Week 9: Generative Adversarial Networks 2

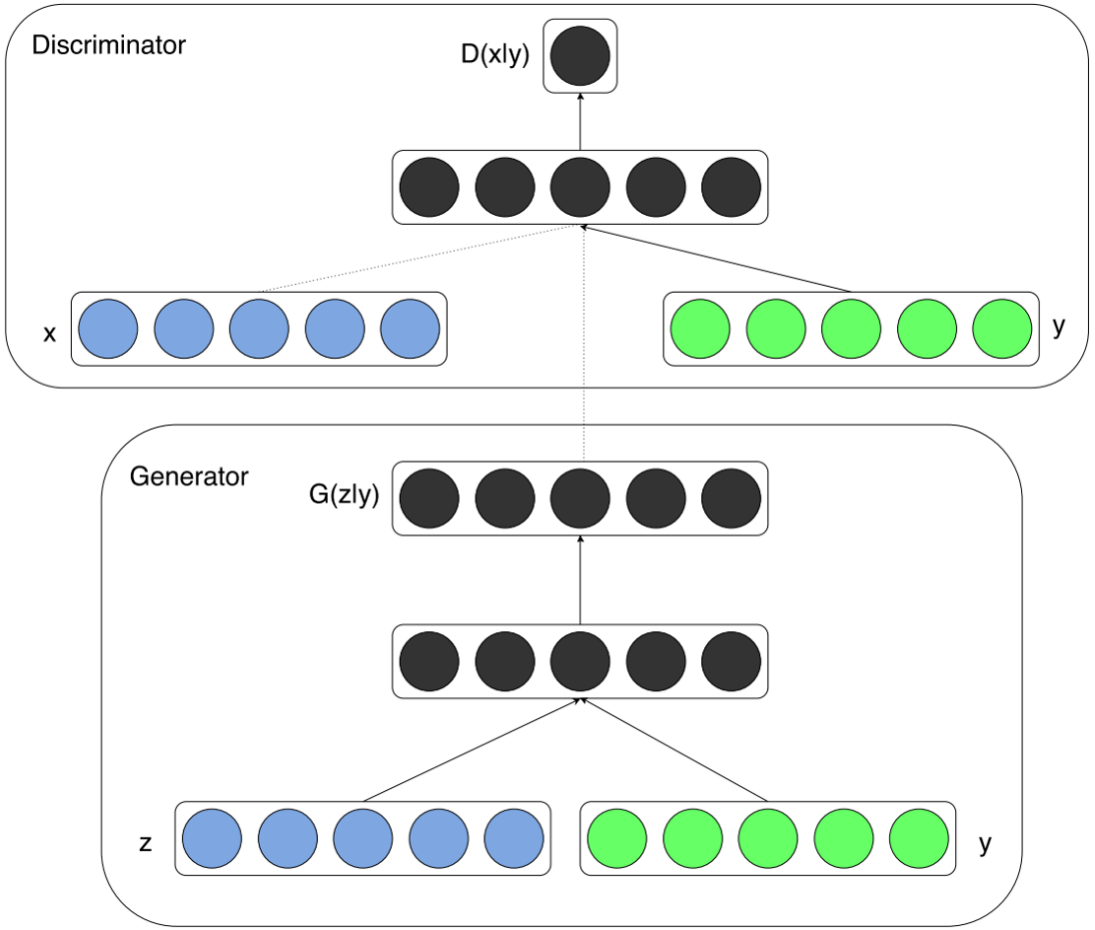
Instructor: Ruixuan Wang
wangruix5@mail.sysu.edu.cn

School of Data and Computer Science
Sun Yat-Sen University

25 April, 2019

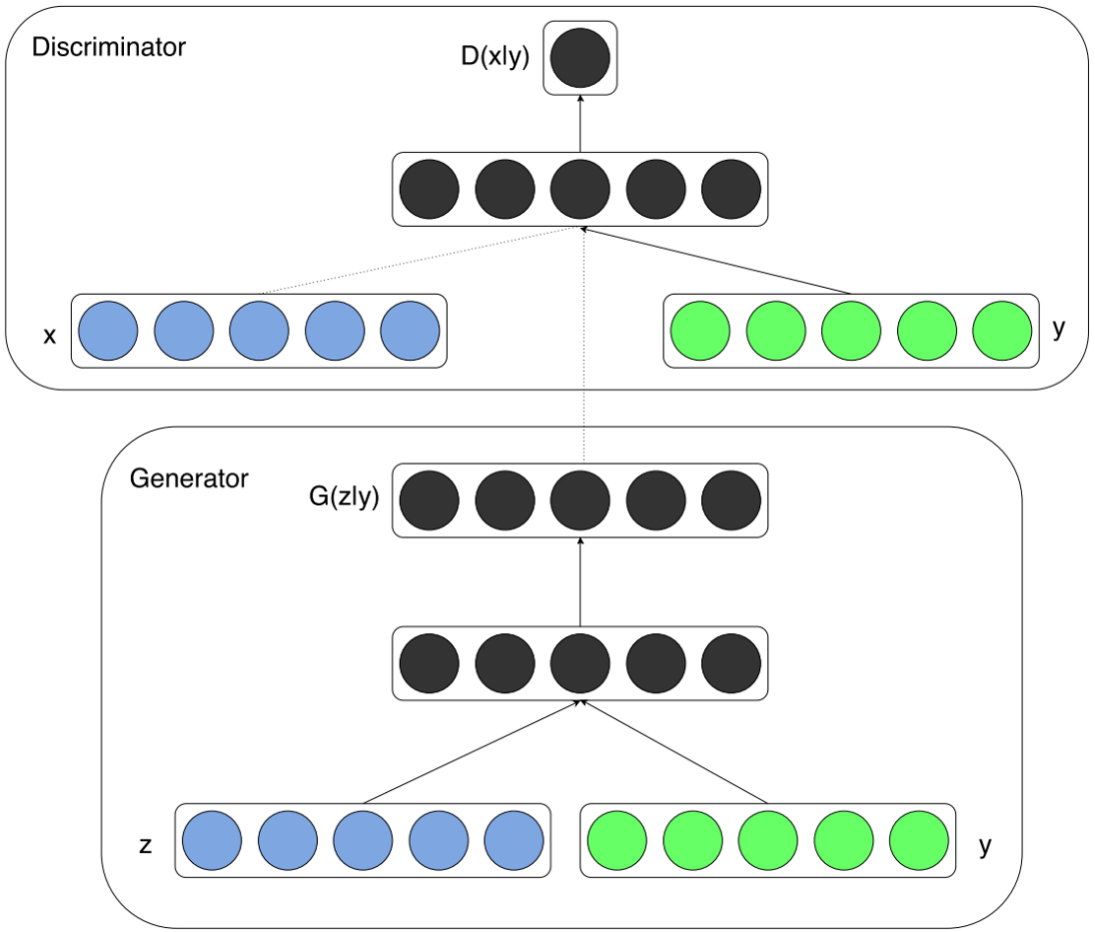
Conditional GAN

- Conditional GAN: code vector 'y' representing 'condition' is considered as part of input to both G and D.



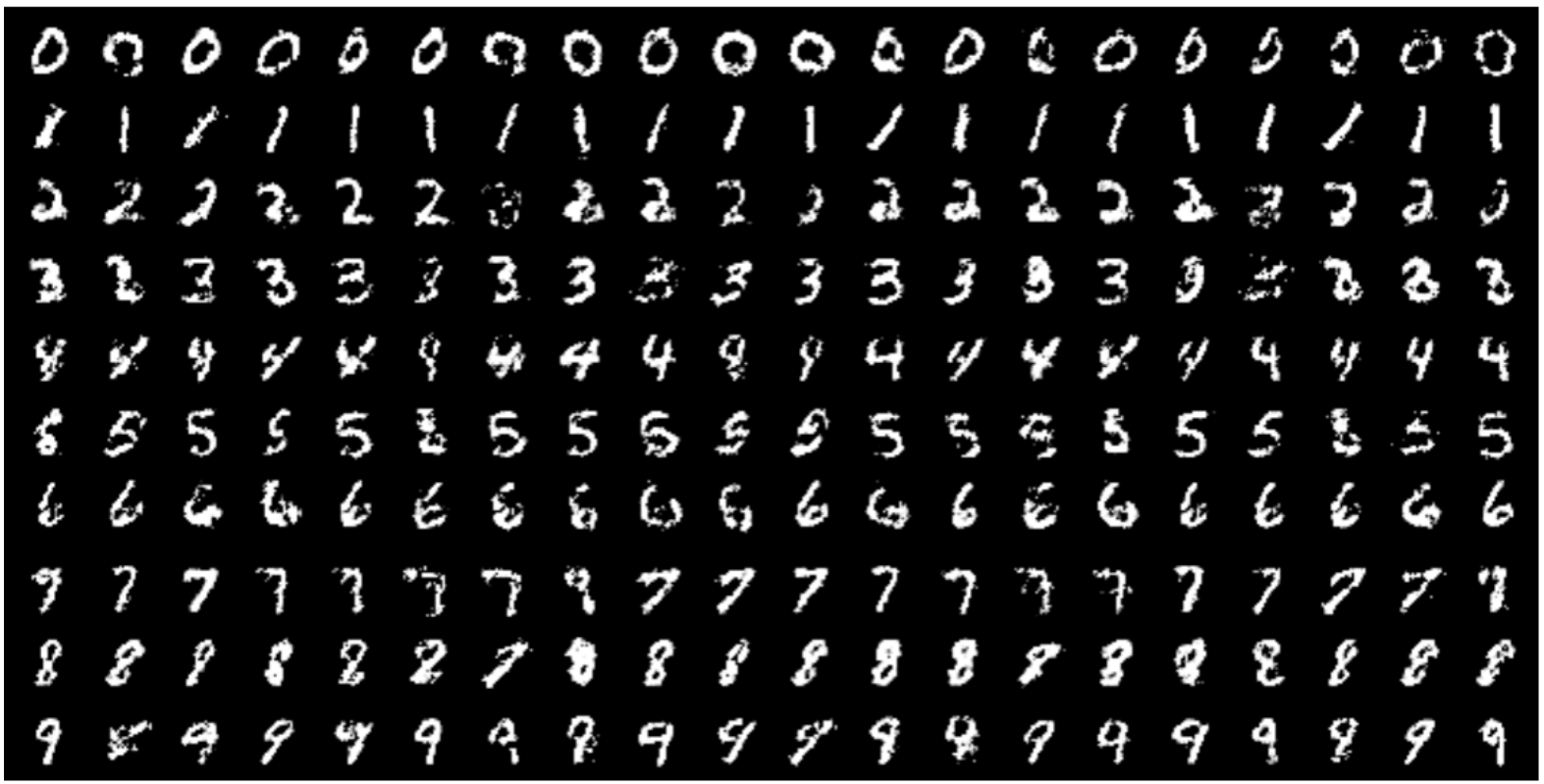
Conditional GAN

- Conditional GAN: code vector 'y' representing 'condition' is considered as part of input to both G and D.
- Condition 'y' could be a hot vector for class label.



Conditional GAN

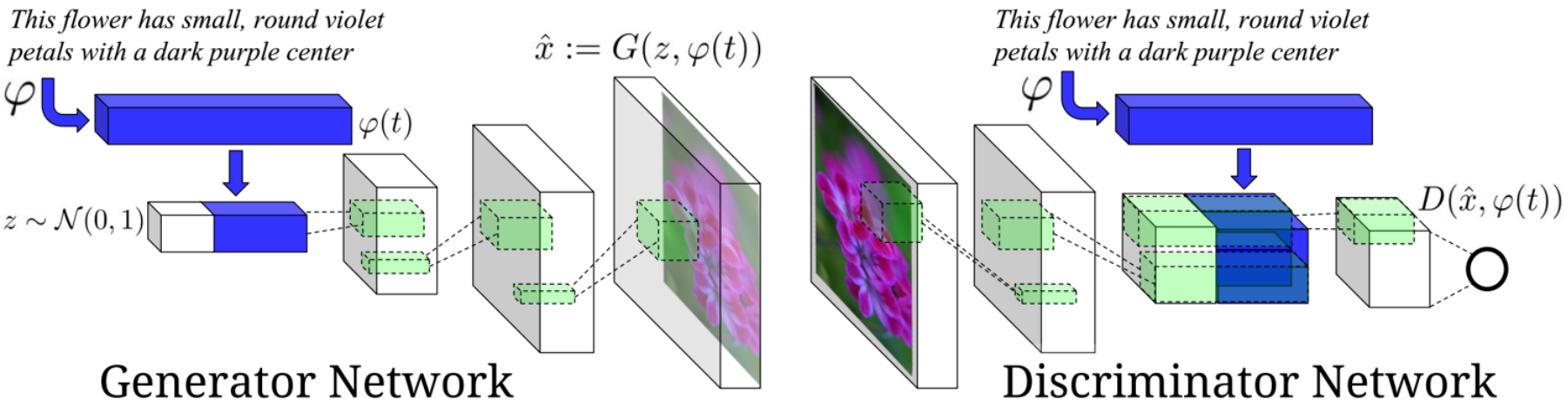
- Each row is conditioned on one digit class label, and each column is a different generated sample



Figures here and in the prev slide from Mirza, Osindero, "Conditional generative adversarial nets", arXiv, 2014

CGAN applications: text to image

- Transform a sentence (text) to a code vector as 'condition'
- In Generator: 'condition' as part of input
- In Discriminator: 'condition' close to output



Figures and table here and in the next 3 slides from Reed, Akata, Yan, Logeswaran, Schiele, Lee "Generative adversarial text to image synthesis", ICML, 2016

CGAN for text to image: alogorithm

1: **Input:** minibatch images x, matching text t, mismatching $\hat{t}$, number of training batch steps S
2: **for** $n = 1$ **to** S **do**
3: $h \leftarrow \varphi(t)$ {Encode matching text description}
4: $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode mis-matching text description}
5: $z \sim \mathcal{N}(0, 1)^Z$ {Draw sample of random noise}
6: $\hat{x} \leftarrow G(z, h)$ {Forward through generator}
7: $s_r \leftarrow D(x, h)$ {real image, right text}
8: $s_w \leftarrow D(x, \hat{h})$ {real image, wrong text}
9: $s_f \leftarrow D(\hat{x}, h)$ {fake image, right text}
10: $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1 - s_w) + \log(1 - s_f))/2$
11: $D \leftarrow D - \alpha \partial \mathcal{L}_D / \partial D$ {Update discriminator}
12: $\mathcal{L}_G \leftarrow \log(s_f)$
13: $G \leftarrow G - \alpha \partial \mathcal{L}_G / \partial G$ {Update generator}
14: **end for**

'Real' data

'Fake' data

CGAN for text to image: result

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

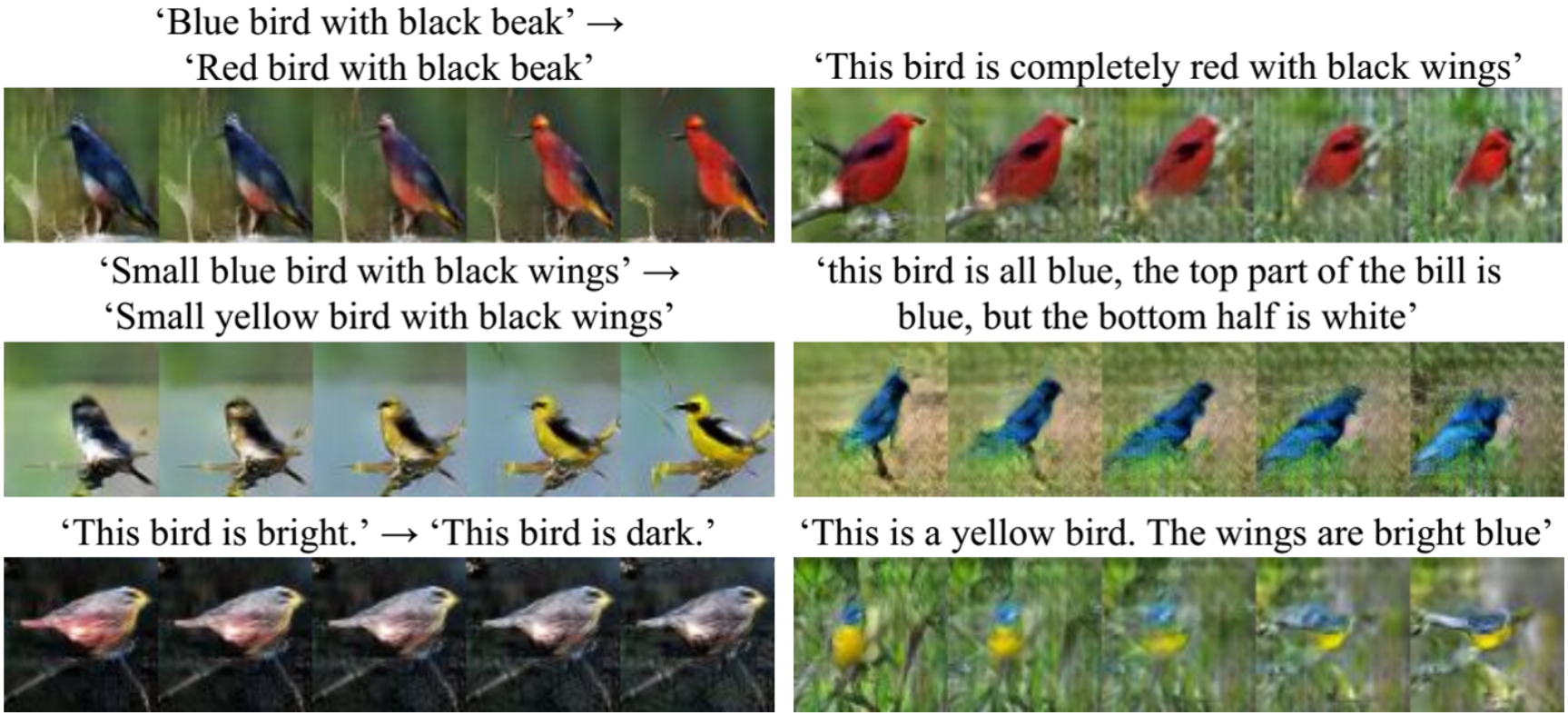
this magnificent fellow is almost all black with a red crest, and white cheek patch.

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

CGAN for text to image: result

- Interpolations within input space is meaningful!



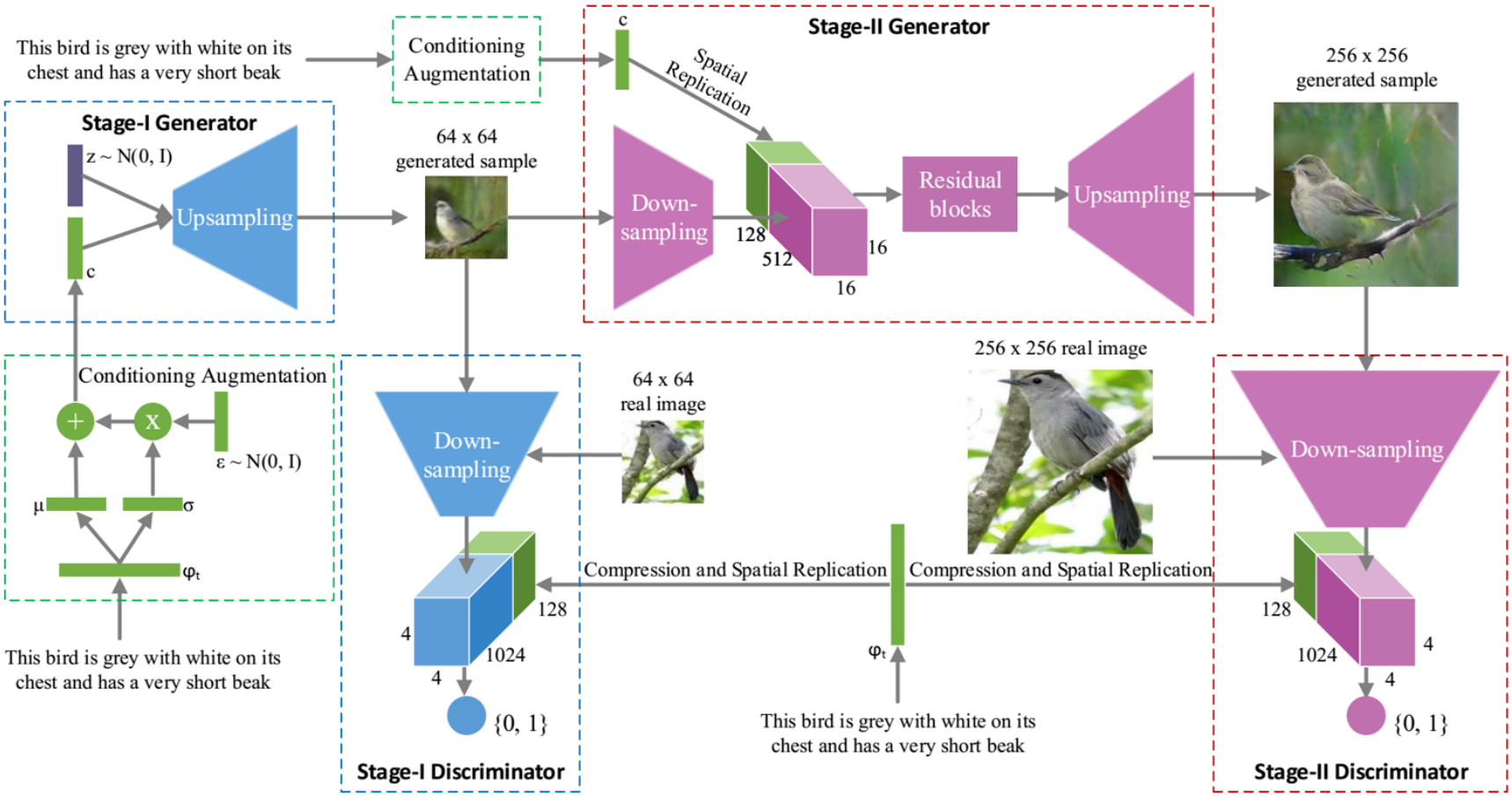
- Left: interpolating between two sentences (fixing noise vector)
- Right: interpolating between two randomly-sampled noise vectors

Issue...

The above text-to-image CGAN did not provide visual details and vidid object parts!

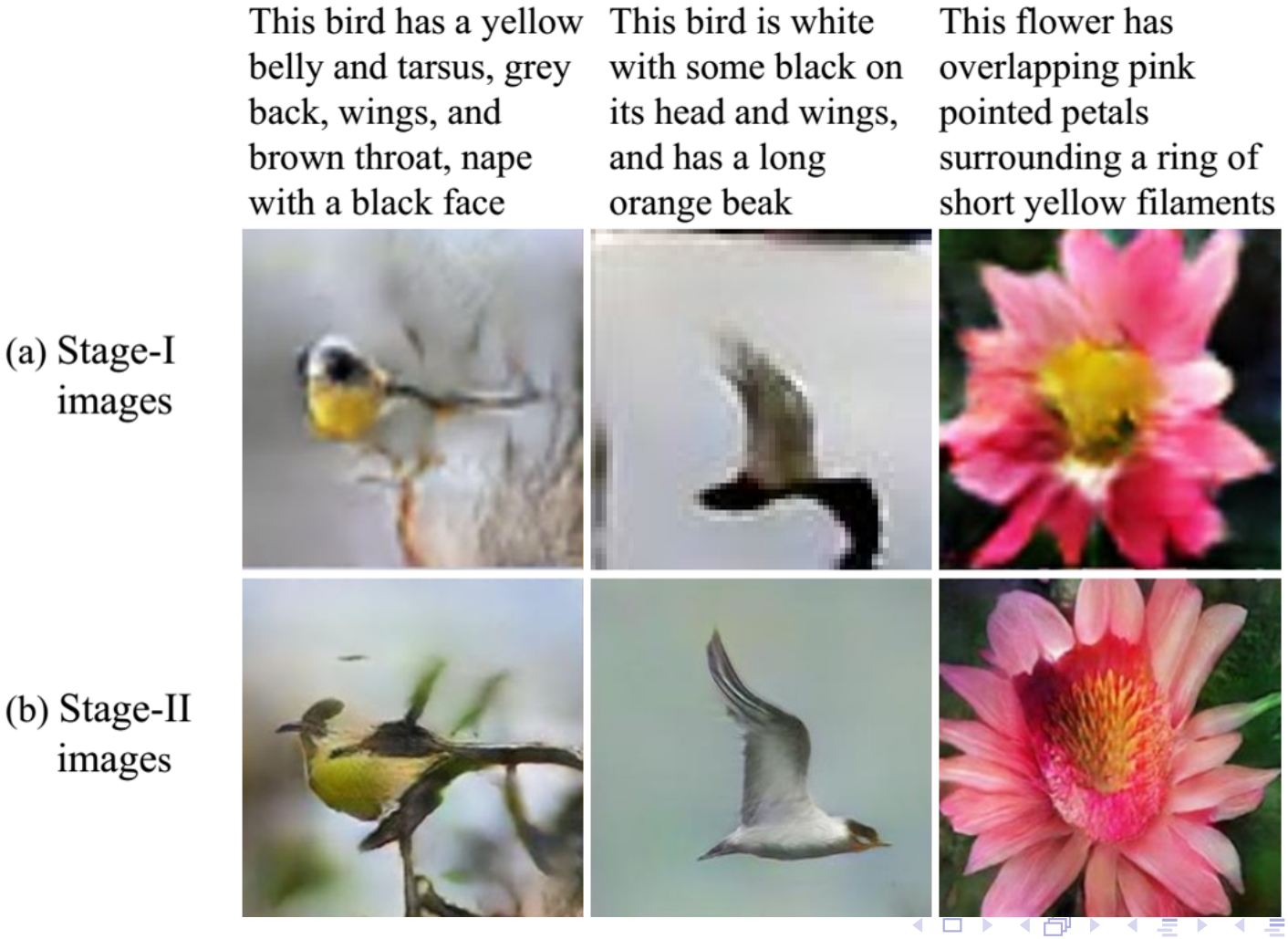
StackGAN: text to photo-realistic images

- StackGAN: stack of two CGANs, with 1st G's output (low-resolution image) as part of input to 2nd CGAN's G



StackGAN: stage effect

- Stage I: sketches rough shapes and basic colors
- Stage II: generate high-res images with photo-realistic details



StackGAN: result

- StackGAN generates high-res images with photo-realistic details (last row)

StackGAN: result (cont')

- StackGAN does not simply copy-paste existing images

StackGAN: result (cont')

- Images by interpolation between two texts are meaningful

This bird is completely red with black wings and pointy beak →
this small blue bird has a short pointy beak and brown on its wings

This bird is completely red with black wings and pointy beak →
The bird has a yellow breast with grey features and a small beak

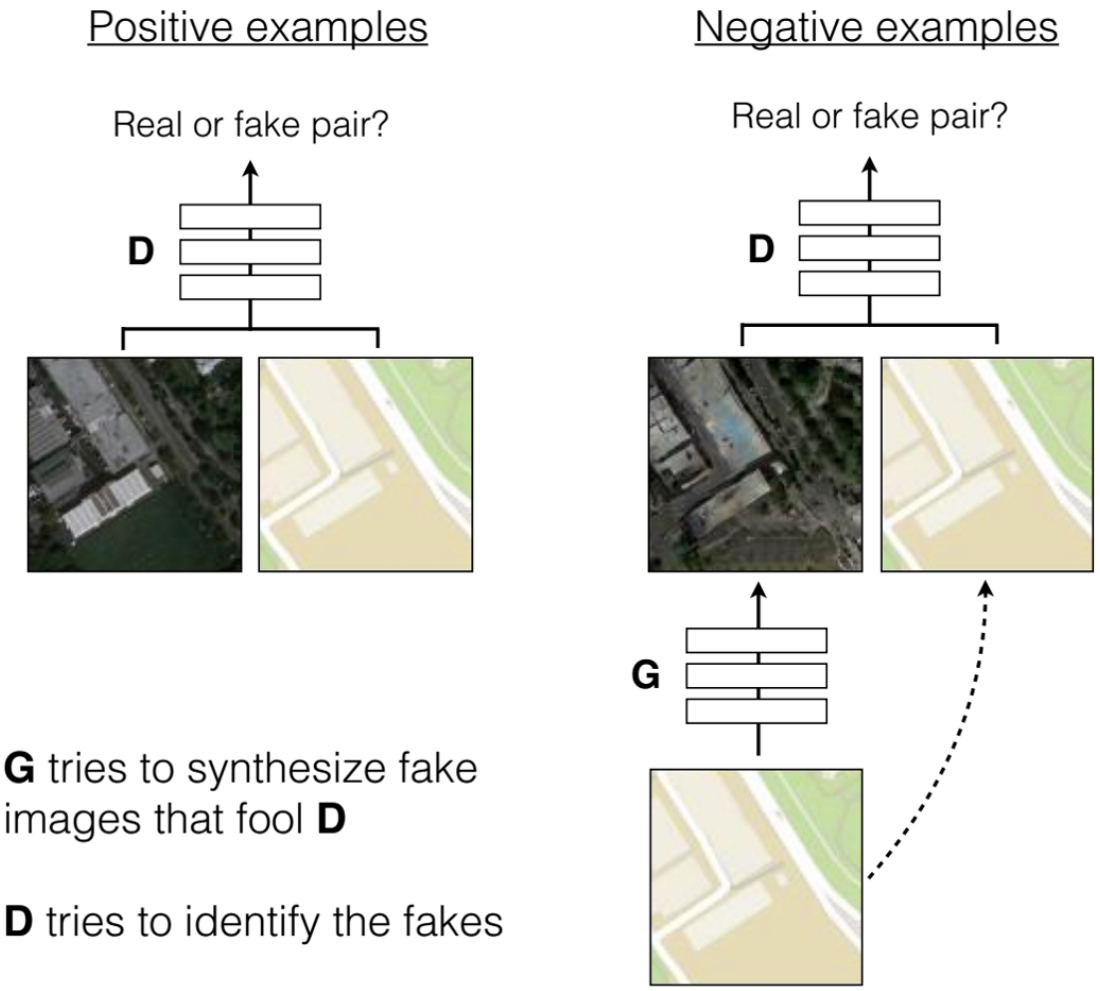
Figures here and in the previous 4 slides from Zhang, Xu, Li, Zhang, Huang, Wang, Metaxas, "StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks", arXiv, 2016

CGAN for more translation applications

CGANs not only for text-to-image translation ...

CGAN for image-image translation

- Image translation: translate one image to another type
- image pair (original & target domain) as input to D



CGAN for image-image translation

- Loss function:

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{x,y\sim p_{data}(x,y)}[\log D(x,y)] + \mathbb{E}_{x\sim p_{data}(x),z\sim p_z(z)}[\log(1-D(x,G(x,z))]$$

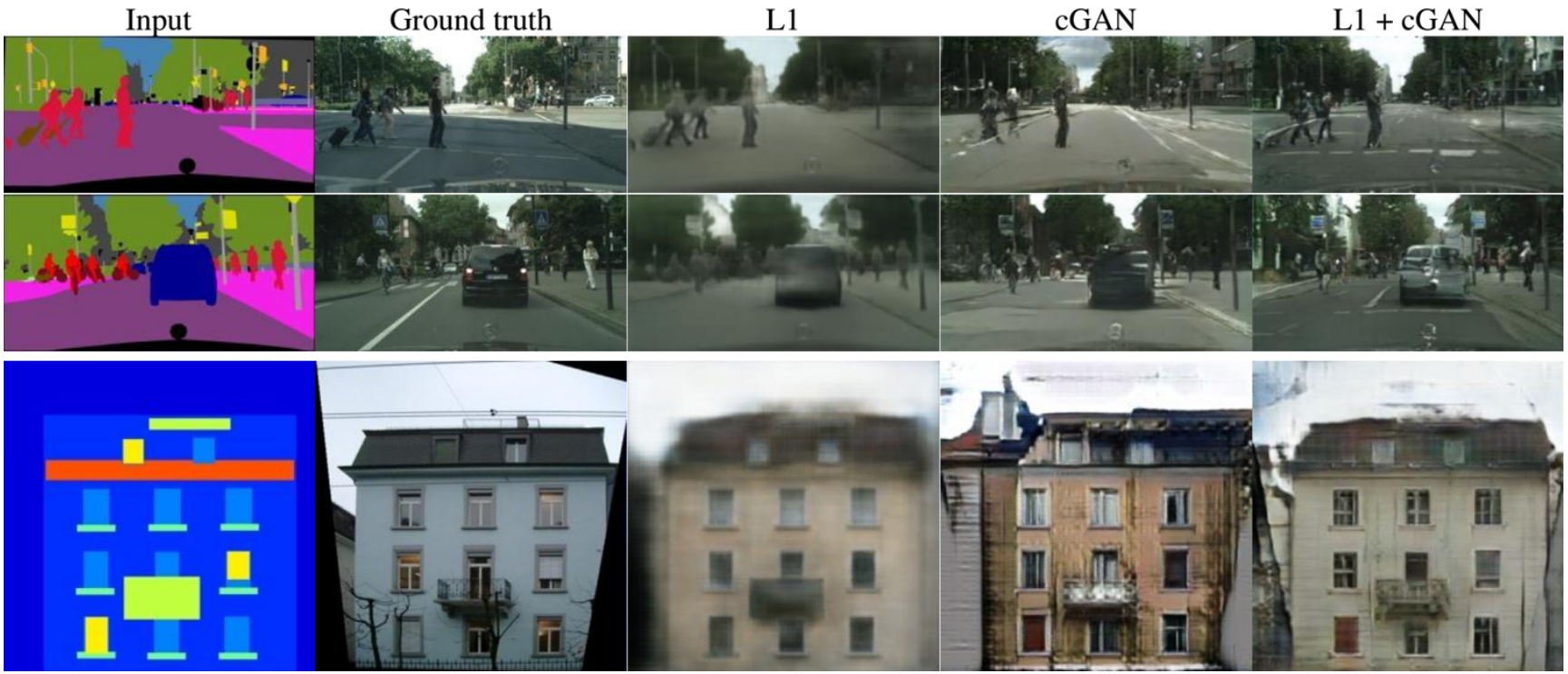
$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y\sim p_{data}(x,y),z\sim p_z(z)}[\|y - G(x,z)\|_1]$$

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda \mathcal{L}_{L1}(G)$$

- $\mathcal{L}_{cGAN}(G,D)$ forces G ouput realistic
- $\mathcal{L}_{L_1}(G)$ forces G ouput close to expected translated image
- Noise vector z almost not affect G output (dropout used here)

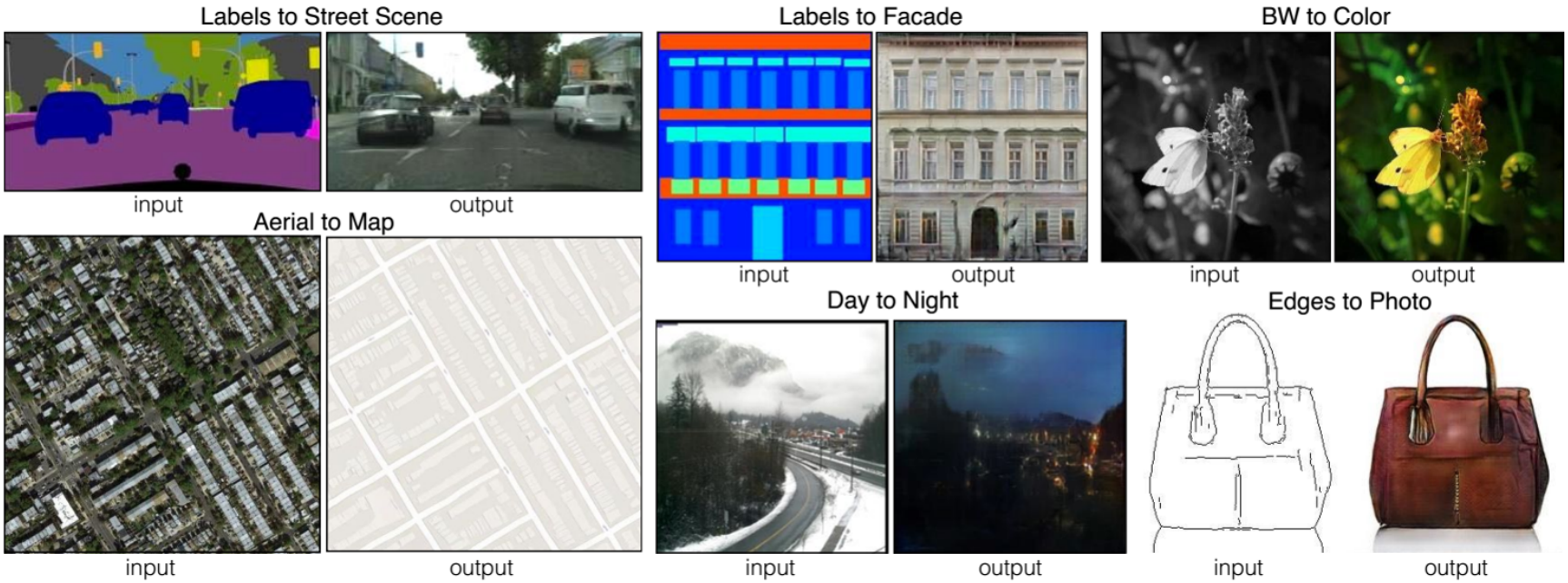
CGAN for image-image translation: loss term effect

- L_1 loss term alone leads to blurry results
- CGAN alone results in some artefacts



CGAN for image-image translation: result

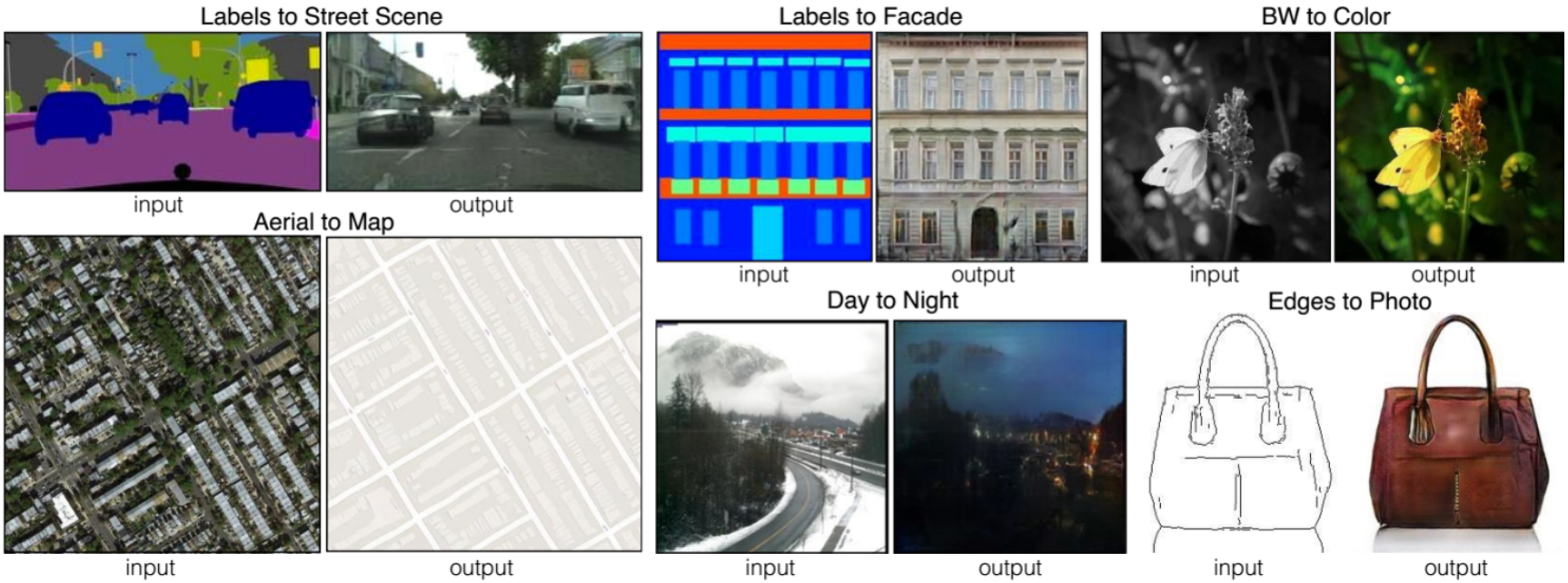
- Image translations between various domains!



Figures here and in the previous 3 slides from Isola, Zhu, Zhou, Efros, "Image-to-image translation with conditional adversarial networks", arXiv, 2016

CGAN for image-image translation: result

- Image translations between various domains!
- Note: need to train a unique CGAN for each translation app!



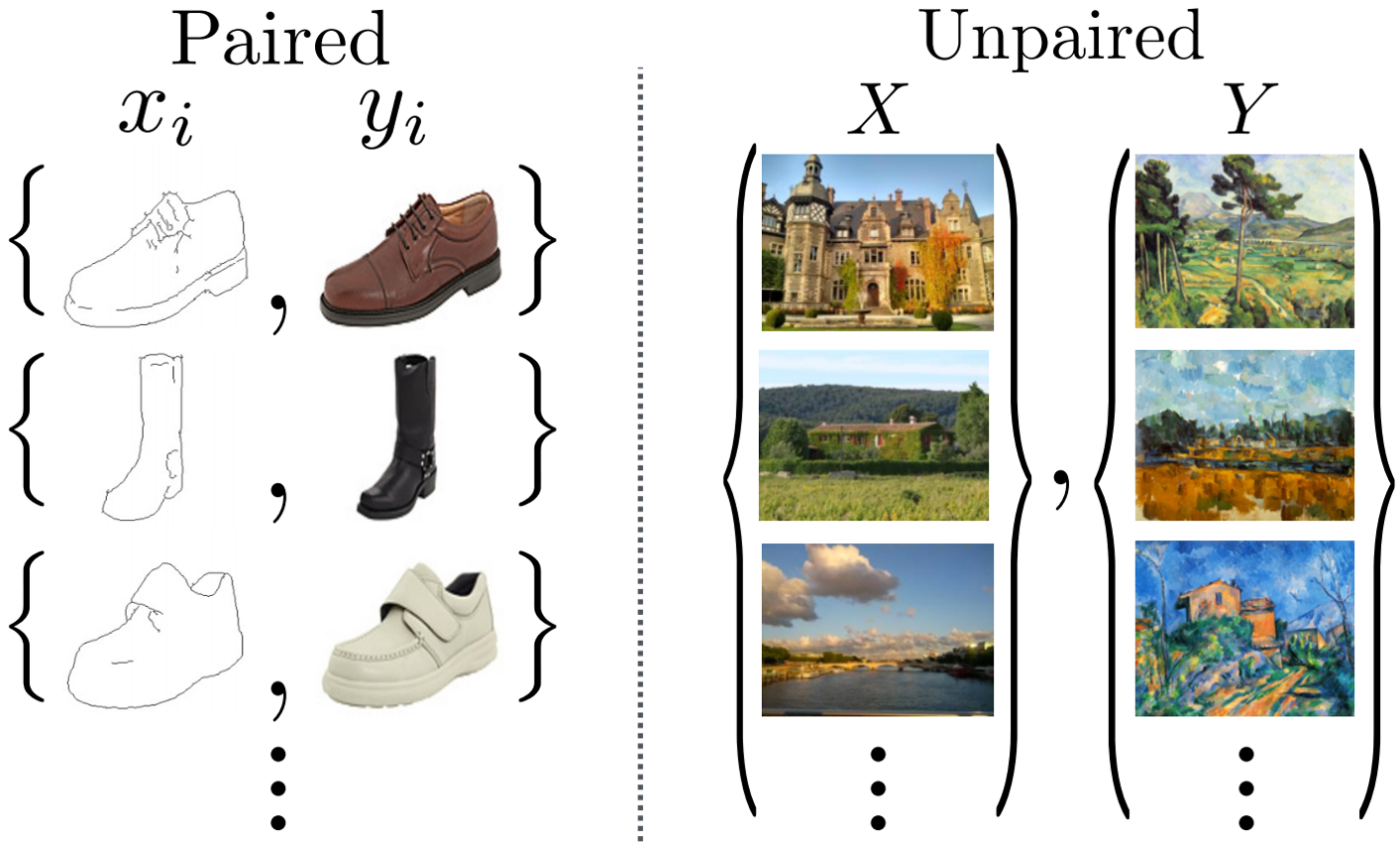
Figures here and in the previous 3 slides from Isola, Zhu, Zhou, Efros, "Image-to-image translation with conditional adversarial networks", arXiv, 2016

Issue of image translation

Difficult to collect enough paired images for training

Good question!

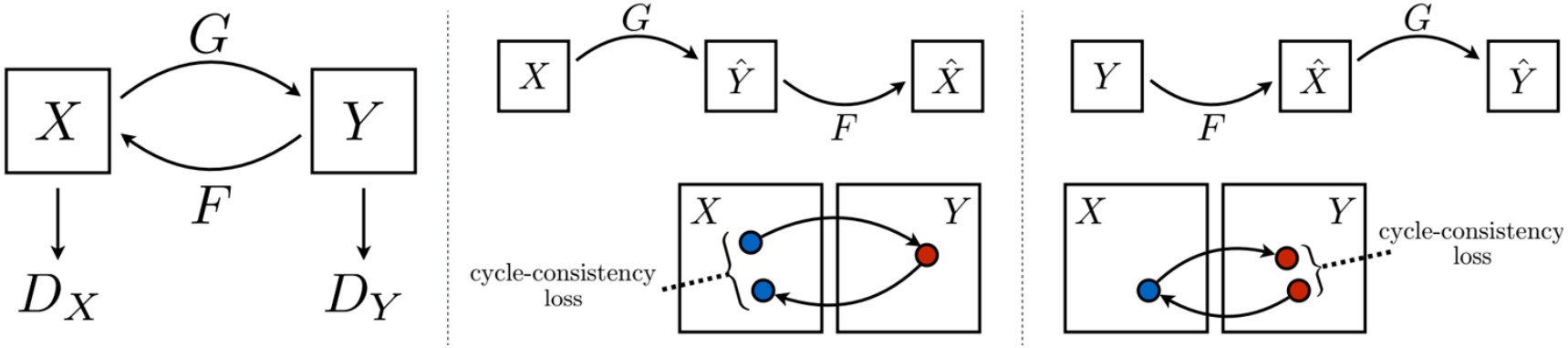
- Can we translate an image from one domain to another without paired images for training?



Figures here and in the next 3 slides from Zhu, Park, Isola, Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks", arXiv, 2017

CycleGAN: unpaired image-image translation

- Idea: translate one image from one domain to another, then translate the translated image back to original domain!

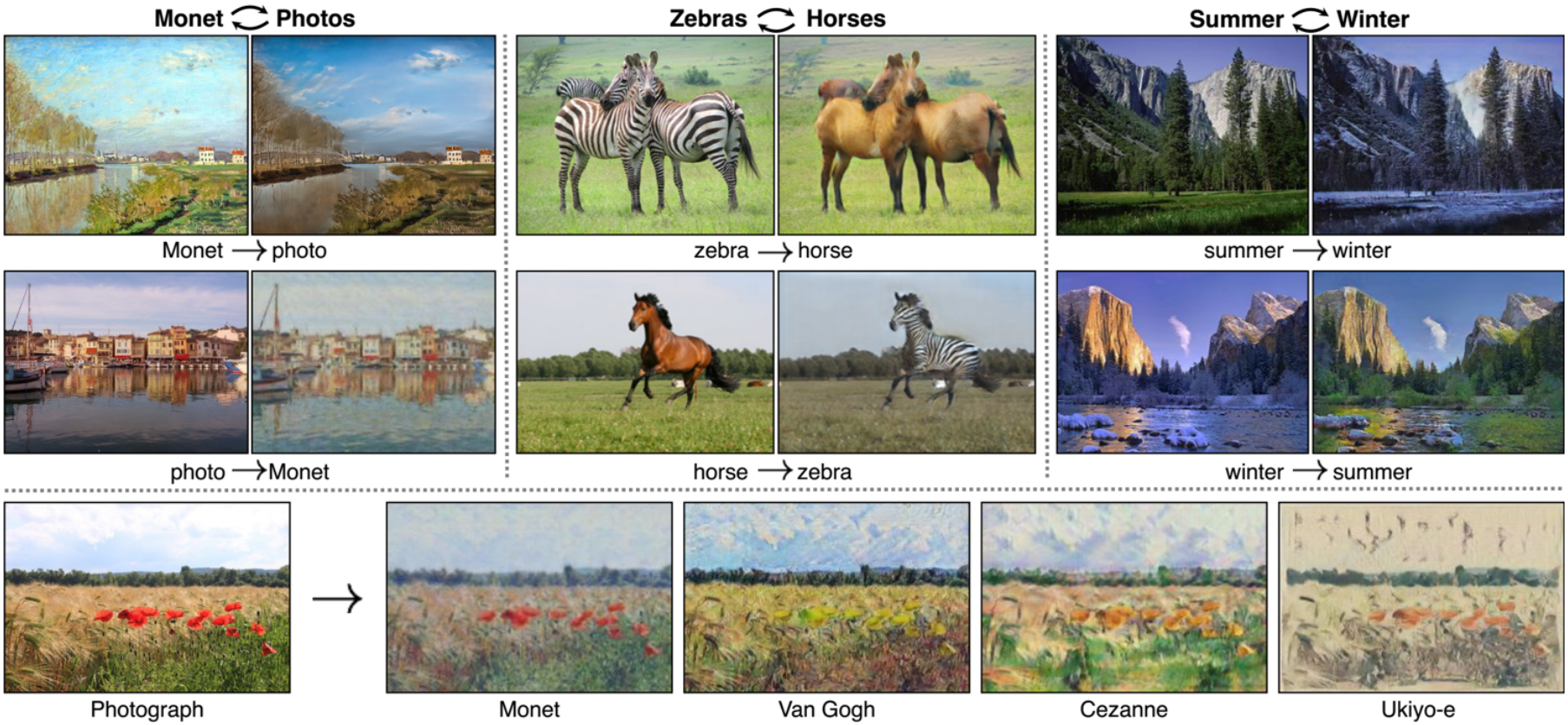


$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]$$

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F),$$

CycleGAN: unpaired image-image translation

- With **unpaired** training set, images can be translated from one domain to another!



CycleGAN: limitations

- Succeed in translating color and textures
- Often fail when requiring geometric changes (below)

Models almost same as CycleGAN

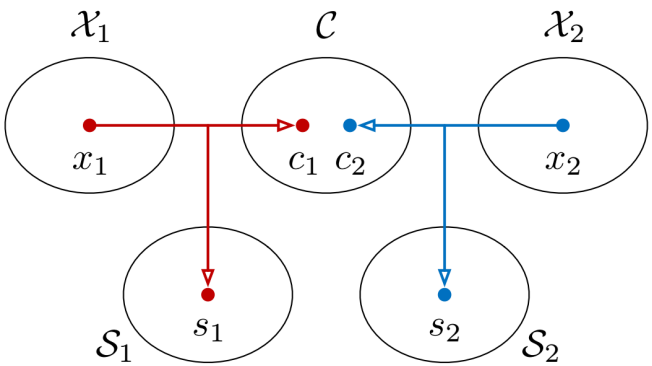
- DualGAN, DiscoGAN were proposed concurrently
- Same model structures as CycleGAN
- Minor differences in G and D, loss, training, etc.

So far ...

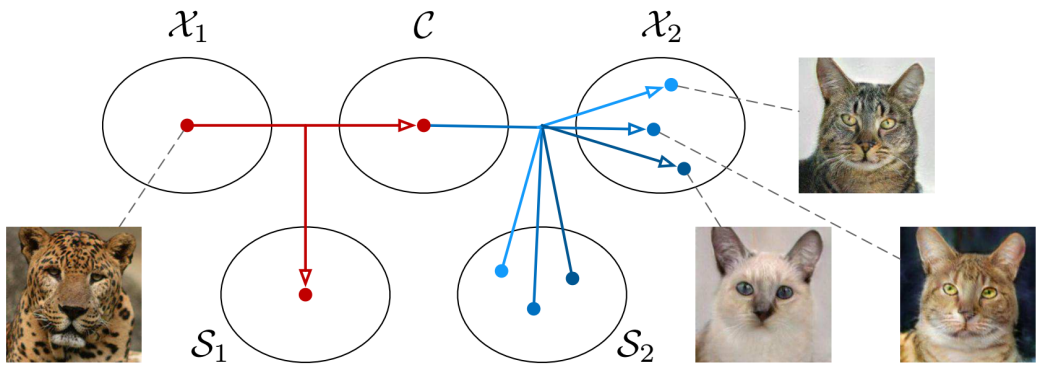
- Image translation is largely one-to-one mapping
- Ideally, translated to multiple results in another domain

MUNIT: multimodal unsupervised image-image translation

- Represent image separately by content code and style code;
- Different domains share content space, but not style space.
- Image translation: recombine content code of image from one domain with a random style code from the other domain.



(a) Auto-encoding

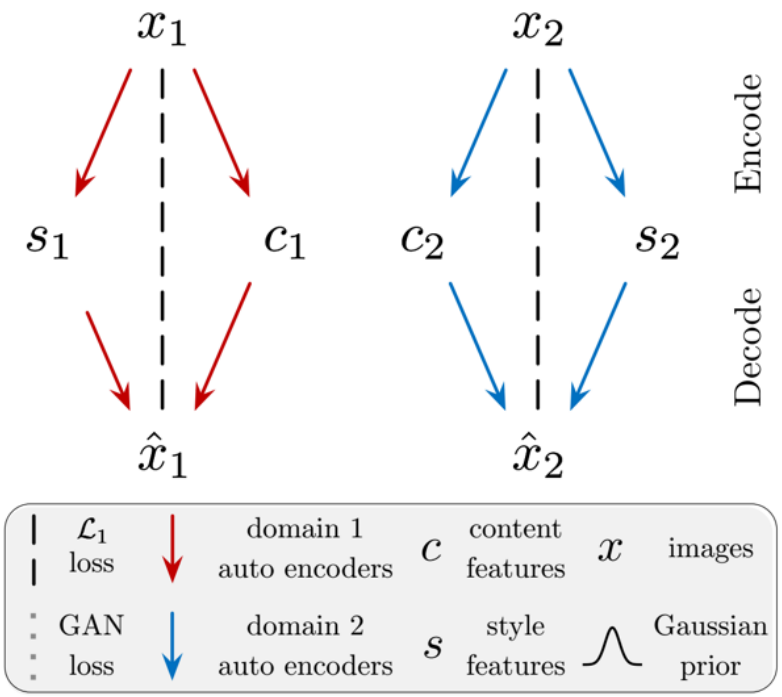


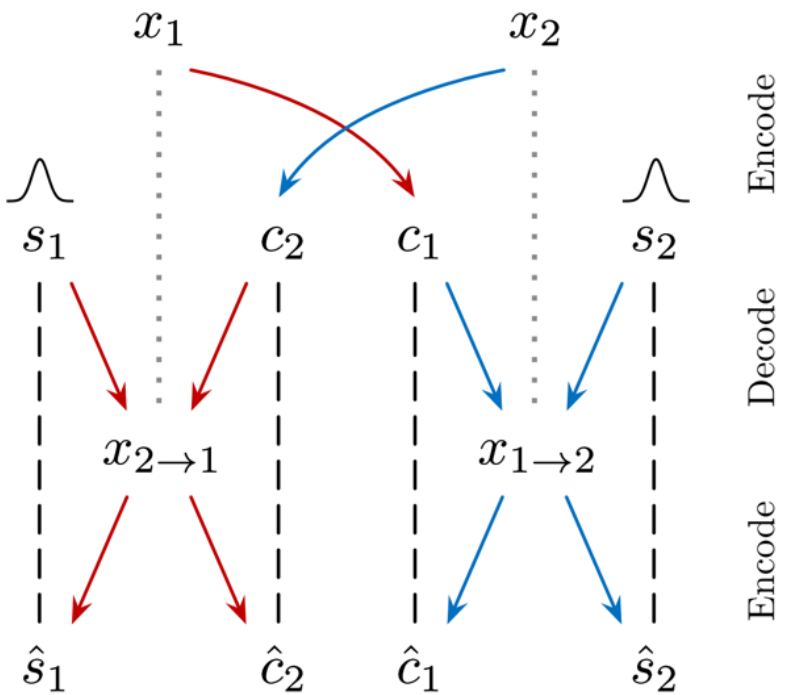
(b) Translation

Figures here and in the next 5 slides from Huang, Liu, Belongie, Kautz, "Multimodal unsupervised image-to-image translation", ECCV, 2018

MUNIT (cont')

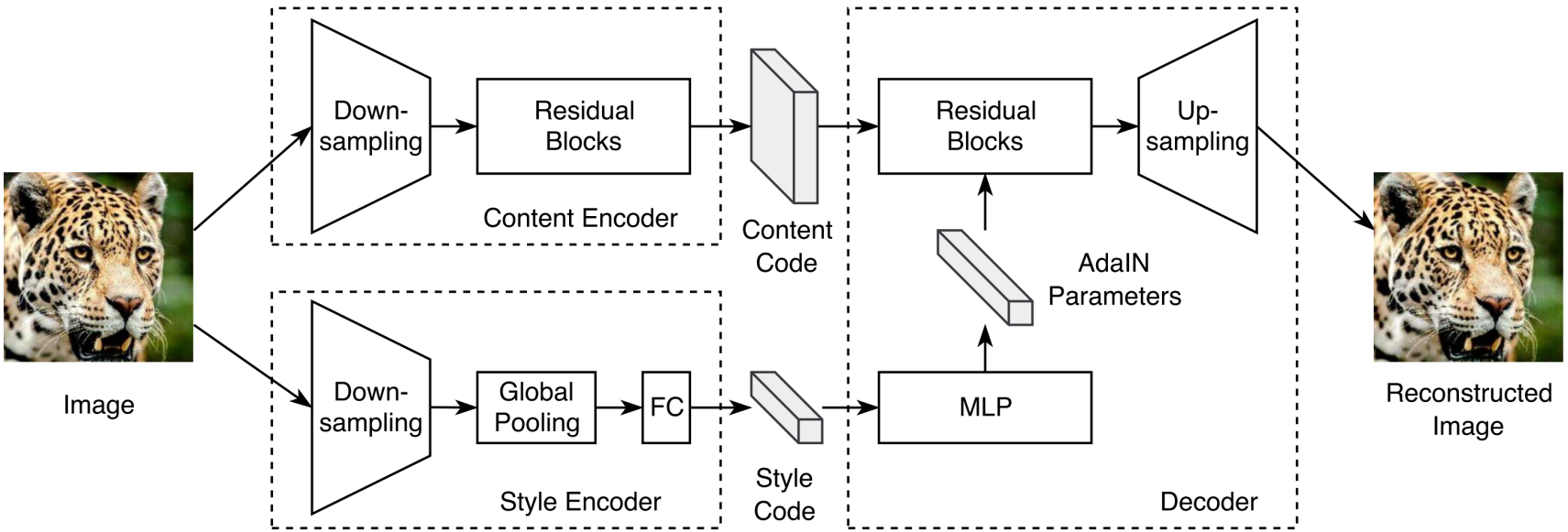
- Two autoencoders (red & blue), one for each domain; latent code of each AE consists of content code c and style code s.
- Two discriminators (not shown).
- 2 GAN objectives (dotted), 6 bidirectional reconstruction losses (dashed) that reconstruct both images and latent codes.





MUNIT (cont')

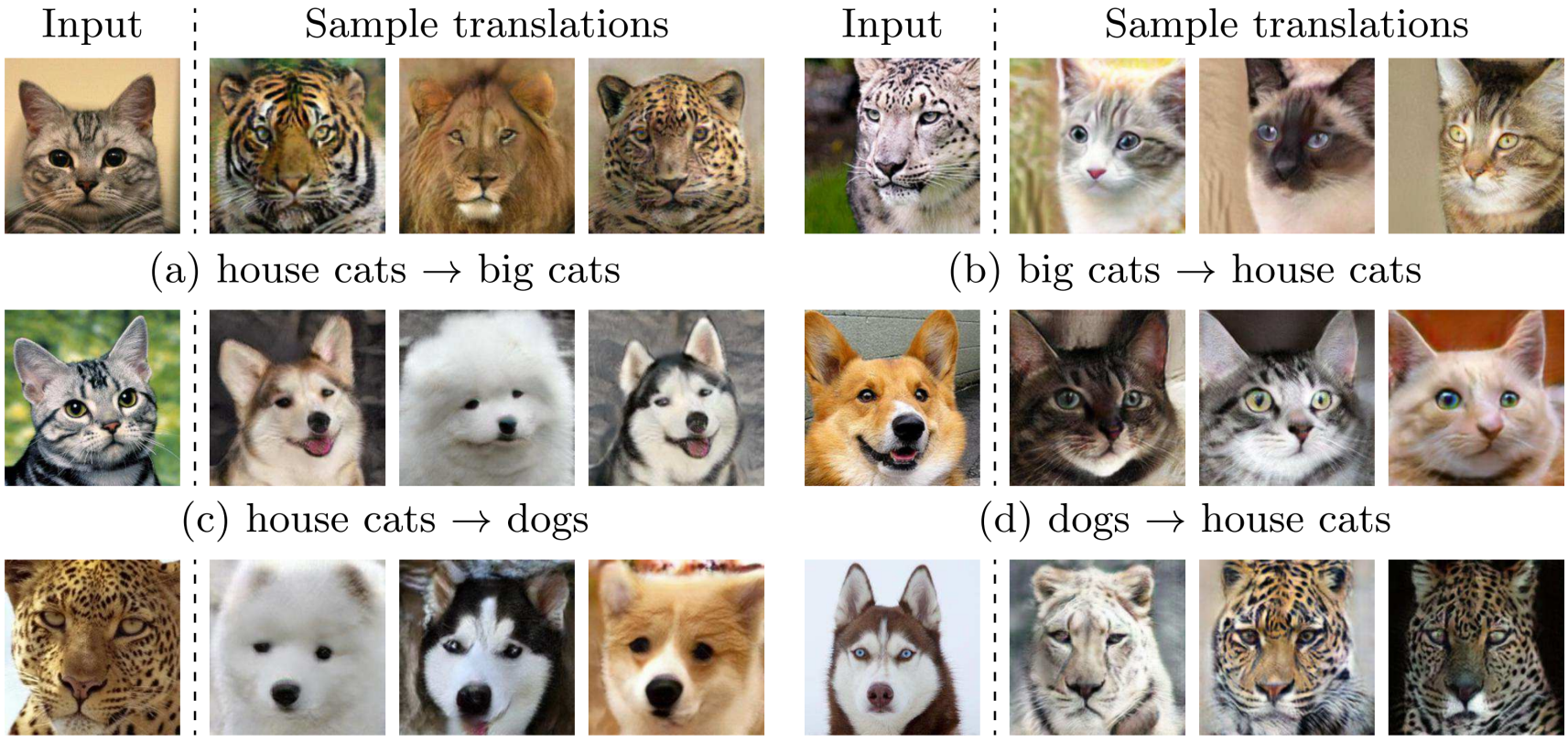
- One autoencoder's architecture
- Decoder is (also) Generator!



Note: AdaIN - adaptive instance normalization

MUNIT: results

- Animal image translation: one to many translations



(a) house cats → big cats

(b) big cats → house cats

(c) house cats → dogs

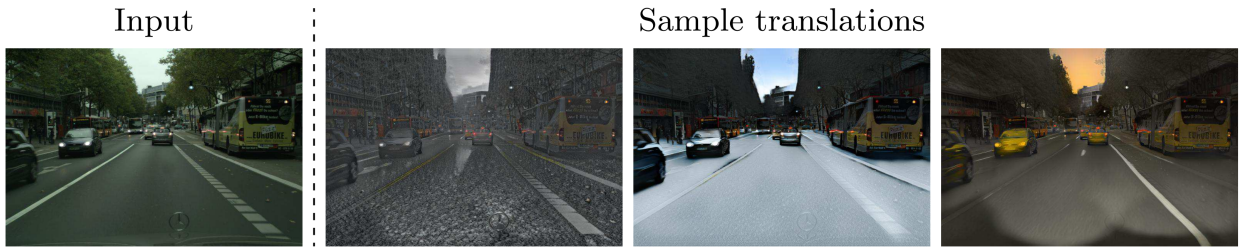
(d) dogs → house cats

(e) big cats → dogs

(f) dogs → big cats

MUNIT: results

- Example results on street scene translation



(a) Cityscape → SYNTHIA

(b) SYNTHIA → Cityscape

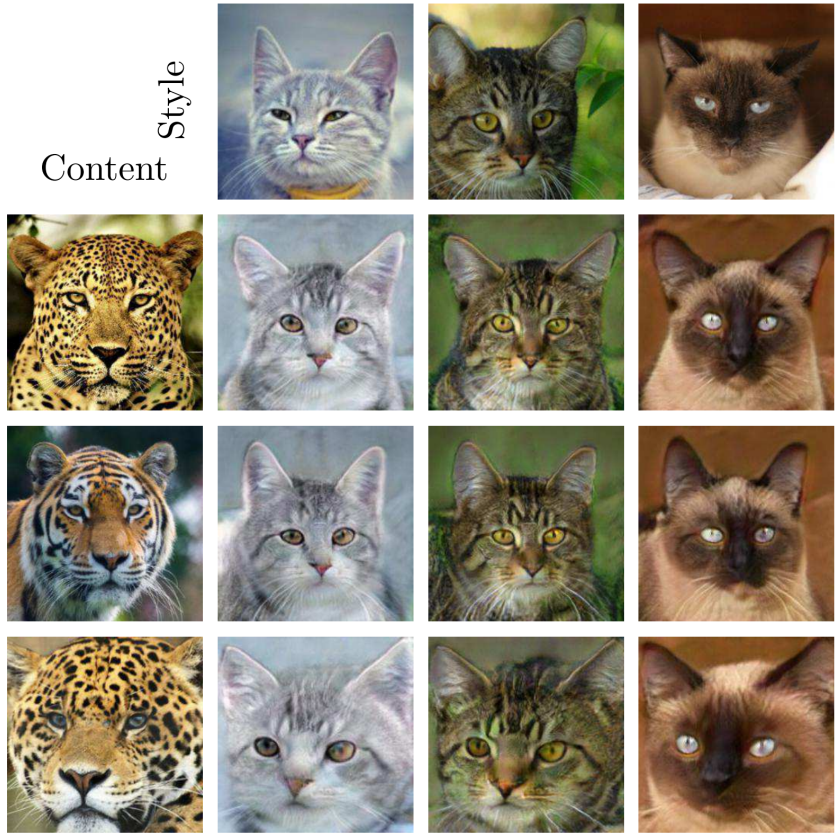
(c) summer → winter

(d) winter → summer

MUNIT: results

- Example-guided image translation

(a) edges → shoes



(b) big cats → house cats

CAN: creative generative networks for arts

- We have seen GANs can transfer art styles!
- We may be satisfied with generated art images

CAN: creative generative networks for arts

- We have seen GANs can transfer art styles!
- We may be satisfied with generated art images
- But artists are NOT! They want to create new arts rather than emulate old arts!

CAN: creative generative networks for arts

- We have seen GANs can transfer art styles!
- We may be satisfied with generated art images
- But artists are NOT! They want to create new arts rather than emulate old arts!
- So far, GANs just emulate old arts!
- Question: how to create new arts!

CAN: creative generative networks for arts

- We have seen GANs can transfer art styles!
- We may be satisfied with generated art images
- But artists are NOT! They want to create new arts rather than emulate old arts!
- So far, GANs just emulate old arts!
- Question: how to create new arts!
- Idea: generator tries to fool D to think it is 'art', and meanwhile tries to confuse D about the style of generated art.

Figure in the next 2 slides from Elgammal, Liu, Elhoseiny, Mazzone, "CAN: creative adversarial networks generating art by learning about styles and deviating from style norms", arXiv, 2017

CAN: creative generative networks for arts

- Training dataset: real art images of K styles

Style classification loss: for a real art image x, try to make Discriminator able to recognize its style type

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

$$\mathbb{E}_{x,\hat{c}\sim p_{data}}[\log D_r(x) + \log D_c(c=\hat{c}|x)] +$$

$$\mathbb{E}_{z\sim p_z}[\log(1-D_r(G(z))) - \sum_{k=1}^{K}(\frac{1}{K}log(D_c(c_k|G(z))+$$

$$(1-\frac{1}{K})log(1-D_c(c_k|G(z)))]$$

For generated art image G(z), try to make D think it's an art image

Style ambiguity loss: for a generated art image G(z), try to make D unable to recognize its style

CAN: creative generative networks for arts

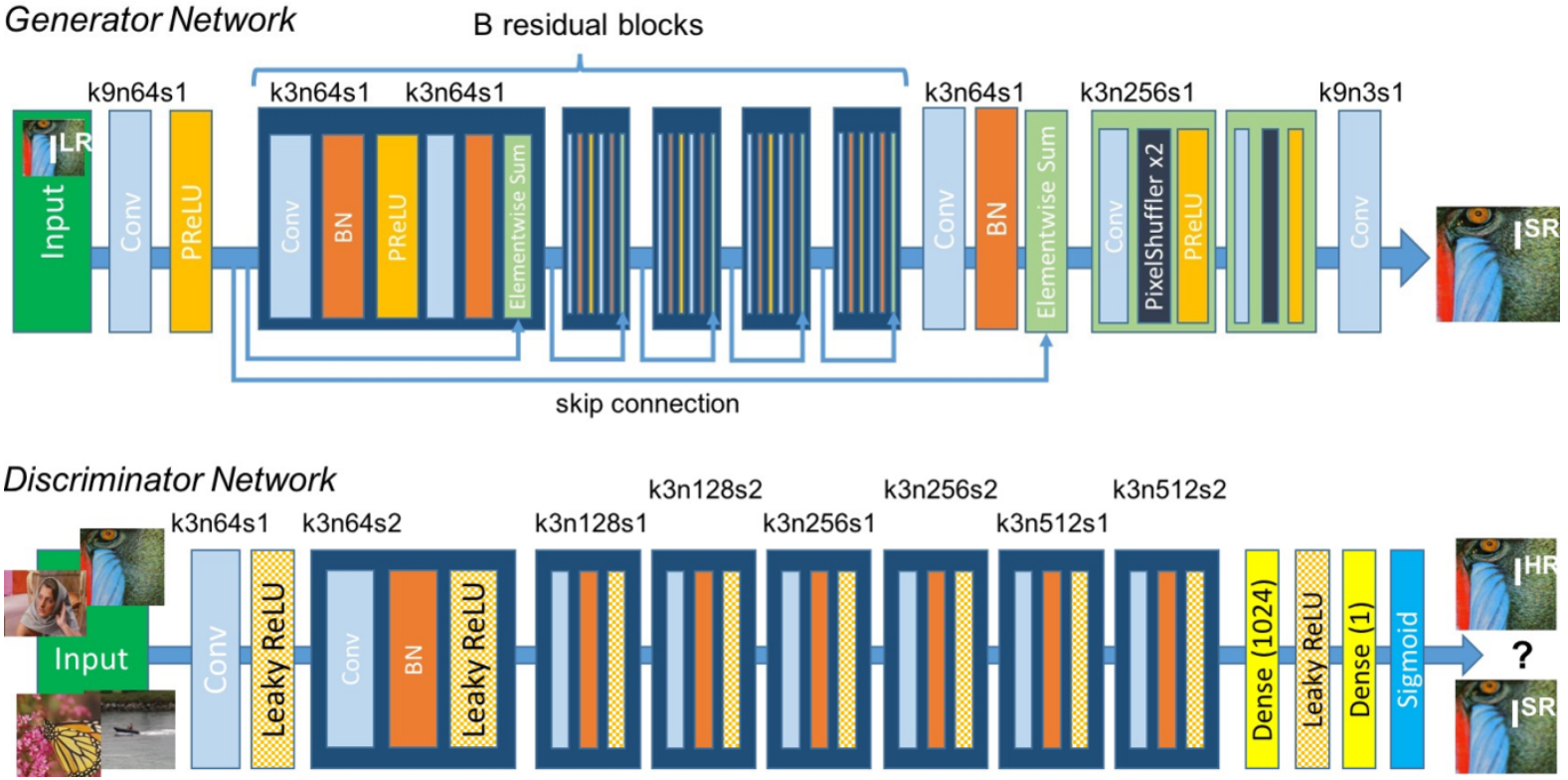
- People confused generated arts (below) with human arts

More applications

GAN applications are more than expected...

SRGAN: GAN for image super-resolution

- G input: low-resolution image; output: super-res image
- G has a FCN-like structure with residual blocks



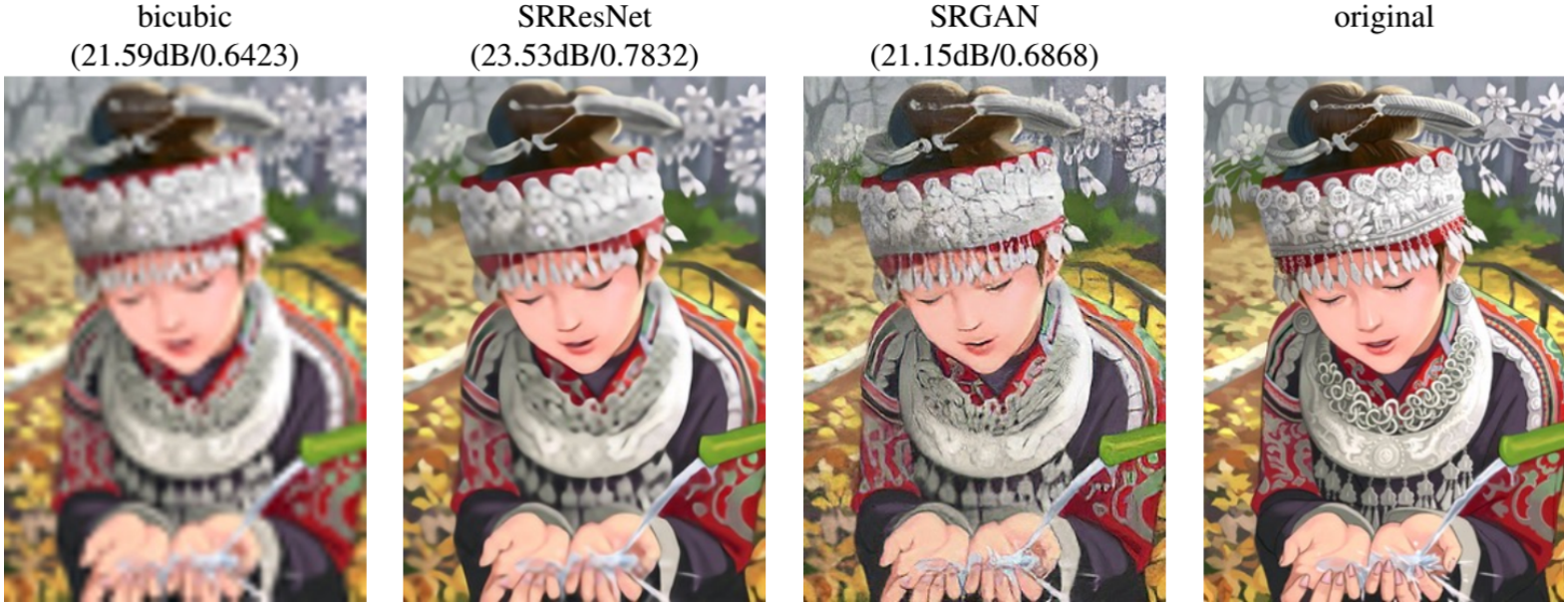
Figures here and in the next 2 slides from Ledig, Theis, Huszar, Caballero, Cunningham, Acosta, Aitken, Tejani, Totz, Wang, Shi, "Photo-realistic single image super-resolution using a generative adversarial network", arXiv, 2016

SRGAN: loss

$$G^* = \arg\min_G \max_D \mathcal{L}_{GAN}(G, D) + \lambda \mathcal{L}_{content}(G)$$

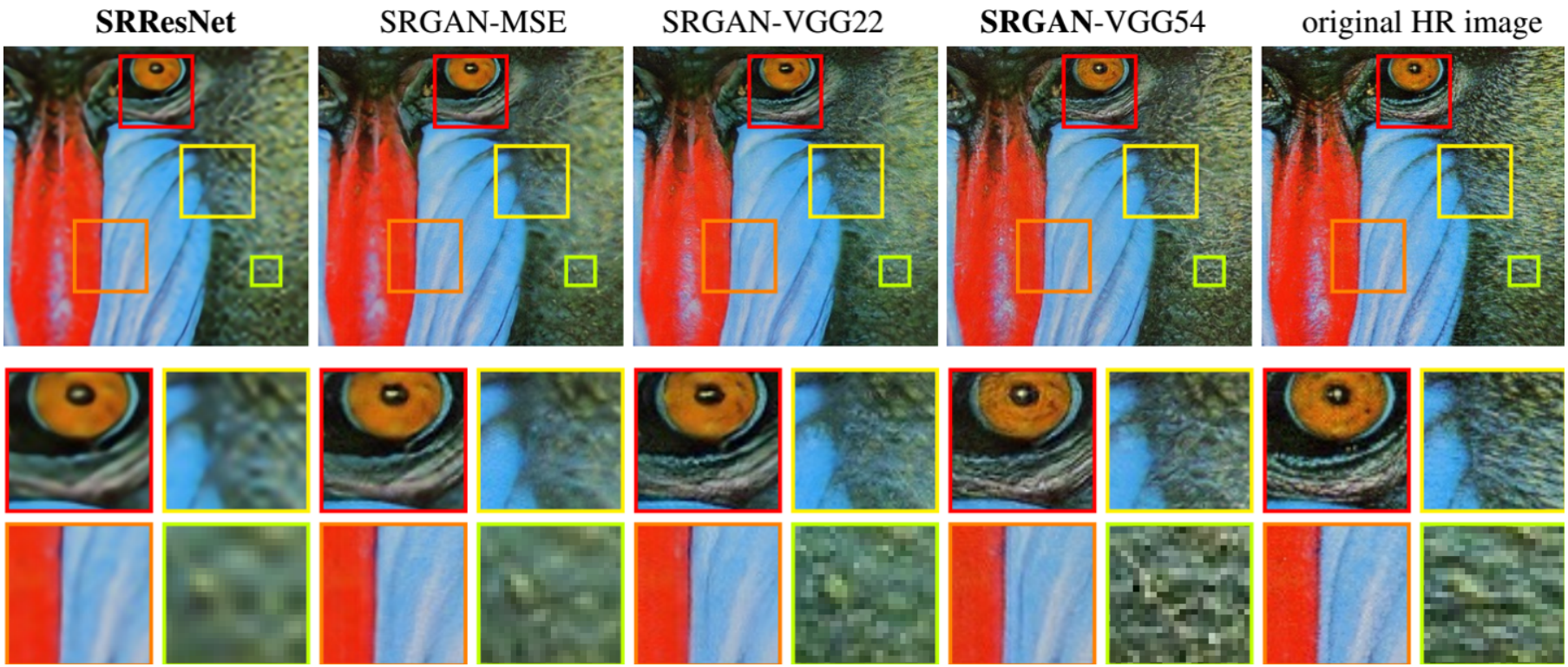
$$\mathcal{L}_{content}(G) = \mathbb{E}_{x,y \sim p_{data}(x,y)} \|f(y) - f(G(x))\|^2$$

- $f(\cdot)$ is feature map from certain higher layer of VGG
- x is a low-resolution version of y



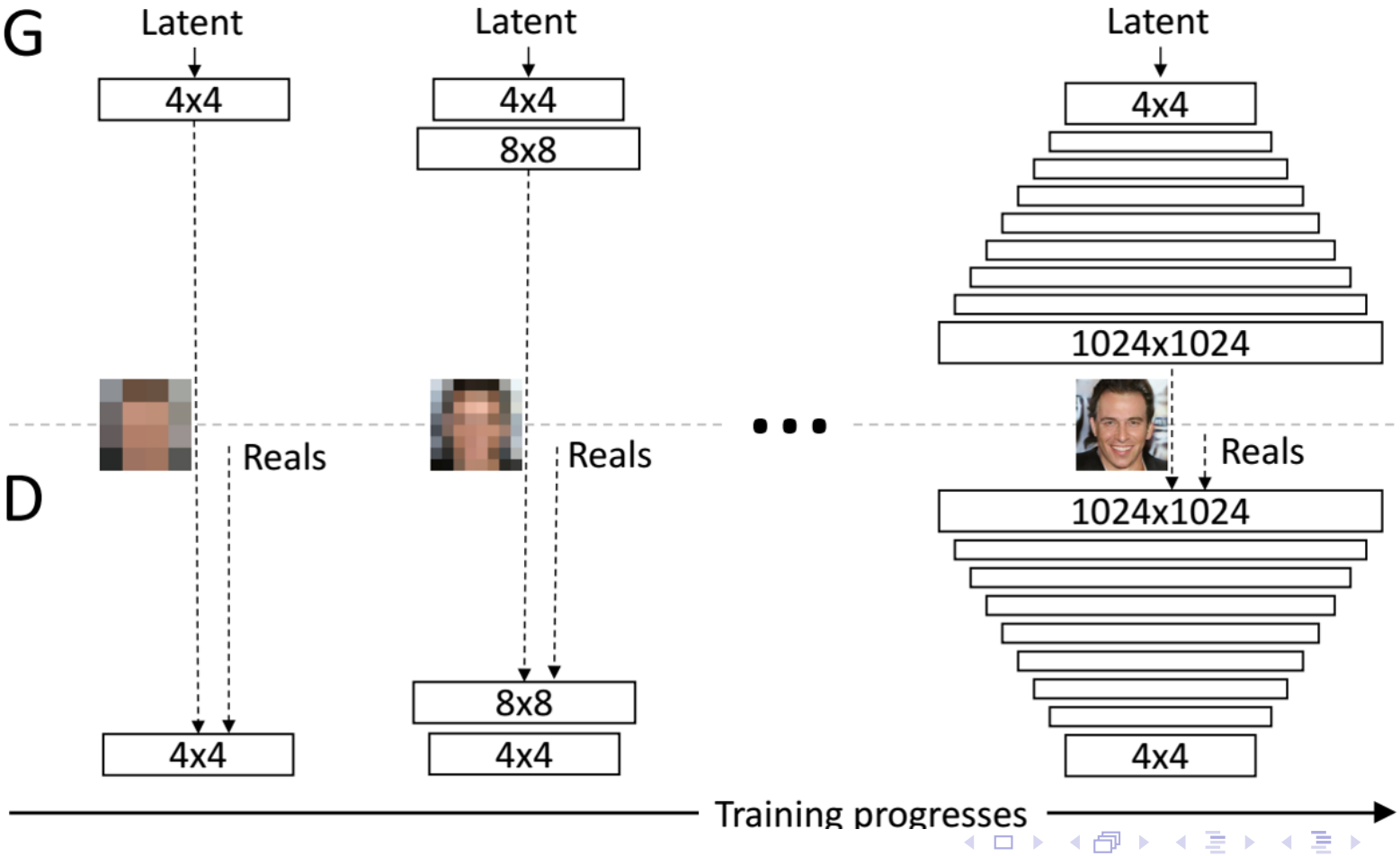
SRGAN: result

- SRGAN with content loss yields better texture details



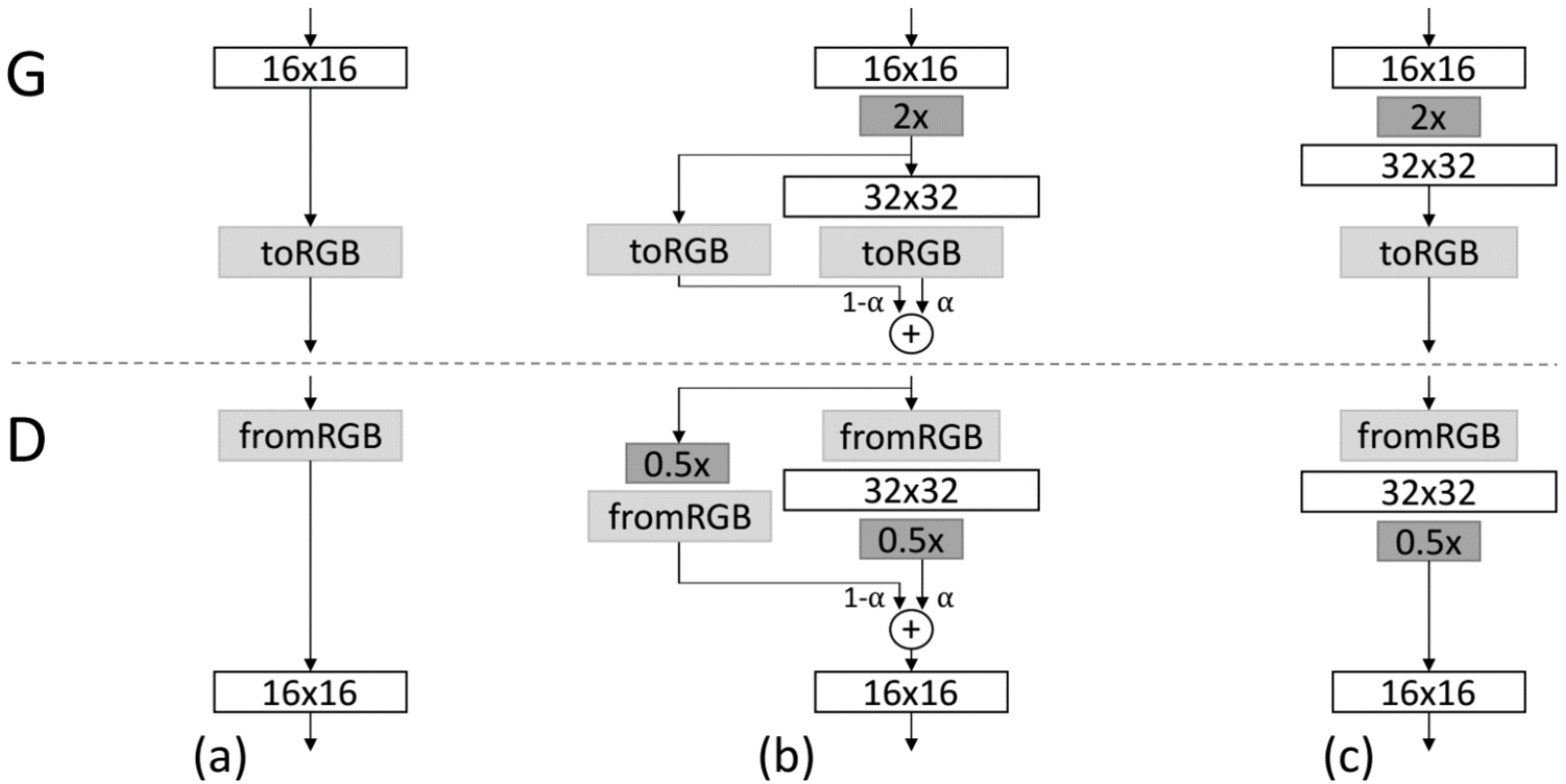
Progressive GAN for higher-resolution images

- Repeat: train a low-resolution GAN, then add one more layer
- Learn to discover large-scale structures, then shift to increasingly finer details



Progressive GAN: training

- During training (Fig. b), α increases gradually from 0 to 1.
- '$2\times$', '$0.5\times$': doubling or halving resolution
- 'toRGB': from feature map to 3-channel with 1×1 kernels



Figures here, in prev slide, and in next 2 slides from Karras, Aila, Laine, Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation", arXiv, 2017

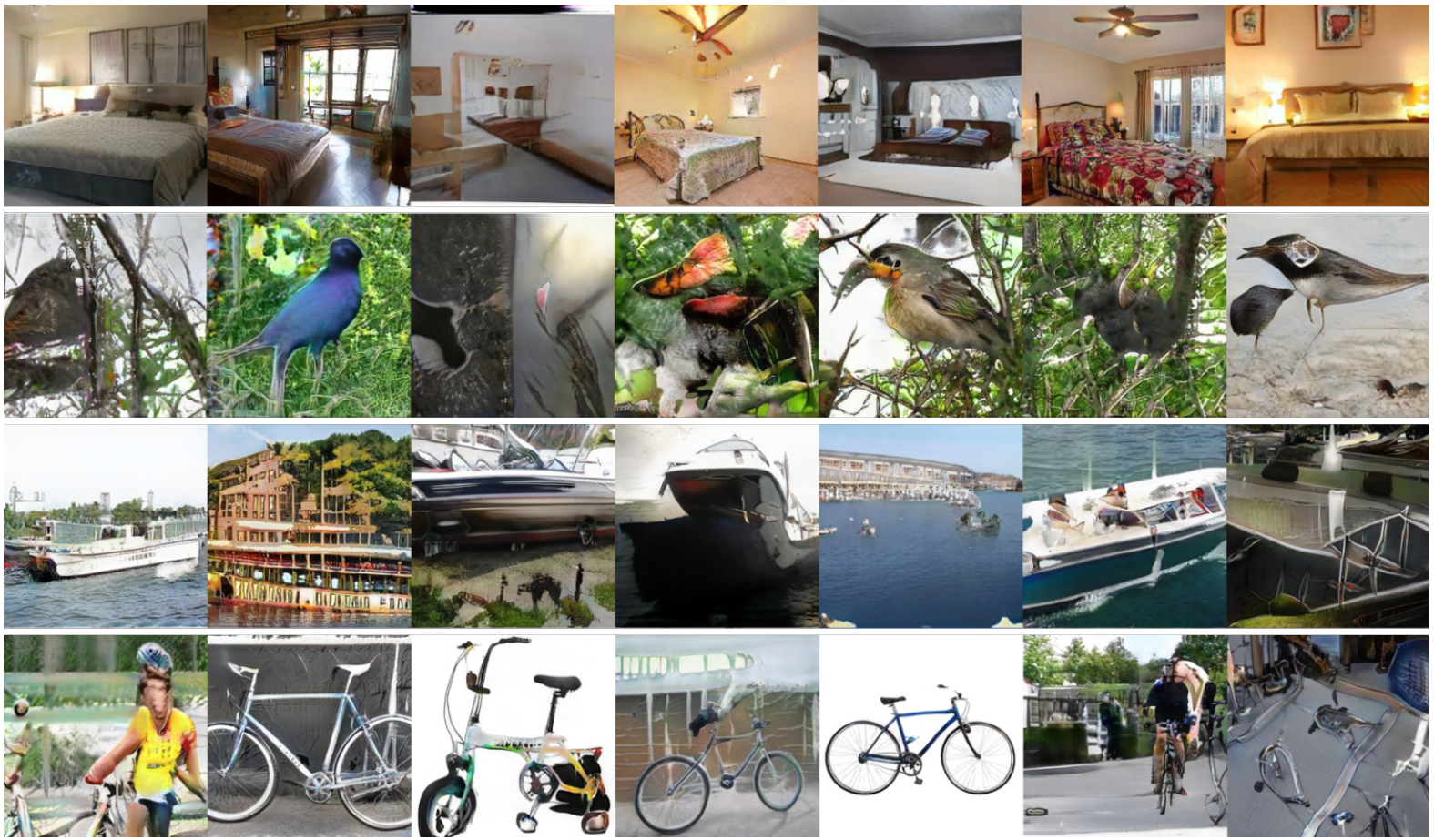
Progressive GAN: result

- Progressive GAN show more details and variations (1024×1024 pixels)

(a) Progressive GAN (b) Baseline method

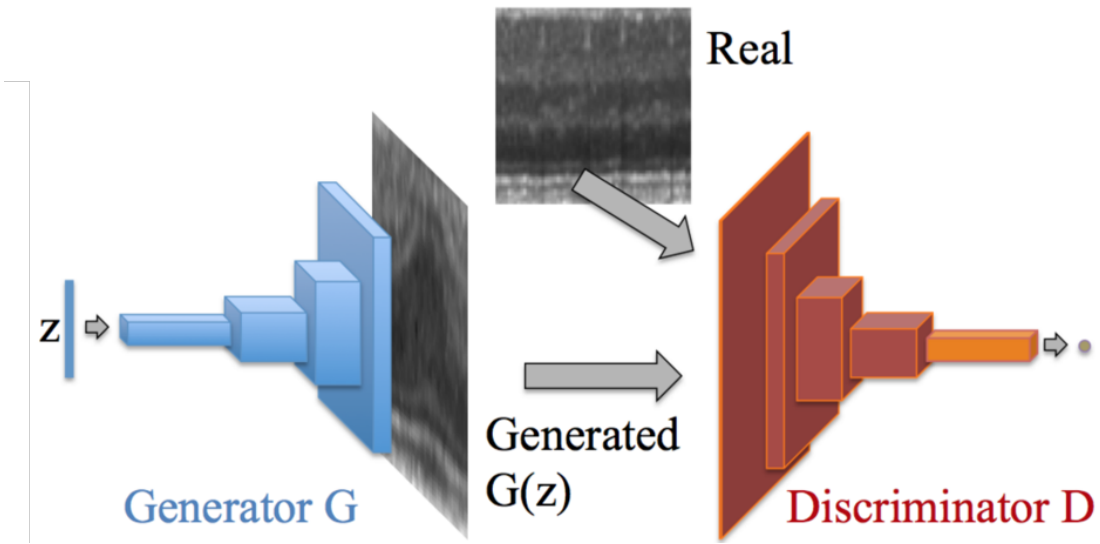
Progressive GAN: result (cont')

- Generated images contain more details and variations, but might not always be semantically correct (256×256 pixels)



One more application: GAN for biomarker discovery

- Use healthy retinal images to train a GAN
- Then use the GAN to detect abnormality in new images
- Idea: if a real image is healthy, there should exist a very similar synthetic image generated by the Generator G



Figures here and below from Schlegl, Seebock, Waldstein, Schmidt-Erfurth, Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery", IPMI, 2017

GAN for biomarker discovery

- Given a new image $\mathbf{x}$, how to find abnormal regions inside?

GAN for biomarker discovery

- Given a new image $\mathbf{x}$, how to find abnormal regions inside?
- Answer: find best input $\mathbf{z}_\gamma$ such that $G(\mathbf{z}_\gamma)$ is as similar to $\mathbf{x}$ as possible.

GAN for biomarker discovery

- Given a new image $\mathbf{x}$, how to find abnormal regions inside?
- Answer: find best input $\mathbf{z}_\gamma$ such that $G(\mathbf{z}_\gamma)$ is as similar to $\mathbf{x}$ as possible.

$$\mathcal{L}(\mathbf{z}_\gamma) = (1-\lambda)\cdot\mathcal{L}_R(\mathbf{z}_\gamma) + \lambda\cdot\mathcal{L}_D(\mathbf{z}_\gamma)$$

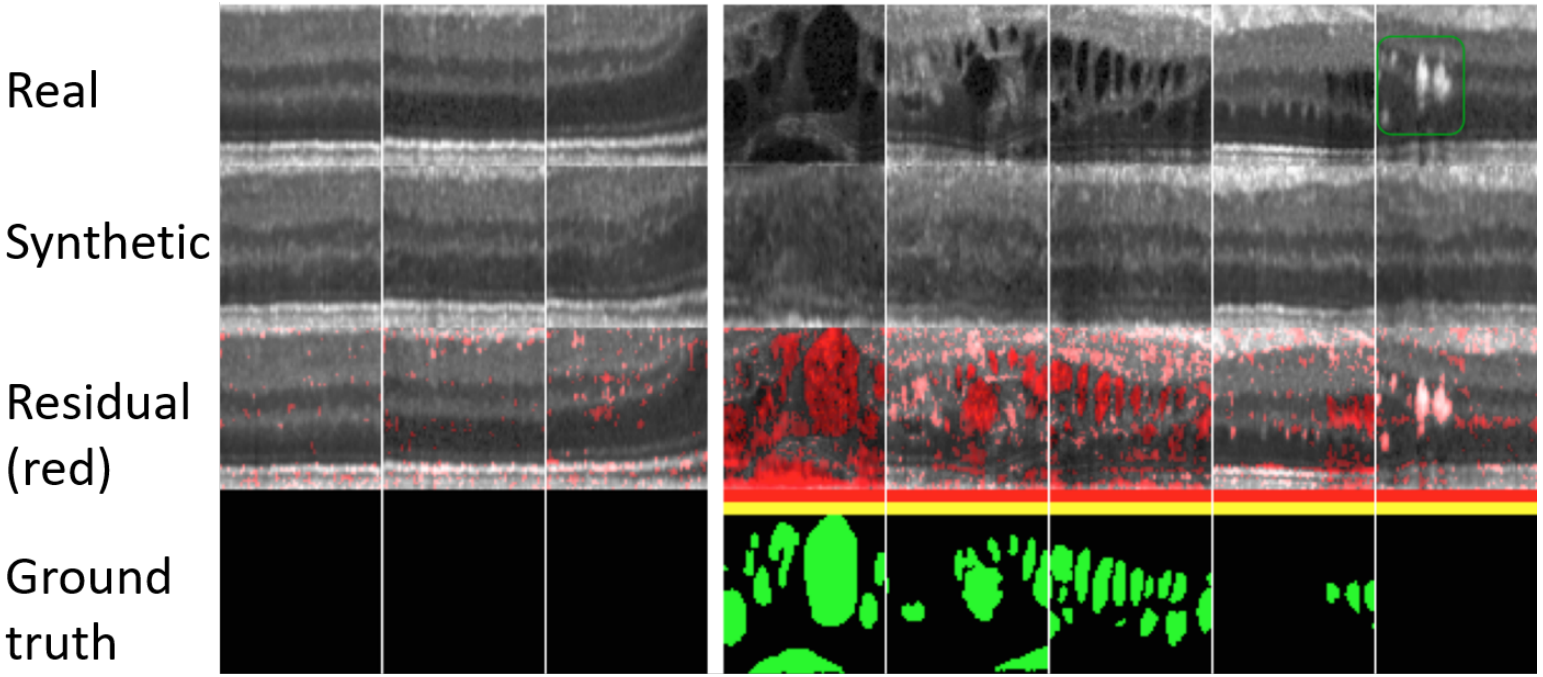
$$\mathcal{L}_R(\mathbf{z}_\gamma) = \sum|\mathbf{x} - G(\mathbf{z}_\gamma)|$$

$$\mathcal{L}_D(\mathbf{z}_\gamma) = \sum|\mathbf{f}(\mathbf{x}) - \mathbf{f}(G(\mathbf{z}_\gamma))|$$

- $\min L_R(\mathbf{z}_\gamma)$ makes $G(\mathbf{z}_\gamma)$ similar to $\mathbf{x}$ in image space
- $\min L_D(\mathbf{z}_\gamma)$ makes $G(\mathbf{z}_\gamma)$ similar to $\mathbf{x}$ in feature space, where $\mathbf{f}(\cdot)$ is a higher layer ouput of D

GAN for biomarker discovery

- Minor difference between a healthy image and corresponding synthesized image
- Obvious differences (red) somewhere between lesion and synthesized images
- Lesion (abnormal) regions automatically detected!



Summary

- Conditional GAN opens the door to huge applications!
- Applications in image, text, and medical domains, etc.
- New applications appear every week!
- GAN Zoo! https://github.com/hindupuravinash/the-gan-zoo

Further reading:

- Yu, Zhang, Wang, Yu, 'SeqGAN: sequence generative adversarial nets with policy gradient', AAAI, 2017
- Chen, Duan, Houthooft, Schulman, Sutskever, Abbeel, 'InfoGAN: interpretable representation learning by information maximizing generative adversarial nets', NIPS, 2016
- ZHu, Zhang, Pathak, Darrell, Efros, Wang, Shechtman, 'Toward multimodal image-to-image translation', NIPS, 2017.