Generative Models II

Jisang Han

onground@korea.ac.kr

KUGODS

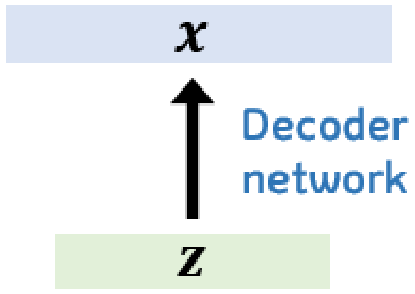
Department of Computer Science and Engineering, Korea University

Korea University GOogle Developer Students
KUGODS

Mathematical meaning

- If a model parameter θ is given, the higher $p_{\theta_*}(x)$ (The probability that the answer we want is x) the better model.
- Train parameters to maximize $p_{\theta_*}(x)$

Decoder



z latent variable의 확률분포

$$p_\theta(z)$$

z가 given일 때 x의 확률분포

$$p_\theta(x|z^{(i)})$$

어떻게 학습?

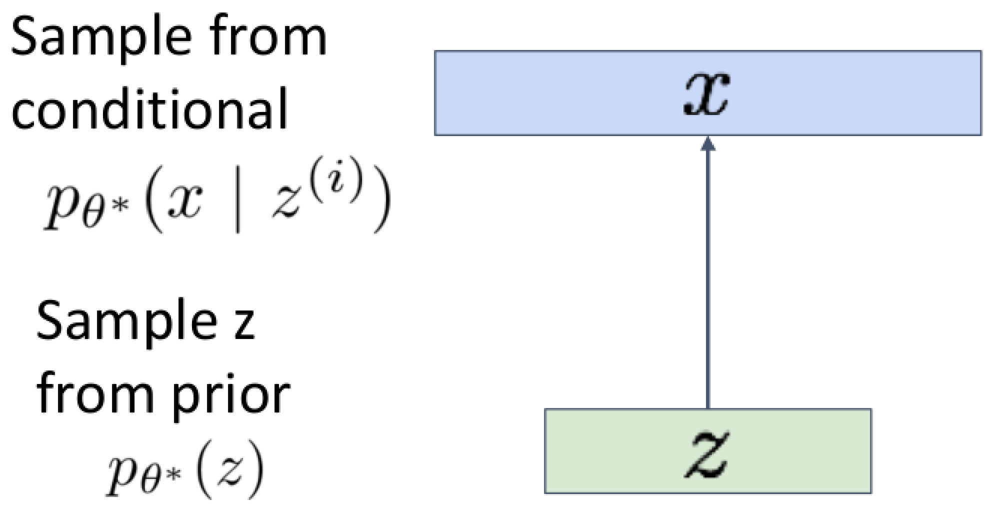
네트워크의 출력값이 있을 때
우리가 원하는 정답 x가 나올 확률이 높길바람

= x의 likelihood를 최대화하는 확률분포 찾자

Maximize

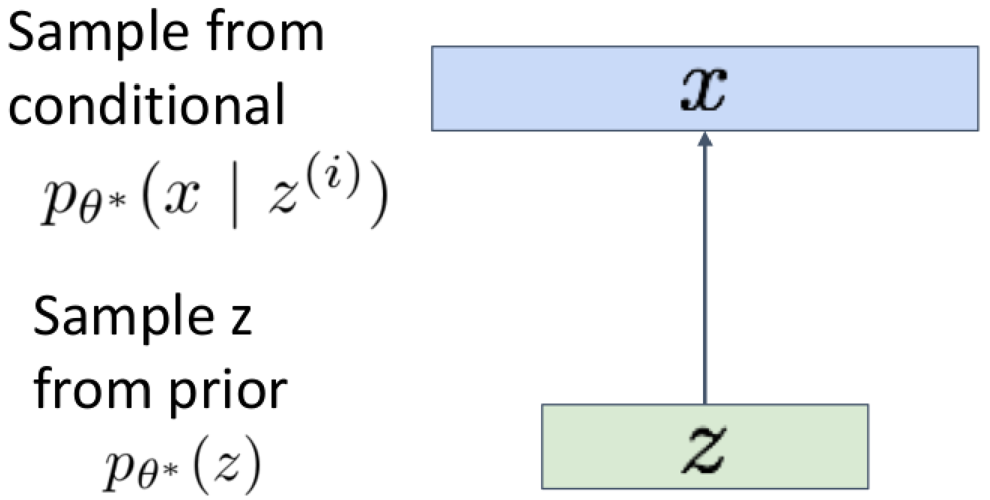
$$p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$$

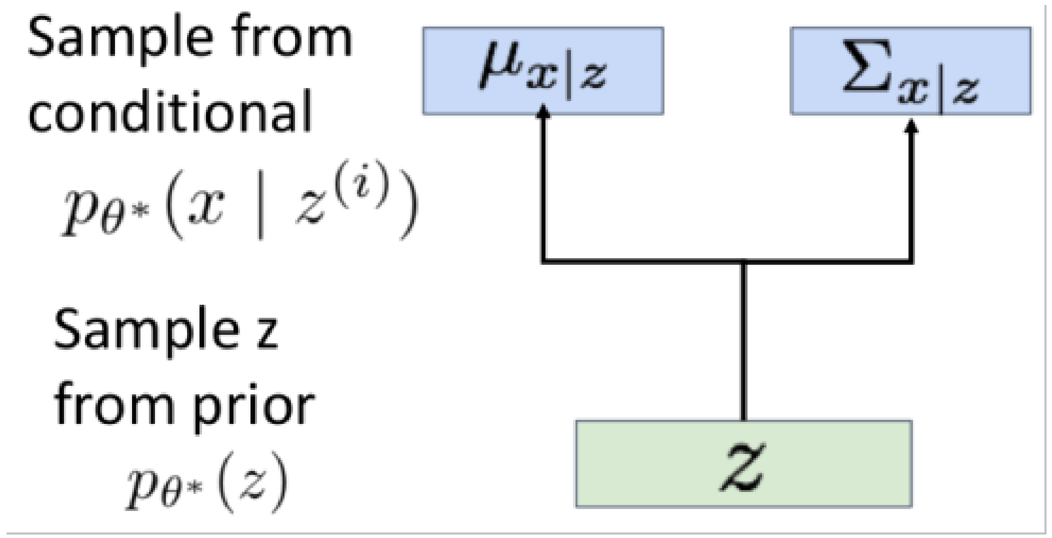
Variational Autoencoders



- When image comes in, $p(z)$ is the appropriate Gaussian distribution for each pixel of image.
- That is, each pixel has a Gaussian distribution with μ, σ
- But if it's a **high resolution** image, it's going to have a lot of values. Therefore, the **diagonal gaussian distribution** is used instead of the general Gaussian distribution. That is, when z is given, there is no covariance between pixels of the generated image. **Pixels are independent.**

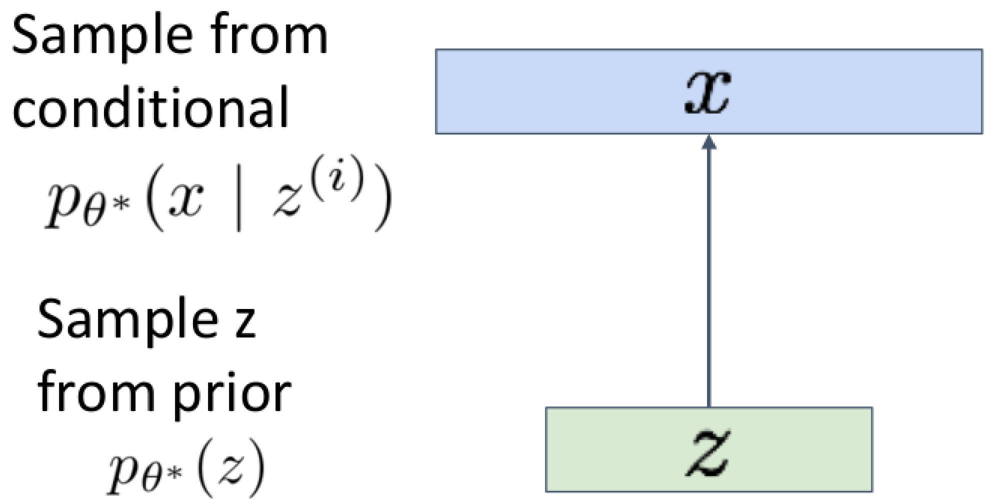
Variational Autoencoders

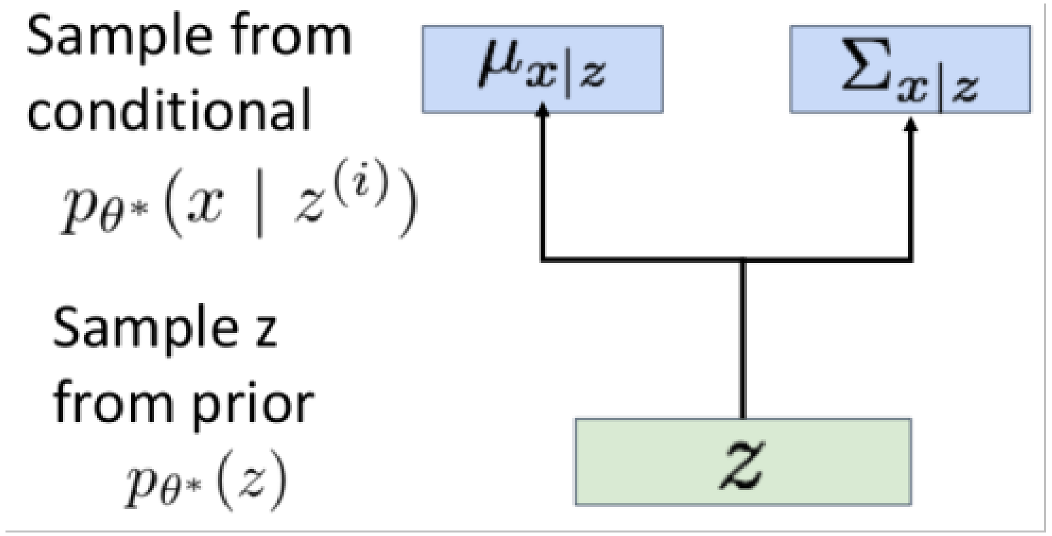




- **Decoder** outputs mean $\mu_{x|z}$ and diagonal covariance $\sum_{x|z}$ for the input z
- Then sample x from the above Gaussian distribution

Variational Autoencoders - Train





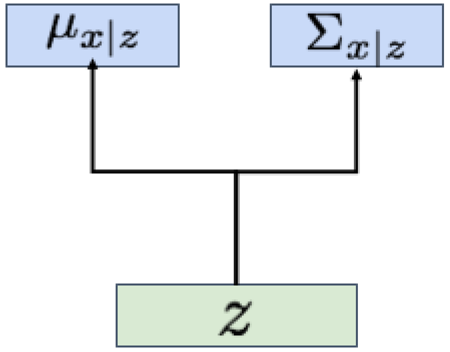
- **Maximize likelihood of data $p_{\theta_*}(x)$**

$$p_\theta(x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(z|x)} \qquad q_\phi(z|x) \approx p_\theta(z|x) \qquad p_\theta(x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(z|x)} \approx \frac{p_\theta(x|z)p_\theta(z)}{q_\phi(z|x)}$$

Variational Autoencoders - Train

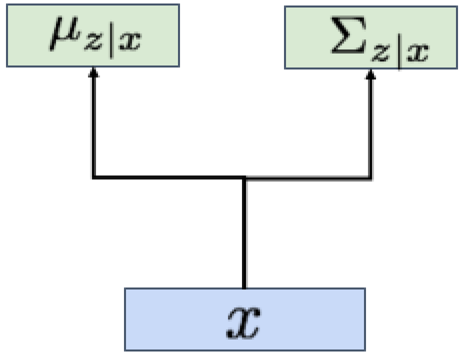
Decoder network inputs latent code z, gives distribution over data x

$$p_\theta(x \mid z) = N(\mu_{x|z}, \Sigma_{x|z})$$



Encoder network inputs data x, gives distribution over latent codes z

$$q_\phi(z \mid x) = N(\mu_{z|x}, \Sigma_{z|x})$$



$$\log p_\theta(x) = \log \frac{p_\theta(x|z)p(z)}{p_\theta(z|x)}$$

$$= \log \frac{p_\theta(x|z)p(z)q_\phi(z|x)}{p_\theta(z|x)q_\phi(z|x)}$$

$$= \log p_\theta(x|z) - \log \frac{q\phi(z|x)}{p(z)} + \log \frac{q_\phi(z|x)}{p_\theta(z|x)}$$

wrap in an expectation since it doesn't depend on z

$$\log p_\theta(x) = E_{z\sim q\phi(z|x)}[\log p_\theta(x)]$$

$$= E_z[\log p_\theta(x|z)] - E_z\left[\log \frac{q_\phi(z|x)}{p(z)}\right] + E_z\left[\log \frac{q_\phi(z|x)}{p_\theta(z|x)}\right]$$

$$= E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x), p(z)) + D_{KL}(q_\phi(z|x), p_\theta(z|x))$$

Variational Autoencoders - Train

$$\log p_\theta(x) = E_{z\sim q\phi(z|x)}[\log p_\theta(x)]$$

$$= E_z[\log p_\theta(x|z)] - E_z\left[\log \frac{q_\phi(z|x)}{p(z)}\right] + E_z\left[\log \frac{q_\phi(z|x)}{p_\theta(z|x)}\right]$$

$$= E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x), p(z)) + D_{KL}(q_\phi(z|x), p_\theta(z|x))$$

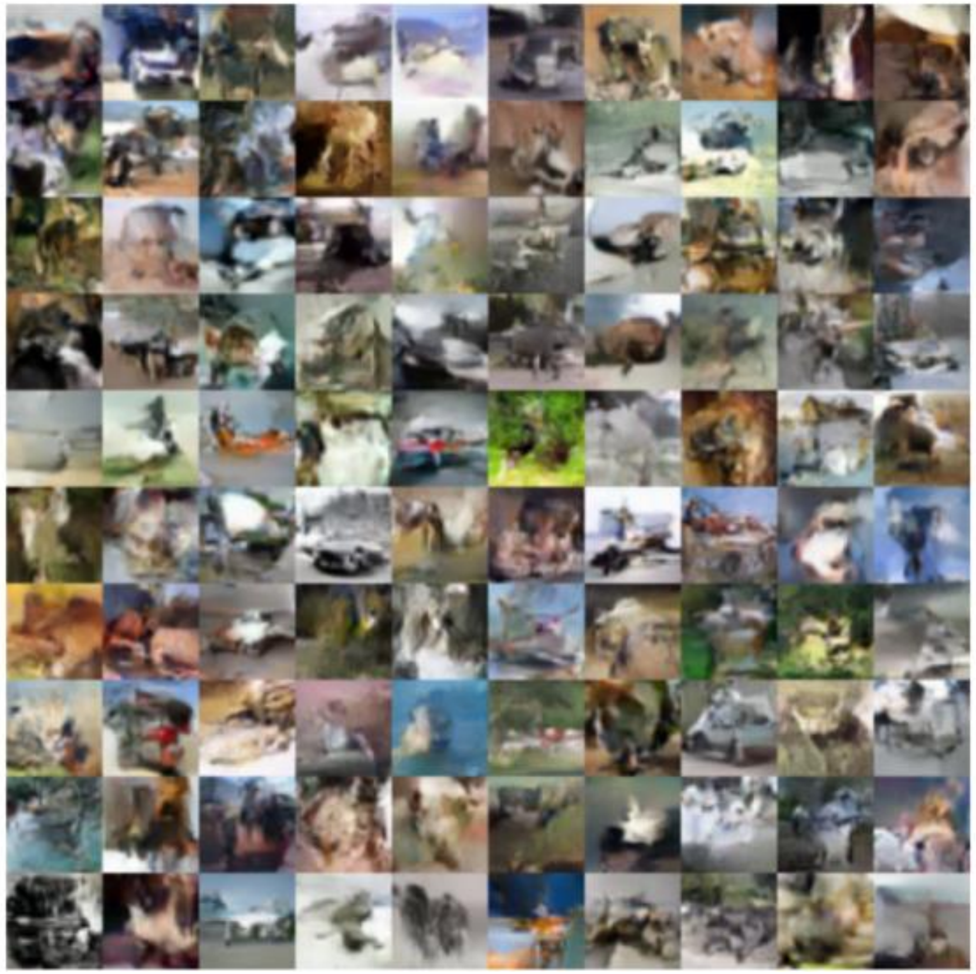
$$\boxed{\log p_\theta(x) \geq E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x), p(z))}$$

Lower bound of likelihood

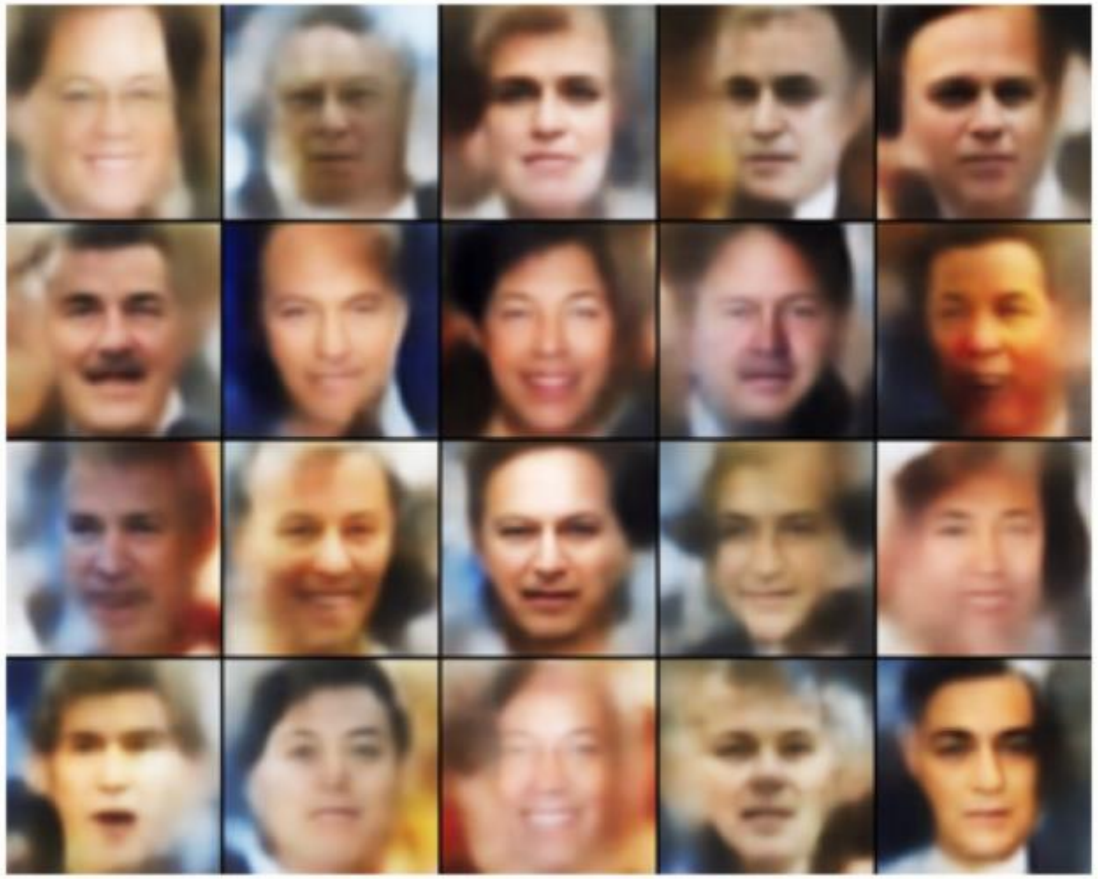
Through this, encoders and decoders are learned jointly to maximize the variable lower bound of data-like hood.

Variational Autoencoders - Train

32x32 CIFAR-10



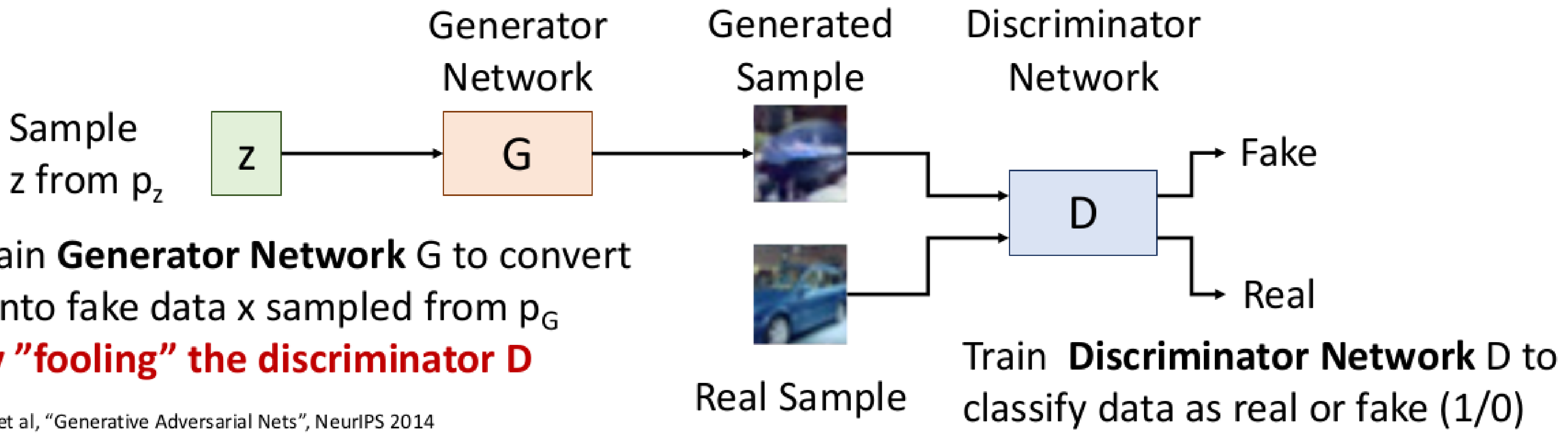
Labeled Faces in the Wild



Generative Adversarial Networks

- **Setup**
  - $p_{\text{data}}(x)$: real data distribution
  - x_i : our train data from $p_{\text{data}}(x)$
- **Idea**
  - Suppose a latent variable z with $p(z)$ which is a simple prior (diagonal Gaussian, unformed distribution, etc.).
  - Sample z from $p(z)$ and pass through **Generative Network $\boldsymbol{G}$.**
  - $\boldsymbol{x} = \boldsymbol{G}(\boldsymbol{z})$
  - Then the x is from Generative distribution p_G.
  - Therefore we want $p_G = p_{\text{data}}$ (Our generative distribution to be real data distribution)

Generative Adversarial Networks



Sample z from p_z
z
Generator Network
G
Generated Sample
Discriminator Network
D
Fake
Real
Real Sample
Train **Generator Network** G to convert z into fake data x sampled from p_G **by "fooling" the discriminator D**
Train **Discriminator Network** D to classify data as real or fake (1/0)

Goodfellow et al, "Generative Adversarial Nets", NeurIPS 2014

- **Generater**
  - By sampling x from p_G, train the model to generate an image that the Discriminator get fooled to think that the image was from p_{data}.
- **Discriminator**
  - Train to discriminate the generated sample and a real sample (real/fake(1/0)).
- Jointly train the two networks. Then p_G will converge to p_{data}.

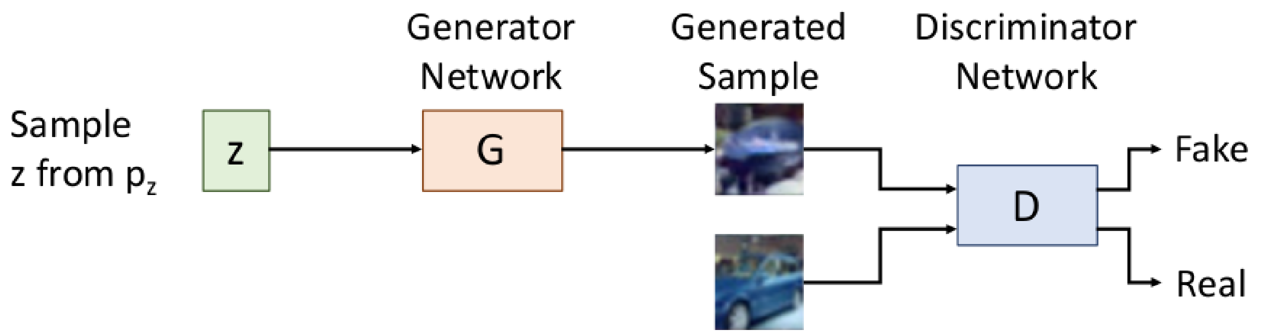
GANs : Training Objective

- Train **Generator G** and **Discriminator D** jointly by **minmax game**.

$$\min_G \max_D \left(E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p(z)}[\log(1 - D(G(z)))]\right)$$

GANs : Training Objective

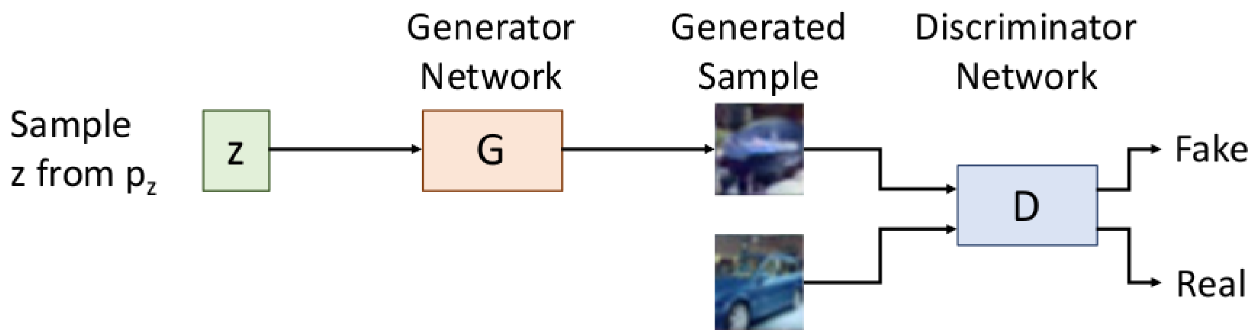
$$\min_{G}\max_{D}\left(\overbrace{E_{x\sim p_{data}}[\log D(x)]}^{\text{Discriminator wants } D(x) = 1 \text{ for real data}} + E_{z\sim p(z)}\underbrace{\overbrace{\left[\log\left(1 - D(G(z))\right)\right]}^{\text{Discriminator wants } D(x) = 0 \text{ for fake data}}}_{\text{Generator wants } D(x) = 1 \text{ for fake data}}\right)$$



- x sampled from p_{data}, which is the real data to be REAL
- If $D(x) < 1$, it passes log term and becomes very small negative value. So we train $D(x) = 1$ that the whole term can be maximized by D.

GANs : Training Objective

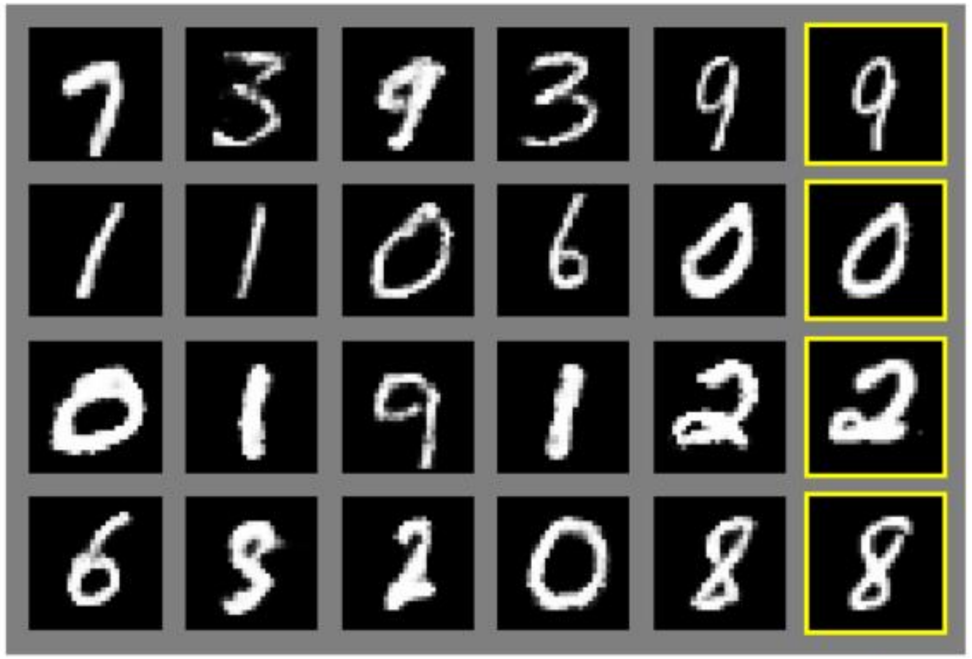
$$\min_{G} \max_{D} \Big(\overbrace{E_{x\sim p_{data}}[\log D(x)]}^{\text{Discriminator wants } D(x)=1 \text{ for real data}} + E_{z\sim p(z)} \overbrace{\underbrace{\Big[\log\big(1 - D(G(z))\big)\Big]}_{\text{Generator wants } D(x)=1 \text{ for fake data}}}^{\text{Discriminator wants } D(x)=0 \text{ for fake data}} \Big)$$

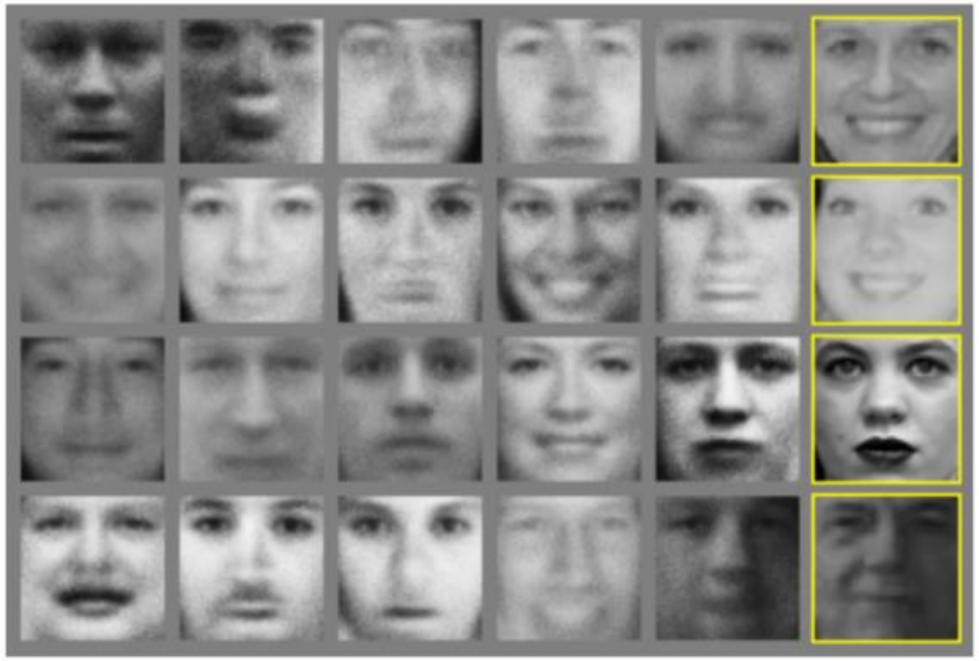


- z sampled from $p(z)$, pass it to Generator G, and G outputs generated sample $G(z)$. Train Discriminator to discriminate the generated sample $G(z)$ is fake. (fake to be FAKE)
- Generator G trains Discriminator D to discriminate $G(z)$ is REAL. (fake to be REAL)

GANs

Generated samples



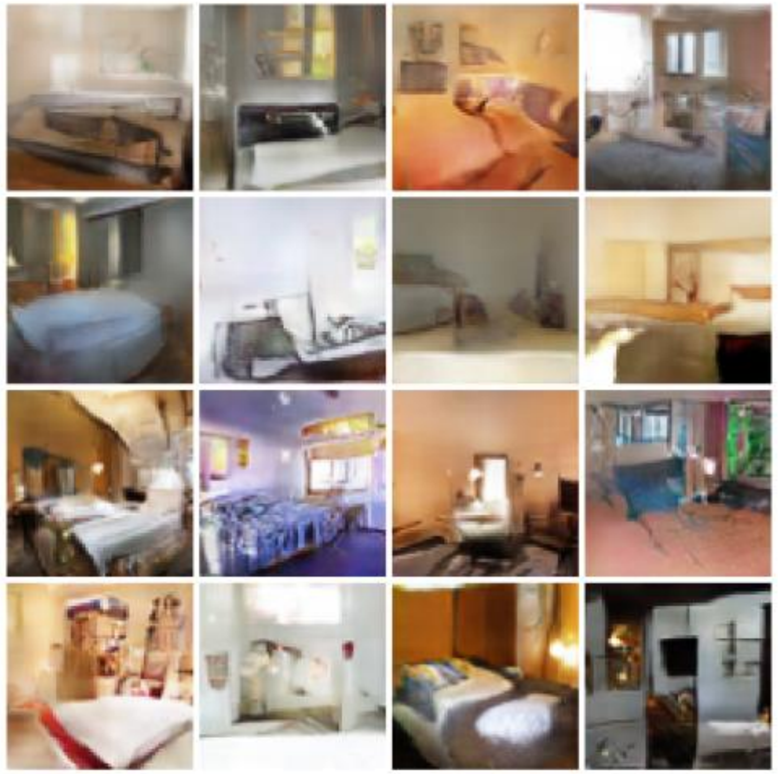


Nearest neighbor from training set

Goodfellow et al, "Generative Adversarial Nets", NeurIPS 2014

GANs

Wasserstein GAN (WGAN)



Arjovsky, Chintala, and Bouttou, “Wasserstein GAN”, 2017

WGAN with Gradient Penalty (WGAN-GP)

Gulrajani et al, “Improved Training of Wasserstein GANs”, NeurIPS 2017

GANs

256 x 256 bedrooms

1024 x 1024 faces

Karras et al, "Progressive Growing of GANs for Improved Quality, Stability, and Variation", ICLR 2018

GANs

512 x 384 cars

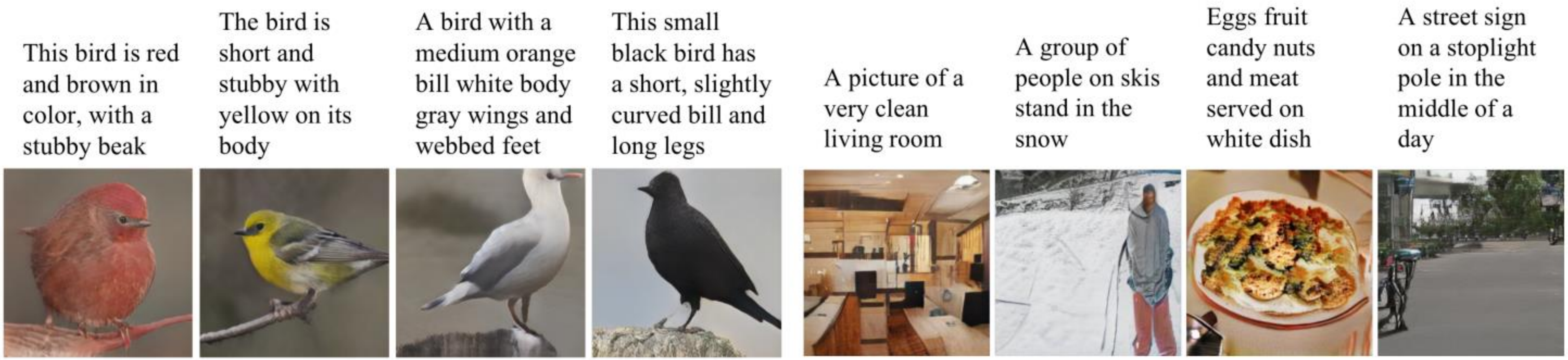
1024 x 1024 faces

Karras et al, “A Style-Based Generator Architecture for Generative Adversarial Networks”, CVPR 2019

GANs

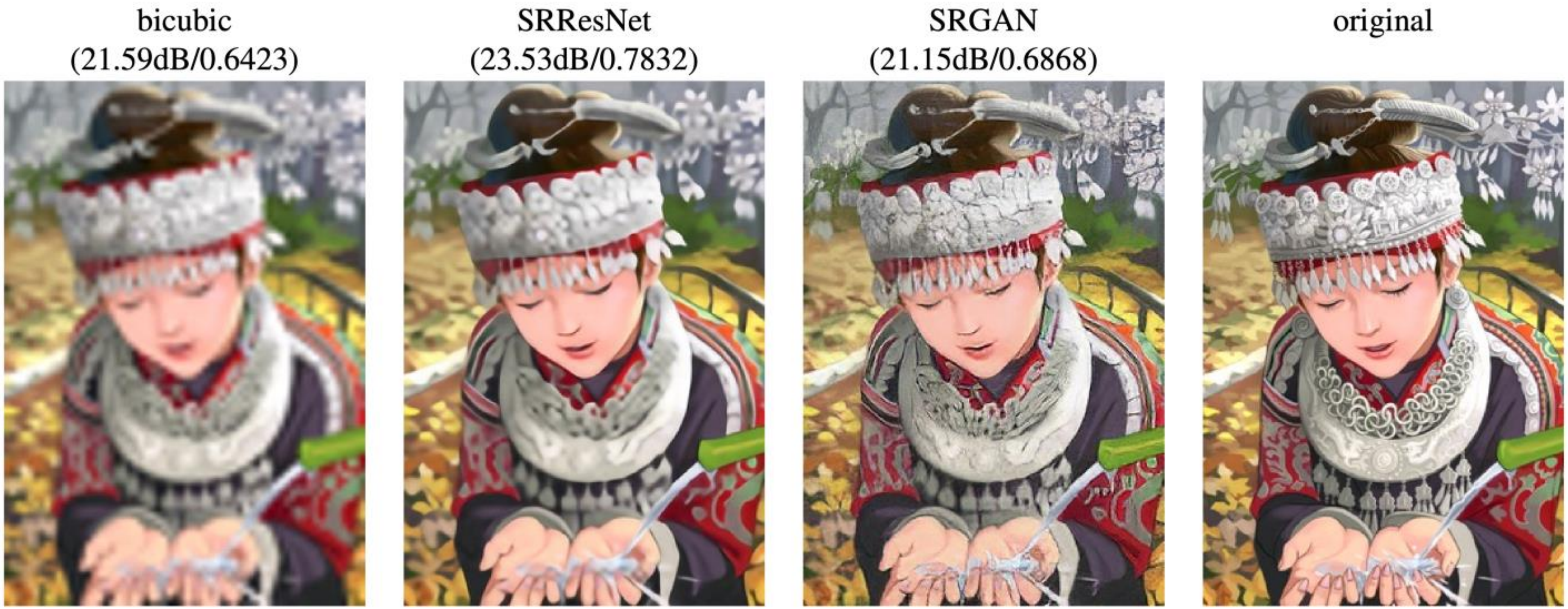
GANs

GANs


This bird is red and brown in color, with a stubby beak
The bird is short and stubby with yellow on its body
A bird with a medium orange bill white body gray wings and webbed feet
This small black bird has a short, slightly curved bill and long legs
A picture of a very clean living room
A group of people on skis stand in the snow
Eggs fruit candy nuts and meat served on white dish
A street sign on a stoplight pole in the middle of a day

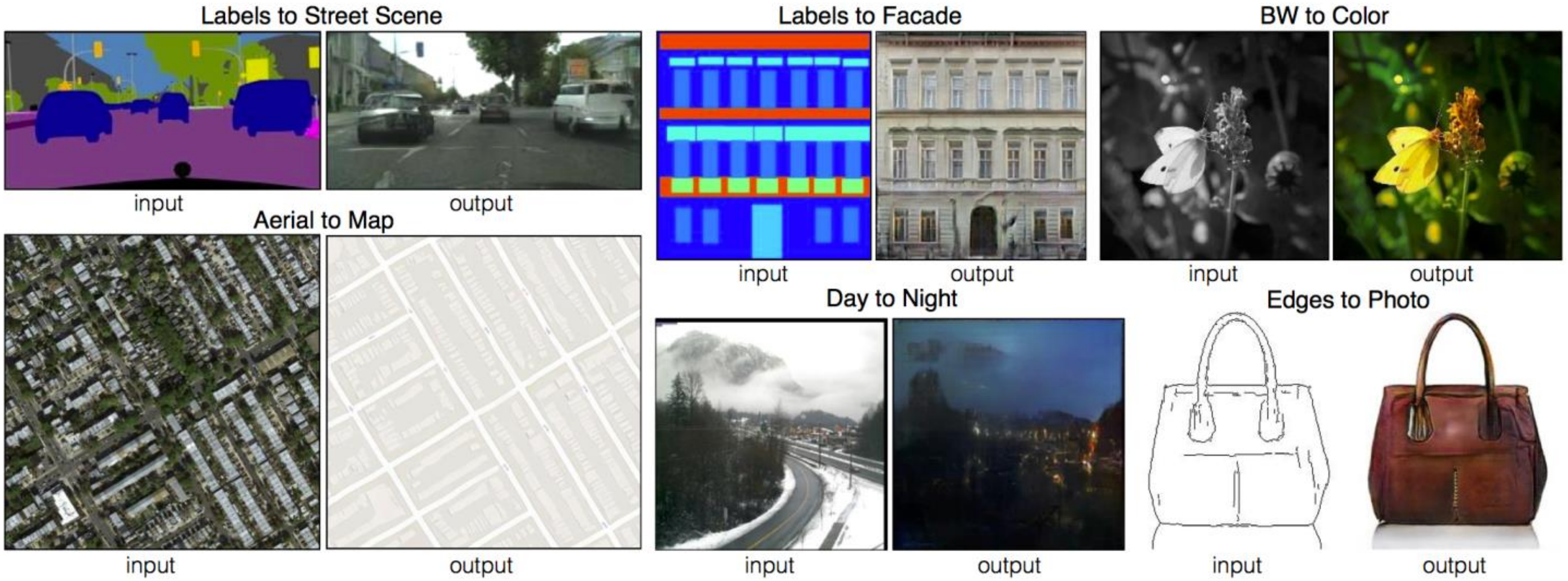
Zhang et al, "StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks.", TPAMI 2018
Zhang et al, "StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks.", ICCV 2017
Reed et al, "Generative Adversarial Text-to-Image Synthesis", ICML 2016

GANs



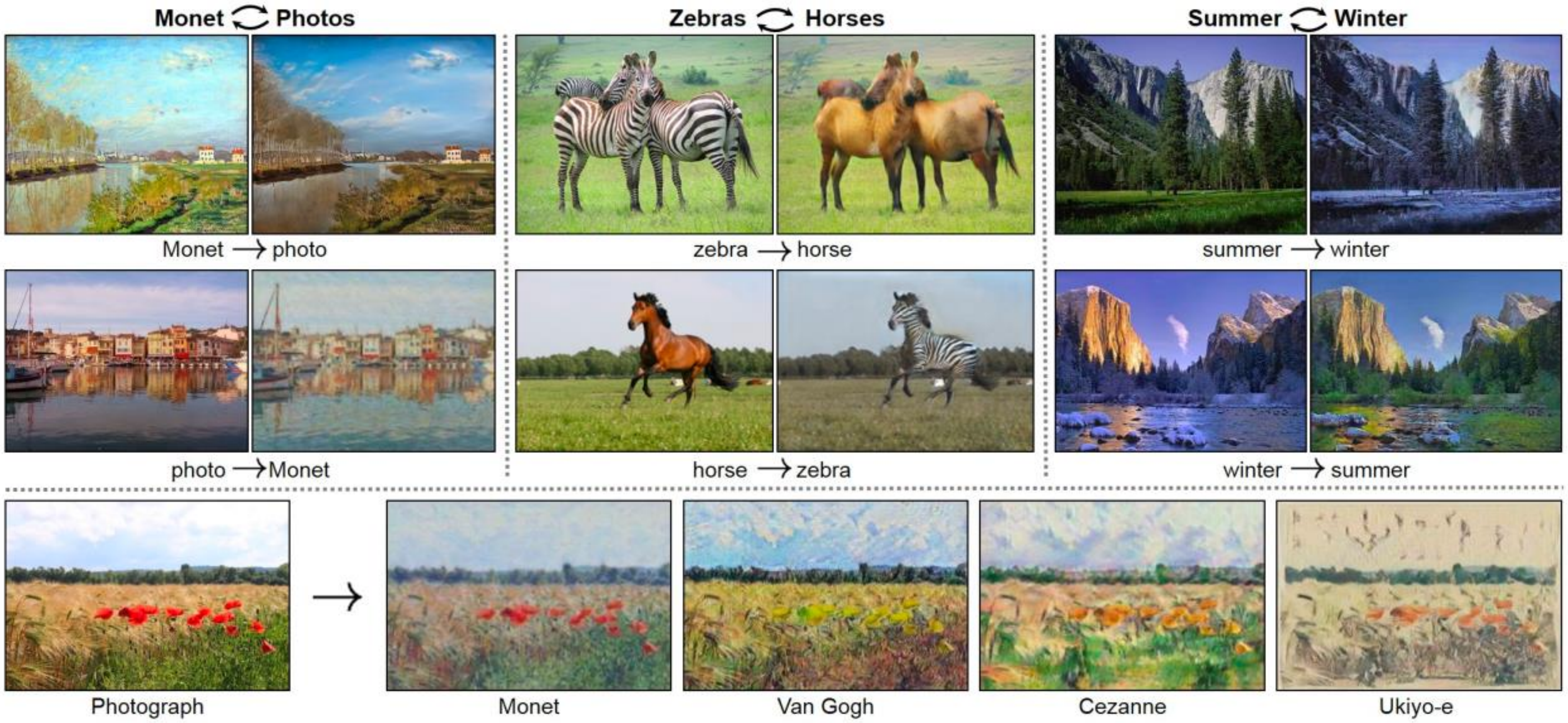
Ledig et al, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", CVPR 2017

GANs



Isola et al, “Image-to-Image Translation with Conditional Adversarial Nets”, CVPR 2017

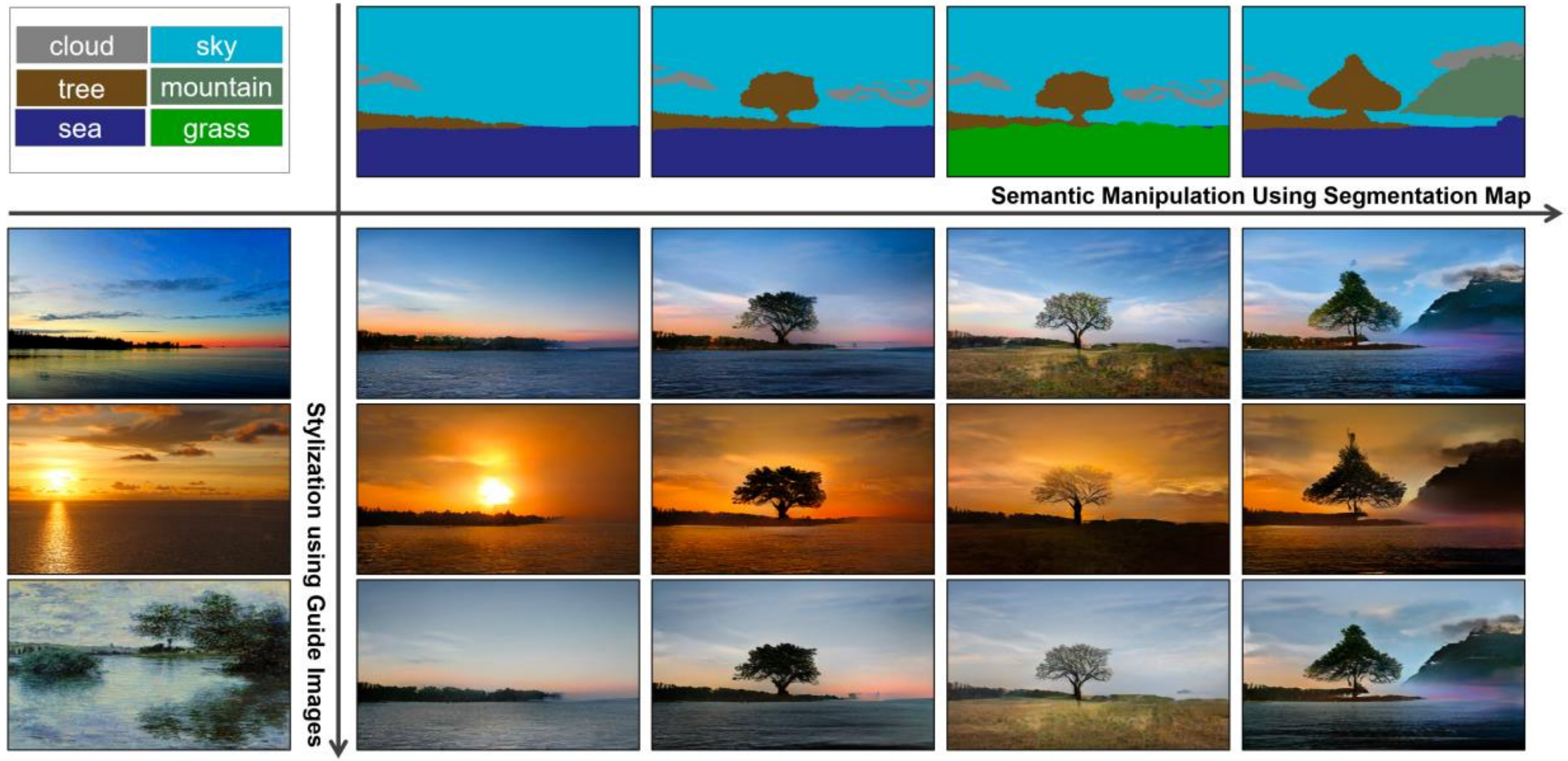
GANs



Zhu et al, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks”, ICCV 2017

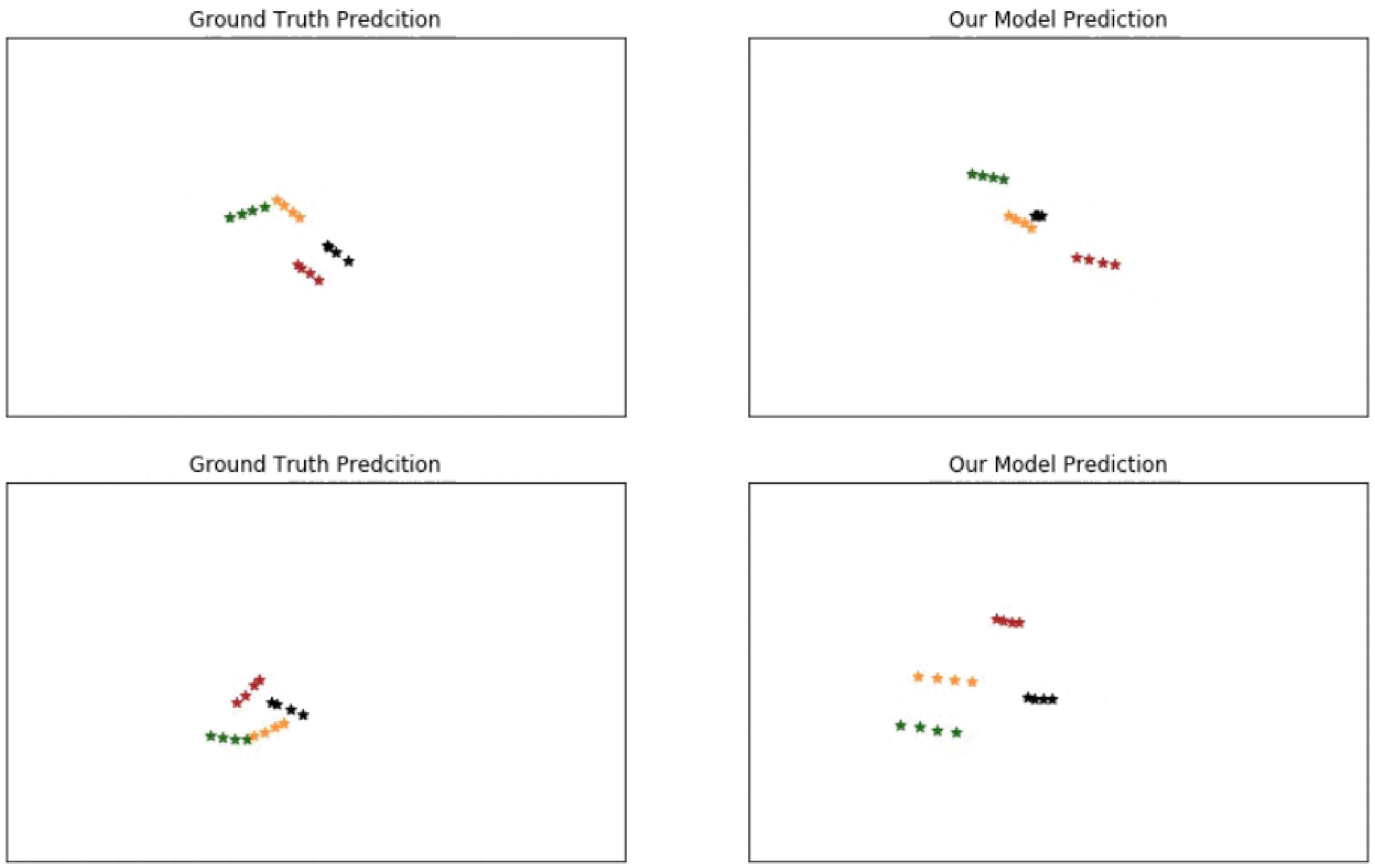
GANs

Label Map to Image


Input: Label Map
cloud
sky
tree
mountain
sea
grass
Semantic Manipulation Using Segmentation Map
Stylization using Guide Images
Input: Style Image

Park et al, "Semantic Image Synthesis with Spatially-Adaptive Normalization", CVPR 2019

GANs



Gupta, **Johnson**, Li, Savarese, Alahi, "Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks", CVPR 2018

Thank you!

Q & A