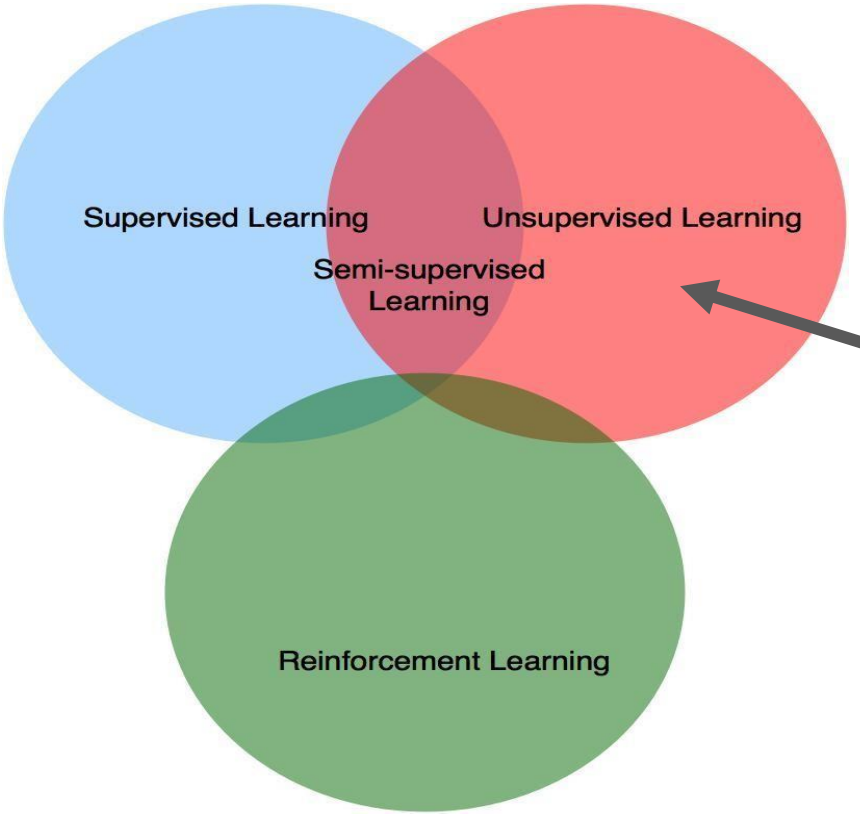
Some presentations about GANs

Contents

1. Introduction

Taxonomy of ML

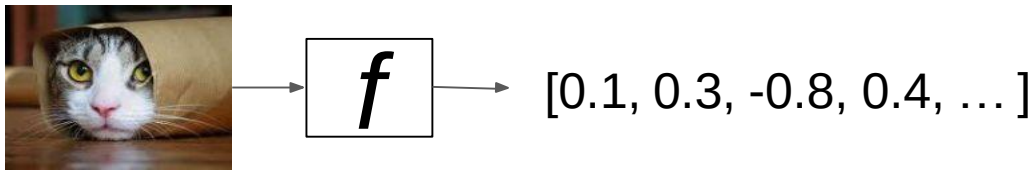


Generative model

From David silver, Reinforcement learning (UCL course on RL, 2015).
2017.03 Namju Kim, https://www.slideshare.net/ssuser77ee21/generative-adversarial-networks-70896091?fbclid=IwAR0_i8VbVH--XWZjh-fD0haqcMTwPLTHvKZ2zGvB8xFXY_RxRHz2lVzFuv8

Unsupervised Learning

- Find deterministic function ***f*** : ***z = f(x)***, x : data, z : **latent**



Generative Model

- Find generation function ***g*** : $x = g(z)$, x : data, z : **latent**

https://www.slideshare.net/ssuser77ee21/generative-adversarial-networks-70896091?fbclid=IwAR0_yJDNB3KAsrRnsszGVHSgQ9rikWsY-QTwu-xst_1bap57T7-Bd79GiLc

Unsupervised learning vs. Generative model

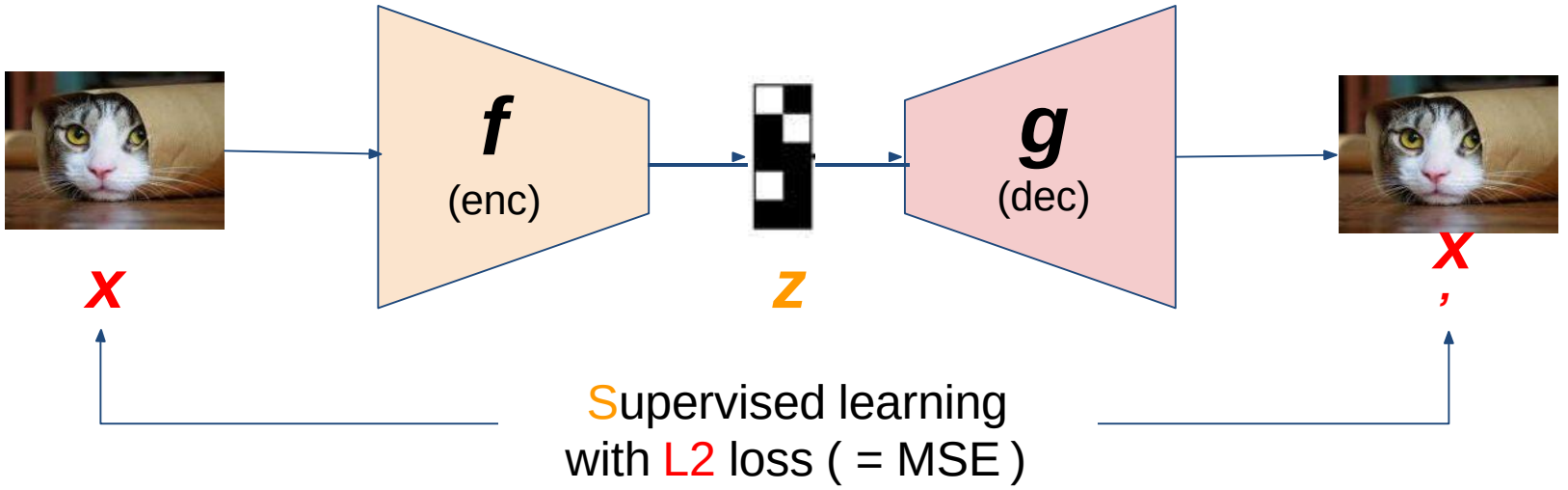
- ***z = f(x)*** *vs.* ***x = g(z)***
- ***P(z\|x)*** *vs.* ***P(x\|z)***
- ***Encoder*** *vs.* ***Decoder (Generator)***
  - P(x, z) needed. (cf : P(y|x) in supervised learning)
    - P(z|x) = P(x, z) / P(x) → P(x) is intractable (ELBO)
    - P(x|z) = P(x, z) / P(z) → P(z) is given. (prior)

$$\theta^* = \arg\max_{\theta} P(X, \boldsymbol{Z}; \theta)$$

https://www.slideshare.net/ssuser77ee21/generative-adversarial-networks-70896091?fbclid=IwAR0_yJDNB3KAsrRnsszGVHSgQ9rikWsY-QTwu-xst_1bap57T7-Bd79GiLc

Stacked autoencoder - SAE

- Use data itself as label → Convert UL into reconstruction SL
- ***z = f(x), x=g(z) → x = g(f(x))***
- https://github.com/buriburisuri/sugartensor/blob/master/sugartensor/example/mnist_sae.py

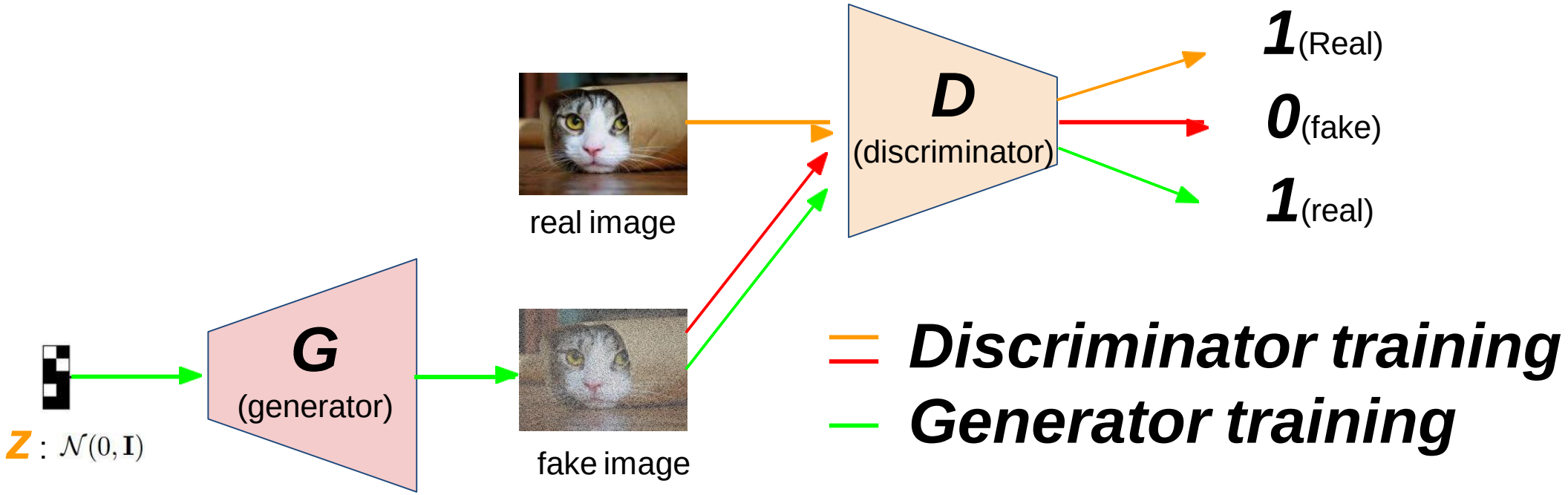


Generative Adversarial Networks - GAN

- Ian Goodfellow et al, “Generative Adversarial Networks”, 2014.
  - **Learnable** cost function
  - Mini-Max game based on Nash Equilibrium
    - Little assumption
    - High fidelity
  - Hard to training - no guarantee to equilibrium.

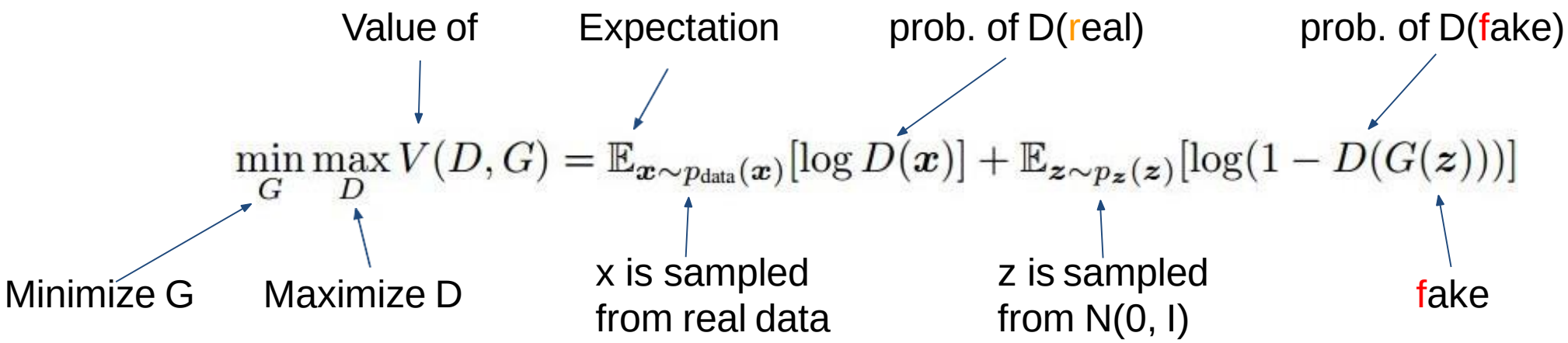
Generative Adversarial Networks - GAN

- Alternate training
- https://github.com/buriburisuri/sugartensor/blob/master/sugartensor/example/mnist_gan.py



Generative Adversarial Networks - GAN

- Mathematical notation



Generative Adversarial Networks - GAN

- Mathematical notation - discriminator

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

Maximize prob. of D(real)

Minimize prob. of D(fake)

BCE(binary cross entropy) with label 1 for real, 0 for fake.

(Practically, **CE** will be OK. or more plausible.)

Generative Adversarial Networks - GAN

- Mathematical notation - generator

$$\min_G V(D,G) = \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \simeq \max_G \mathbb{E}_{z\sim p_z(z)}[\log(D(G(z)))]$$

Maximize prob. of D(fake)

BCE(binary cross entropy) with label 1 for fake.

(Practically, **CE** will be OK. or more plausible.)

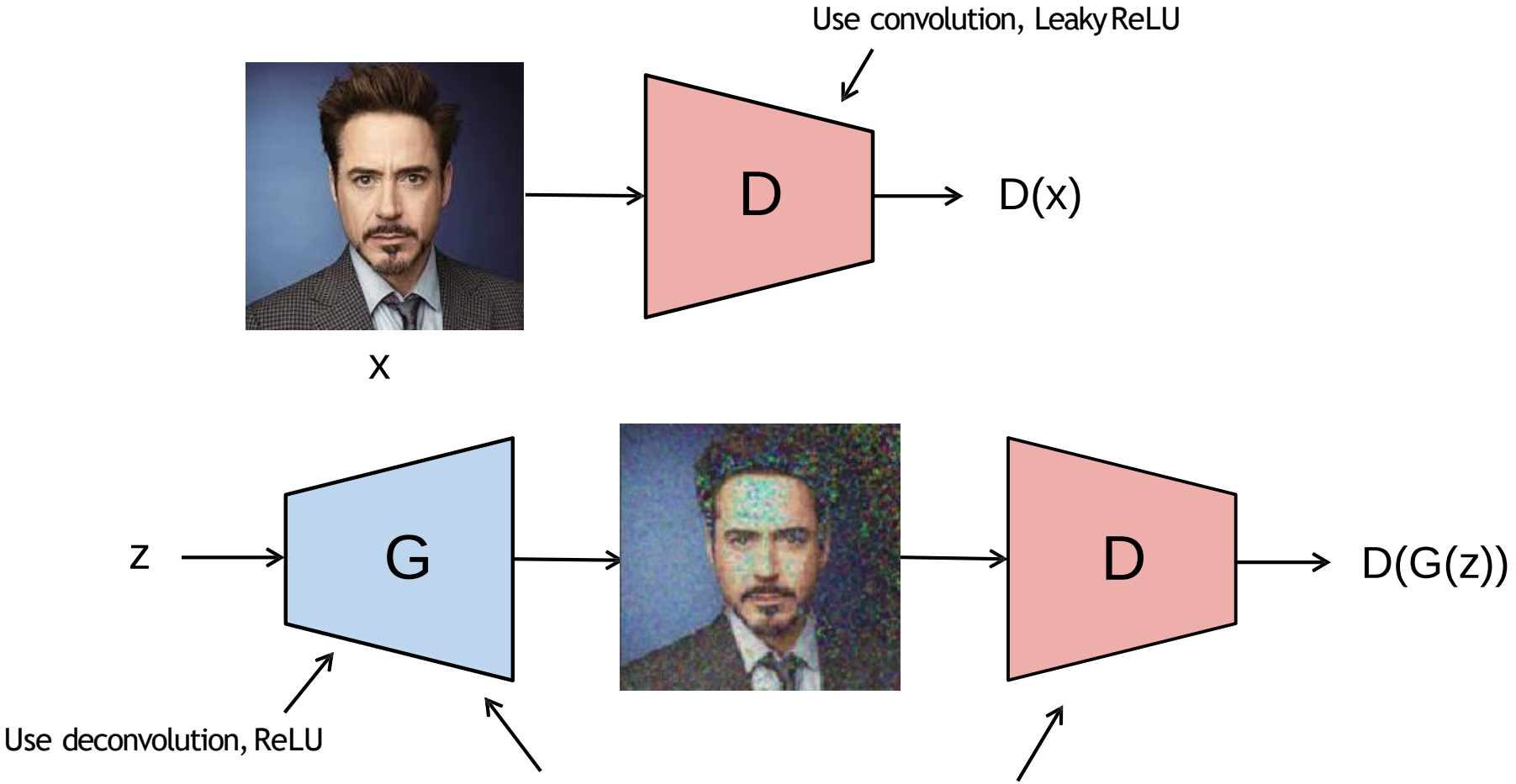
2. Development of GANs

https://www.slideshare.net/NaverEngineering/1-gangenerative-adversarial-network

Pitfall of GAN

- No guarantee to equilibrium
  - Mode collapsing
  - Oscillation
  - No indicator when to finish
- All generative model
  - Evaluation metrics (Human turing test)
  - Overfitting test (Nearest Neighbor) → GAN is robust !!!
  - Diversity test
  - Wu et al. On the quantitative analysis of decoder-based generative model, 2016

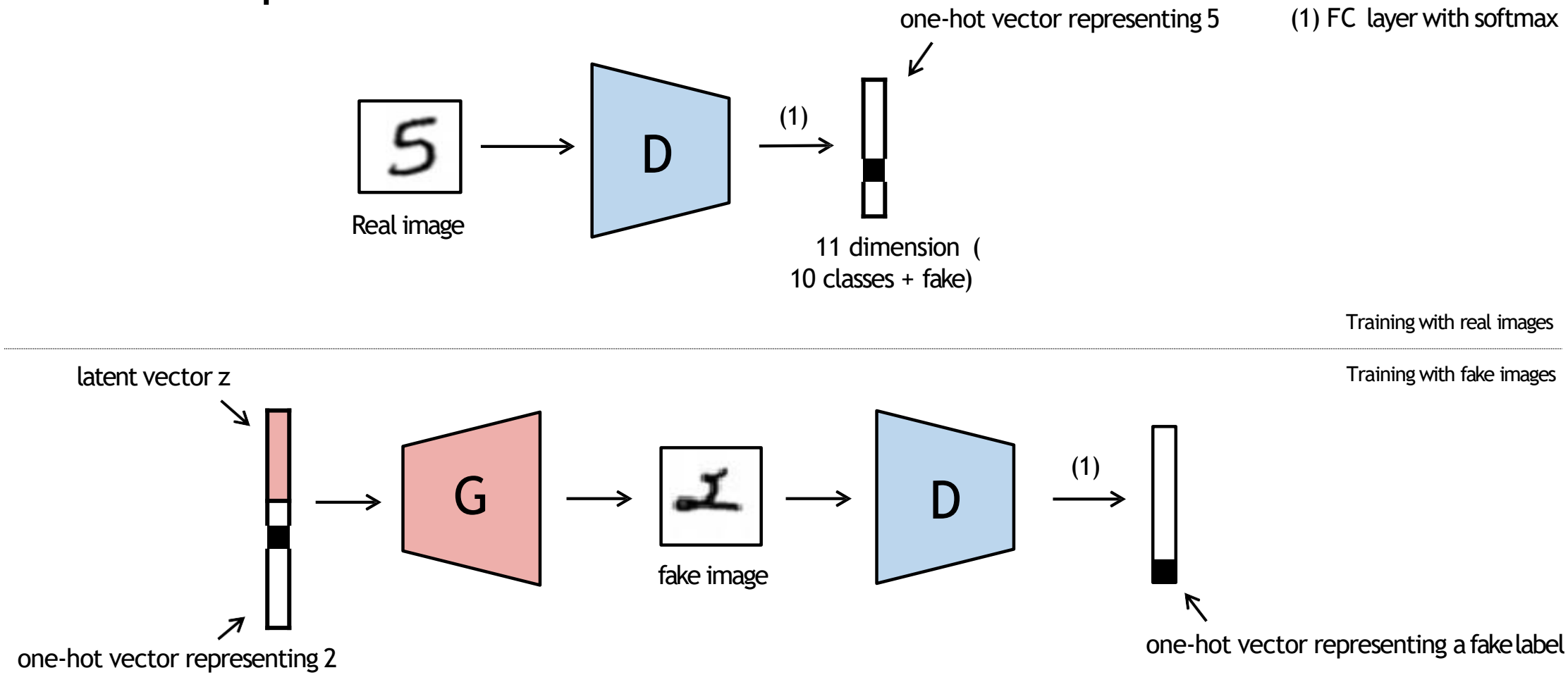
DCGAN



- No pooling layer (Instead strided convolution)
- Use batch normalization
- Adam optimizer(lr=0.0002, beta1=0.5, beta2=0.999)

SGAN

- Semi-Supervised GAN

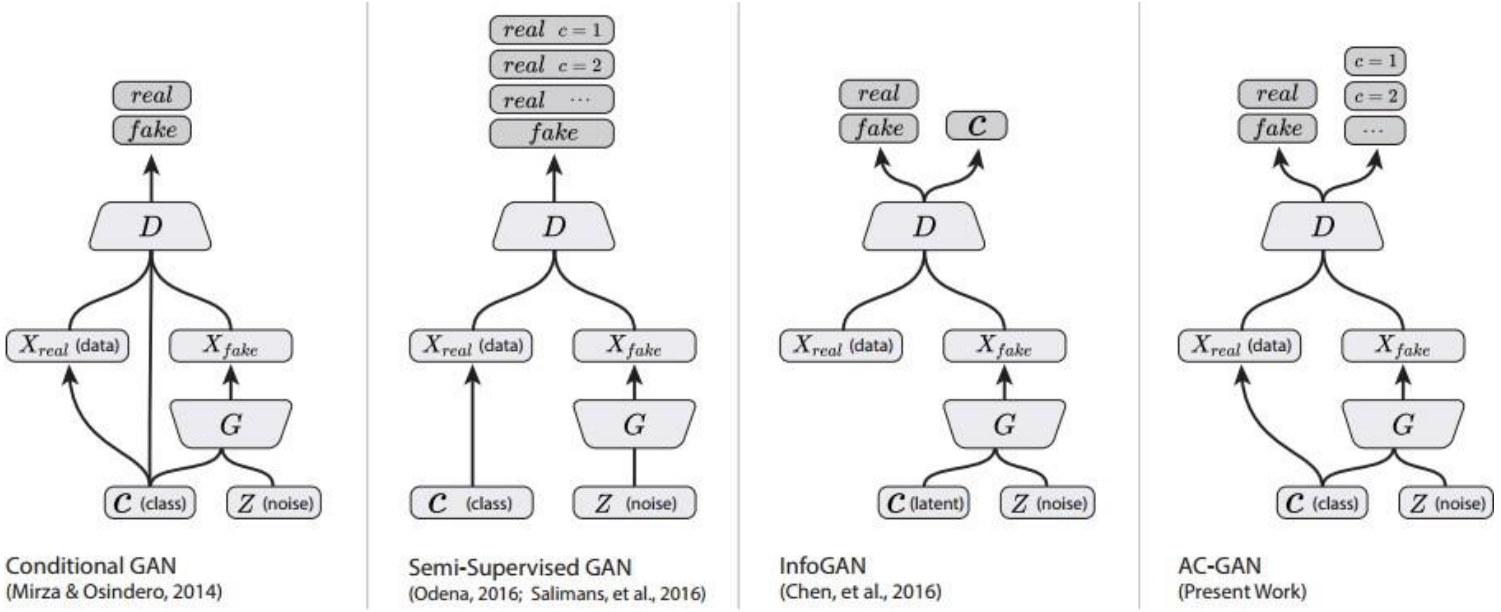


Augustus Odena et al. Semi-Supervised Learning with Generative Adversarial Netwoks, 2016

ACGAN

- Auxiliary Classifier GAN(ACGAN), 2016

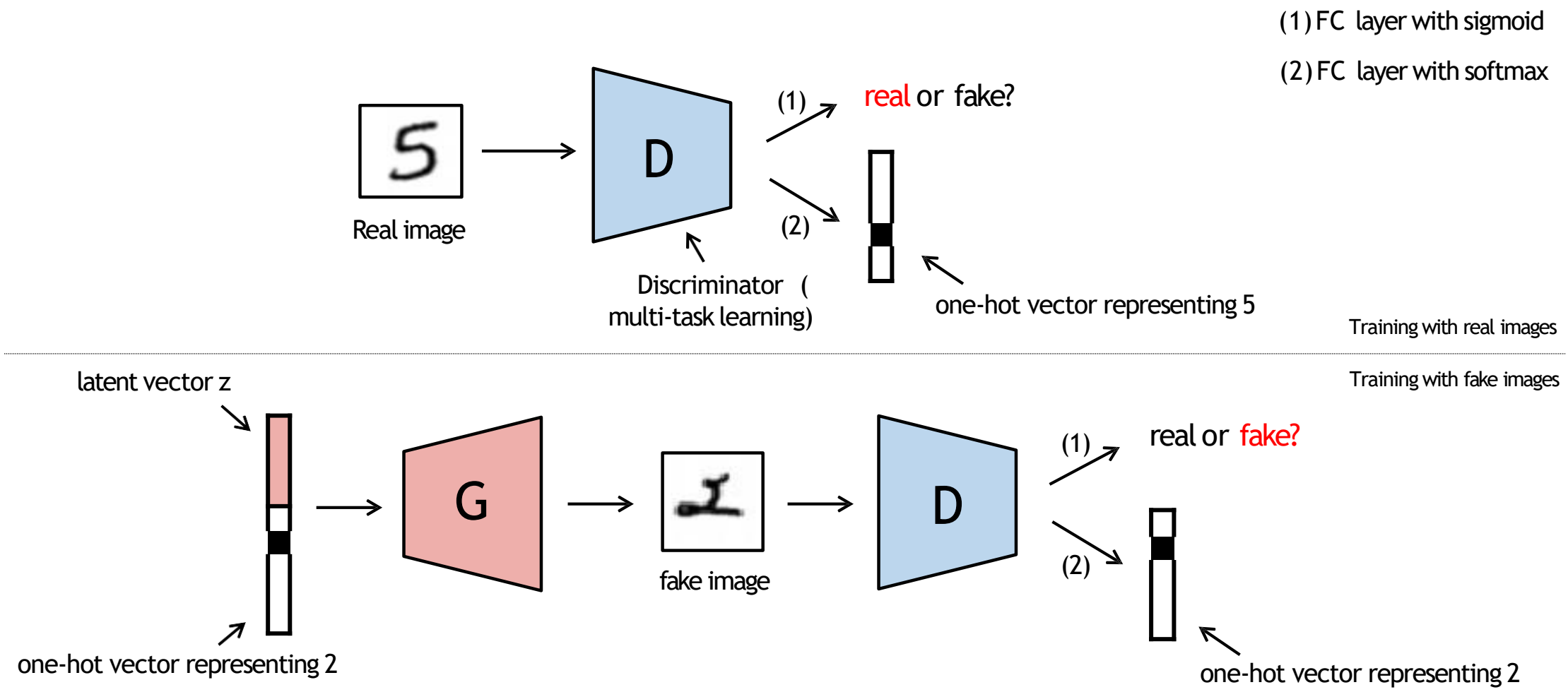
  Proposed a new method for improved training of GANs using class labels.



Augustus Odena et al. Conditional Image Synthesis with Auxilary Classifier, 2016

ACGAN

- How does it work?



3. Image translation with GANs

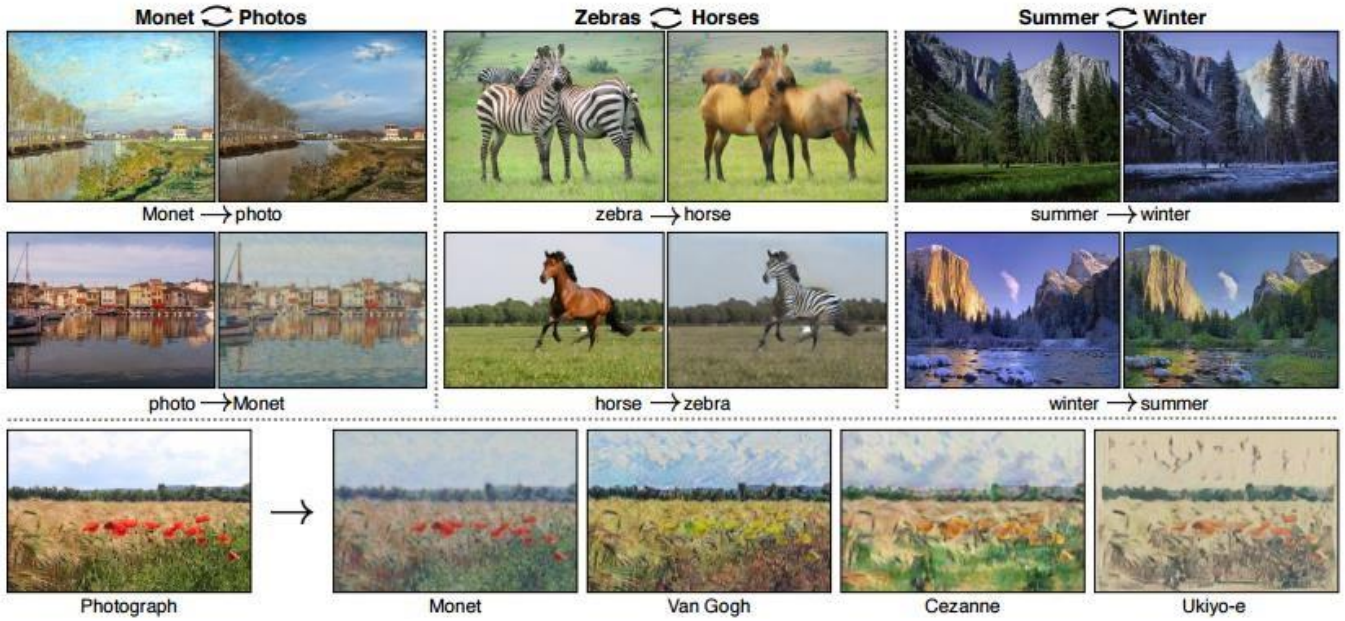
1.

2. 박태성, Finding connections among images using CycleGAN, https://www.slideshare.net/NaverEngineering/finding-connections-among-images-using-cyclegan?from_action=save

CycleGAN

- CycleGAN: Unpaired Image-to-ImageTranslation

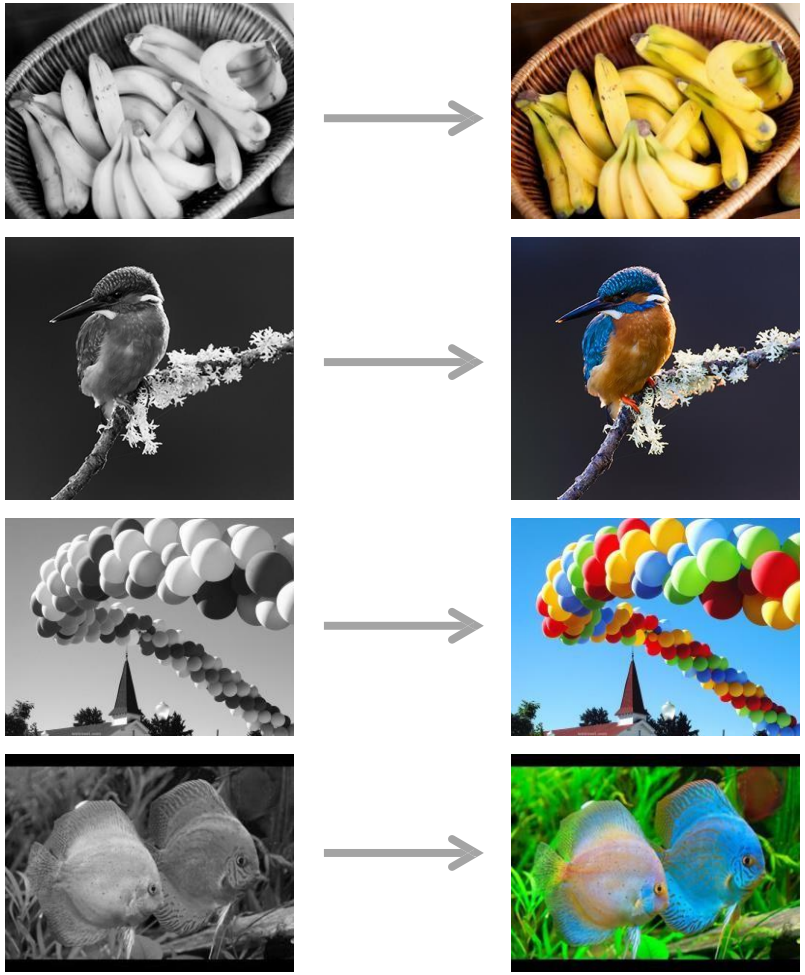
presents a GAN model that transfer an image from a source domain A to a target domain B in the absence of paired examples.



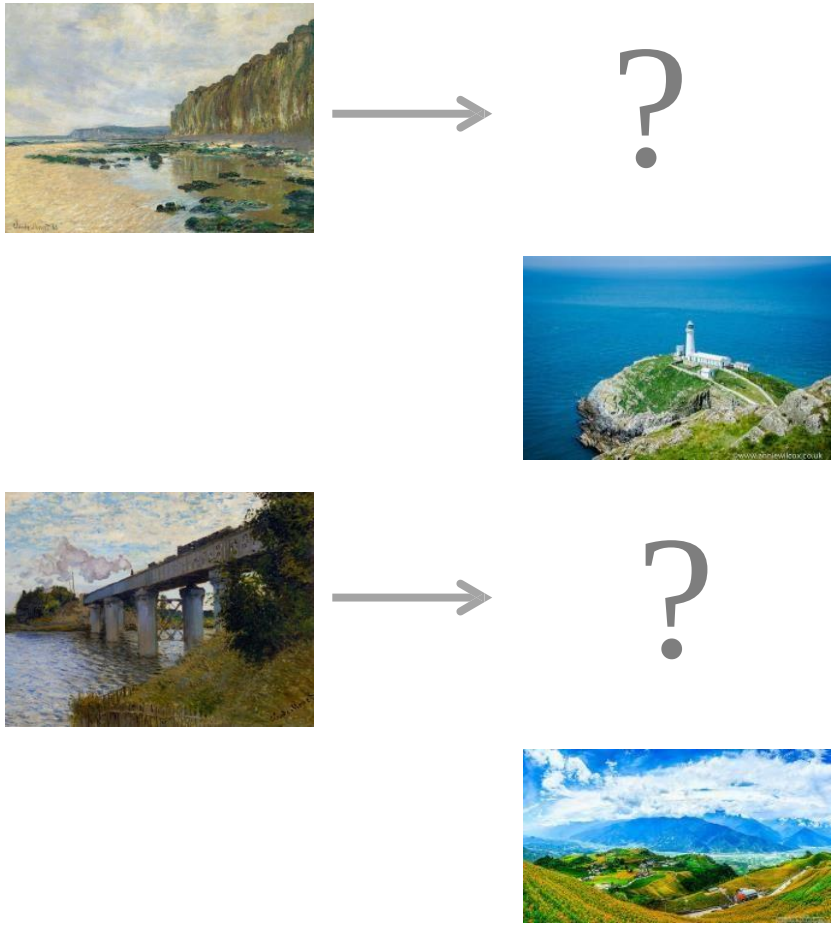
Jun-Yan Zhu et al. Unpaired Image-to-Image Translation using Cycle Consistent Adversarial Networks, 2017

CycleGAN (Zhu et al. 2017)

pix2pix



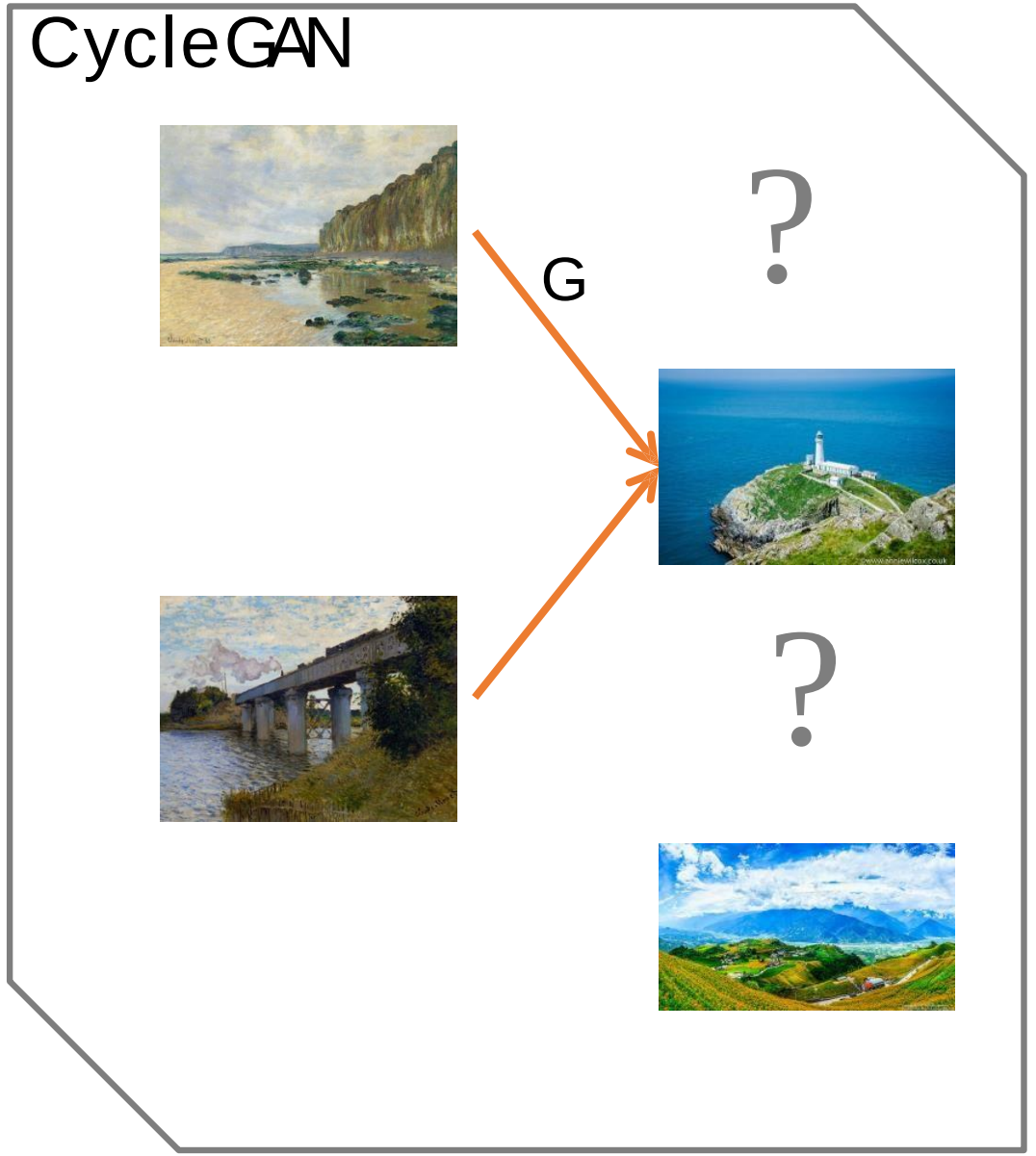
CycleGAN



CycleGAN (Zhu et al. 2017)

Loss: $L_{GAN}\ (G(x), y)$

G(x) should just look photorealistic

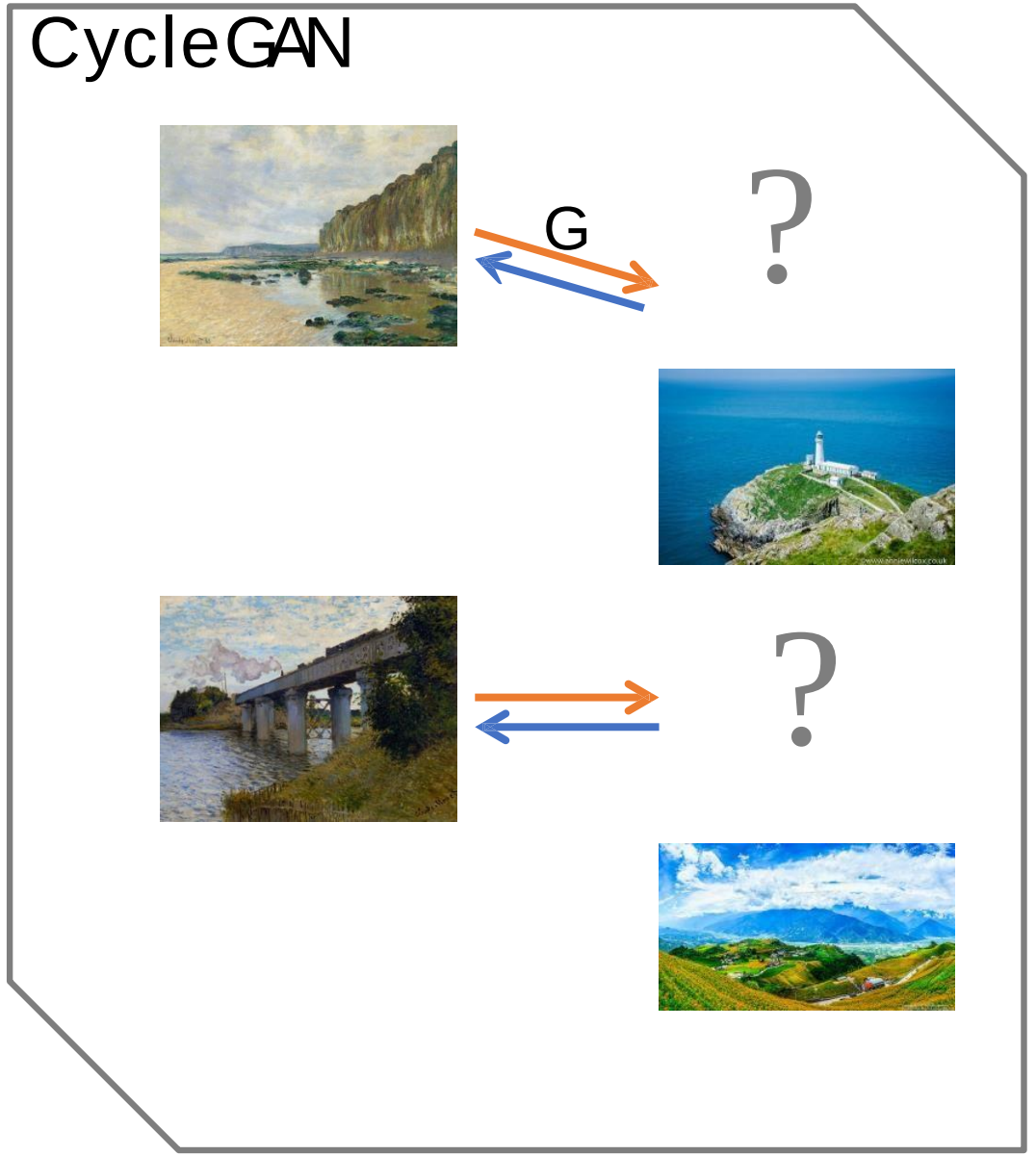


CycleGAN (Zhu et al. 2017)

Loss: $L_{GAN}\ (G(x), y)$

G(x) should just look photorealistic

and be able to reconstruct *x*



CycleGAN (Zhu et al. 2017)

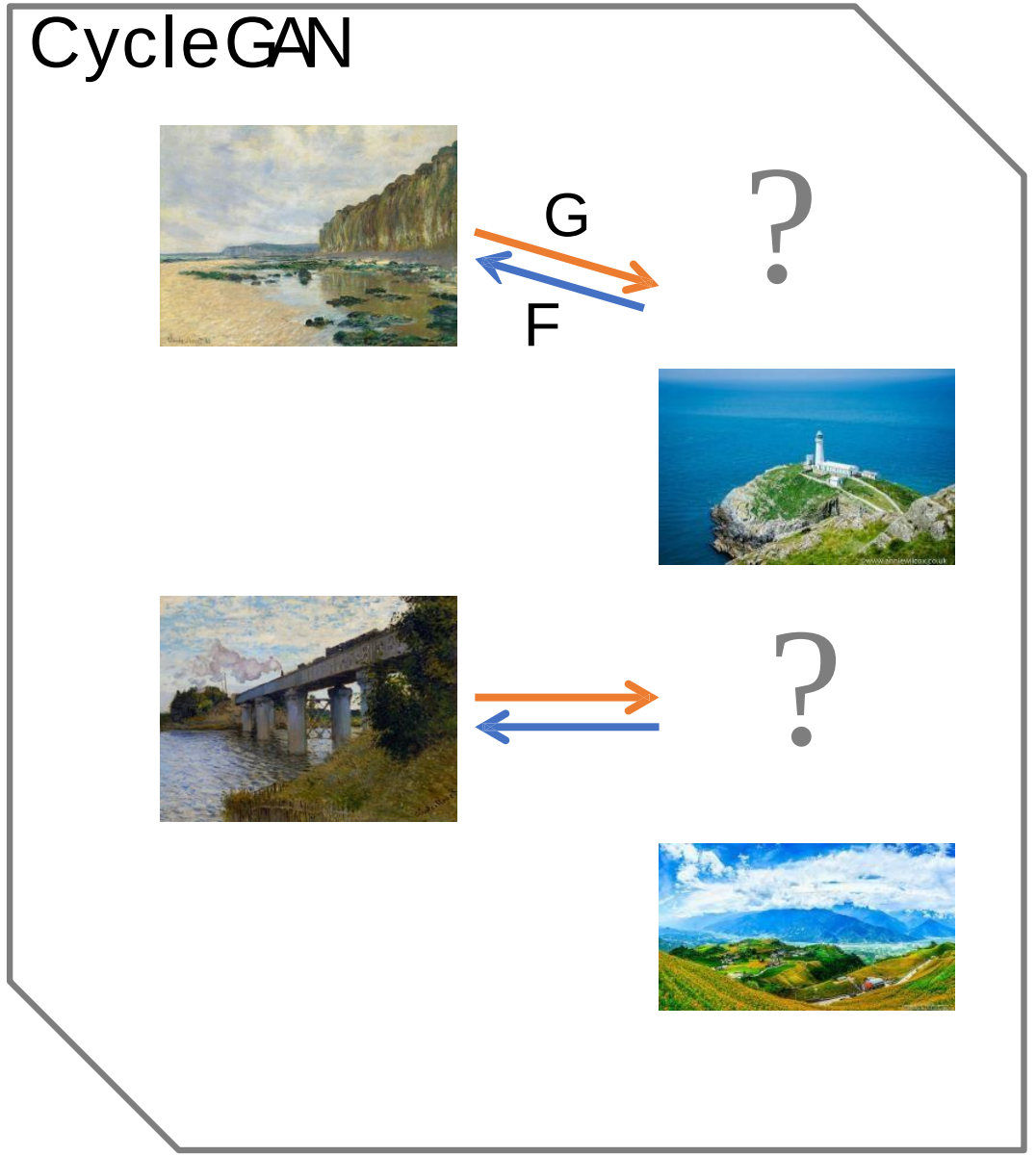
Loss

$$L_{GAN}\ (G(x), y) + \left\| F\left(G(x)\right) - x \right\|_1$$

G(x) should just look photorealistic

and *F(G(x))* should be *F(G(x))* = *x*,

where *F* is the inverse deep network

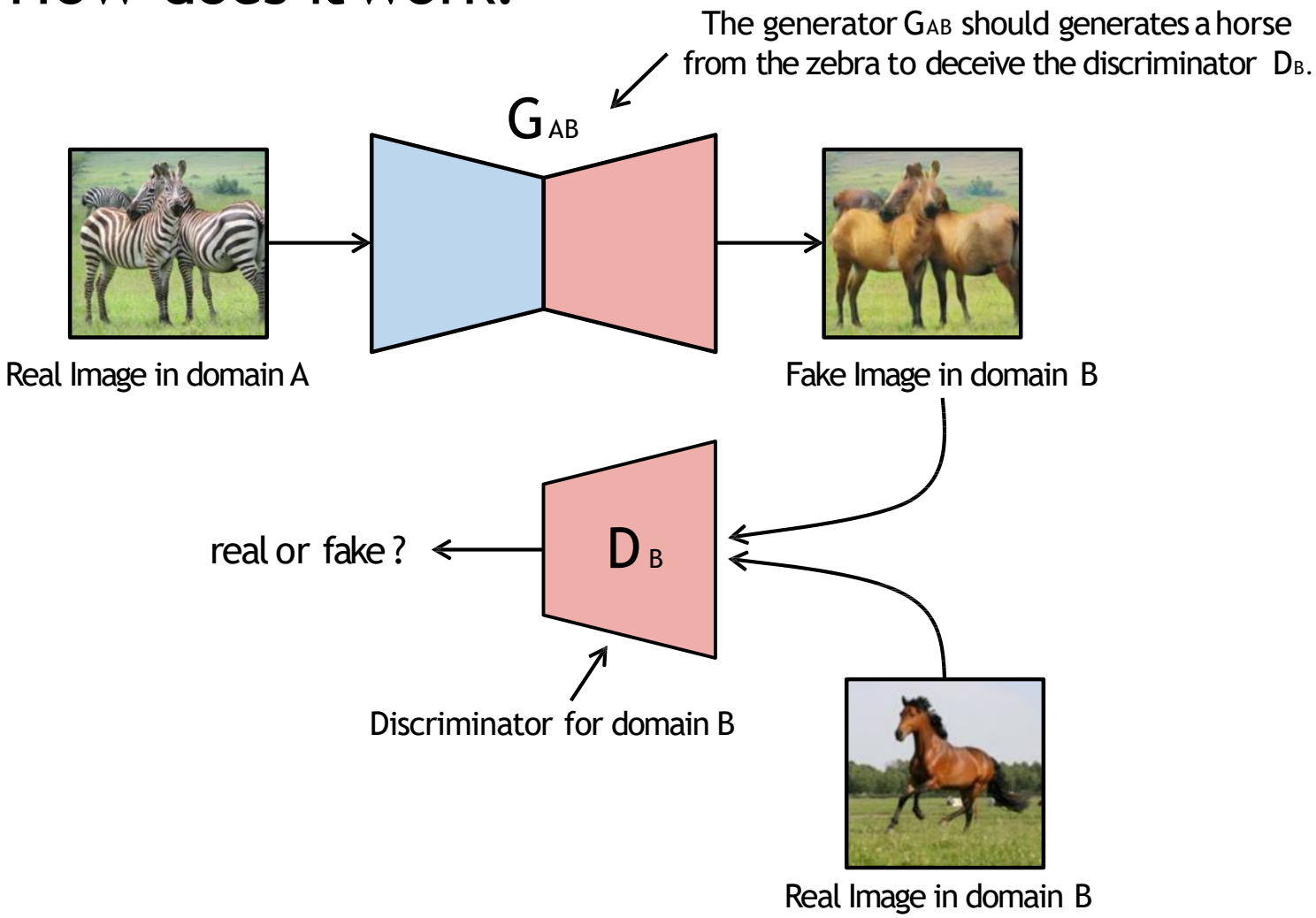


Ablation Study on Cityscapes dataset

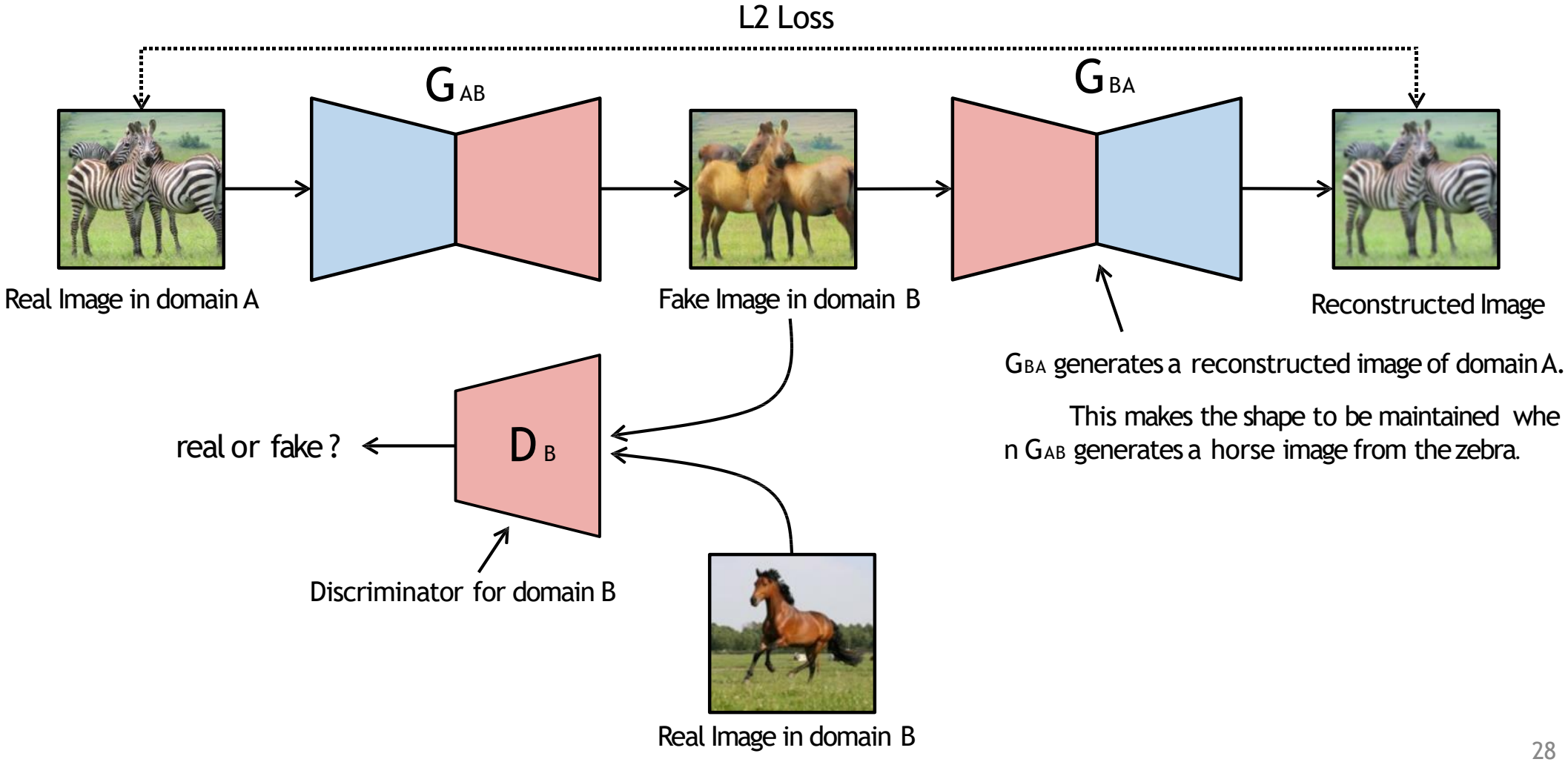
$$L_{GAN}\ (G(x), y) + \|F(G(x)) - x\|_1 \quad + \quad L_{GAN}\ (F(y), x) + \|G(F(y)) - y\|_1$$

CycleGAN

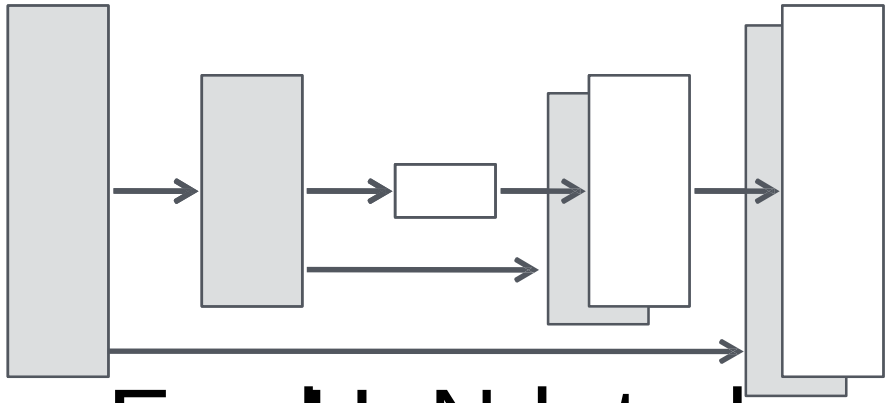
- How does it work?



- How does it work?



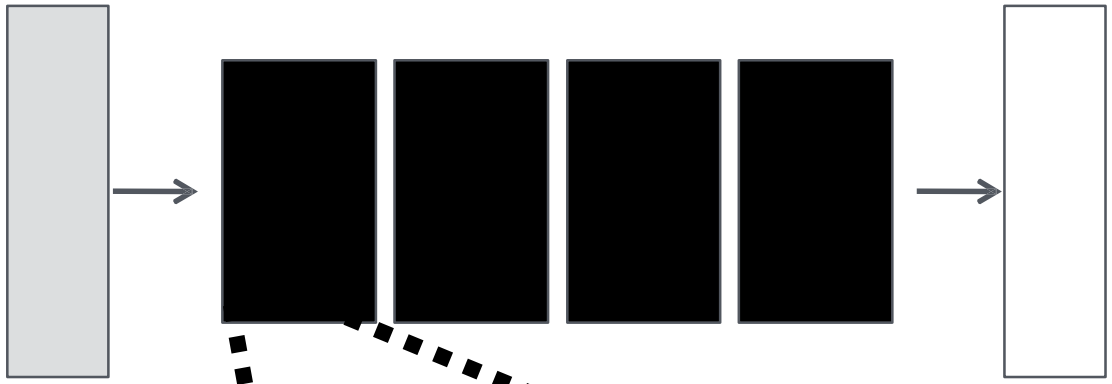
Training Details: Generator G

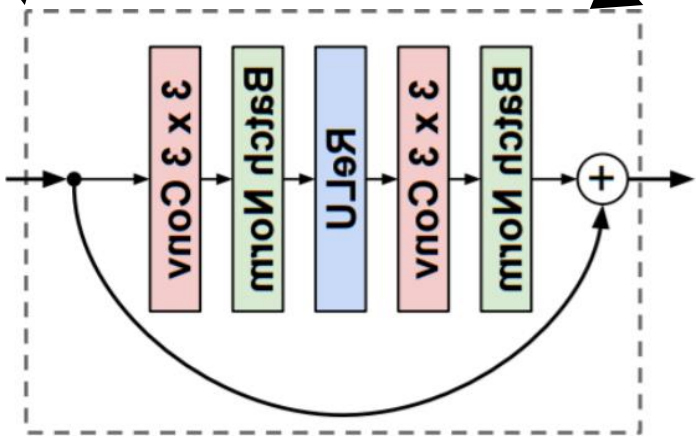


U-Net: Encoder-Decoder

[Ronneberger et al.]

ResNet [He et al.] [Johnson et al.]



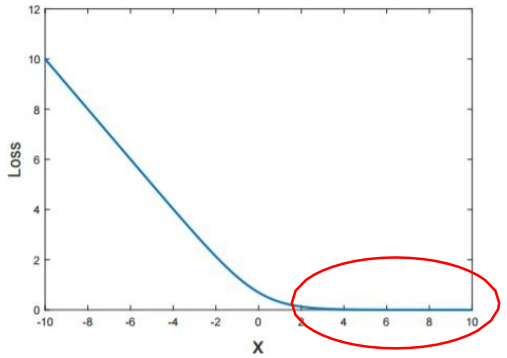


- Both have skip connections
- ResNet: fewer parameters better for ill-posed problems

Training Details: Objective

- GANs with cross-entropy loss

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))],$$

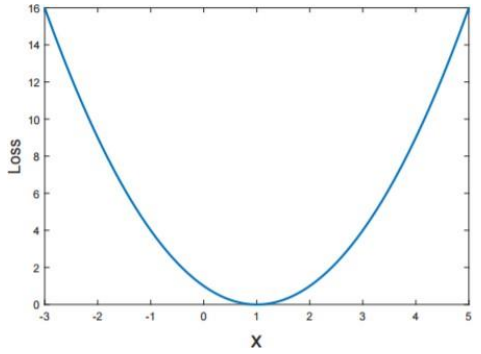


Vanishing gradients

- Least square GANs [Mao et al. 2016]

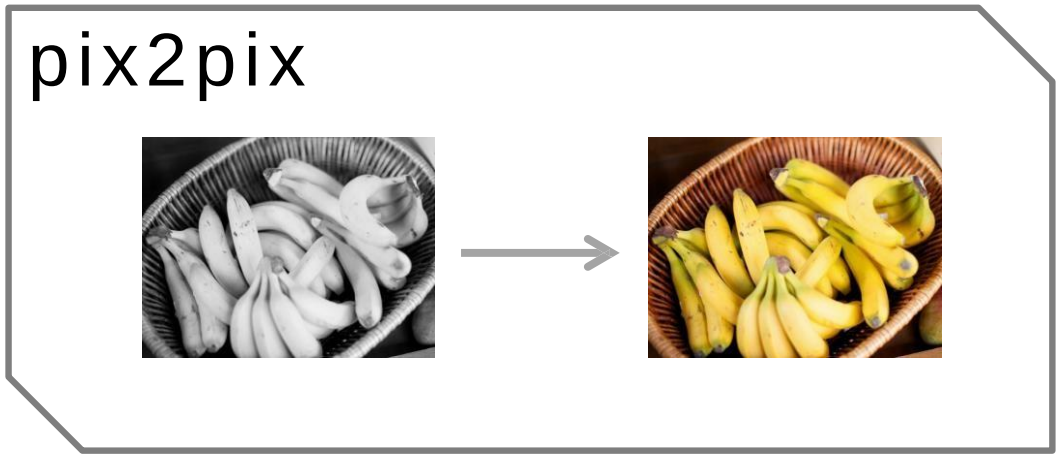
  Stable training + better results

$$\mathcal{L}_{\text{LSGAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[(D_Y(y) - 1)^2] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[D_Y(G(x))^2],$$



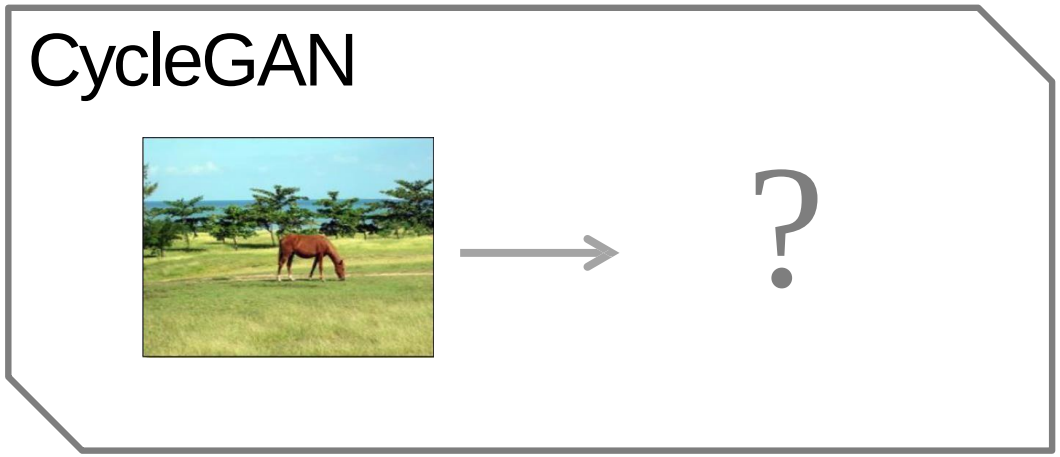
Combine with as much L1 loss as possible

- L1 loss as a *stable guiding force* in GAN training



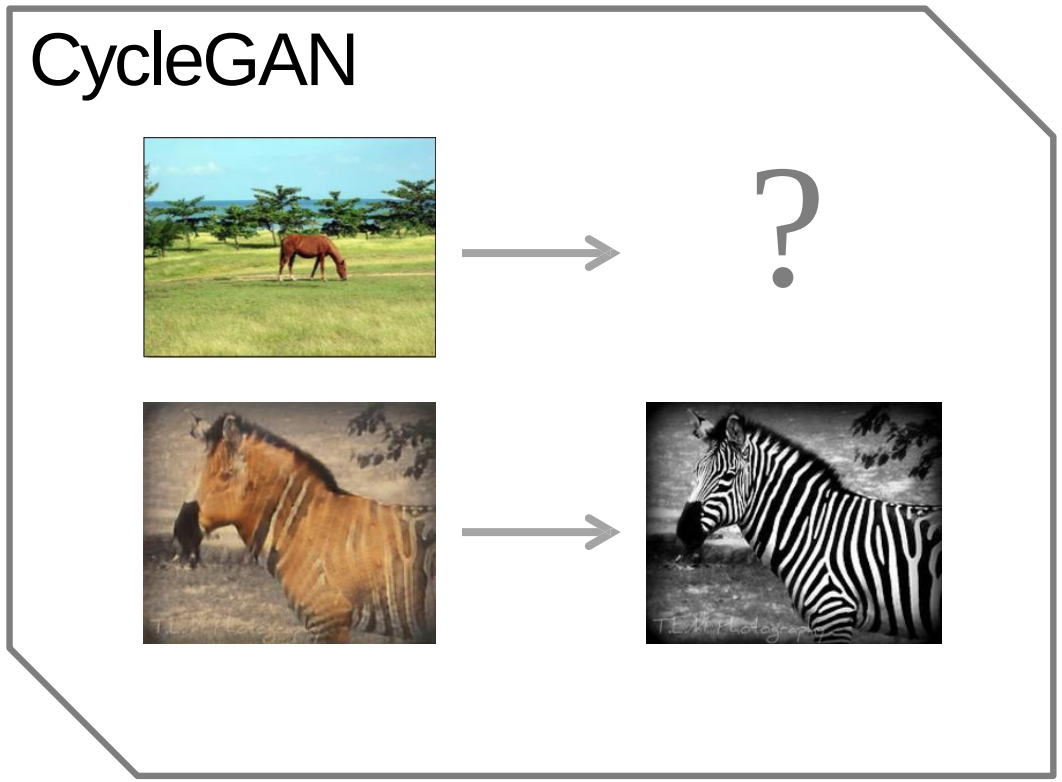
Combine with as much L1 loss as possible

- L1 loss as a *stable guiding force* in GAN training



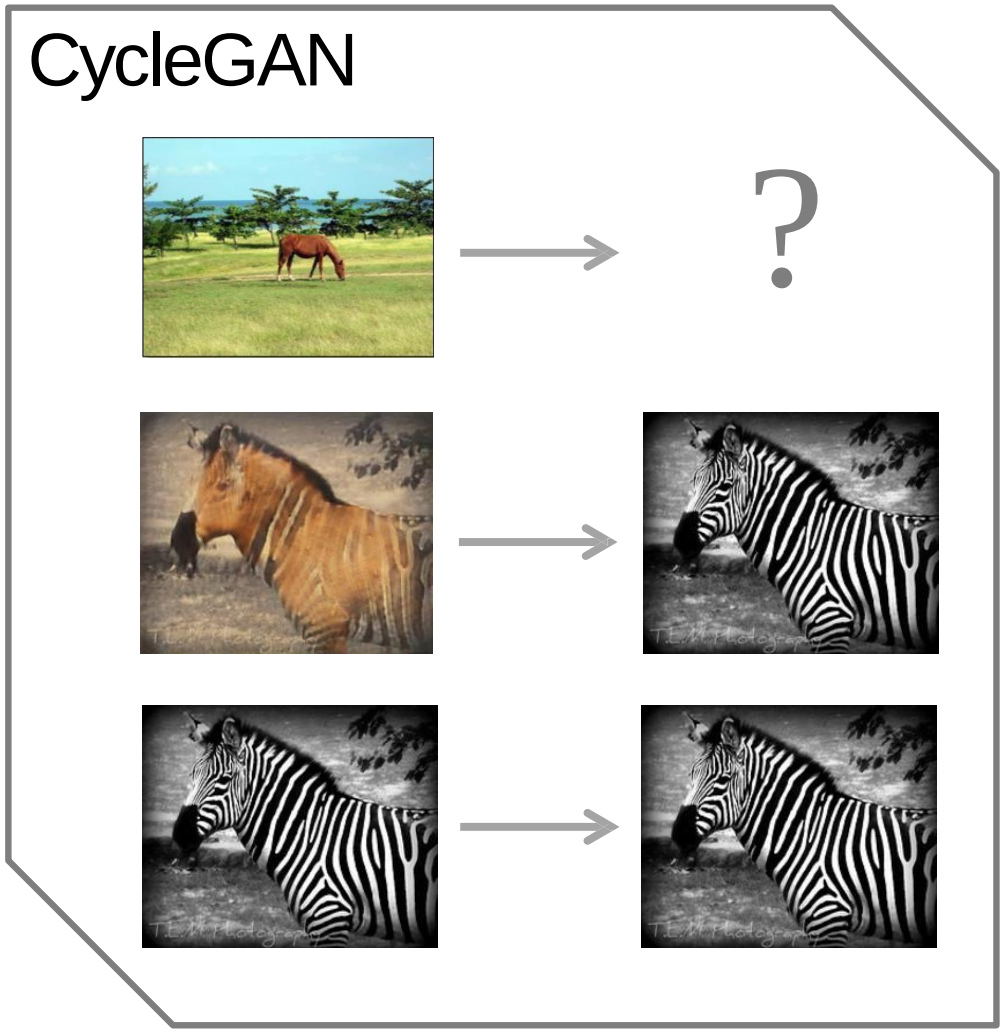
Combine with as much L1 loss as possible

- L1 loss as a *stable guiding force* in GAN training



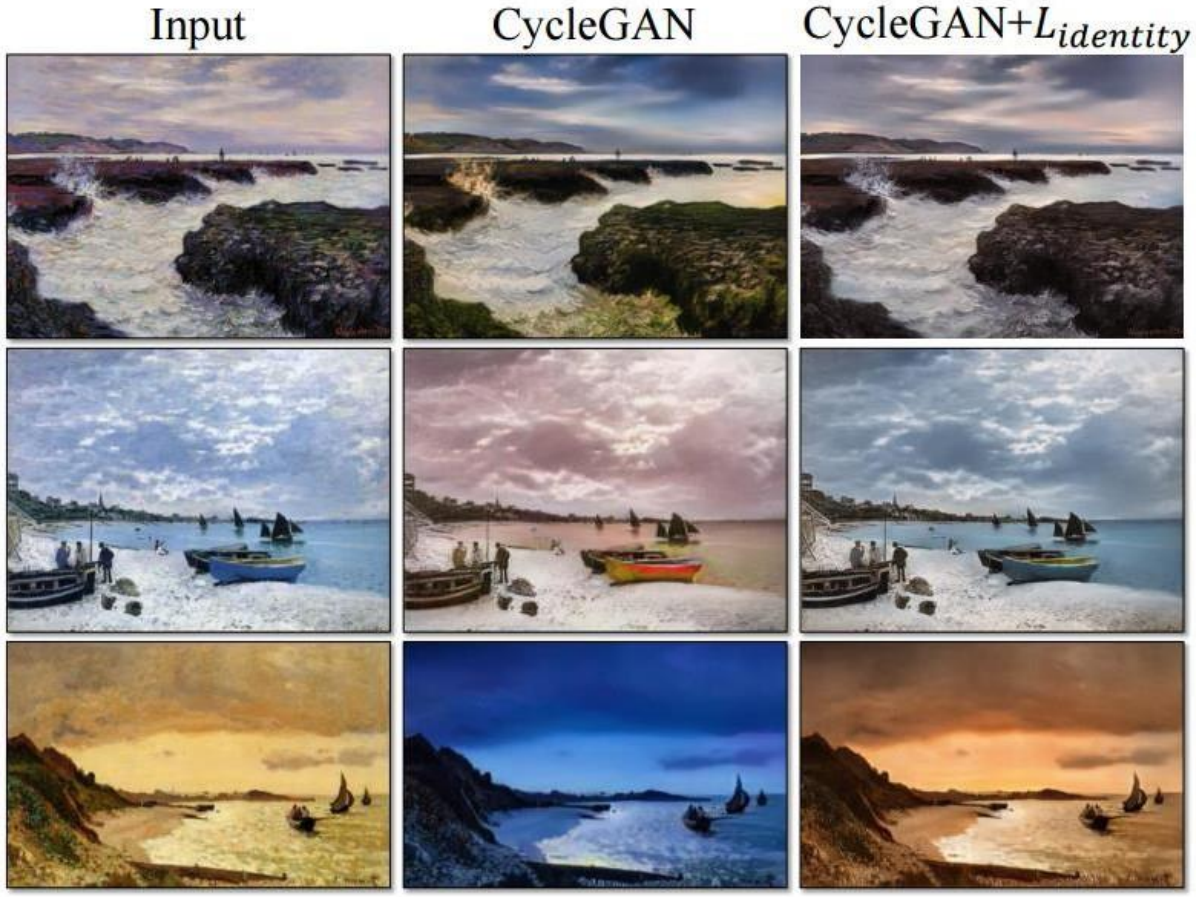
Combine with as much L1 loss as possible

- L1 loss as a *stable guiding force* in GAN training



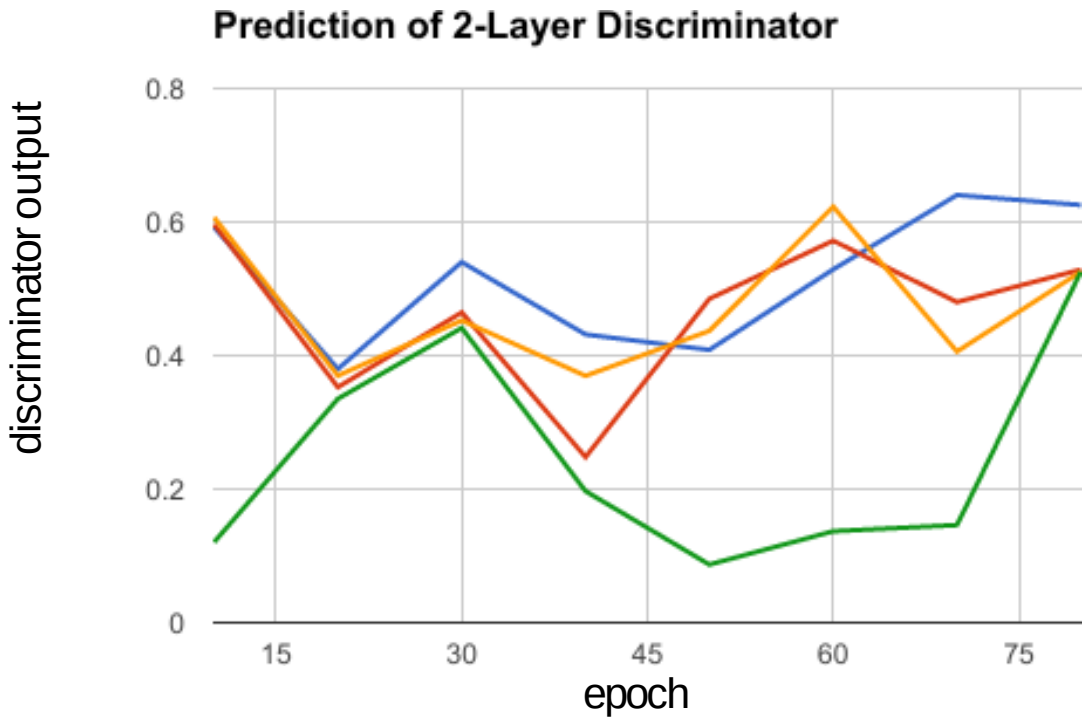
Combine with as much L1 loss as possible

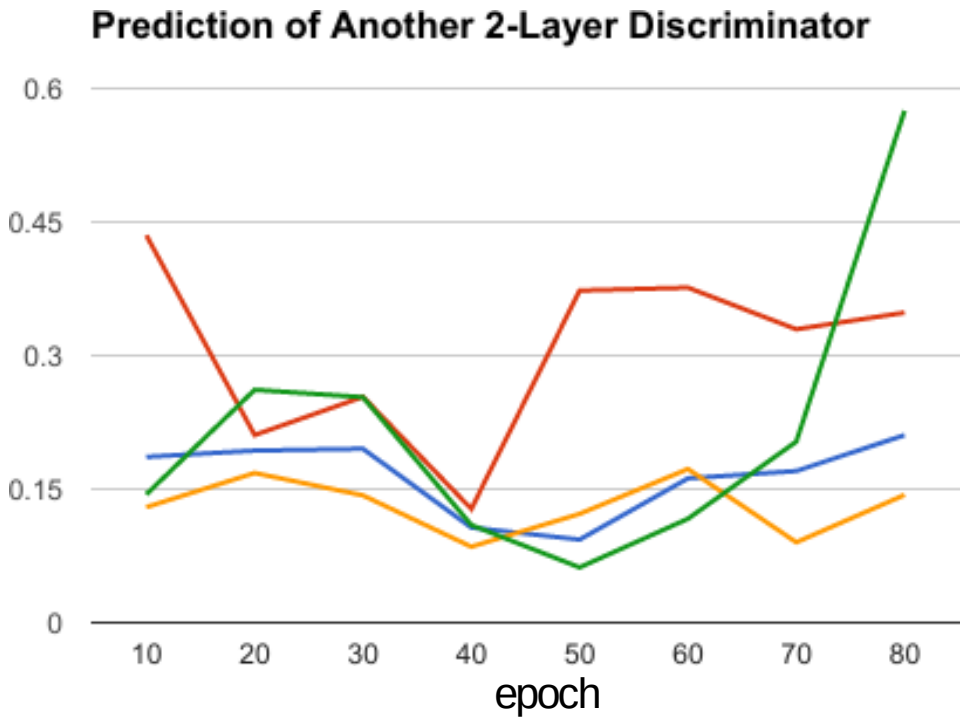
- L1 loss as a *stable guiding force* in GAN training



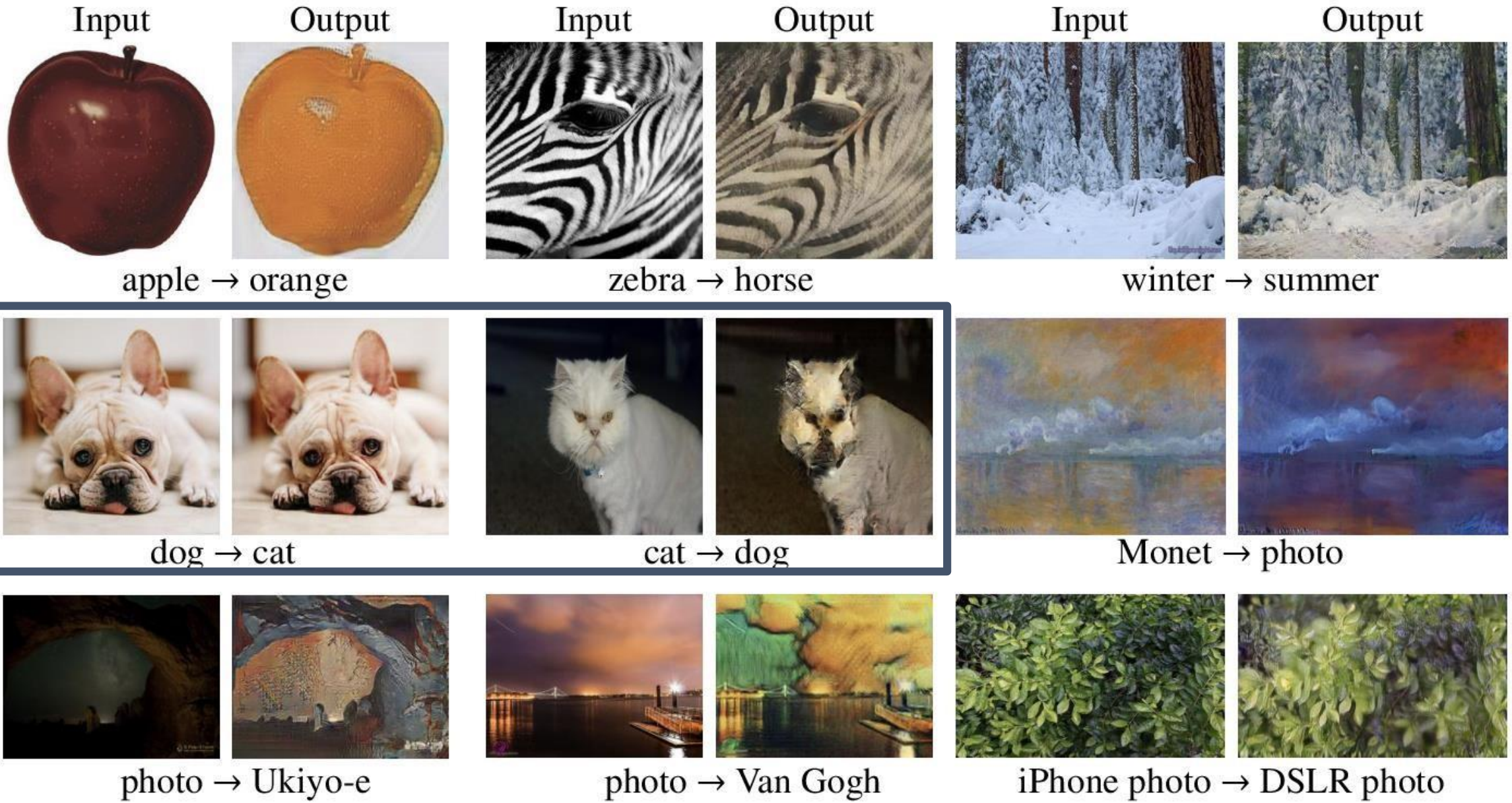
Training Details: replay buffer

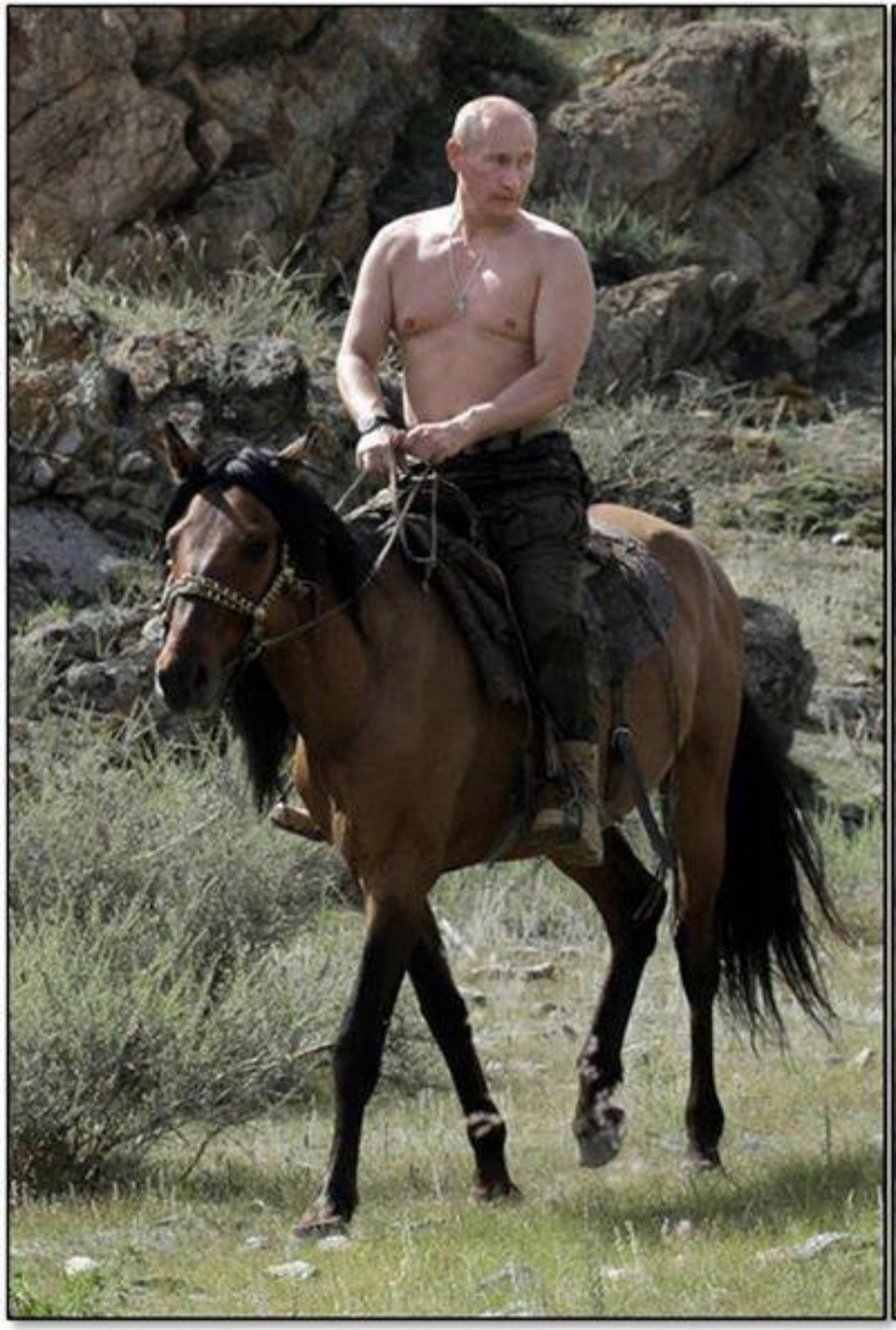
- … because the discriminators can take very different trajectories in training

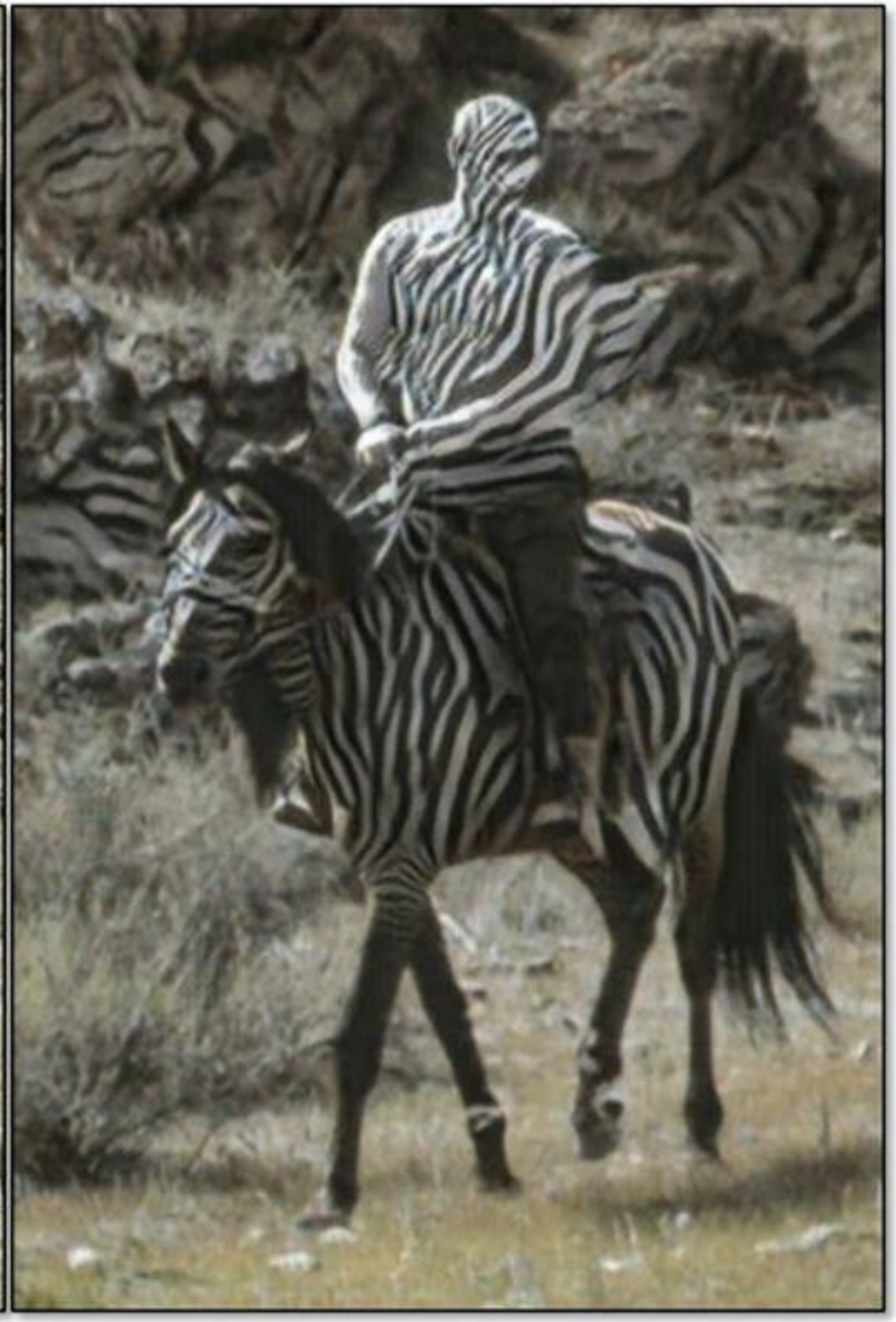




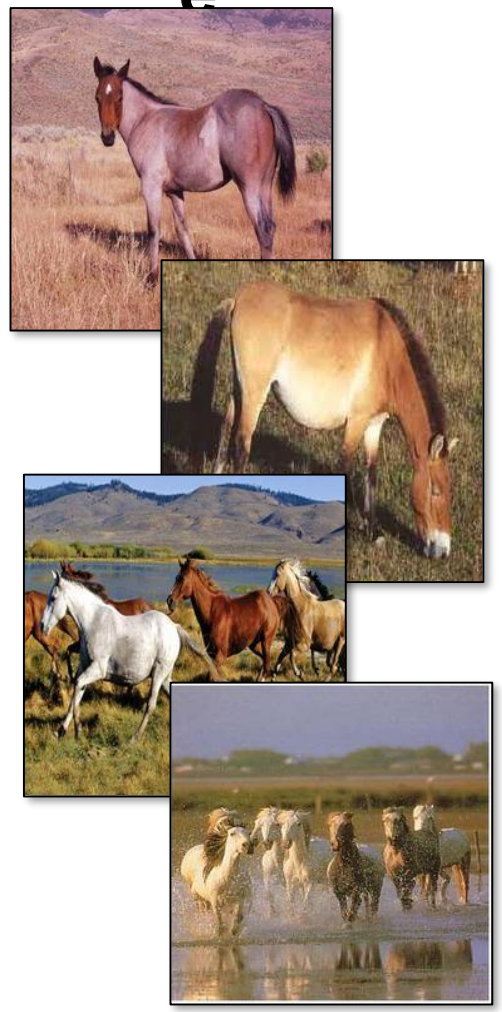
Failure cases





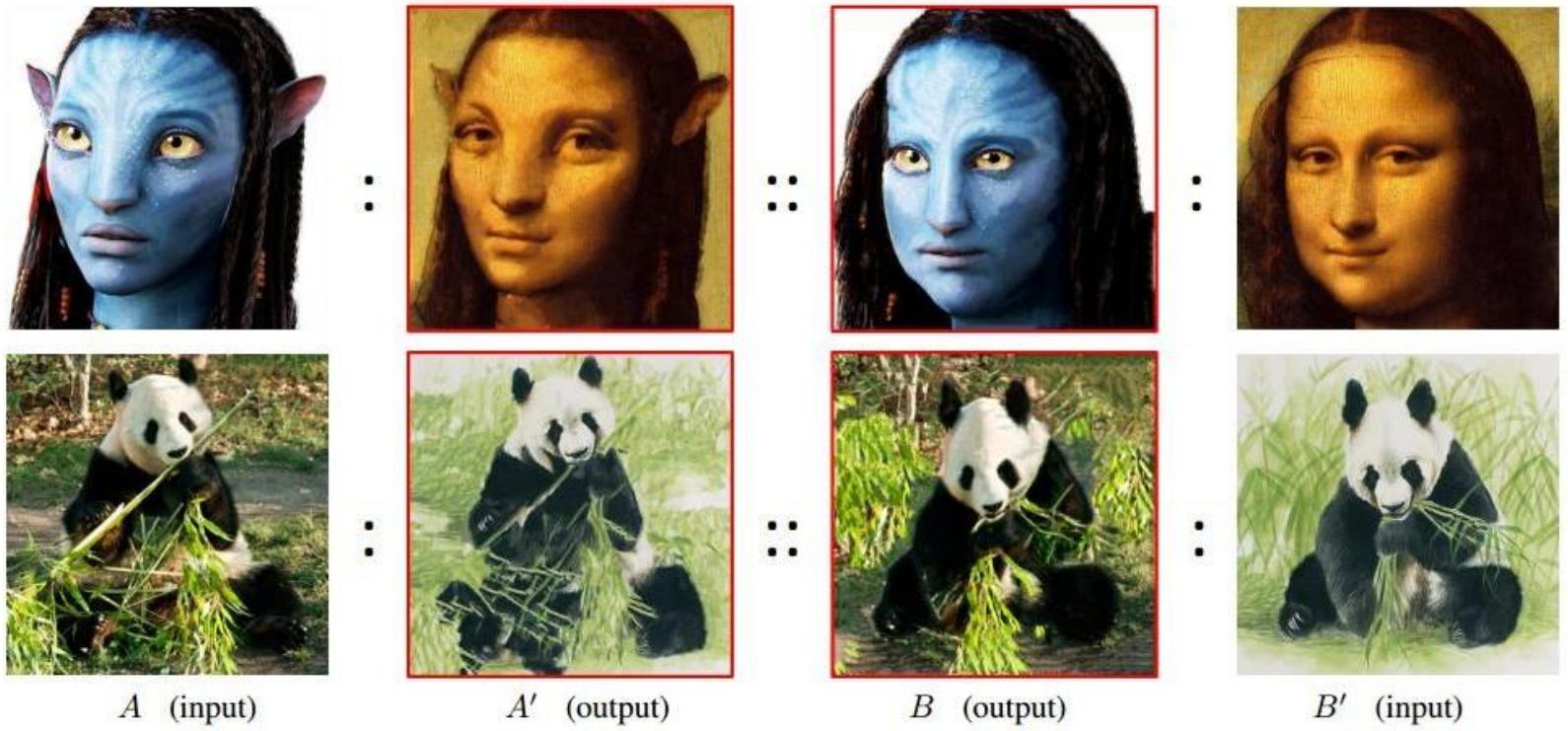


ImageNet “Wild horse”



Latest Work

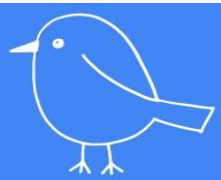
- Visual Attribute Transfer



Jing Liao et al. Visual Attribute Transfer through Deep Image Analogy, 2017

Latest Work

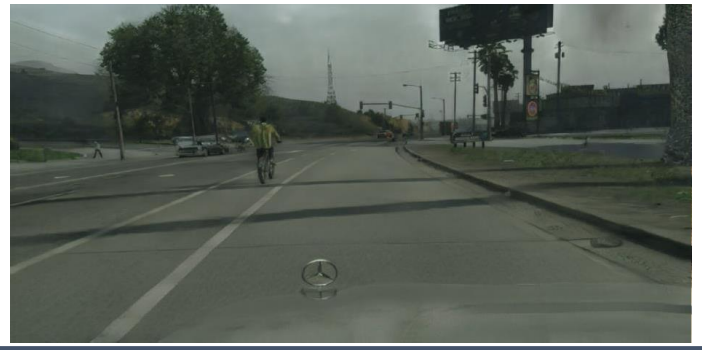
Extensions



- User-Interactive Image Colorization

Richard Zhang et al. Real-Time User-Guided Image Colorization with Learned Deep Prioirs, 2017

Domain Adaption with CycleGAN



Train on CycleGAN data

Test on real images

	Per-class accuracy	Per-pixel accuracy
Oracle (Train and test on Real)	60.3	93.1
Train on CG, test on Real	17.9	54.0
FCN in the wild [Hoffman et al.]	27.1 (+6.0)	-
Train on CycleGAN, test on Real	**34.8 (+16.9)**	**82.8**

[Tzeng et al.] In submission, Slides from Junyan Zhu

StarGANs

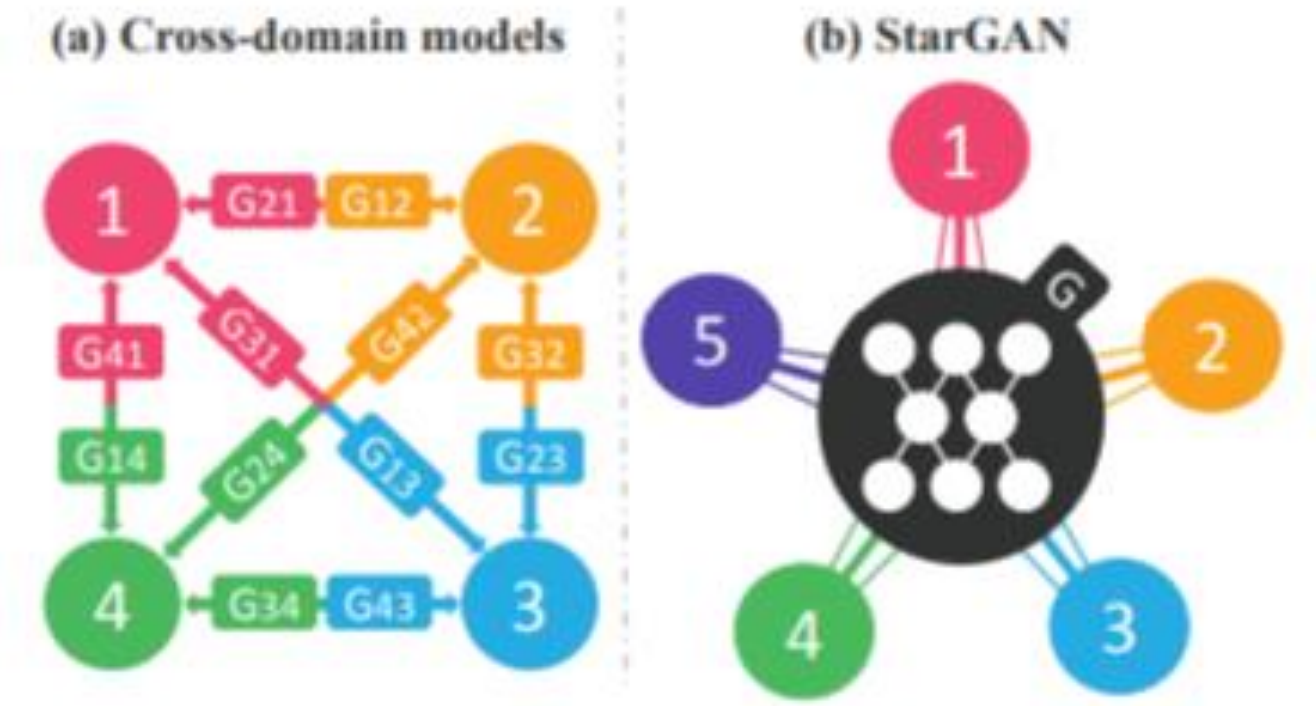


Figure 2. Comparison between cross-domain models and our proposed model, StarGAN. (a) To handle multiple domains, cross-domain models should be built for every pair of image domains. (b) StarGAN is capable of learning mappings among multiple domains using a single generator. The figure represents a star topology connecting multi-domains.

StarGANs

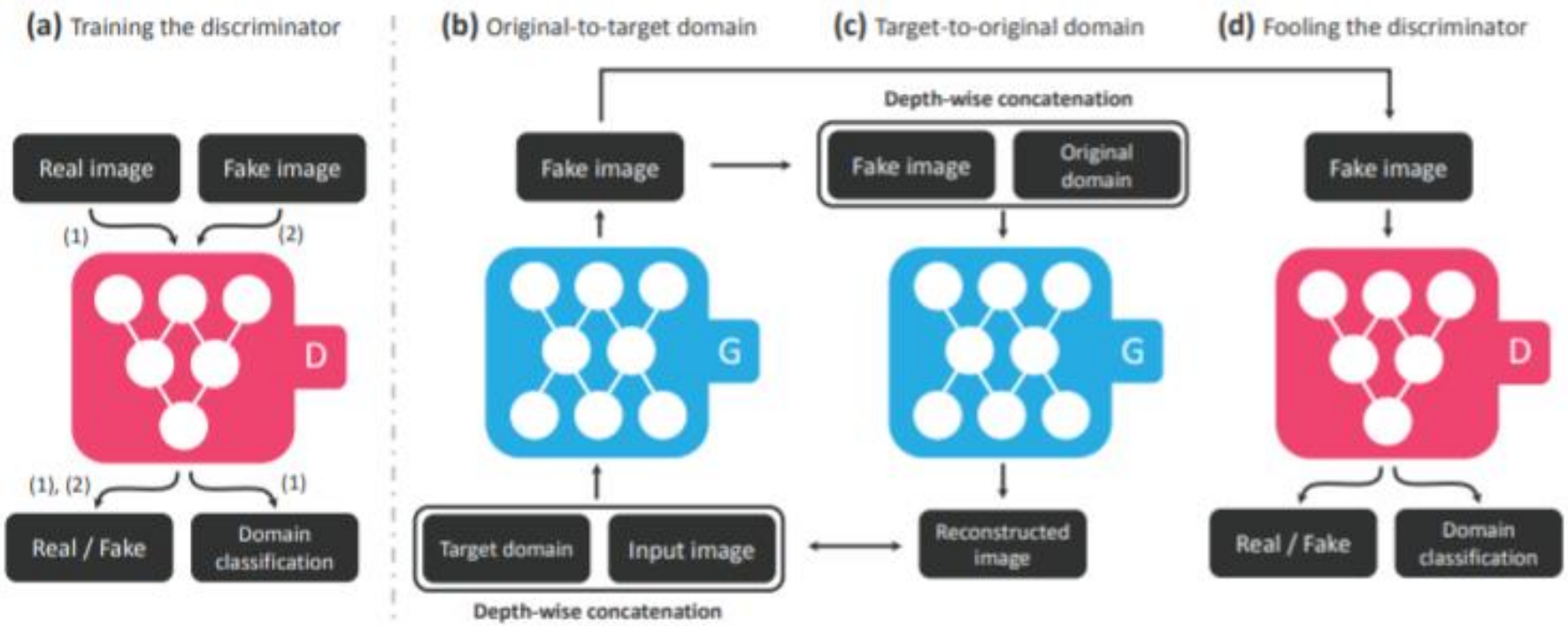
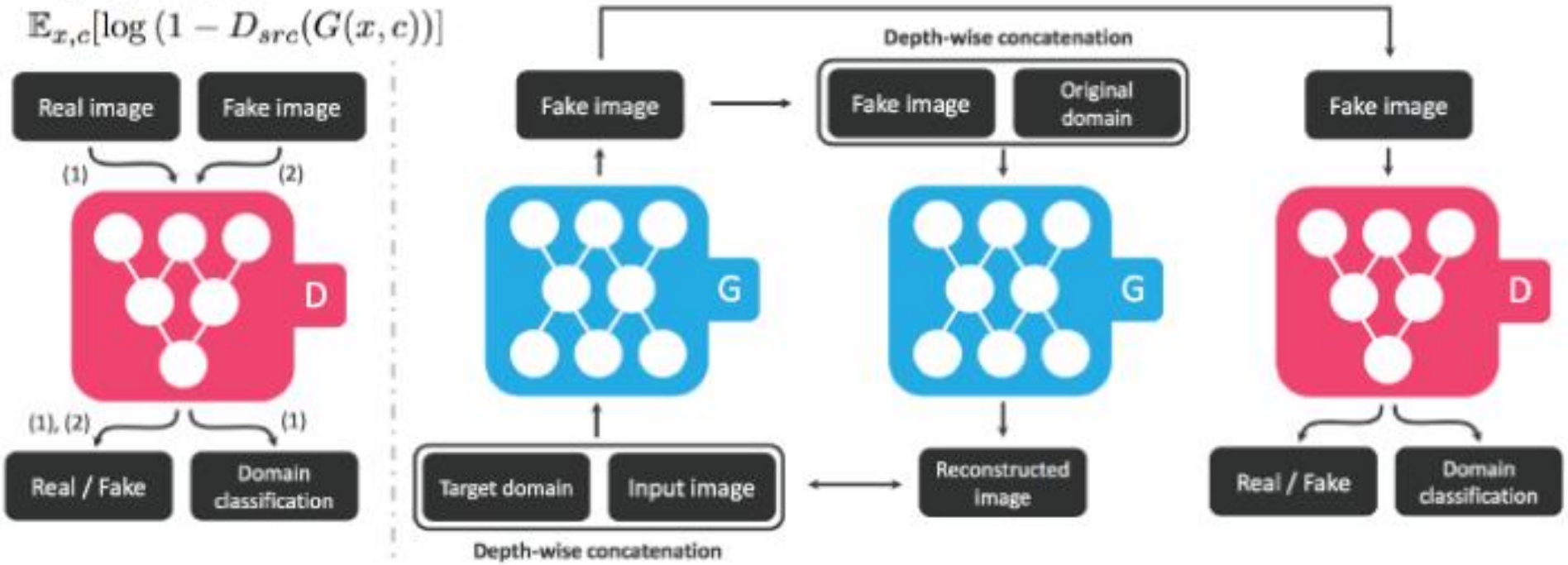


Figure 3. Overview of StarGAN, consisting of two modules, a discriminator D and a generator G. **(a)** D learns to distinguish between real and fake images and classify the real images to its corresponding domain. **(b)** G takes in as input both the image and target domain label and generates an fake image. The target domain label is spatially replicated and concatenated with the input image. **(c)** G tries to reconstruct the original image from the fake image given the original domain label. **(d)** G tries to generate images indistinguishable from real images and classifiable as target domain by D.

$$\mathcal{L}_{adv} = \mathbb{E}_x [\log D_{src}(x)] + \mathbb{E}_{x,c}[\log (1 - D_{src}(G(x,c))]$$

$$\mathcal{L}^f_{cls} = \mathbb{E}_{x,c}[-\log D_{cls}(c|G(x,c))]$$



$$\mathcal{L}^r_{cls} = \mathbb{E}_{x,c'}[-\log D_{cls}(c'|x)]$$

$$\mathcal{L}_{rec} = \mathbb{E}_{x,c,c'}[||x - G(G(x,c),c')||_1]$$

$$\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls}\,\mathcal{L}^r_{cls},$$

$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls}\,\mathcal{L}^f_{cls} + \lambda_{rec}\,\mathcal{L}_{rec},$$

StarGANs

Adversarial Loss. To make the generated images indistinguishable from real images, we adopt an adversarial loss

$$\mathcal{L}_{adv} = \mathbb{E}_x [\log D_{src}(x)] + \mathbb{E}_{x,c}[\log (1 - D_{src}(G(x, c)))], \quad (1)$$

StarGANs

Domain Classification Loss. For a given input image x and a target domain label c, our goal is to translate x into an output image y, which is properly classified to the target domain c. To achieve this condition, we add an auxiliary classifier on top of D and impose the domain classification loss when optimizing both D and G. That is, we decompose the objective into two terms: a domain classification loss of real images used to optimize D, and a domain classification loss of fake images used to optimize G. In detail, the former is defined as

$$\mathcal{L}_{cls}^{r} = \mathbb{E}_{x,c'}[-\log D_{cls}(c'|x)], \tag{2}$$

where the term $D_{cls}(c'|x)$ represents a probability distribution over domain labels computed by D. By minimizing this objective, D learns to classify a real image x to its corresponding original domain c'. We assume that the input image and domain label pair (x, c') is given by the training data. On the other hand, the loss function for the domain classification of fake images is defined as

$$\mathcal{L}_{cls}^{f} = \mathbb{E}_{x,c}[-\log D_{cls}(c|G(x,c))]. \tag{3}$$

In other words, G tries to minimize this objective to generate images that can be classified as the target domain c.

$$\mathcal{L}_{cls}^{r} = \mathbb{E}_{x,c'}[-\log D_{cls}(c'|x)], \tag{2}$$

$$\mathcal{L}_{cls}^{f} = \mathbb{E}_{x,c}[-\log D_{cls}(c|G(x,c))]. \tag{3}$$

StarGANs

Reconstruction Loss. By minimizing the adversarial and classification losses, G is trained to generate images that are realistic and classified to its correct target domain. However, minimizing the losses (Eqs. (1) and (3)) does not guarantee that translated images preserve the content of its input images while changing only the domain-related part of the inputs. To alleviate this problem, we apply a cycle consistency loss [9, 33] to the generator, defined as

$$\mathcal{L}_{rec} = \mathbb{E}_{x,c,c'}[||x - G(G(x,c),c')||_1], \quad (4)$$

where G takes in the translated image $G(x,c)$ and the original domain label c' as input and tries to reconstruct the original image x. We adopt the L1 norm as our reconstruction loss. Note that we use a single generator twice, first to translate an original image into an image in the target domain and then to reconstruct the original image from the translated image.

StarGANs

Full Objective. Finally, the objective functions to optimize G and D are written, respectively, as

$$\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls}\,\mathcal{L}^r_{cls}, \tag{5}$$

$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls}\,\mathcal{L}^f_{cls} + \lambda_{rec}\,\mathcal{L}_{rec}, \tag{6}$$

where λ_{cls} and λ_{rec} are hyper-parameters that control the relative importance of domain classification and reconstruction losses, respectively, compared to the adversarial loss. We use $\lambda_{cls} = 1$ and $\lambda_{rec} = 10$ in all of our experiments.

StarGANs

3.2. Training with Multiple Datasets

An important advantage of StarGAN is that it simultaneously incorporates multiple datasets containing different types of labels, so that StarGAN can control all the labels at the test phase. An issue when learning from multiple datasets, however, is that the label information is only *partially known* to each dataset. In the case of CelebA [19] and RaFD [13], while the former contains labels for attributes such as hair color and gender, it does not have any labels for facial expressions such as 'happy' and 'angry', and vice versa for the latter. This is problematic because the complete information on the label vector c' is required when reconstructing the input image x from the translated image $G(x, c)$ (See Eq. (4)).

StarGANs

Mask Vector. To alleviate this problem, we introduce a mask vector m that allows StarGAN to ignore unspecified labels and focus on the explicitly known label provided by a particular dataset. In StarGAN, we use an n-dimensional one-hot vector to represent m, with n being the number of datasets. In addition, we define a unified version of the label as a vector

$$\tilde{c} = [c_1, ..., c_n, m], \tag{7}$$

where $[\cdot]$ refers to concatenation, and c_i represents a vector for the labels of the i-th dataset. The vector of the known label c_i can be represented as either a binary vector for binary attributes or a one-hot vector for categorical attributes. For the remaining $n-1$ unknown labels we simply assign zero values. In our experiments, we utilize the CelebA and RaFD datasets, where n is two.

References

1. Namju Kim, Generative adversarial networks, https://www.slideshare.net/ssuser77ee21/generative-adversarial-networks-70896091?fbclid=IwAR0_i8VbVH--XWZjh-fD0haqcMTwPLTHvKZ2zGvB8xFXY_RxRHz2lVzFuv8
2. Yunjey Choi, 1시간 만에 GAN(Generative Adversarial Network) 완전 정복하기, https://www.slideshare.net/NaverEngineering/1-gangenerative-adversarial-network
3. 박태성, Finding connections among images using CycleGAN, https://www.slideshare.net/NaverEngineering/finding-connections-among-images-using-cyclegan?from_action=save
4. PR-001: Generative adversarial nets by Jaejun Yoo, https://www.youtube.com/watch?v=L3hz57whyNw&t=55s&index=2&list=PL0oFI08O71gKjGhaWctTPvvM7_cVzsAtK
5. Jaegul Choo, Image Generation 180703-3: pix2pix, CycleGAN, StarGAN, https://www.youtube.com/watch?v=v_IRC2e-IZI

Thank you.