

Generative Adversarial Networks for Face Recognition: A practical view – Part II

Arnold Wiliem

The University of Queensland

a.wiliem@uq.edu.au ; arnold.wiliem@ieee.org



THE UNIVERSITY
OF QUEENSLAND
AUSTRALIA



Tutorial slides are available for download

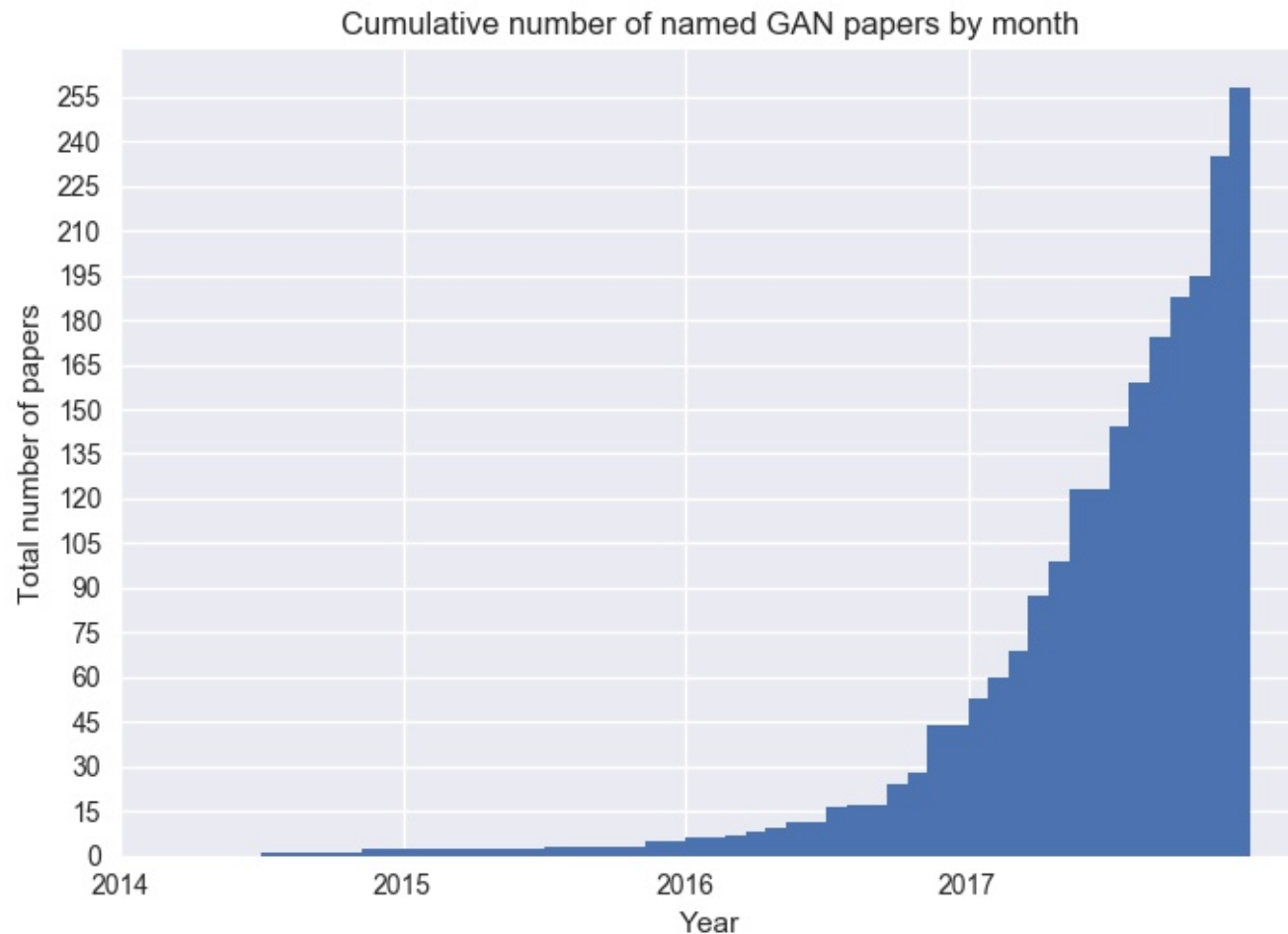
- https://outbox.eait.uq.edu.au/uqawilie/ICB2018_tutorial/

- Aims for this tutorial to provide
 - a brief introduction to GAN
 - basic understanding of GAN and recent advancements
 - how GAN can be used for face recognition problem
- This tutorial will **not**
 - provide in-depth technical and theoretical discussion on GAN
 - provide complete review of important papers
- More detailed technical treatments can be found from excellent tutorials that have been delivered previously
 - <https://arxiv.org/pdf/1701.00160.pdf>
 - https://github.com/mingyuliutw/cvpr2017_gan_tutorial

Agenda

- Brief introduction on GAN
- Recent works in GAN
- GAN for image-to-image translation
- GAN applications for face recognition

Why talk about GAN?



[1406.2661] Generative Adversarial Networks - arXiv

<https://arxiv.org/abs/1406.2661>

by IJ Goodfellow - 2014 - Cited by 2245 - Related articles

Jun 10, 2014 - Abstract: We propose a new framework for estimating **generative** models via an **adversarial** process, in which we simultaneously train two models: a **generative** model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data ...

- With this exponential growth of the field, it is easy to miss new works
- Solution?
 - See the GAN Zoo:
<https://deephunt.in/the-gan-zoo-79597dc8c347>

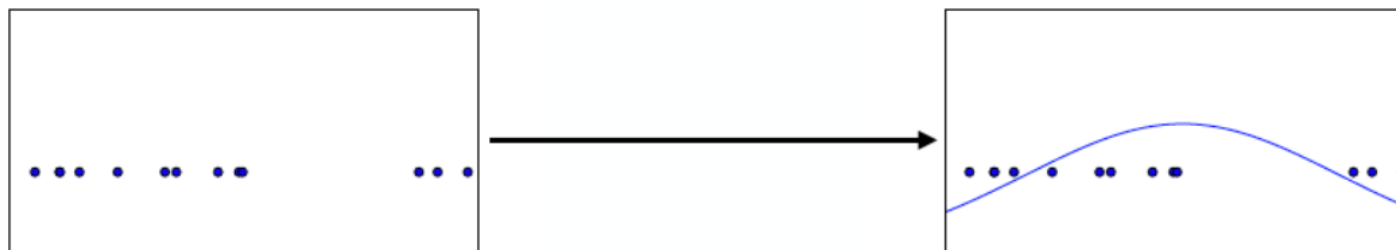
Image credit GAN Zoo

What is GAN?

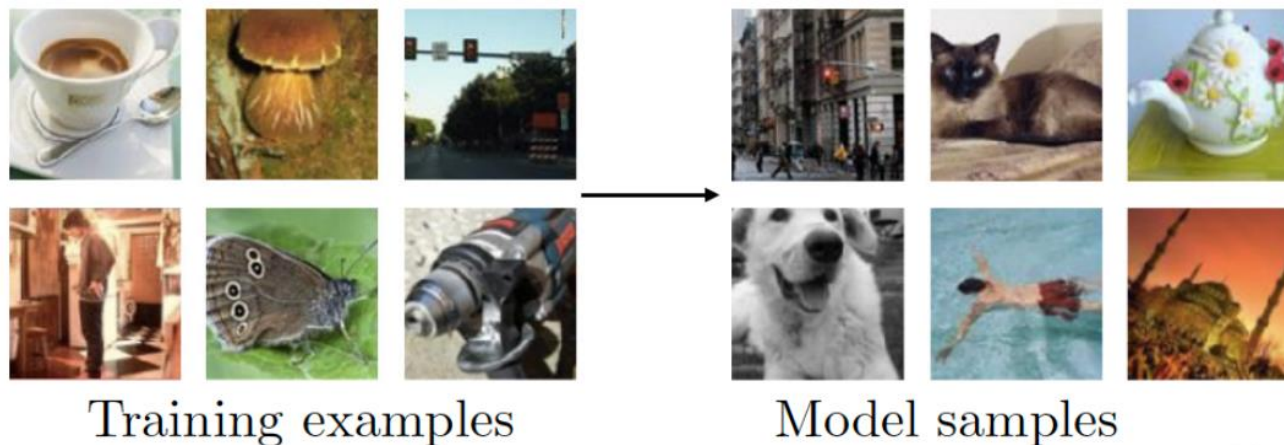
- Generative Adversarial Networks (GAN)

- A generative model \rightarrow models that are only capable of generating data rather than an estimate of the density function

- Density estimation



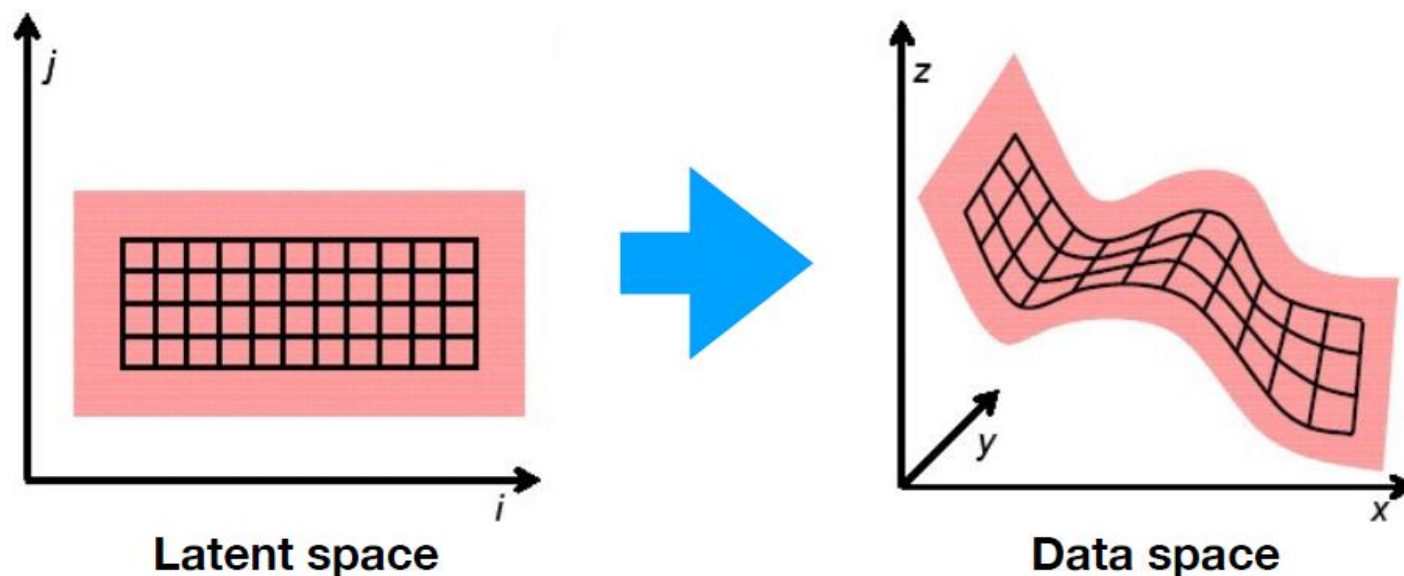
- Sample generation



Images credit Goodfellow, 2016

Manifold assumption

- Data lie approximately on a manifold much lower dimension than the input space



Slide credit Ming-Yu Liu 2017
Images credit Ward, A. D. et. al. 2007

What is GAN?

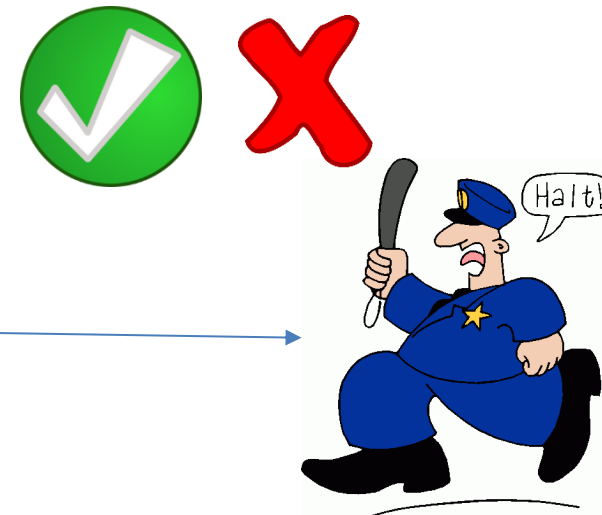
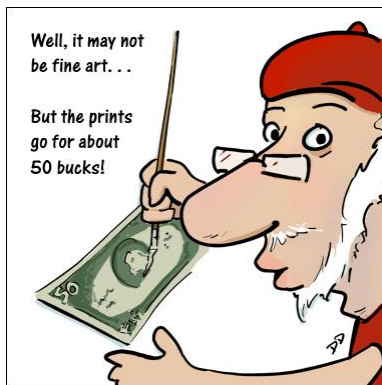
- Consists of two networks trained end-to-end against each other: Generator (G) and Discriminator (D)
- Analogy from [Goodfellow, NIPS 2014]

data distribution. The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistinguishable from the genuine articles.

[Goodfellow, NIPS 2014] – Goodfellow et. al. Generative Adversarial Networks. NIPS 2014

What is GAN? (cont.)

- In reality,
 - **Corrupted** police and **corrupted** counterfeiters



What is GAN? (cont.)

- But, there are many other deep generative models
 - Deep generative models that can learn via the principle of maximum likelihood differ with respect to how they represent or approximate the likelihood

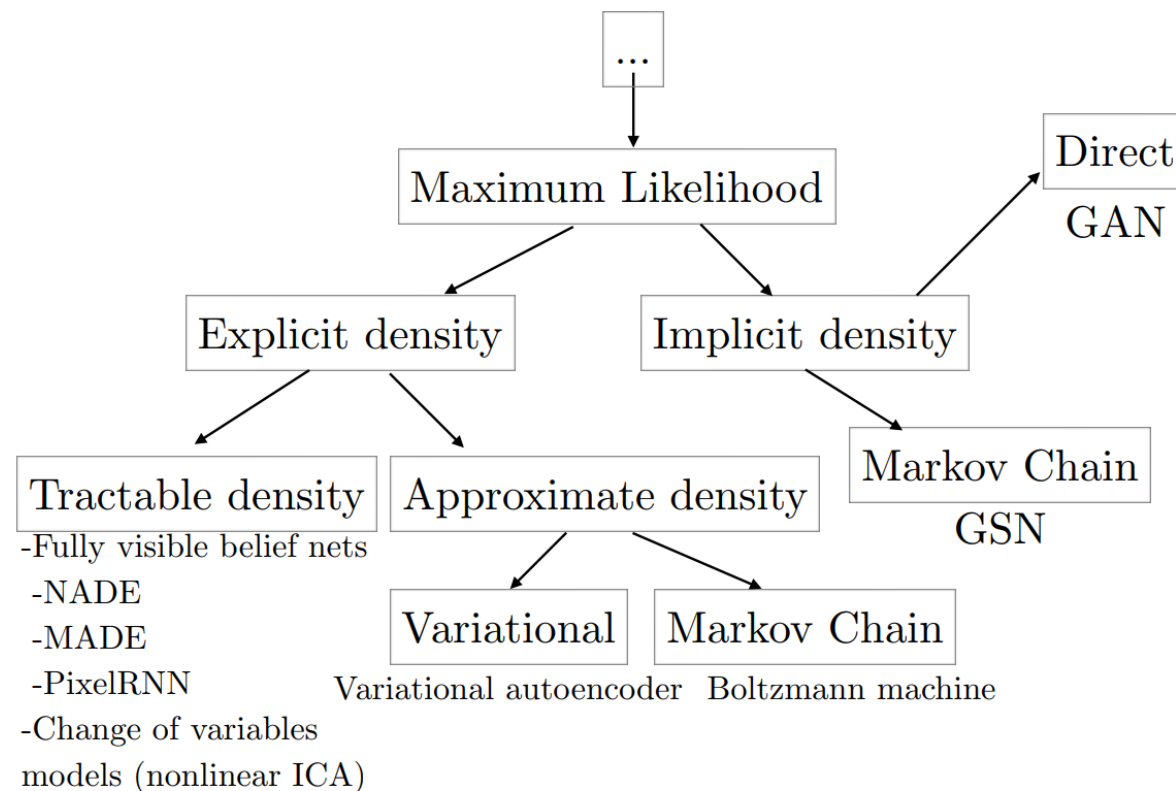
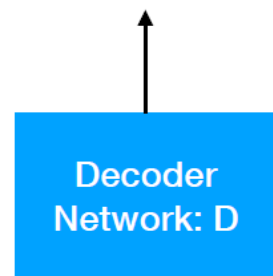


Image credit Goodfellow 2016

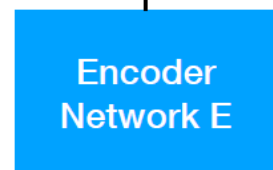
What is GAN? (cont.)

■ Generative models via autoencoders and GMM

Reconstructed image



Latent code



Input image

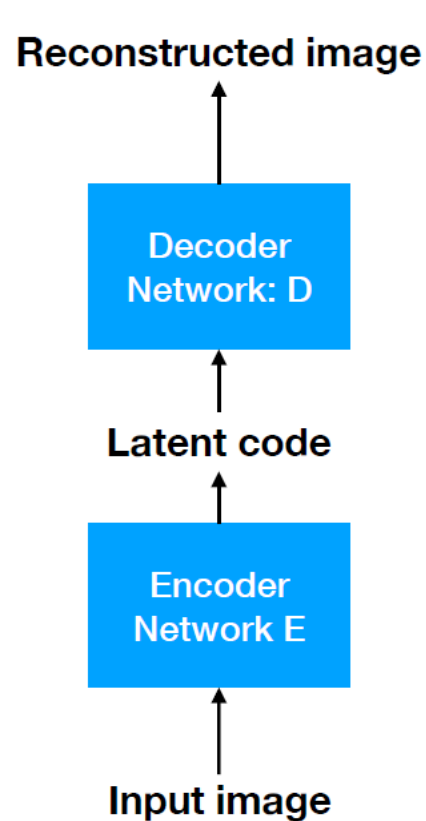
- Two-stage process
 1. Apply an auto encoder to embed images in a low dimensional space

$$\min_{E,D} \sum_{x^{(j)}} ||x^{(j)} - D(E(x^{(j)}))||^2$$

2. Fit a GMM to the embeddings $E(x^{(j)})$
- Once GMM has been learned, then a latent can be sampled and the decoder D can be used to generate a novel image
 - Shortcomings
 - Not end-to-end (two-stage process)
 - The latent code can be still high-dimensional
 - Tend to memorise samples

Slide adapted from Ming-Yu Liu 2017

What is GAN (cont.)



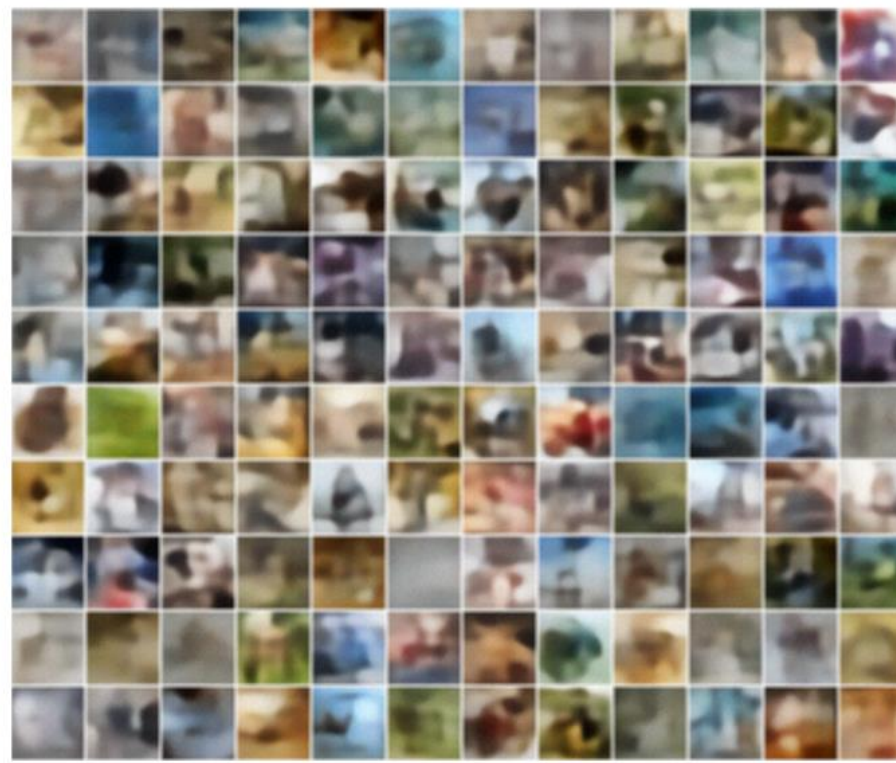
- Variational Auto encoder can be used to make the learning end-to-end
- By constraining the latent code comes from a Gaussian distribution
- The encoder is modified so it can generate Gaussian the distribution parameters (mean and covariance matrix)
- Latent code Sampling can be done from the generated distribution

What is GAN? (cont.)

- Samples from Variational AutoEncoder (VAE)



LFW dataset



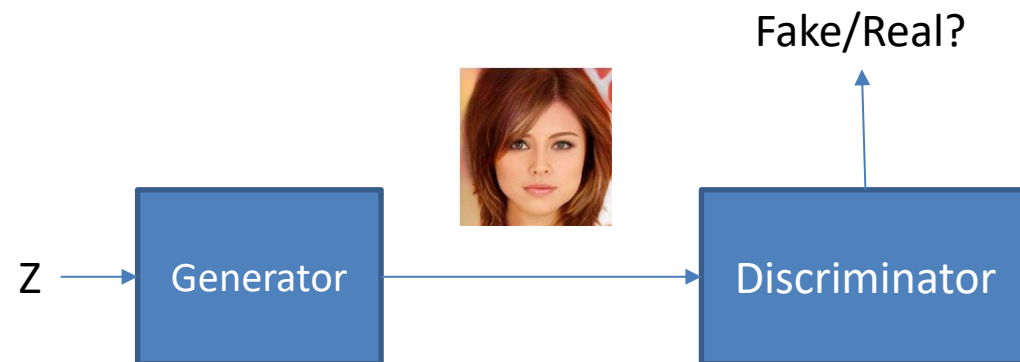
ImageNet dataset

Slide credit Courville 2017

What is GAN? (cont.)

- The Euclidean loss makes this to regress to the mean problem and rendering blurry image
- How to make the generated image more realistic?
 - Perceptual loss? But how to hand-craft this?

What is GAN? (cont.)



- GANs were designed so that the Nash equilibrium for a GAN game corresponds to recovering the real data distribution [Goodfellow, 2016]
- The discriminator can be considered as a “perceptual loss”

What is GAN? (cont.)

- GAN Objective

$$\min_G \max_D E_{x \sim p_X} [\log D(x)] + E_{z \sim p_Z} [\log(1 - D(G(z)))]$$

p_X : Data distribution,
usually represented by samples.

$p_{G(Z)}$: Model distribution, where
 Z is usually modeled as uniform or Gaussian.

What is GAN? (cont.)

- Alternating gradient updates between G and D
 - Step 1: Fix G and perform a gradient step to maximise

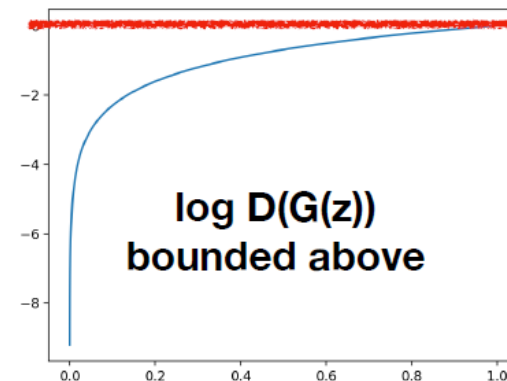
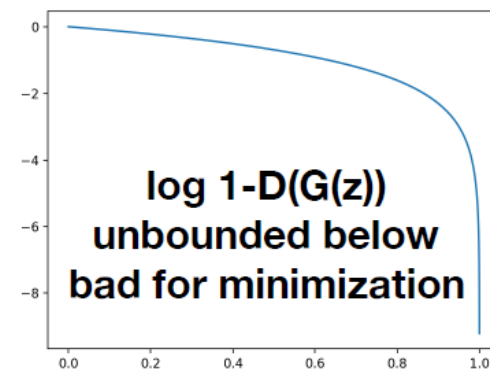
$$\max_D E_{x \sim p_X} [\log D(x)] + E_{z \sim p_Z} [\log(1 - D(G(z)))]$$

- Step 2: Fix D and perform a gradient step to
 - (in theory)

$$\min_G E_{z \sim p_Z} [\log(1 - D(G(z)))]$$

- (in practice)

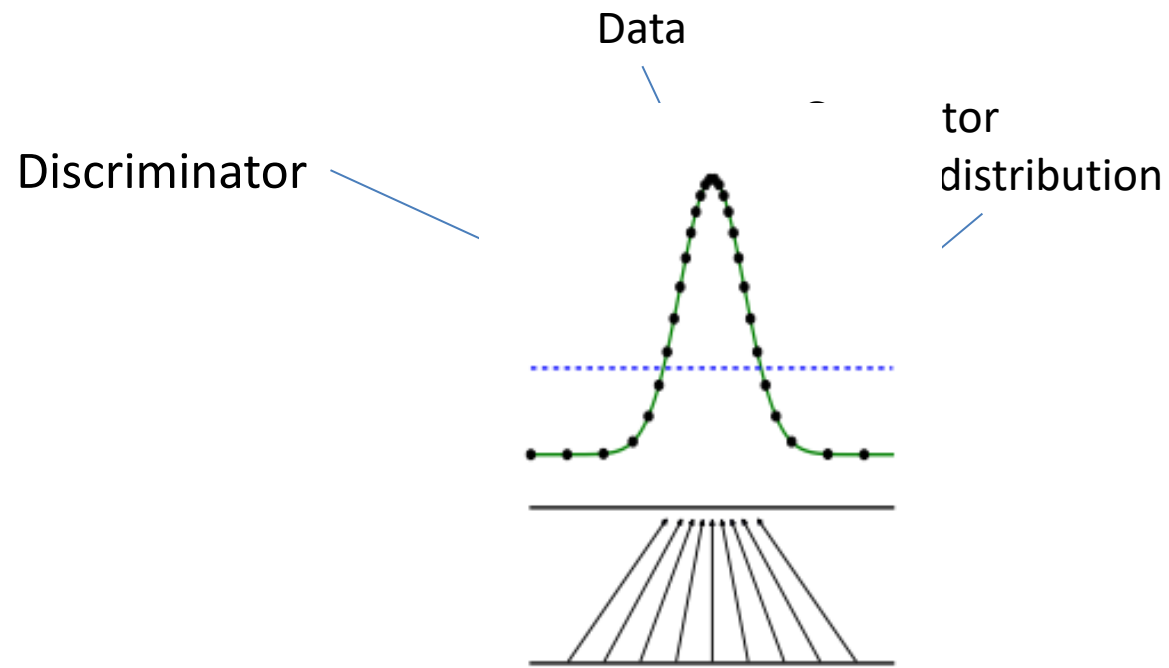
$$\max_G E_{z \sim p_Z} [\log D(G(z))]$$



Slide credit Ming-Yu Liu 2017

What is GAN? (cont.)

- Iteration 2—max G, fix D



Does GAN work?

■ Does it converge?

Theorem 1. *The global minimum of the virtual training criterion $C(G)$ is achieved if and only if $p_g = p_{\text{data}}$. At that point, $C(G)$ achieves the value $-\log 4$.*

Proof. For $p_g = p_{\text{data}}$, $D_G^*(\mathbf{x}) = \frac{1}{2}$, (consider Eq. 2). Hence, by inspecting Eq. 4 at $D_G^*(\mathbf{x}) = \frac{1}{2}$, we find $C(G) = \log \frac{1}{2} + \log \frac{1}{2} = -\log 4$. To see that this is the best possible value of $C(G)$, reached only for $p_g = p_{\text{data}}$, observe that

$$\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [-\log 2] + \mathbb{E}_{\mathbf{x} \sim p_g} [-\log 2] = -\log 4$$

and that by subtracting this expression from $C(G) = V(D_G^*, G)$, we obtain:

$$C(G) = -\log(4) + KL \left(p_{\text{data}} \left\| \frac{p_{\text{data}} + p_g}{2} \right\| \right) + KL \left(p_g \left\| \frac{p_{\text{data}} + p_g}{2} \right\| \right) \quad (5)$$

where KL is the Kullback–Leibler divergence. We recognize in the previous expression the Jensen–Shannon divergence between the model’s distribution and the data generating process:

$$C(G) = -\log(4) + 2 \cdot JSD(p_{\text{data}} \| p_g) \quad (6)$$

The training stops when D performance in recognising fake/real from G images (fake) and real images is on par with the random chance ($\frac{1}{2}$)

Has some unrealistic assumptions:

- Infinite data
- Large (maybe infinite) capacity networks

Does GAN work? (cont.)

- GAN has non-convergence problem!
 - GAN training is theoretically guaranteed to converge if we can modify the density functions directly, but!
 - Instead, we modify G (sample generation function)
 - We represent G and D as highly non-convex parametric functions
 - “Oscillation”: can train for a very long time, generating very many different categories of samples, without clearly generating better samples
 - Mode collapse: most severe form of non-convergence

Slide credit Goodfellow 2016

Does GAN work? (cont.) – Mode collapse

- GAN objective function

$$\min_G \max_D E_{x \sim p_X} [\log D(x)] + E_{z \sim p_Z} [\log(1 - D(G(z)))]$$

- The way we solve it: Alternating gradient descent (simultaneous gradient descent)
 - This strategy does not clearly privilege min max over max min or vice versa
 - We use it in the hope that it will behave like min max but it often behaves like max min → problem!

$$\min_G \max_D V(G, D) \neq \max_D \min_G V(G, D)$$

Does GAN work? (cont.) – Mode collapse

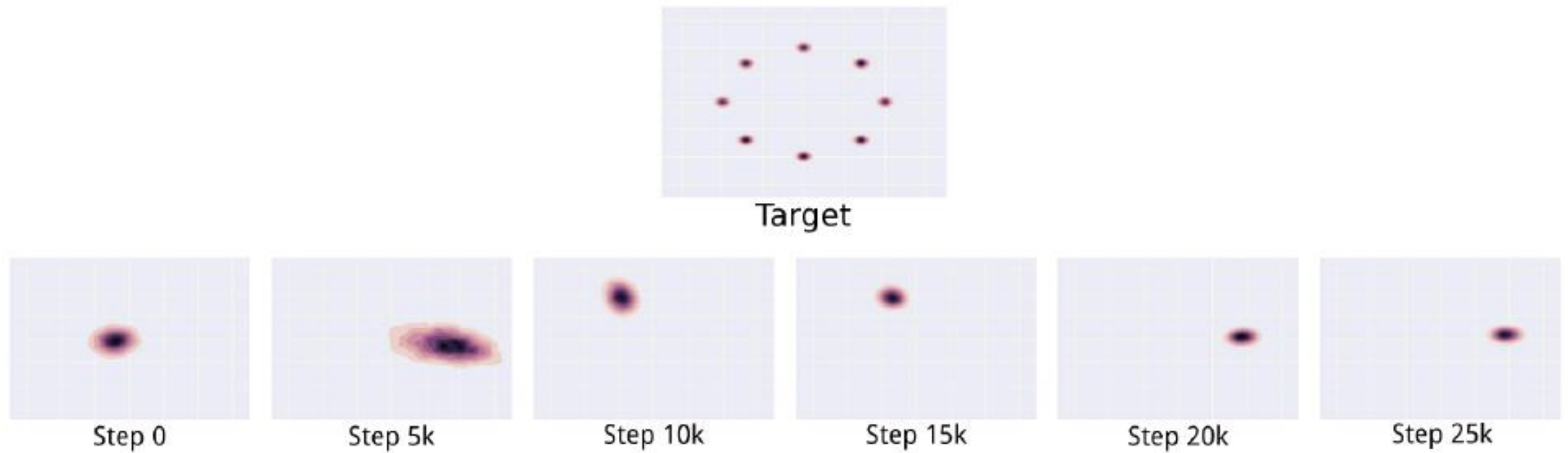
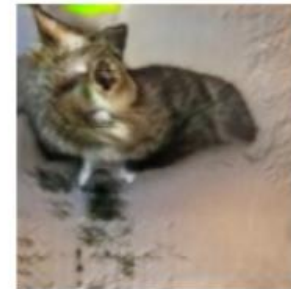


Figure credit, Metz et. al. 2016
Slide credit Goodfellow 2016

Does GAN work? (cont.)

- Many other problems – Problem with global structure

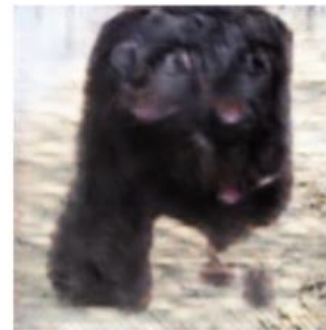


(Goodfellow 2016)

Slide credit Goodfellow 2016

Does GAN work? (cont.)

- Problem with counting



(Goodfellow 2016)

Slide credit Goodfellow 2016

Does GAN work? (cont.)

- Many works are devoted to address these open problems
 - Non-convergence in GANs
 - Addressing mode collapse
 - Tricks to get better training
 - Objective functions
 - Network architectures
 - Improving GAN image resolution
- GAN method evaluation is also a problem

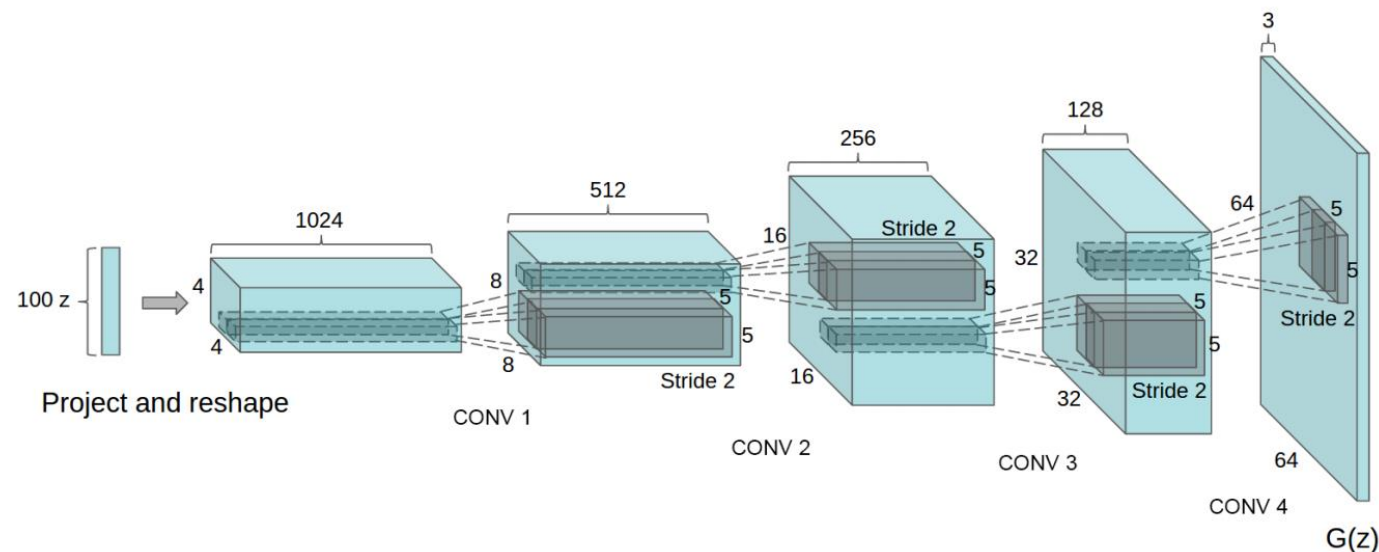
Adapted from Ming-Yu Liu 2017

- So far, we have discussed
 - GAN falls in the sample generation methods
 - Manifold assumption in GAN
 - GAN structure and how it differs compared to the other deep generative models
 - GAN convergence
 - GAN problems
 - Mode collapse → may be caused by the AGD
 - Global structure
 - Counting
 - Evaluation methods

Agenda

- Brief introduction on GAN
- Recent works in GAN
- GAN for image-to-image translation
- GAN applications for face recognition

Deep Convolutional GAN (DCGAN)



- Radford, et. al. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2016
- Stabilise GAN with some architecture constraints:
 - Pooling layer \rightarrow strided convolutions (discriminator) and fractionally-strided (generator)
 - Use batchnorm in both generator and discriminator
 - Fully connected layer \rightarrow average pooling
 - ReLu activation for all layers except the output, which uses Tanh
 - LeakyReLU for discriminator

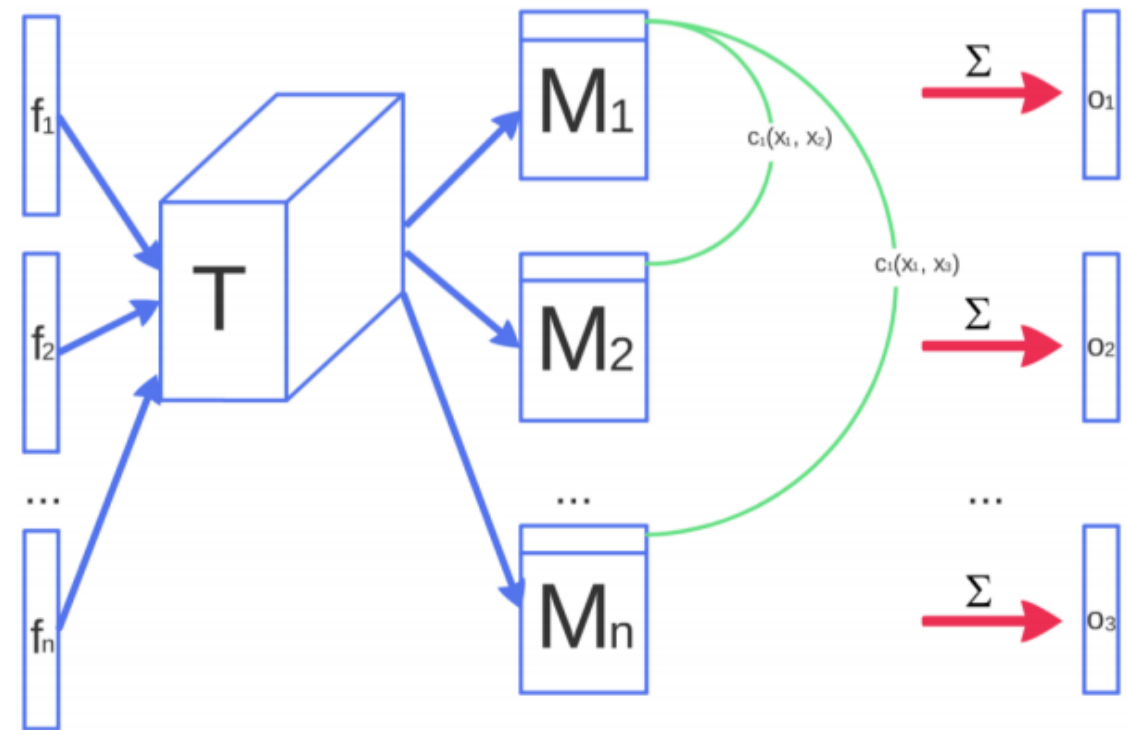
Images credit Radford 2016

Improved Techniques for Training GANs

- Salimans et. al. Improved Techniques for Training GANs, NIPS 2016
- Introduced a bag of tricks to address GAN training such as
 - One-sided label smoothing
 - Replaces 0 and 1 for a classifier with smoothed values, such as 0.9 or 0.1
 - This will not reduce the discriminator performance, but only the confidence
 - When using the traditional GAN objective function, an overly confident discriminator can cause problems
 - Large confidence translates to large gradient signal for the generator → most of the time it is bad

Improved Techniques for Training GANs (cont.)

- Minibatch discrimination
 - Attempts to address mode collapse
 - In the early stage of collapse, the G generates similar images for various inputs
 - Possible solution: check the similarity of N generated images, and use this information as a side information fed into the discriminator during training

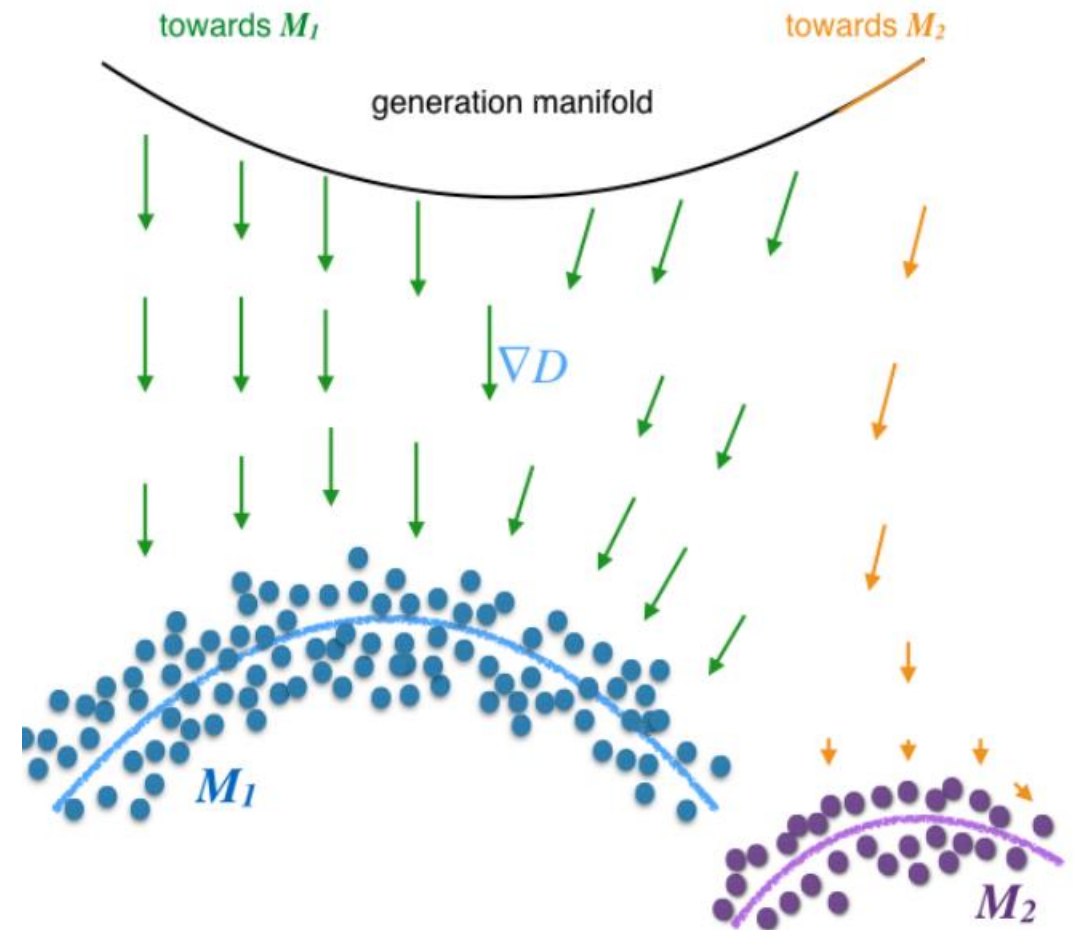


Images credit Salimans 2016

More tricks from GANHacks

- Can be found at <https://github.com/soumith/ganhacks>
 - Use ADAM Optimizer!
 - Kingma and Ba. Adam: A Method for Stochastic Optimization. ICLR 2015

- Che et. al. Mode Regularized Generative Adversarial Networks, ICLR 2017
- Tackles mode collapse and mode missing
- Mode missing \rightarrow modes represented with small amount of training data will not be generated by the generator G
- The discriminator functional shape has significant factor to GAN training
 - Discriminator is used as the training metric
 - Using it as the sole metric could be bad as there is **no direct control** to its functional shape



Images credit Che 2017

■ Solution

■ Use Autoencoder idea as a regulariser

- Encoder that encodes an image into the space of vector z
- To control the discriminator functional shape → addressing mode collapse
 - In most cases, data manifold and generated image manifold are disjoint!
 - By using an autoencoder we can measure the distance
 - The distance between the real sample and the generated sample produced by an encoder should be small
 - This is equivalent to first training a point to point mapping between two manifolds and then trying to minimize the expected distance between the points on these two manifolds
- To control the mode missing problem
 - Make the discriminator visits all the modes

Controls Discriminator functional shape Controls mode missing problem

$$T_G = -\mathbb{E}_z[\log D(G(z))] + \mathbb{E}_{x \sim p_d}[\lambda_1 d(x, G \circ E(x)) + \lambda_2 \log D(G \circ E(x))] \quad (1)$$

$$T_E = \mathbb{E}_{x \sim p_d}[\lambda_1 d(x, G \circ E(x)) + \lambda_2 \log D(G \circ E(x))] \quad (2)$$

- Arjovsky et. al. Wasserstein GAN, 2016
- Main problem: The KL divergence used in the original GAN formulation is not defined (becomes infinity) when real data and the generated data manifolds are not overlapped

GAN: minimize Jensen-Shannon divergence between p_X and $p_{G(Z)}$

$$JS(p_X || p_{G(Z)}) = KL(p_X || \frac{p_X + p_{G(Z)}}{2}) + KL(p_{G(Z)} || \frac{p_X + p_{G(Z)}}{2})$$

- Solution: use different distance to better measure the differences between these two manifolds/distributions → Earth mover distance

$$EM(p_X, p_{G(Z)}) = \inf_{\gamma \in \Pi(p_X, p_{G(Z)})} E_{(x,y) \sim \gamma} [|x - y|]$$

WGAN (cont.)





- It allows the discriminator is fully trained before training the generator
- Meaningful loss metric → could be used as a guide in training

Images credit Arjovsky 2016

- Mao et. al. Least Squares Generative Adversarial Networks, 2016

Still use a classifier but replace cross-entropy loss with Euclidean loss.

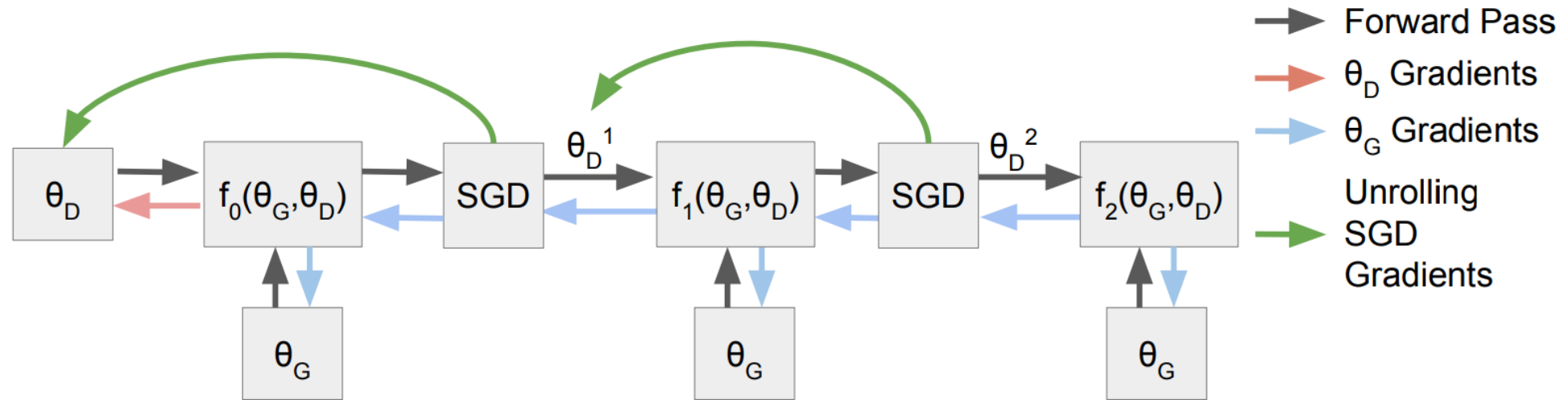
Discriminator	
GAN	$\min_D E_{x \sim p_X} [-\log D(x)] + E_{z \sim p_Z} [-\log(1 - D(G(z)))]$
	
LSGAN	$\min_D E_{x \sim p_X} [(D(x) - 1)^2] + E_{z \sim p_Z} [D(G(z))^2]$
Generator	
GAN	$\min_G E_{z \sim p_Z} [-\log D(G(z))]$
	
LSGAN	$\min_G E_{z \sim p_Z} [(D(G(z)) - 1)^2]$

61

Slide credit Ming-Yu Liu, 2017

- Metz et. al. Unrolled Generative Adversarial Networks, ICLR 2017
- We want to solve the minimax $\min_G \max_D V(G, D)$
- We do **not** want to solve maximin $\max_D \min_G V(G, D)$
- Recall the issue in the Alternating Gradient Descend (AGD) that in practice could solve the maximin not the minimax
- UnrolledGAN attempts to address this by solving the $\max_D V(G, D)$, before solving the generator minimisation

UnrolledGAN (cont.)

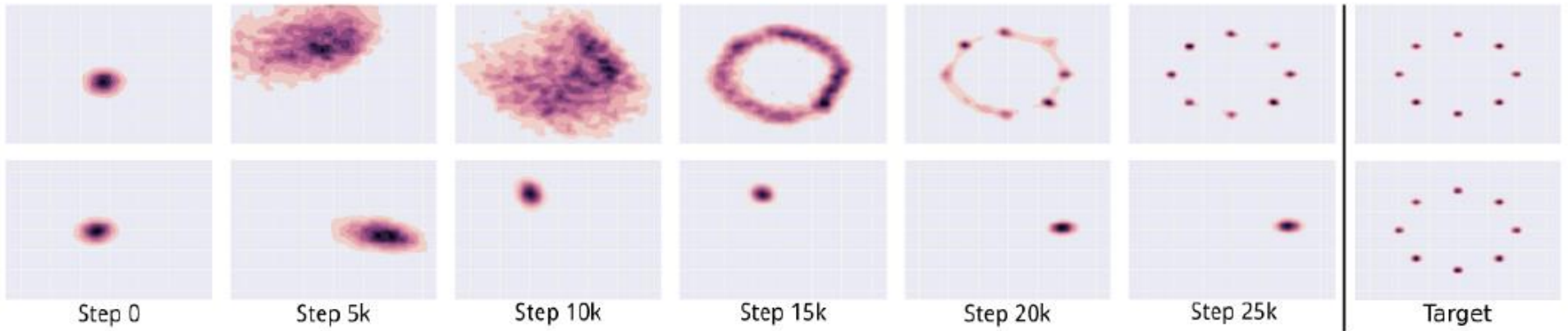


- The generator is updated by considering the future updates of the discriminator
- The trick is to use a surrogate objective function that will allow the unrolling

Images credit Metz 2017

UnrolledGAN (cont.)

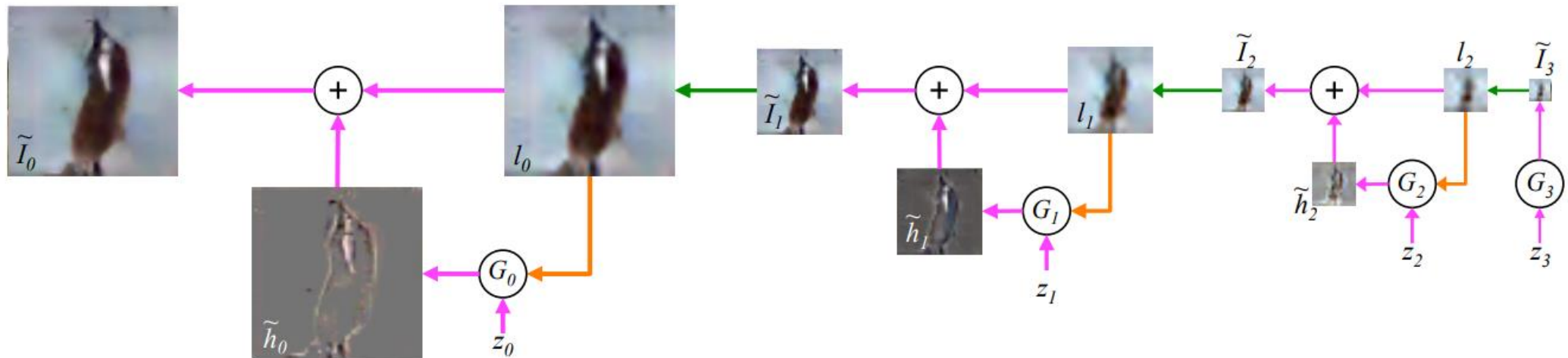
UnrolledGAN



GAN

Images credit Metz 2017

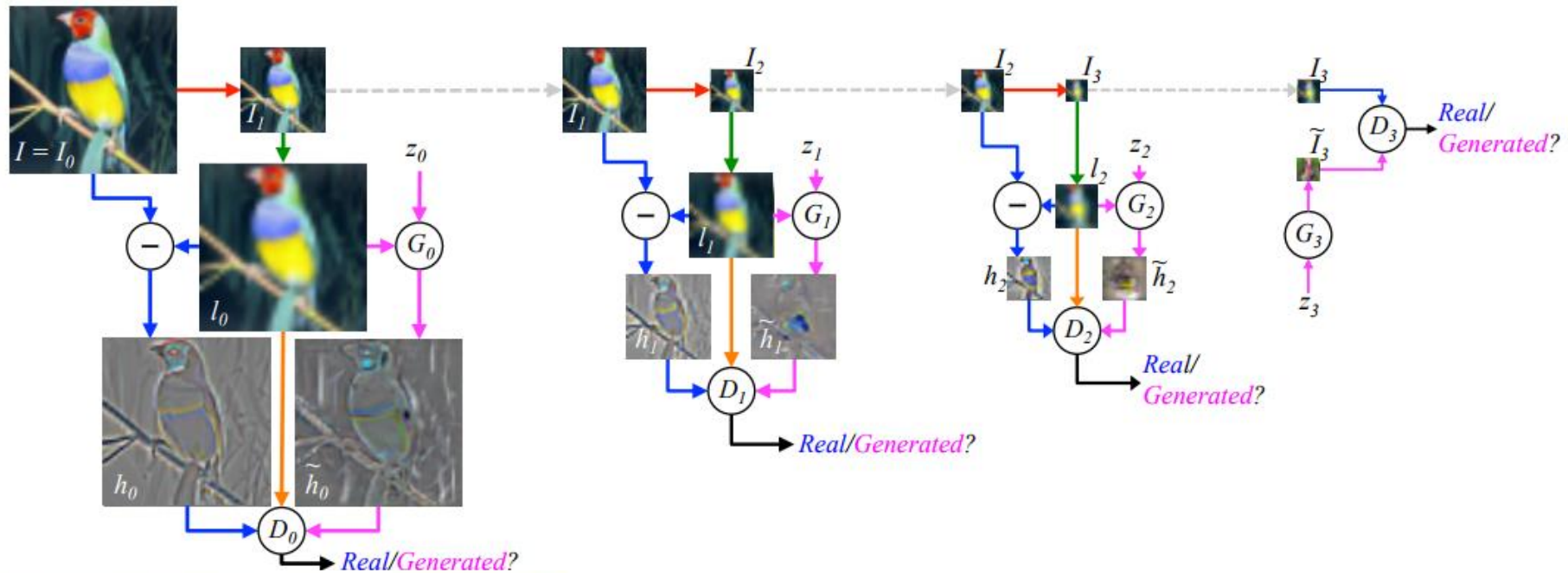
- Denton et. al. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks, NIPS 2015



- Can generate images with high resolution such as 96x96
- The generator learns the residual image which will be added into the upsampled image

Images credit Denton 2017

LAPGAN (cont.)



Images credit Denton 2015

- Shrivastava et. al. Learning from Simulated and Unsupervised Images through Adversarial Training, CVPR 2017 **best paper award!**
- Uses GAN to generate realistic training sample
 - First generate synthetic image that carries the domain information
 - Uses GAN to make the synthetic image more realistic

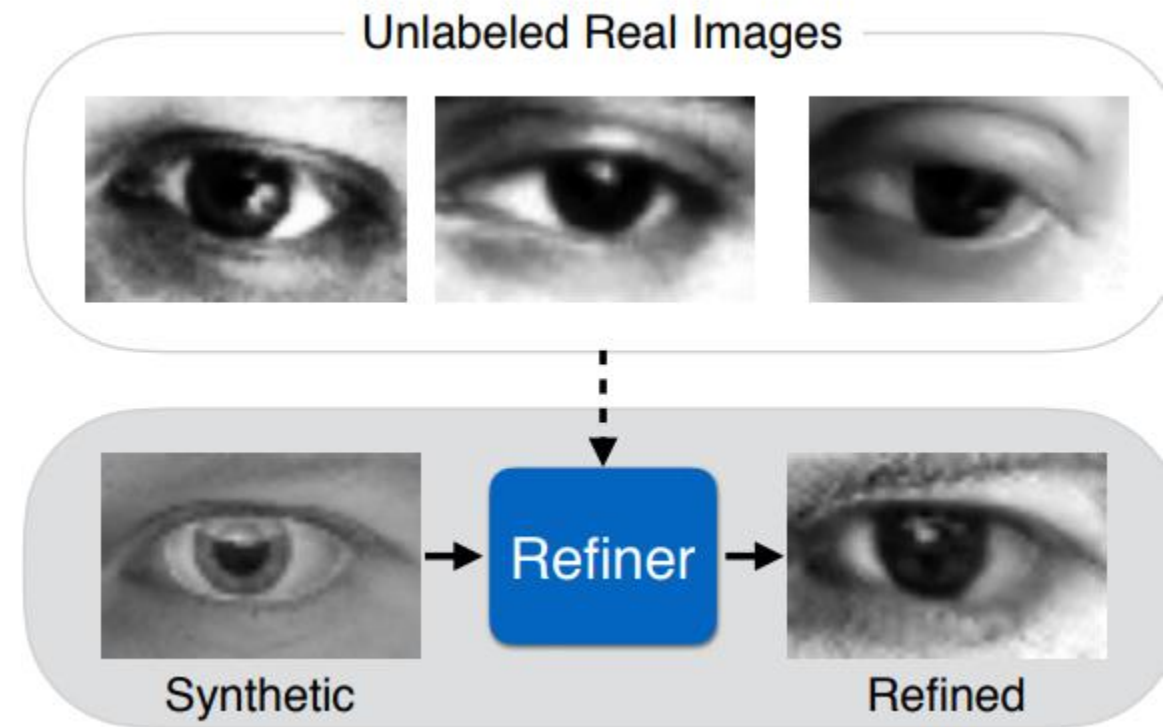


Image credit Shrivastava 2017

Recap

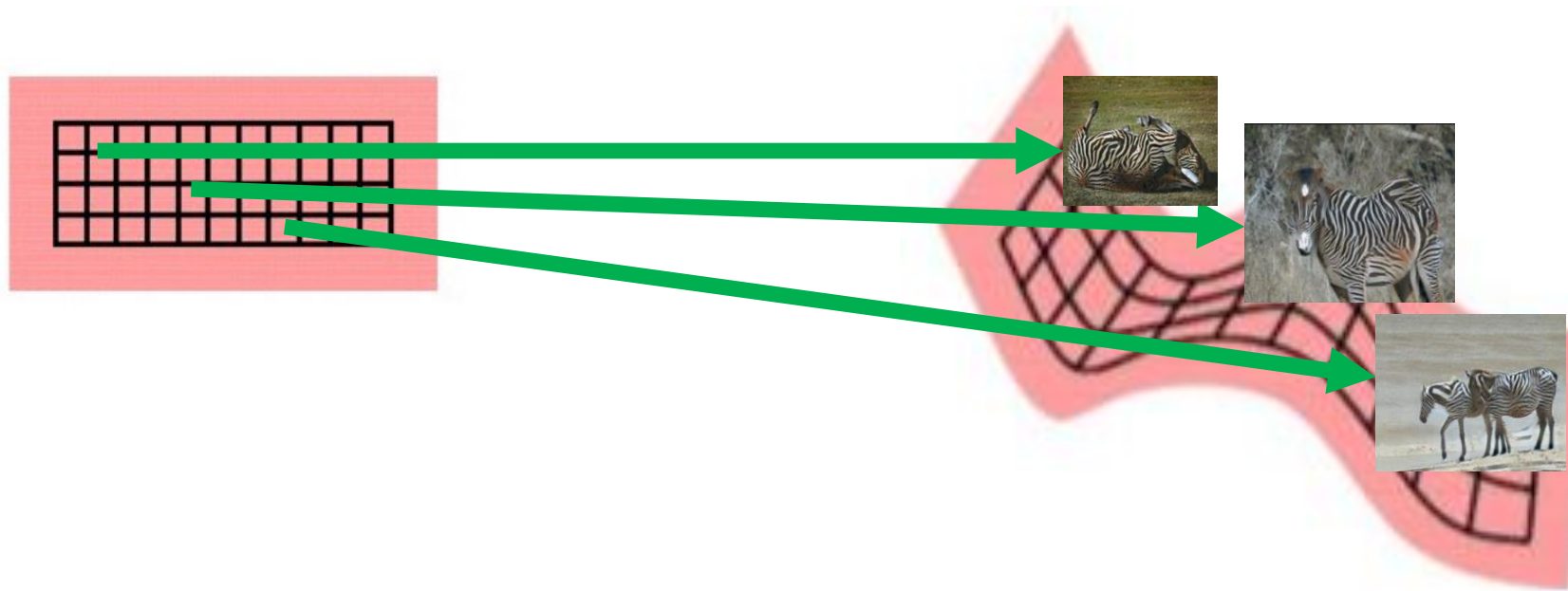
- So far, we have discussed
 - DCGAN → propose a stable architecture for GAN
 - Some tricks to address mode collapse
 - Minibatch discrimination
 - autoencoder regularisation
 - Some works propose to change the distance used in the objective function to address mode collapse and improve training stability
 - WGAN
 - LSGAN
 - UnrolledGAN → propose a more principled way of solving mode collapse by attempting to solve the GAN objective in a better way than the AGD
 - LapGAN → propose a way to increase the generated image resolution
 - SimGAN → propose to use GAN to make the training data more realistic

Agenda

- Brief introduction on GAN
- Recent works in GAN
- GAN for image-to-image translation
- GAN applications for face recognition

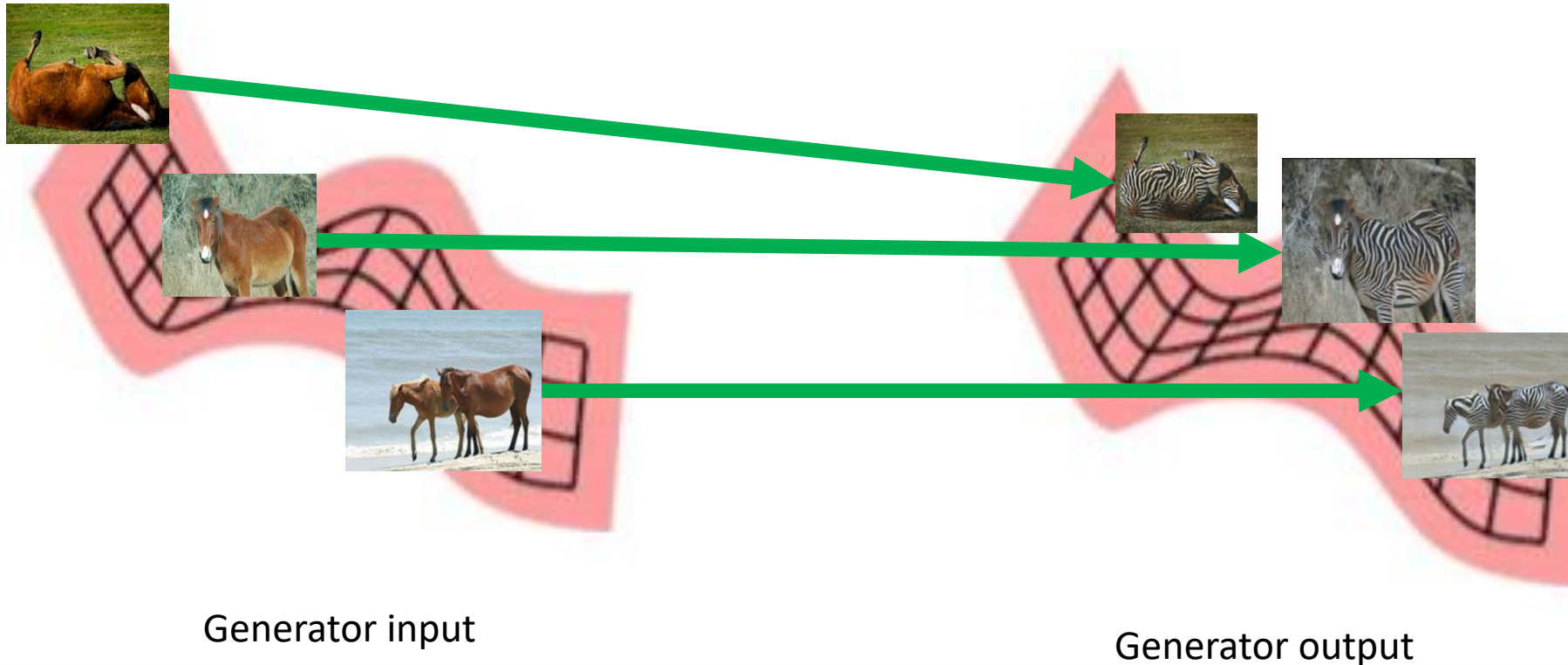
Generator input

- Initial GAN formulation defines the Generator input as a random vector z



Generator input (cont.)

- The vector z dimensionality can be increased significantly into an image



Conditional GAN

- Mirza and Osindero, Conditional Generative Adversarial Nets, 2014
- Adds additional information besides the vector z
- But, CGAN is not sufficient for image translation
- No guarantee that the conditionally generated image is related to the source image in a desired way
- In the supervised setting,
 - Dataset = $\{(x_1^{(1)}, x_2^{(1)}), (x_1^{(2)}, x_2^{(2)}), \dots, (x_1^{(N)}, x_2^{(N)})\}$
 - This can be easily addressed

Slide adapted from Ming-Yu Liu, 2017

Supervised image-to-image translation

- We have a dataset of paired images between the source image and the generated image:

$$\{(x_1^{(1)}, x_2^{(1)}), (x_1^{(2)}, x_2^{(2)}), \dots, (x_1^{(N)}, x_2^{(N)})\}$$

- Problem: What we can do to make the generator learns a reasonable translation?
 - Possible solution: add a regulariser into the learning objective

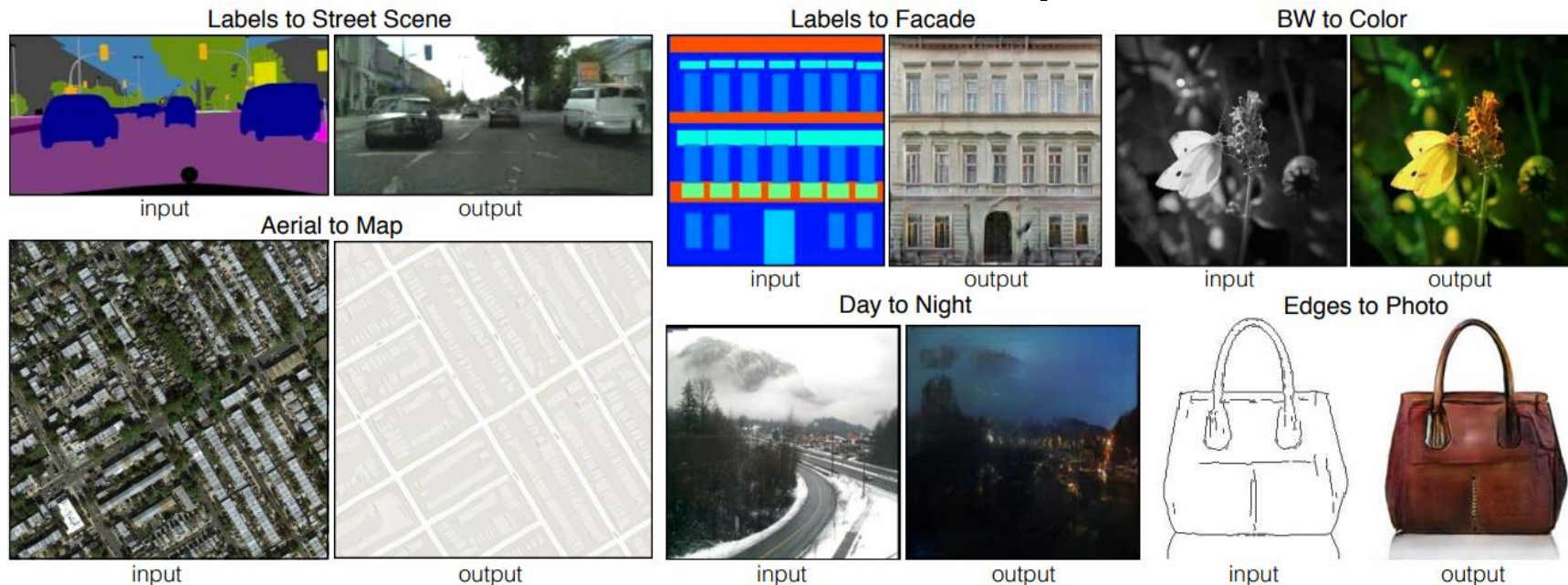
- Ledig et. al. "Photo-realistic image superresolution using a generative adversarial networks", CVPR 2017
- Added content loss

$$||x - x_2^{(i)}||_2 + ||\text{VGG}(x) - \text{VGG}(x_2^{(i)})||_2$$



Images credit Ledig 2017

- Isola et. al. Image-to-Image Translation with Conditional Adversarial Networks, CVPR 2017
- Added L1 loss: $G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$ $\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z} [\|y - G(x, z)\|_1]$
- L1 loss is used to reduce the blurry effect caused by L2 norm



Images credit Isola 2017

Unsupervised image-to-image translation

- In many cases, acquiring a dataset with paired images is not possible
 - Expensive
 - Time consuming
 - Simply impossible
- But, acquiring a dataset consisting of two sets of images from two different domains is still possible

$$Data_1 = \{x_1^n\}_{n=1}^N \quad Data_2 = \{x_2^m\}_{m=1}^M$$

- Additional assumptions/constraints are required to ensure desirable translation

- Liu and Tuzel, Coupled Generative Adversarial Networks, NIPS 2016
- Has two generators and two discriminators for each domain
- The regularisation is achieved by weight sharing between networks
- The random vector z is assumed to have high level semantic information
 - Weights from lower layers of both generators are shared
 - Weights from upper layers of both discriminators are shared

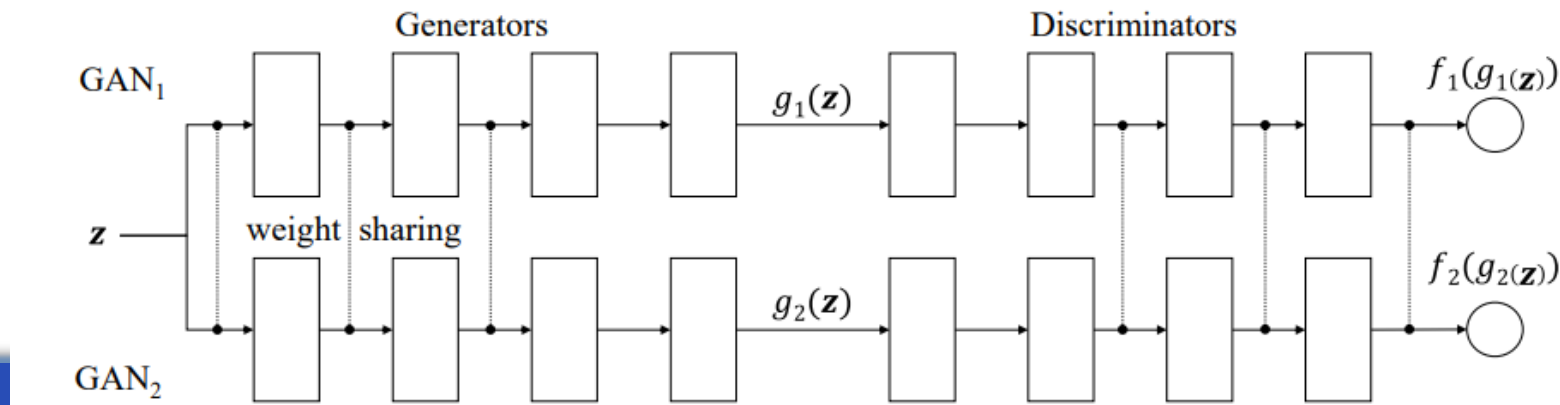
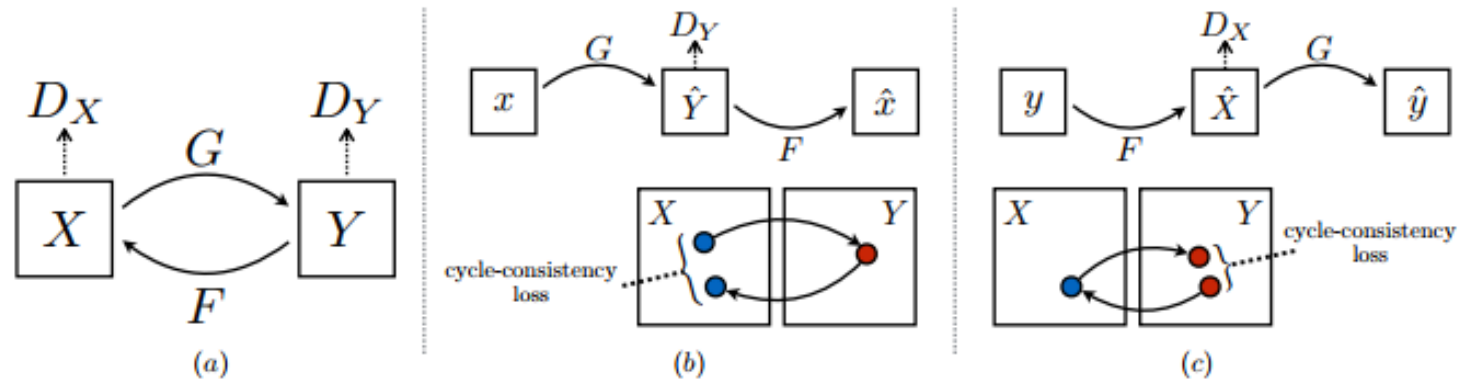


Image credit Liu 2016

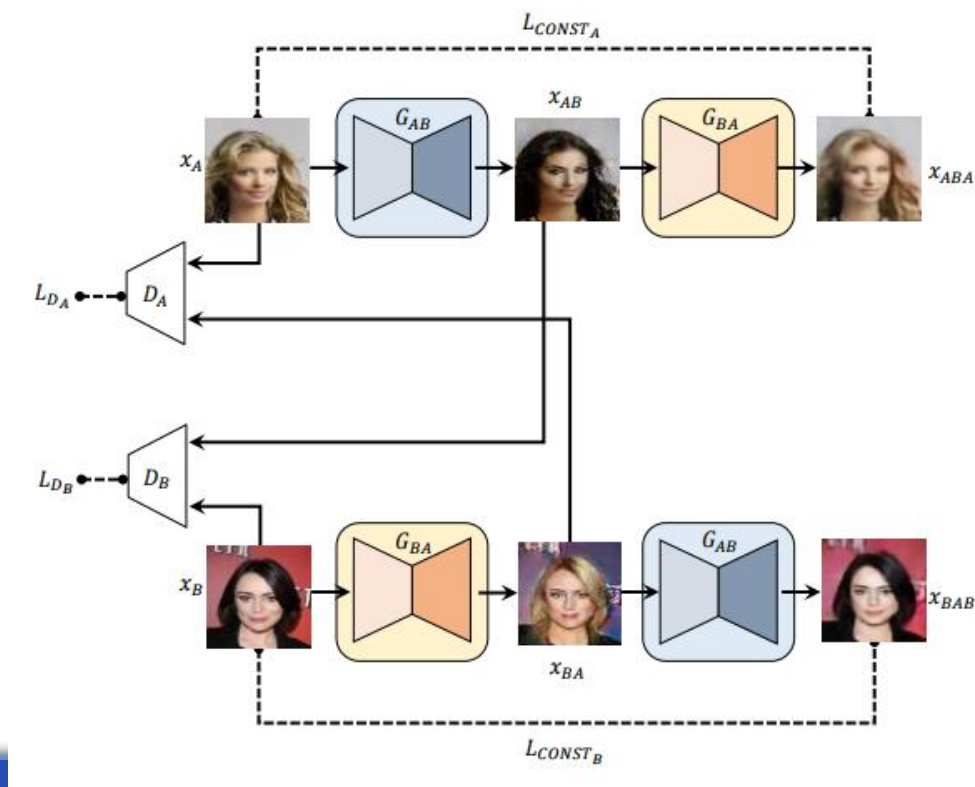
CycleGAN

- Zhu, Park et. al. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, ICCV 2017
- Has two generators, G and F . Two discriminators for the two domains

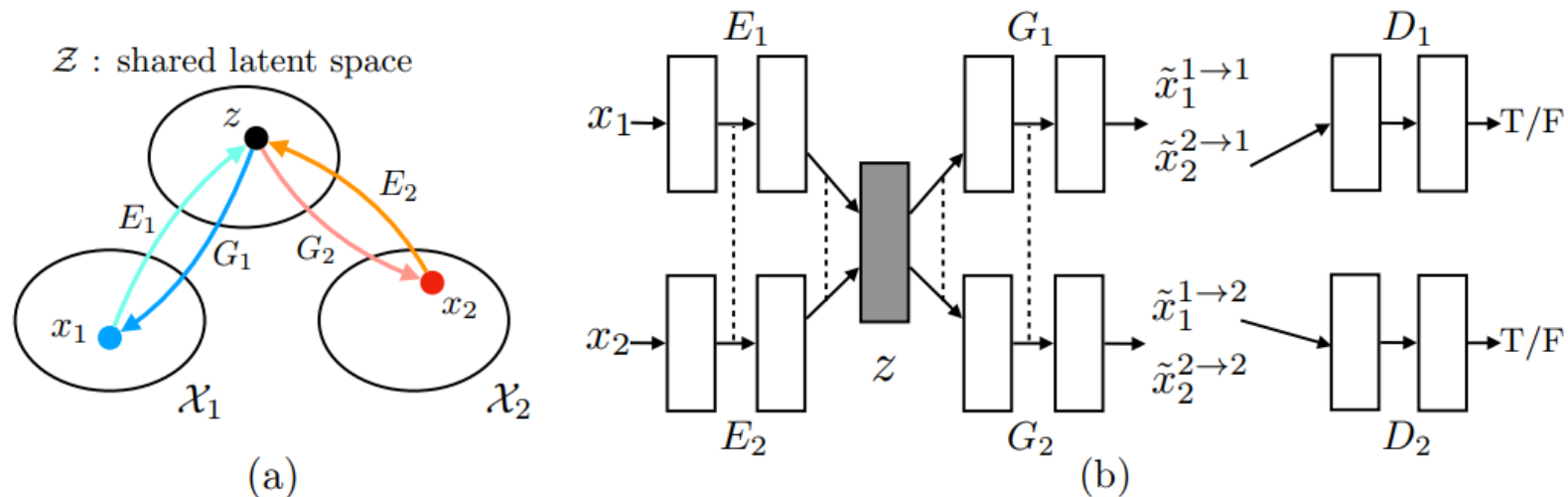


Images credit Zhu 2017

- Kim et. al. Learning to Discover Cross-Domain Relations with Generative Adversarial Networks, ICML 2017



- Liu et. al. Unsupervised Image-to-Image Translation Networks, NIPS 2017
- Shared latent space assumption \rightarrow images from different domains share the same representation in the latent space



Images credit Liu 2017

- So far, we have discussed
 - To perform image-to-image translation the random vector z is replaced by image data
 - Constraints and assumptions are normally required to achieve desirable translation results
 - There are two problems:
 - Paired/supervised image-to-image translation \rightarrow translation error can be measured
 - Unpaired/unsupervised image-to-image translation
 - Weight sharing (CoGAN)
 - Cycle consistency regularisation (CycleGAN, DiscoGAN, UnitGAN)
 - Shared latent space assumption (UnitGAN)

Agenda

- Brief introduction on GAN
- Recent works in GAN
- GAN for image-to-image translation
- GAN applications for face recognition

Possible GAN applications in face recognition domain

- Data scarcity issue
 - Improving your face recognition model accuracy
 - Add more variations to faces either in query or gallery set
 - Add more fake face images
 - Heterogenous face recognition
 - Utilising the existing face recognition model to perform recognition to the other domain
- Privacy preservation issue
 - Make your facial attribute invisible
 - Fool the existing face recognition system

- Tran et. al. Disentangled Representation Learning GAN for Pose-Invariant Face Recognition, CVPR 2017
- Learns identity representation invariant to pose
- The identity representation is combined with pose code to generate the face image with the pose specified by the pose code

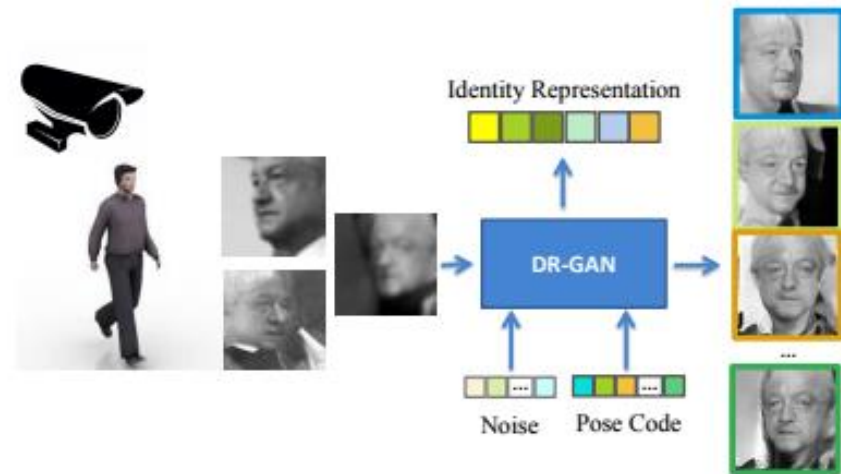
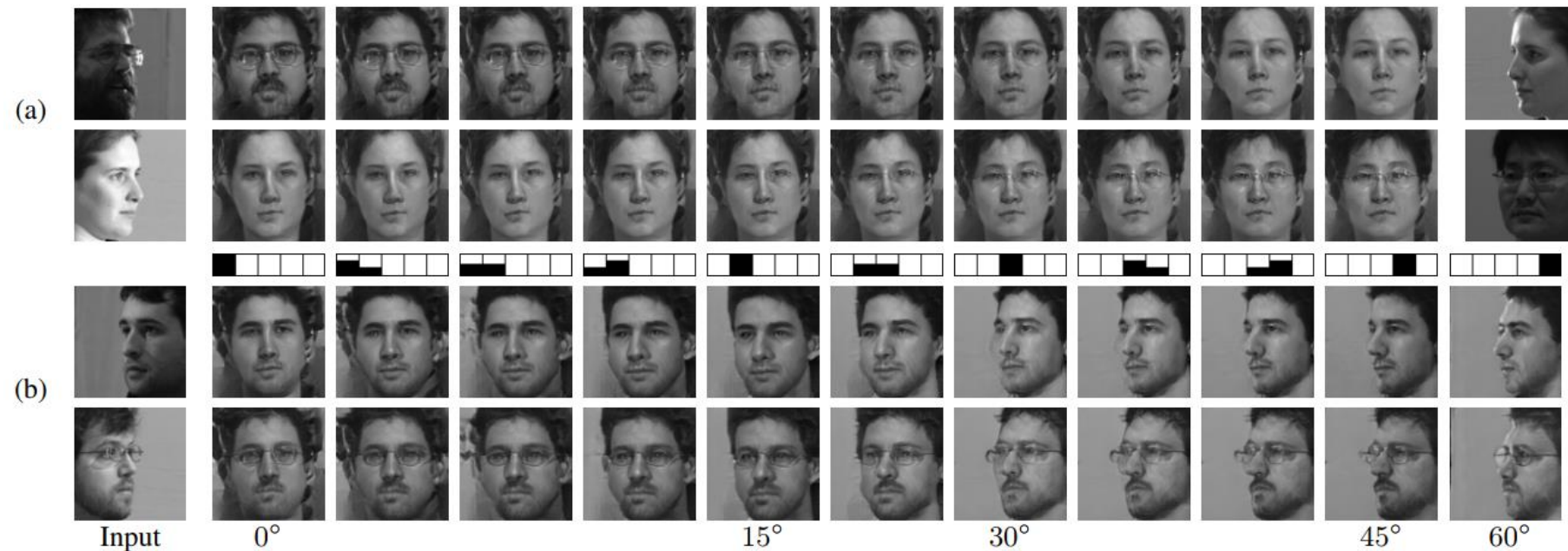
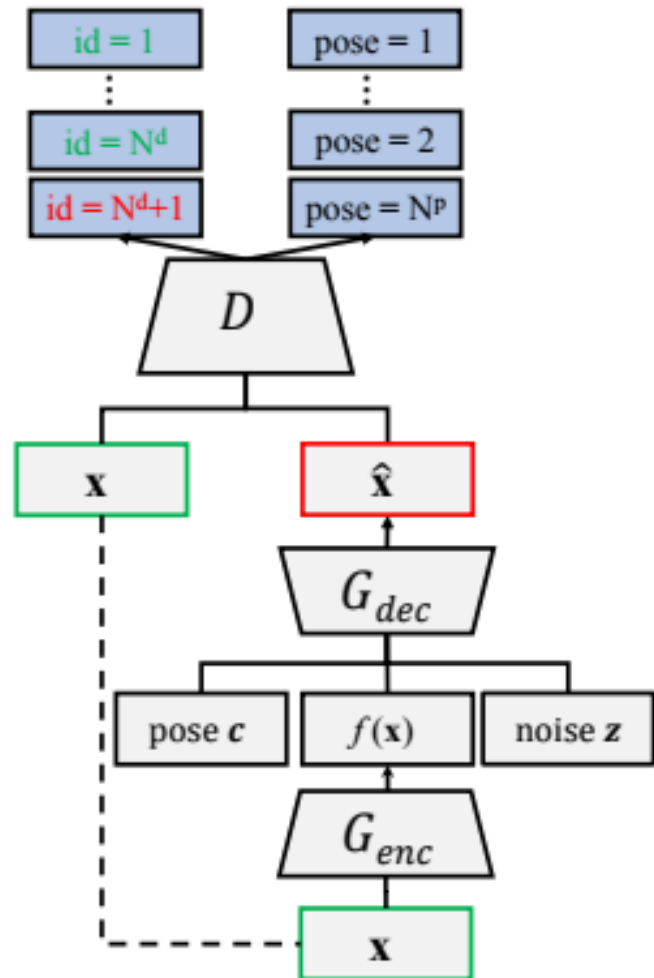


Image credit Tran 2017

DR-GAN (cont.)

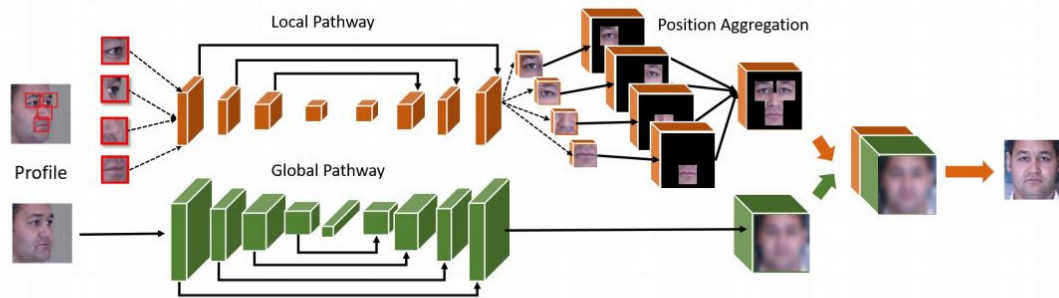


Images credit Tran 2017

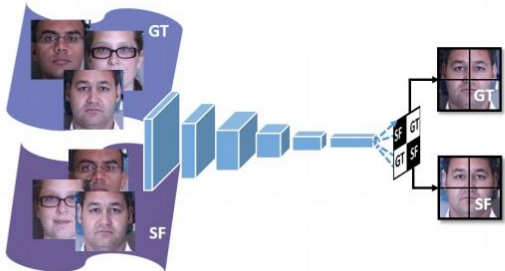
TP-GAN

- Huang et. al. Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis, ICCV 2017
- Uses two GANs: for local and global transformation
- Many constrains imposed in the training including the identity preserving loss

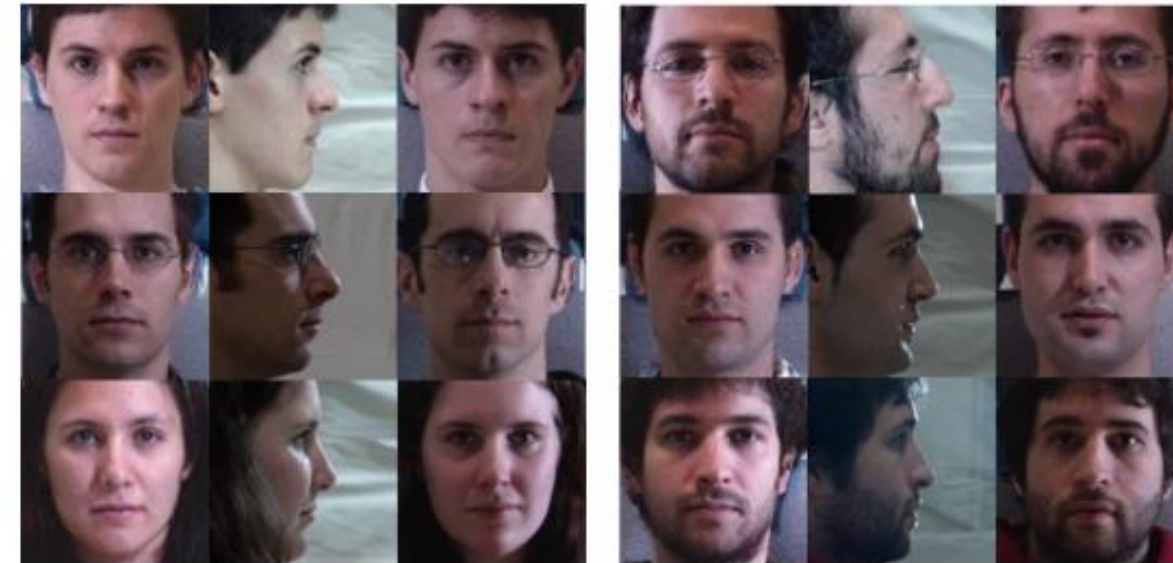
Two-pathway Generator Network



Discriminator Network



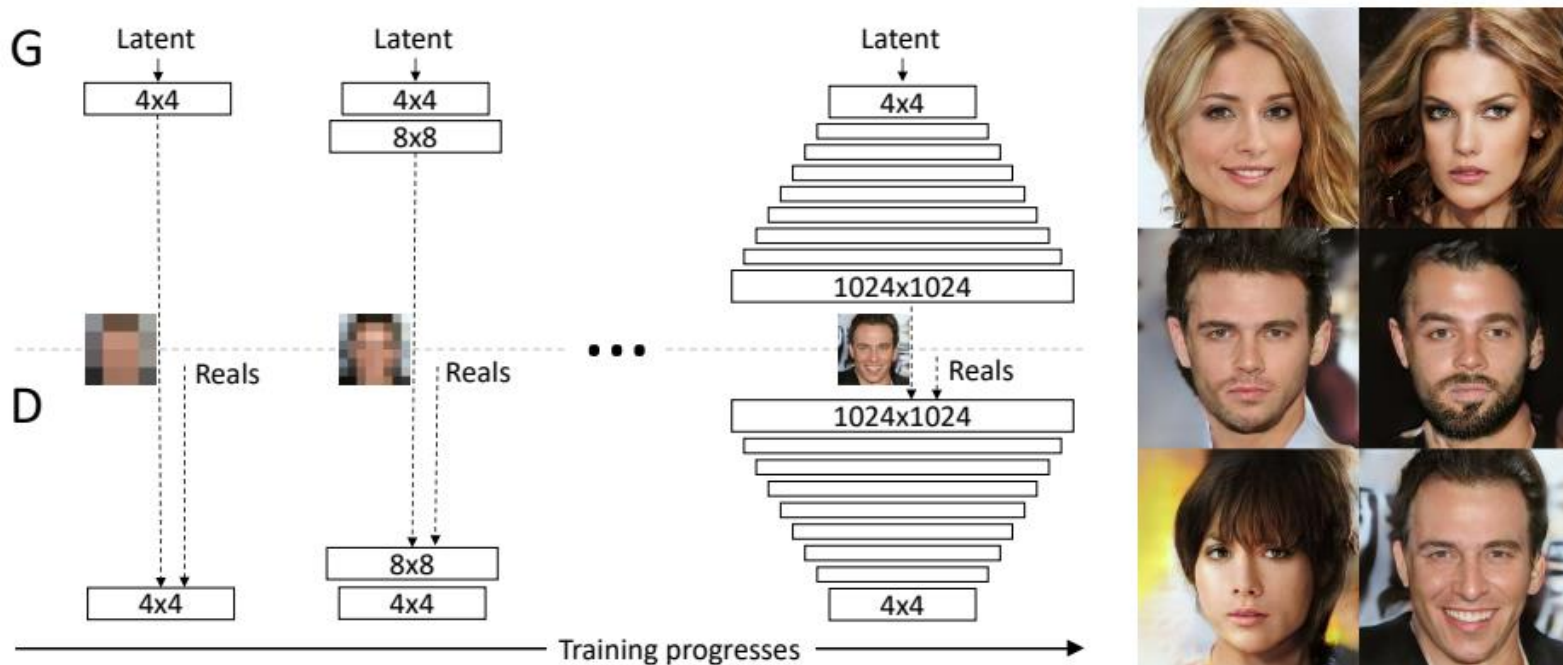
Recognition via Generation



Images credit Huang 2017

Generating high resolution fake face images

- Karras et. al. Progressive Growing of GANs for Improved Quality, Stability, and Variation, ICLR 2018 **oral**
- Uses a novel training methodology by progressively training different resolution of GANs → 2-6 times faster training time



Images credit Karras 2018

Face generation for low-shot learning

- Choe et. al. Face Generation for Low-shot Learning using Generative Adversarial Networks, ICCV Workshop 2017
 - N-shot learning \rightarrow N training samples per class
 - Low-shot learning \rightarrow small training samples per class
 - 1-shot learning \rightarrow one training sample per class
 - Zero-shot learning \rightarrow no training sample per class
- Low-shot learning problem in deep learning
 - Base training, B dataset which has many samples per class
 - Low-shot novel training, Ls dataset which has small samples per class
 - The classes between B and Ls do not overlap
 - Goal: to learn a model that performs good on both Base and Low-shot sets

Face generation for low-shot learning (cont.)



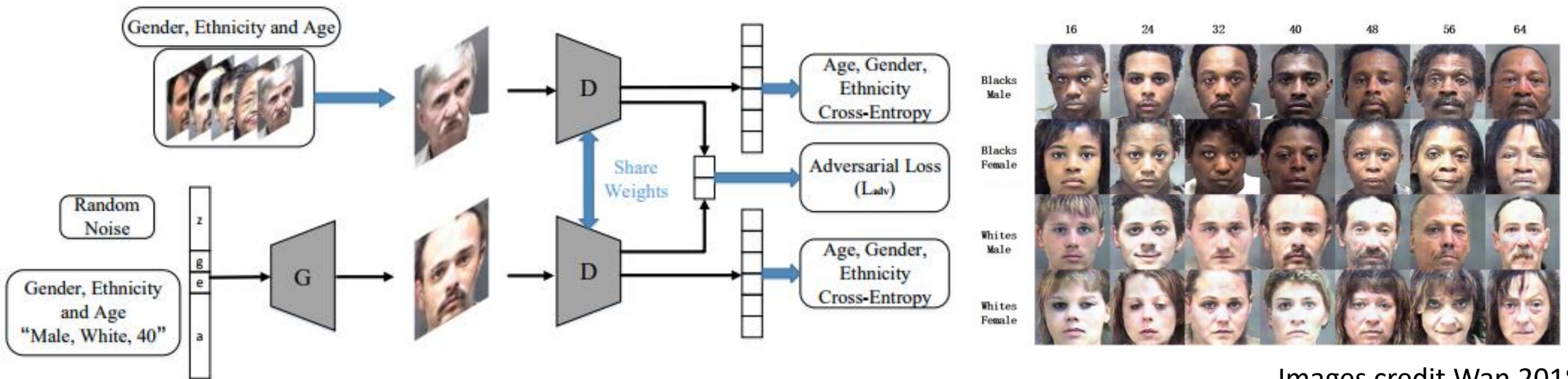
(a) Glasses attribute



(b) Smile attribute

Images credit Choe 2017

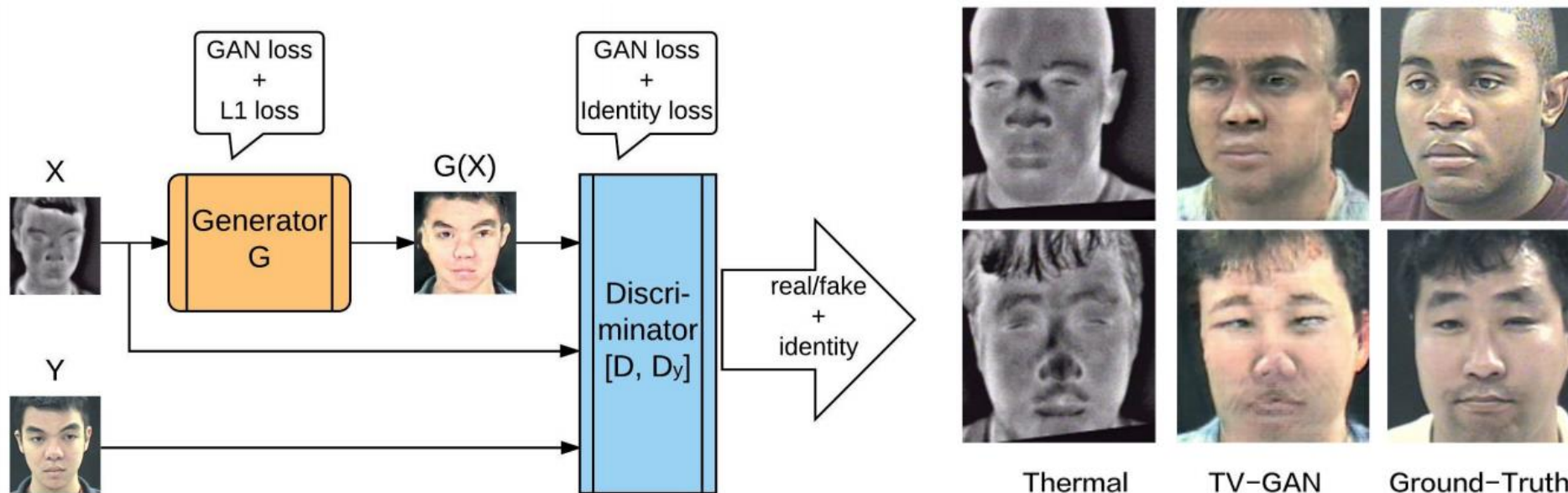
- Wan et. al. Fine-grained Multi-attribute Adversarial Learning for Face Generation of Age, Gender and Ethnicity, ICB 2018



Images credit Wan 2018

Heterogenous face recognition

- Zhang, Wiliem et. al. TV-GAN: Generative Adversarial Network based Thermal to Visible Face Recognition, ICB 2018
- To use the existing face recognition model in visible domain for thermal domain without fine tuning
 - Extend pix2pix with additional identity preserving loss
 - Tested in challenging scenarios where faces are unaligned and not normalised



Images credit Zhang 2018

TV-GAN (cont.)

- Identity preservation loss is shown to be effective. But!
 - More constraints are required to preserve other facial attributes

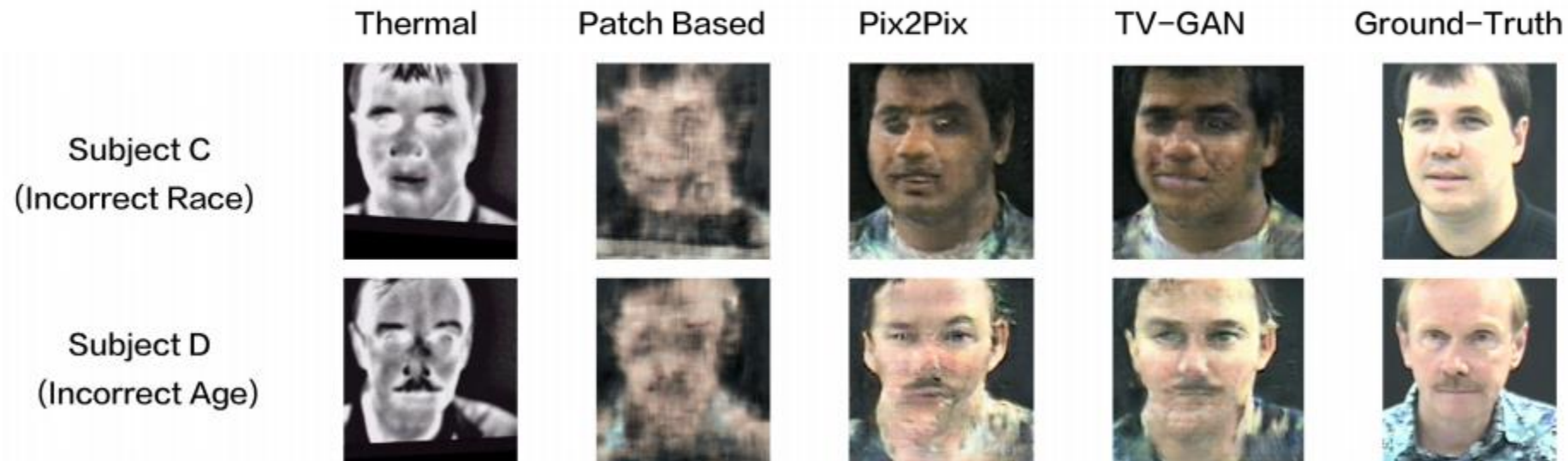


Image credit Zhang 2018

GAN for privacy

- Mirjalili et. al. Semi-Adversarial Networks: Convolutional Autoencoders for Imparting Privacy to Face Images, ICB 2018
- to impair existing face analysis system for correctly extracting facial attribute on a person face → very useful in social media

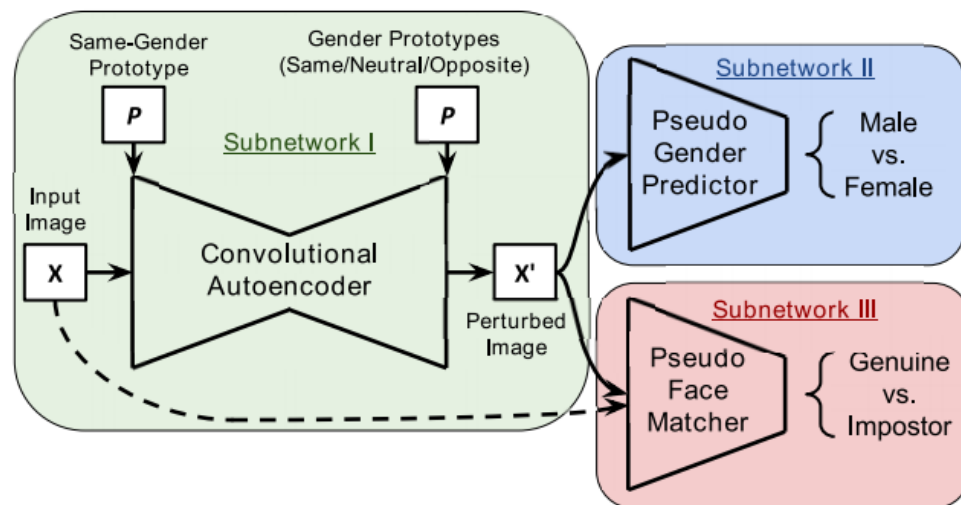


Image credit Mirjalili, 2018

- Data scarcity issue
 - Improving your face recognition model accuracy
 - Add more variations to faces either in query or gallery set
 - Add more fake face images
 - Heterogenous face recognition
 - Utilising the existing face recognition model to perform recognition to the other domain
- Privacy preservation issue
 - Make your facial attribute invisible
 - Fool the existing face recognition system

- Brief discussion about GAN and its problems
- Some recent results that are aimed to tackle these open problems
- Image-to-image translation works
- GAN for extending the capabilities of the existing face recognition systems

How to start working with GAN?

- A quick demo...