CASE SCHOOL OF ENGINEERING
CASE WESTERN RESERVE UNIVERSITY

CSDS 600: Deep Generative Models

Generative Adversarial Network (2)

Yu Yin (yu.yin@case.edu)

Case Western Reserve University

https://yin-yu.github.io/

Generative Adversarial Network (GAN)

- Improved GANs
  - Problems in GANs
  - Wasserstein GAN (WGAN)
  - WGAN -GP
- Selected GANs
  - Conditional GAN
  - CycleGAN, DualGAN, DiscoGAN
  - High-Resolution Image Generation: Progressive GAN, StyleGAN

Problems in GANs

- Mode collapse

  Generator collapse to parameters that produces the same outputs

- Vanishing gradients

  Well-trained discriminator makes gradient of generator vanished

- Hard to achieve Nash equilibrium to a two-player non-cooperative game

  Each model updates its own objective function

Wasserstein GAN (WGAN)

Wasserstein distance or Earth Mover's distance

- A measure of the distance between two probability distributions:

$$W(p_{\text{data}}, p_g) = \inf_{\gamma \in \Pi(p_{\text{data}}, p_g)} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|$$

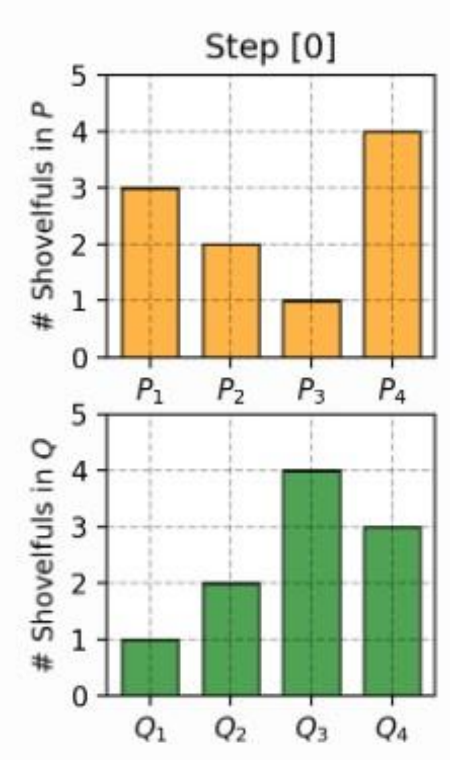
- Intuitively, minimal total amount of work to transform one heap of dirt into the other
- Work is defined as the amount of dirt in a chunk times the distance it was moved

Wasserstein GAN (WGAN)

Wasserstein distance: A Discrete Example

- $W(P,Q)$: the minimum amount of work from distribution P to Q

$$P_1 = 3, P_2 = 2, P_3 = 1, P_4 = 4$$
$$Q_1 = 1, Q_2 = 2, Q_3 = 4, Q_4 = 3$$



If we label the cost to pay to make P_i and Q_i match as δ_i , we would have :

$$\delta_{i+1} = \delta_i + P_i - Q_i$$

https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html#wasserstein-gan-wgan

Wasserstein GAN (WGAN)

Wasserstein distance: A Discrete Example

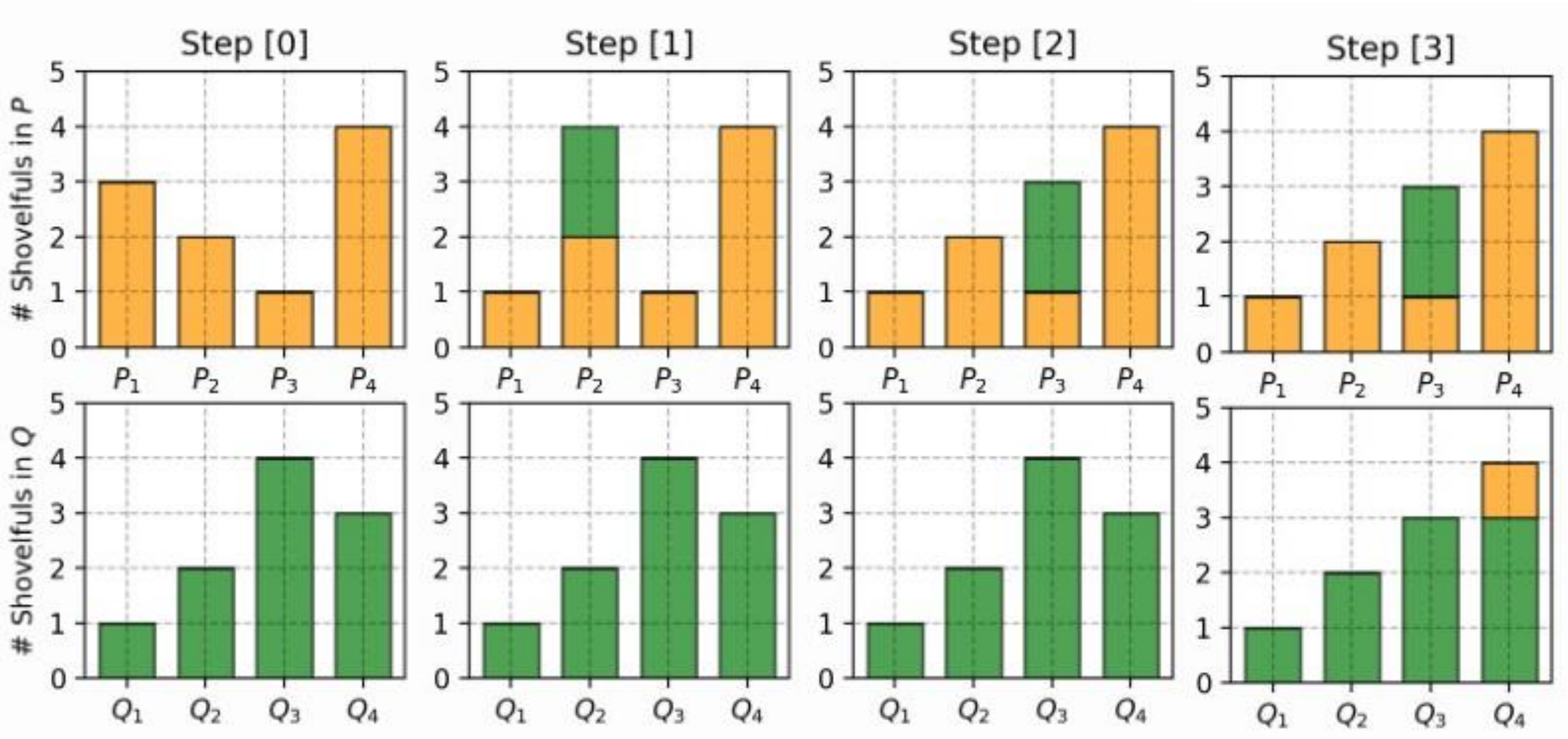
- $W(P,Q)$: the minimum amount of work from distribution P to Q

$$\delta_{i+1} = \delta_i + P_i - Q_i$$

So, we have :

$$\delta_0 = 0$$
$$\delta_1 = 0 + 3 - 1 = 2$$
$$\delta_2 = 2 + 2 - 2 = 2$$
$$\delta_3 = 2 + 1 - 4 = -1$$
$$\delta_4 = -1 + 4 - 3 = 0$$

$$W(P,Q) = 5$$



https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html#wasserstein-gan-wgan

Wasserstein GAN (WGAN)

Wasserstein distance

- In continuous probability domain

$$W(p_{\text{data}}, p_g) = \inf_{\gamma \in \Pi(p_{\text{data}}, p_g)} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|$$

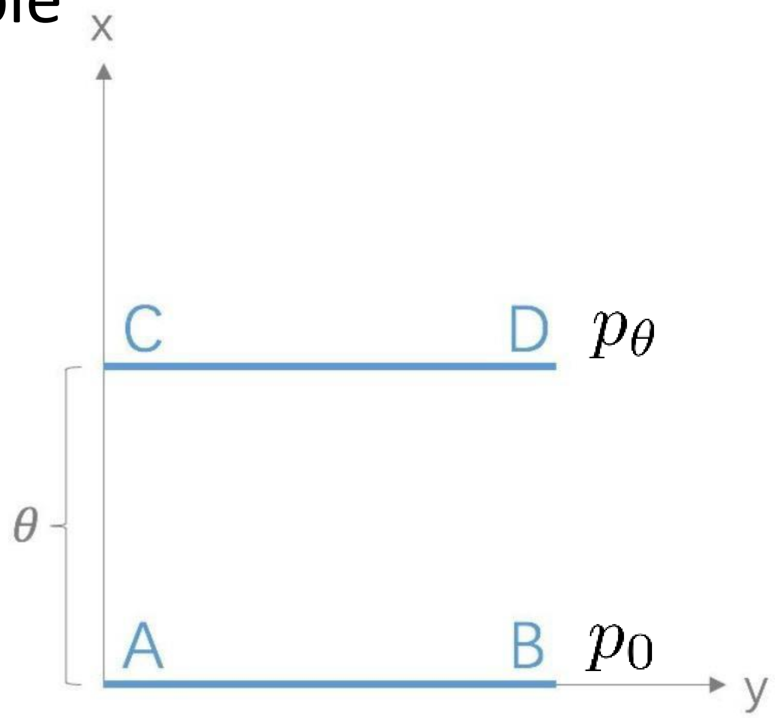
  - $\Pi(p_{\text{data}}, p_g)$ is the set of all possible joint probability distributions between p_{data} and p_g
  - Infimum over joint distribution γ (each γ corresponds to one dirt transport plan like in example in a slide before)

Wasserstein GAN (WGAN)

- Why Wasserstein is better than JS or KL divergence

  When two distributions are located without overlaps, Wasserstein distance can still provide a meaningful and smooth representation of the distance.

- Example



Distance between two distributions are:

$$KL(p_0 \parallel p_\theta) = KL(p_\theta \parallel p_0) = \begin{cases} \infty & \text{if } \theta \neq 0, \\ 0 & \text{if } \theta = 0 \end{cases}$$

$$JS(p_0 \parallel p_\theta) = \begin{cases} \log 2 & \text{if } \theta \neq 0, \\ 0 & \text{if } \theta = 0 \end{cases}$$

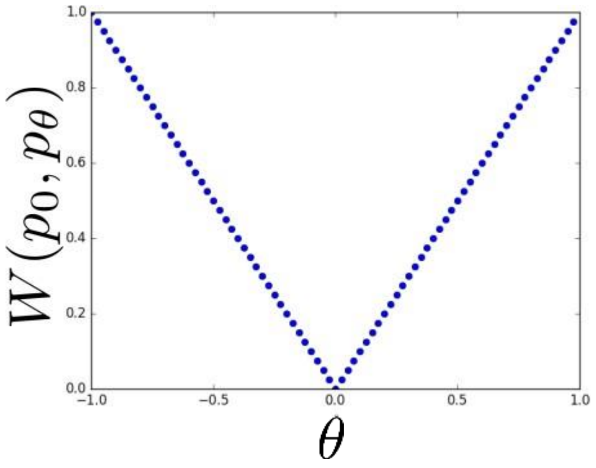
$$W(p_0, p_\theta) = |\theta|$$

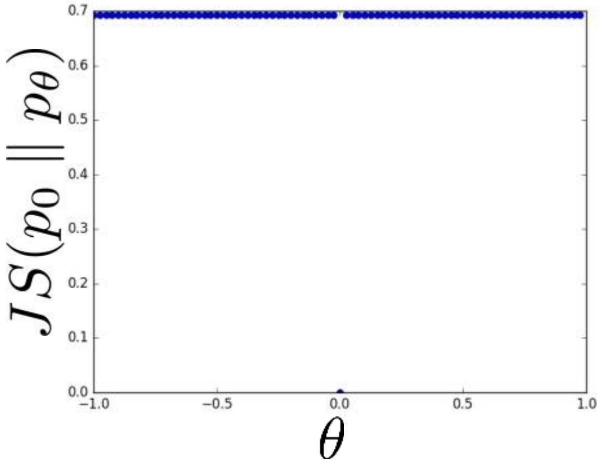
Wasserstein GAN (WGAN)

- Why Wasserstein is better than JS or KL divergence

  When two distributions are located without overlaps, Wasserstein distance can still provide a meaningful and smooth representation of the distance.

- Example





- W-distance is “better” than JSD, and JSD is “better ” than KLD.
- W-distance is a better way to measure the distance between two distributions when their support sets hardly have intersection.

Arjovsky, M. "Wasserstein generative adversarial networks." *ICML*, 2017.

Wasserstein GAN (WGAN)

- Kantorovich-Rubinstein duality
- Lipschitz Continuity
- Wasserstein GAN

- Now we attempt to design a method to minimize the W-distance between p_{data} and p_g

$$W(p_{\text{data}}, p_g) = \boxed{\inf_{\gamma \in \Pi(p_{\text{data}}, p_g)}} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|$$

Obviously, calculating the above estimation is an intractable problem.

Wasserstein GAN (WGAN)

- Kantorovich-Rubinstein duality
- Lipschitz Continuity
- Wasserstein GAN

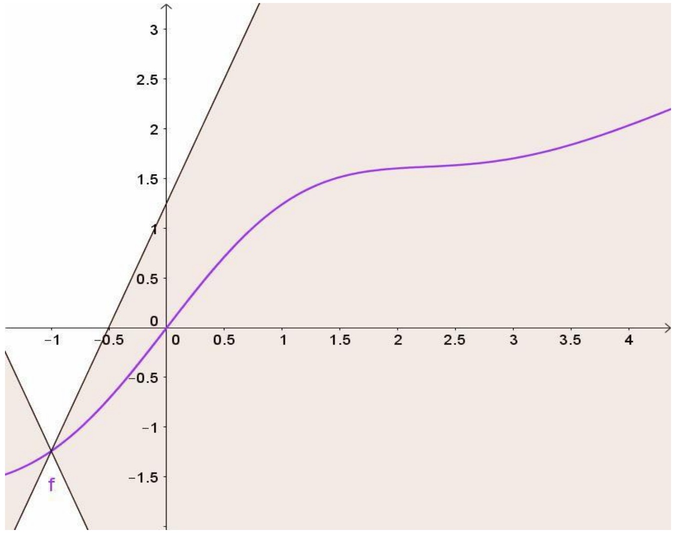
- Using Kantorovich-Rubinstein duality [Villani, 2009], Wasserstein distance becomes:

$$W(p_{\text{data}}, p_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim p_{\text{data}}}[f(x)] - \mathbb{E}_{x \sim p_g}[f(x)]$$

  - The Supremum is over all the 1-Lipschitz functions $f : \mathcal{X} \to \mathbb{R}$

Wasserstein GAN (WGAN)

- Kantorovich-Rubinstein duality
- Lipschitz Continuity
- Wasserstein GAN



- In particular, a real-valued function $f: R^n \rightarrow R$ is called Lipschitz continuous if there exists a positive real constant K such that, for all $x_1, x_2 \in R^n$:

$$|f(x_1) - f(x_2)| \leq K||x_1 - x_2||$$

- If a function is derivable and its gradient is bounded, then it is Lipschitz continuous
- To enforce the Lipschitz constraint, clamp the weights to a fixed box (e.g., $\mathcal{W} = [-0.01, 0.01]^\ell$, where ℓ is dimension of parameter $w \in \mathcal{W}$

Wasserstein GAN (WGAN)

- Kantorovich-Rubinstein duality
- Lipschitz Continuity
- Wasserstein GAN

- Now we introduce our new objective
  - To minimize $W(p_{data}||p_g) = \max_{||f||_L \le 1} \mathbb{E}_{x \sim p_{data}} f(x) - \mathbb{E}_{x \sim p_g} f(x)$
  - Equivalent to $\min_G W(p_{data}||p_g) = \min_G \max_{||f||_L \le 1} \mathbb{E}_{x \sim p_{data}} f(x) - \mathbb{E}_{x \sim p_g} f(x)$
  - Equivalent to $\min_G W(p_{data}||p_g) = \min_G \max_{||D||_L \le 1} \mathbb{E}_{x \sim p_{data}} D(x) - \mathbb{E}_{x \sim p_g} D(x)$

Wasserstein GAN (WGAN)

- Kantorovich-Rubinstein duality
- Lipschitz Continuity
- Wasserstein GAN

- How to optimize this objective $\min_G W(p_{data}||p_g) = \min_G \max_{||D||_L \leq 1} \mathbb{E}_{x \sim p_{data}} D(x) - \mathbb{E}_{x \sim p_g} D(x)$
  - First step, fix *G* update *D*: $\max_{||D||_L \leq 1} \mathbb{E}_{x \sim p_{data}} D(x) - \mathbb{E}_{x \sim p_g} D(x)$
  - Second step, fix *D* update *G*: $\min_G \mathbb{E}_{x \sim p_{data}} D(x) - \mathbb{E}_{x \sim p_g} D(x)$
  - Obviously, the key is the first step: maximize $\mathbb{E}_{x \sim p_{data}} D(x) - \mathbb{E}_{x \sim p_g} D(x)$, while keeping the $||D||_L \leq 1$ condition by weights clipping

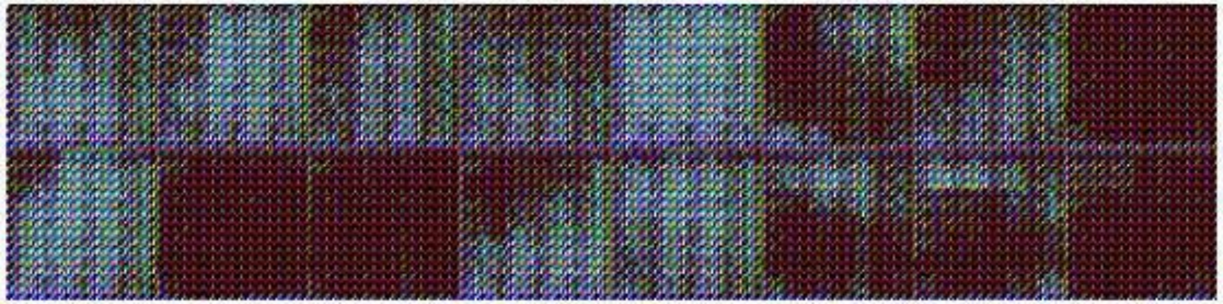
Wasserstein GAN (WGAN)

- Kantorovich-Rubinstein duality
- Lipschitz Continuity
- Wasserstein GAN

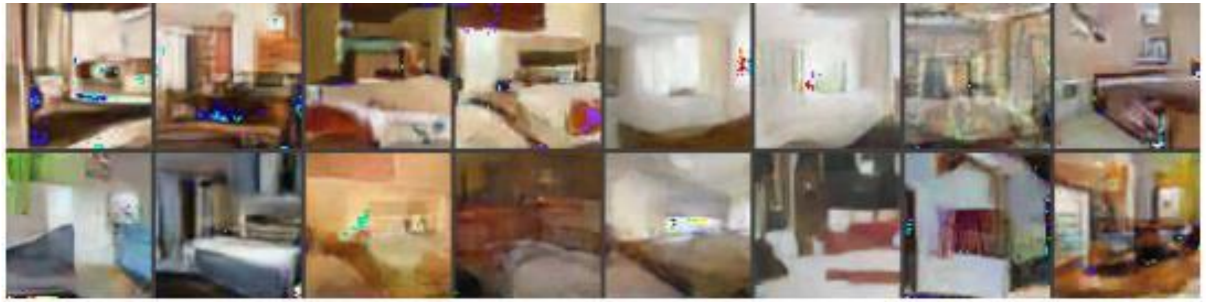
- Algorithm
    1. Sample a batch {z^i, x^i}
    2. Fix *G*, updating *D* with objective: $\max_D \mathbb{E}_{x\sim p_{data}} D(x) - \mathbb{E}_{x\sim p_g} D(x)$
    3. Clip every weight of *D* to [-1, 1]
    4. Fix *D*, updating *G* with objective: $\min_G \mathbb{E}_{x\sim p_{data}} D(x) - \mathbb{E}_{x\sim p_g} D(x)$

WGAN **GAN**

Algorithms trained with a DCGAN generator. Both produce high quality samples



Algorithms trained with smaller generator and without batch normalization. The standard GAN failed to learn while the WGAN still was able to produce samples.



Algorithms trained with MLP generator. Vanilla GAN does mode collapse, while WGAN still produces good samples

Arjovsky, et. al., Wasserstein GAN, ICML 2017

WGAN-GP

- To maintain Lipschitz constraint WGAN uses weight clamping
  - But it is naïve and no guaranteed method
  - Weight clamping leads to optimization difficulties sometimes

- Works are proposed to improve the method for maintaining Lipschitz constraint
  - Improved training of Wasserstein GANs (WGAN-GP) [Gulrajani, et. al., 2017]

    Use gradient penalty to maintain Lipschitz constraint

$$\mathbb{E}_{\hat{x}\sim p_{\hat{x}}}\left[\left(\left\|\nabla_{\hat{x}} D(\hat{x})\right\|_2 - 1\right)^2\right]$$

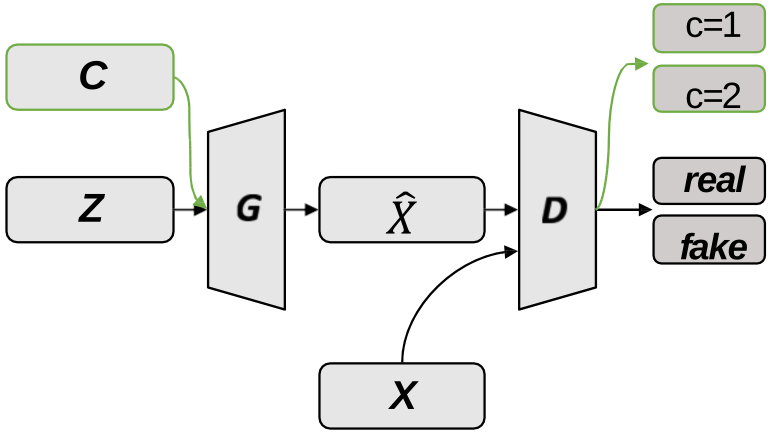
where $\hat{x} = \varepsilon x + (1-\varepsilon)G(z)$

Generative Adversarial Network (GAN)

- Improved GANs
  - Problems in GANs
  - Wasserstein GAN (WGAN)
  - WGAN -GP
- Selected GANs
  - Conditional GAN
  - CycleGAN, DualGAN, DiscoGAN
  - High-Resolution Image Generation: Progressive GAN, StyleGAN

Conditional GAN

- A Simple Example: Auxiliary Classifier GANs



$$\mathcal{L}_D = \mathbb{E}_{x\sim p_{data}}[logD_x(x)] + \mathbb{E}_{z\sim p_z}[log(1 - D_x(G(z,c)))]$$

$$\mathbb{E}_{x\sim p_{data}}[logD_c(x)] + \mathbb{E}_{z\sim p_z}[log(1 - D_c(G(z,c)))]$$

$$\mathcal{L}_G = \mathbb{E}_{x\sim p_{data}}[logD_x(G(z,c))] + \mathbb{E}_{z\sim p_z}[logD_c(G(z,c))]$$

monarch butterfly　　goldfinch　　daisy

Multi-modal problem: one problem has multiple solutions $p(x|c, z)$

Conditional GAN

“Class” conditional generative models

$P(X =$  $|Y = Cat)$

“Text” conditional generative models

$P(X =$  $|Y = \text{“}a\ flower\ with\ white\ petals\ and\ yellow\ stamen\text{”})$

“Text-image” conditional generative models

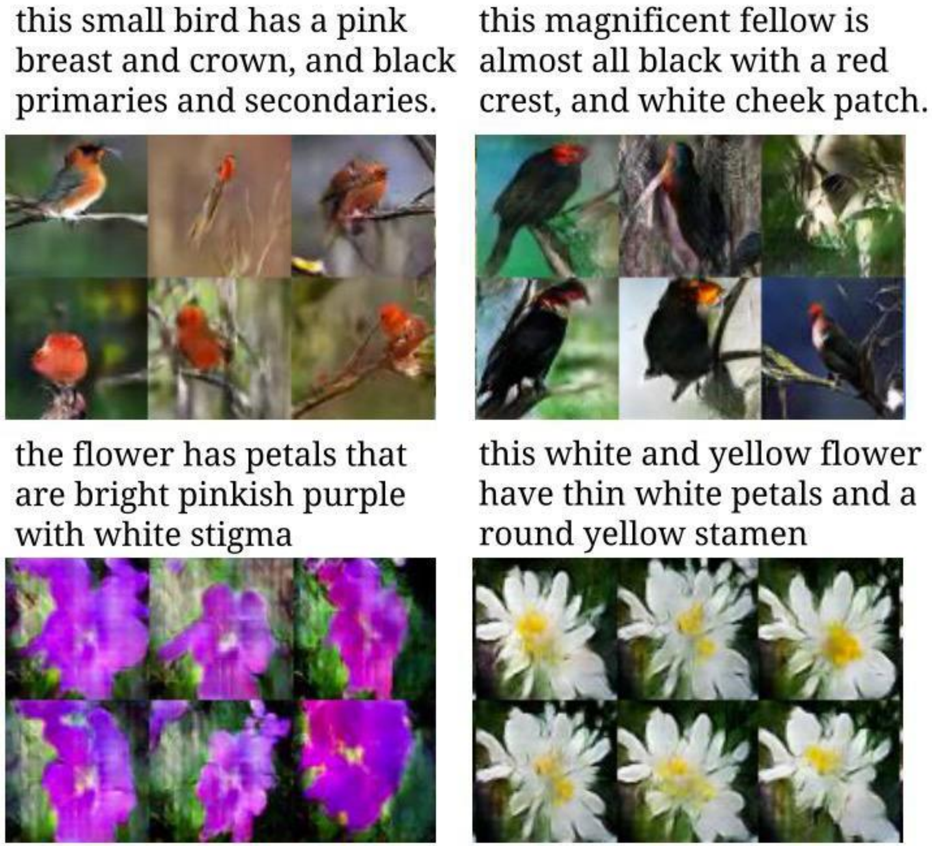
Joint distribution

$P(X =$  $|Y_1 =$  $, Y_2 = \text{“}a\ yellow\ bird\ with\ grey\ wings\text{”})$

- Generative Adversarial Text to Image Synthesis. S. Reed, Z. Akata et al. ICML. 2016.
- Semantic Image Synthesis via Adversarial Learning. H. Dong, S. Yu et al. ICCV 2017.

Conditional GAN

- Text-to-image synthesis: Another Multi-modal generation problem



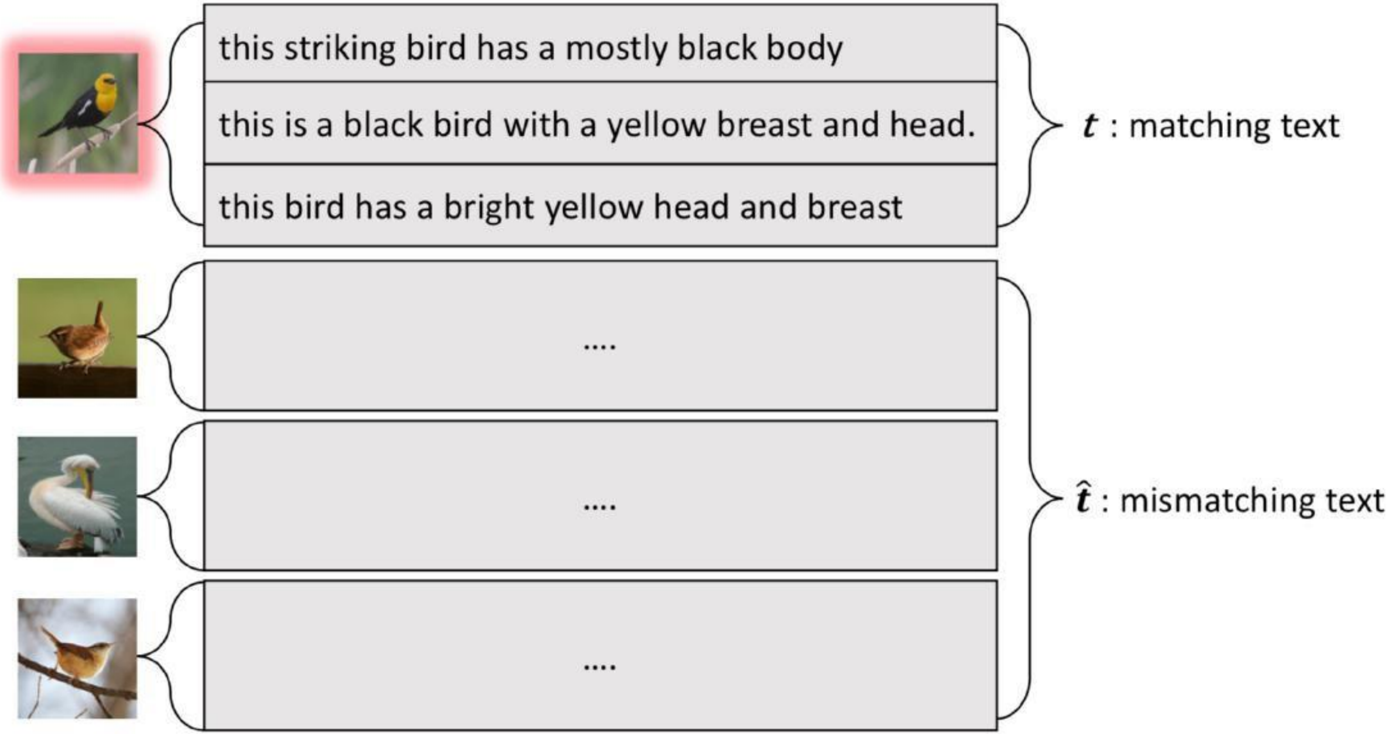
Classic multi-modal problem

P(t, z)

Generative Adversarial Text to Image Synthesis. *S. Reed, Z. Akata et al. ICML. 2016.*

Conditional GAN

- Text-to-image synthesis

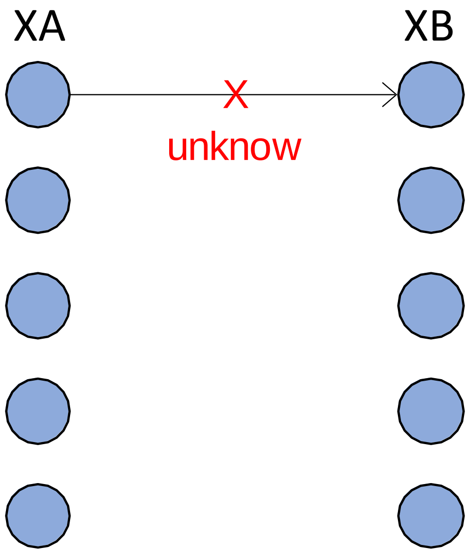


Generative Adversarial Text to Image Synthesis. *S. Reed, Z. Akata et al. ICML. 2016.*

CycleGAN, DualGAN, and DiscoGAN

- **Unpaired** Image-to-Image Translation

Data from two domains without known the mappings
(Learn the unknown mappings)



$\boldsymbol{X_B} = G_{A2B}(\boldsymbol{X_A})$, $\boldsymbol{X_A} = G_{B2A}(\boldsymbol{X_B})$

XA

XB

Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *J. Zhu, T. Park et al. ICCV 2017.*

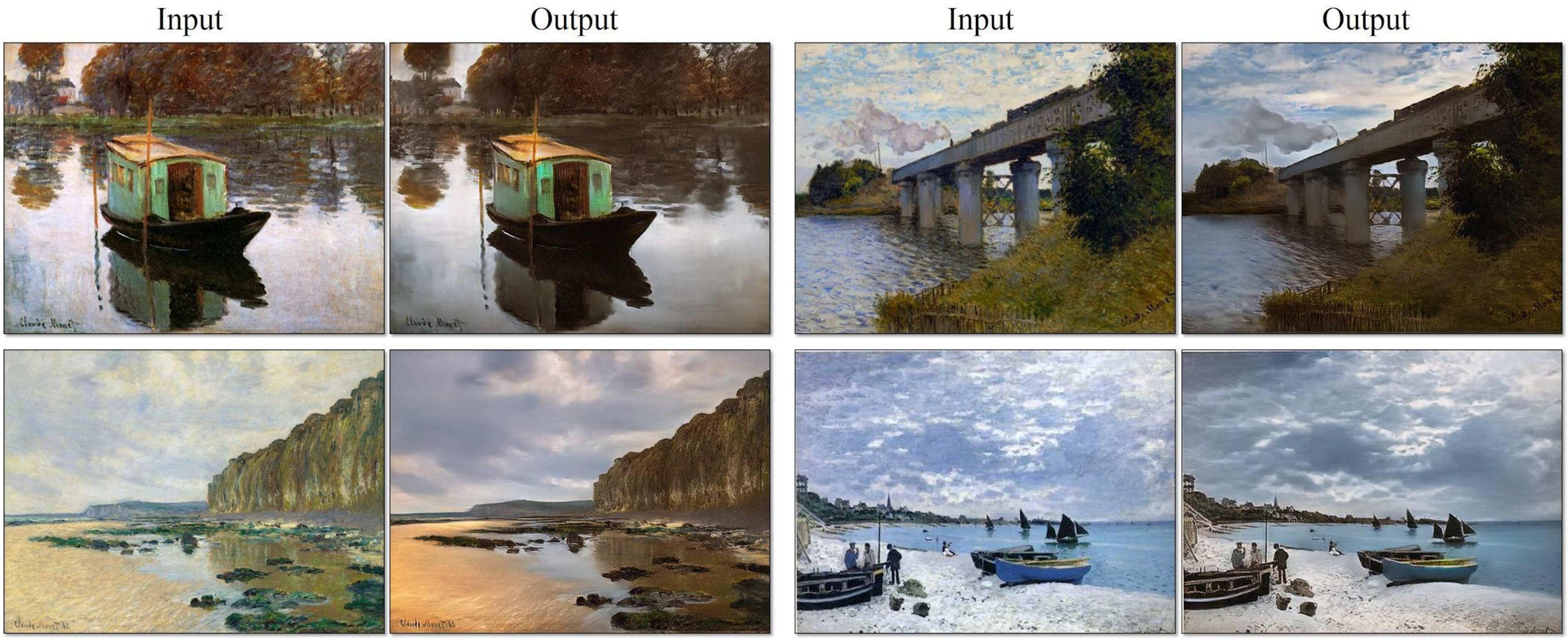
Applications

Style Transfer

Implementing Neural Artistic Style Transfer, Josh Whittle, Apr 2017

Applications

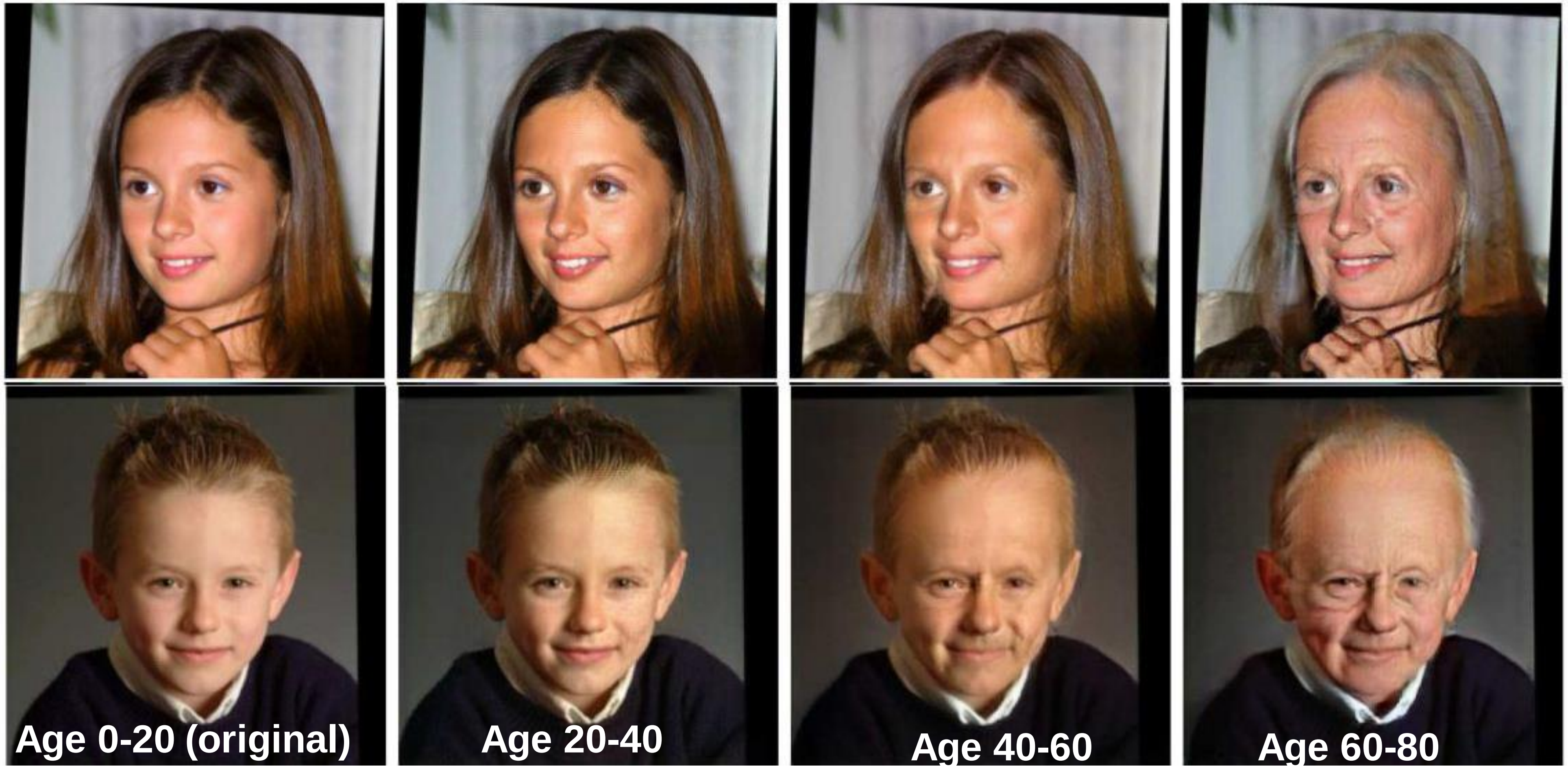
Style Transfer



Images from Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, Jun-Yan Zhu et al. ICCV 2017

Applications

Face Aging

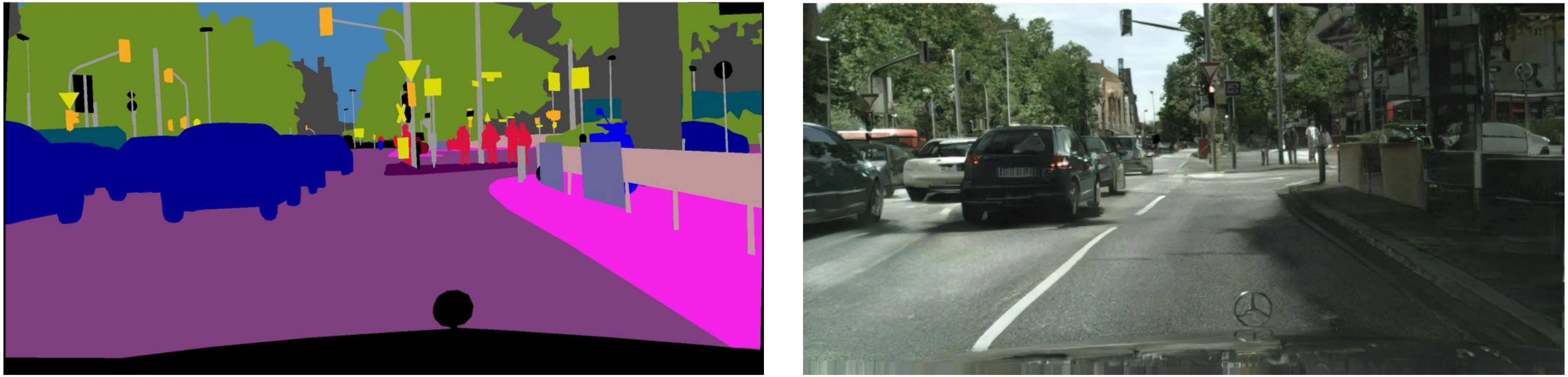


Generative Adversarial Style Transfer Networks for Face Aging, CVPR, June 2018

Applications

Generate Image from Segment Labels

Input labels | Synthesized image

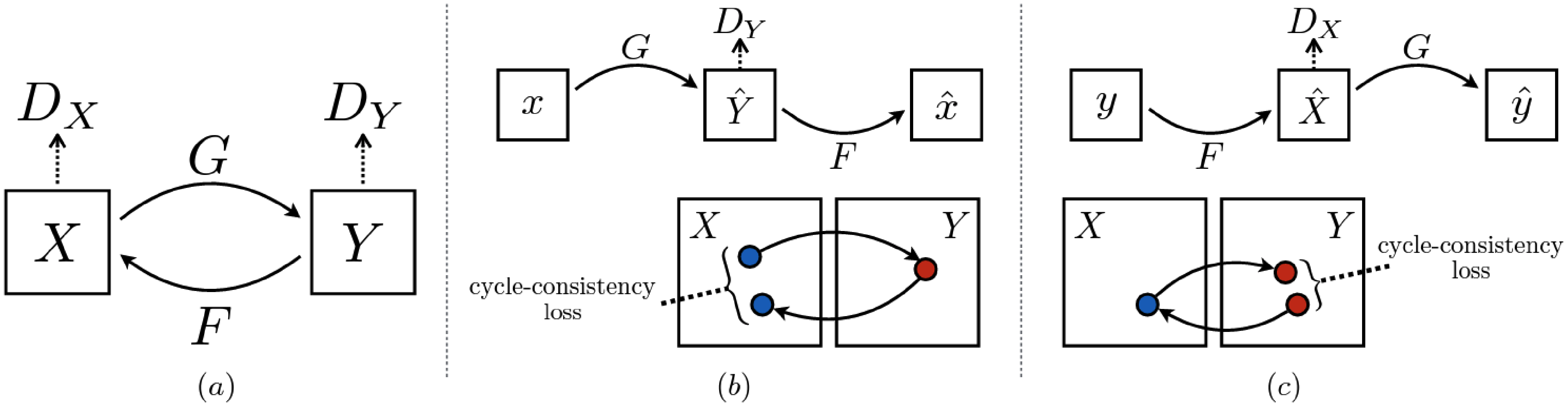


High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs, Wang et al., Nov 2017

CycleGAN, DualGAN, and DiscoGAN

- Motivation: no guarantee that input x (from domain x) & output y (from domain y) will have any meaningful relationship

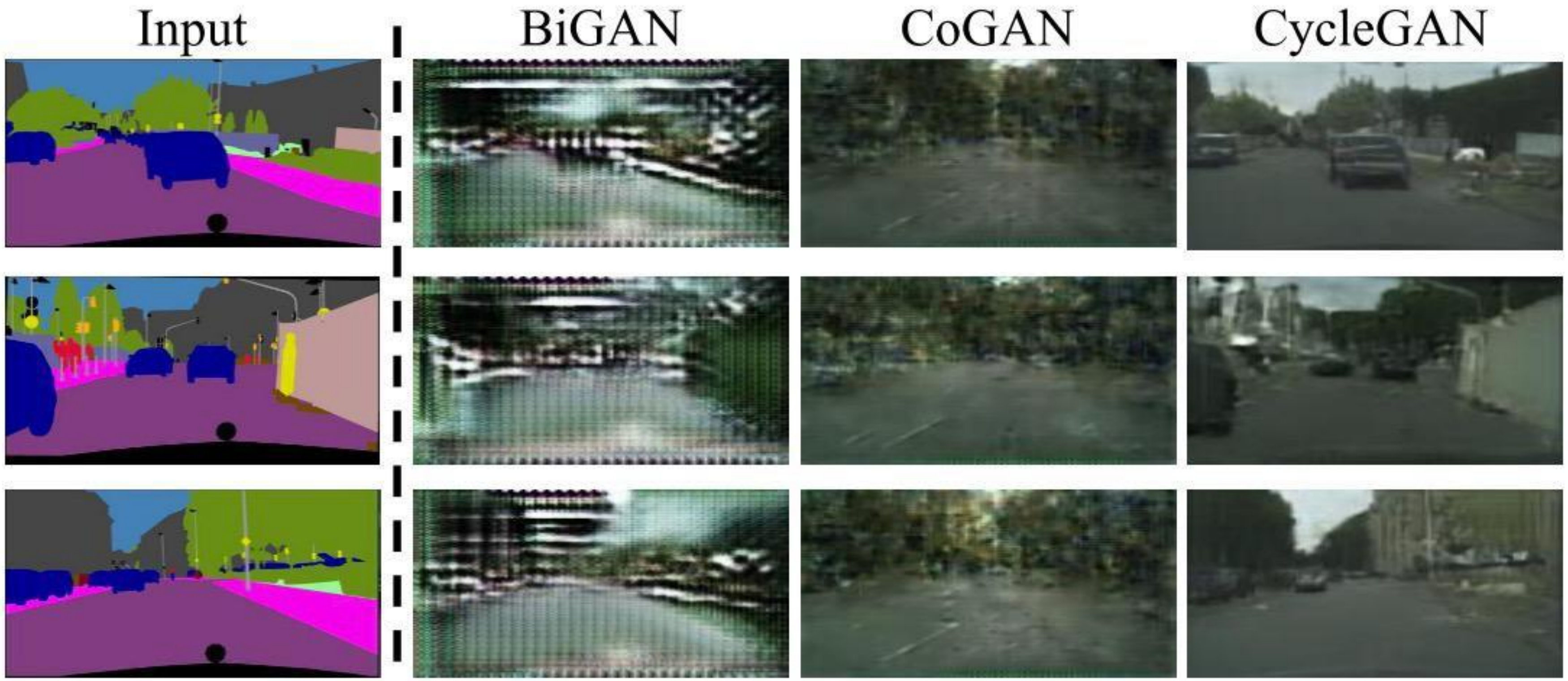
- Cycle-consistency loss + adversarial loss



Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *J. Zhu, T. Park et al. ICCV 2017.*

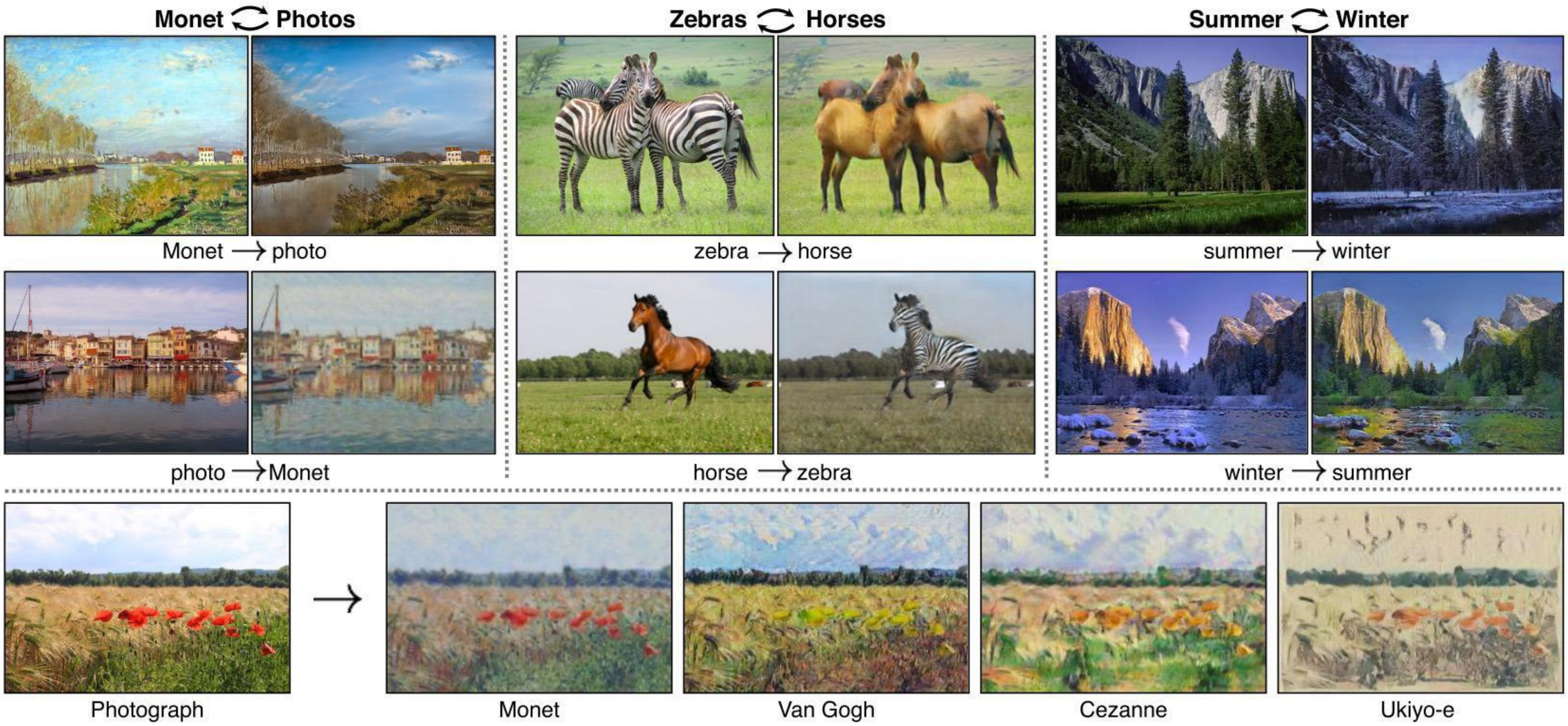
CycleGAN

- Importance of cycle-consistency loss



Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *J. Zhu, T. Park et al. ICCV 2017.*

CycleGAN



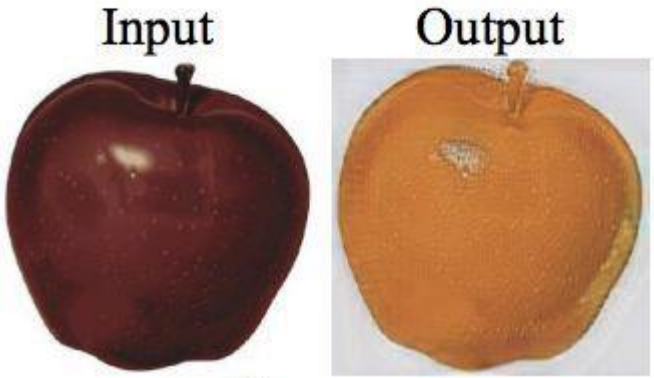
Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *J. Zhu, T. Park et al. ICCV 2017.*

CycleGAN

Limitations

- **The good**: CycleGAN works well for texture or color changes
- **The bad**: Geometric changes (e.g apple <-> orange) have little success
- Dataset characteristic can also cause confusion, e.g ImageNet does not contain images of a man riding a horse/zebra

Still a gap between results from paired training data and those achieved by the unpaired method

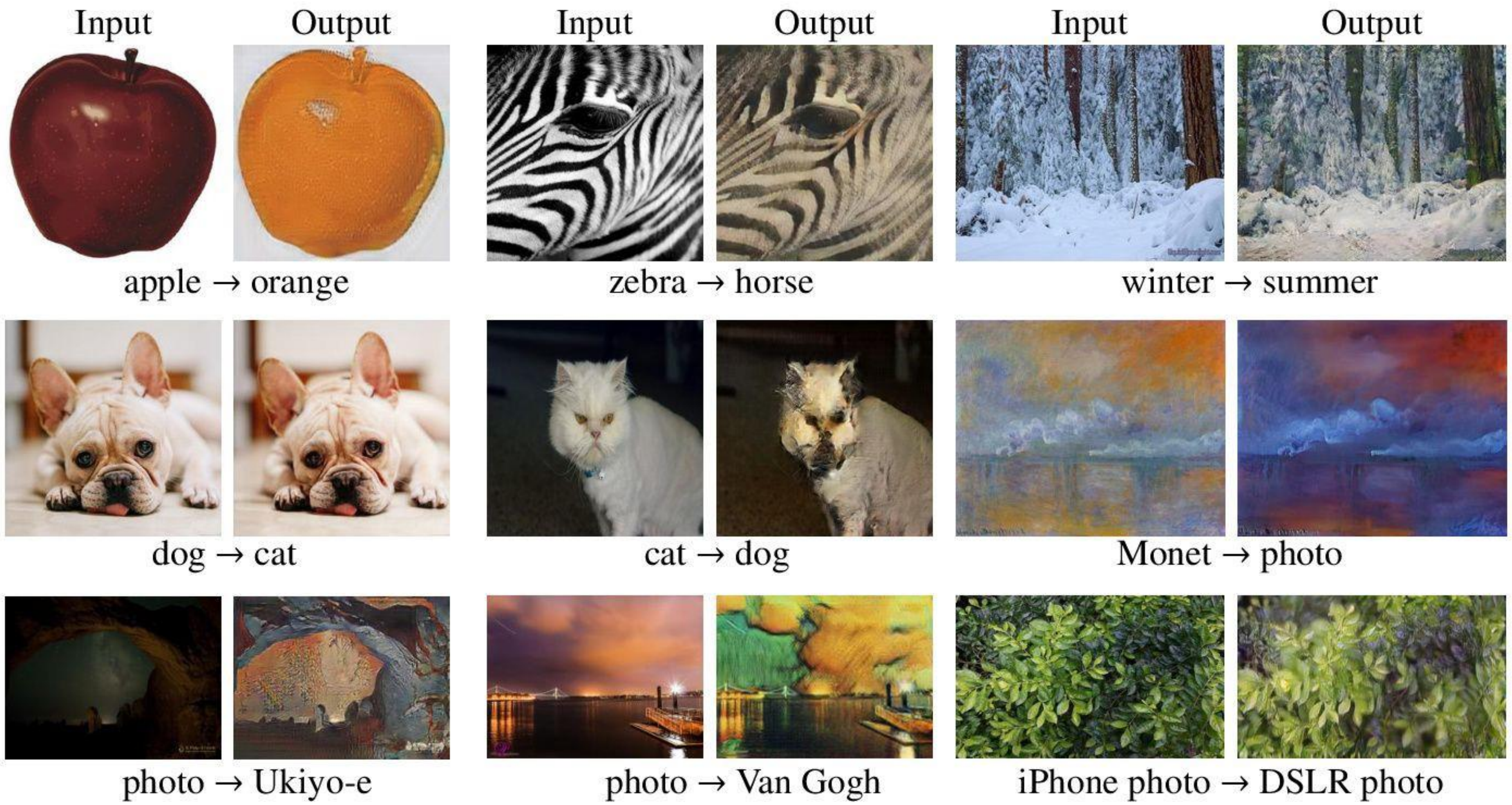


apple → orange

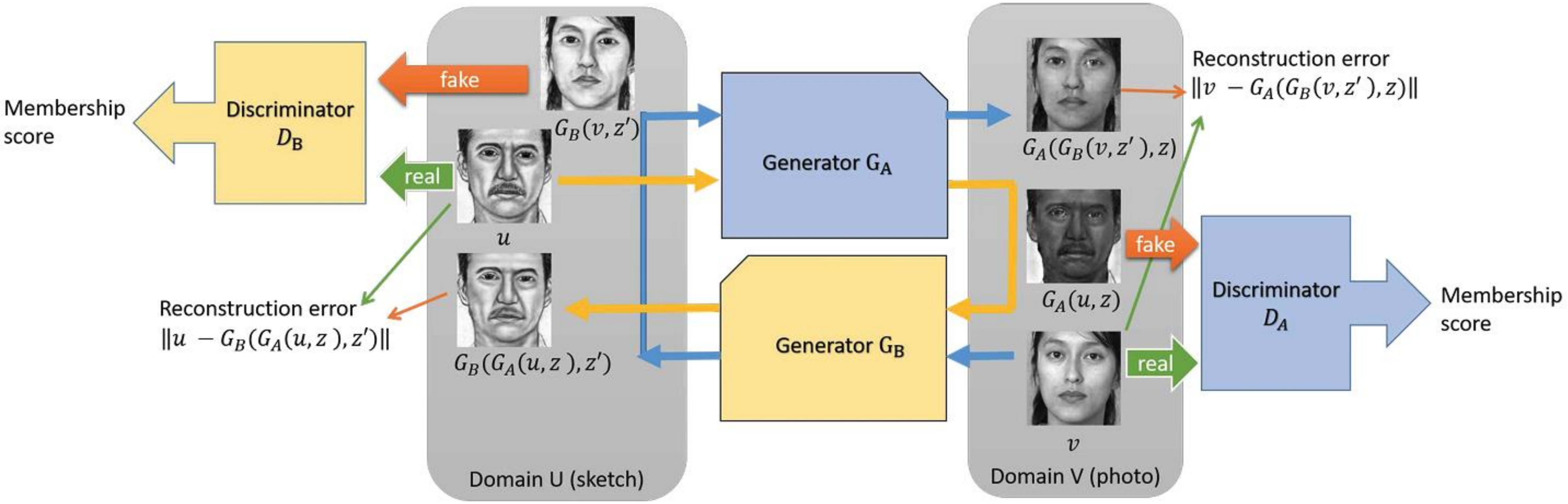
Horse → Zebra

CycleGAN

Failures results:



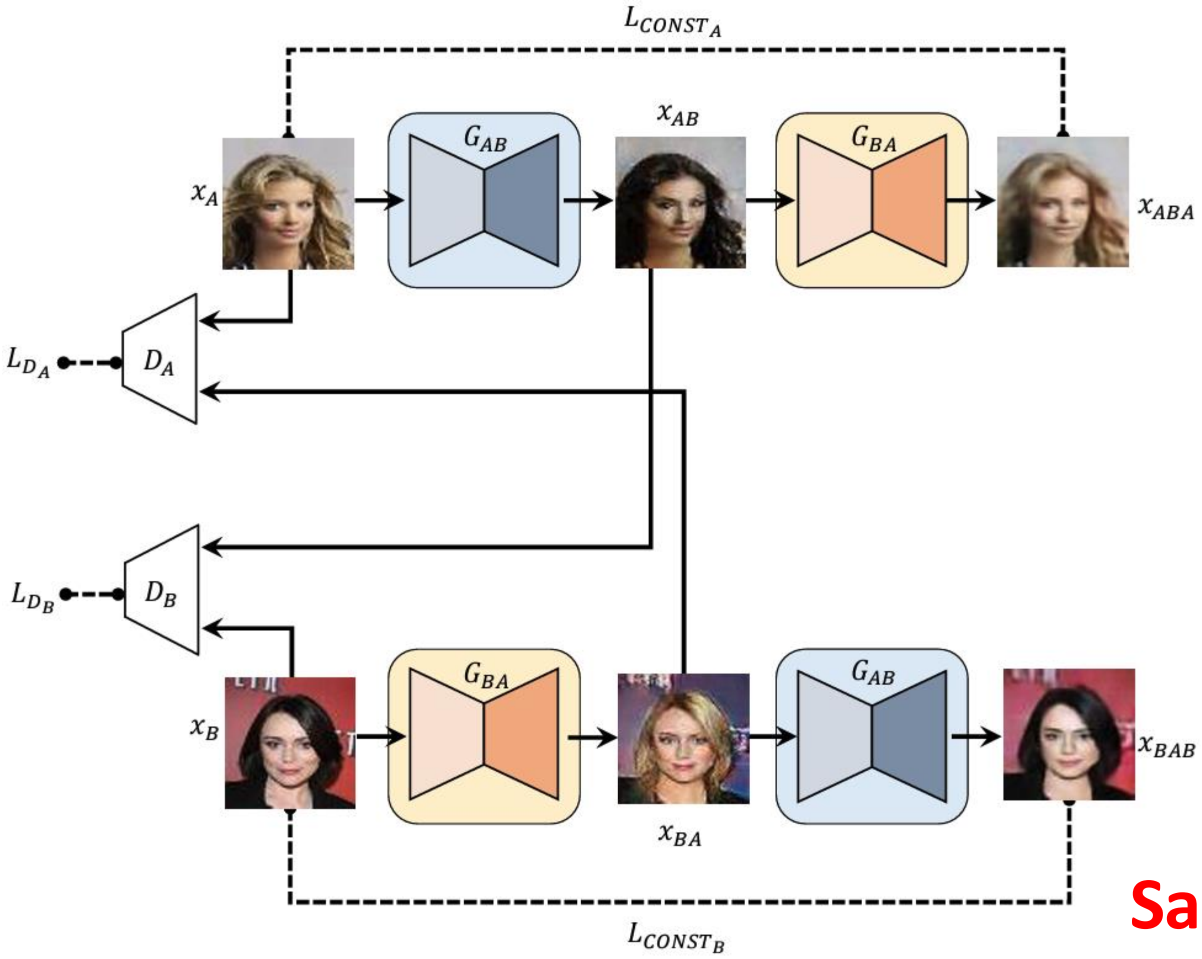
CycleGAN, **DualGAN**, and DiscoGAN



Same Core Idea Underlied

DualGAN: Unsupervised Dual Learning for Image-to-Image Translation, Zili Yi et al. ICCV 2017
DiscoGAN: Learning to Discover Cross-Domain Relations with Generative Adversarial Networks, Taeksoo Kim et al. ICML 2017

CycleGAN, DualGAN, and **DiscoGAN**



Same Core Idea Underlied

DualGAN: Unsupervised Dual Learning for Image-to-Image Translation, Zili Yi et al. ICCV 2017
DiscoGAN: Learning to Discover Cross-Domain Relations with Generative Adversarial Networks, Taeksoo Kim et al. ICML 2017

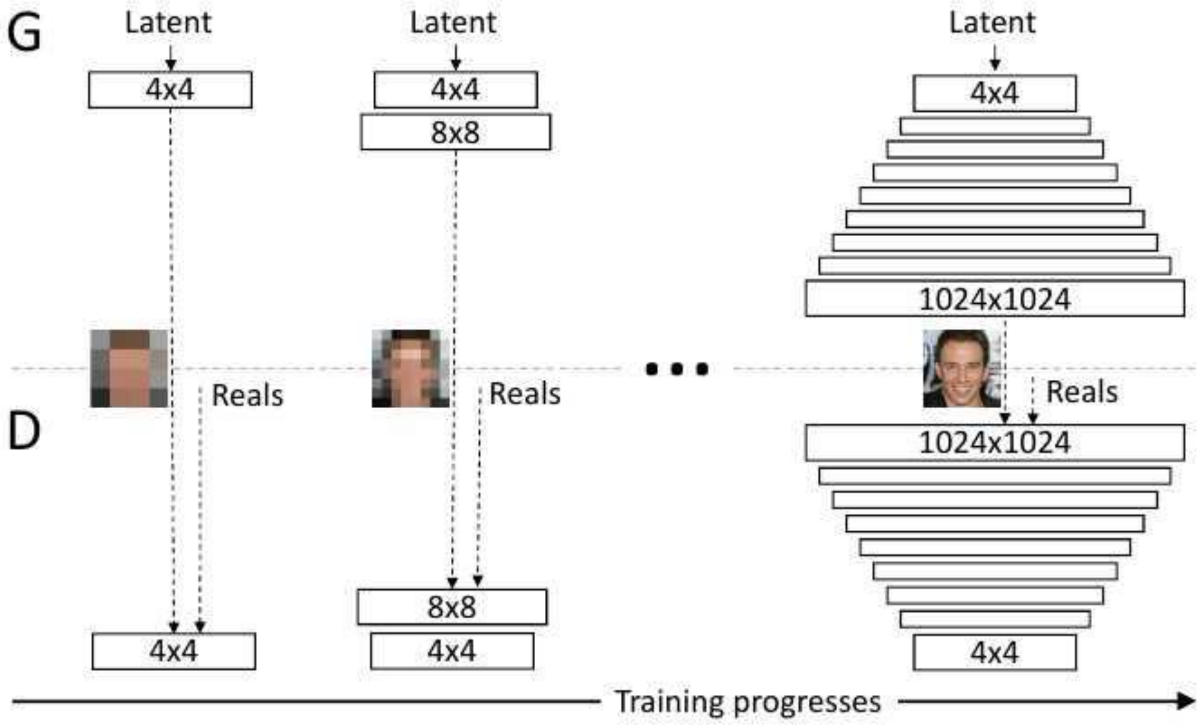
Progressive GAN

- GANs produce sharp images
  - But only in fairly small resolutions and with somewhat limited variation
- Training continues to be unstable despite recent progress
- Generating high resolution image is difficult
  - It is easier to tell the generated images from training images in high-res images

Karras, et. al., Progressive growing of GANs for improved quality, stability, and variation, ICLR 2018

Progressive GAN

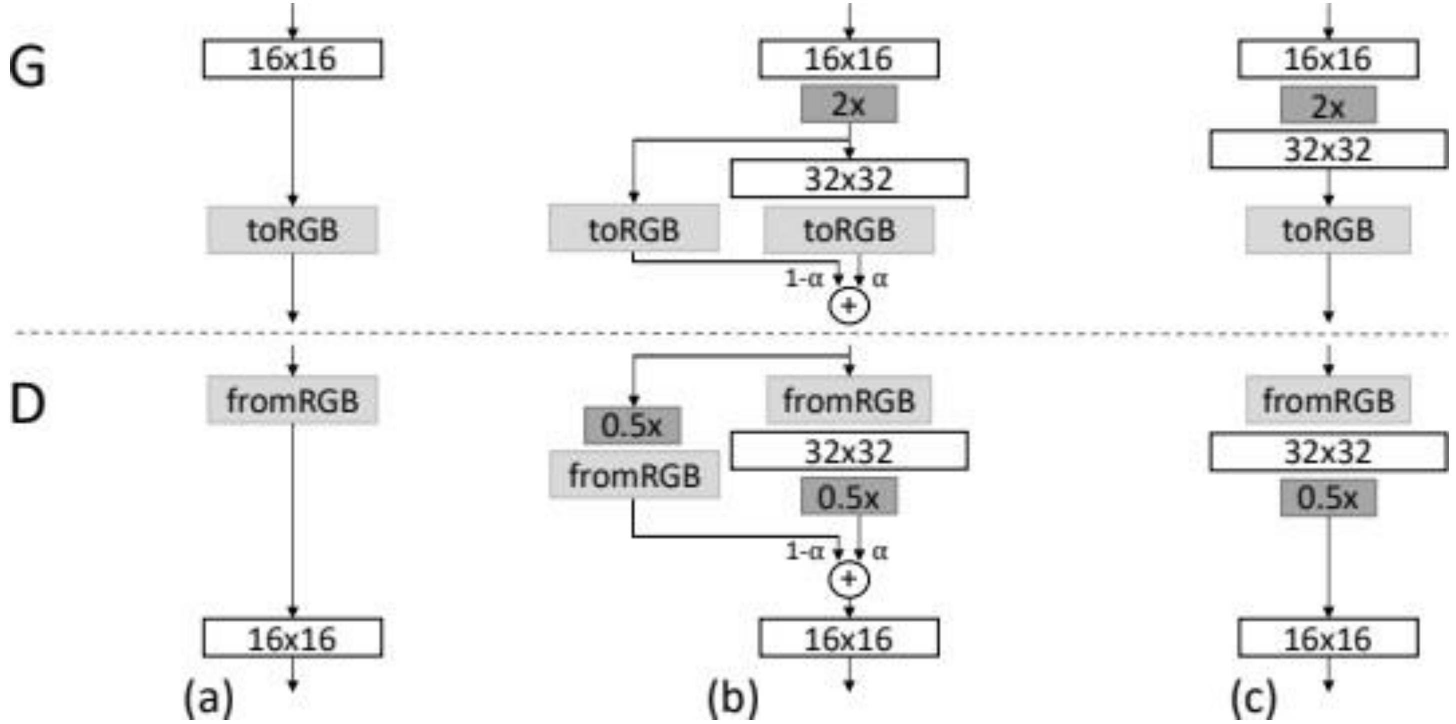
- Grow both generator and discriminator progressively
- Start learning from easier low-resolution images
- Add new layers that introduce higher-resolution details as the training progress



Karras, et. al., Progressive growing of GANs for improved quality, stability, and variation, ICLR 2018

Progressive GAN

- Fade in the new layers smoothly



Transition from 16 × 16 images (a) to 32 × 32 images (c). During the transition (b) we treat the layers that operate on the higher resolution like a residual block, whose weight α increases linearly from 0 to 1

Progressive GAN results

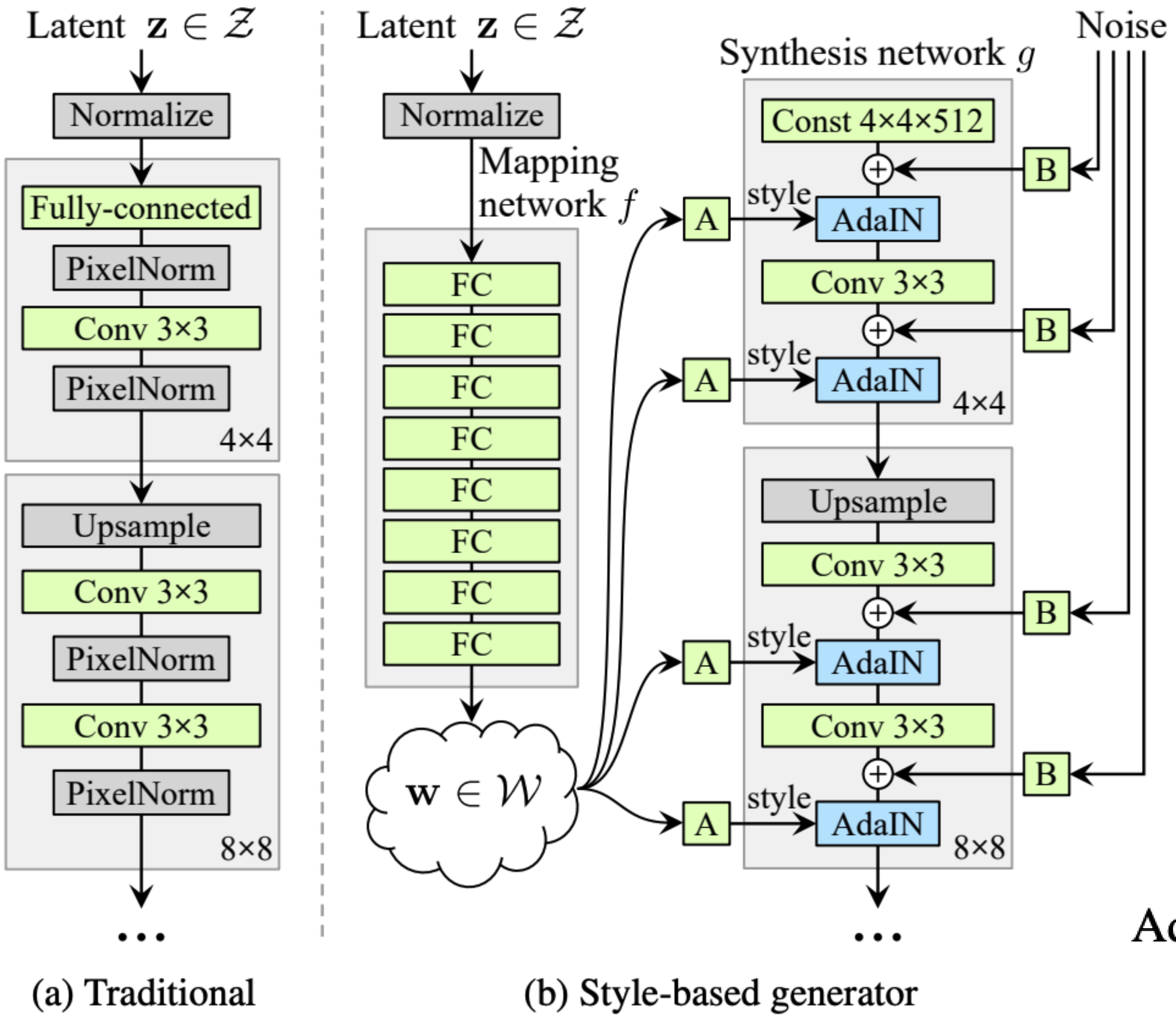
1024x1024 images generated using the Celeba-HQ

LSUN other categories generated image (256x256)

Mao et al. (2016b) (128 × 128) Gulrajani et al. (2017) (128 × 128) Our (256 × 256)

LSUN bedroom

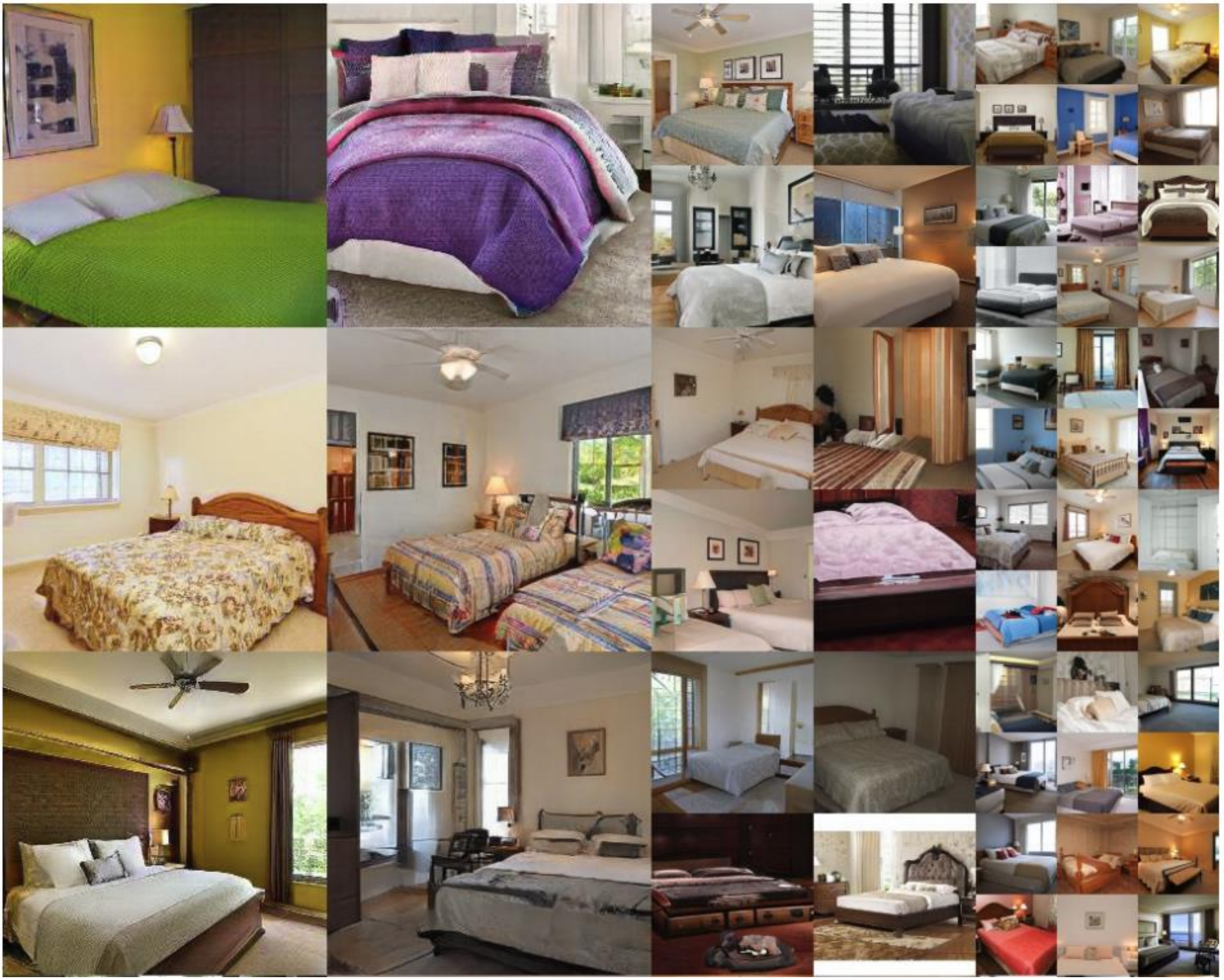
StyleGAN

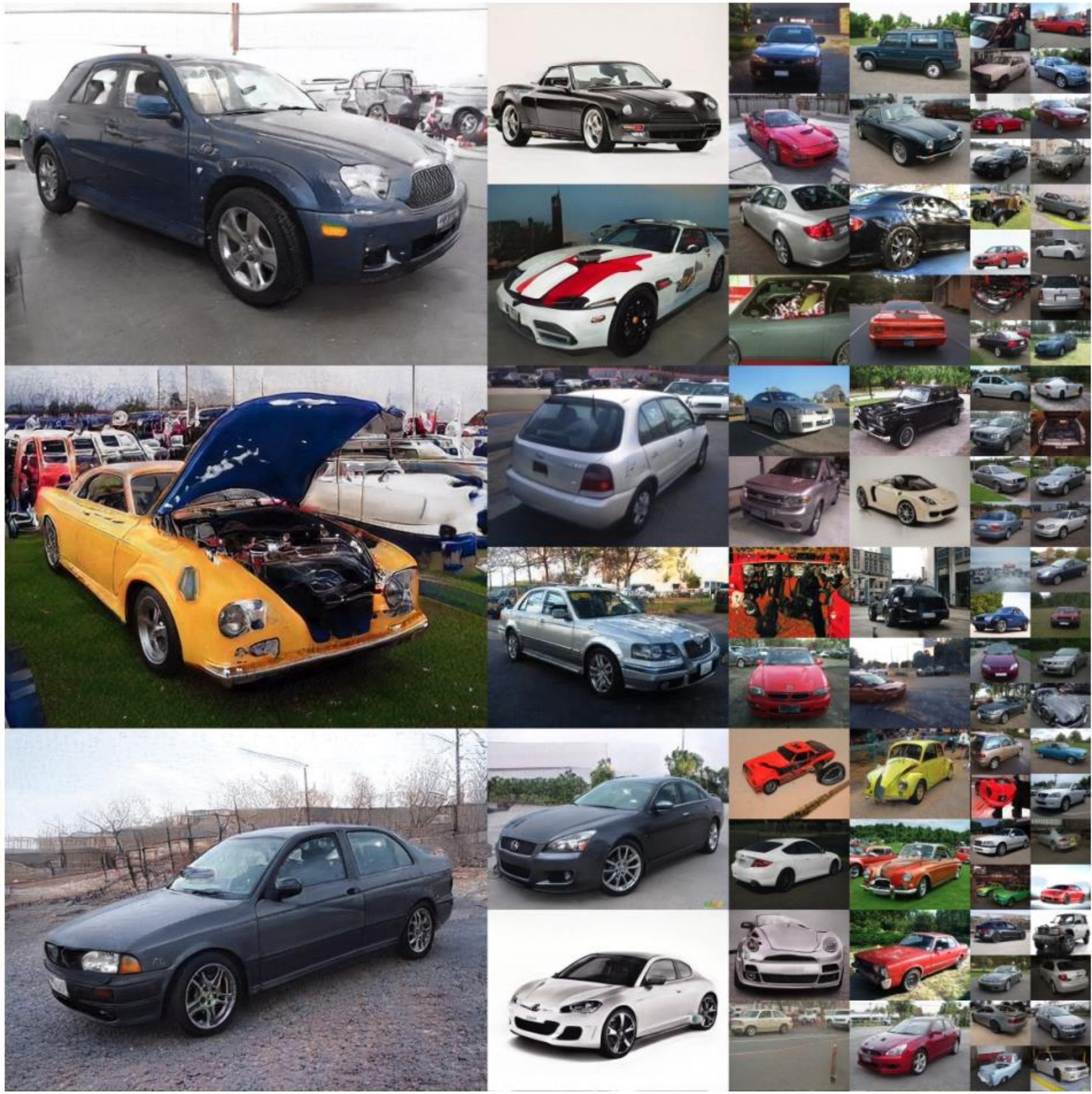


$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i}$$

Karras, T. "A style-based generator architecture for generative adversarial networks." *CVPR*. 2019.

Karras, T. "A style-based generator architecture for generative adversarial networks." *CVPR*. 2019.





Karras, T. "A style-based generator architecture for generative adversarial networks." *CVPR*. 2019.

Next

- GAN Inversion
  - Challenge
  - Optimization-based method
  - Encoder-based method
  - Walking on the Latent Space
  - …

Thank You

- Questions?

- Email: yu.yin@case.edu

CASE SCHOOL OF ENGINEERING
CASE WESTERN RESERVE UNIVERSITY

Reference slides

- Hao Dong. Deep Generative Models
- https://lilianweng.github.io/posts/2018-10-13-flow-models/
- Yunhun Jang, Hyungwon Choi, Sangwoo Mo and Sungsoo Ahn, EE807: Recent Advances in Deep Learning.
- Kenny Green, Rotem Shaul, Chang Liu, Domain Transfer, 2018