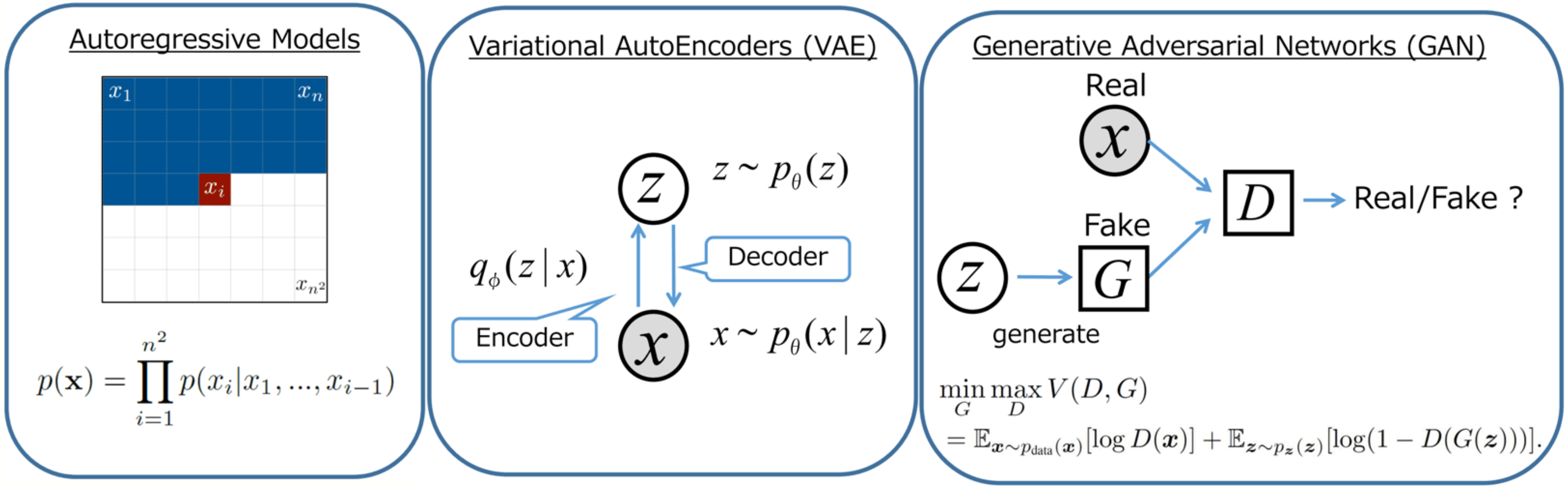
Generative Adversarial Networks

- GAN was first introduced by Ian Goodfellow et al in 2014



Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems*. 2014.

Roadmap

- Why study generative modeling?
- How do GANs work?
- Tips and tricks
- Research frontiers

Why study generative models?

- Have been used in generating images, videos, poems, some simple conversation.

- This co-evolution approach might have far-reaching implications.

Single Image Super-Resolution

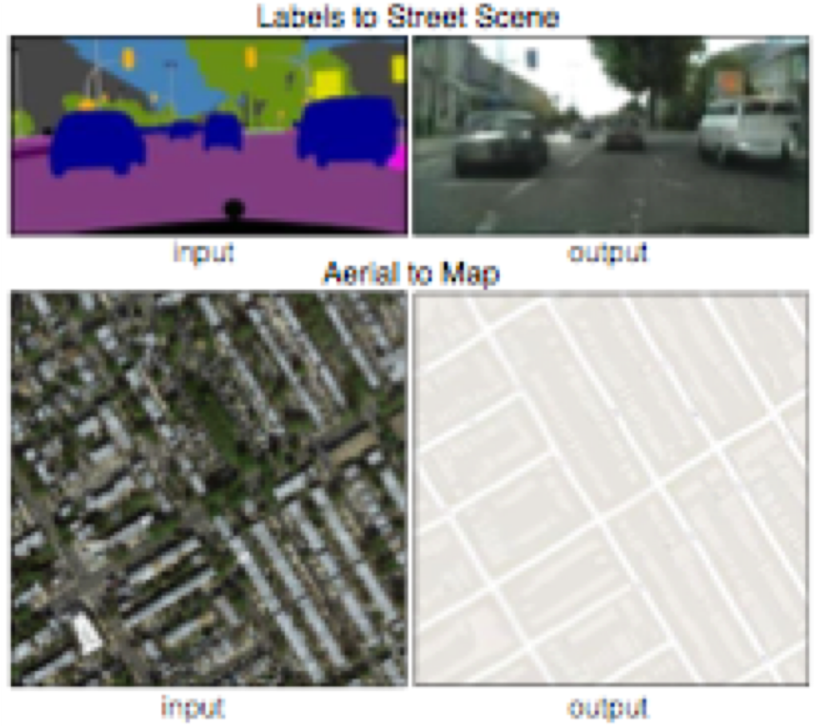
original | bicubic (21.59dB/0.6423) | SRResNet (23.44dB/0.7777) | SRGAN (20.34dB/0.6562)

(Ledig et al 2016)

iGAN

(Zhu et al 2016)

Image to Image Translation



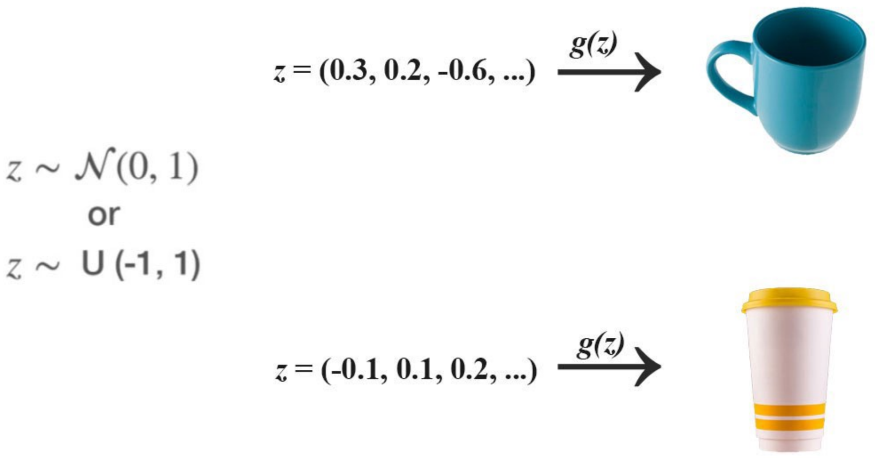
(Isola et al 2016)

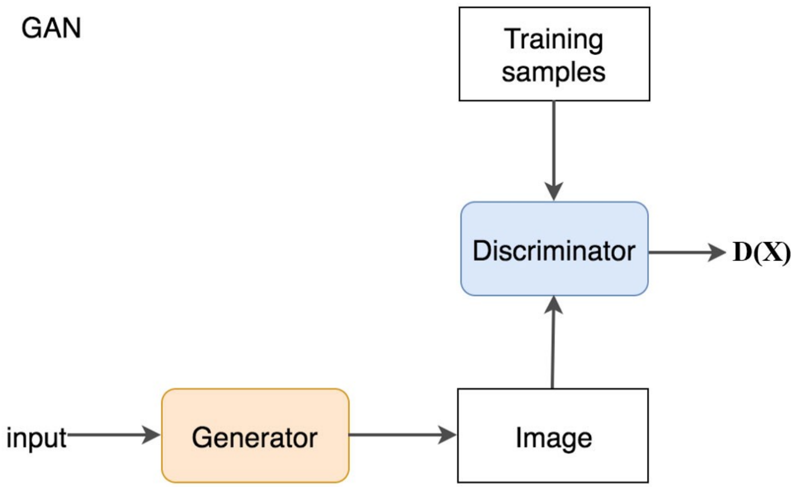
Roadmap

- Why study generative modeling?
- How do GANs work?
- Tips and tricks
- Research frontiers

What are GANs?

- Generative Adversarial Networks (GAN):
  - Generator Network
  - Discriminator Network
- Worst case input of one network is produces by another network





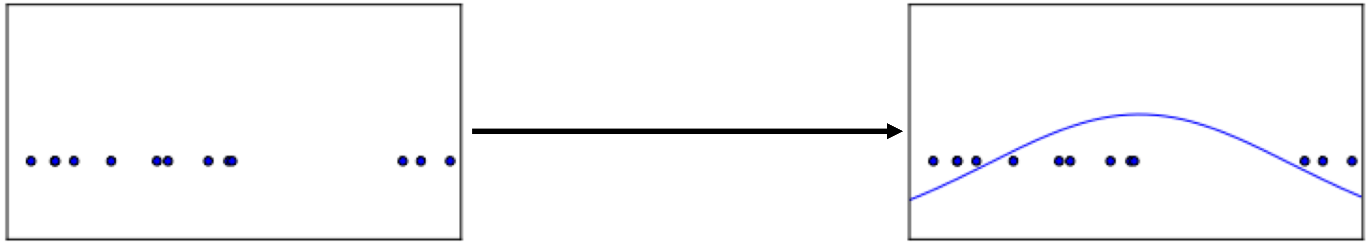
Generative Model

- A **generative** model tries to learn the joint probability of the input data and labels simultaneously i.e. *P(x,y)*.

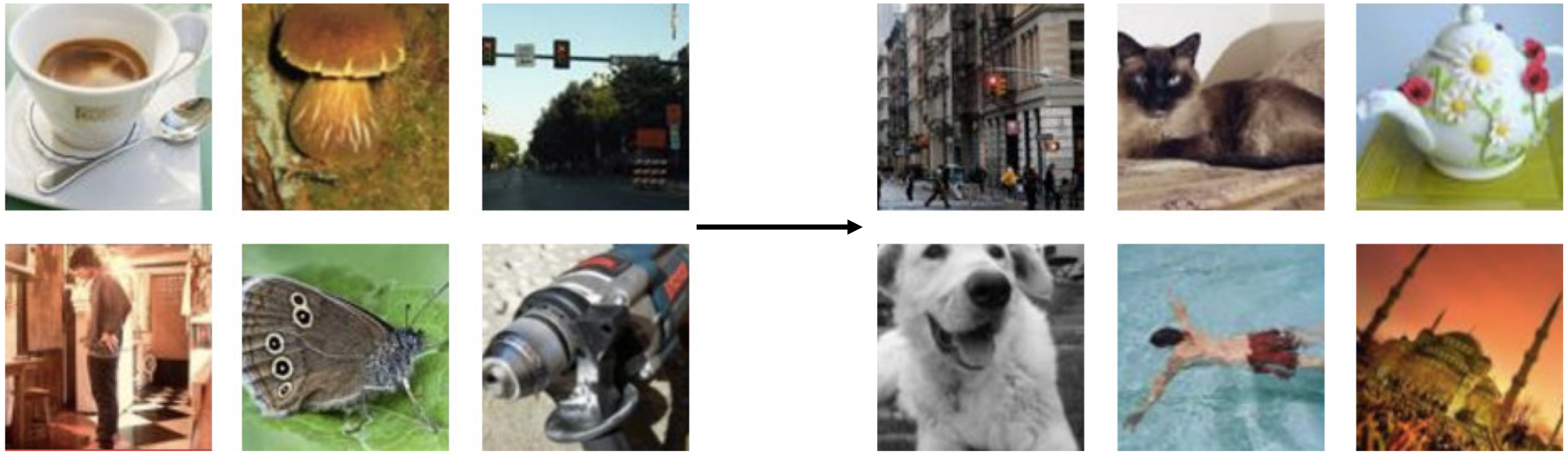
- Potential to understand and explain the underlying structure of the input data even when there are no labels.

Generative Model

- Density estimation



- Sample generation



Training examples Model samples

Discriminative Model

- A **discriminative** model learns a function that maps the input data (*x*) to some desired output class label (*y*).

- In probabilistic terms, they directly learn the conditional distribution $P(y|x)$.

Minimax Game

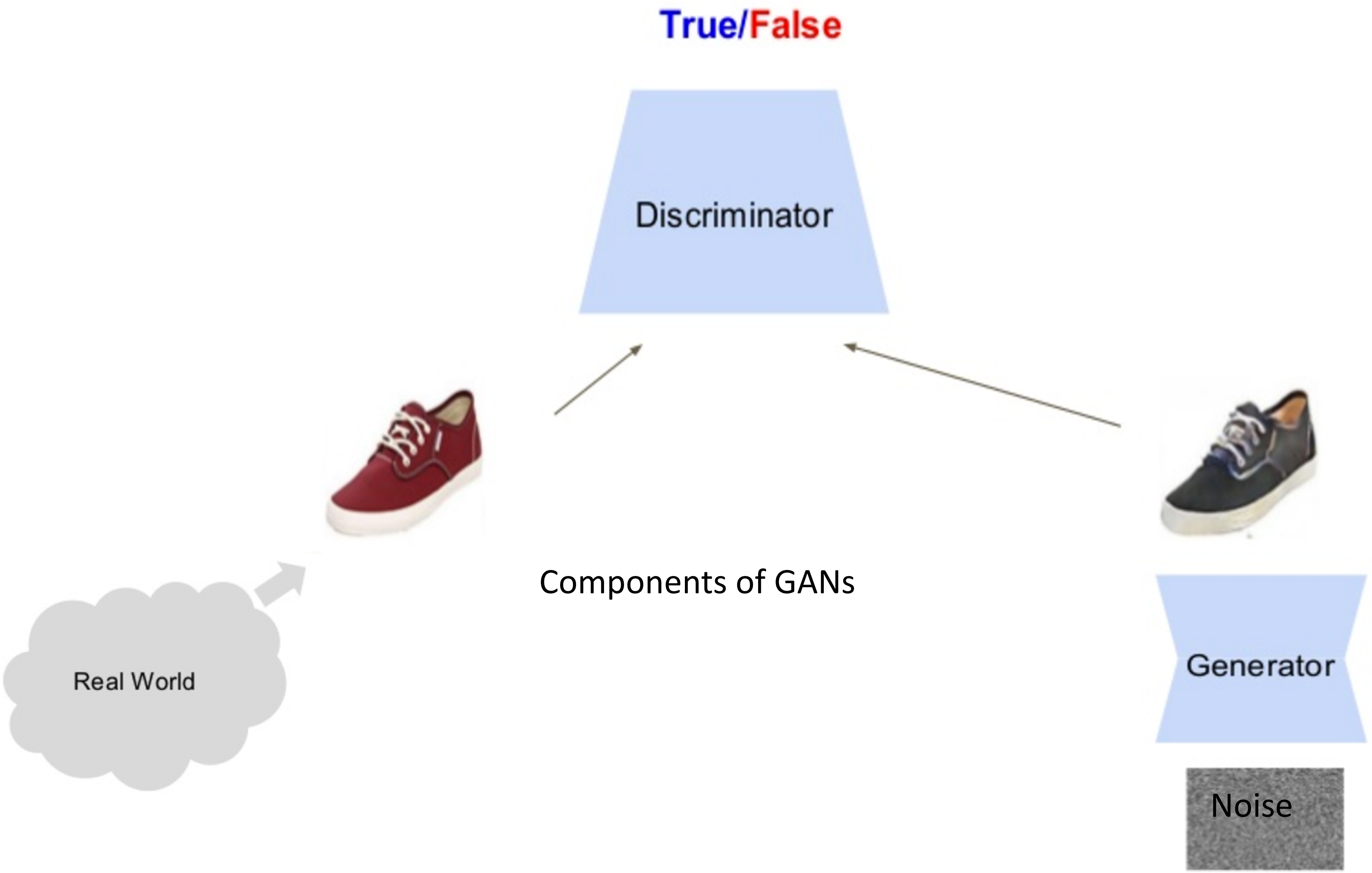
$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

-Equilibrium is a saddle point of the discriminator loss

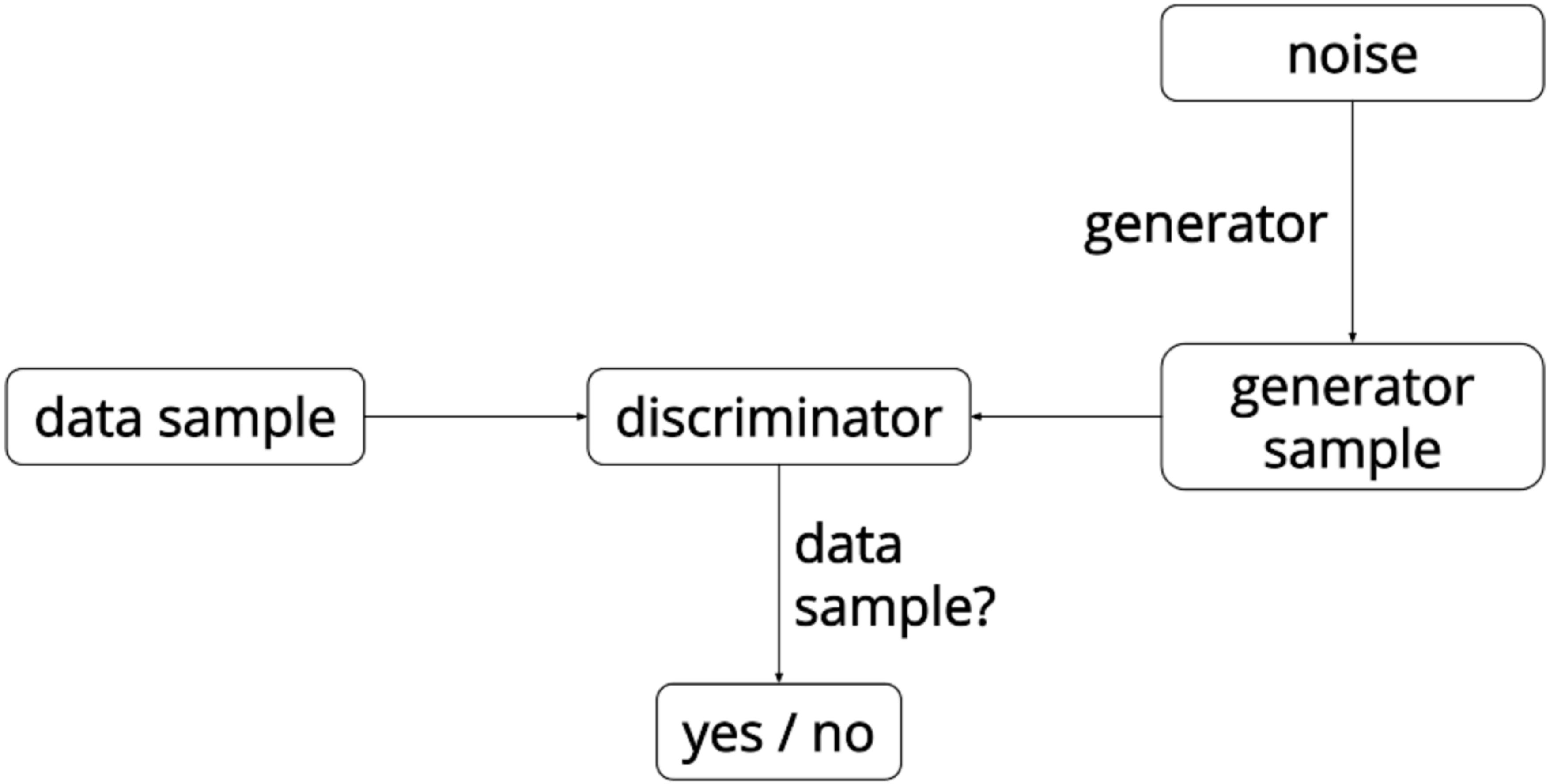
-Resembles Jensen-Shannon divergence

-Generator minimizes the log-probability of the discriminator being correct



Components of GANs

Slide credit – Victor Garcia



Overview of GANs
Source: https://ishmaelbelghazi.github.io/ALI

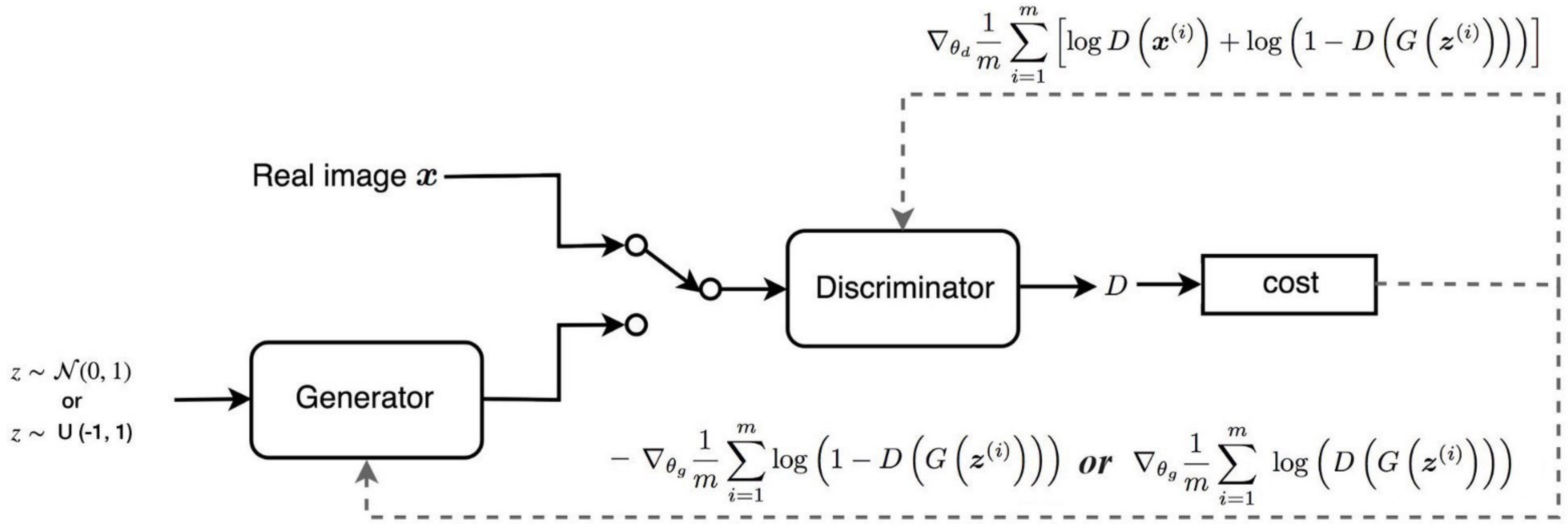
Roadmap

- Why study generative modeling?
- How do GANs work?
- <u>Tips and tricks</u>
- Research frontiers

How to train GANs?

- Objective of generative network - increase the error rate of the discriminative network.
- Objective of discriminative network – decrease binary classification loss.
- Discriminator training - backprop from a binary classification loss.
- Generator training - backprop the negation of the binary classification loss of the discriminator.

How to train GAN



Labels improve subjective sample quality

- Learning a conditional model *p(y|x)* often gives much better samples from all classes than learning *p*(*x*) does (Denton et al 2015)
- Even just learning *p(x,y)* makes samples from *p(x)* look much better to a human observer (Salimans et al 2016)
- Note: this defines three categories of models (no labels, trained with labels, generating condition on labels) that should not be compared directly to each other

Batch Norm

- Given inputs $\boldsymbol{X}$={$\boldsymbol{x}^{(1)}$, $\boldsymbol{x}^{(2)}$, .., $\boldsymbol{x}^{(m)}$}
- **Compute mean and standard deviation of features of *X***
- **Normalize features (subtract mean, divide by standard deviation)**
- **Normalization operation is part of the graph**
  - **Backpropagation computes the gradient through the normalization**
  - **This avoids wasting time repeatedly learning to undo the normalization**

Balancing *G* and *D*

- Usually the discriminator “wins”
- This is a good thing—the theoretical justifications are based on assuming *D* is perfect
- Usually *D* is bigger and deeper than *G*
- Sometimes run *D* more often than *G*. Mixed results.
- Do not try to limit *D* to avoid making it “too smart”
  - Use non-saturating cost
  - Use label smoothing

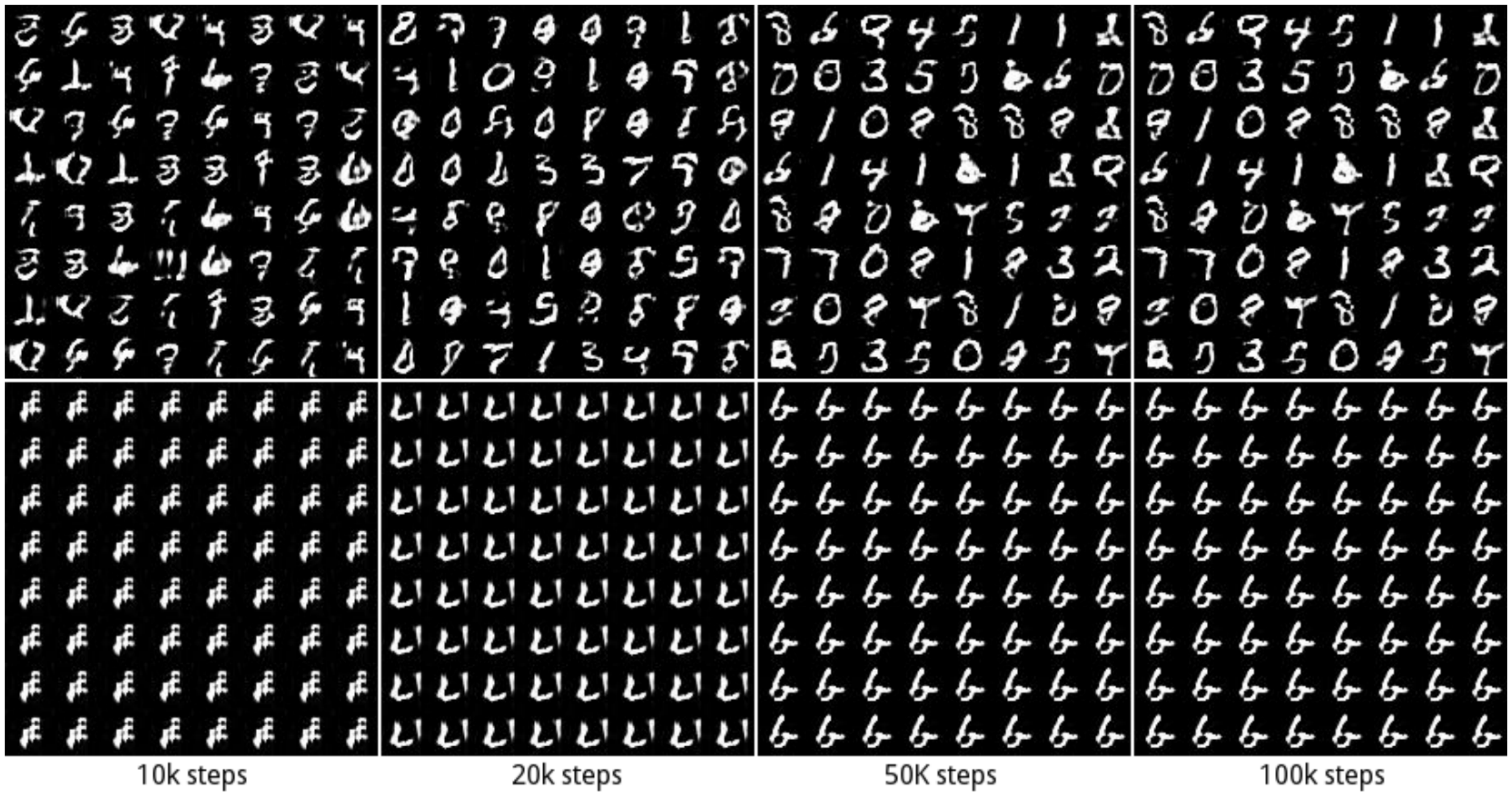
Roadmap

- Why study generative modeling?
- How do GANs work?
- Tips and tricks
- Research frontiers

Non-convergence in GANs

- Optimization algorithms often approach a saddle point or local minimum rather than a global minimum
- Game solving algorithms may not approach an equilibrium at all

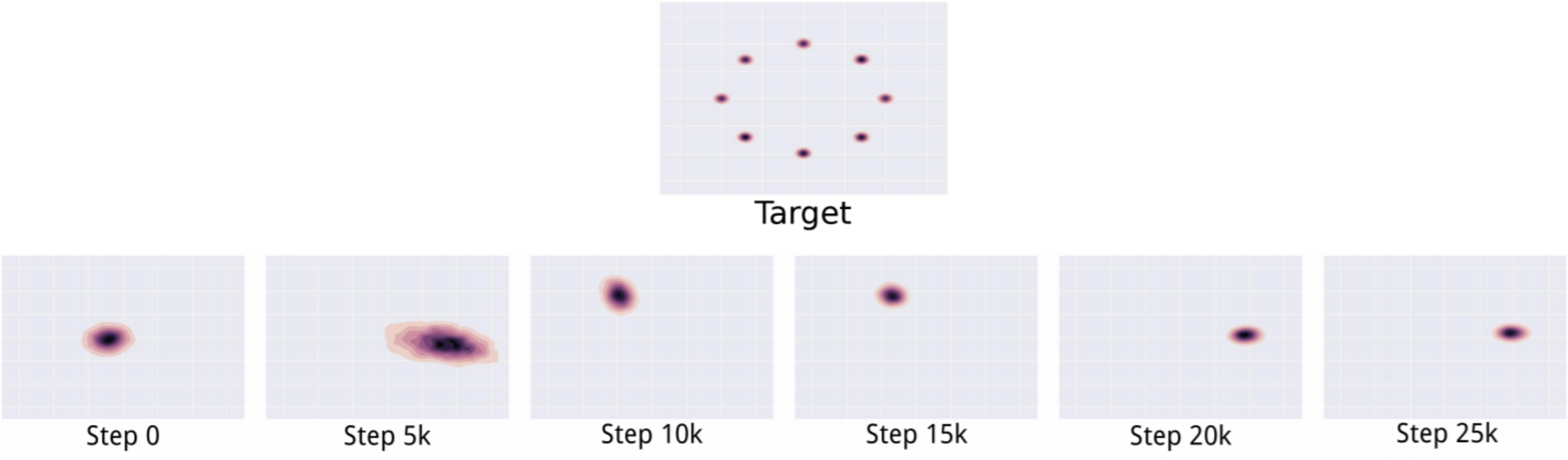
Mode Collapse



Mode Collapse

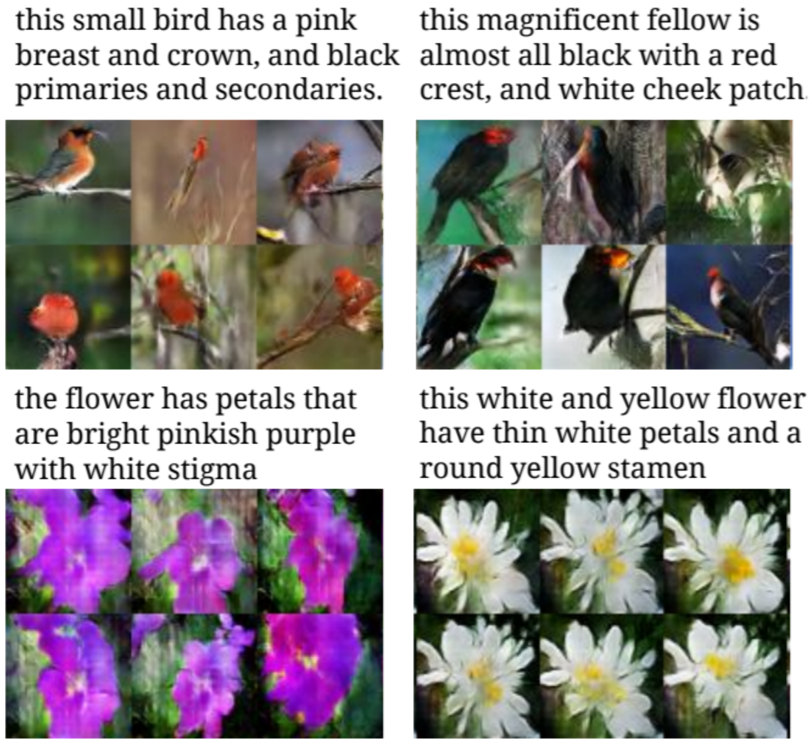
$$\min_G \max_D V(G, D) \neq \max_D \min_G V(G, D)$$

- *D* in inner loop: convergence to correct distribution
- *G* in inner loop: place all mass on most likely point



(Metz et al 2016)

Mode collapse causes low output diversity



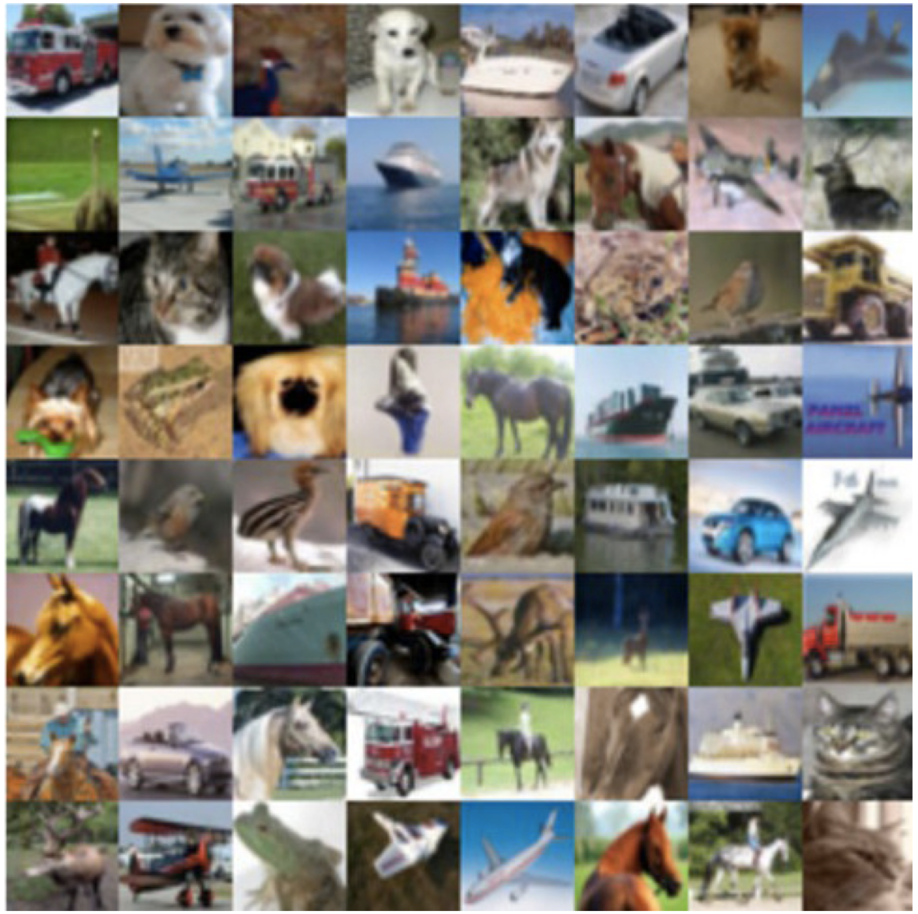
(Reed et al 2016)

(Reed et al, submitted to ICLR 2017)

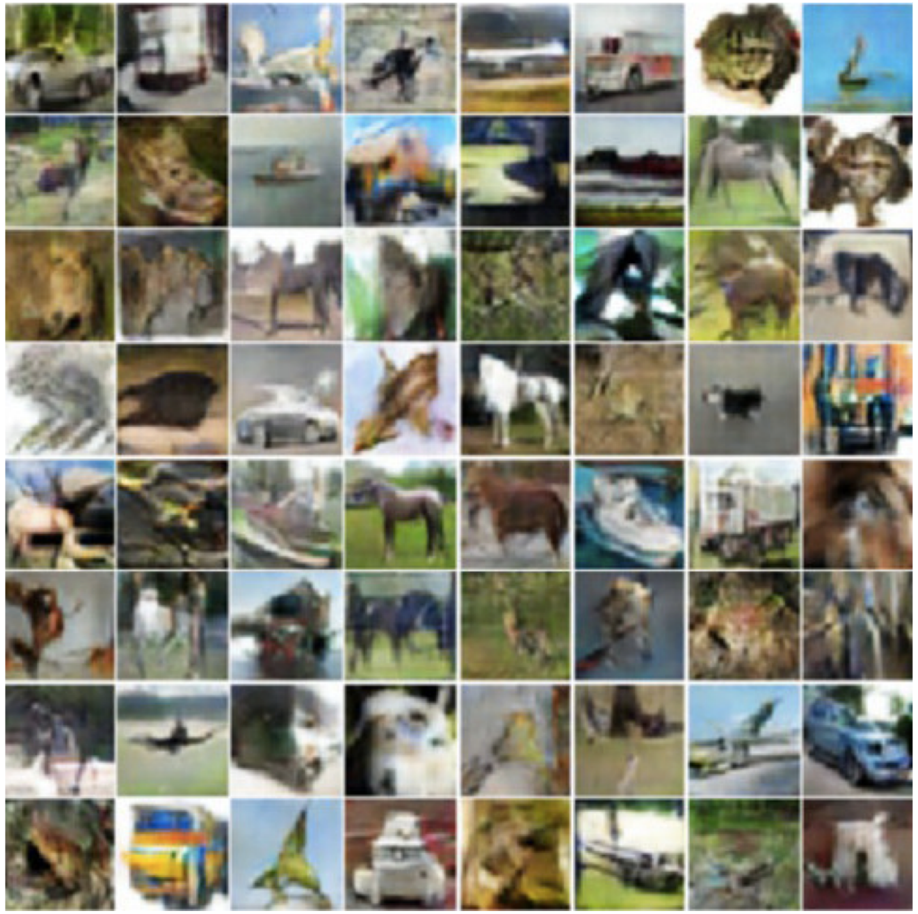
Minibatch Features

- Add minibatch features that classify each example by comparing it to other members of the minibatch (Salimans et al 2016)
- Nearest-neighbor style features detect if a minibatch contains samples that are too similar to each other

Minibatch GAN on CIFAR



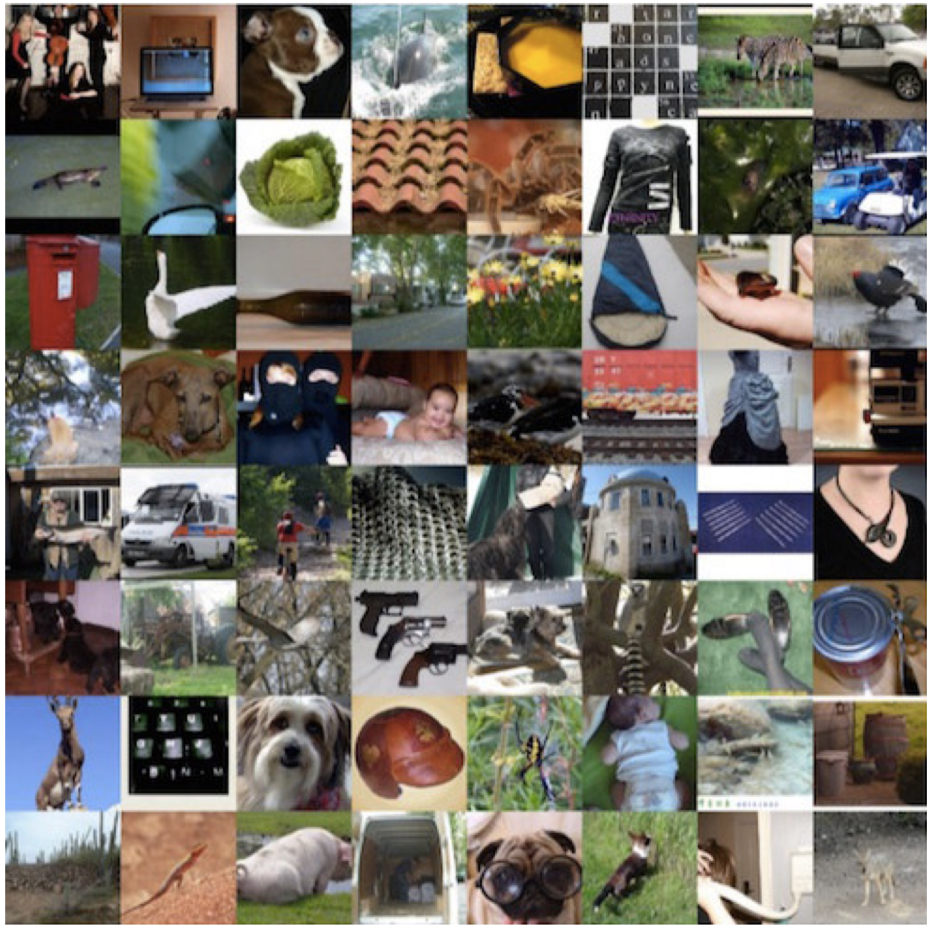
Training Data



Samples

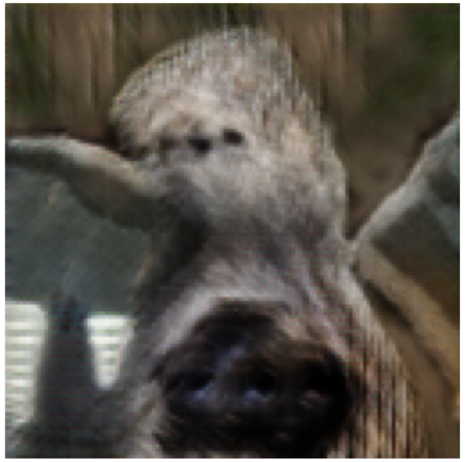
(Salimans et al 2016)

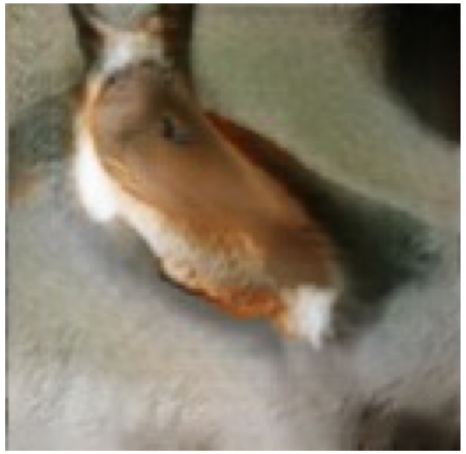
Minibatch GAN on ImageNet



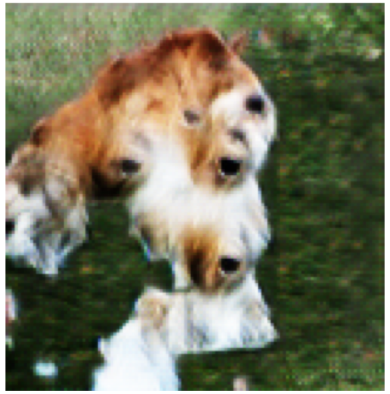
(Salimans et al 2016)

Cherry-Picked Results

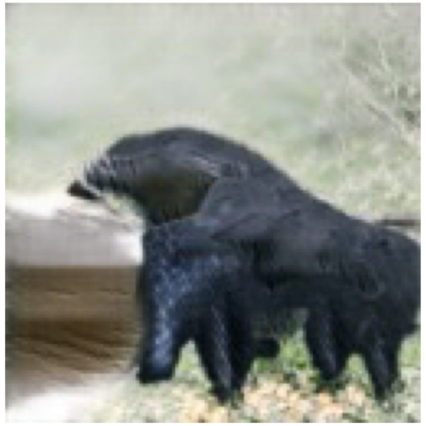



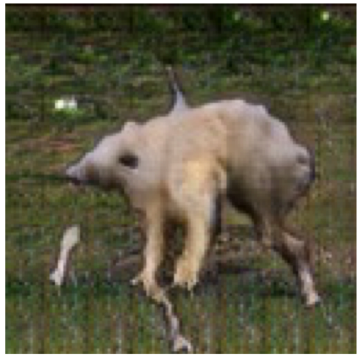
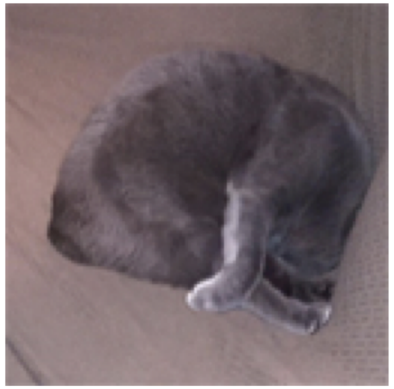
Problems with Counting





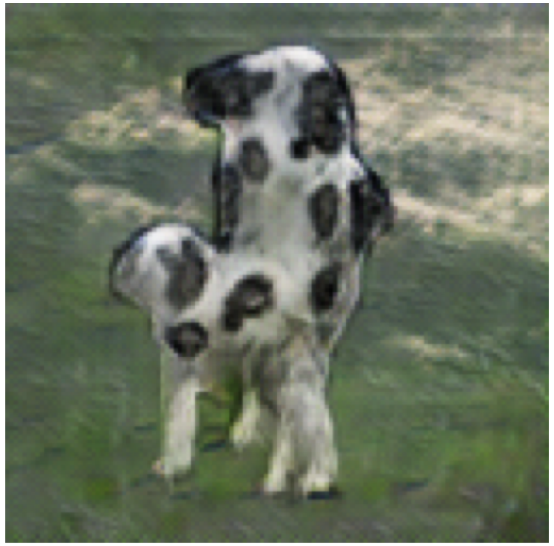

Problems with Perspective



Problems with Global Structure



Vector Space Arithmetic

Man with glasses

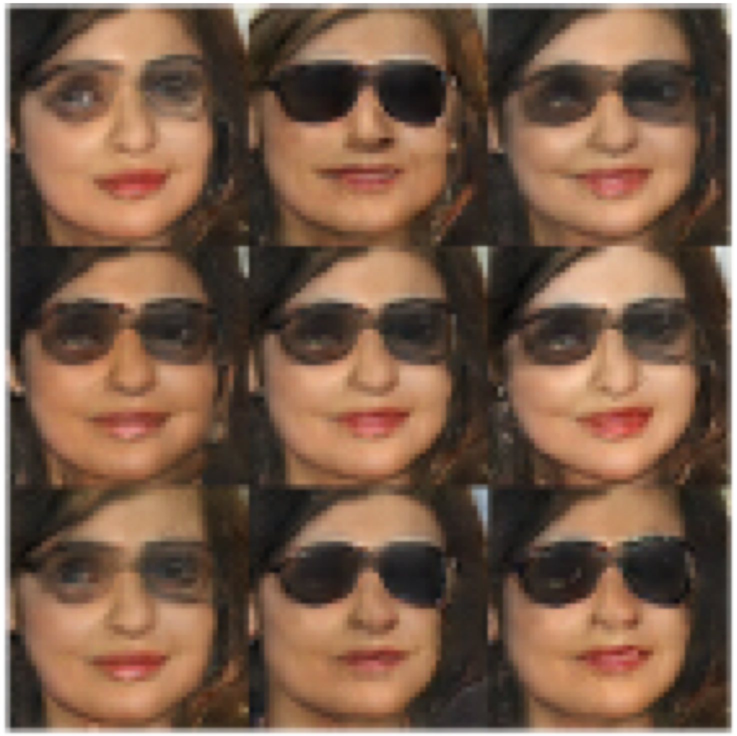
-

Man

+

Woman

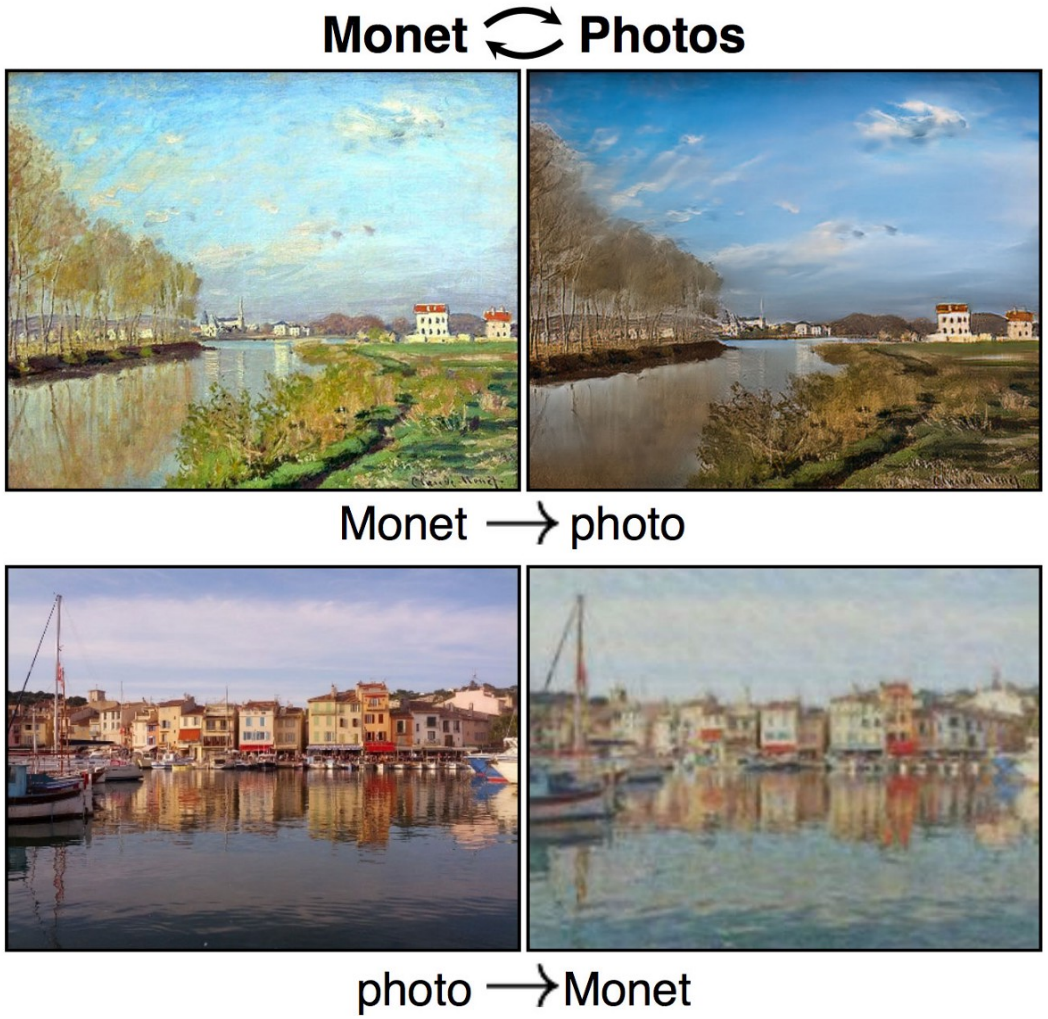
=

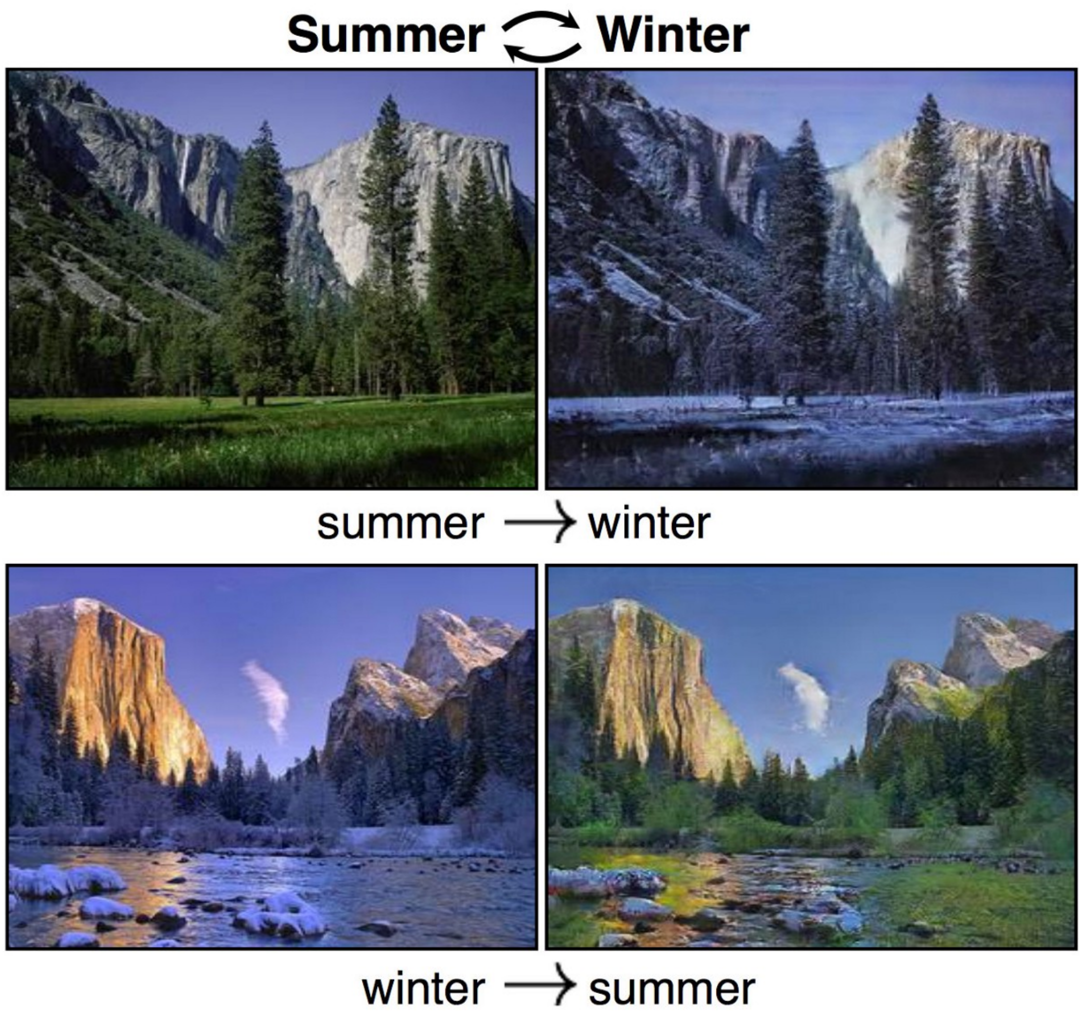


Woman with Glasses

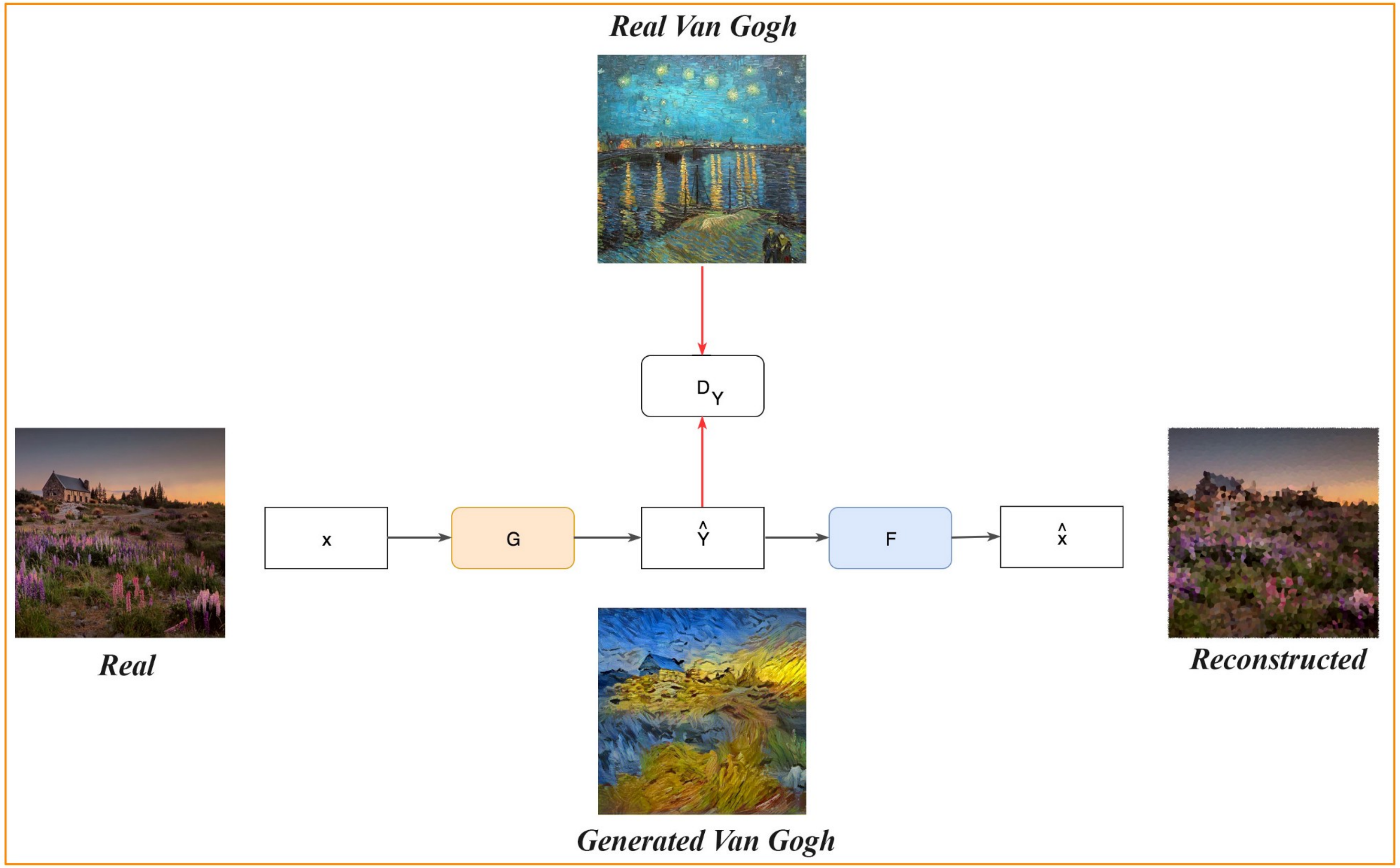
(Radford et al, 2015)

Style transfer

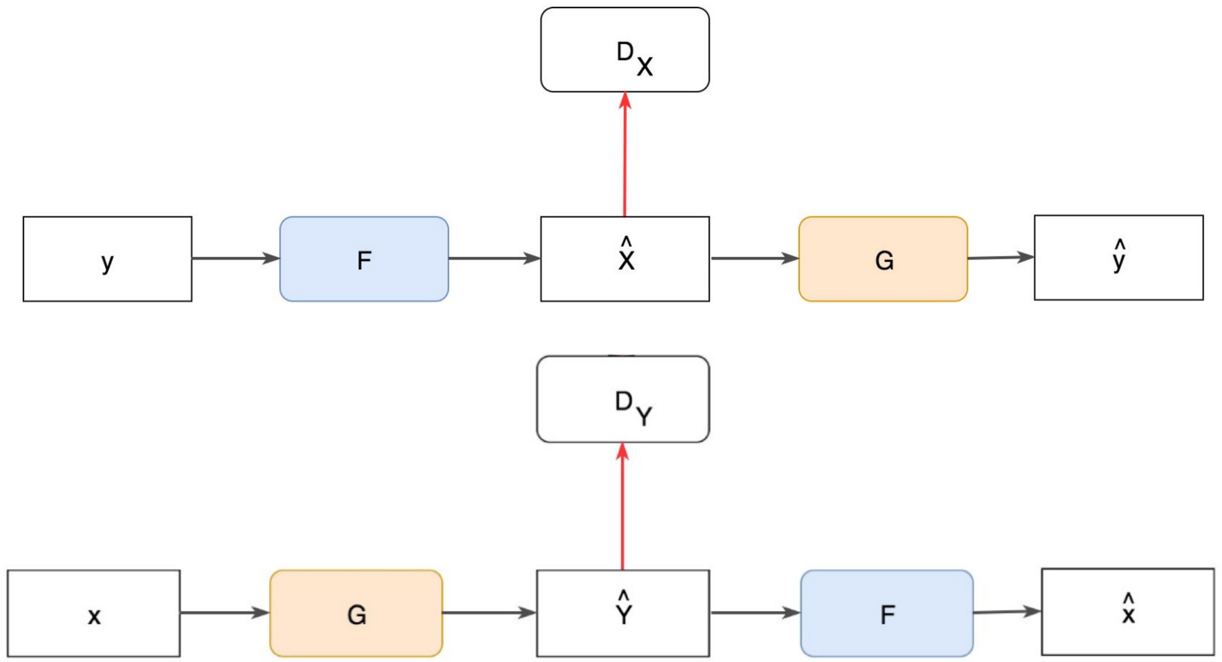




CycleGAN


Real Van Gogh
D_Y
x
G
Ŷ
F
x̂
Real
Reconstructed
Generated Van Gogh

CycleGAN – Network Design



- A generator ***G*** to convert a real image to a Van Gogh style picture.
- A generator ***F*** to convert a Van Gogh style picture to a real image.
- A discriminator ***D*** to identify real or generated Van Gogh pictures.

CycleGAN – Loss Function

Final objective function:

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

- Cycle consistency loss (reconstruction loss):

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

- Adversarial loss

$$\begin{aligned}
&\mathcal{L}_{\text{GAN}}(G, D, X, Y) \\
&\quad \textit{For } G, \text{ minimize } \mathbb{E}_{x \sim p_{\text{data}}(x)}[(D(G(x)) - 1)^2] \\
&\quad \textit{For } D, \text{ minimize } \mathbb{E}_{y \sim p_{\text{data}}(y)}[(D(y) - 1)^2] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[D(G(x))^2]
\end{aligned}$$

CycleGAN – Style Transfer

CycleGAN – Season Transfer

winter Yosemite → summer Yosemite

summer Yosemite → winter Yosemite

CycleGAN – Photo Enhancement

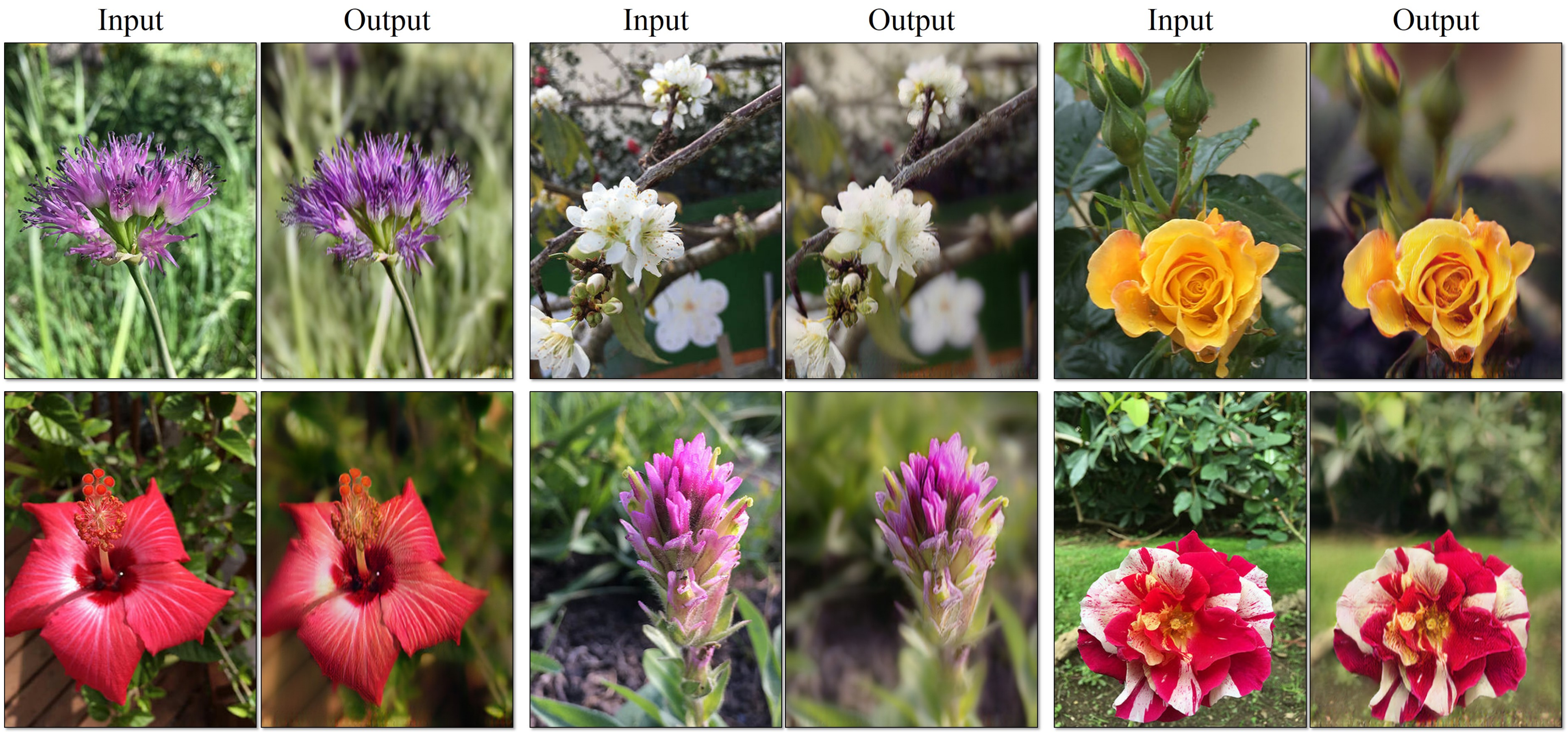
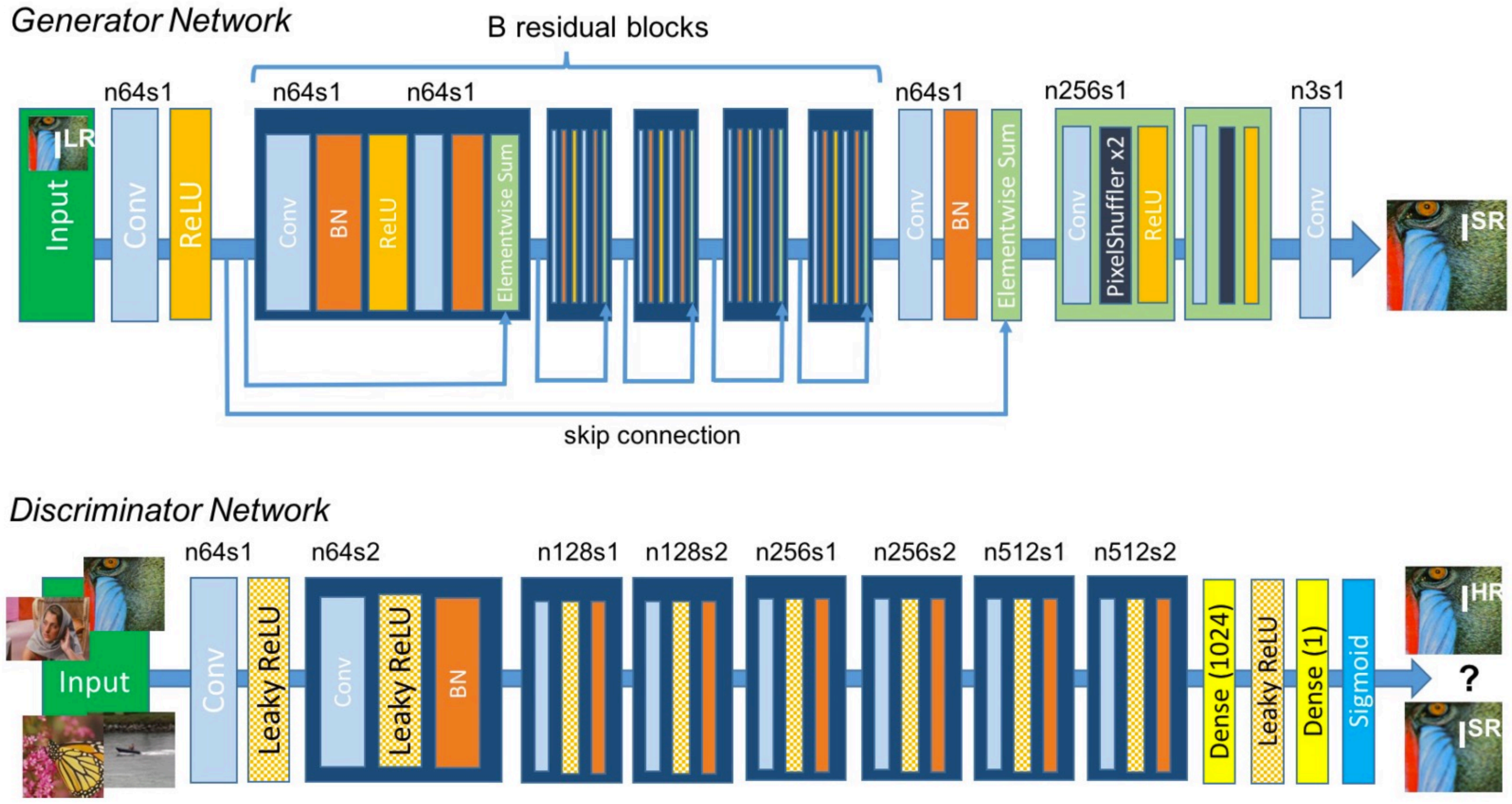


Image Super-resolution

bicubic (21.59dB/0.6423) | SRResNet (23.53dB/0.7832) | SRGAN (21.15dB/0.6868) | original

Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]

SRGAN – Network Design



Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *CVPR*17

SRGAN – Loss Function

- Loss function for the generator

$$l^{SR} = \underbrace{\underbrace{l^{SR}_{X}}_{\text{content loss}} + \underbrace{10^{-3} l^{SR}_{Gen}}_{\text{adversarial loss}}}_{\text{perceptual loss (for VGG based content losses)}}$$

- Adversarial loss:

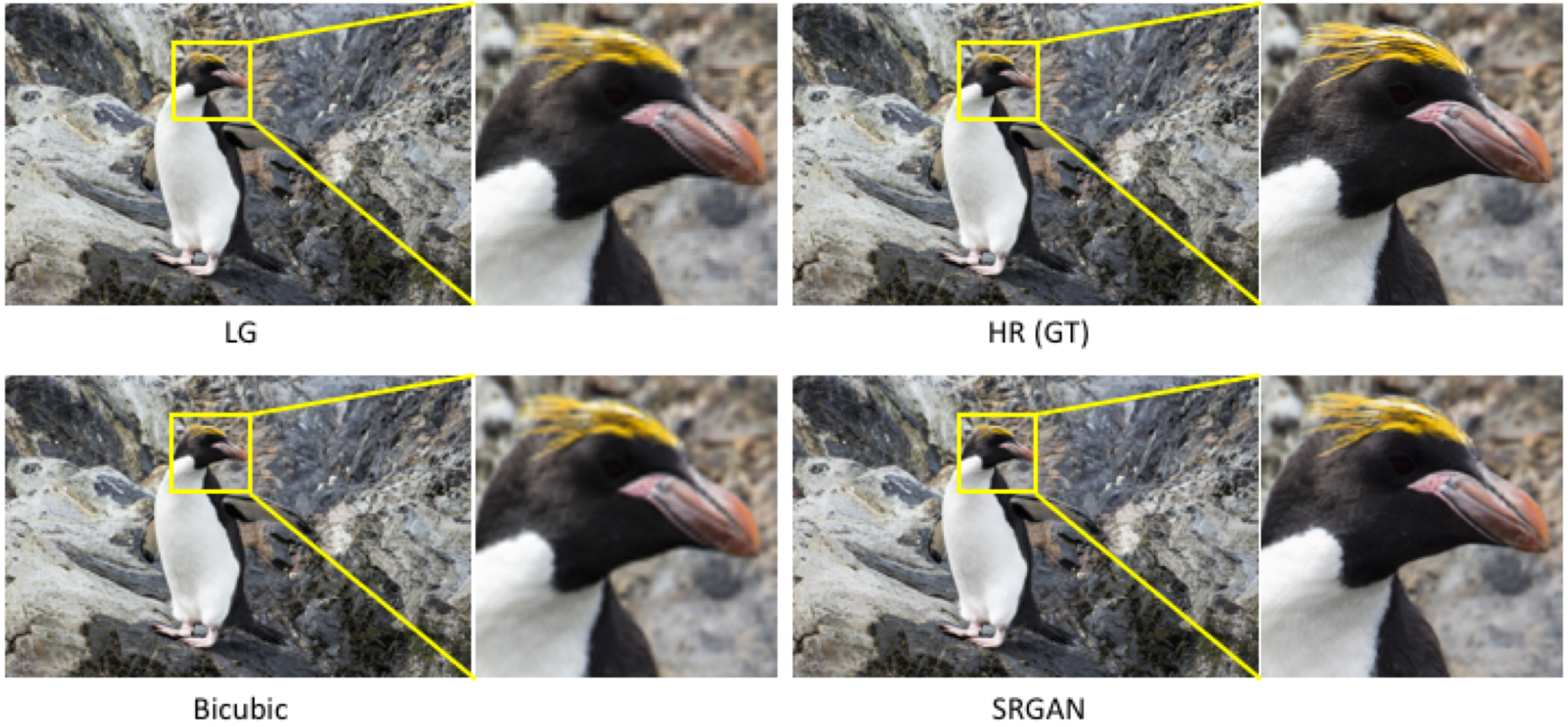
$$l^{SR}_{Gen} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$

- Content loss:

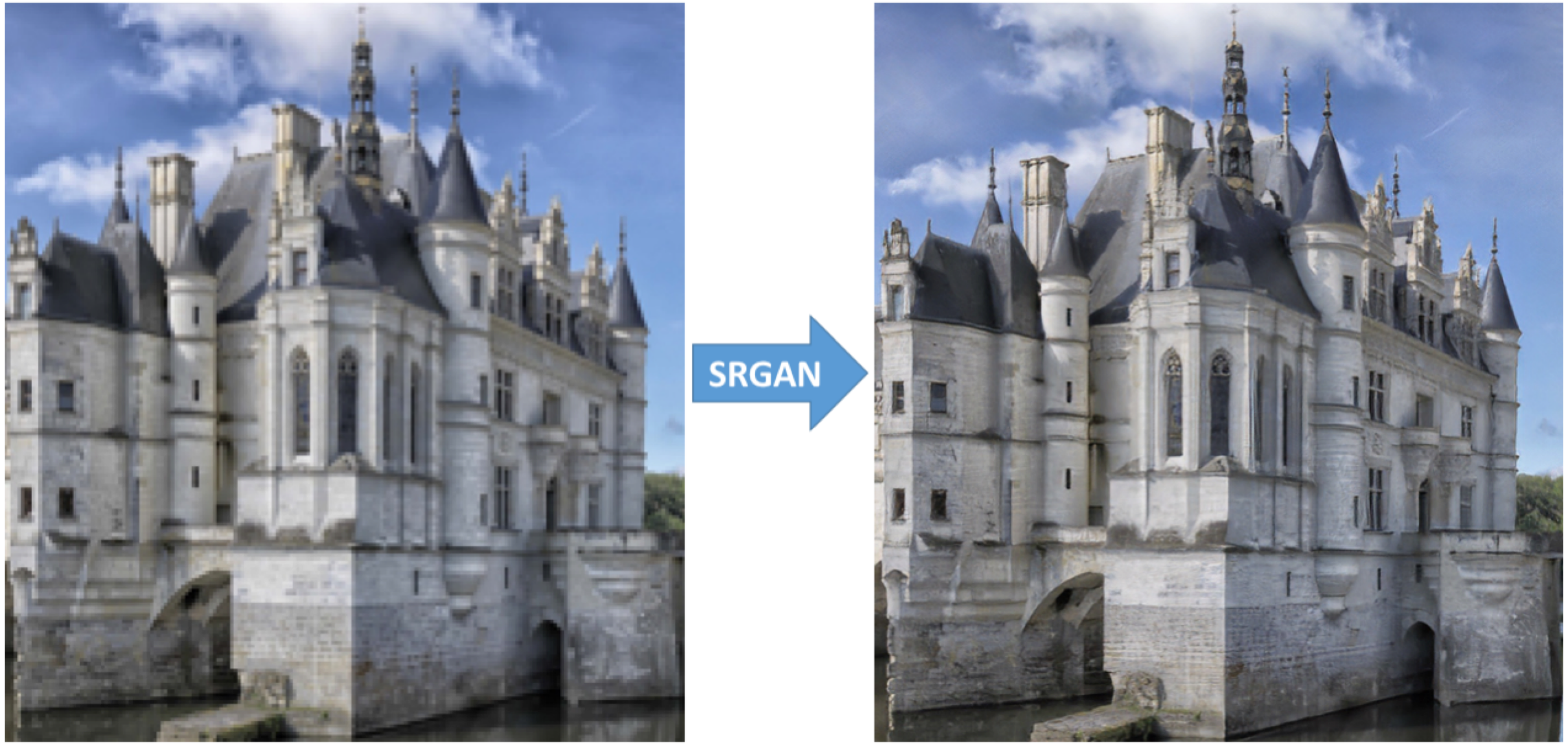
$$l^{SR}_{VGG/i.j} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2$$

$\phi_{i,j}$ **The feature map for the j-th convolution (after activation) before the i-th maxpooling layer.**

SRGAN - Results

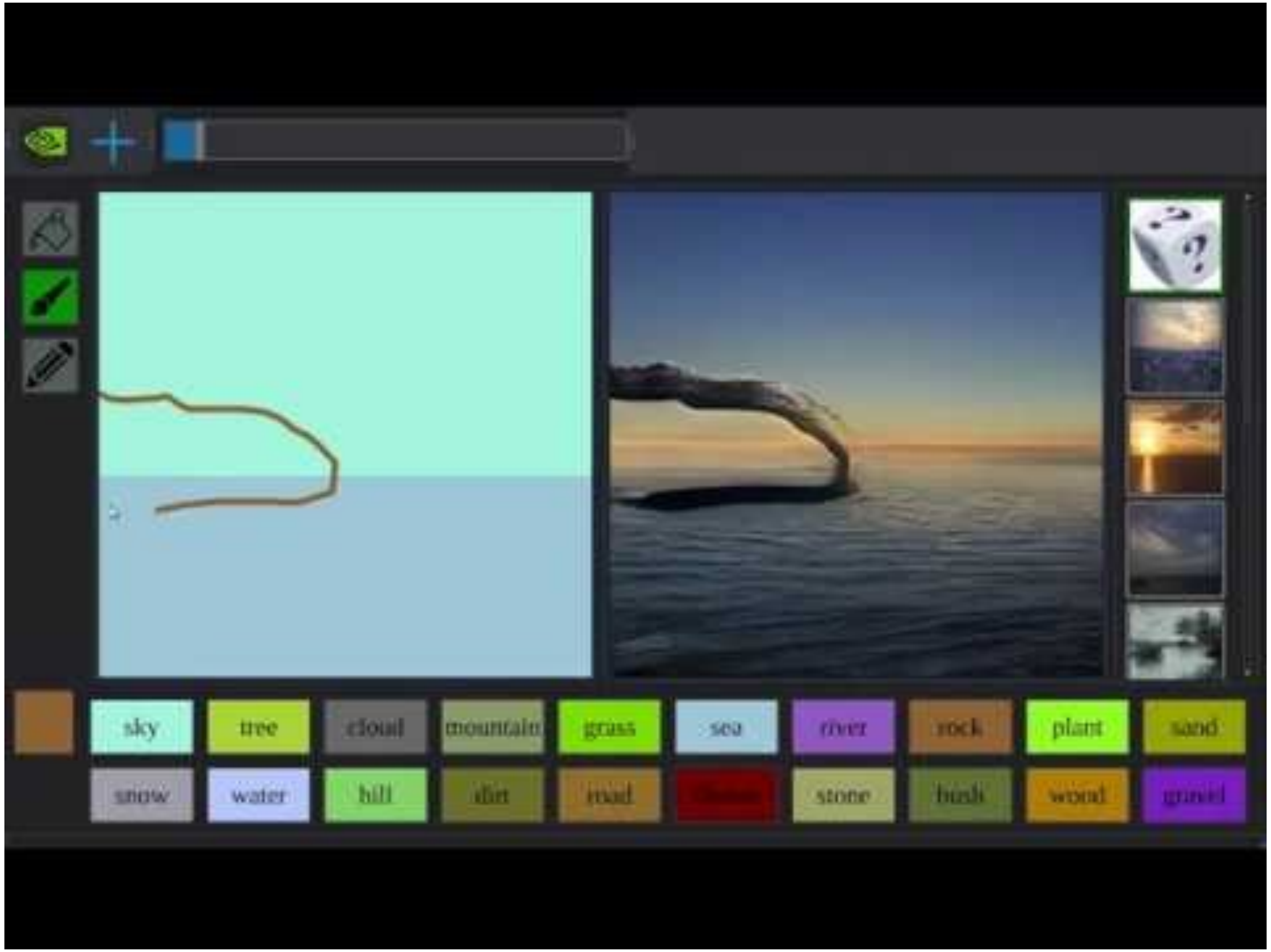


SRGAN - Results



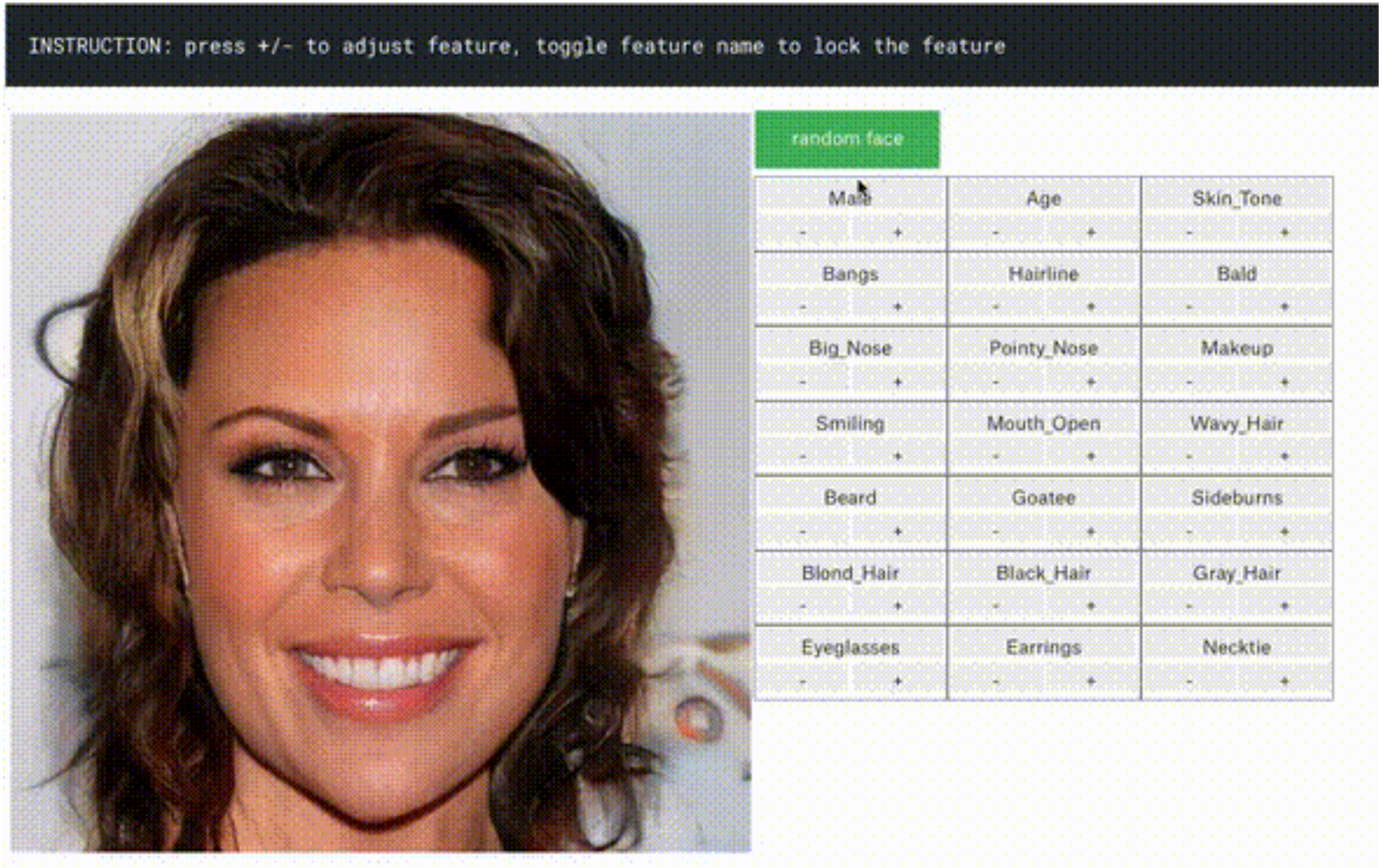
LG Image **Generated Image**

iGAN



Park, Taesung, et al. "Semantic Image Synthesis with Spatially-Adaptive Normalization." *arXiv*19.

Transparent Latent-space GAN (TL-GAN)

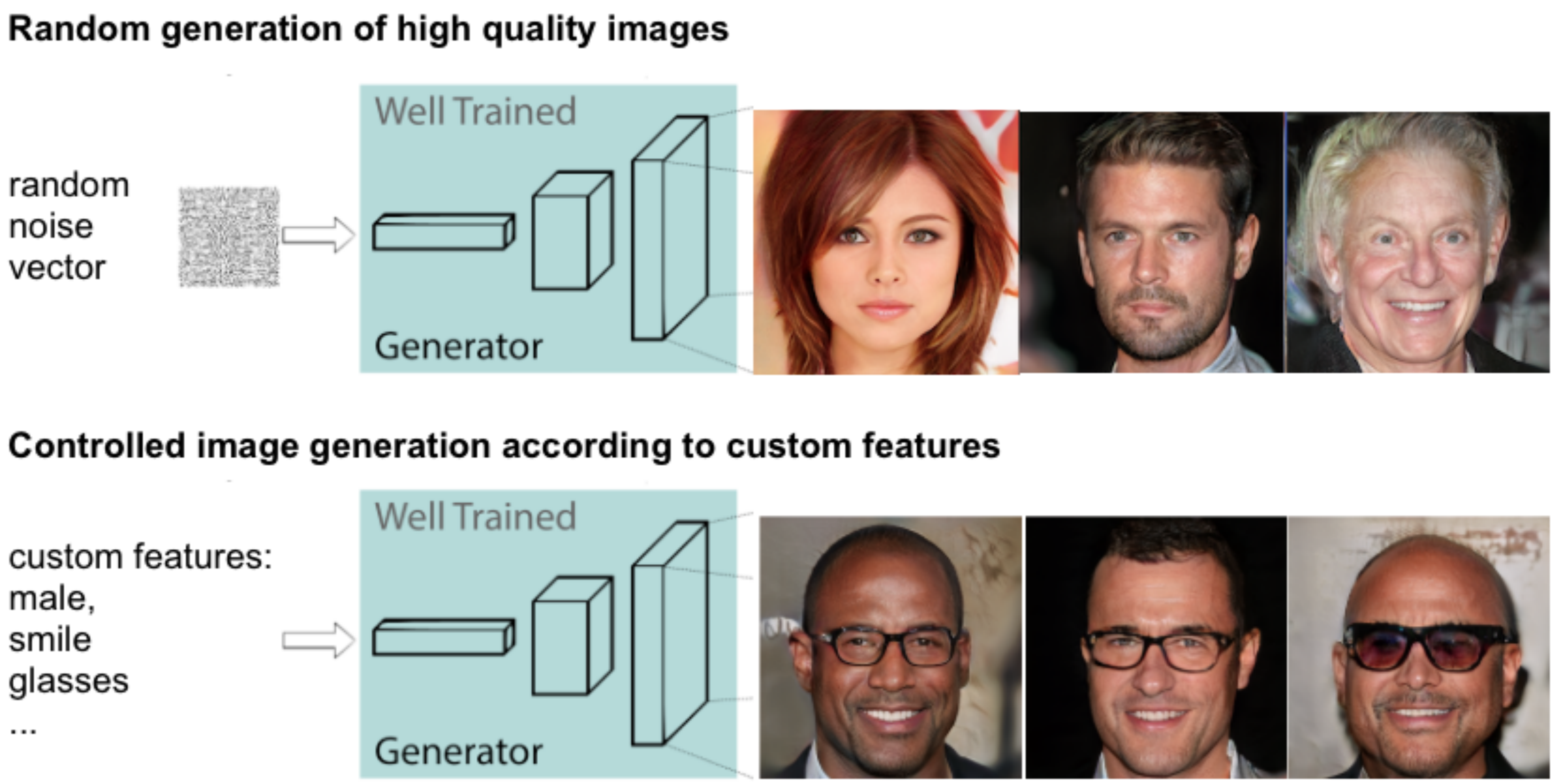


Progressive Growing of GANs

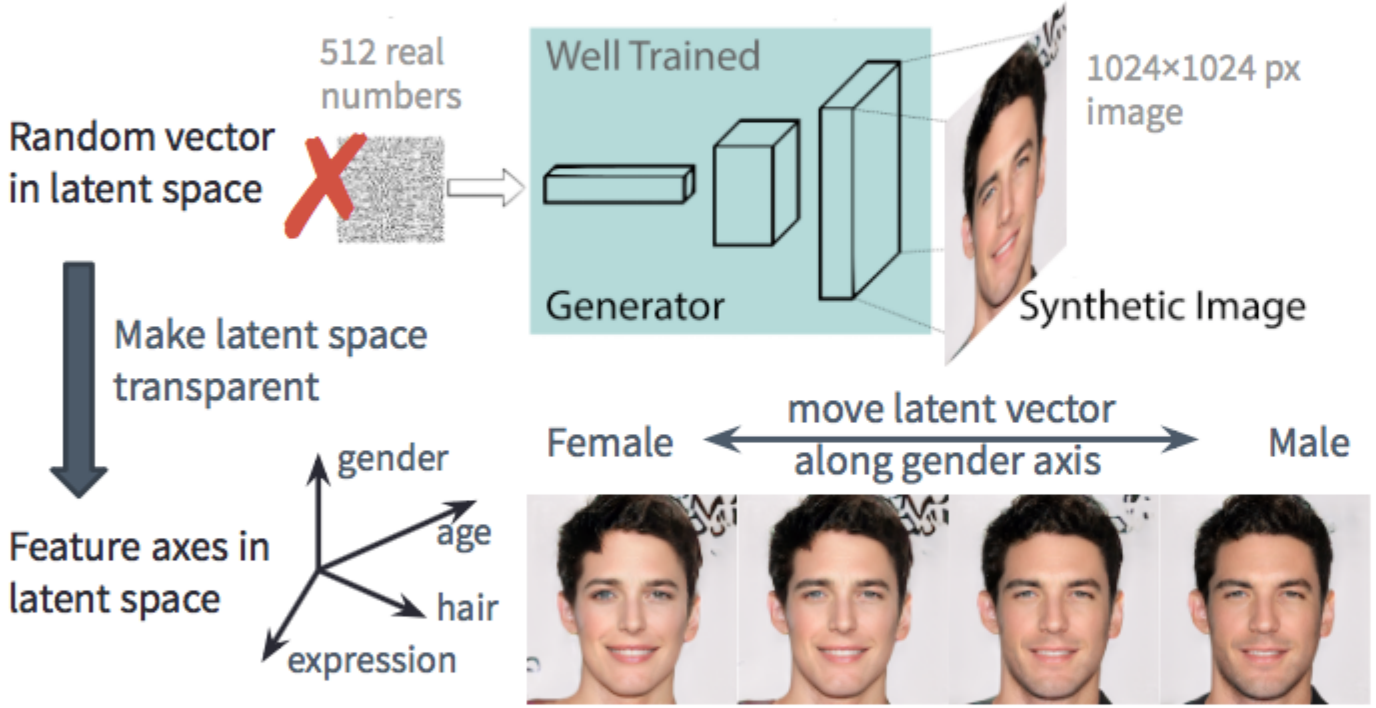
Figure: synthetic images generated by [pg-GAN]() from Nvidia. None of these images are real!

Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation.“ ICLR18

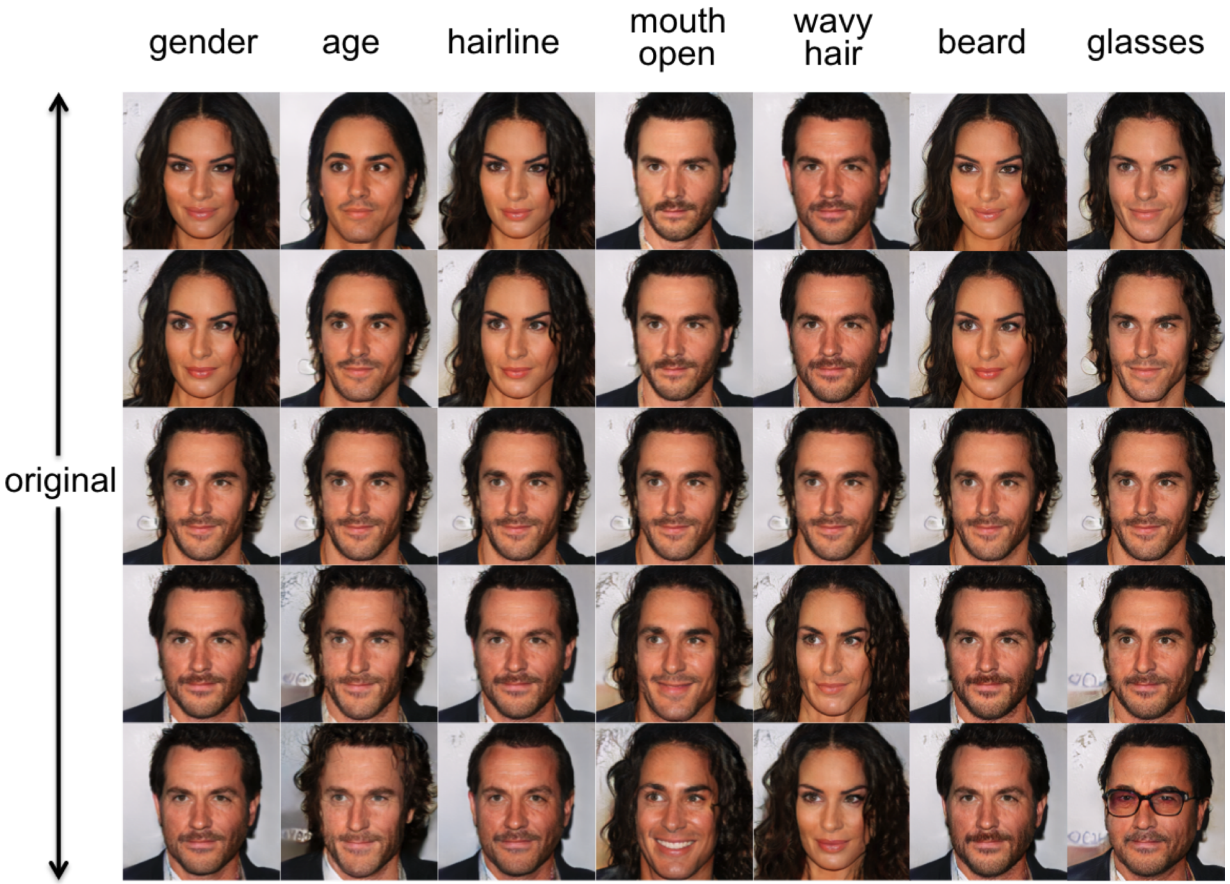
Conditional GAN



TL-GAN


512 real
numbers
Random vector
in latent space
Well Trained
Generator
1024×1024 px
image
Synthetic Image
Make latent space
transparent
Feature axes in
latent space
gender
age
hair
expression
Female
move latent vector
along gender axis
Male

TL-GAN – More results



Pose Guided Person Image Generation

Ma, Liqian, et al. "Pose guided person image generation." *Advances in Neural Information Processing Systems*. 2017.

Pose Guided Person Image Generation

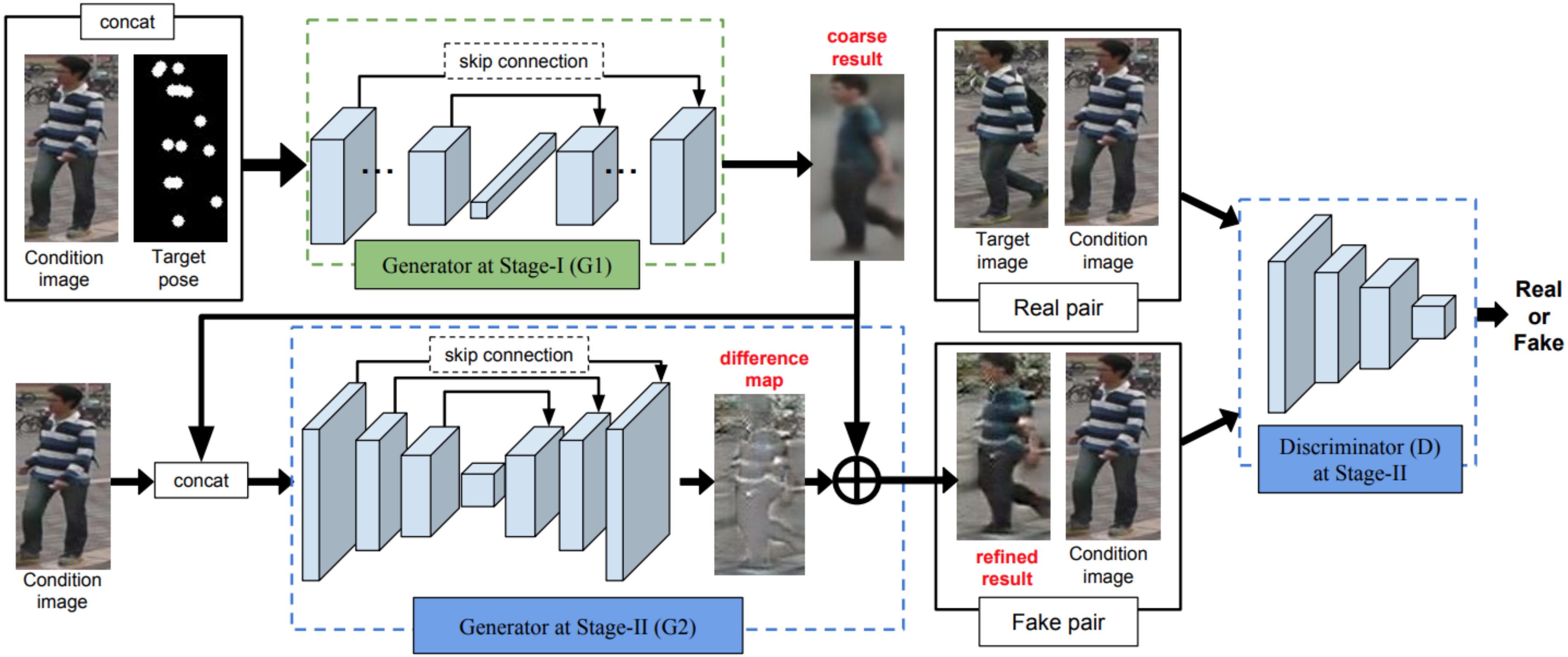
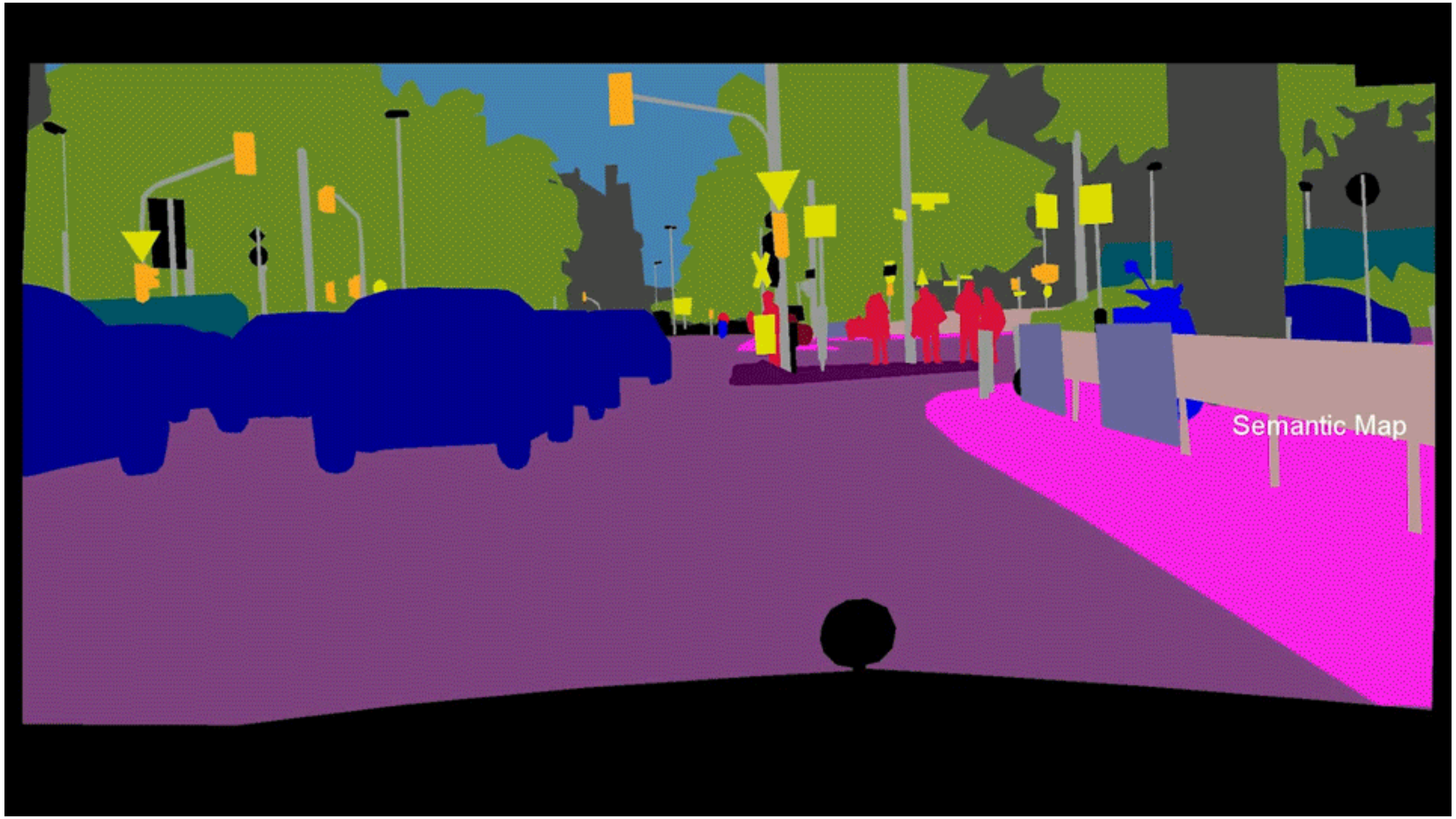


Image to Image Translation



Wang, Ting-Chun, et al. "High-resolution image synthesis and semantic manipulation with conditional gans." *CVPR*18.

Image to Image Translation

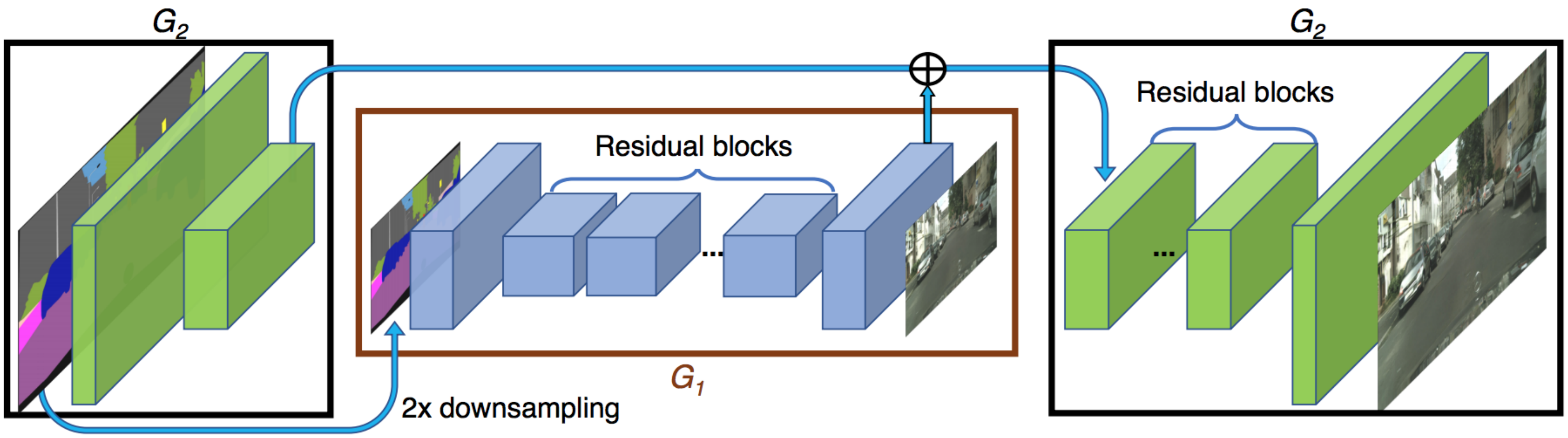
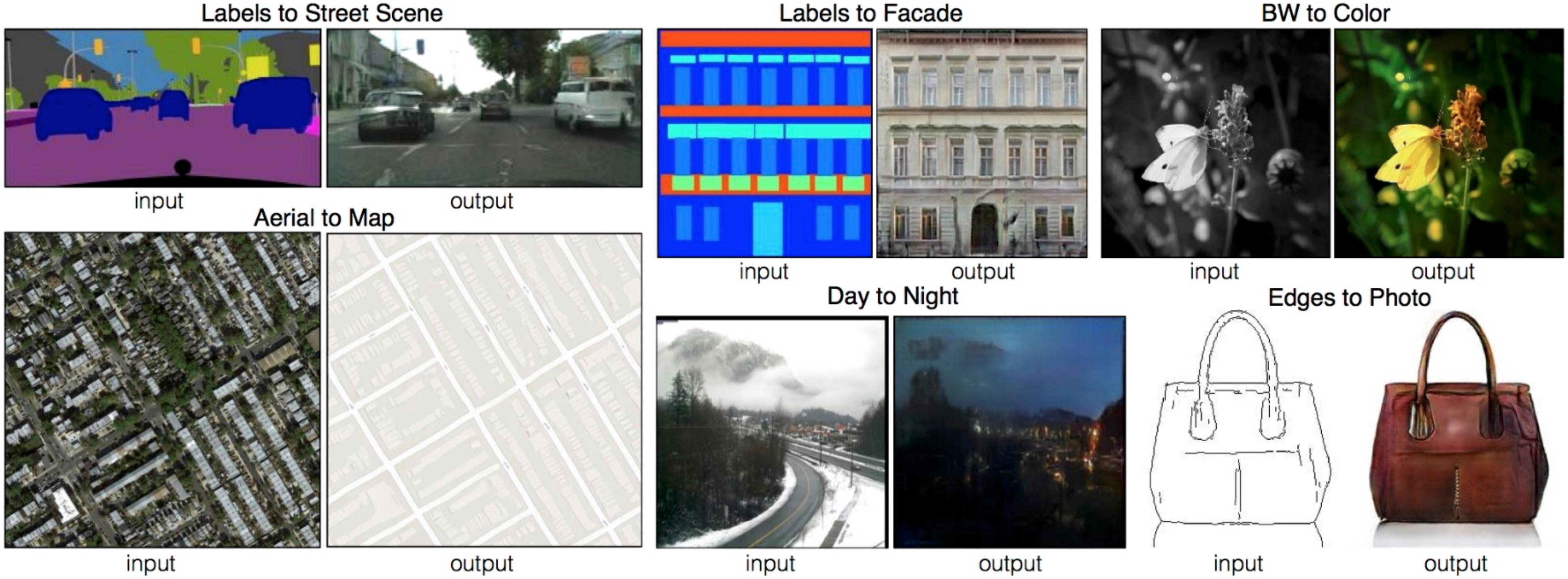


Figure 2: Network architecture of our generator. We first train a residual network G_1 on lower resolution images. Then, another residual network G_2 is appended to G_1 and the two networks are trained jointly on high resolution images. Specifically, the input to the residual blocks in G_2 is the element-wise sum of the feature map from G_2 and the last feature map from G_1.

Image to Image Translation



Text to image (StackGAN)

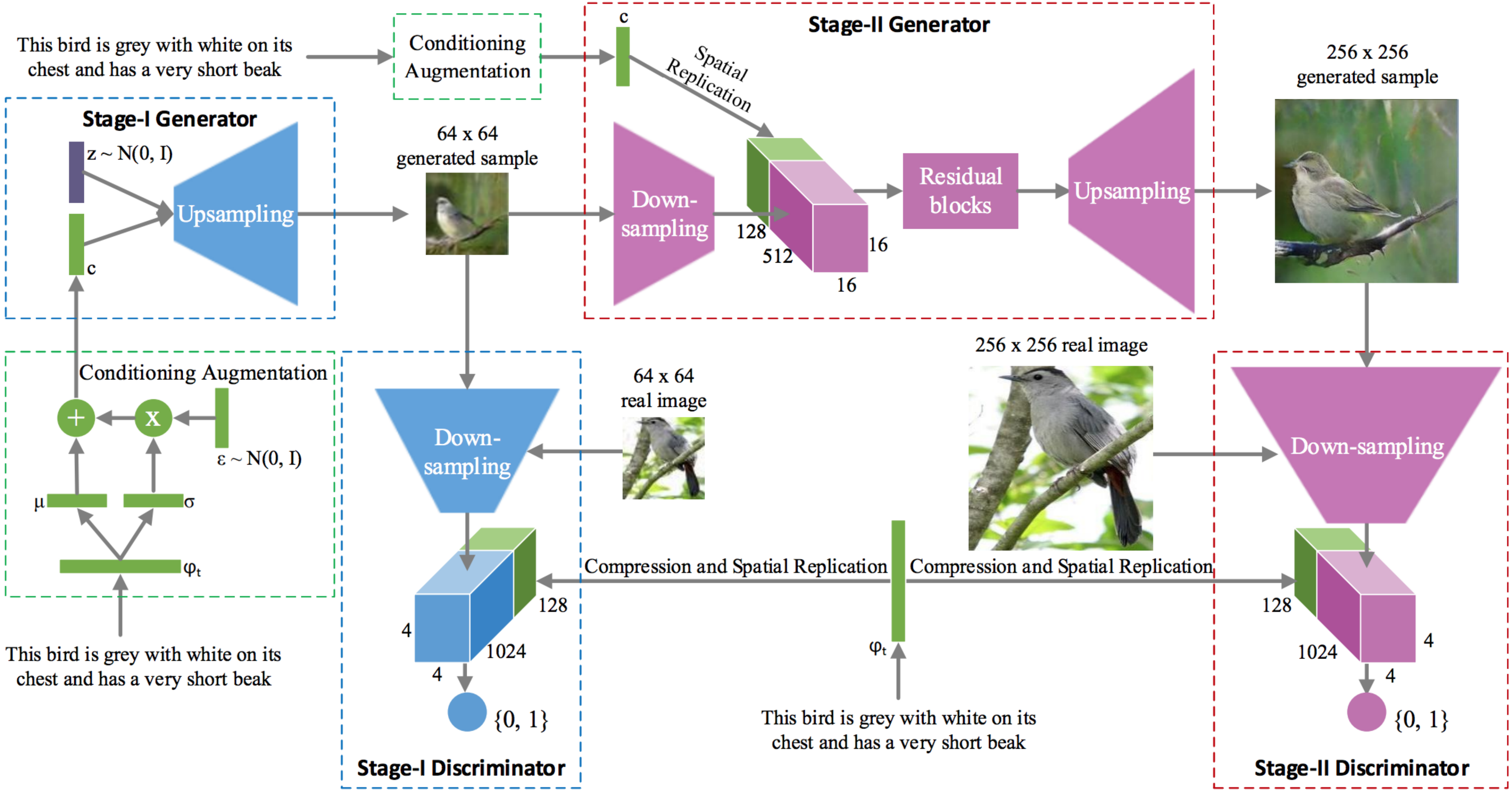


Figure 2. The architecture of the proposed StackGAN. The Stage-I generator draws a low resolution image by sketching rough shape and basic colors of the object from the given text and painting the background from a random noise vector. The Stage-II generator generates a high resolution image with photo-realistic details by conditioning on both the Stage-I result and the text again.

More details and References

- Ian Goodfellow:
  https://www.youtube.com/watch?v=YpdP_0-IEOw
- Radford, (generate voices also here)
  https://www.youtube.com/watch?v=KeJINHjyzOU
- Tips for training GAN: https://github.com/soumith/ganhacks