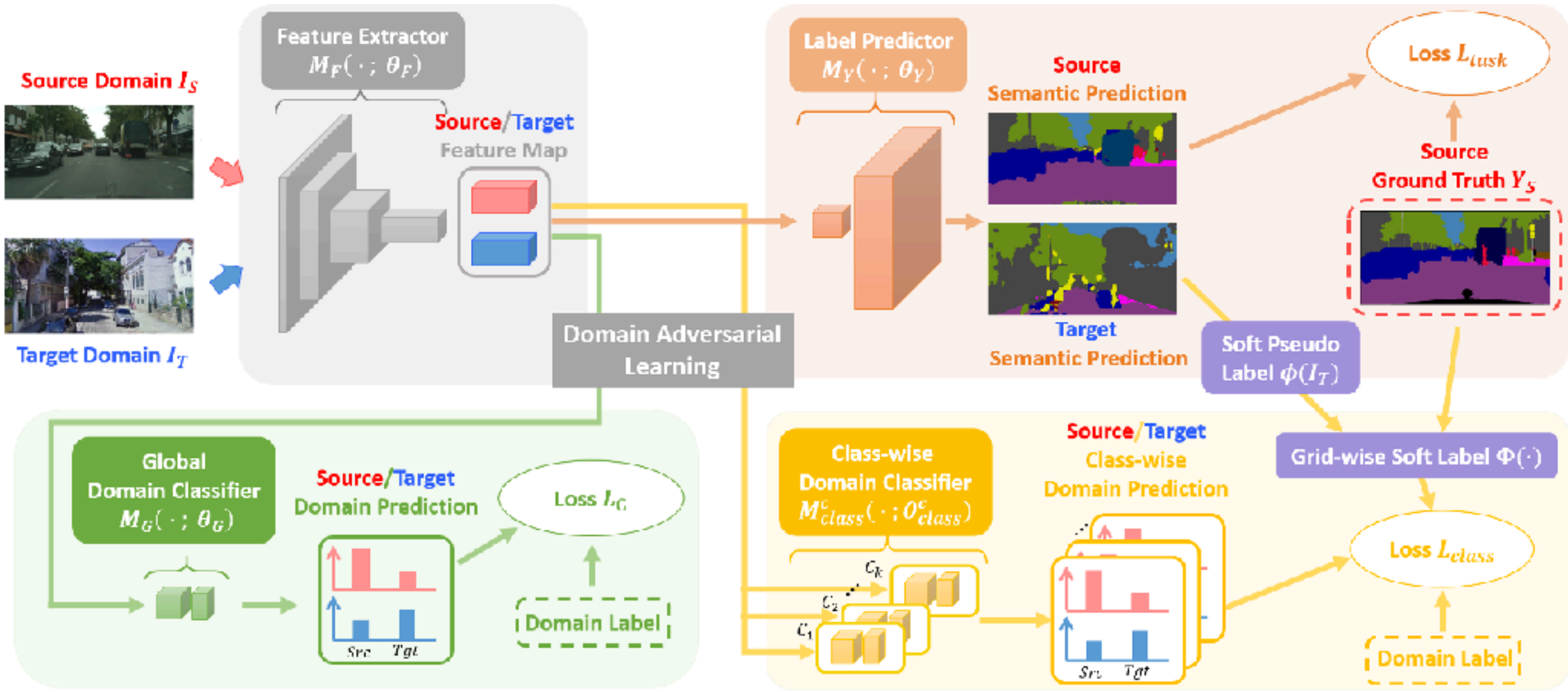
Target domain: lots of **unlabeled** data



Before Adaptation **After Adaptation**

[Ref: Tzeng et al., Adversarial Discriminative Domain Adaptation, 2017]

Adversarial Loss for Domain Adaptation

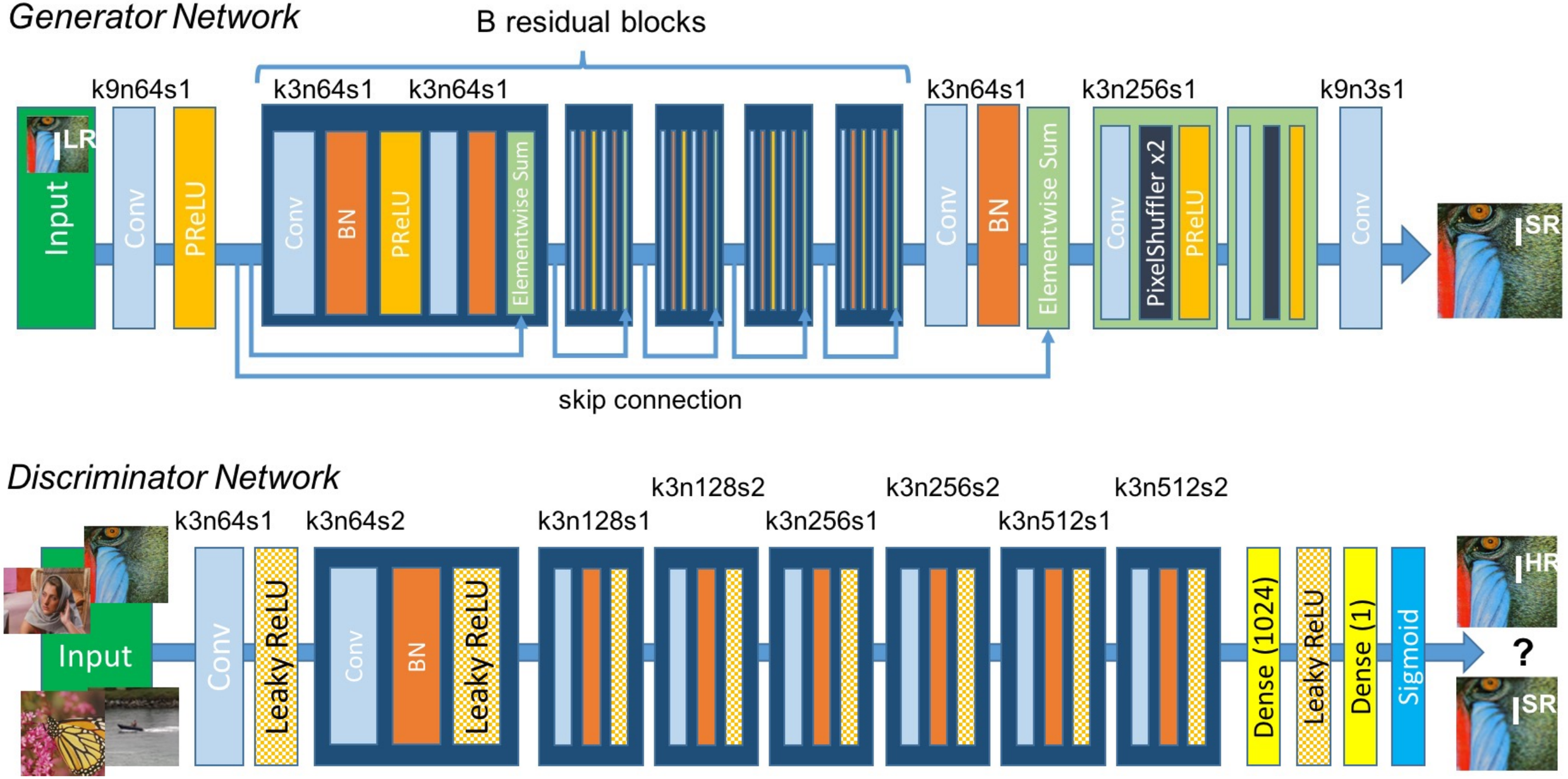


[Ref: Chen et al., No more discrimination: Cross city adaptation of road scene segmenters, ICCV'17]

Outlines

- Discriminative versus Generative Models
- Going Into Deep Generative Models
- From Autoencoder to Variational Autoencoder (VAE)
- From VAE to Generative Adversarial Network (GAN)
- **Various Applications**
- Understanding the latent space: disentanglement

Application on Image Super-Resolution



[Ref: Ledig et al.,
Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, ArXiv2016]

Application on Image Super-Resolution

deep residual net
optimised for MSE

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

4x up-sampling

[Ref: Ledig et al.,
Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, ArXiv2016]

Application on Image Inpainting

corrupting mask

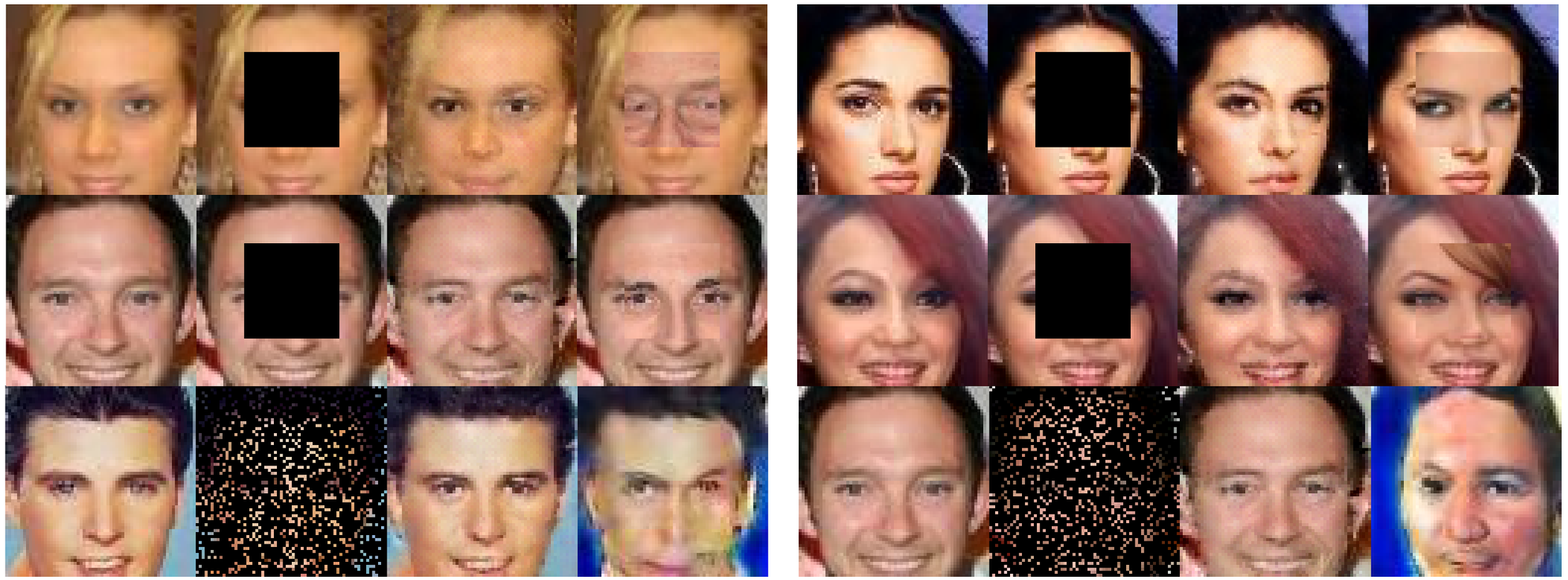
corrupted image

$$\mathcal{L}_{contextual}(\mathbf{z}) = \|\mathbf{M} \odot G(\mathbf{z}) - \mathbf{M} \odot \mathbf{y}\|_1$$

$$\mathcal{L}_{perceptual}(\mathbf{z}) = \log(1 - D(G(\mathbf{z})))$$

} optimising to find z

$$\mathbf{x}_{reconstructed} = \mathbf{M} \odot \mathbf{y} + (1 - \mathbf{M}) \odot G(\hat{\mathbf{z}})$$



[Ref: Yeh et al.,
Semantic Image Inpainting with Perceptual and Contextual Losses, ArXiv2016]

資訊工程學系
Department of Computer Science

Application on Interactive Image Editing

find the latent representation for X^R

$$z^0 = \arg\min_{z \in \tilde{\mathbb{Z}}} \mathcal{L}(G(z),\, x^R)$$

[Ref: Zhu et al.,
Generative Visual Manipulation on the Natural Image Manifold, ECCV2016]

(a) original photo

(e) different degree of image manipulation

Project

(b) projection on manifold

(c) Editing UI

(d) smooth transition between the original and edited projection

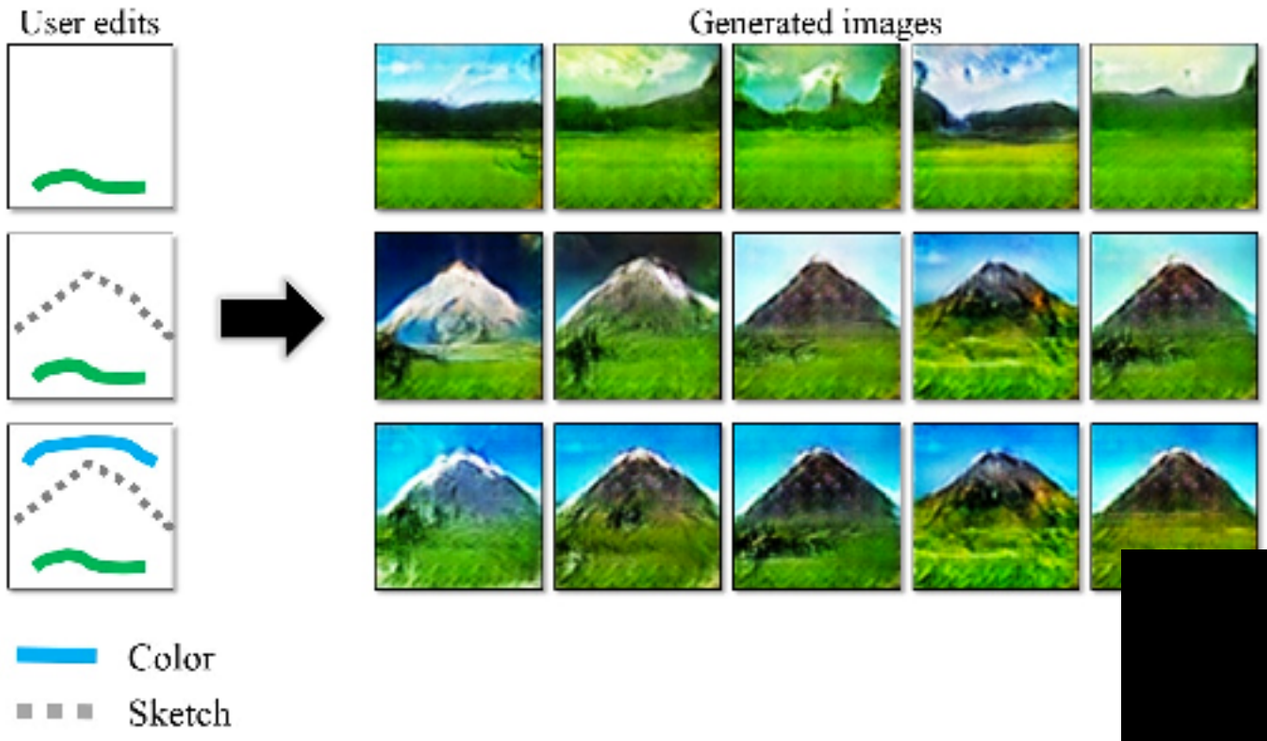
Edit Transfer
apply similar edit on others

$$f_g(x) = v_g \quad \textbf{certain editing effect}$$

$$z^* = \arg\min_{z \in \mathbb{Z}} \Big\{ \underbrace{\sum_g \|f_g(G(z)) - v_g\|^2}_{\text{data term}} + \underbrace{\lambda_s \cdot \|z - z_0\|^2}_{\substack{\text{manifold} \\ \text{smoothness}}} + E_D \Big\}$$

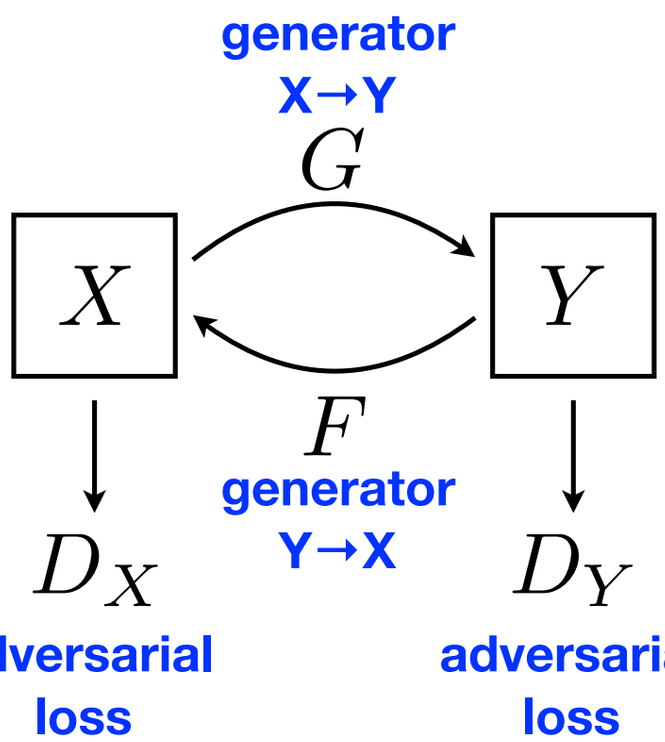
$$E_D = \lambda_D \cdot \log(1 - D(G(z))) \quad \textbf{adversarial loss for synthesising realistic images}$$

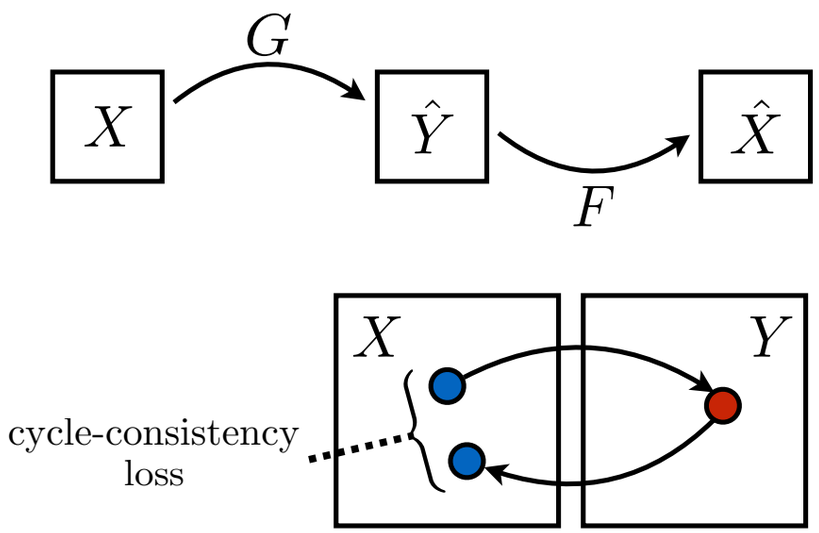
Application on Interactive Image Editing

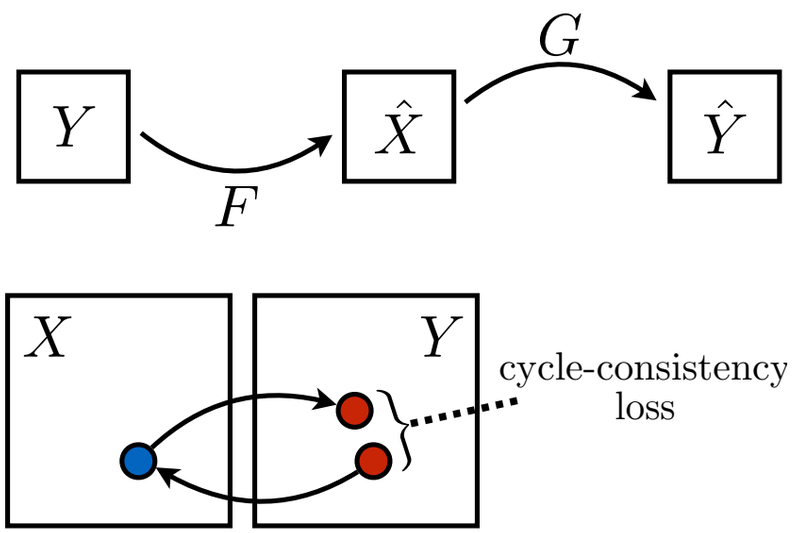


[redacted]

Application on Image-to-Image Translation





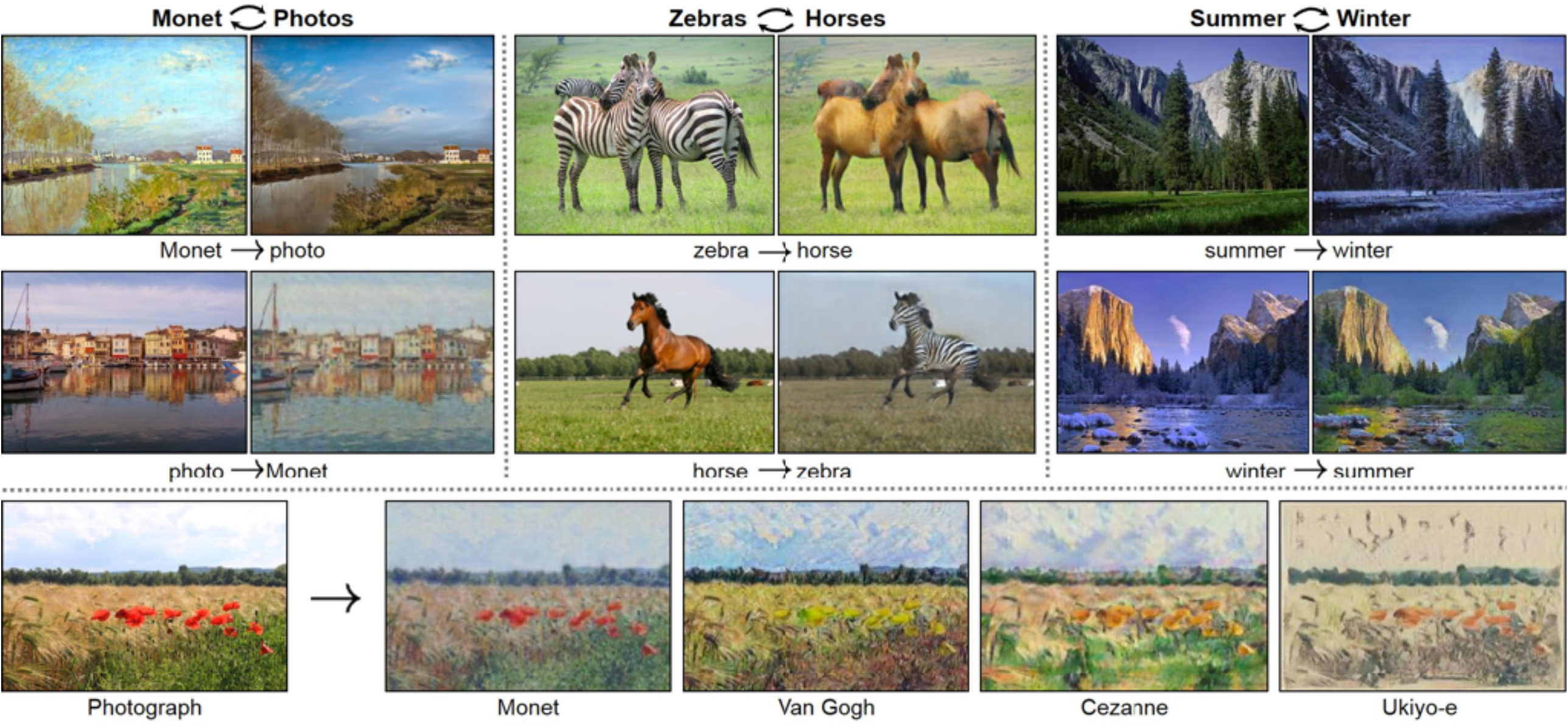


[Ref: Zhu et al., Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, ArXiv2017] **CycleGAN**

[Ref: Kim et al., Learning to Discover Cross-Domain Relations with Generative Adversarial Networks, ArXiv2017] **DiscoGAN**

[Ref: Yi et al., DualGAN: Unsupervised Dual Learning for Image-to-Image Translation, ArXiv2017] **DualGAN**

Application on Image-to-Image Translation

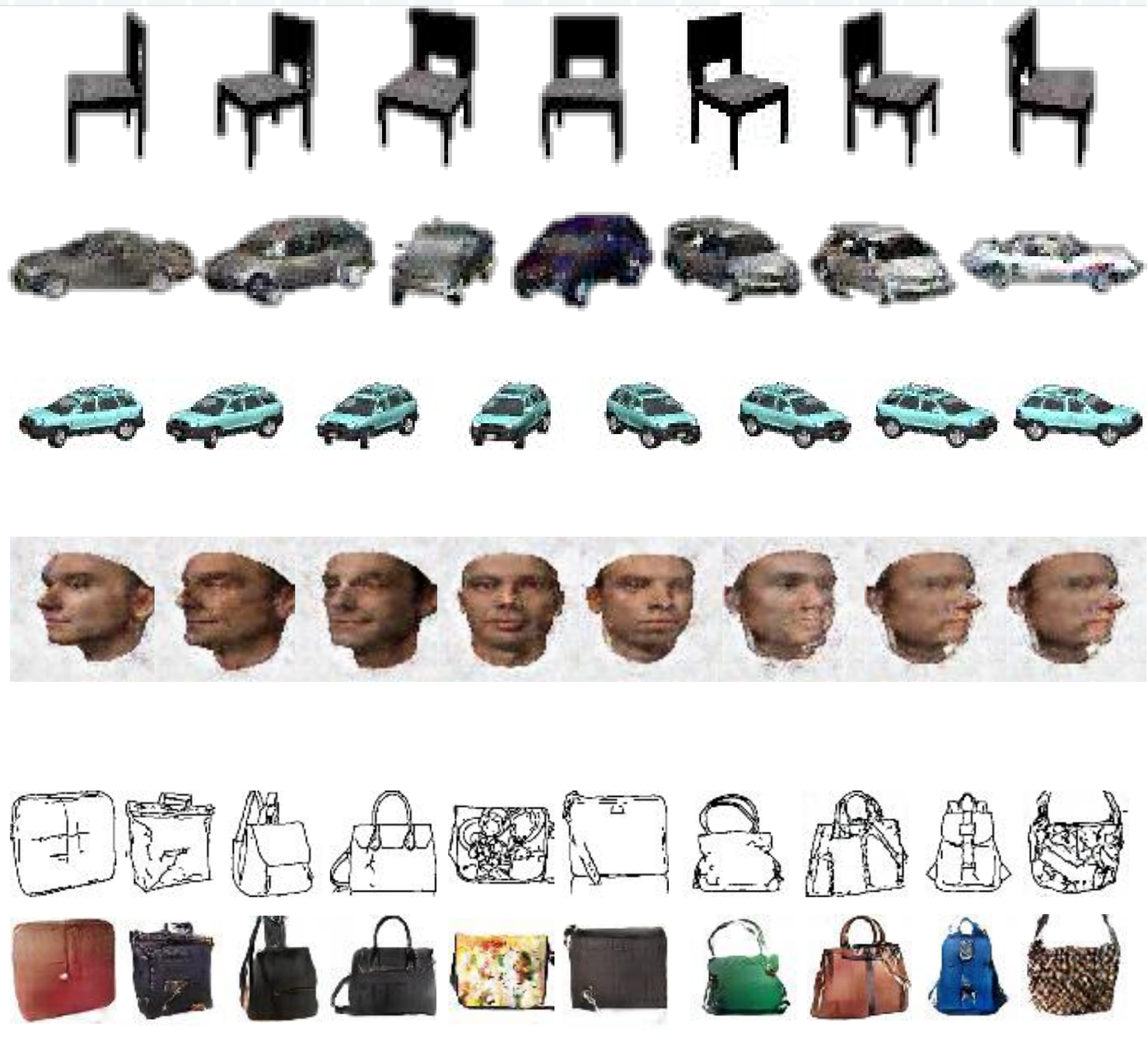


[Ref: Zhu et al., Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, ArXiv2017] **CycleGAN**

[Ref: Kim et al., Learning to Discover Cross-Domain Relations with Generative Adversarial Networks, ArXiv2017] **DiscoGAN**

[Ref: Yi et al., DualGAN: Unsupervised Dual Learning for Image-to-Image Translation, ArXiv2017] **DualGAN**

Application on Image-to-Image Translation

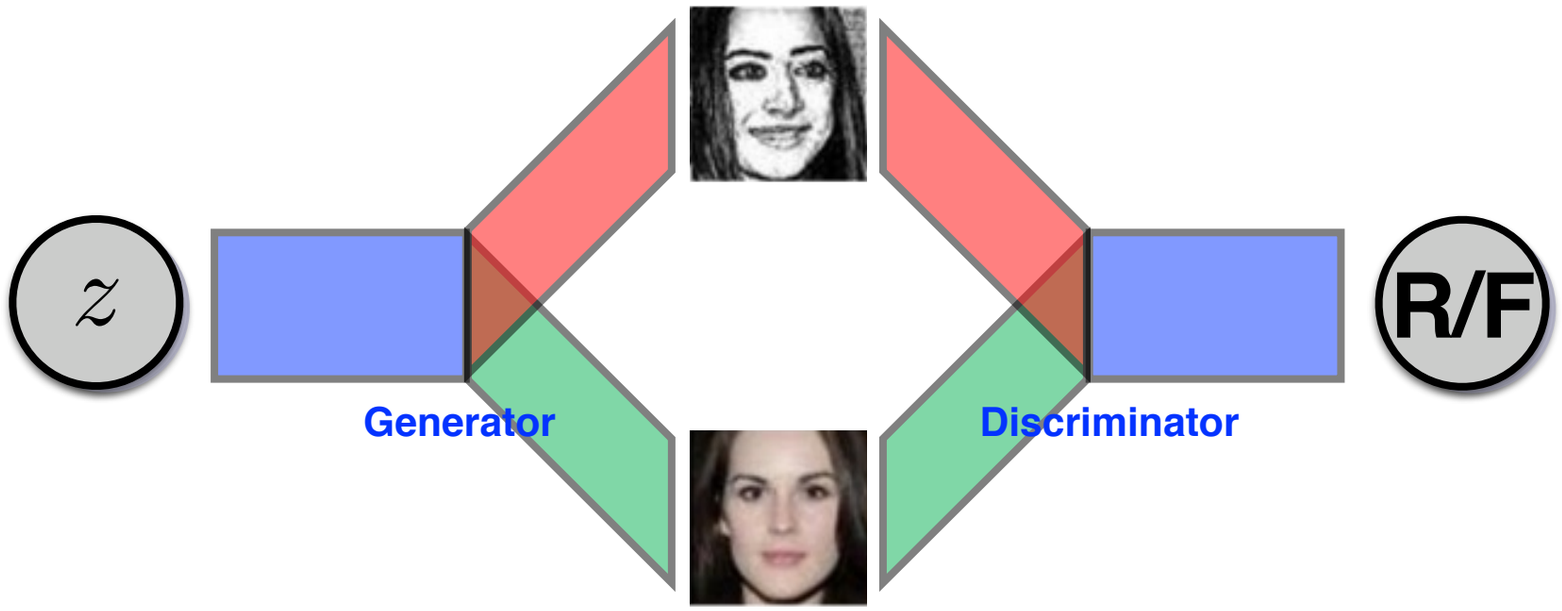


[Ref: Zhu et al., Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, ArXiv2017] **CycleGAN**

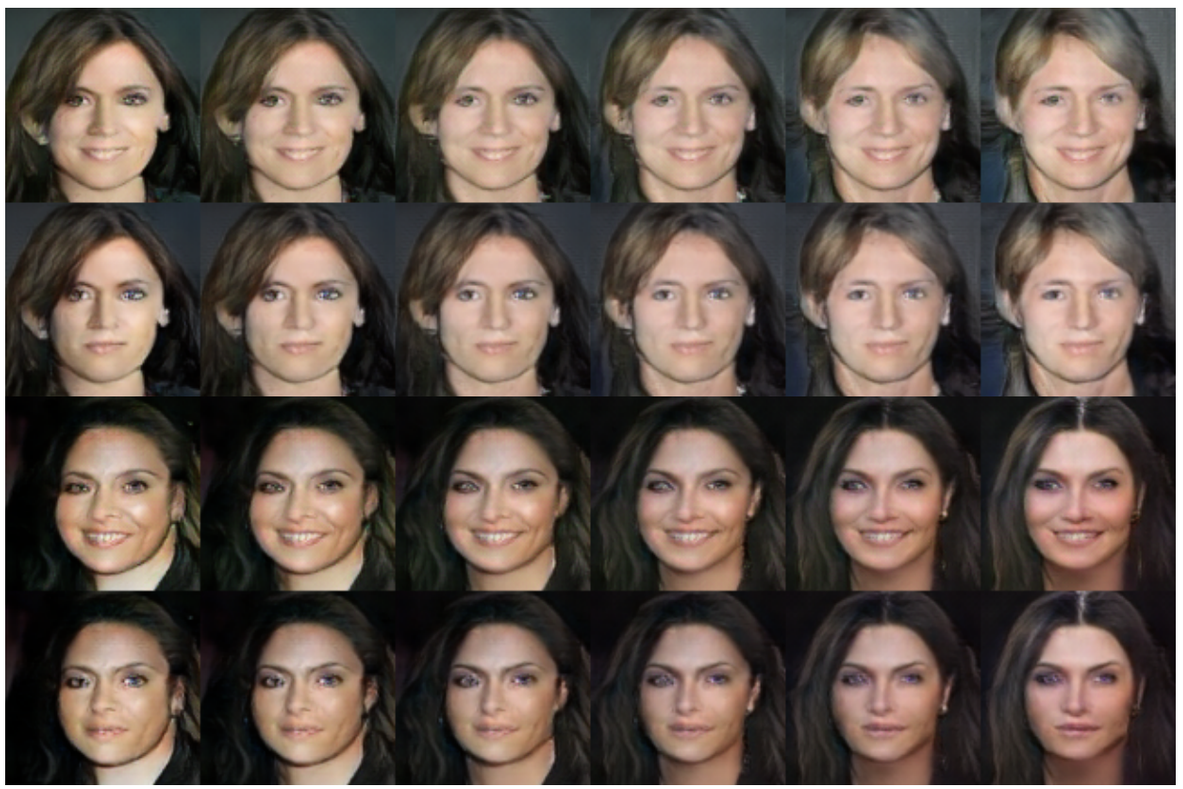
[Ref: Kim et al., Learning to Discover Cross-Domain Relations with Generative Adversarial Networks, ArXiv2017] **DiscoGAN**

[Ref: Yi et al., DualGAN: Unsupervised Dual Learning for Image-to-Image Translation, ArXiv2017] **DualGAN**

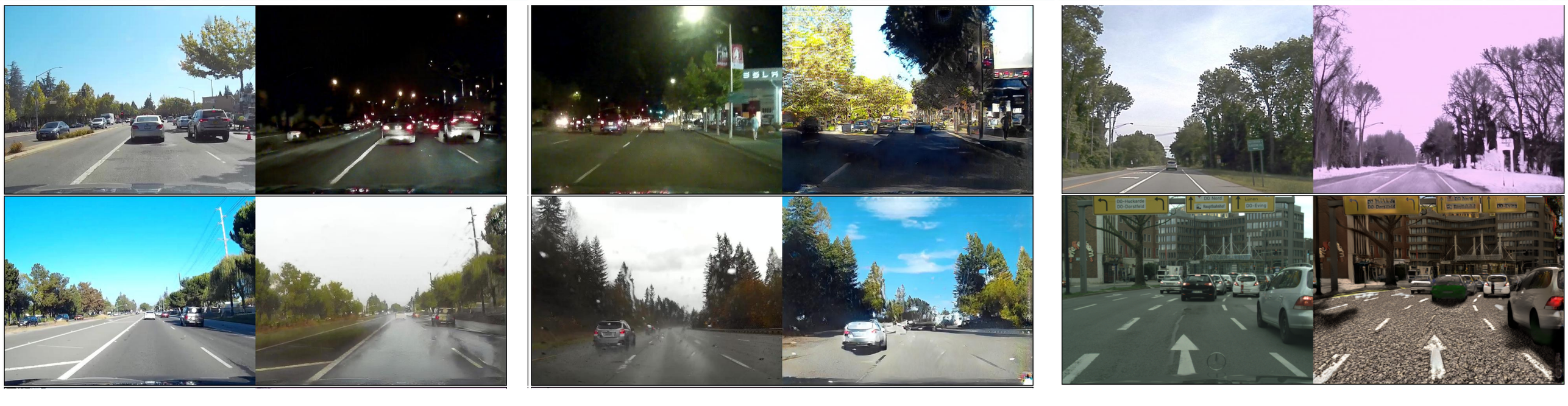
Application on Image-to-Image Translation

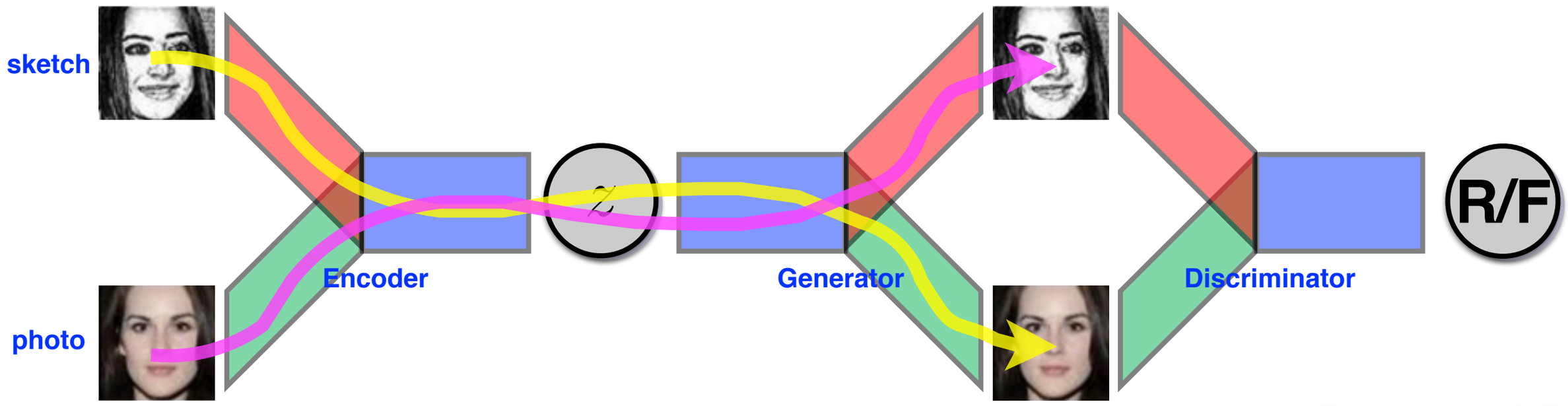


[Ref: Liu et al., Co-GAN, NIPS'16]



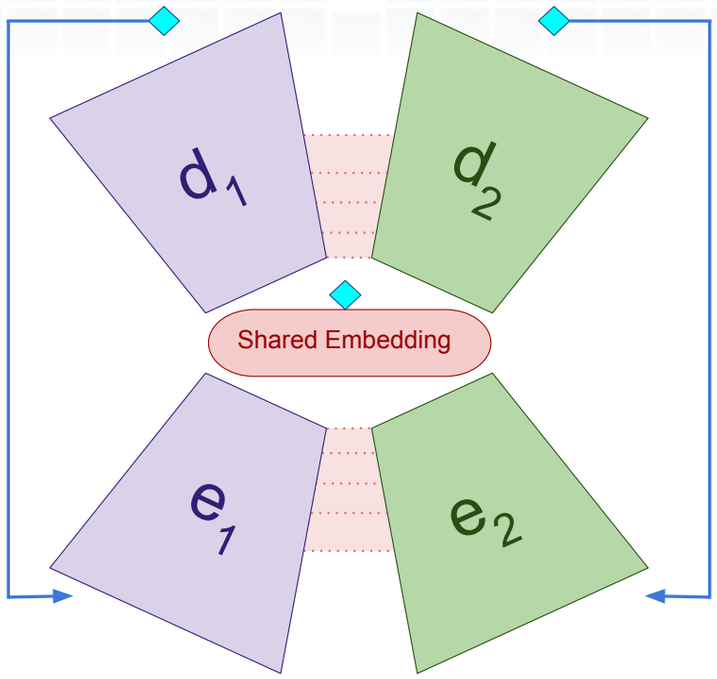
Application on Image-to-Image Translation



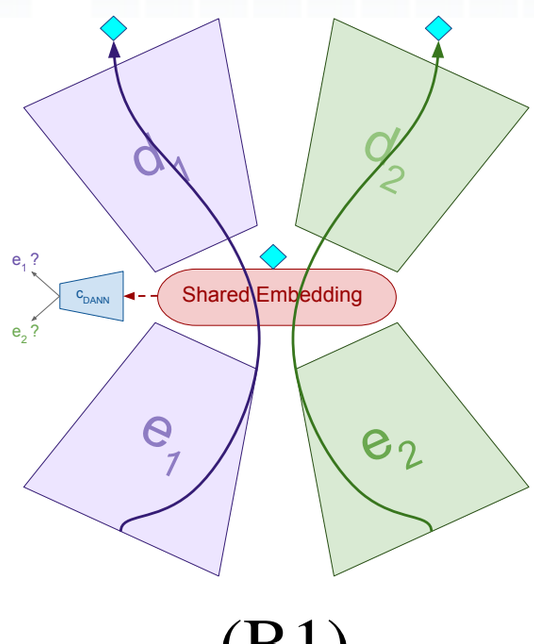


[Ref: Liu et al., UNIT, NIPS'17]

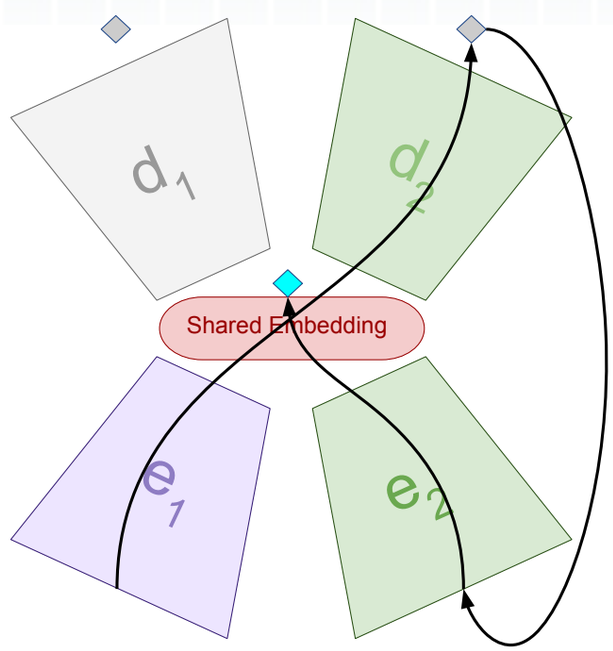
Application on Image-to-Image Translation



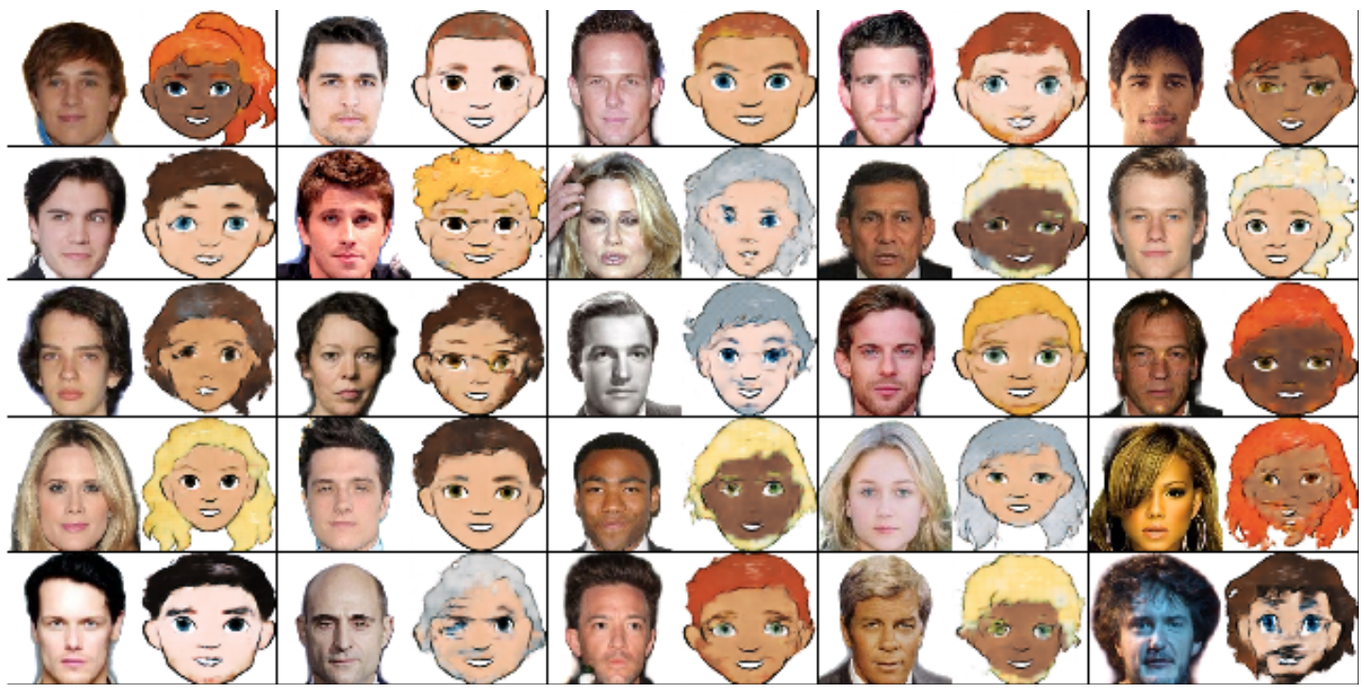
(A) High-level view of the XGAN architecture



(B1) Domain-adversarial autoencoder



(B2) Semantic consistency



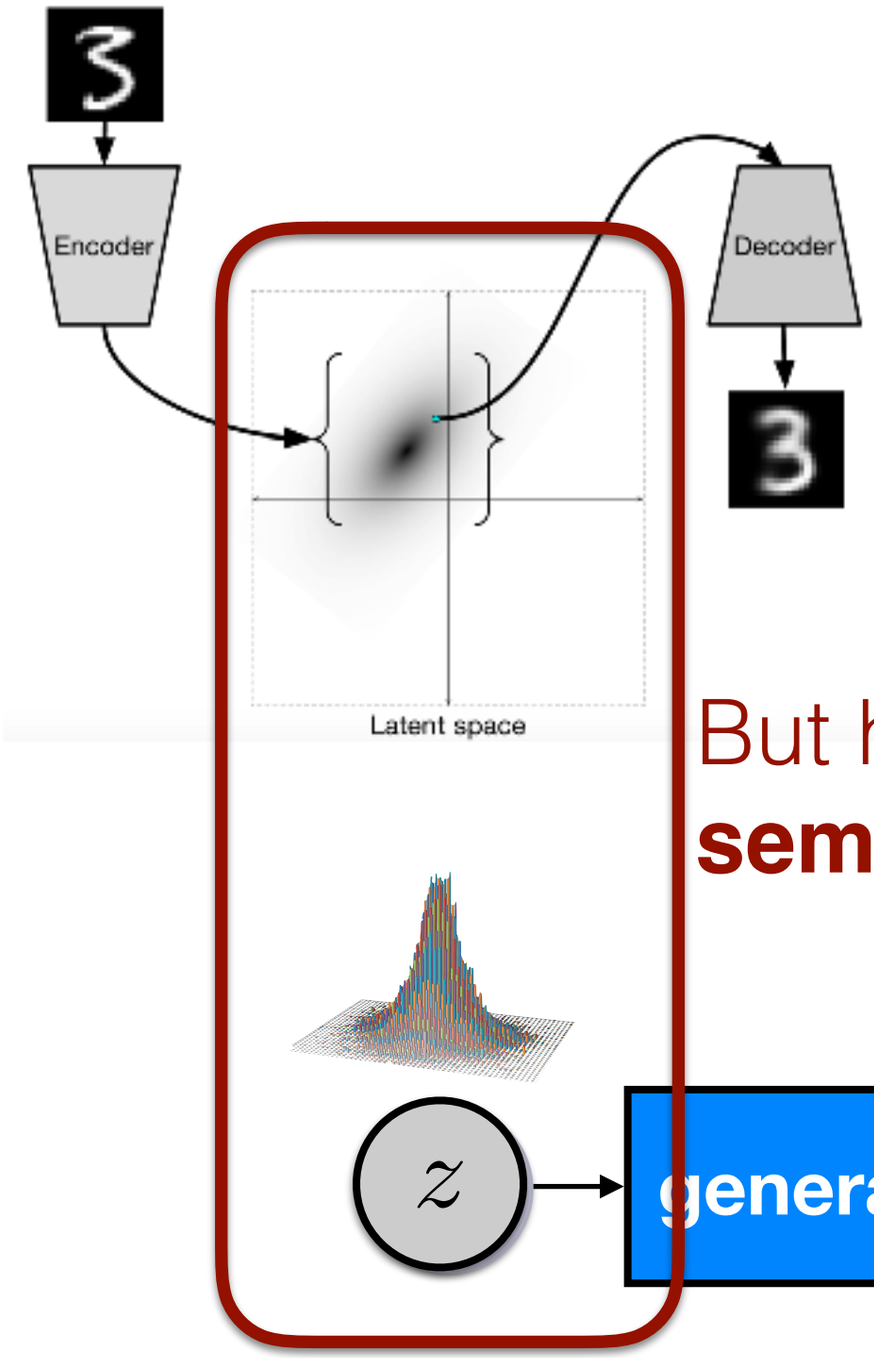
[Ref: Royer et al.,XGAN, ArXiv 1711.05139]

Outlines

- Discriminative versus Generative Models
- Going Into Deep Generative Models
- From Autoencoder to Variational Autoencoder (VAE)
- From VAE to Generative Adversarial Network (GAN)
- Various Applications
- **Understanding the latent space: disentanglement**

Disentangling Latent Space

- Yes, we are now able to generate some images…



But how do we understand the **semantics** of the latent space?

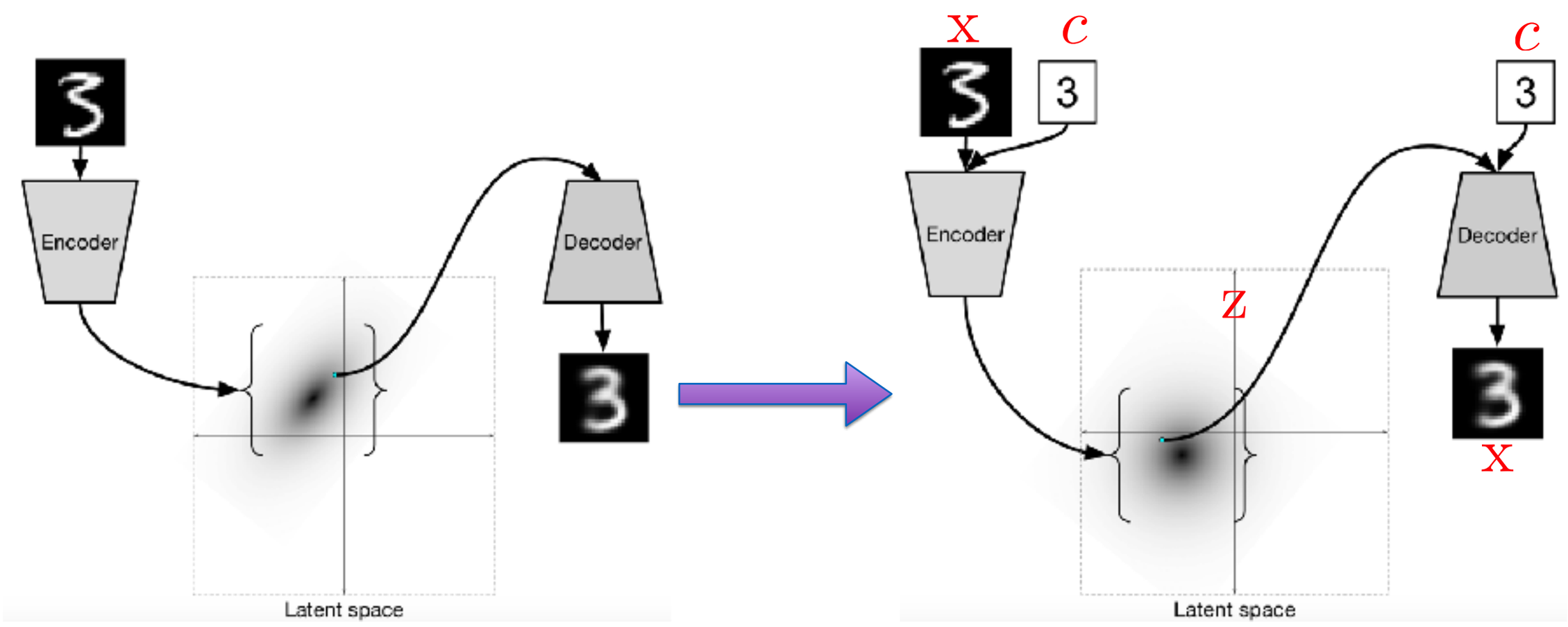
Disentangling Latent Space

- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)

- Different methods based on supervised or unsupervised learning as well as their base models: VAE, GAN, or mixed.

Disentangling Latent Space - VAE (Supervised)

- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)



conditional VAE

Disentangling Latent Space - VAE (Supervised)

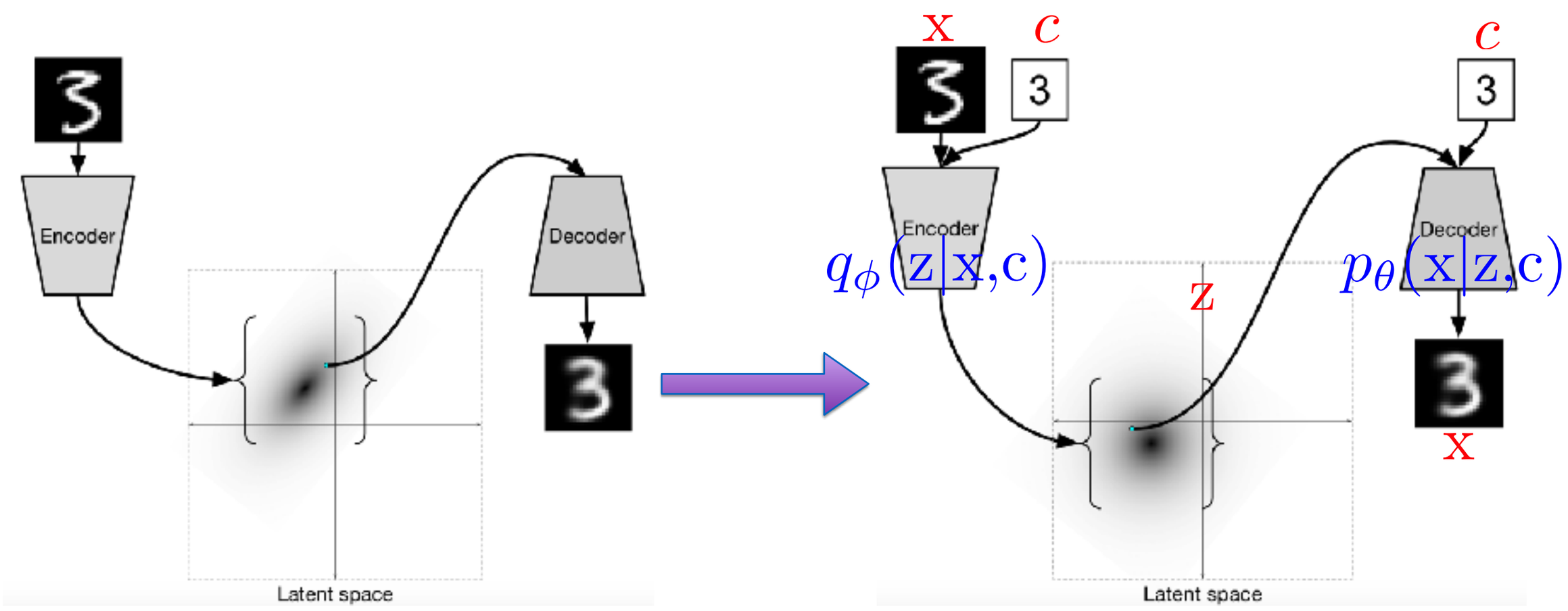
- **Conditional VAE**
  - Given training data $\mathbf{x}$ and corresponding labels $\mathbf{c}$, we would consider conditional distribution $p_\theta(\mathrm{x}|\mathrm{c})$

$$
\begin{aligned}
KL(q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})||p_\theta(\mathrm{z}|\mathrm{x},\mathrm{c})) &= \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})} \log \frac{q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})}{p_\theta(\mathrm{z}|\mathrm{x},\mathrm{c})} \\
&= \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})} \log \frac{q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})p_\theta(\mathrm{x}|\mathrm{c})}{p_\theta(\mathrm{z}|\mathrm{x},\mathrm{c})p_\theta(\mathrm{x}|\mathrm{c})} \\
&= \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})} \log \frac{q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})p_\theta(\mathrm{x}|\mathrm{c})}{p_\theta(\mathrm{x},\mathrm{z}|\mathrm{c})} \\
&= KL(q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})||p_\theta(\mathrm{x},\mathrm{z}|\mathrm{c})) + \log p_\theta(\mathrm{x}|\mathrm{c})) \\
\text{then } \log p_\theta(\mathrm{x}|\mathrm{c}) &= KL(q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})||p_\theta(\mathrm{z}|\mathrm{x},\mathrm{c})) + \mathcal{L}(\theta,\phi;\mathrm{x},\mathrm{c}) \\
\text{where } \mathcal{L}(\theta,\phi;\mathrm{x},\mathrm{c}) &= -KL(q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})||p_\theta(\mathrm{x},\mathrm{z}|\mathrm{c})) \\
&= \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})}[\log p_\theta(\mathrm{x},\mathrm{z}|\mathrm{c}) - \log q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})] \\
&= -KL(q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})||p_\theta(\mathrm{z}|\mathrm{c})) + \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})} \log p_\theta(\mathrm{x}|\mathrm{c},\mathrm{z})
\end{aligned}
$$

[https://zhuanlan.zhihu.com/p/25518643] **conditional VAE**

Disentangling Latent Space - VAE (Supervised)

- **Conditional VAE**
  - Given training data $\mathbf{x}$ and corresponding labels $\mathbf{c}$, we would consider conditional distribution $p_\theta(\mathrm{x}|\mathrm{c})$



impose prior **reconstruction**

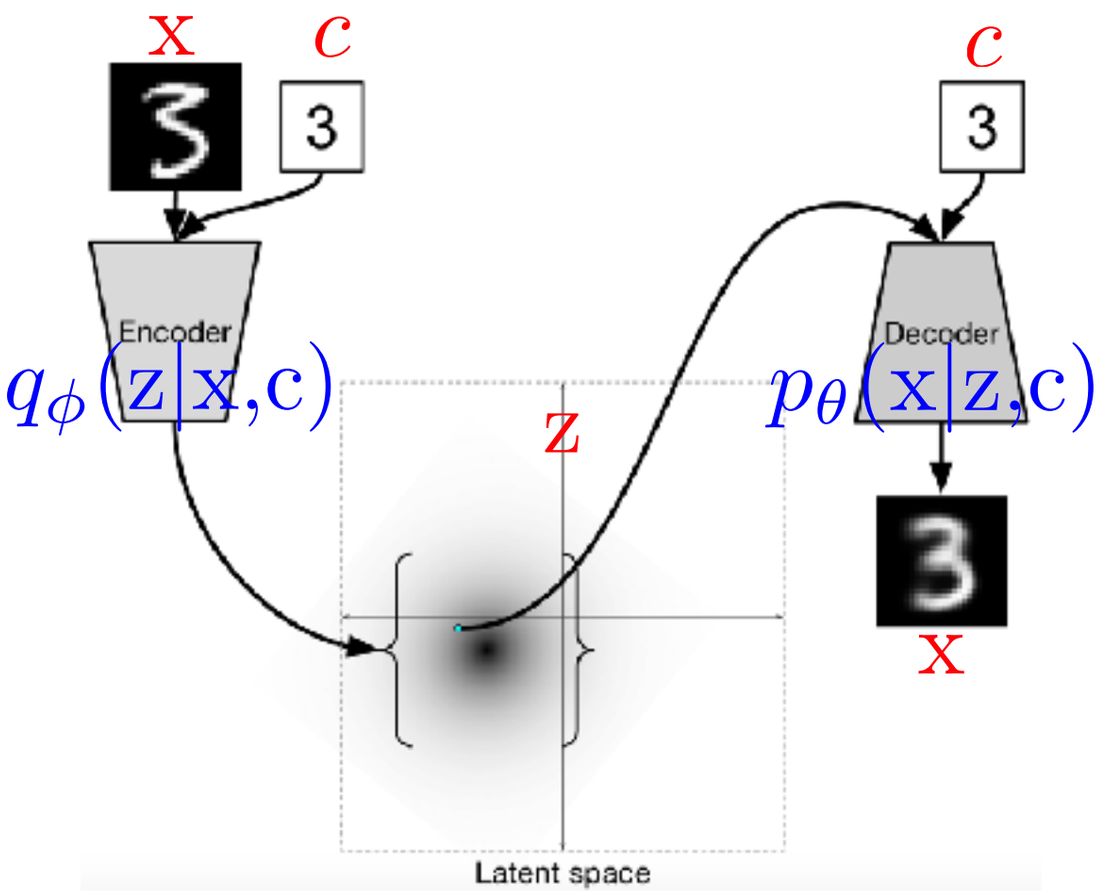
$$= -KL(q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})||p_\theta(\mathrm{z}|\mathrm{c})) + \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})} \log p_\theta(\mathrm{x}|\mathrm{c},\mathrm{z})$$

[https://zhuanlan.zhihu.com/p/25518643] **conditional VAE**

Disentangling Latent Space - VAE (Supervised)

- **Conditional VAE**
  - Given training data $\mathbf{x}$ and corresponding labels $\mathbf{c}$, we would consider conditional distribution $p_\theta(\mathrm{x}|\mathrm{c})$

I prefer to imagine that the encoder cleans up factor related to labels. z now is invariant to c, e.g. stroke width or angle

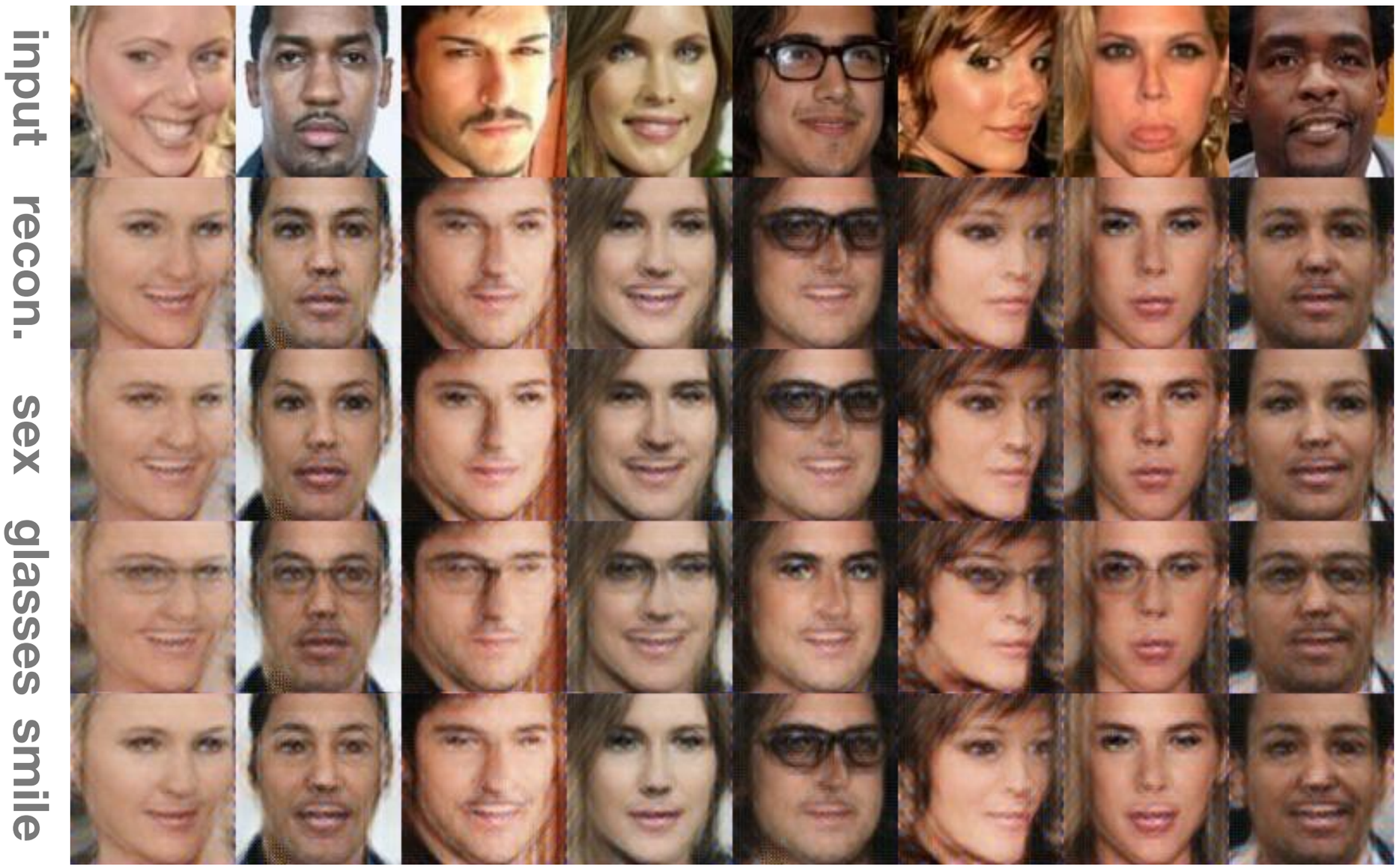


$$= \underbrace{-KL(q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})||p_\theta(\mathrm{z}|\mathrm{c}))}_{\text{impose prior}} + \underbrace{\mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x},\mathrm{c})} \log p_\theta(\mathrm{x}|\mathrm{c},\mathrm{z})}_{\text{reconstruction}}$$

[https://zhuanlan.zhihu.com/p/25518643] conditional VAE

Disentangling Latent Space - VAE (Supervised)

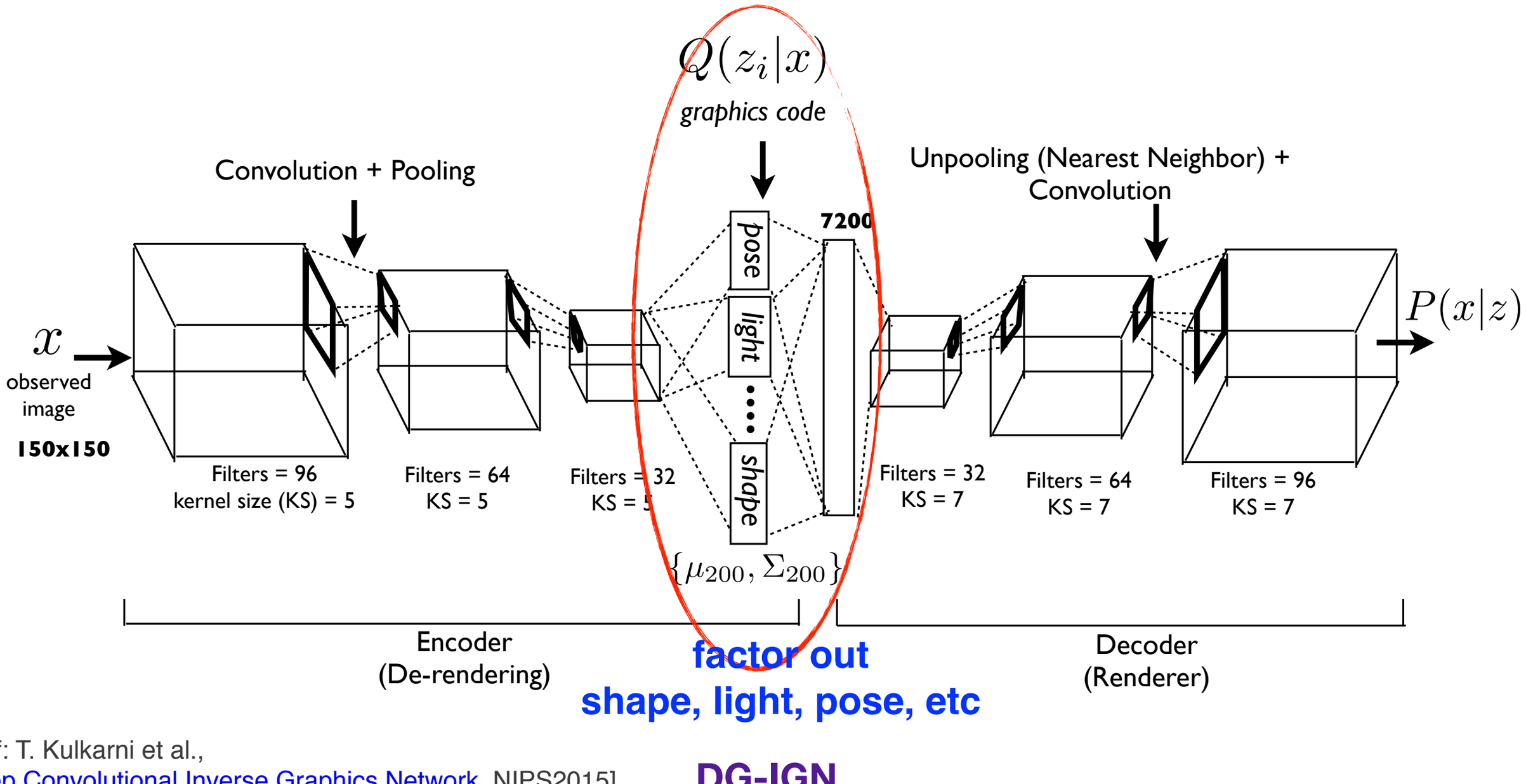
- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)



conditional VAE

Disentangling Latent Space - VAE (Supervised)

- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)



factor out
shape, light, pose, etc

[Ref: T. Kulkarni et al.,
Deep Convolutional Inverse Graphics Network, NIPS2015]

DG-IGN

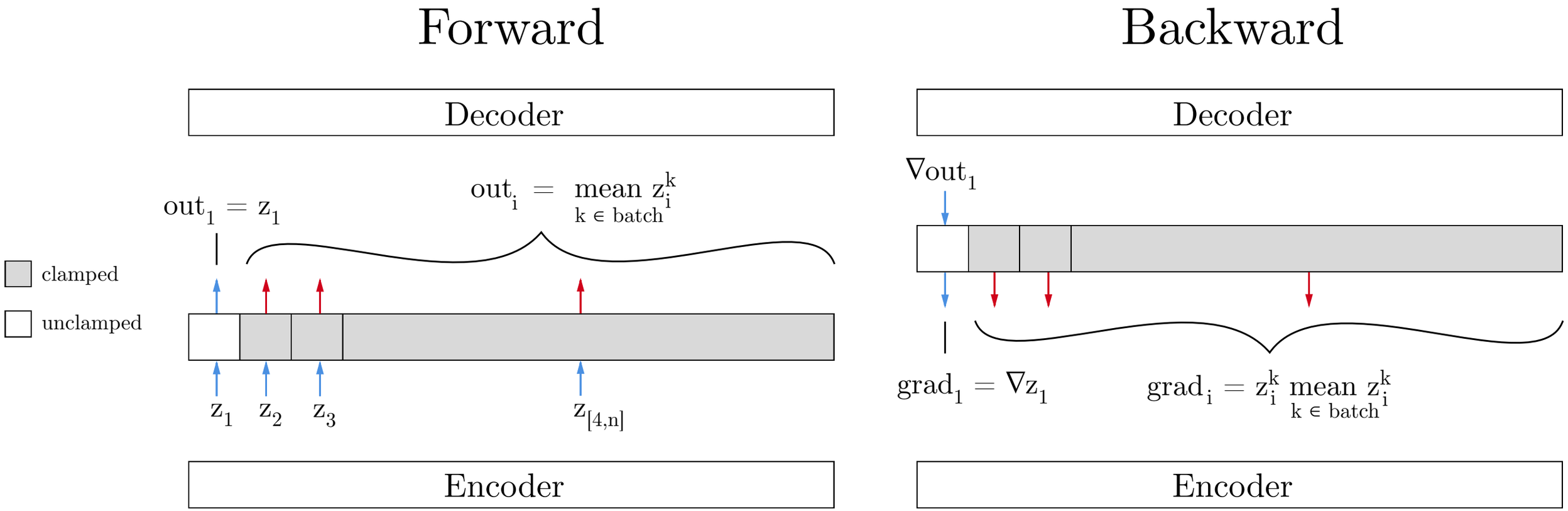
Disentangling Latent Space - VAE (Supervised)

- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)

$$\mathrm{Z} = \boxed{\mathbf{z_1} \mid \mathbf{z_2} \mid \mathbf{z_3} \mid \mathbf{z_{[4,n]}}}$$

$$\text{corresponds to} \quad \phi \quad \alpha \quad \phi_{\mathrm{L}} \quad \text{intrinsic properties (shape, texture, etc)}$$

training on a minibatch in which only ϕ, the azimuth angle, changes:

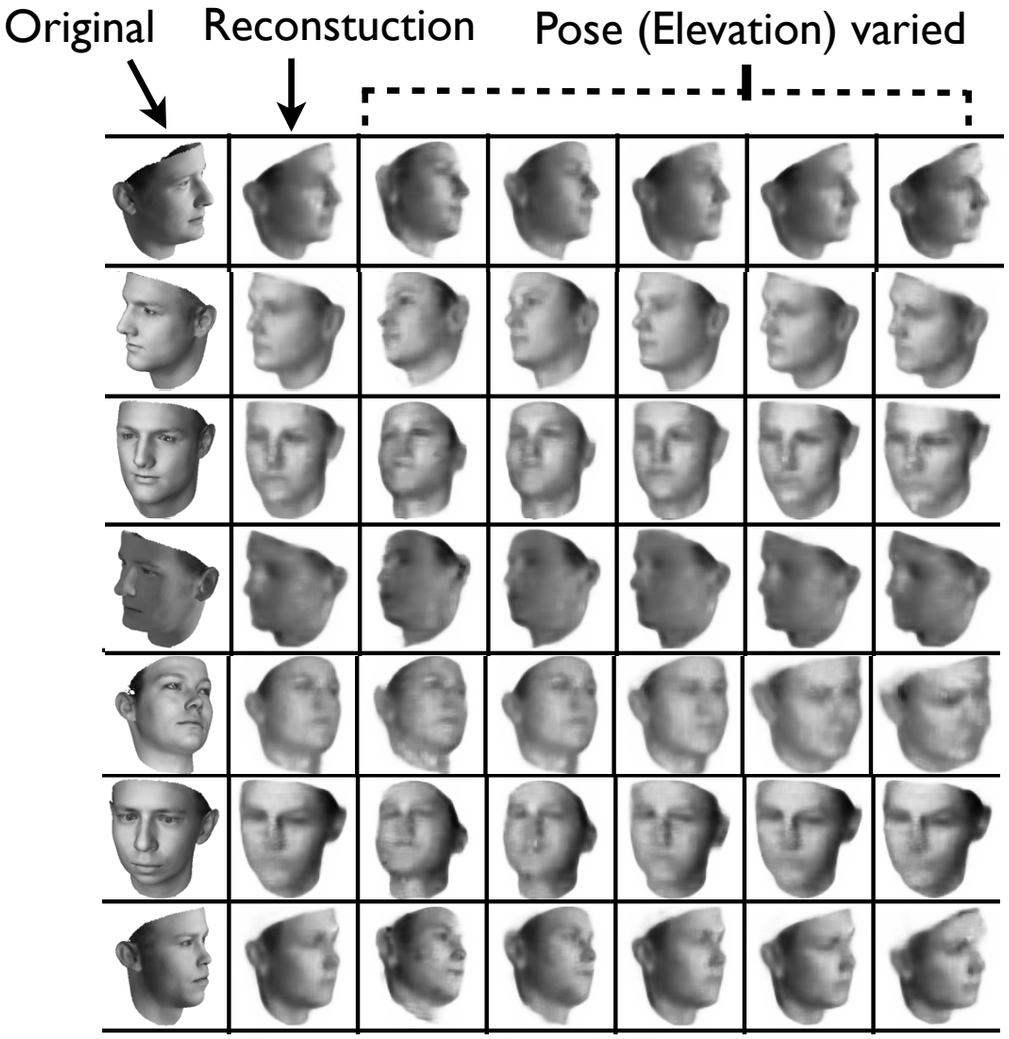


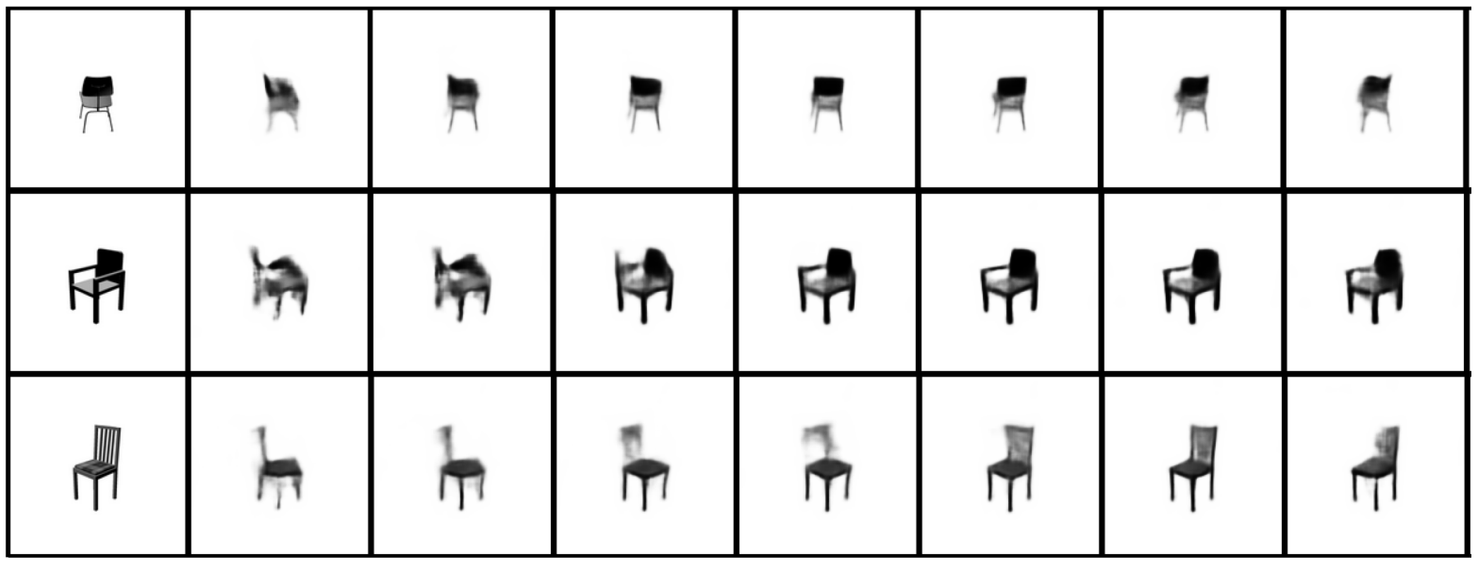
[Ref: T. Kulkarni et al., Deep Convolutional Inverse Graphics Network, NIPS2015]

DG-IGN **heavy data consumption**

Disentangling Latent Space - VAE (Supervised)

- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)



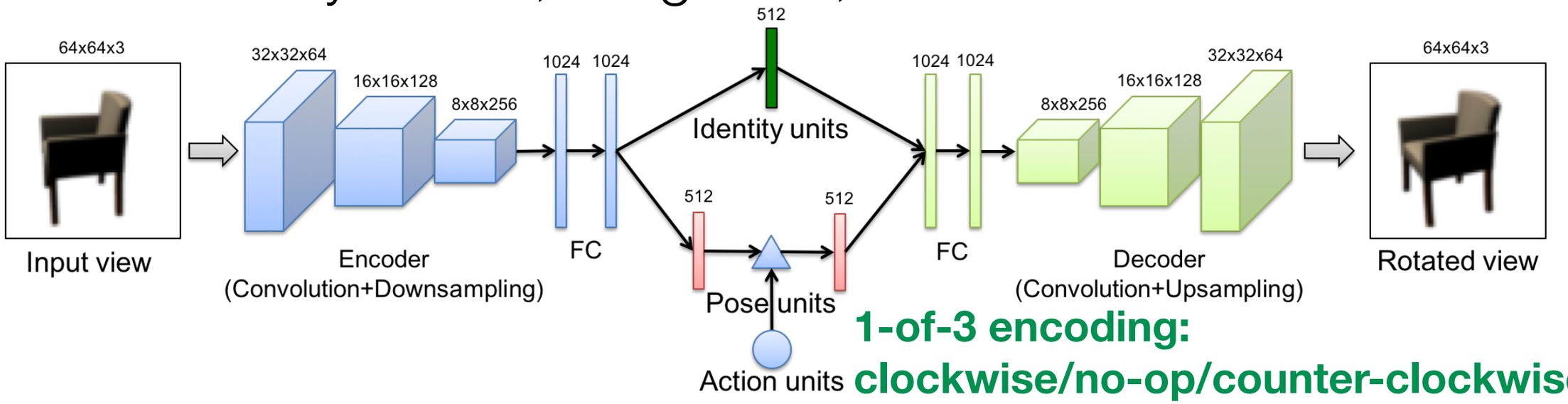


[Ref: T. Kulkarni et al.,
Deep Convolutional Inverse Graphics Network, NIPS2015]

DG-IGN

Disentangling Latent Space - VAE (Supervised)

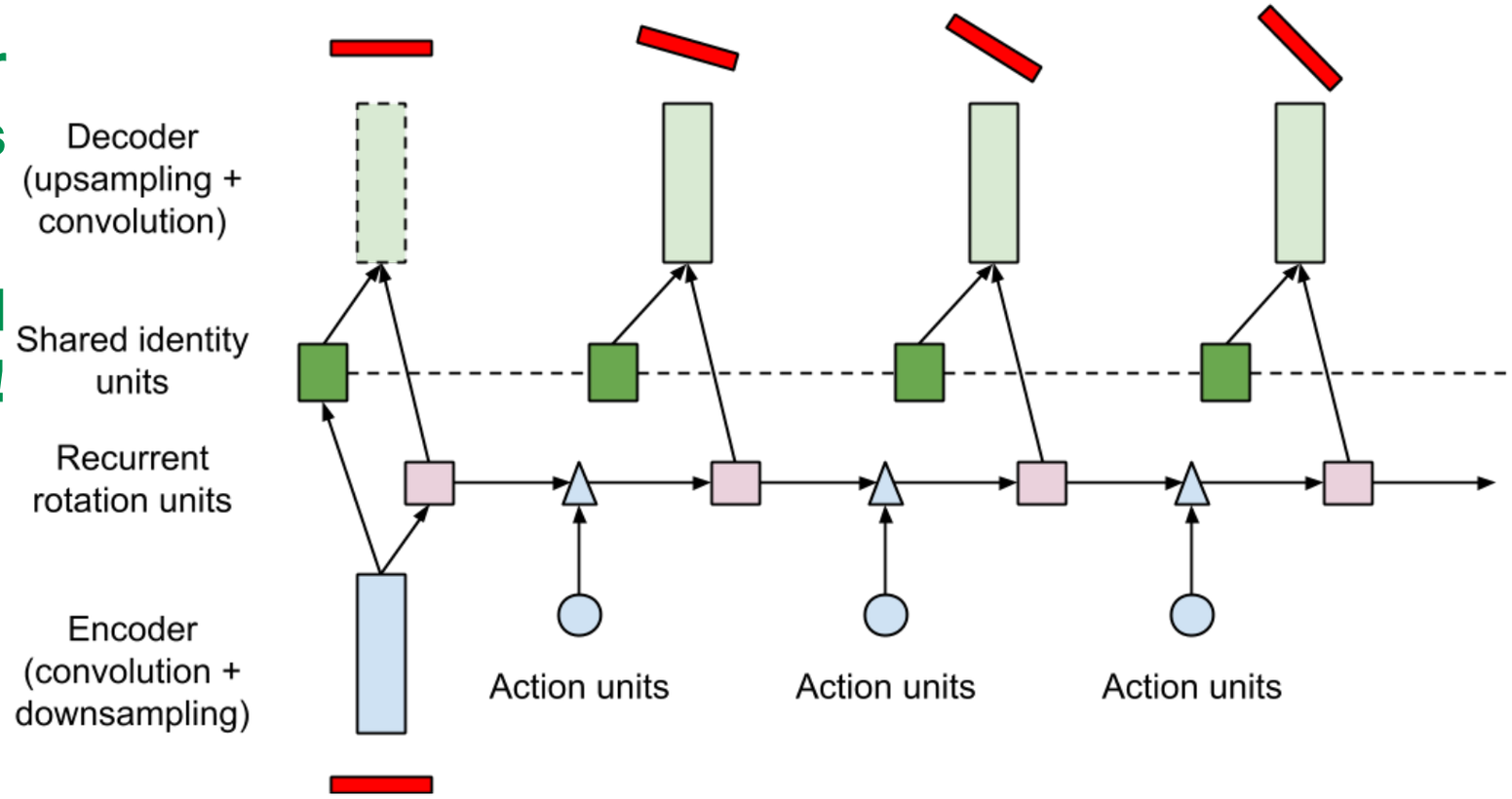
- Weakly-supervised Disentangling with Recurrent Transformations for 3D View Synthesis, Yang et al., NIPS'15.



1-of-3 encoding: clockwise/no-op/counter-clockwise

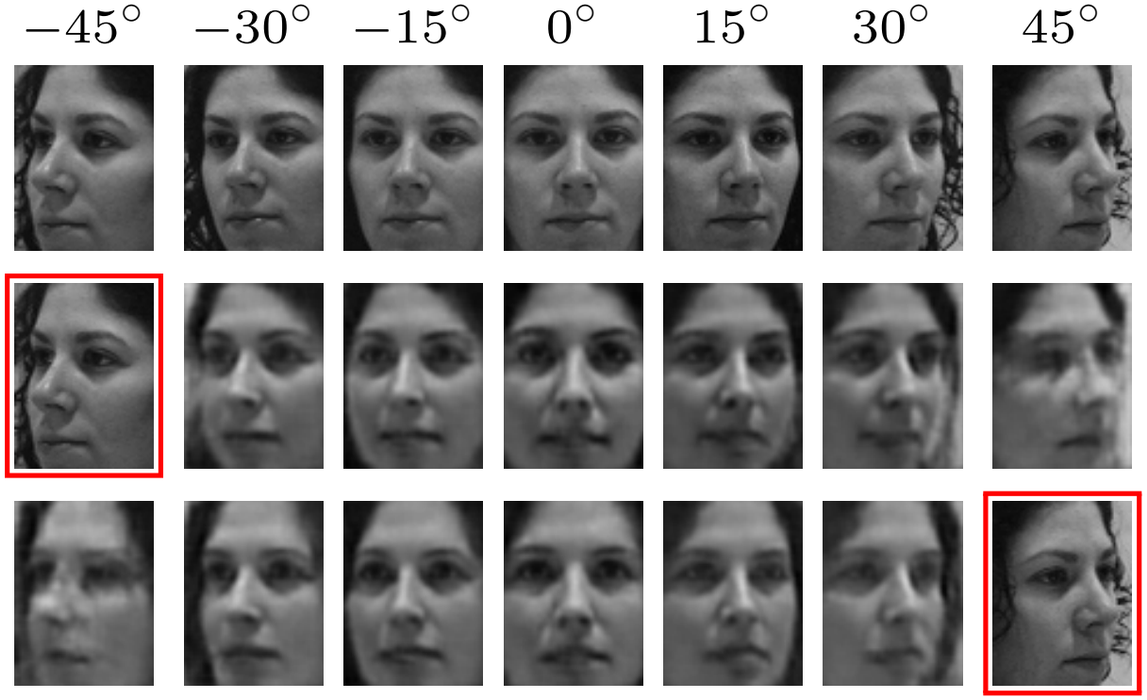
reconstruction error evaluated on all frames

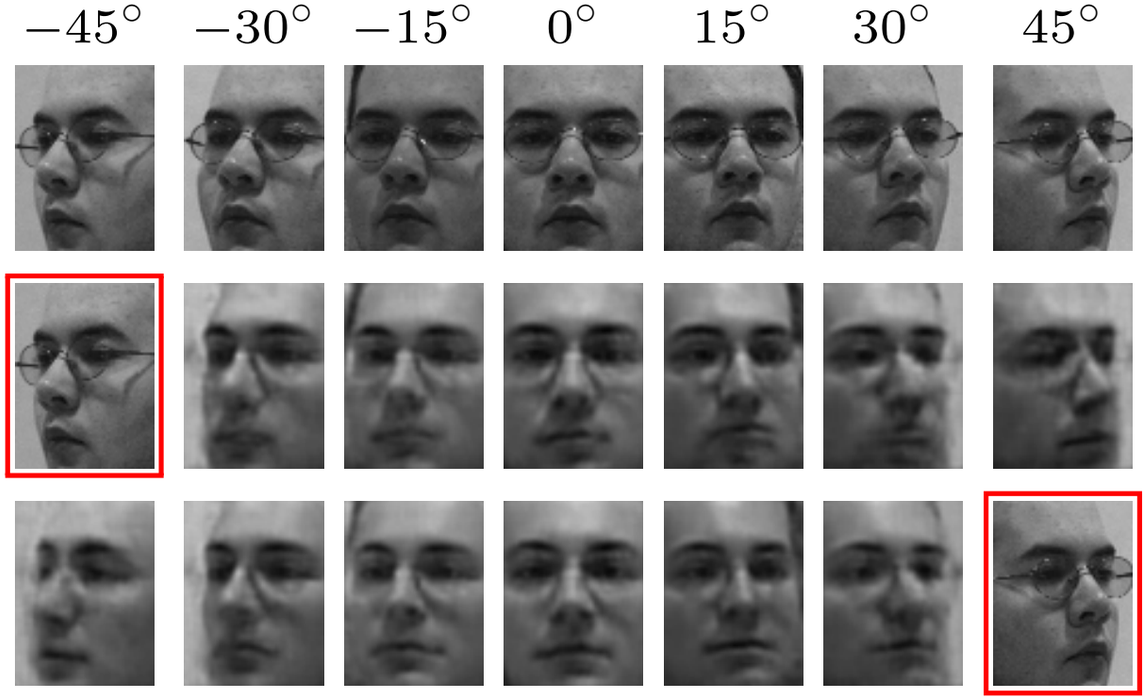
same id all the time!



Disentangling Latent Space - VAE (Supervised)

- Weakly-supervised Disentangling with Recurrent Transformations for 3D View Synthesis, Yang et al., NIPS'15.



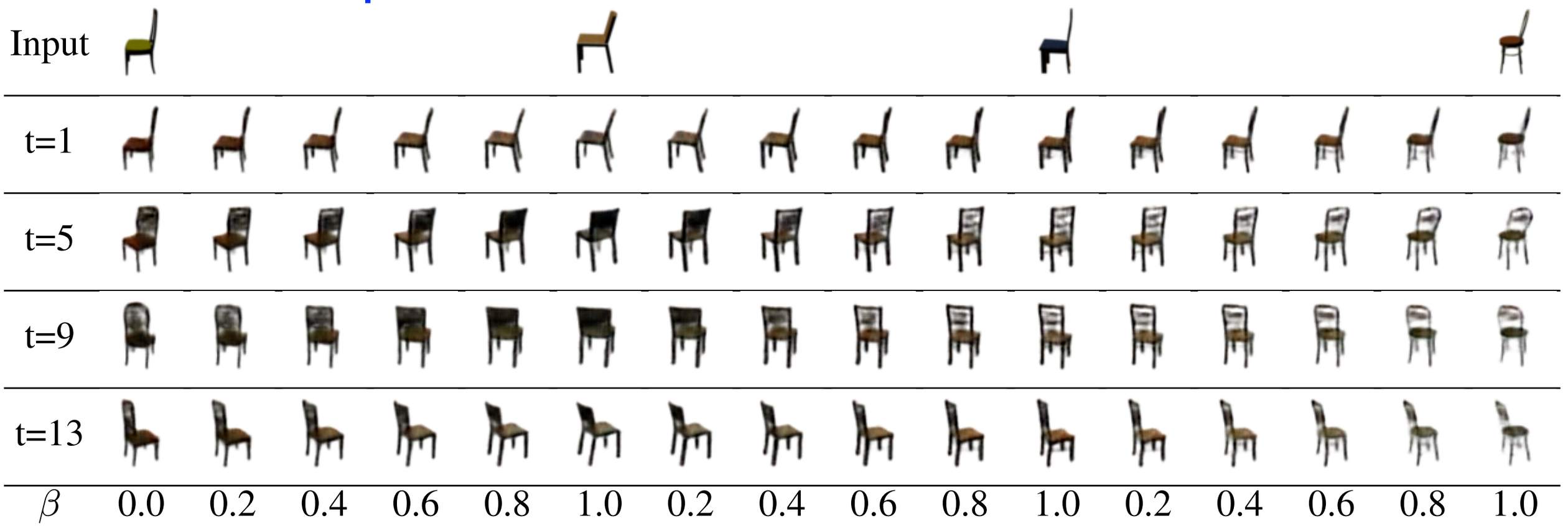


資訊工程學系
Department of Computer Science

Disentangling Latent Space - VAE (Supervised)

- Weakly-supervised Disentangling with Recurrent Transformations for 3D View Synthesis, Yang et al., NIPS'15.

interpolation between different chairs as well as rotation



Disentangling Latent Space - VAE (Unsupervised)

- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)

remember VAE story? **reconstruction** **impose prior**

$$\max_{\phi,\theta} \mathbb{E}_{x\sim\mathbf{D}}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]\right] \quad \text{subject to} \quad D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) < \epsilon$$

$$\mathcal{F}(\theta,\phi,\beta;\mathbf{x},\mathbf{z}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta(D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) - \epsilon)$$ **rewrite by Lagrangian**

bigger β limits the capacity of z more,
together with maximising likelihood of data,
it ends up with learning more efficient representation

[Ref: Higgins et al.,
β-VAE: LEARNING BASIC VISUAL CONCEPTS WITH A CONSTRAINED VARIATIONAL FRAMEWORK, ICLR2017]

β-VAE

Disentangling Latent Space - VAE (Unsupervised)

- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)

need to eyeball the disentangled results in order to know which variable is being factored out

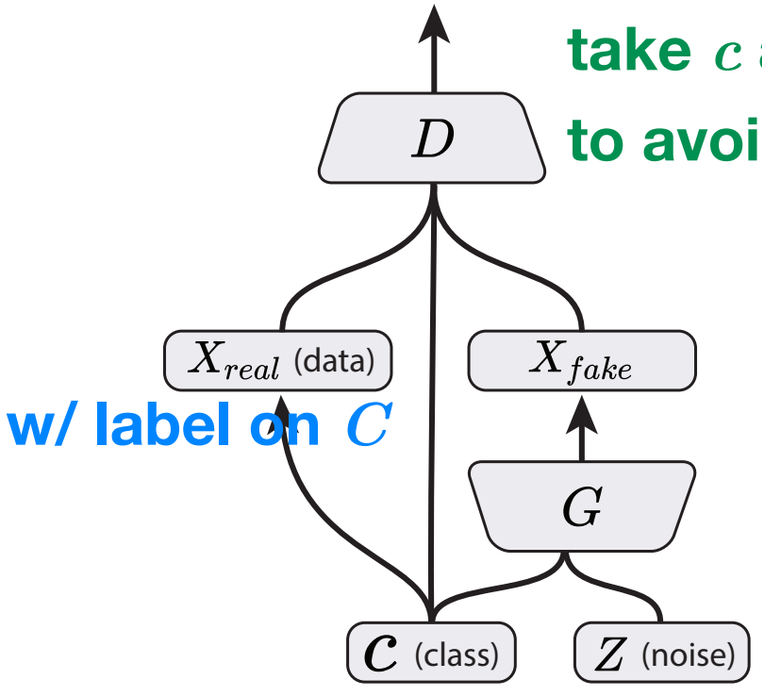
β-VAE

Disentangling Latent Space - GAN (Supervised)

- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

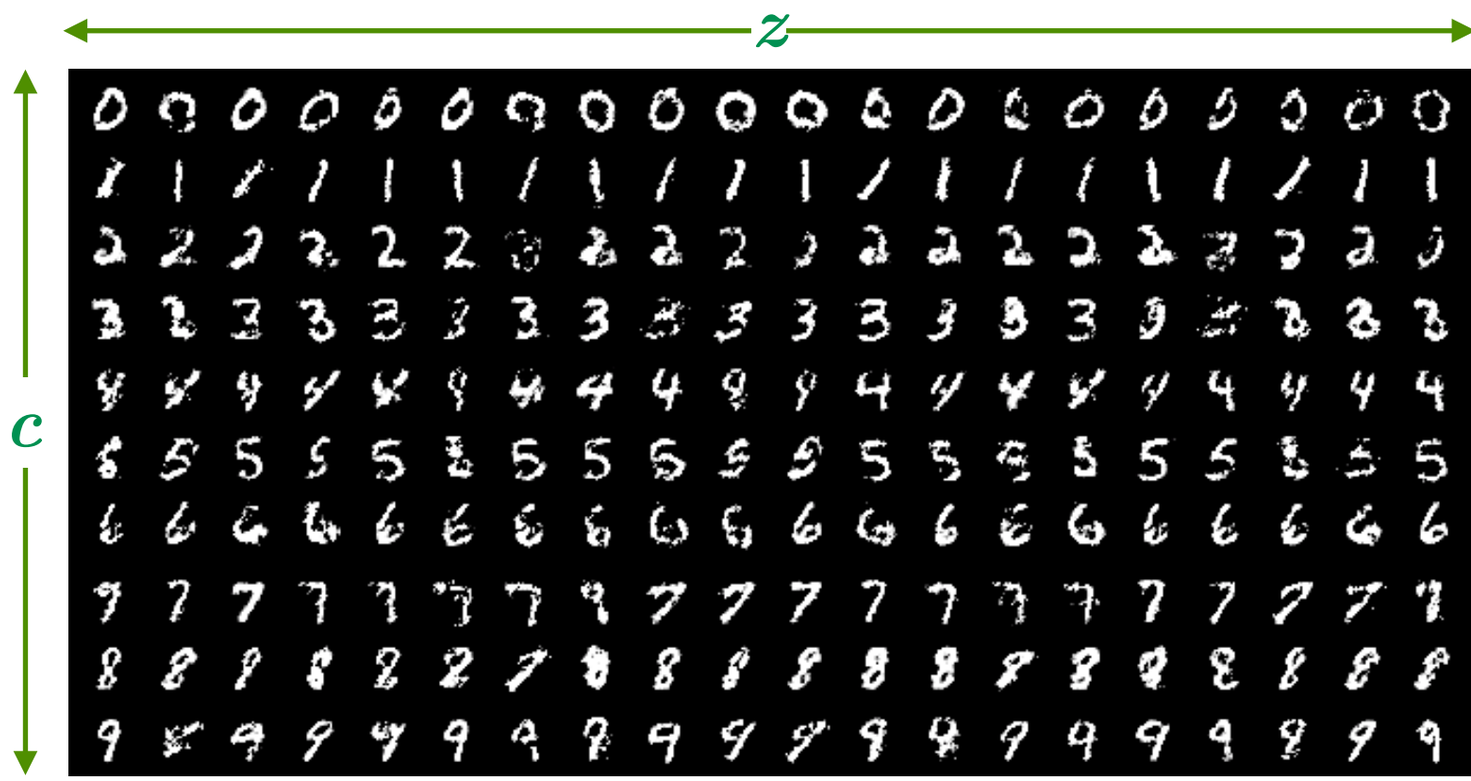
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))]$$



take c as also input for D

to avoid ignoring c in G

w/ label on C

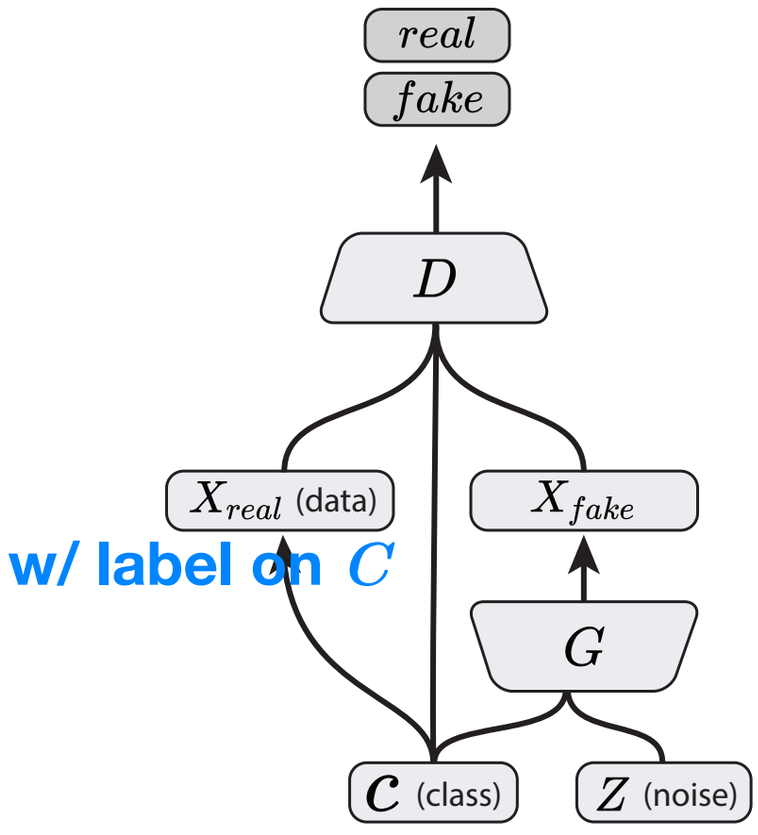


conditional GAN

[Ref: Mirza et al., Conditional generative adversarial nets, CoRR2014]

Disentangling Latent Space - GAN (Supervised)

- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)



w/ label on C

User tags + annotations	Generated tags
montanha, trem, inverno, frio, people, male, plant life, tree, structures, transport, car	taxi, passenger, line, transportation, railway station, passengers, railways, signals, rail, rails
food, raspberry, delicious, homemade	chicken, fattening, cooked, peanut, cream, cookie, house made, bread, biscuit, bakes
water, river	creek, lake, along, near, river, rocky, treeline, valley, woods, waters

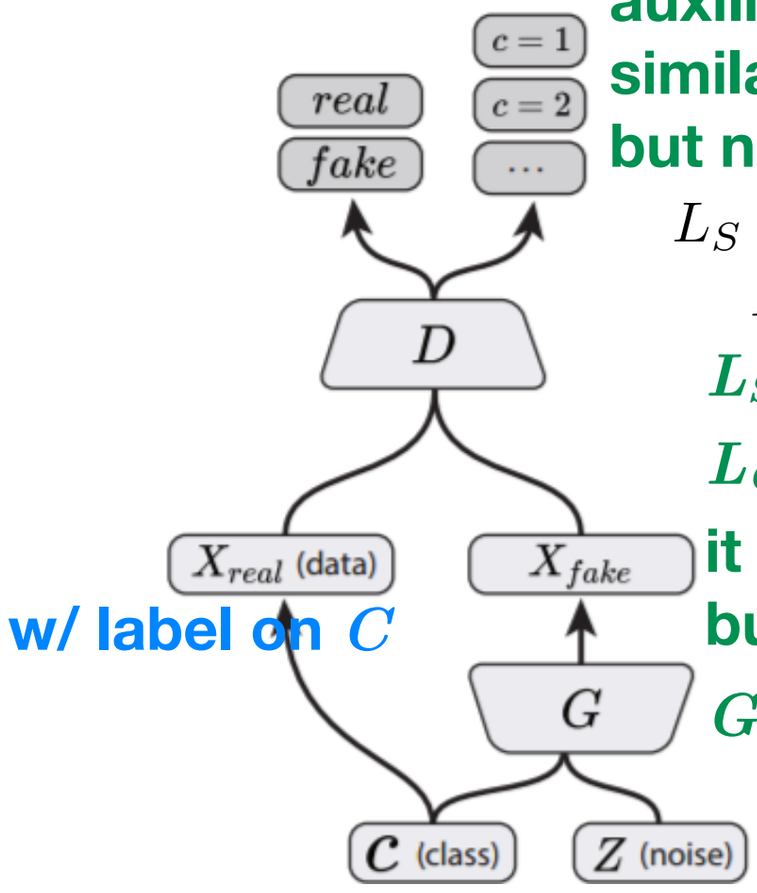
condition on images, generate text tags

[Ref: Mirza et al.,
Conditional generative adversarial nets, CoRR2014]

conditional GAN

Disentangling Latent Space - GAN (Supervised)

- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)



w/ label on C

auxiliary classifier:
similar to the idea of conditional GAN
but now we use the power of multi-task learning

$$L_S = E[\log P(S = real \mid X_{real})] + E[\log P(S = fake \mid X_{fake})]$$
$$L_C = E[\log P(C = c \mid X_{real})] + E[\log P(C = c \mid X_{fake})]$$

L_S as in original GAN, make synthesize data realistic;
L_C as class/attribute classifier, can be imagined that it not only needs to predict class/attribute accurately, but has similar idea as deep-dream: Given c, modify $G(z,\ c)$ to make it more likely to be classified as c

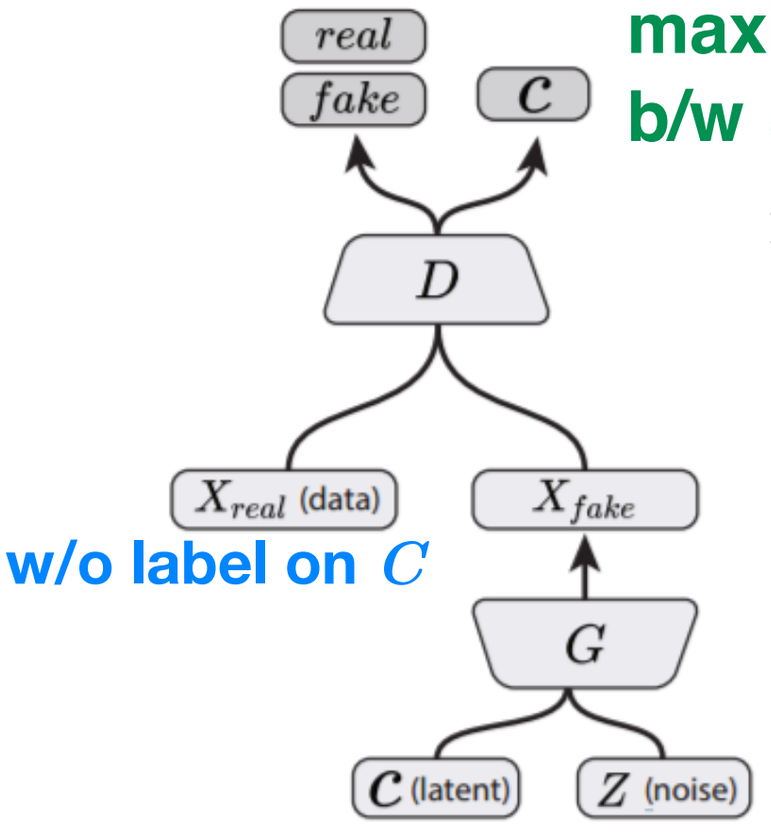
switching smile

AC-GAN

[Ref: Odena et al., CONDITIONAL IMAGE SYNTHESIS WITH AUXILIARY CLASSIFIER GANS, ArXiv2017]

Disentangling Latent Space - GAN (Unsupervised)

- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)



w/o label on C

maximize mutual information $I(c;\ G(z,\ c))$ b/w c and $G(z,\ c)$, otherwise G might ignore c

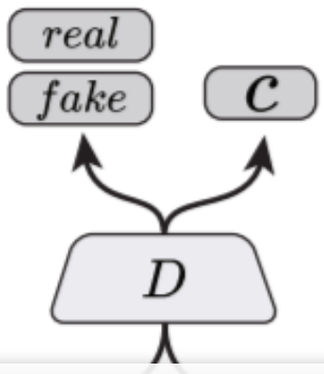
$$\min_G \max_D V_I(D,G) = V(D,G) - \lambda I(c;G(z,c))$$

[Ref: Chen et al.,
InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, NIPS2016]

Info-GAN

Disentangling Latent Space - GAN (Unsupervised)

- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)



maximize mutual information $I(c;\ G(z,\ c))$ b/w c and $G(z,\ c)$, otherwise G might ignore c

$$\min_G \max_D V_I(D,G) = V(D,G) - \lambda I(c;G(z,c))$$

w/o label on c

$$I(c;G(z,c)) = H(c) - H(c|G(z,c))$$
$$= E_{x\sim G(z,c)}\left[E_{c'\sim P(c|x)}\left[logP(c'|x)\right]\right] + H(c)$$
$$= E_{x\sim G(z,c)}\left[KL(P(\cdot|x)\parallel Q(\cdot|x)) + E_{c'\sim P(c|x)}\left[logQ(c'|x)\right]\right] + H(c)$$
$$\geq E_{x\sim G(z,c)}\left[E_{c'\sim P(c|x)}\left[logQ(c'|x)\right]\right] + H(c)$$

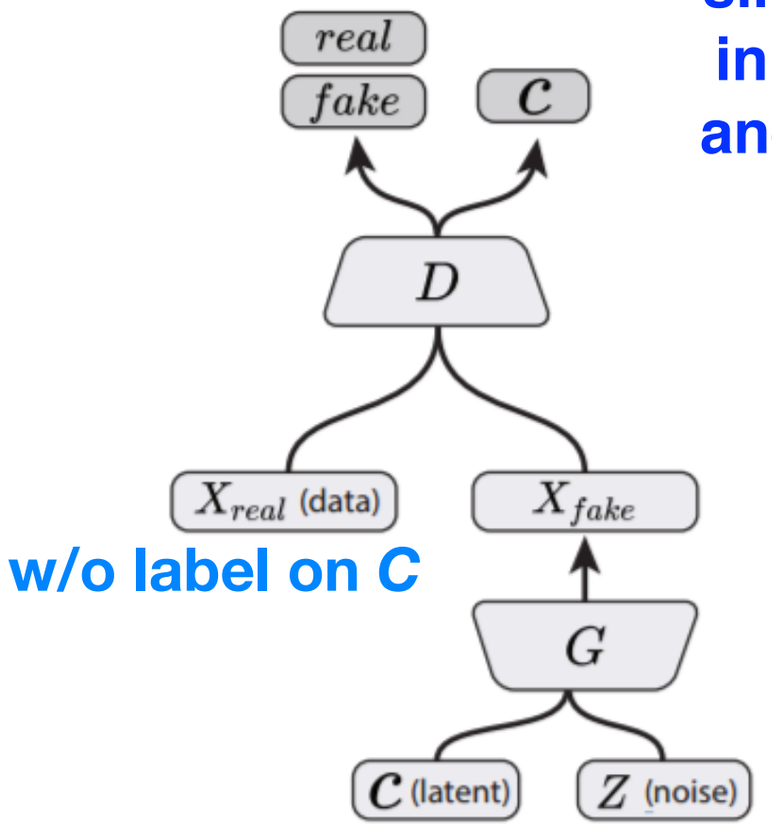
generator **c classifier**

[Ref: Chen et al.,
InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, NIPS2016]

Info-GAN

Disentangling Latent Space - GAN (Unsupervised)

- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)



w/o label on C

similar to β-VAE, need to eyeball disentangled results in order to know which variable is being factored out and the disentangled factor might vary along iteration

male/female

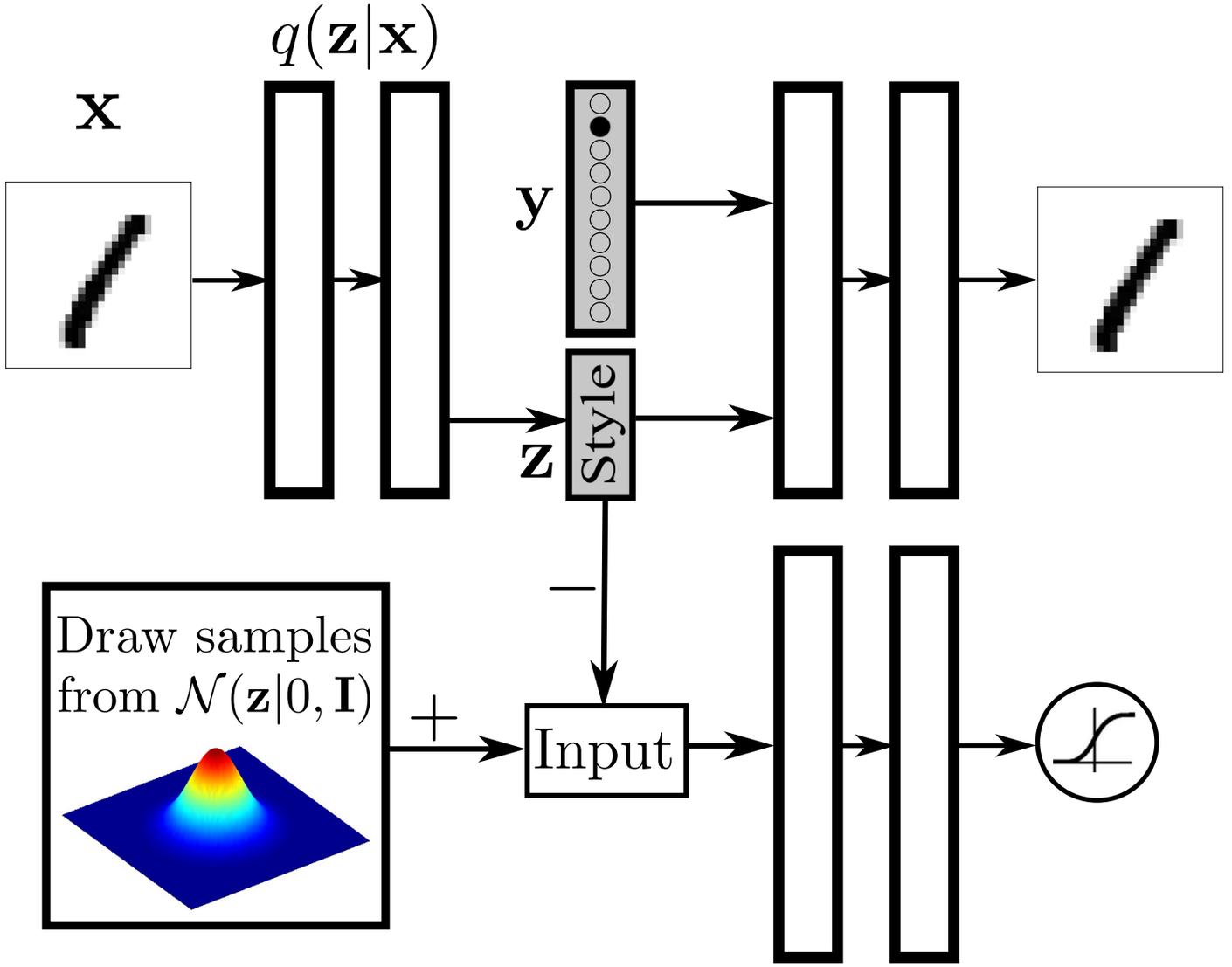
pose

[Ref: Chen et al.,
InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, NIPS2016]

Info-GAN

Disentangling Latent Space - AAE (Supervised)

- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)



AAE

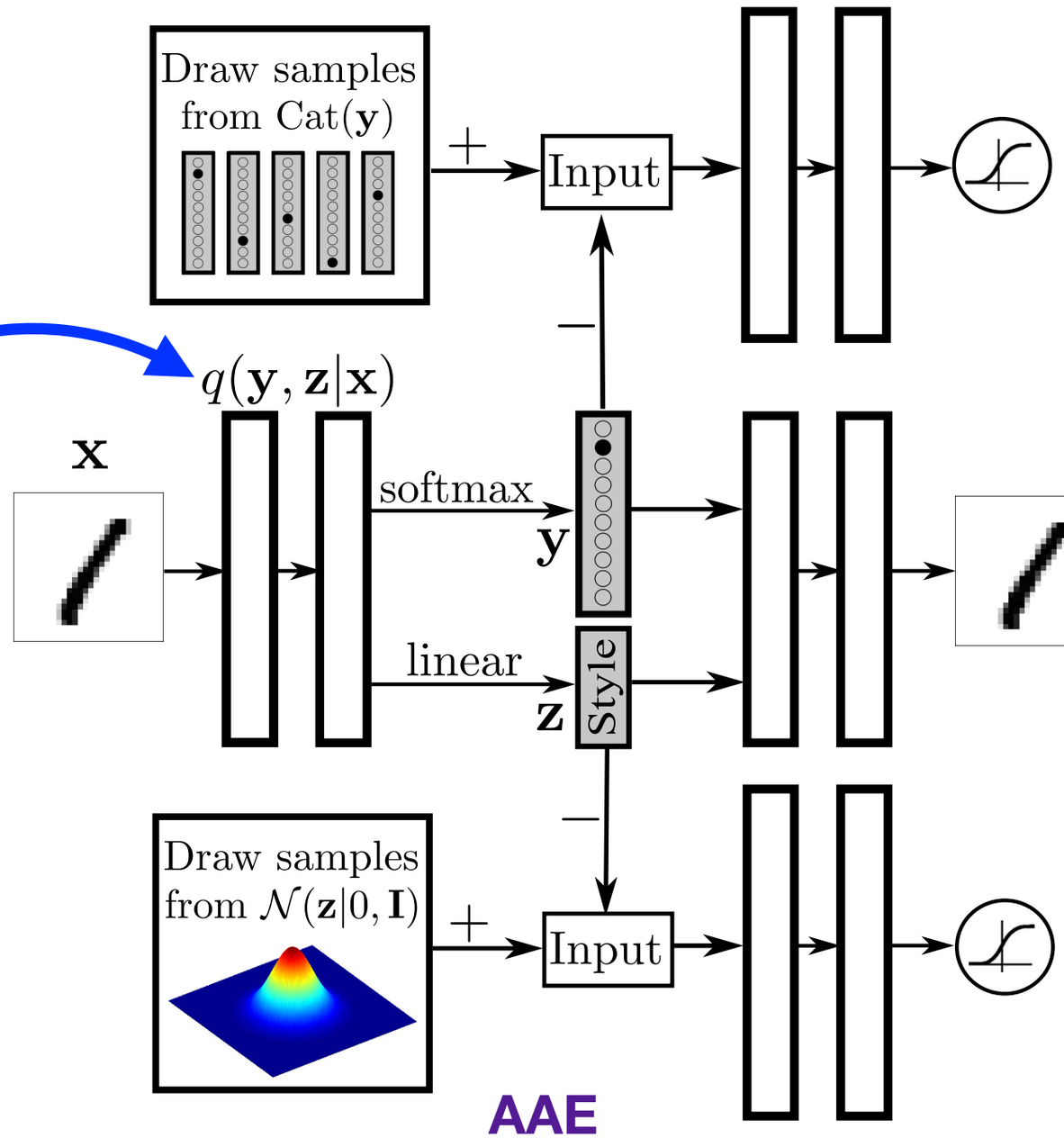
Disentangling Latent Space - AAE (Semi-Supervised)

- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)

label prediction based on encoder $q(\mathbf{y}|\mathbf{x})$ for unlabeled data should follow $\mathrm{Cat}(\mathbf{y})$

$q(\mathbf{y}|\mathbf{x})$ is trained on labeled data

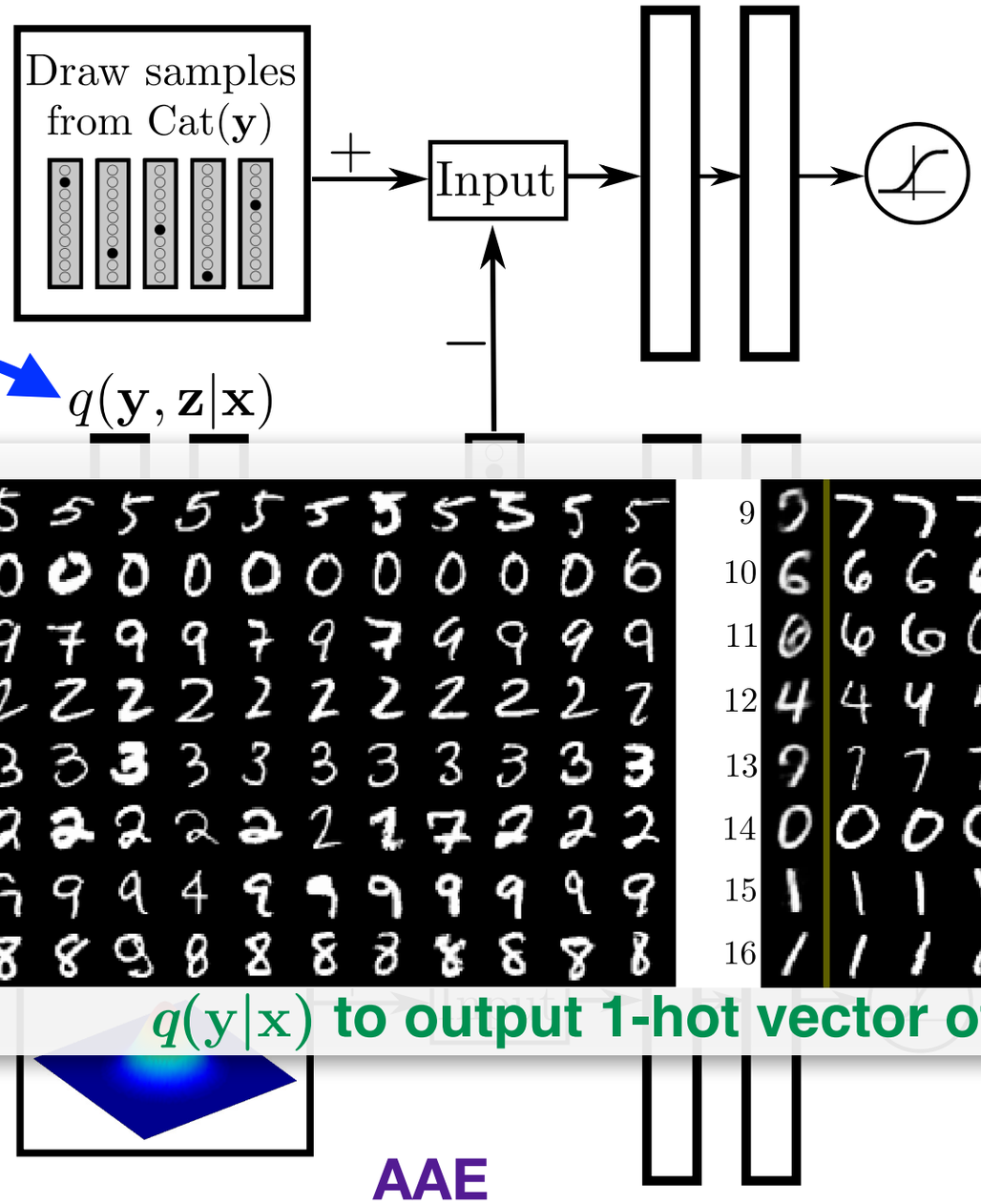
By removing this part, can make the model to be unsupervised. We can ask $q(\mathbf{y}|\mathbf{x})$ to output 1-hot vector.



AAE

Disentangling Latent Space - AAE (Unsupervised)

- Decompose the latent space into two parts, **disentanglement**:
  - interpretable latent variables c
  - uninterpretable latent variables z (c-invariant)



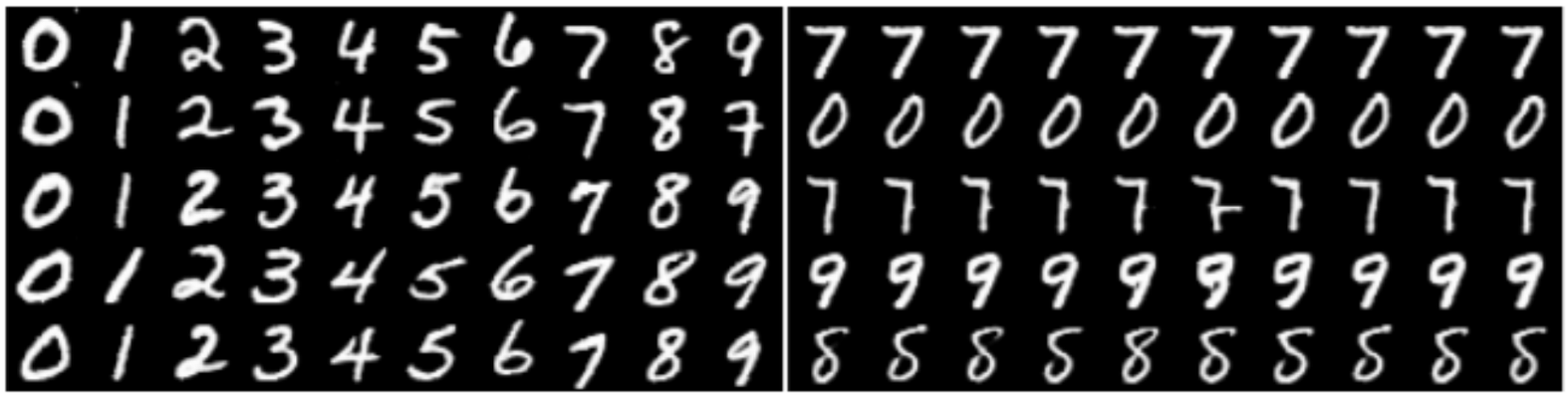
label prediction based on encoder $q(\mathbf{y}|\mathbf{x})$ for unlabeled data should follow $\mathrm{Cat}(\mathbf{y})$

$q(\mathbf{y}|\mathbf{x})$ is trained on labeled data

By removing this part, can make the model to be unsupervised. We can ask $q(\mathbf{y}|\mathbf{x})$ to output 1-hot vector.

$q(\mathbf{y}|\mathbf{x})$ to output 1-hot vector of length 16

Disentangling Latent Space - More Examples

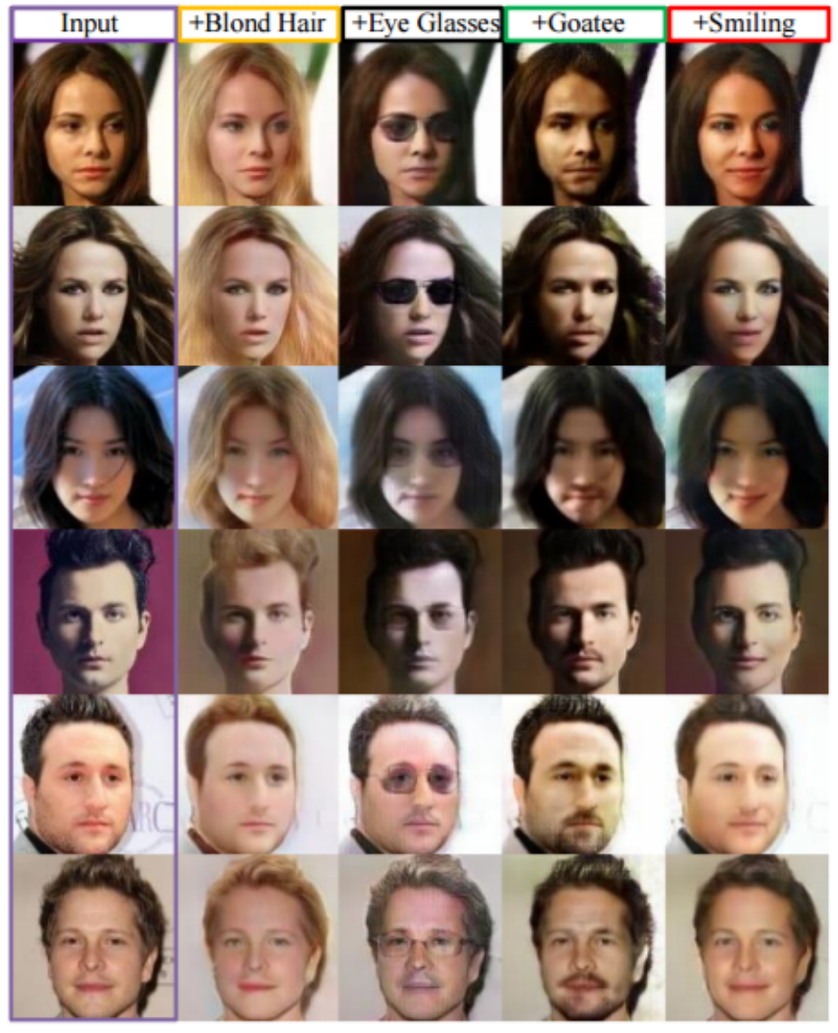


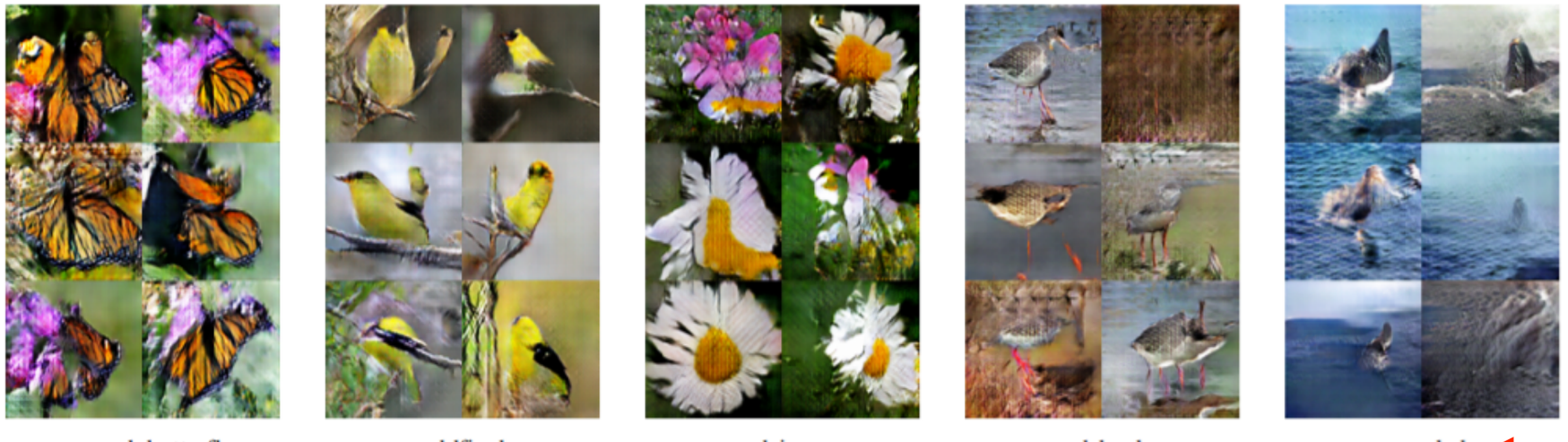
(a) Varying c_1 on InfoGAN (Digit type)

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

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

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

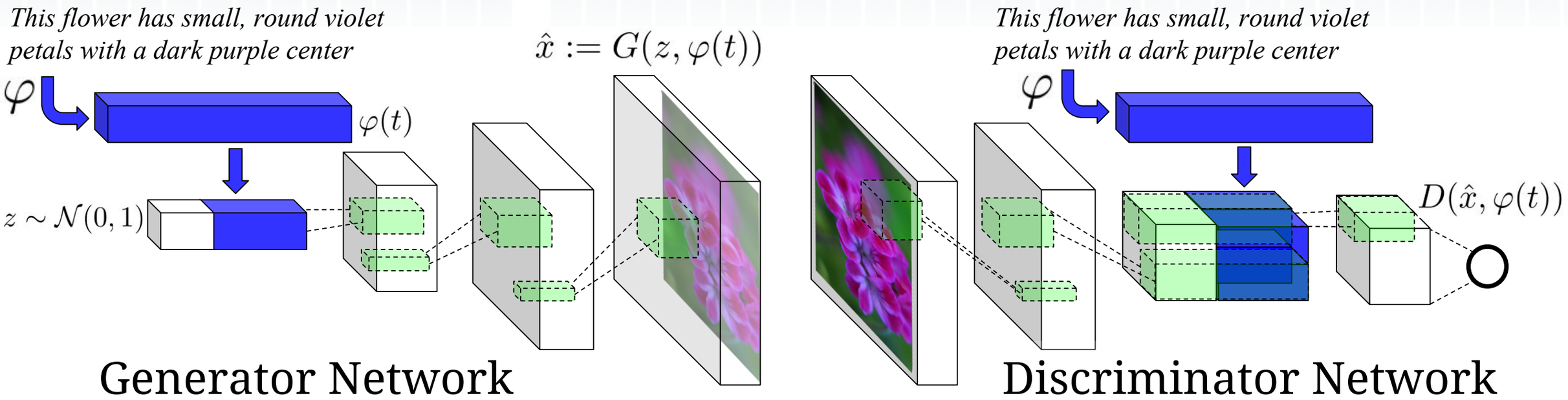




monarch butterfly goldfinch daisy redshank grey whale ← c

Disentangling Latent Space - Application on Text-to-Image Translation

資訊工程學系
Department of Computer Science

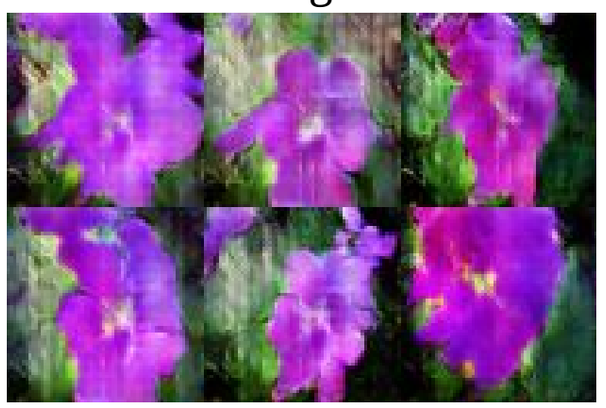


based on conditional-GAN framework

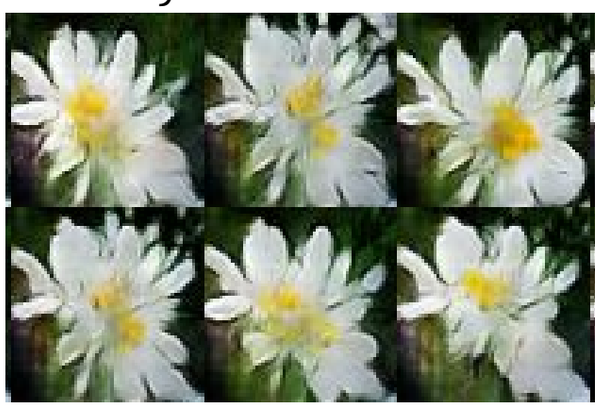
this small bird has a pink breast and crown, and black primaries and secondaries.

this magnificent fellow is almost all black with a red crest, and white cheek patch.

the flower has petals that are bright pinkish purple with white stigma



this white and yellow flower have thin white petals and a round yellow stamen



[Ref: Reed et al.,
Generative Adversarial Text to Image Synthesis, ICML2016]

Disentangling Latent Space - More Models

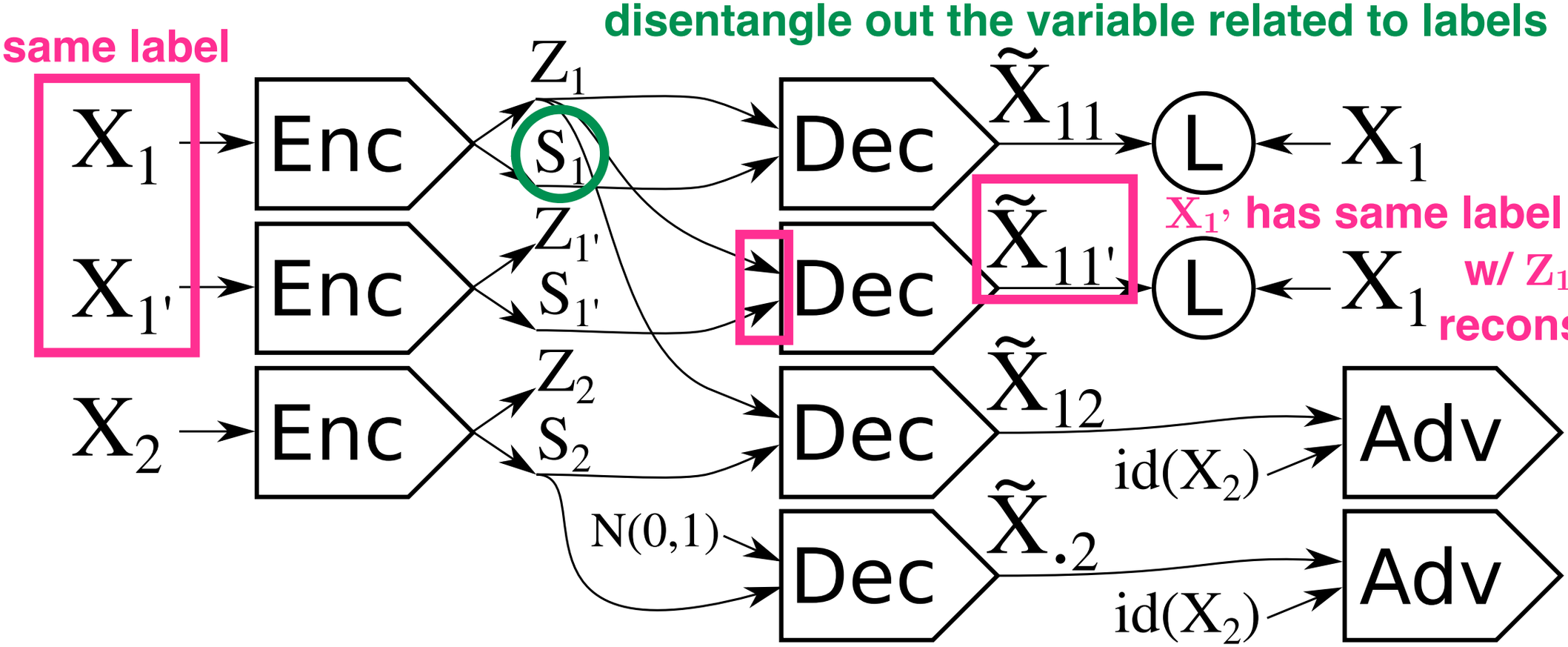
- Disentangling factors of variation in deep representations using adversarial training, Mathieu et al., NIPS'16

X_1 and $X_{1'}$ are with the same label,

whereas X_2 can have any label

to learn encoder which can produce disentanglement

disentangle out the variable related to labels



$X_{1'}$ has same label S_1 as X_1 w/ Z_1 from X_1 reconstruct X_1

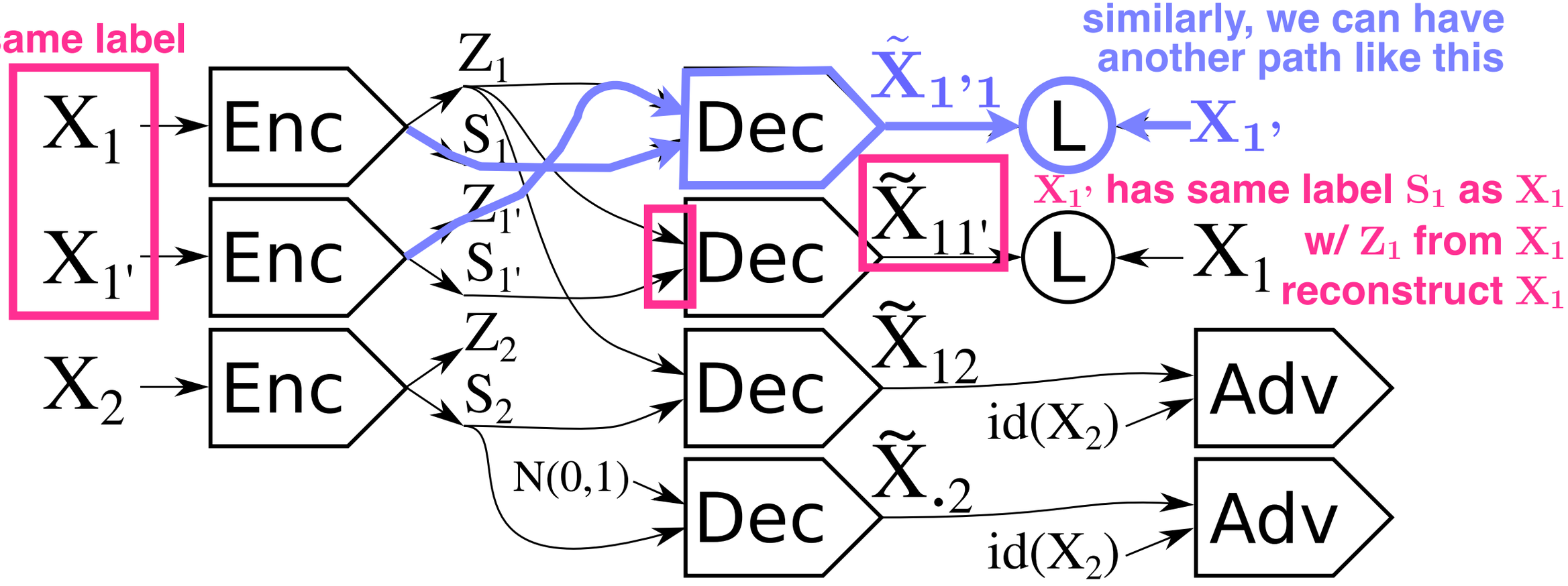
Disentangling Latent Space - More Models

- Disentangling factors of variation in deep representations using adversarial training, Mathieu et al., NIPS'16

X_1 and $X_{1'}$ are with the same label,

whereas X_2 can have any label

to learn encoder which can produce disentanglement



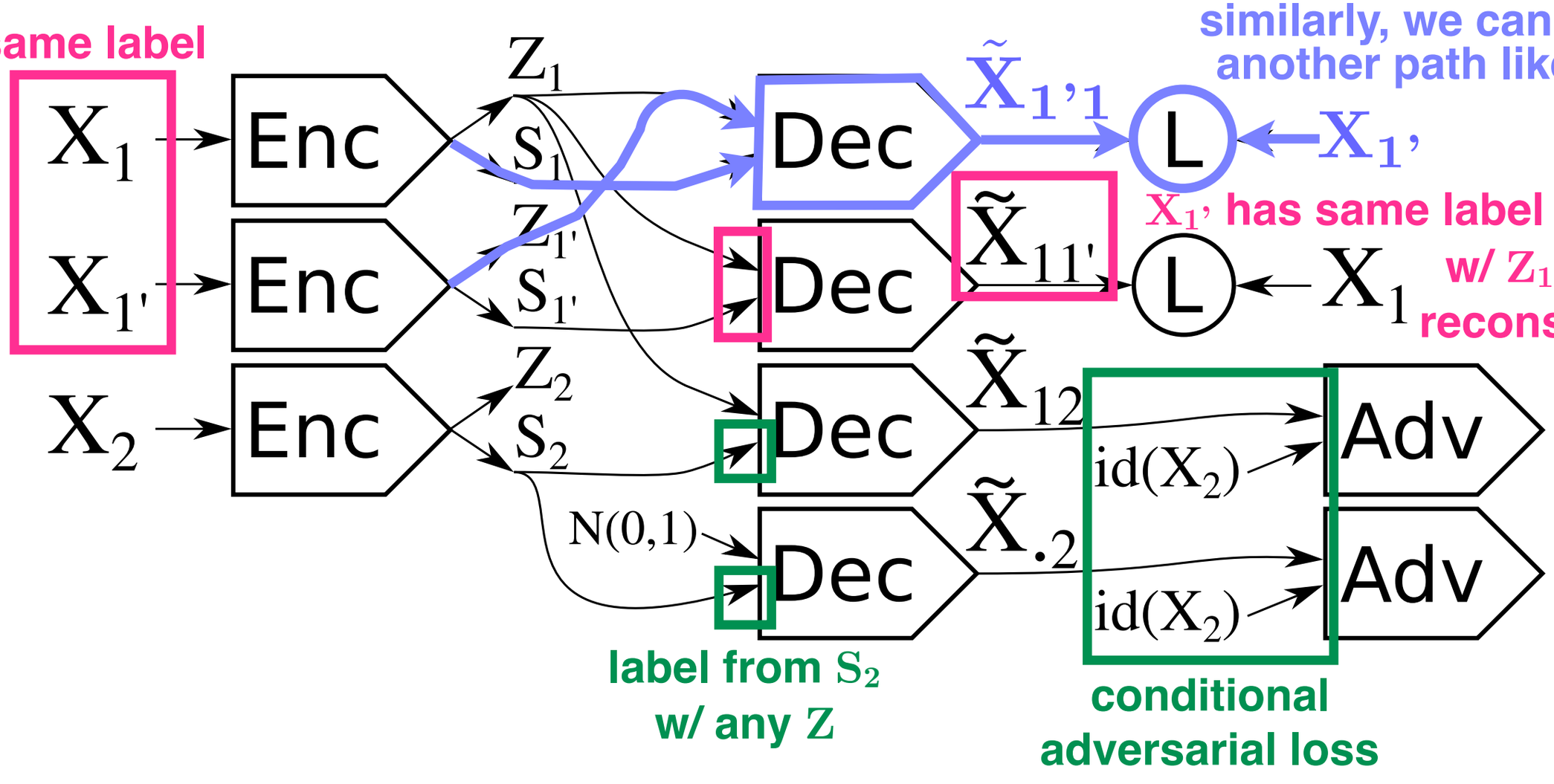
Disentangling Latent Space - More Models

- Disentangling factors of variation in deep representations using adversarial training, Mathieu et al., NIPS'16

X_1 and $X_{1'}$ are with the same label,

whereas X_2 can have any label

to learn encoder which can produce disentanglement

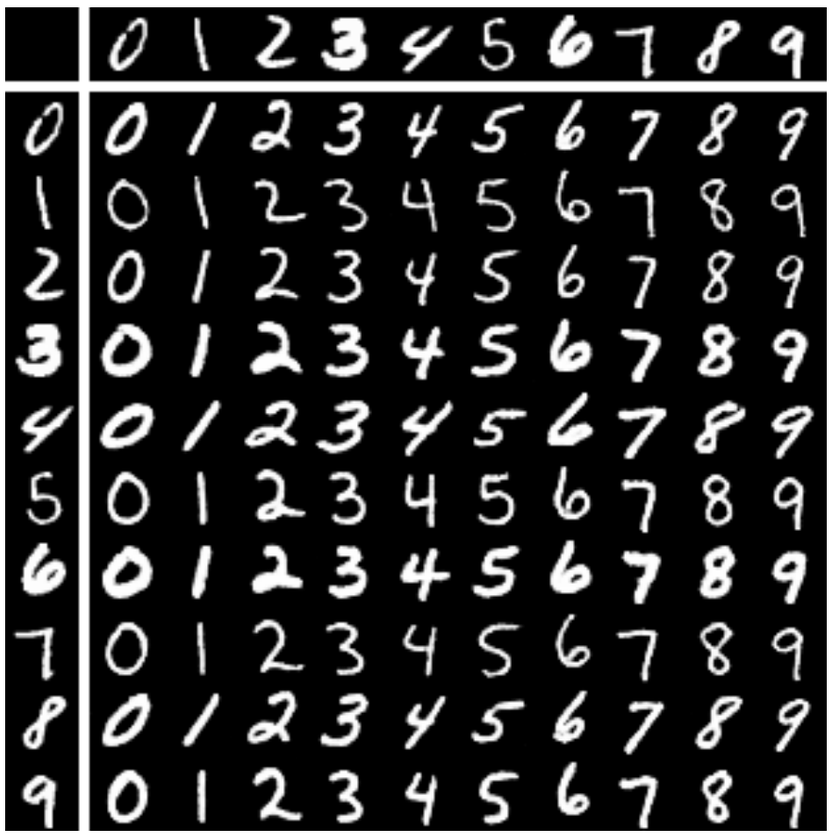


資訊工程學系
Department of Computer Science

Disentangling Latent Space - More Models

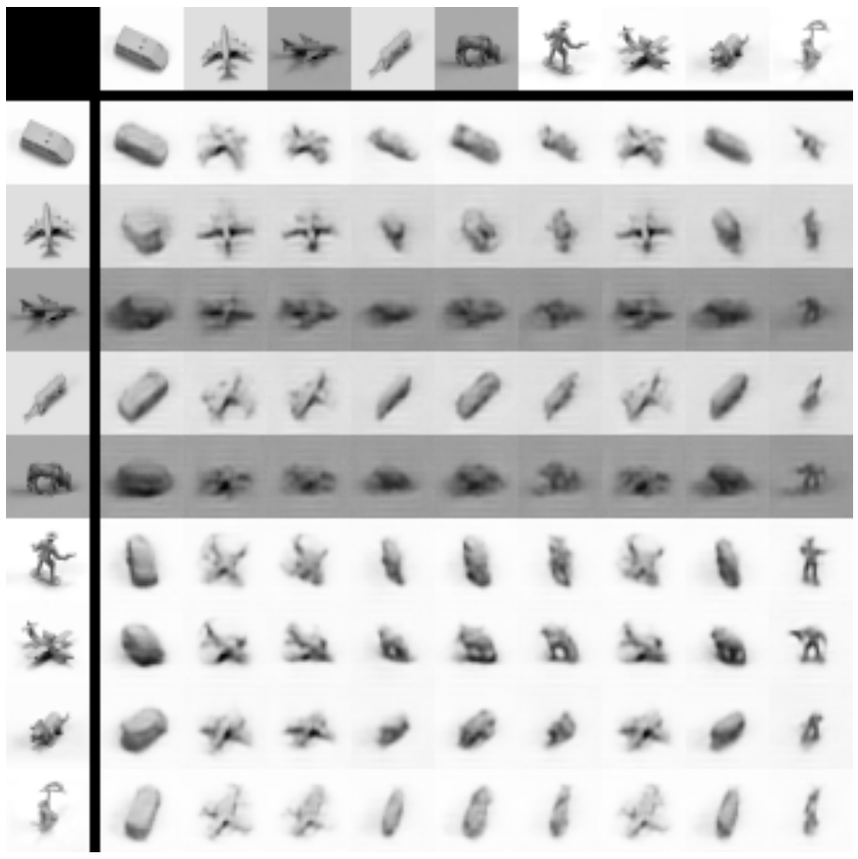
- Disentangling factors of variation in deep representations using adversarial training, Mathieu et al., NIPS'16

Z from left column (style)
S from top row (digit class)



Z from left column (pose)
S from top row (character)

Z from left column (pose)
S from top row (object)



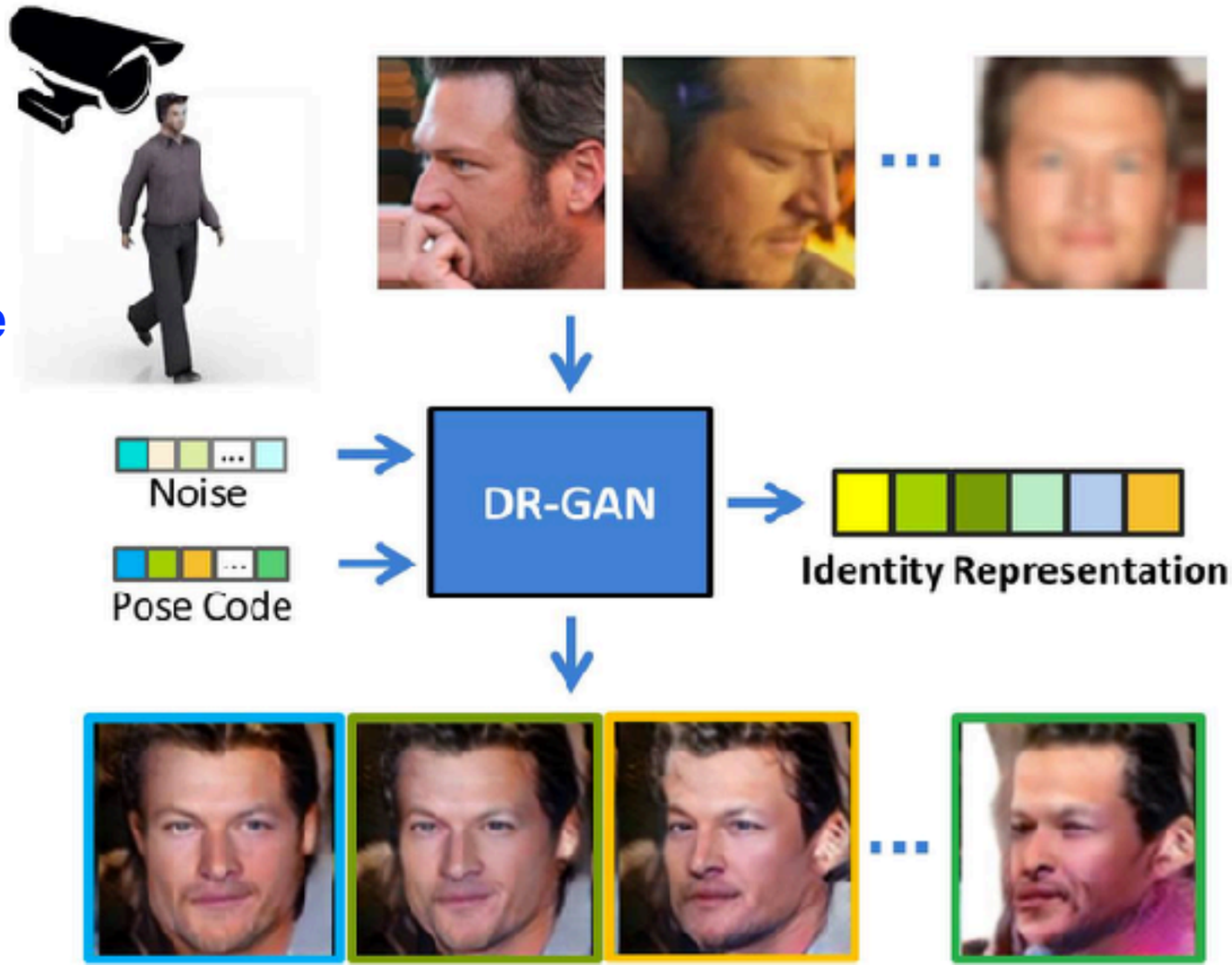
Disentangling Latent Space - More Models

- Disentangled Representation Learning GAN for Pose-Invariant Face Recognition, Tran et al., CVPR'17

Given a face image $\mathbf{x}$ w/ label $\mathbf{y} = \{y^d, y^p\}$, where y^d represents identity label and y^p for pose.

Objective is to:

1) **learn a pose-invariant identity representation**
2) **synthesize a face image $\hat{\mathbf{x}}$ with the *same* identity y^d but a *different* pose specified by pose code c.**



Disentangling Latent Space - More Models

- Disentangled Representation Learning GAN for Pose-Invariant Face Recognition, Tran et al., CVPR'17

objective for D:

$$E_{\mathbf{x},\mathbf{y}\sim p_d(\mathbf{x},\mathbf{y})}[\log D^d_{y^d}(\mathbf{x}) + \log D^p_{y^p}(\mathbf{x})] +$$

real data and labels — N+1 classifier for identity — classifier for pose

$$E_{\substack{\mathbf{x},\mathbf{y}\sim p_d(\mathbf{x},\mathbf{y}),\\ \mathbf{z}\sim p_z(\mathbf{z}),\mathbf{c}\sim p_c(\mathbf{c})}}[\log(D^d_{N^d+1}(G(\mathbf{x},\mathbf{c},\mathbf{z})))]$$

fake class

objective for G:

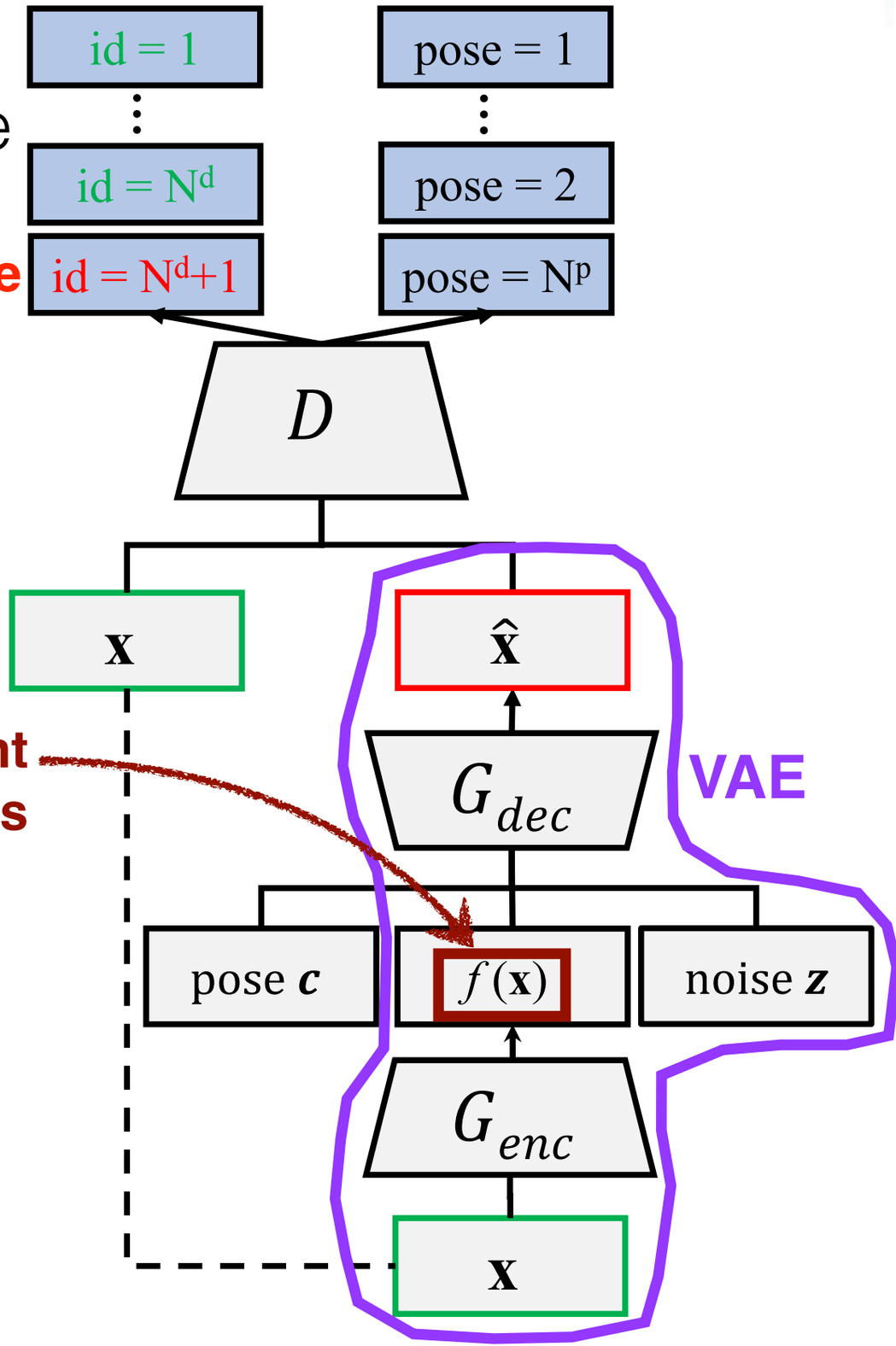
$$E_{\substack{\mathbf{x},\mathbf{y}\sim p_d(\mathbf{x},\mathbf{y}),\\ \mathbf{z}\sim p_z(\mathbf{z}),\mathbf{c}\sim p_c(\mathbf{c})}}[\log(D^d_{y^d}(G(\mathbf{x},\mathbf{c},\mathbf{z})))+$$

make $f(\mathrm{x})$ pose invariant

$$\log(D^p_{y^t}(G(\mathbf{x},\mathbf{c},\mathbf{z})))]$$

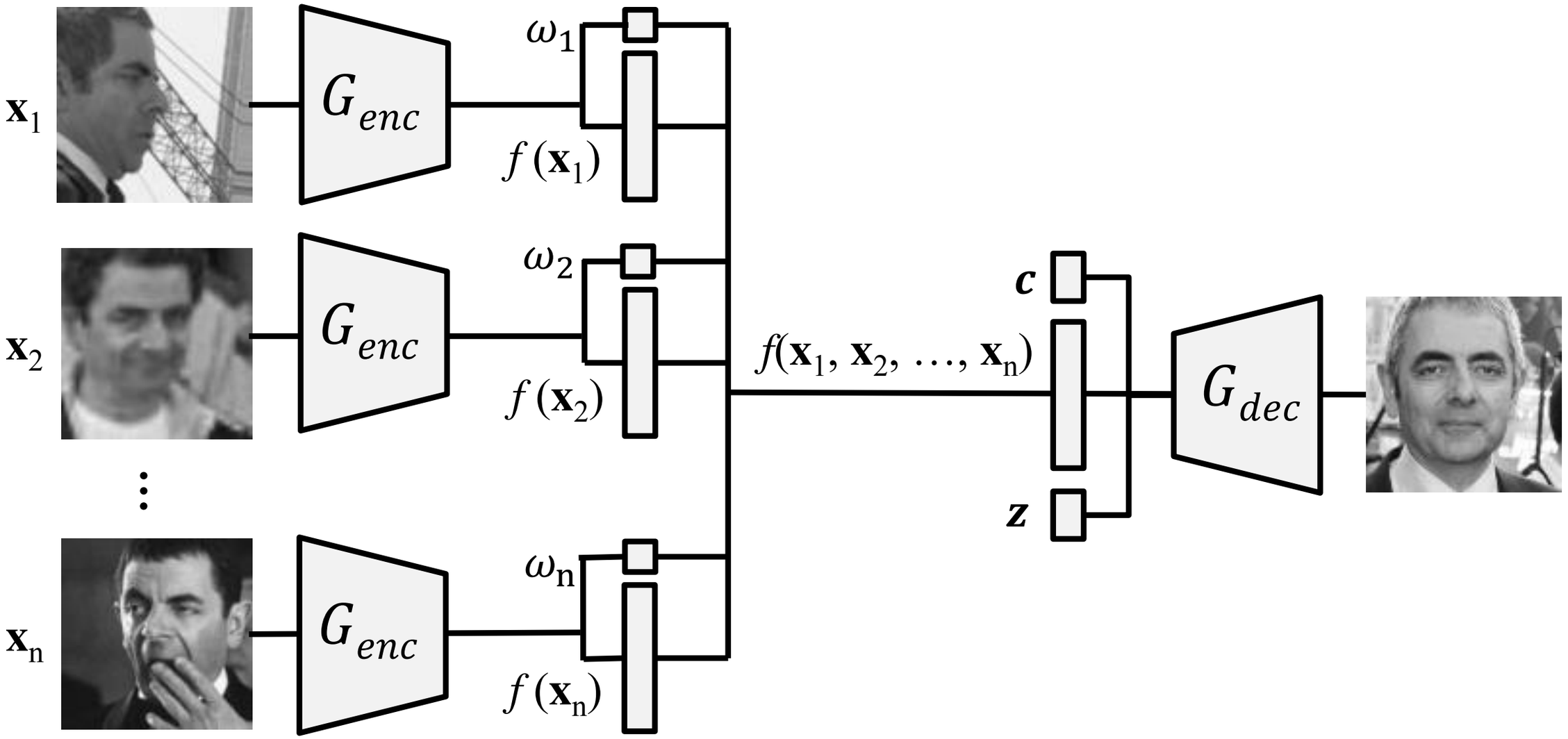
any pose change should not affect identity recognition

predict the pose which it was assigned



Disentangling Latent Space - More Models

- Disentangled Representation Learning GAN for Pose-Invariant Face Recognition, Tran et al., CVPR'17

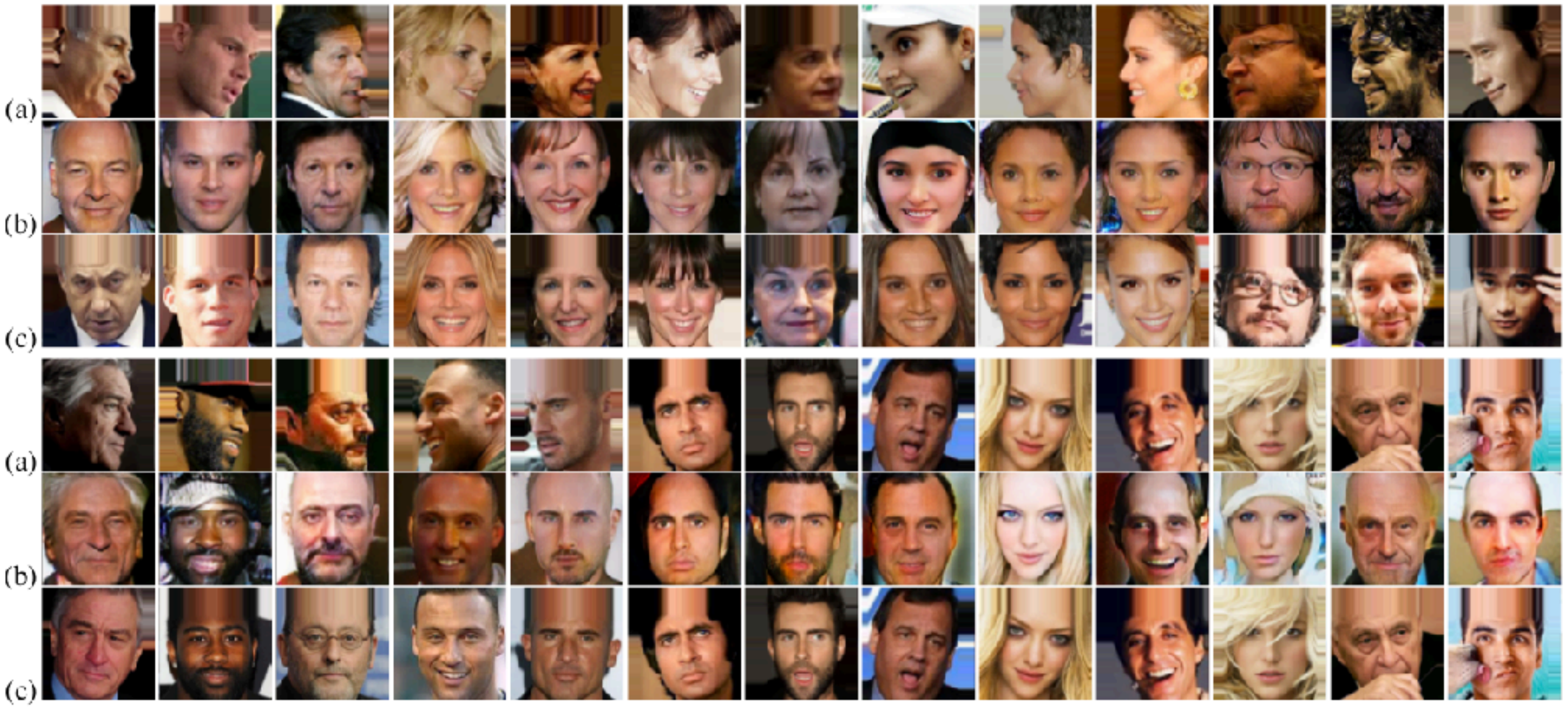


From an image set of a subject, we can fuse the features to a single representation via dynamically learnt coefficients ω and synthesize images in any pose

Disentangling Latent Space - More Models

- Disentangled Representation Learning GAN for Pose-Invariant Face Recognition, Tran et al., CVPR'17

(a) input, (b) frontalized faces by DR-GAN, (c) real frontal faces.



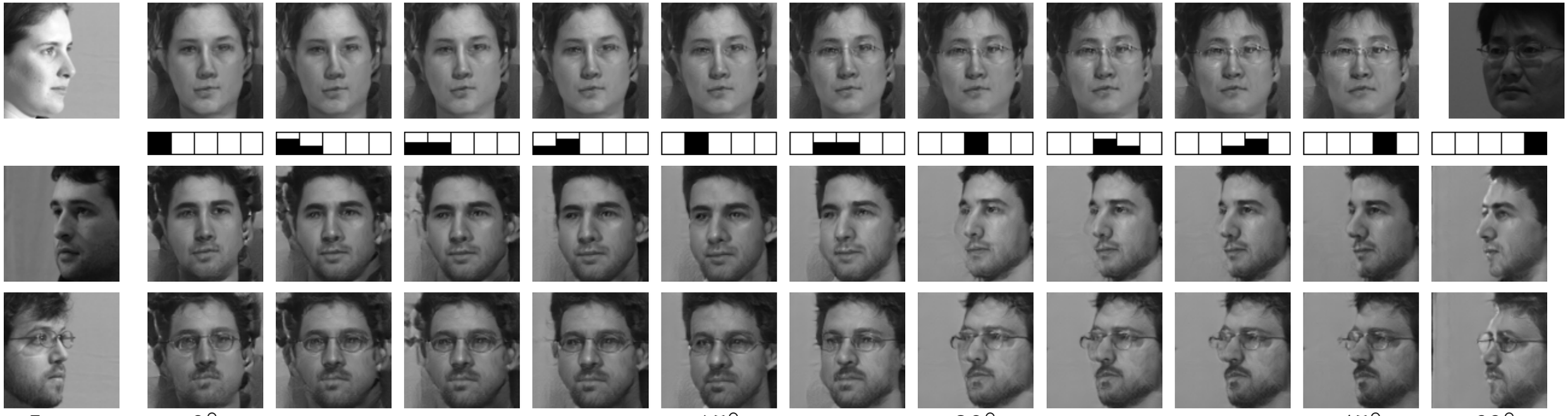
Disentangling Latent Space - More Models

- Disentangled Representation Learning GAN for Pose-Invariant Face Recognition, Tran et al., CVPR'17

(a) interpolation across identities, (b) interpolation across "discrete" poses

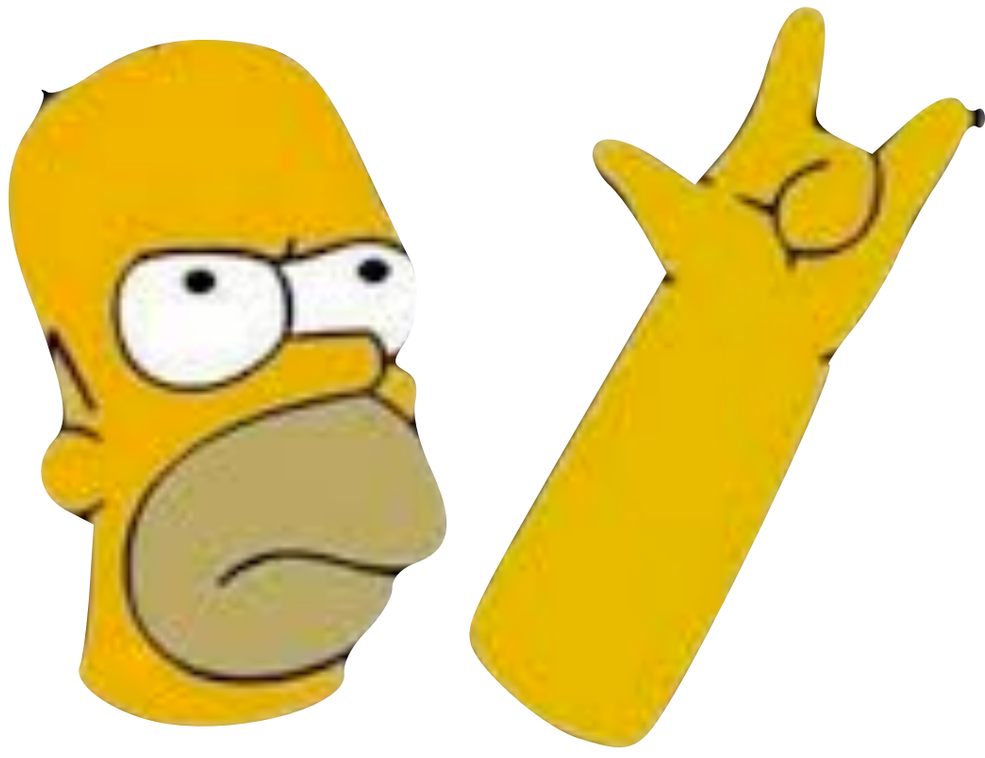
(a)

(b)



yes, of course they also provide state-of-the-art performance on face recognition task, towards benchmark with pose variance

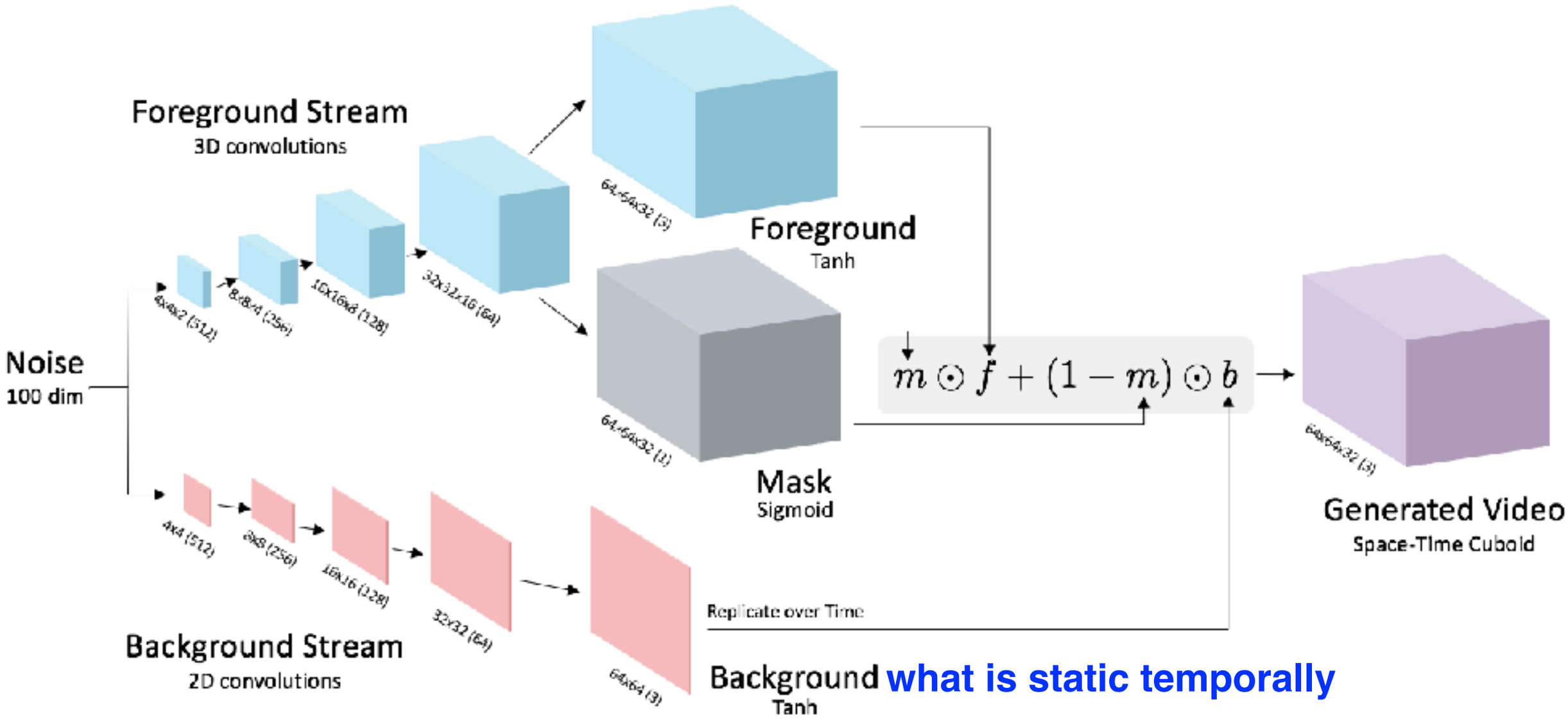
Sick of Equations and Static Images?



Letz go to something with dynamics!

Video Generative Adversarial Network:)

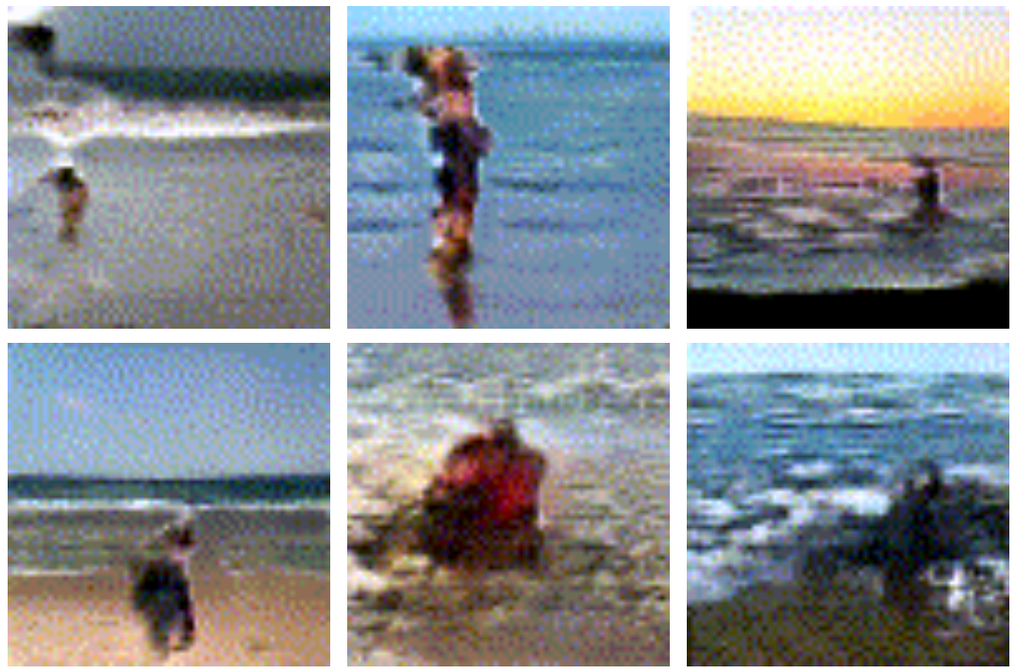
- Generating Videos with Scene Dynamics, Vondrick et al., NIPS'16



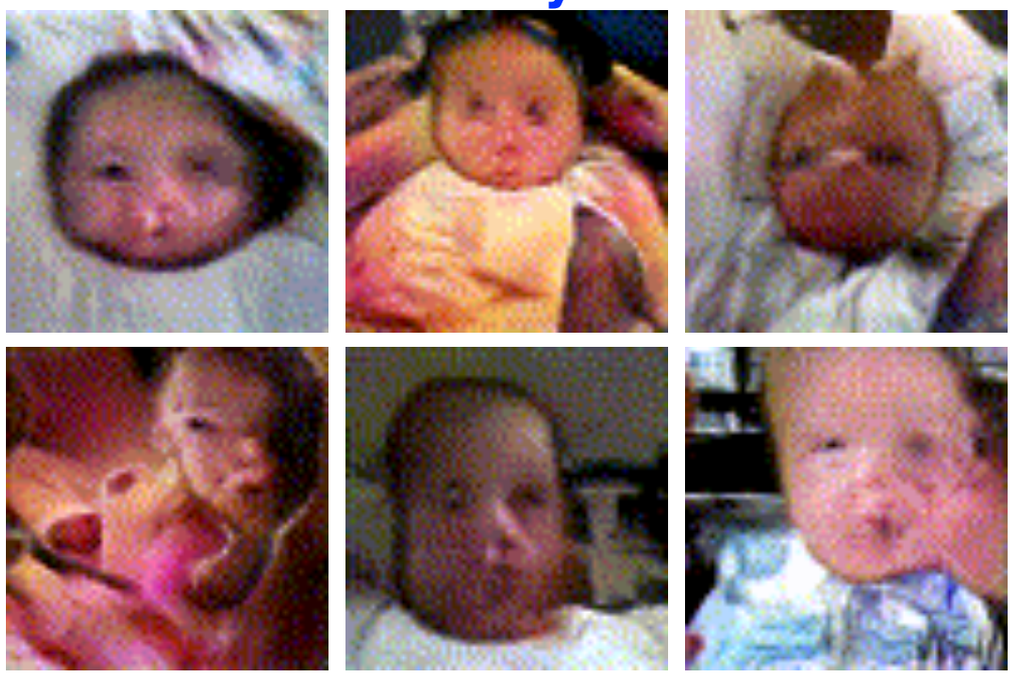
Video Generative Adversarial Network:)

- Generating Videos with Scene Dynamics, Vondrick et al., NIPS'16

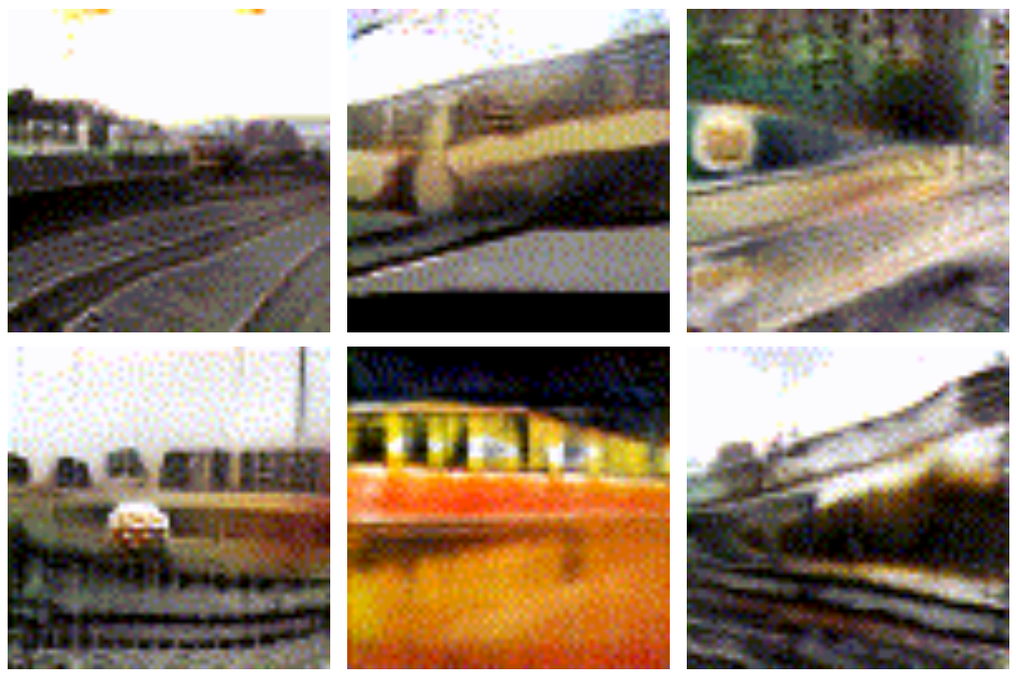
beach



baby



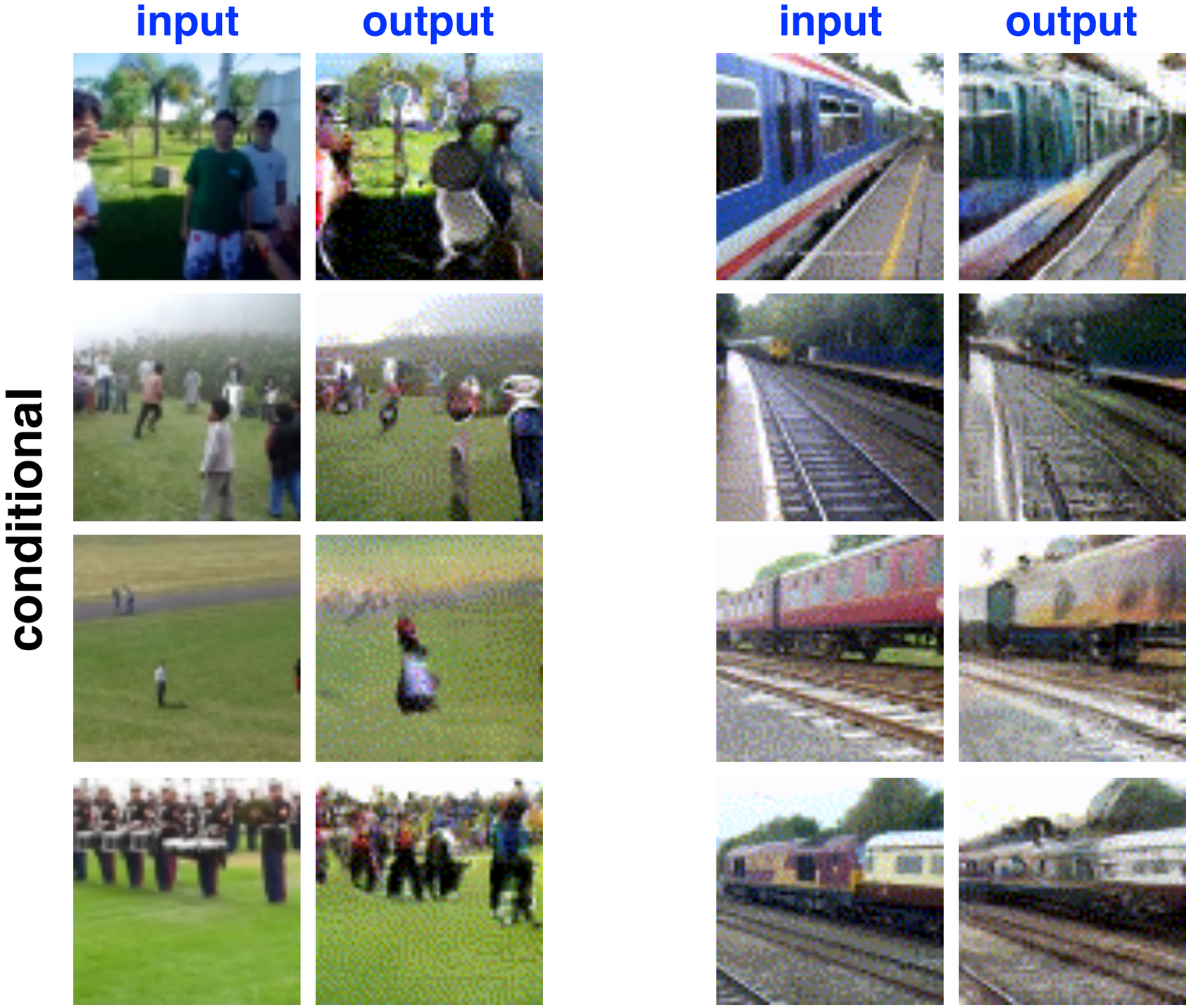
train station



golf

Video Generative Adversarial Network:)

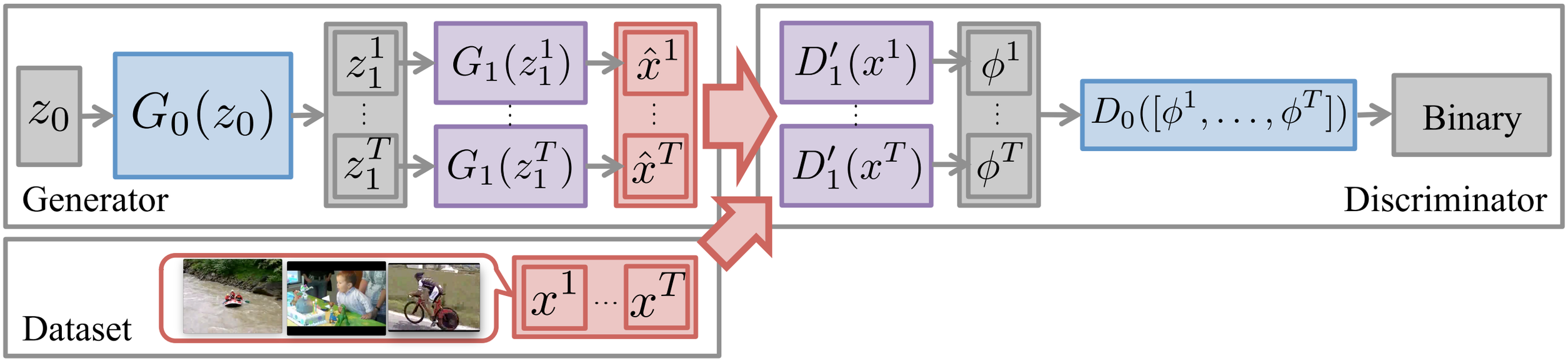
- Generating Videos with Scene Dynamics, Vondrick et al., NIPS'16



資訊工程學系
Department of Computer Science

Video Generative Adversarial Network

- Temporal Generative Adversarial Network, Saito et al., ArXiv'16



Two step training:

1) Image-based: Train image-based generator G_1 and discriminator D_1

2) Video-based: Keep G_1 and D_1 fixed.

Now video-based generator G_0 yields a set $\{z_1^1 \ldots z_1^T\}$ from z_0 ,

then G_1 transforms them into video frames $\{\hat{x}^1 \ldots \hat{x}^T\}$.

D_1 now acts to extract feature ϕ of each frame (from real or fake).

The temporal discriminator D_0 exploits $\{\phi^1 \ldots \phi^T\}$ and evaluate whether these frames are from the real videos or the generator.

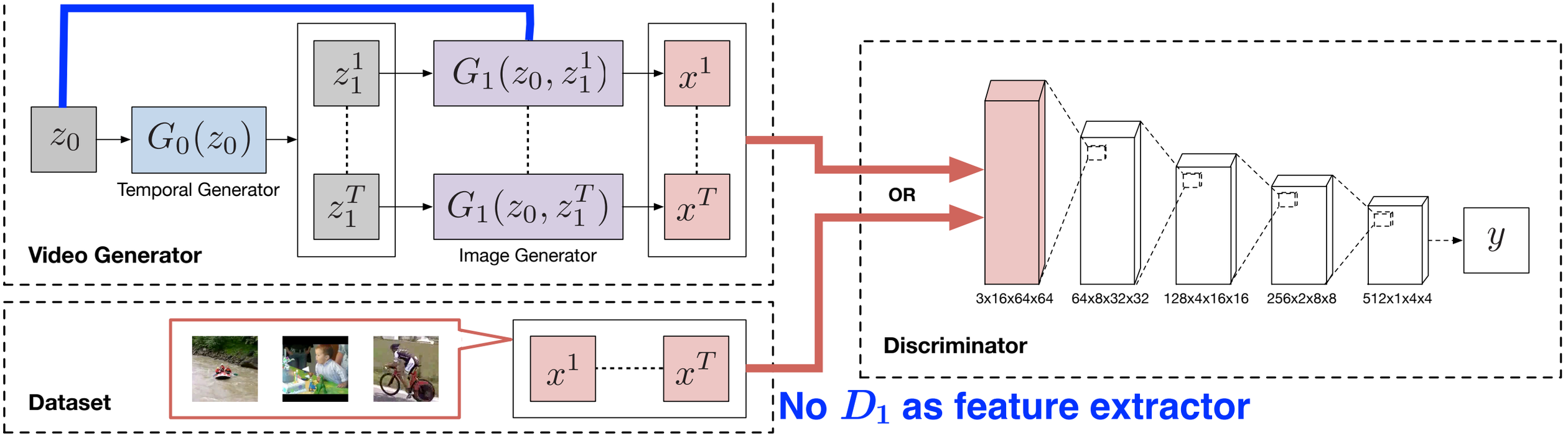
資訊工程學系
Department of Computer Science

Video Generative Adversarial Network

- Temporal GANs with Singular Value Clipping, Saito et al., ICCV'17

related to WGAN to handle mode collapse

Image generator G_1 now sees the global z_0 which is invariable w.r.t. time.
it is empirically observed that z_0 has a significant role in suppressing a sudden change of the action of the generated video.



No D_1 as feature extractor

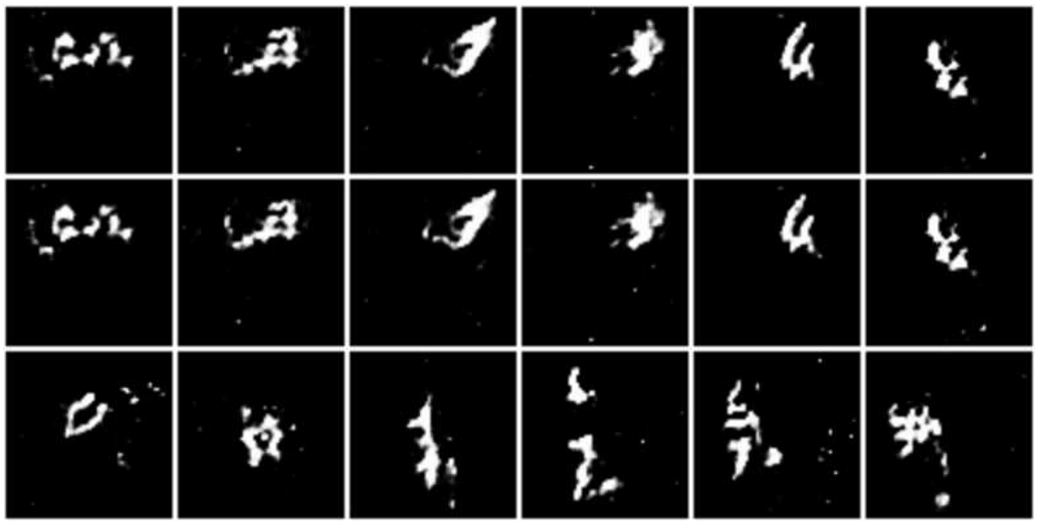
~~**Two step training**~~

資訊工程學系
Department of Computer Science

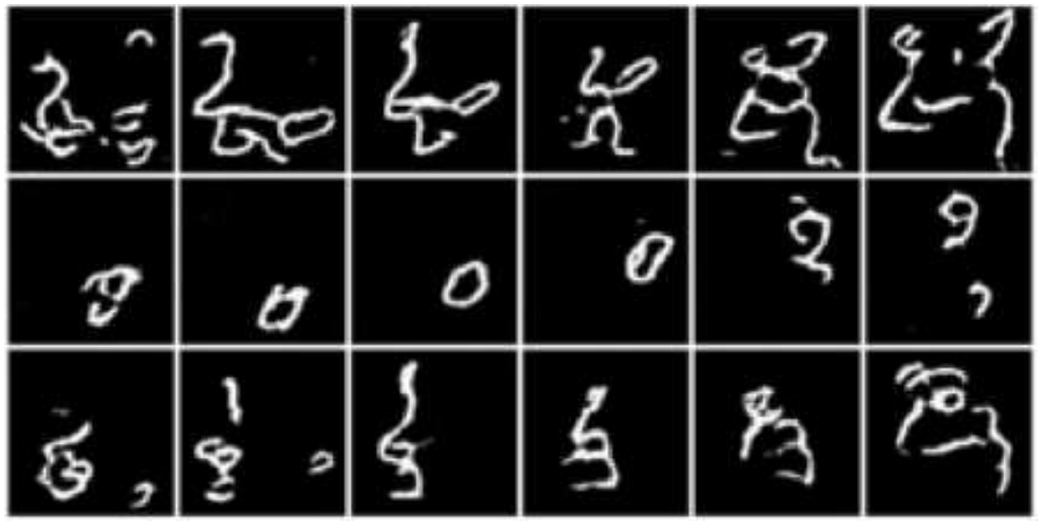
Video Generative Adversarial Network

- Temporal GANs with Singular Value Clipping, Saito et al., ICCV'17

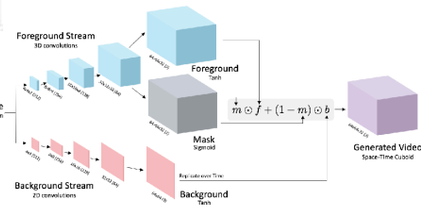
Frame 1 Frame 16 Frame 1 Frame 16

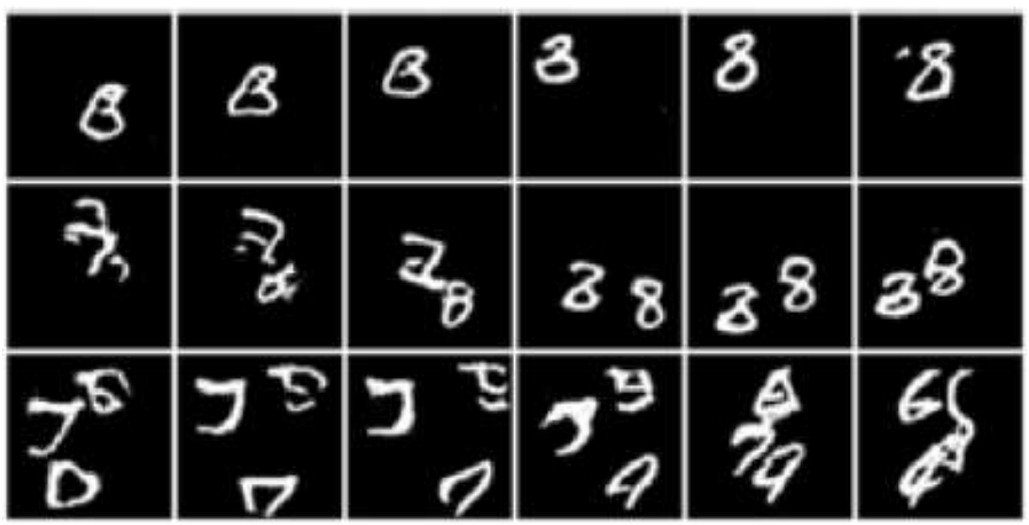


(a) 3D model (GAN)

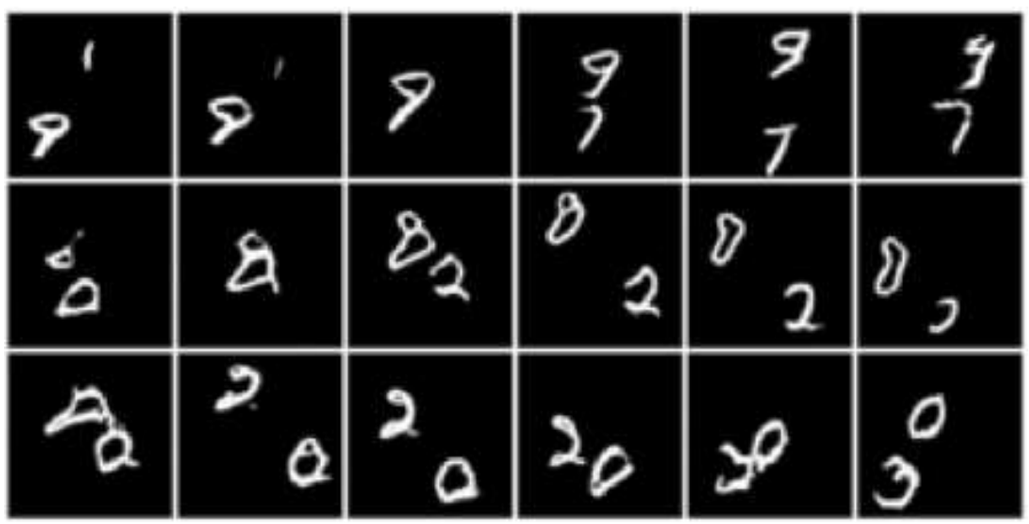


(b) 3D model (WGAN w/ SVC)





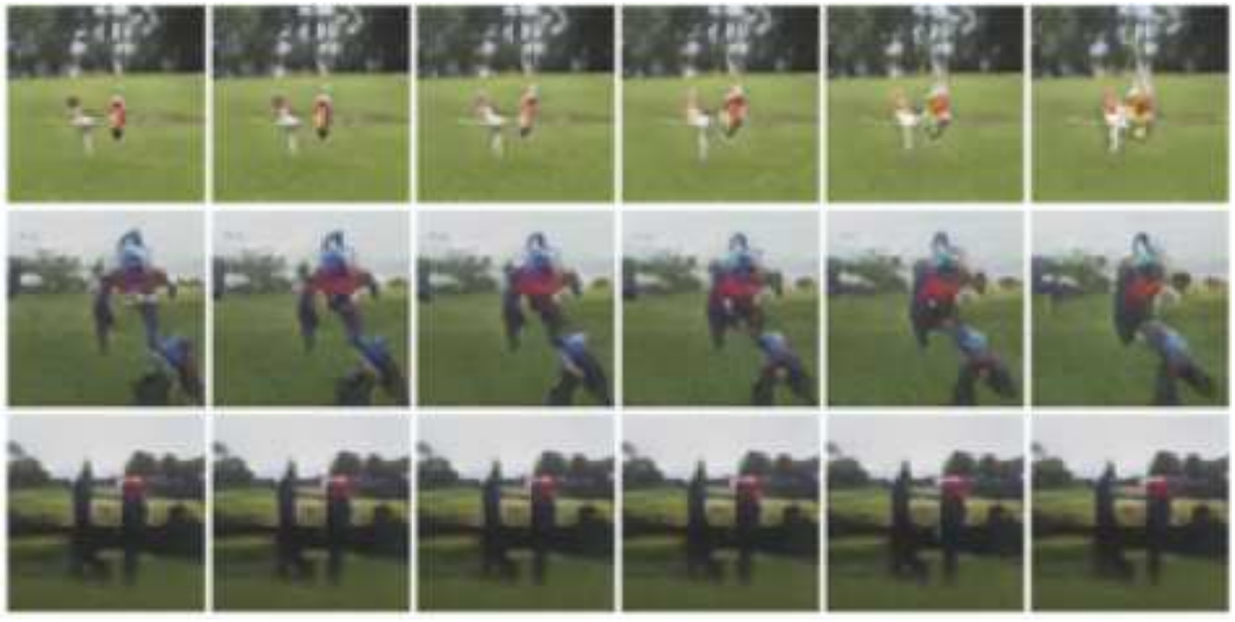
(c) TGAN (SVC, $G_1(z_1^t)$)



(d) TGAN (SVC, $G_1(z_0, z_1^t)$)

Video Generative Adversarial Network

- Temporal GANs with Singular Value Clipping, Saito et al., ICCV'17



conditional

BaseballPitch

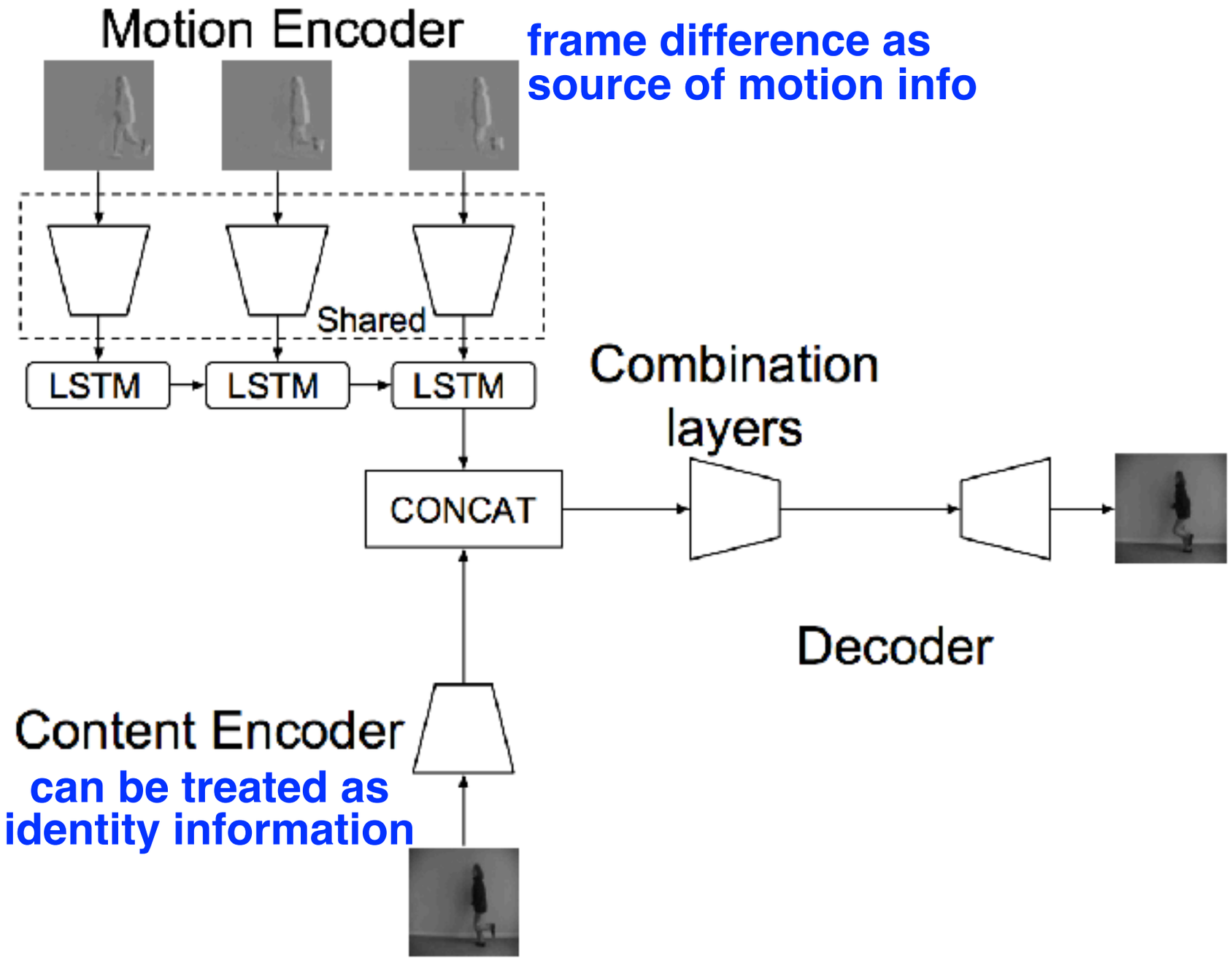
IceDancing

Video Generative Adversarial Network

- Temporal GANs with Singular Value Clipping, Saito et al., ICCV'17

More Works to Come About Video Generative Models

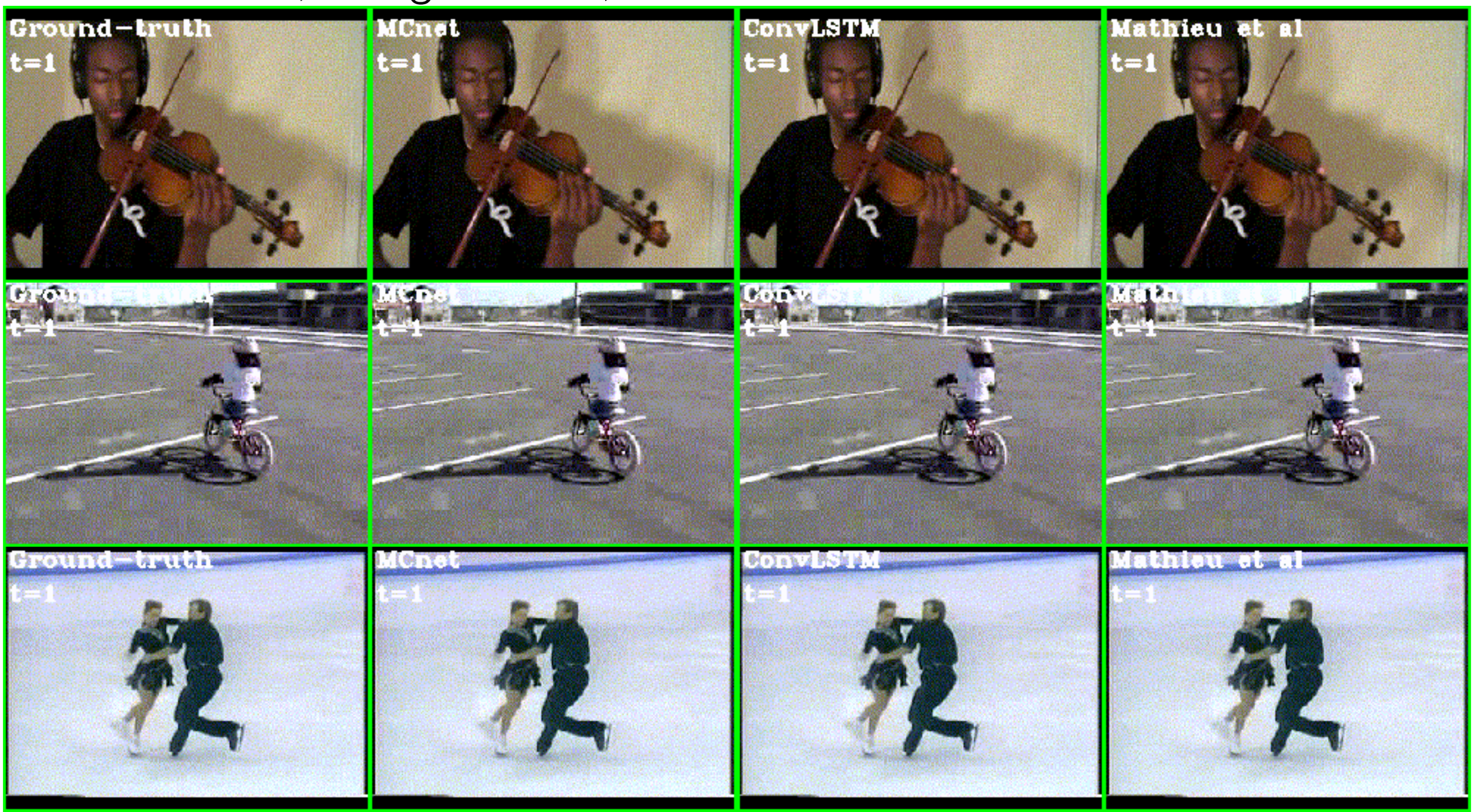
- Decomposing Motion and Content for Natural Video Sequence Prediction, Villegas et al., ICLR'17



More Works to Come About Video Generative Models

- Decomposing Motion and Content for Natural Video Sequence Prediction, Villegas et al., ICLR'17

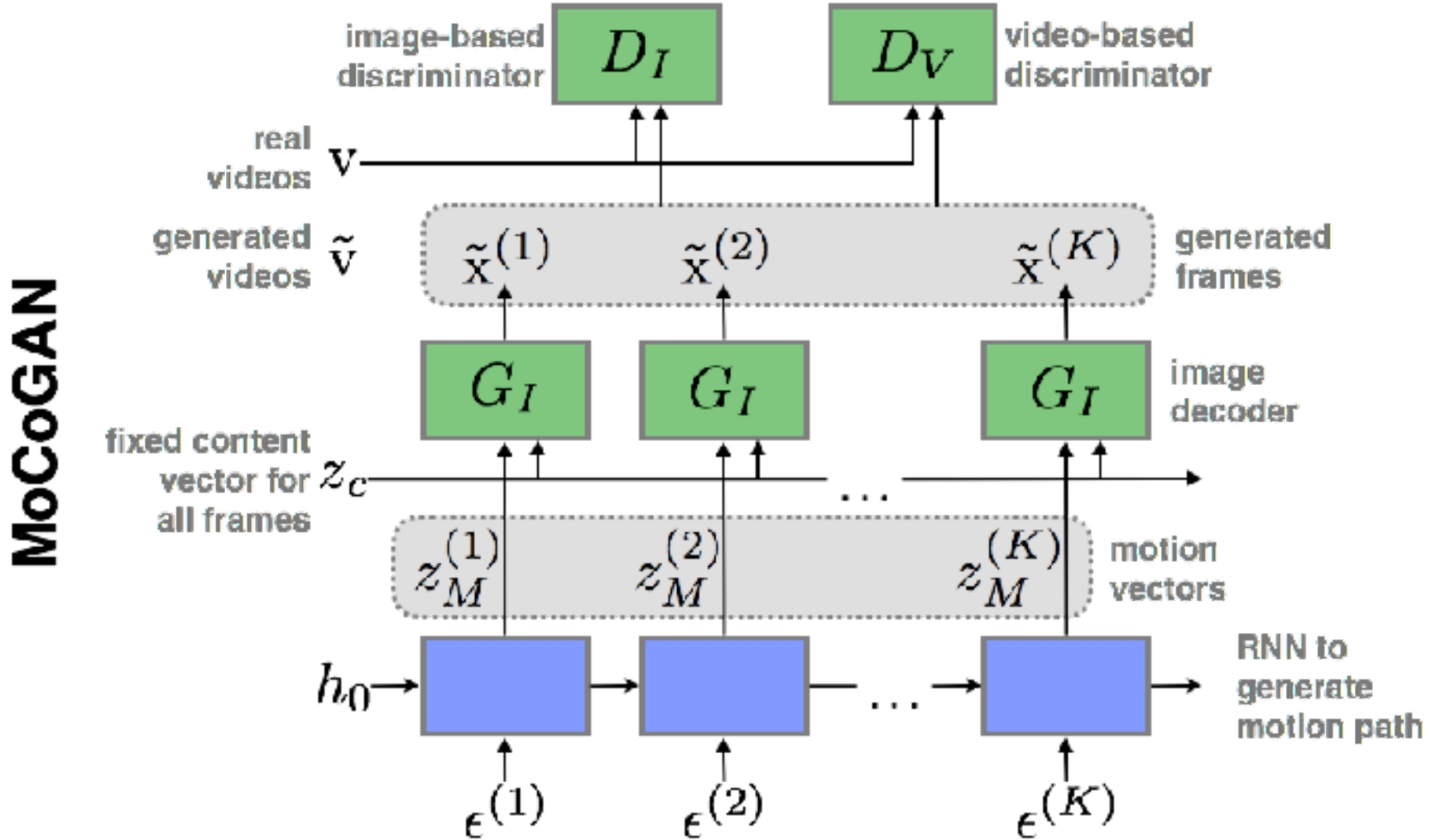
green as input, red as prediction



More Works to Come About Video Generative Models

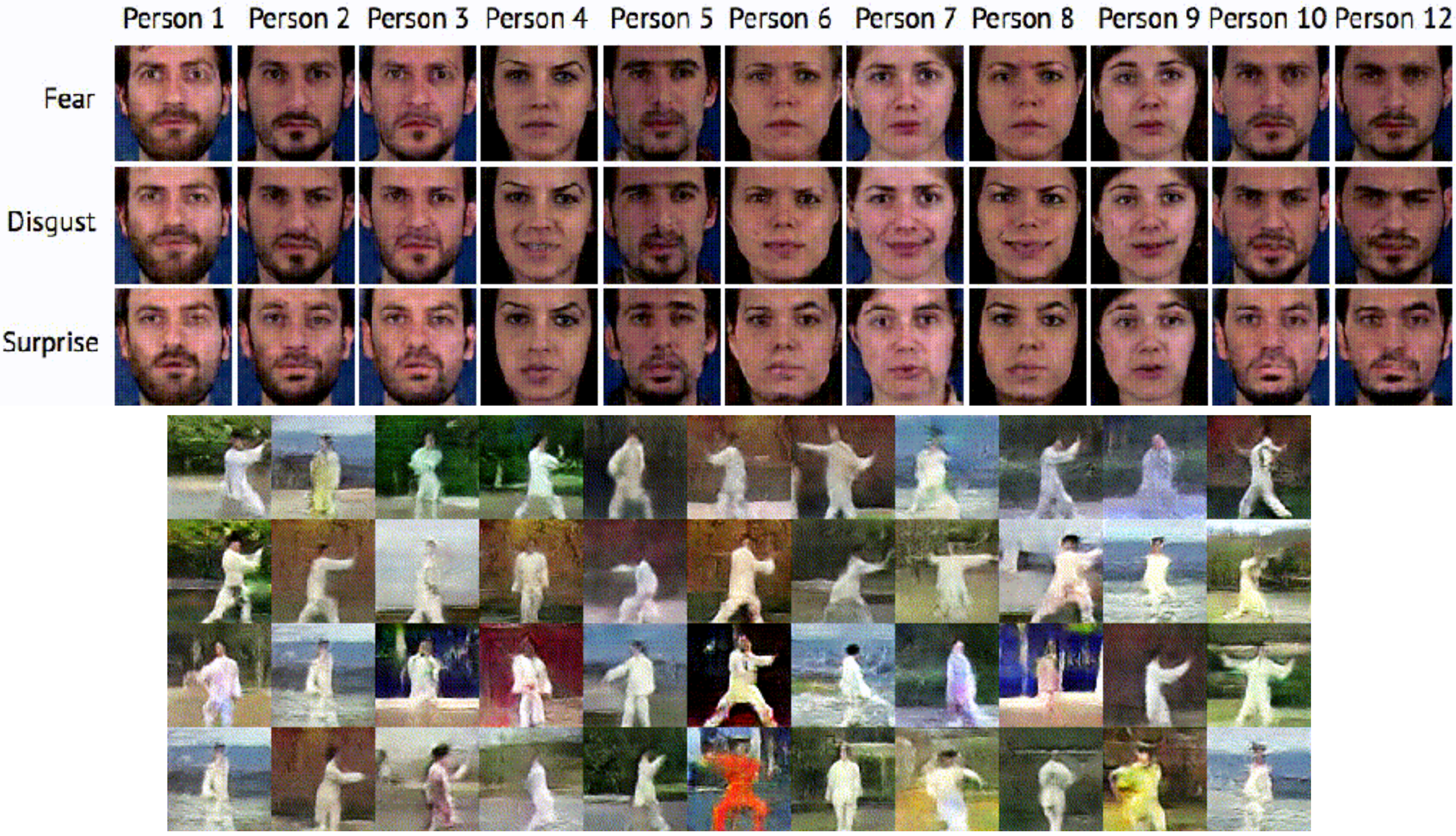
- MoCoGAN, Tulyakov et al., ArXiv'17

A video is represented as a combination of a vector in the content subspace and a trajectory in the motion subspace



More Works to Come About Video Generative Models

- MoCoGAN, Tulyakov et al., ArXiv'17

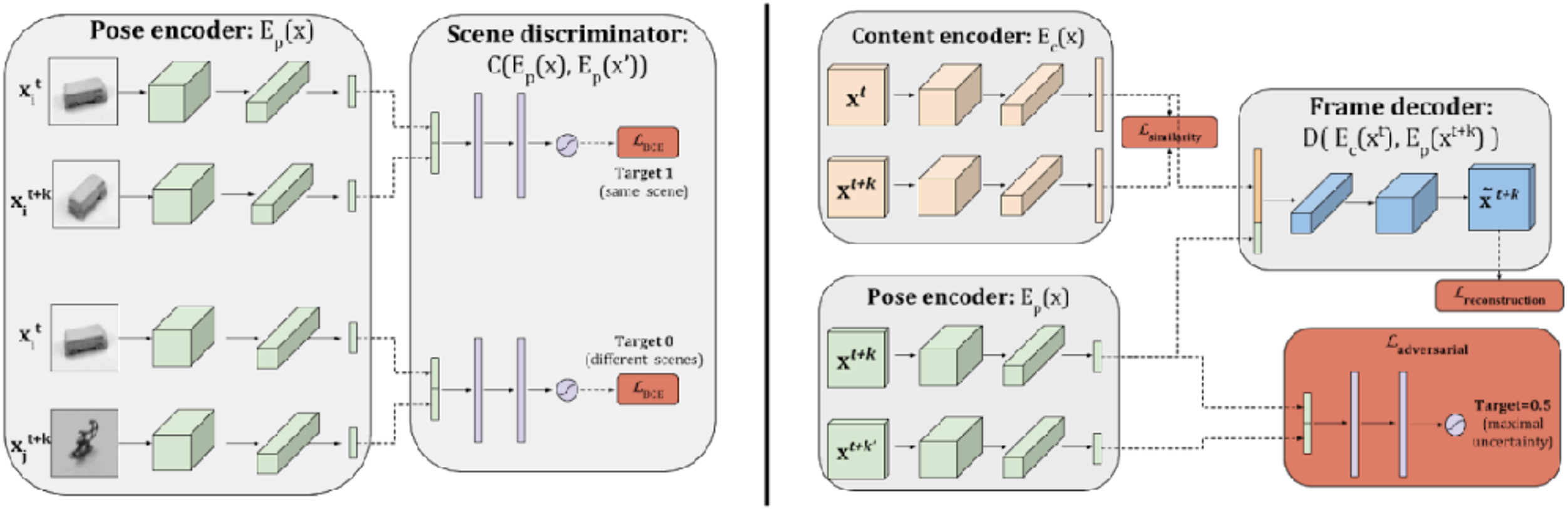


More Works to Come About Video Generative Models

- Unsupervised Learning of Disentangled Representations from Video, Denton et al., NIPS'17

content vector invariant within a video

$$\mathcal{L}_{similarity}(E_c) = ||E_c(x^t) - E_c(x^{t+k})||_2^2$$

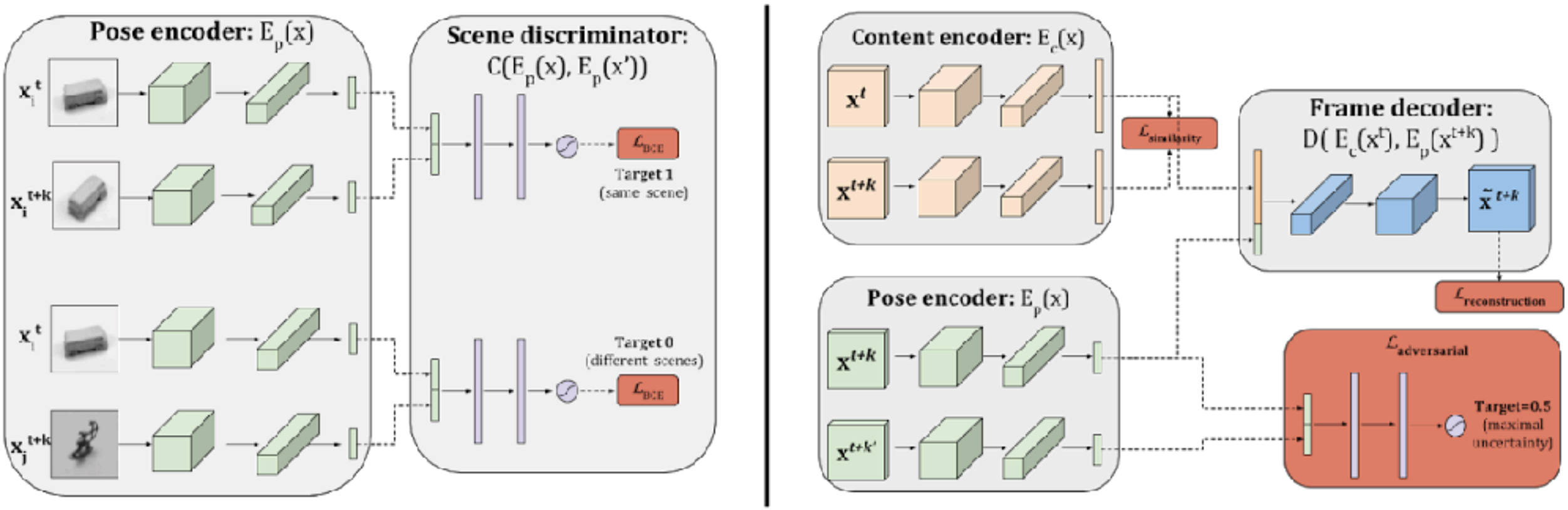


$$\mathcal{L}_{reconstruction}(E_c, E_p, D) = ||D(E_c(x^t), E_p(x^{t+k})) - x^{t+k}||_2^2$$

consistent content from x^t, with pose from x^{t+k}, should be able to reconstruct frame x^{t+k}

More Works to Come About Video Generative Models

- Unsupervised Learning of Disentangled Representations from Video, Denton et al., NIPS'17



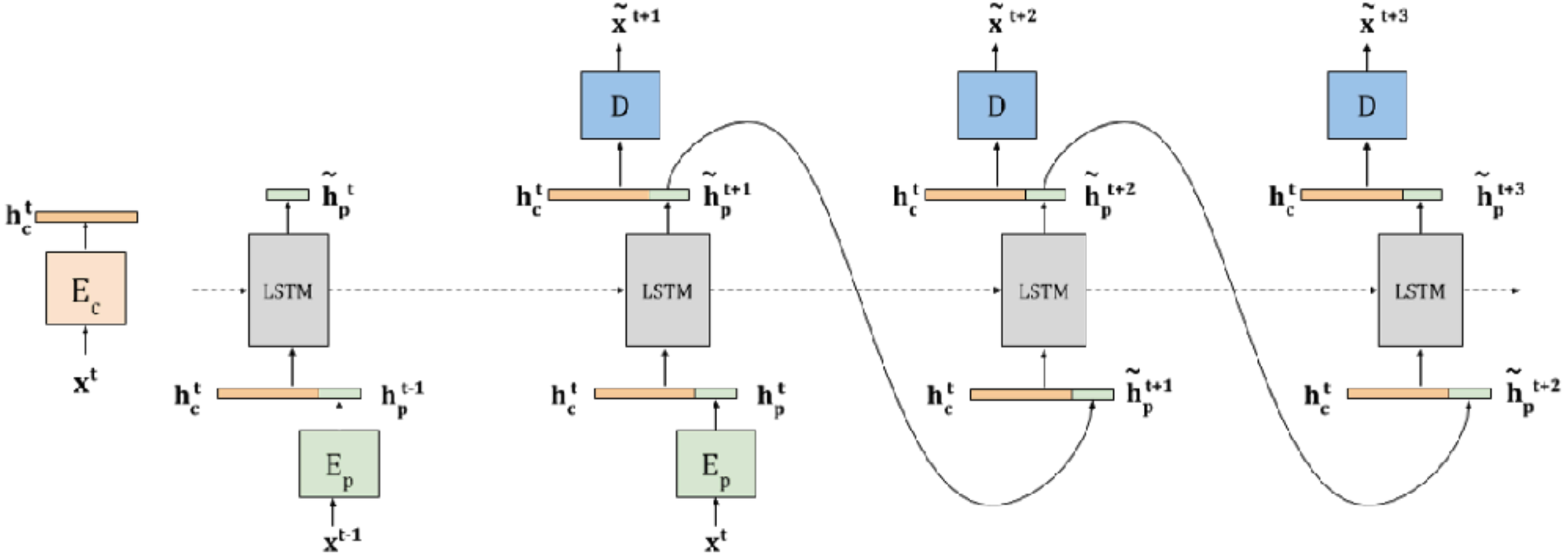
content vectors invariant within a video, but distinct between videos

→ pose features (after Ep) shouldn't carry any info about identity of objects within a clip

→ scene discriminator C attempt to classify the pair as being from same/different video

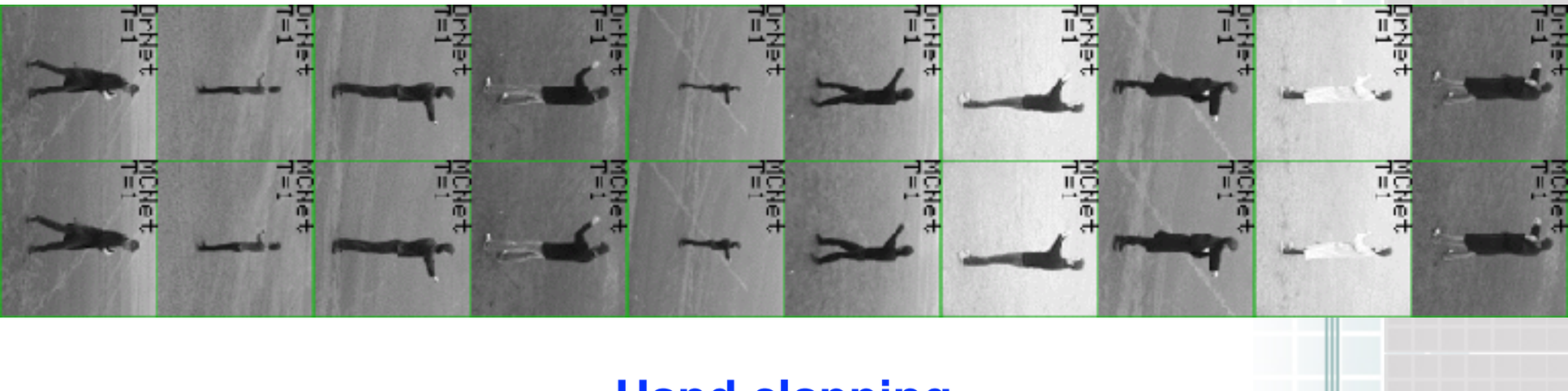
More Works to Come About Video Generative Models

- Unsupervised Learning of Disentangled Representations from Video, Denton et al., NIPS'17

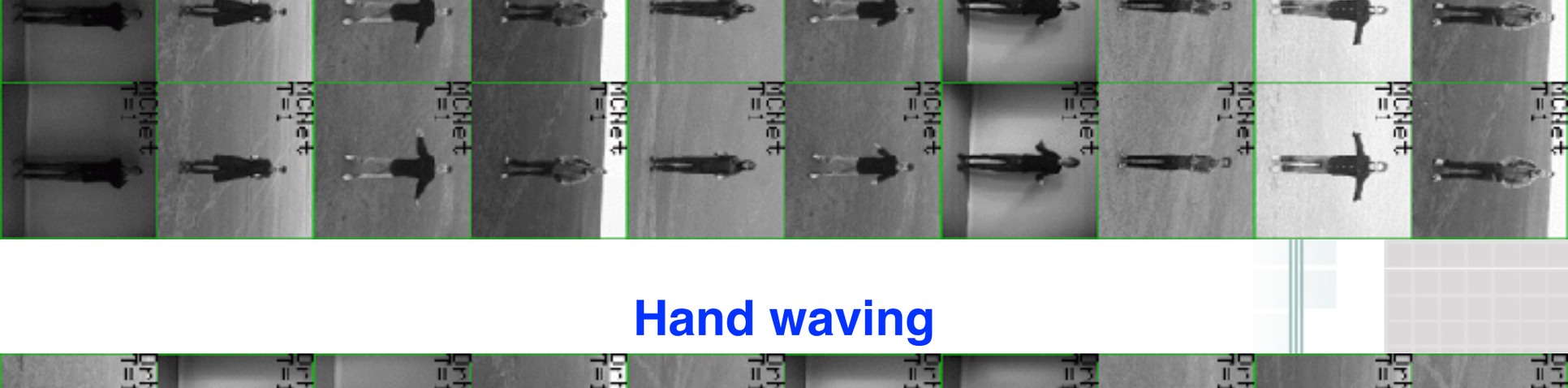


Video prediction: **apply a standard LSTM model to the pose features, conditioning on the content features from the last observed frame**

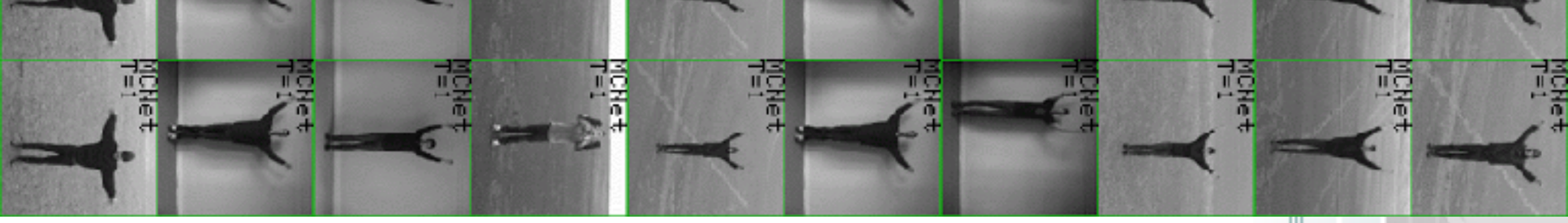
boxing



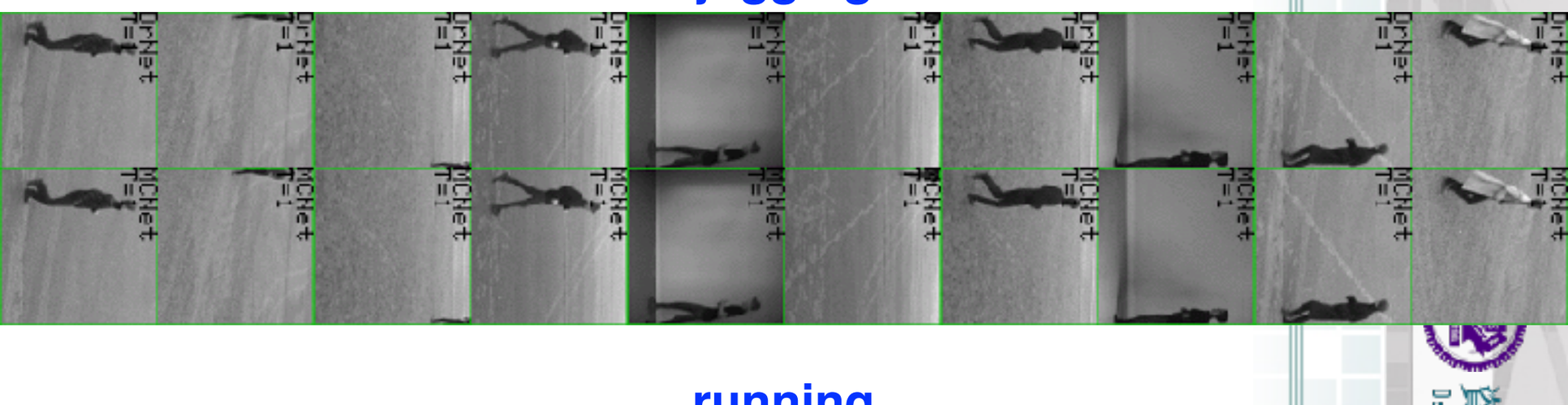
Hand clapping



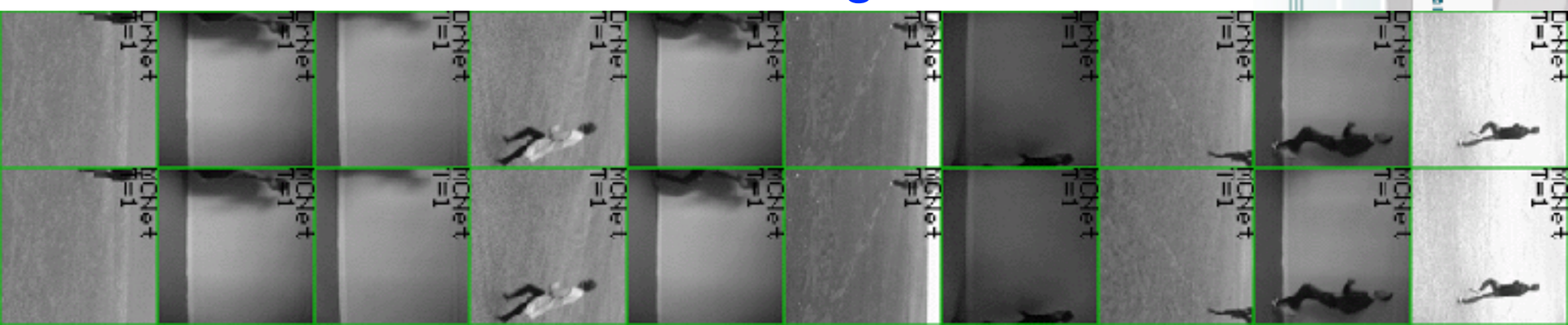
Hand waving



jogging



running



國立交通大學
National Chiao Tung University

Thanks! Questions?

Enriched Vision Applications Laboratory