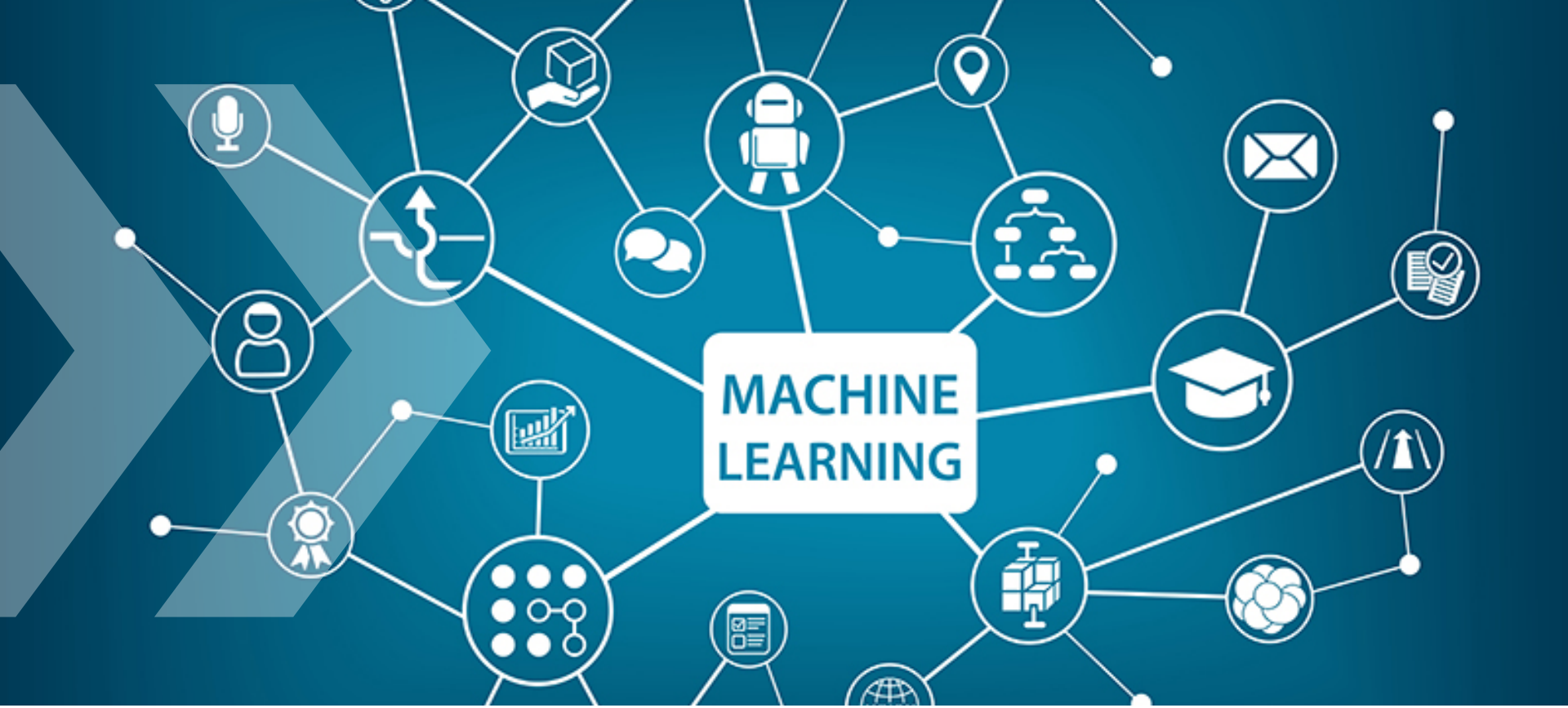


CS489/698: Intro to ML

Lecture 15: Generative Adversarial Networks

UNIVERSITY OF WATERLOO

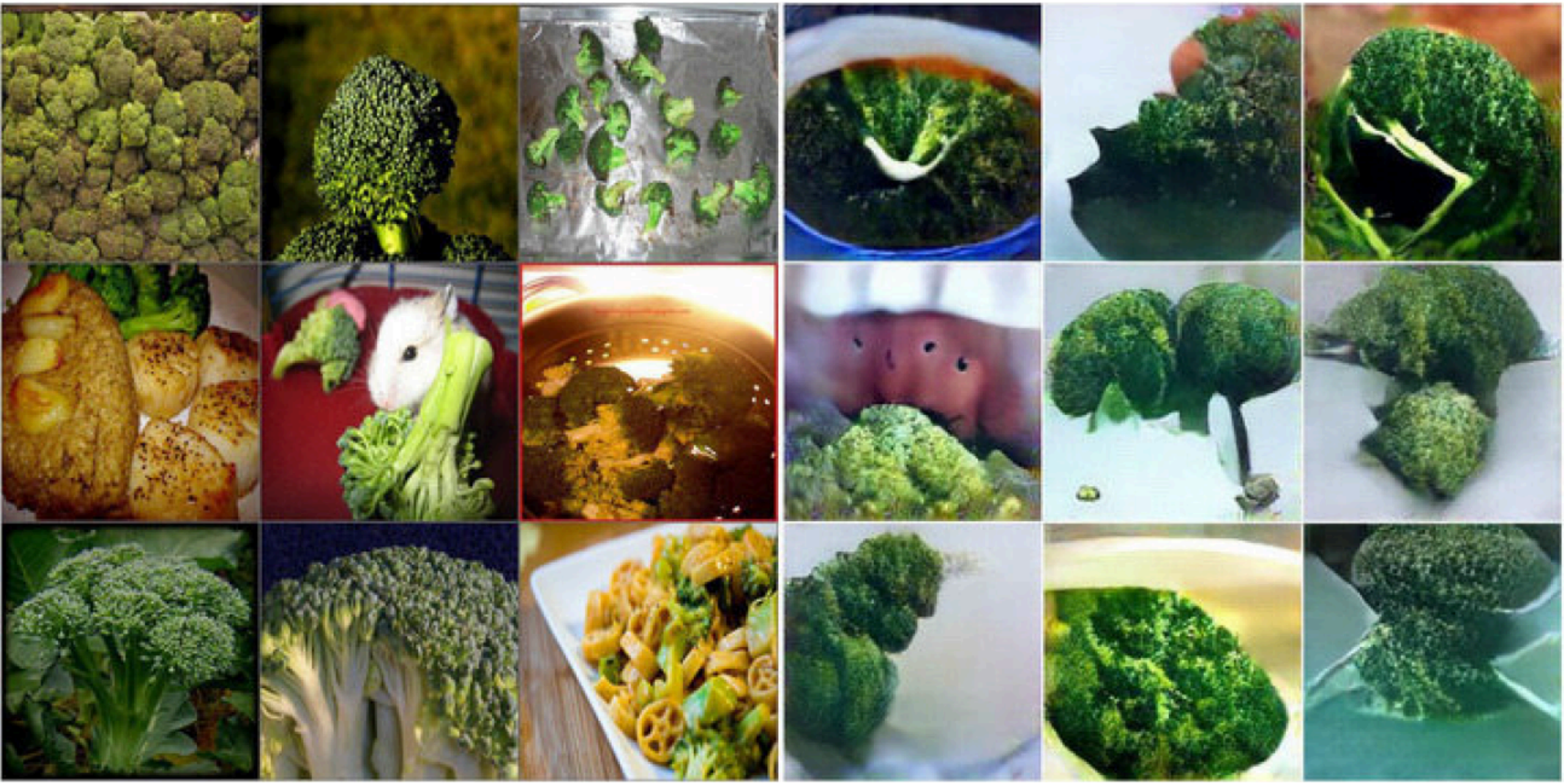
Outline

- Motivation
- Formulation
- Optimization
- Advanced

Some Examples (Nguyen et al, CVPR'17)

Examples cont’

Examples cont’



Generative Models

- Given training data from p_{data}

- Generate new samples from p_{model}

- Want $p_{model} \sim p_{data}$

The "Obvious" Approach

- Use training data to estimate density $p_{model} \sim p_{data}$
  - Which algorithm?
- Then sample from p_{model}
- What might go wrong?
  - Estimating density is hard, very hard…
  - Sampling from high-d density is hard, very hard…

Why Generative Models?

- If we can generate, we must know the objects so well!
  - Feature extraction; pre-training
- Semi-supervised learning
- Planning
- And the cool applications

Image Super-Resolution (Ledig et al, CVPR'17)

bicubic
(21.59dB/0.6423)

SRResNet
(23.53dB/0.7832)

SRGAN
(21.15dB/0.6868)

original

Interactive Image Generation
(Zhu et al, ECCV'16)

UNIVERSITY OF WATERLOO

Neural photo editing (Brock et al, ICLR'17)

Image to Image Translation
(Isola et al, CVPR'17)

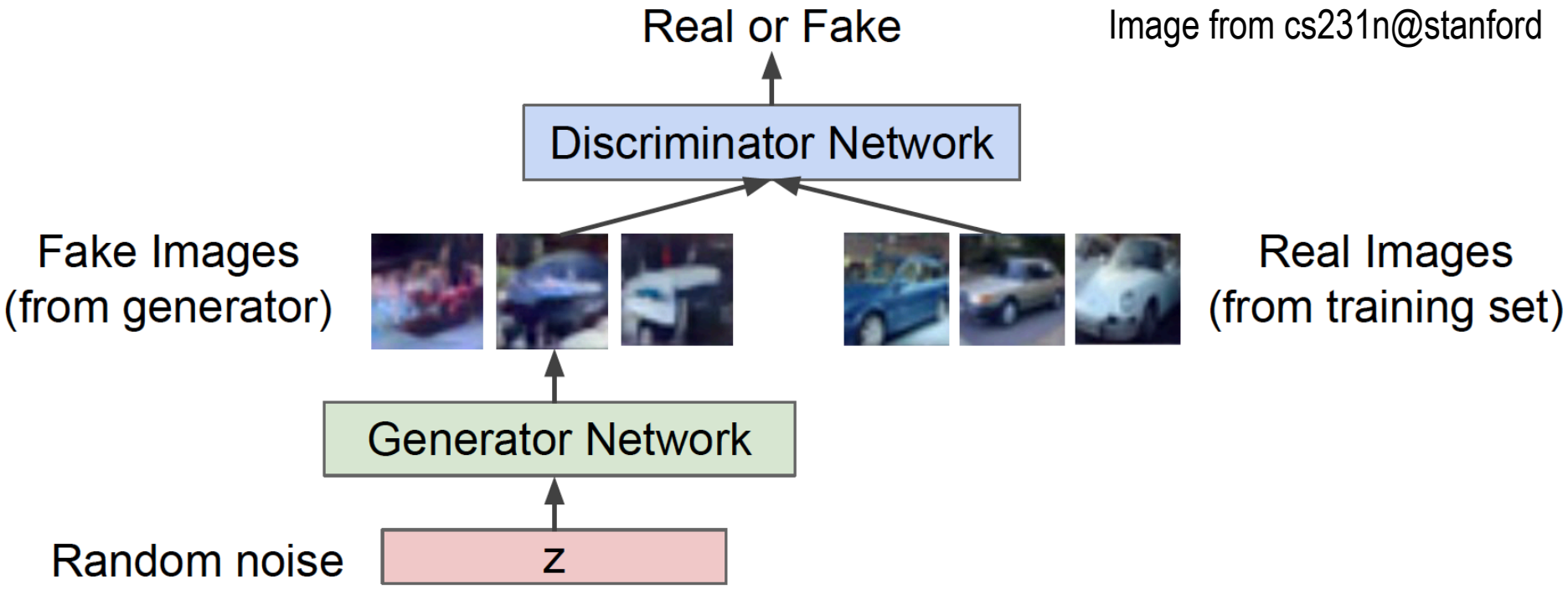
Outline

- Motivation
- Formulation
- Optimization
- Advanced

Generative Adversarial Networks
(Goodfellow et al, NIPS'14)

- A generator proposes samples
- A discriminator examines samples
  - Accept if sample "resembles" training data
  - Reject otherwise
- How can generator "fool" discriminator?

Key Idea



- No density estimation
- Directly generate image from random noise!

Minimax Game

$$\min_G \max_D \; \mathbf{E}_{x \sim p_{\text{data}}}[\log D(x)] + \mathbf{E}_{z \sim p_z}[\log(1 - D(G(z)))]$$

- D(x): probability of x coming from same distribution as training data
- Discriminator D: max prob of training data and min prob of generated sample
- Generator G: max prob of generated sample (fool discriminator)
- Zero-sum game: your loss is my gain

Why Would It Work?

$$\min_G \max_D \; \mathbf{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbf{E}_{z\sim p_z}[\log(1 - D(G(z)))]$$

change of variable

$$\min_G \max_D \; \mathbf{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbf{E}_{x\sim p_{\text{model}}}[\log(1 - D(x))]$$

- Fix G, when is D optimal?
  - $D(x) = p_{data}(x) / [p_{data}(x) + p_{model}(x)]$

$$\min_G \text{JS}(p_{\text{data}} \| p_{\text{model}})$$

- Plug in optimal D, when is G optimal?
  - $p_{model} = p_{data}$

Minimax Theorem

$$\min_G \max_D \; \mathbf{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbf{E}_{x\sim p_{\text{model}}}[\log(1 - D(x))]$$

$$\max_D \min_G \; \mathbf{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbf{E}_{x\sim p_{\text{model}}}[\log(1 - D(x))]$$

- Fix D, when is G optimal?
  - collapsing on modes of D
- Plug in optimal G, when is D optimal?
  - D(x) = 1/2

Outline

- Motivation
- Formulation
- Optimization
- Advanced

Representing D and G

- So far assumed D and G can be arbitrarily "smart"
- In practice, $D(x) = D(x; \theta_d)$ and $G(z; \theta_g)$
  - Each is a DCNN, with θ being the weights
- Run SGD to find best θ_d and θ_g

Algorithm (Goodfellow et al, NIPS'14)

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

end for

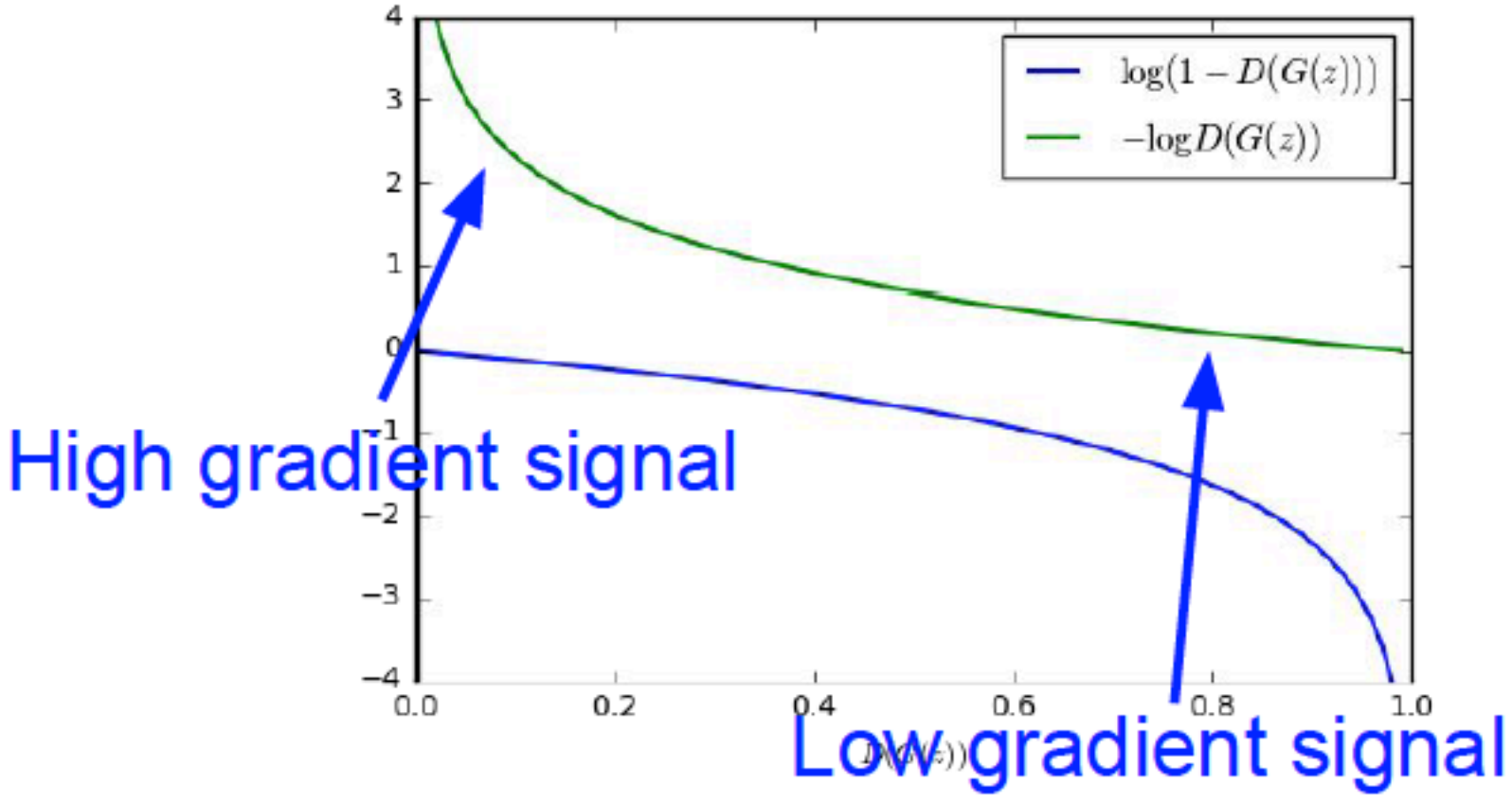
- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

Trick for Saturating Gradient

Image from cs231n@stanford



In practice

$$\max_D \mathbf{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbf{E}_{x\sim p_{\text{model}}}[\log(1-D(x))]$$

~~$$\min_G \mathbf{E}_{x\sim p_{\text{model}}}[\log(1-D(x))]$$~~

$$\min_G \mathbf{E}_{x\sim p_{\text{model}}}[-\log D(x)]$$

$$\min_G \max_D \mathbf{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbf{E}_{x\sim p_{\text{model}}}[-\log D(x)]$$

Fix G, what is the optimal D?

Plug in optimal D, what is the optimal G?

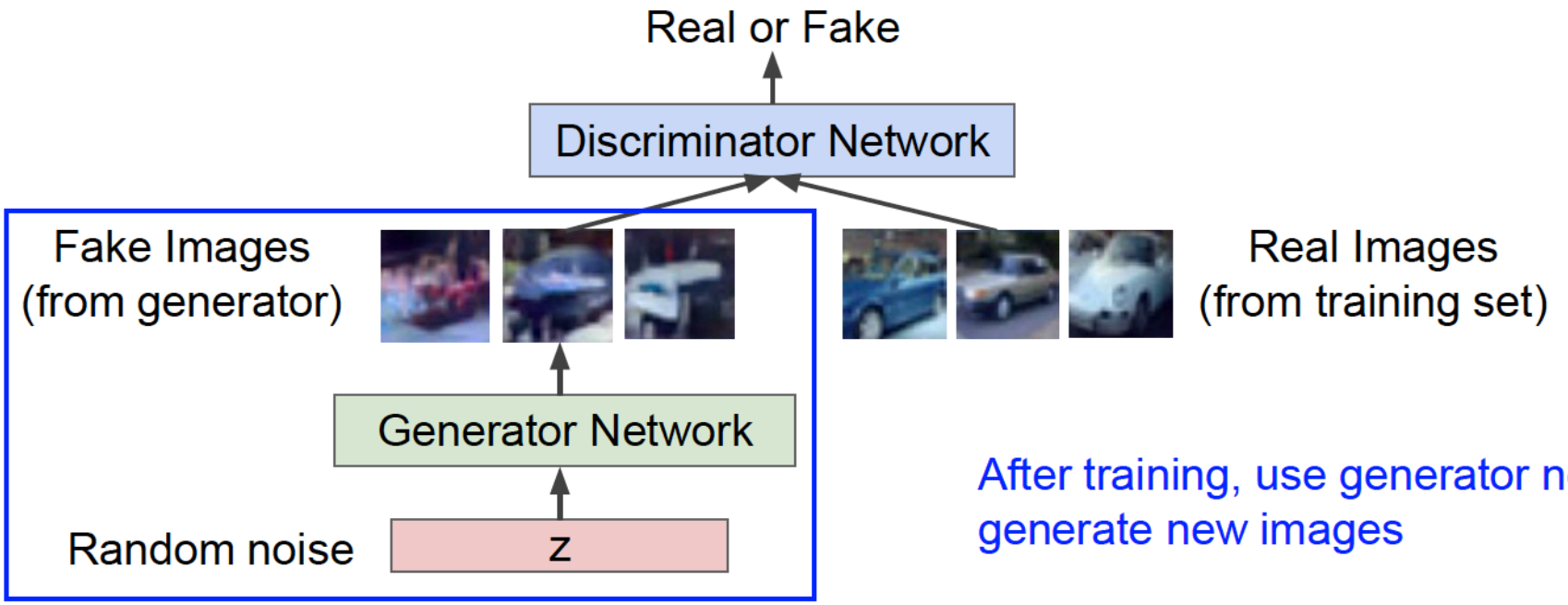
$$\max_D \min_G \mathbf{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbf{E}_{x\sim p_{\text{model}}}[-\log D(x)]$$

Fix D, what is the optimal G?

Plug in optimal G, what is the optimal D?

Testing

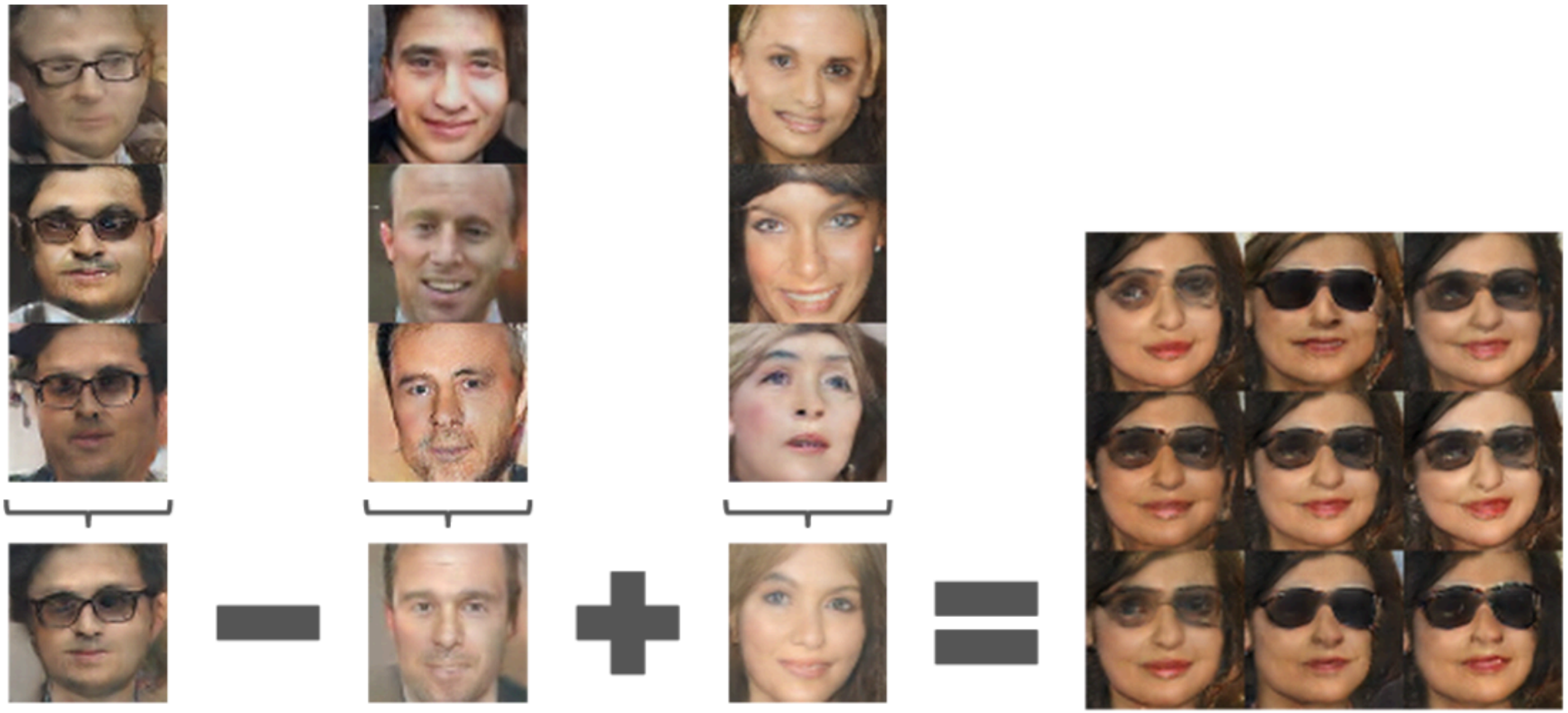
Image from cs231n@stanford



After training, use generator network to generate new images

More Intriguing Examples
(Radford et al, ICLR'16)

More Intriguing Examples cont'
(Radford et al, ICLR'16)



Outline

- Motivation
- Formulation
- Optimization
- Advanced

Wasserstein GAN (Arjovsky et al, ICML'17)

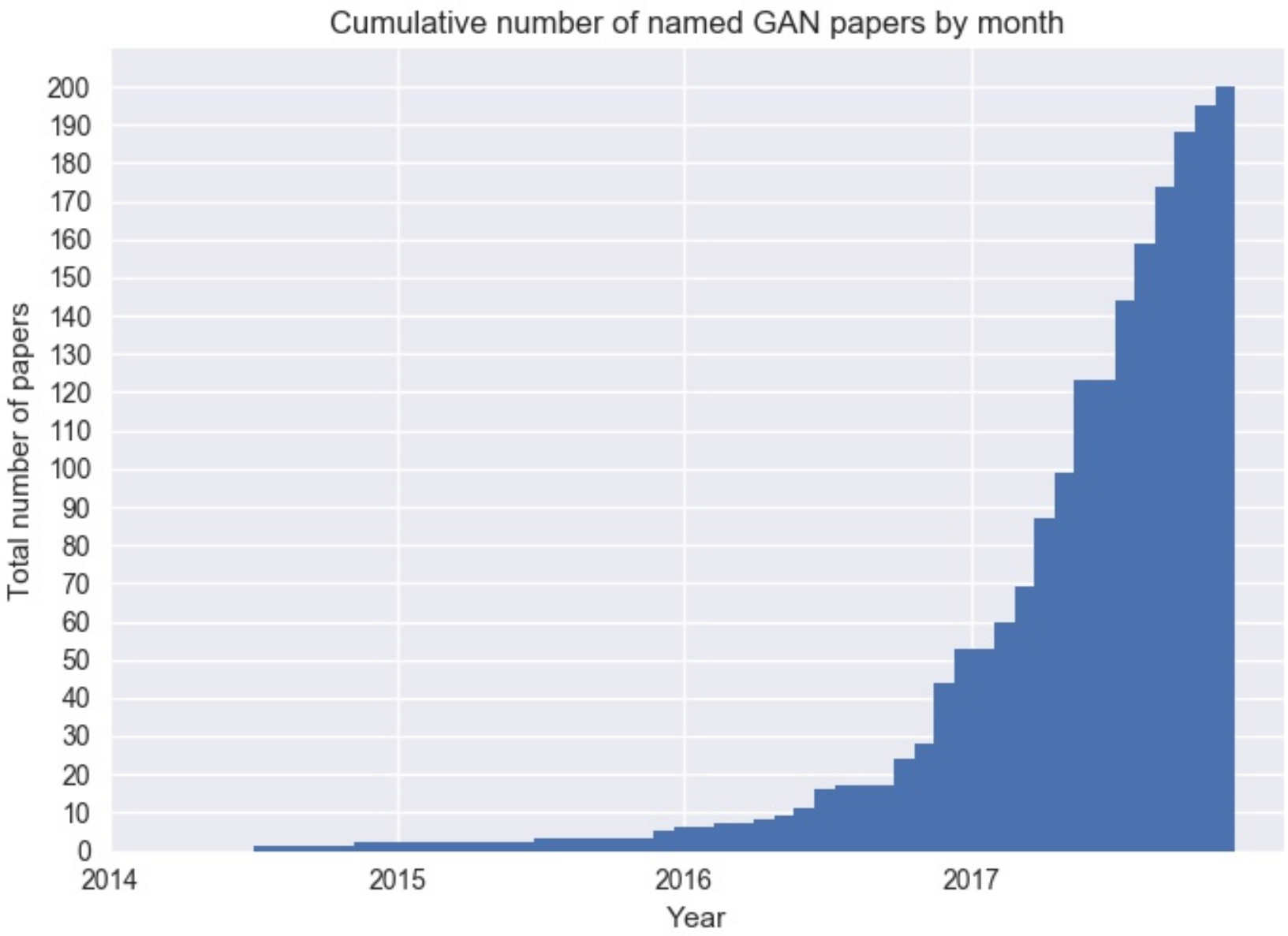
- GAN objective
  - Could be discontinuous

$$\min_G \mathrm{JS}(p_{\mathrm{data}} \| p_{\mathrm{model}})$$

- Better objective
  - Much better behaved
  - Connection to kernels

$$\min_G \mathrm{W}(p_{\mathrm{data}}, p_{\mathrm{model}})$$

The GAN Zoo (https://github.com/hindupuravinash/the-gan-zoo)



Questions?