Generative Adversarial Networks

Ian J. Goodfellow, Jean Pouget-Abadie,[*] Mehdi Mirza, Bing Xu, David Warde-Farley,
Sherjil Ozair,[†] Aaron Courville, Yoshua Bengio[‡]
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Academic Seminar – Paper Presentation

Presenter: Khue Luu

Content

- Creator of Generative Adversarial Networks (GAN)
- Generative vs Discriminative Models
- What is GAN?
- Overview of GAN structure
- GAN Training
- Value function
- Alternating loss function
- GAN – proof of optimality
- GAN - advantages and disadvantages
- GAN evolutions and applications
- Companies using GAN

Creator of GAN

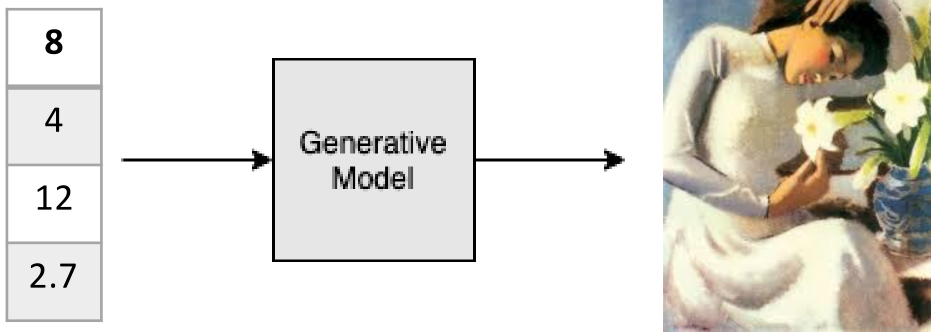
Ian Goodfellow

- Director of Machine Learning in the Special Projects Group at Apple.
- Research scientist at Google Brain
- Lead author of the textbook *Deep Learning*
- Listed as one of the *Innovators Under 35* by MIT Technology Reviews
- Invented GAN in 2014

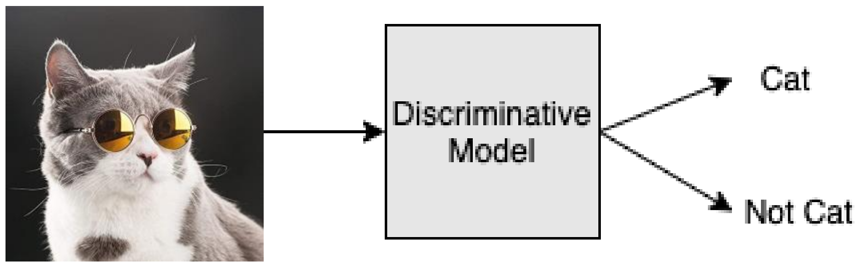
Generative vs Discriminative models

Generative models
learn to produce realistic examples

Random noise



Discriminative models
distinguish between classes



Noise, Class → Features

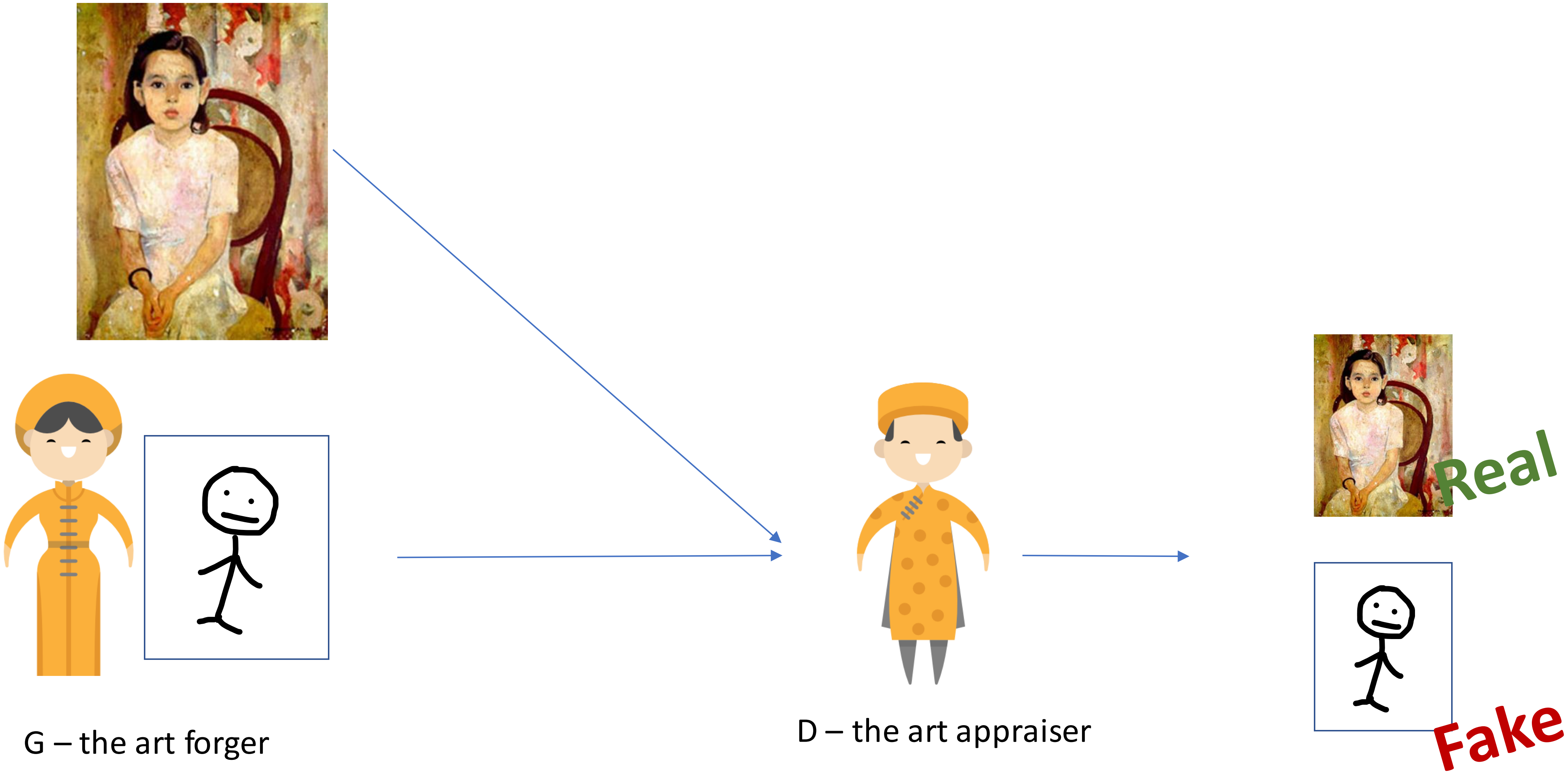
$\xi \quad Y \longrightarrow X$

$$P(X|Y)$$

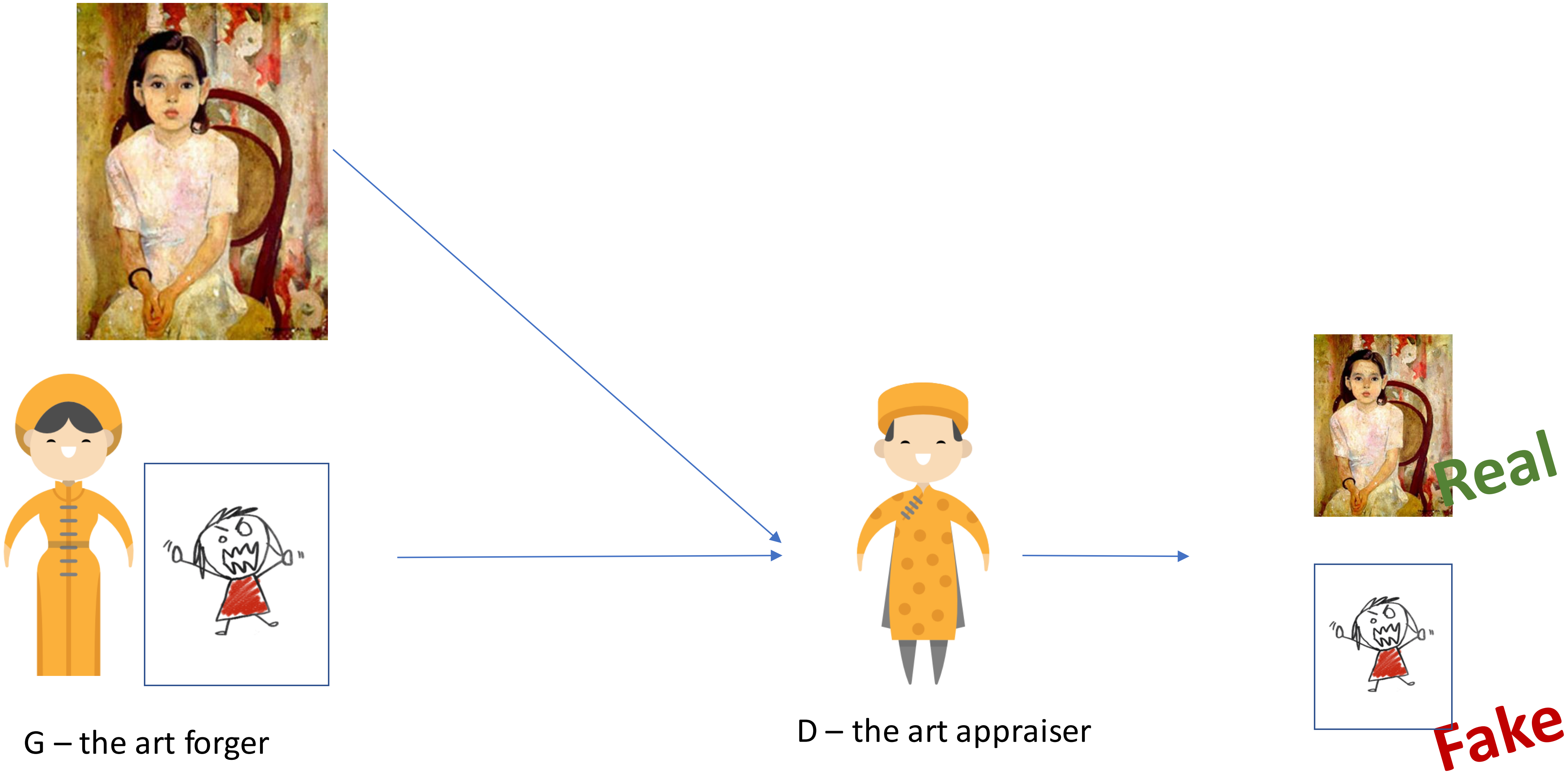
Features → Class

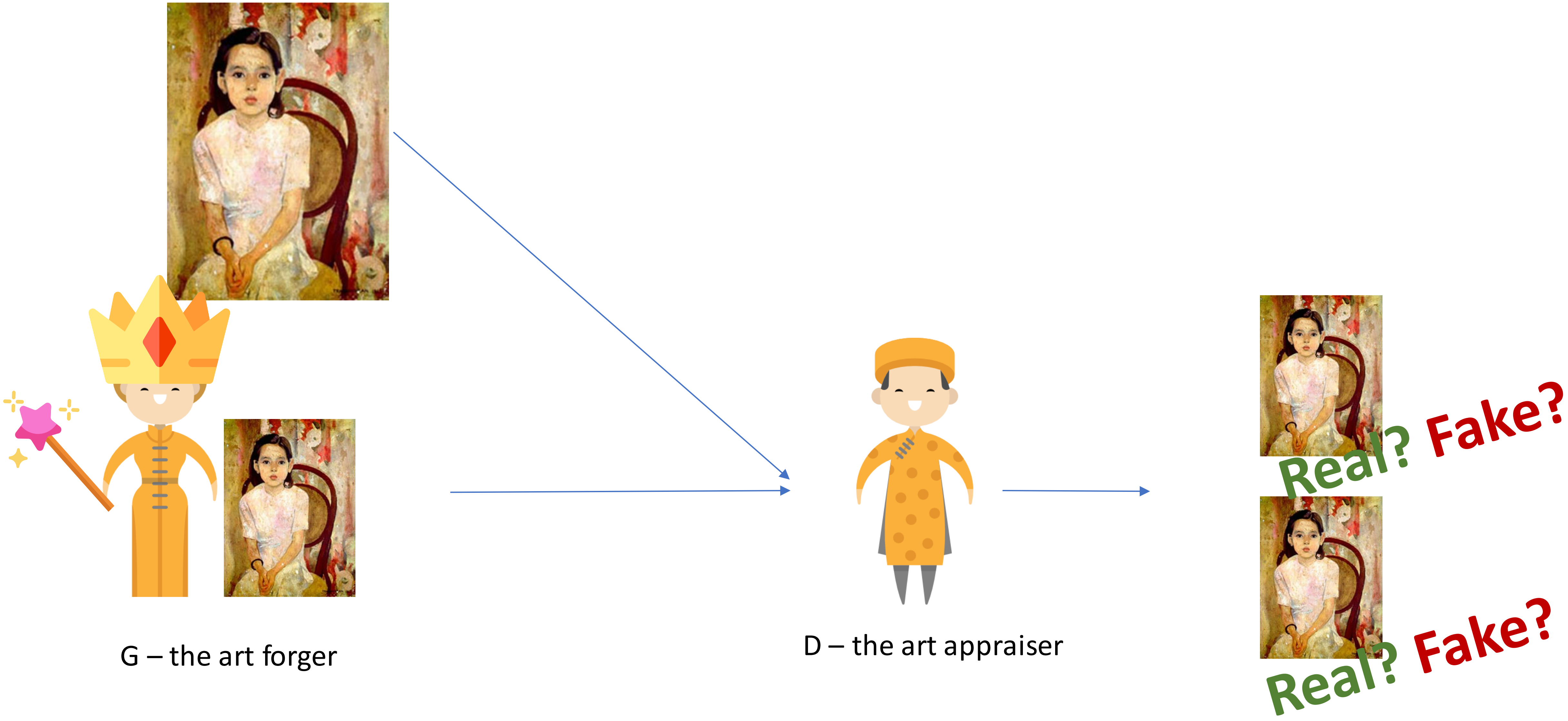
$X \longrightarrow Y$

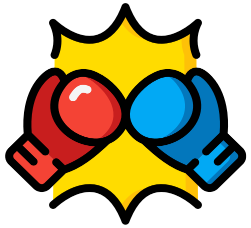
$$P(Y|X)$$



G – the art forger
D – the art appraiser
Real
Fake


G – the art forger
D – the art appraiser
Real
Fake


G – the art forger
D – the art appraiser
Real? Fake?
Real? Fake?

Generator  Discriminator

What is GAN?

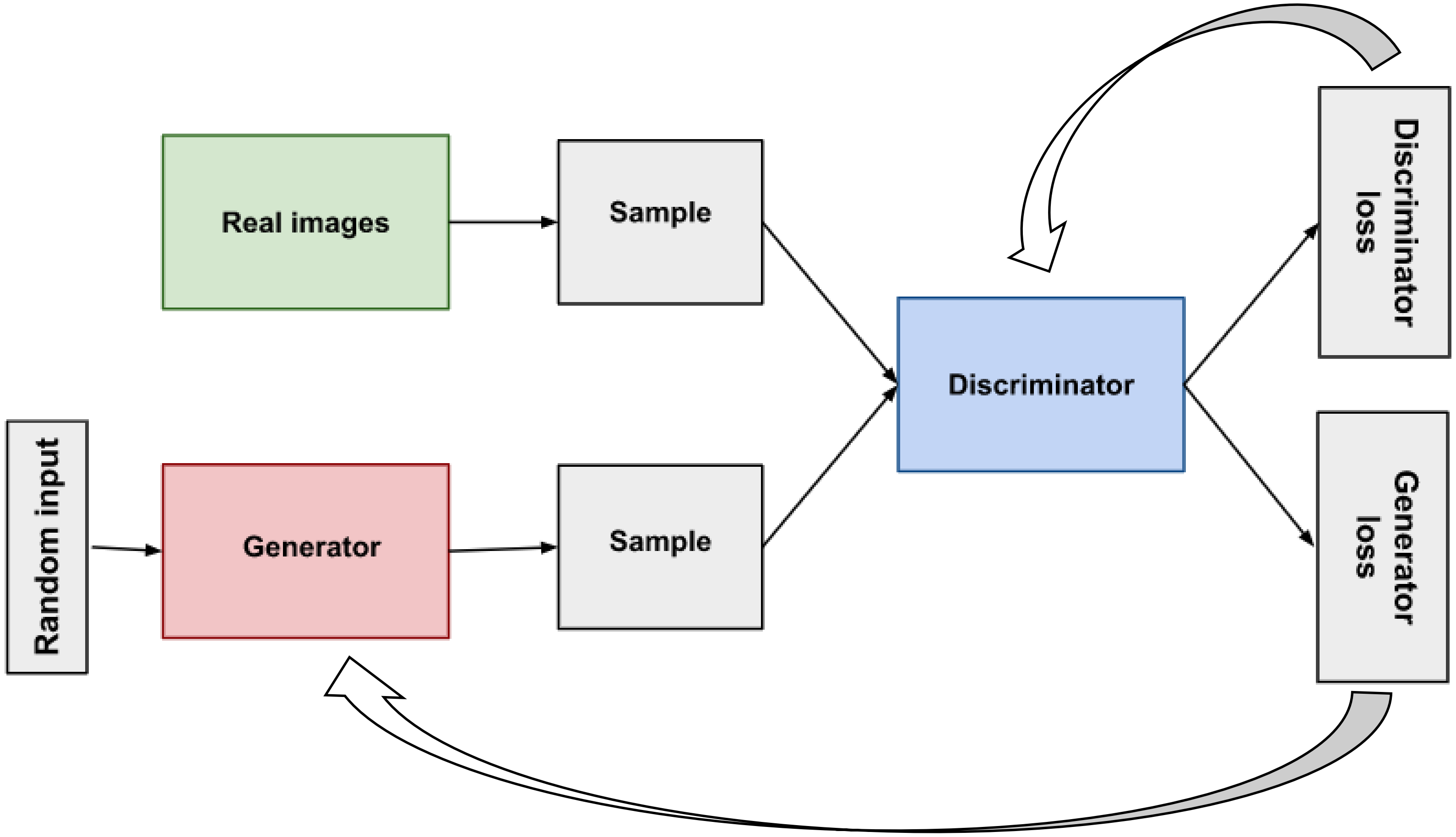
- GANs are composed of two models that compete with each other and reach a point where realistic examples are produced by the generator.
- The generator learns to make fake look real
- The discriminator learns to distinguish real from fake.

REAL

FAKE

Overview of GAN structure

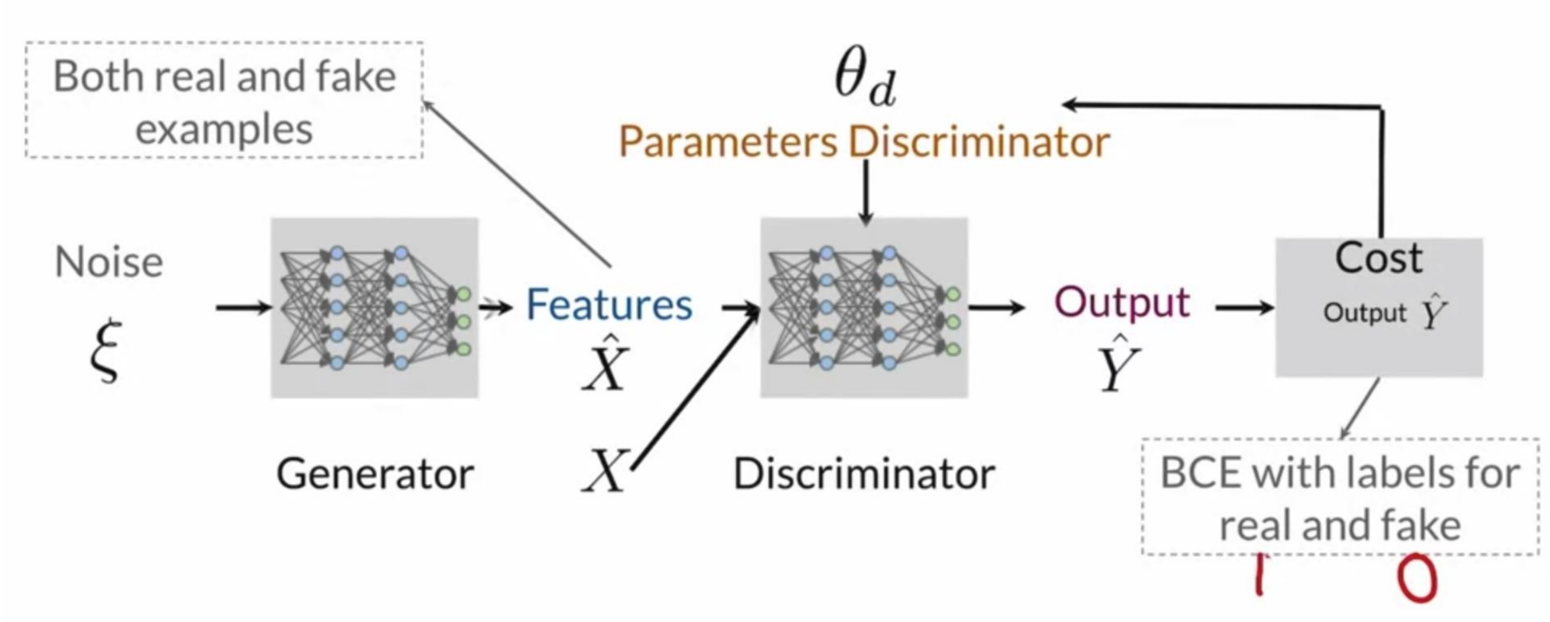
- Both the generator and the discriminator are neural networks.
- The generator output is connected directly to the discriminator input.
- Through backpropagation, the discriminator's classification provides a signal that the generator uses to update its weights.



https://developers.google.com/machine-learning/gan/gan_structure

GAN Training – Discriminator Training

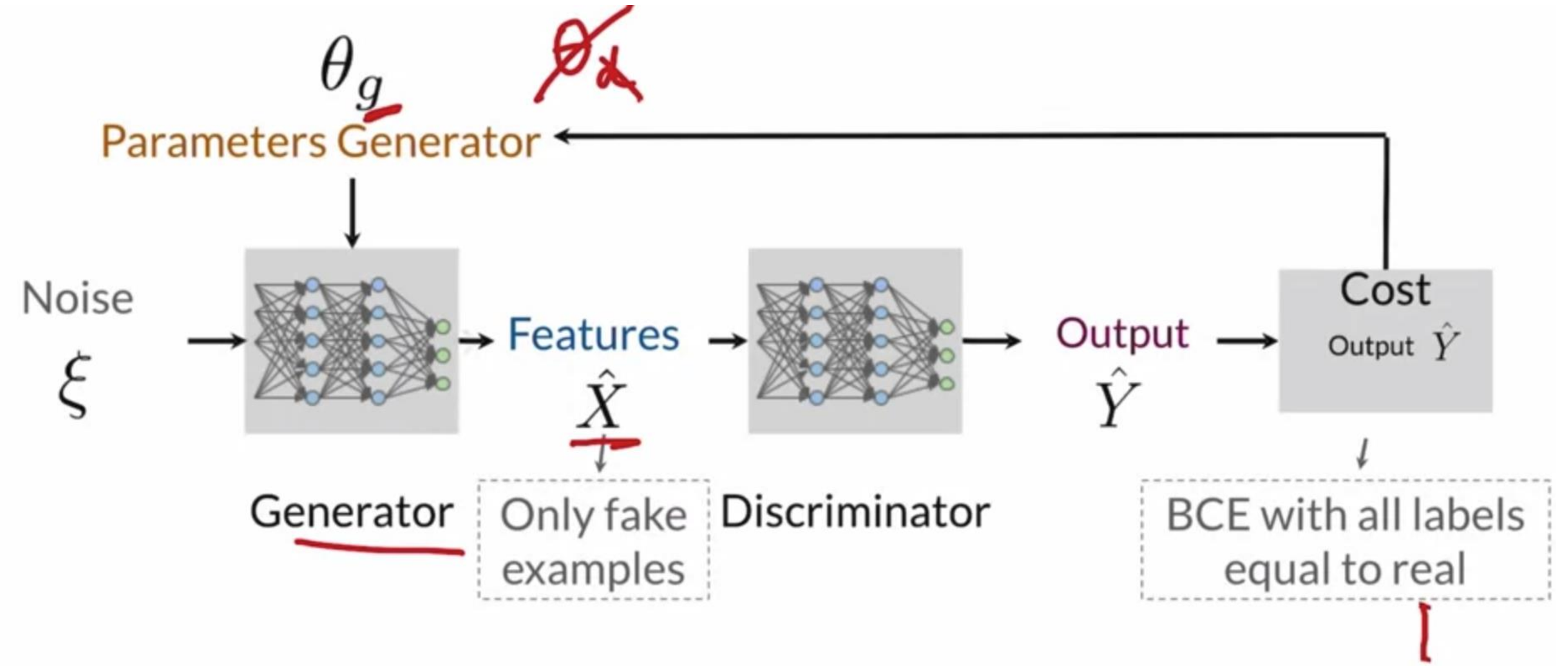
1. The discriminator classifies both real data and fake data from the generator.
2. The discriminator loss penalizes the discriminator for misclassifying a real instance as fake or a fake instance as real.
3. The discriminator updates its weights
   through backpropagation from the discriminator loss through the discriminator network.



Sharon Zhou "GAN for good" Coursera

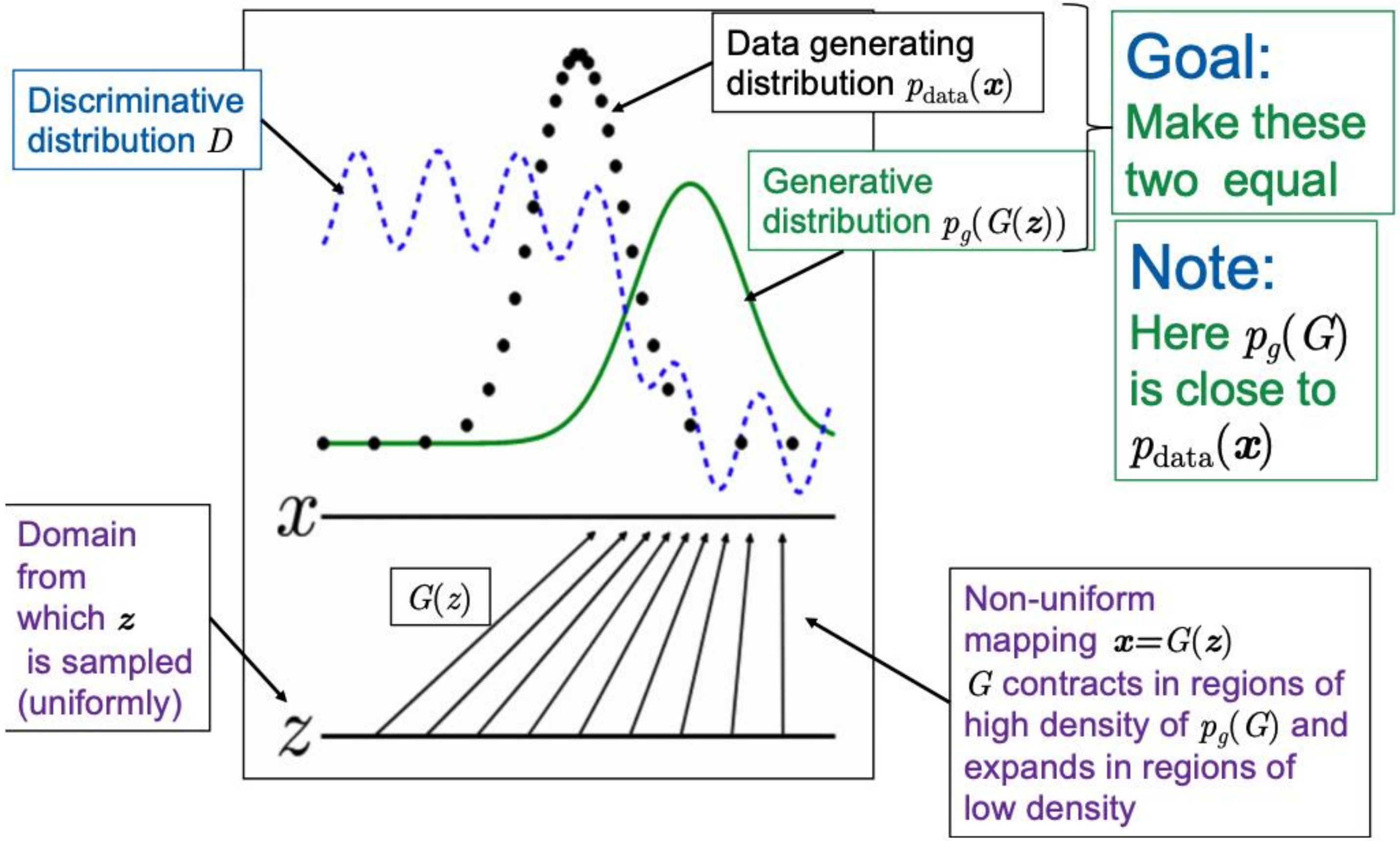
GAN Training – Generator Training

1. Sample random noise.
2. Produce generator output from sampled random noise.
3. Get discriminator "Real" or "Fake" classification for generator output.
4. Calculate loss from discriminator classification.
5. Backpropagate through both the discriminator and generator to obtain gradients.
6. Use gradients to change only the generator weights.



Sharon Zhou "GAN for good" Coursera

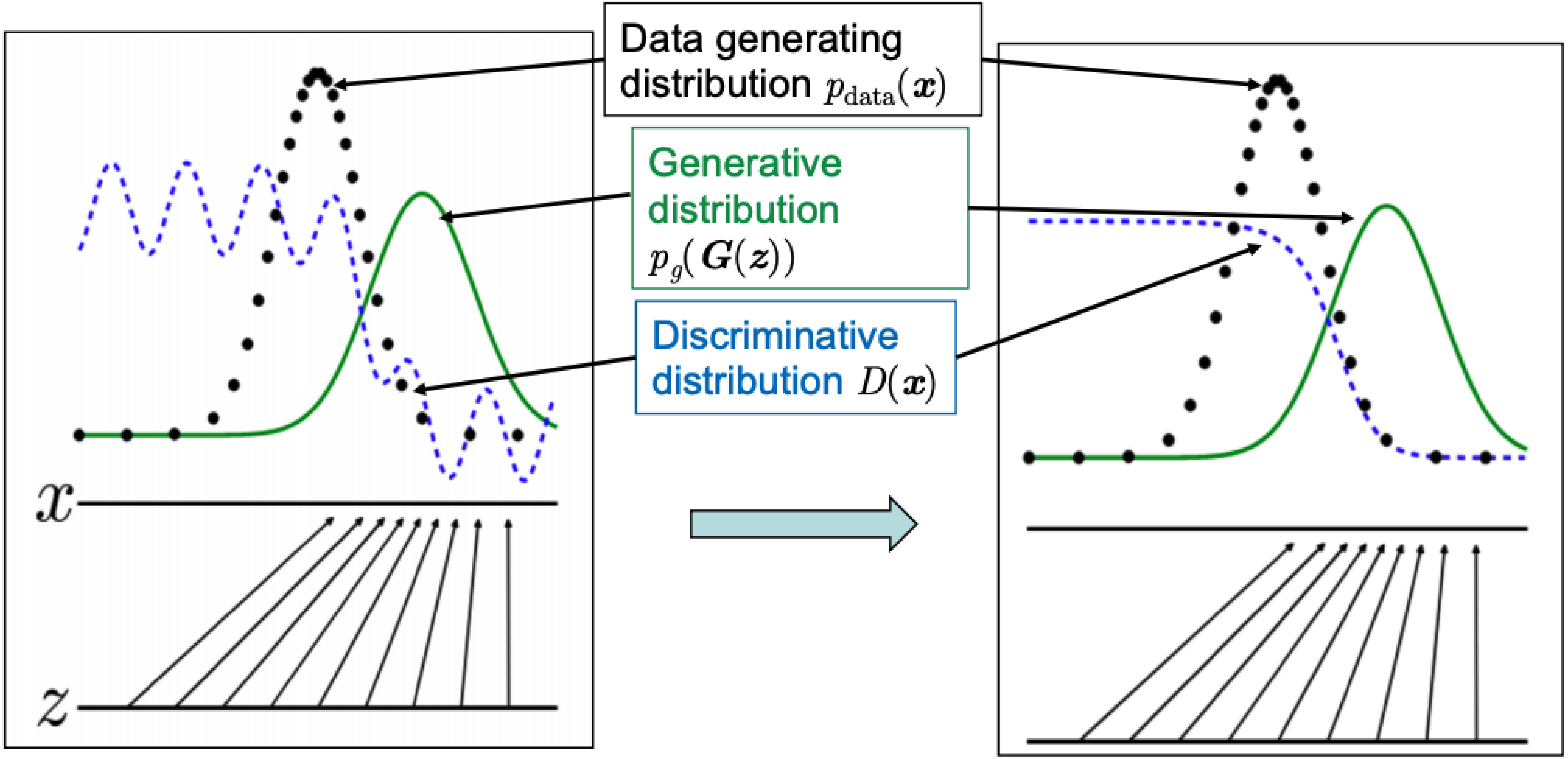
GAN Training - Intuition



https://cedar.buffalo.edu/~srihari/CSE676/22.2-GAN%20Theory.pdf

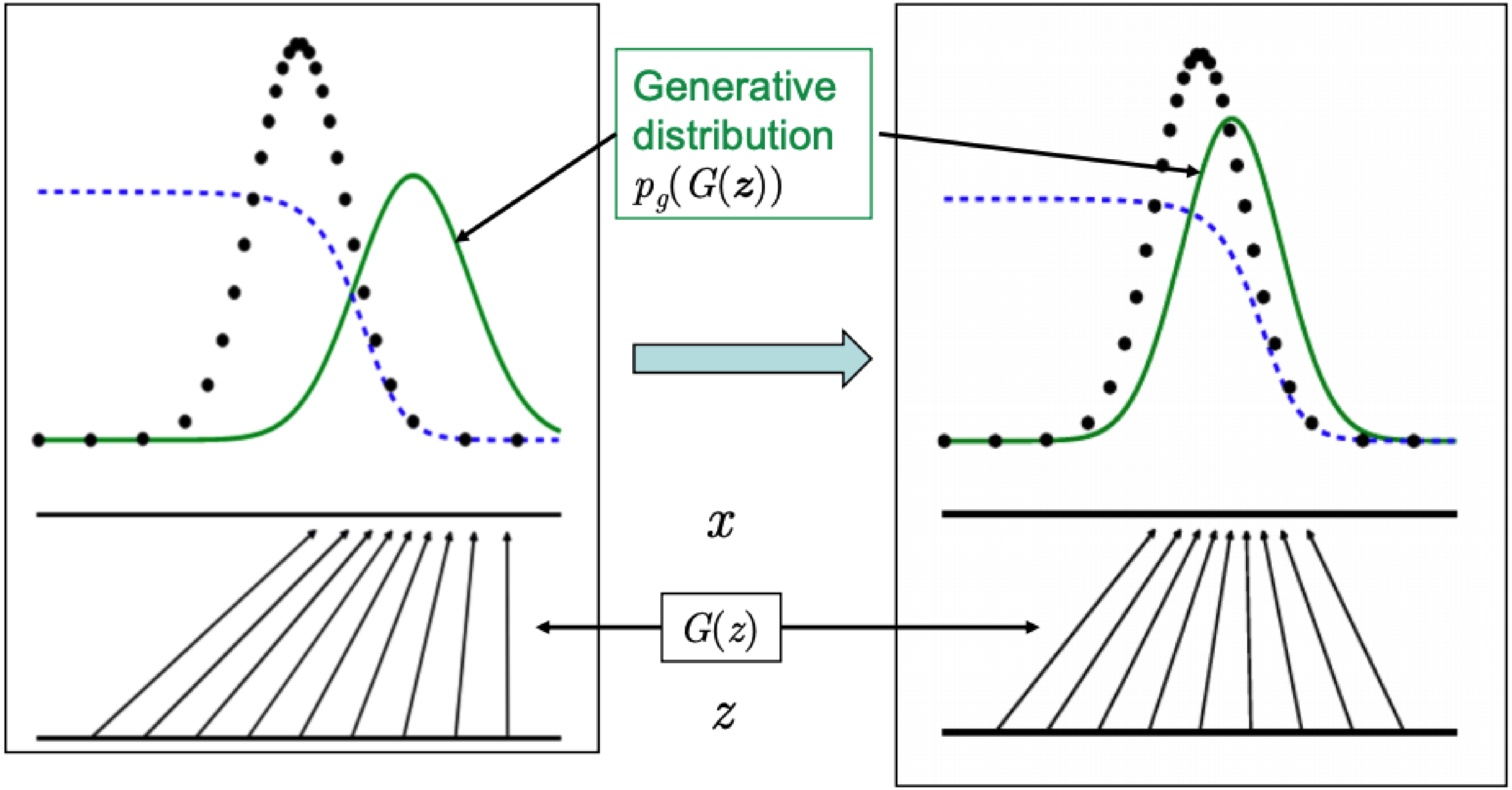
GAN Training - Intuition

$$D^*(\boldsymbol{x}) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$



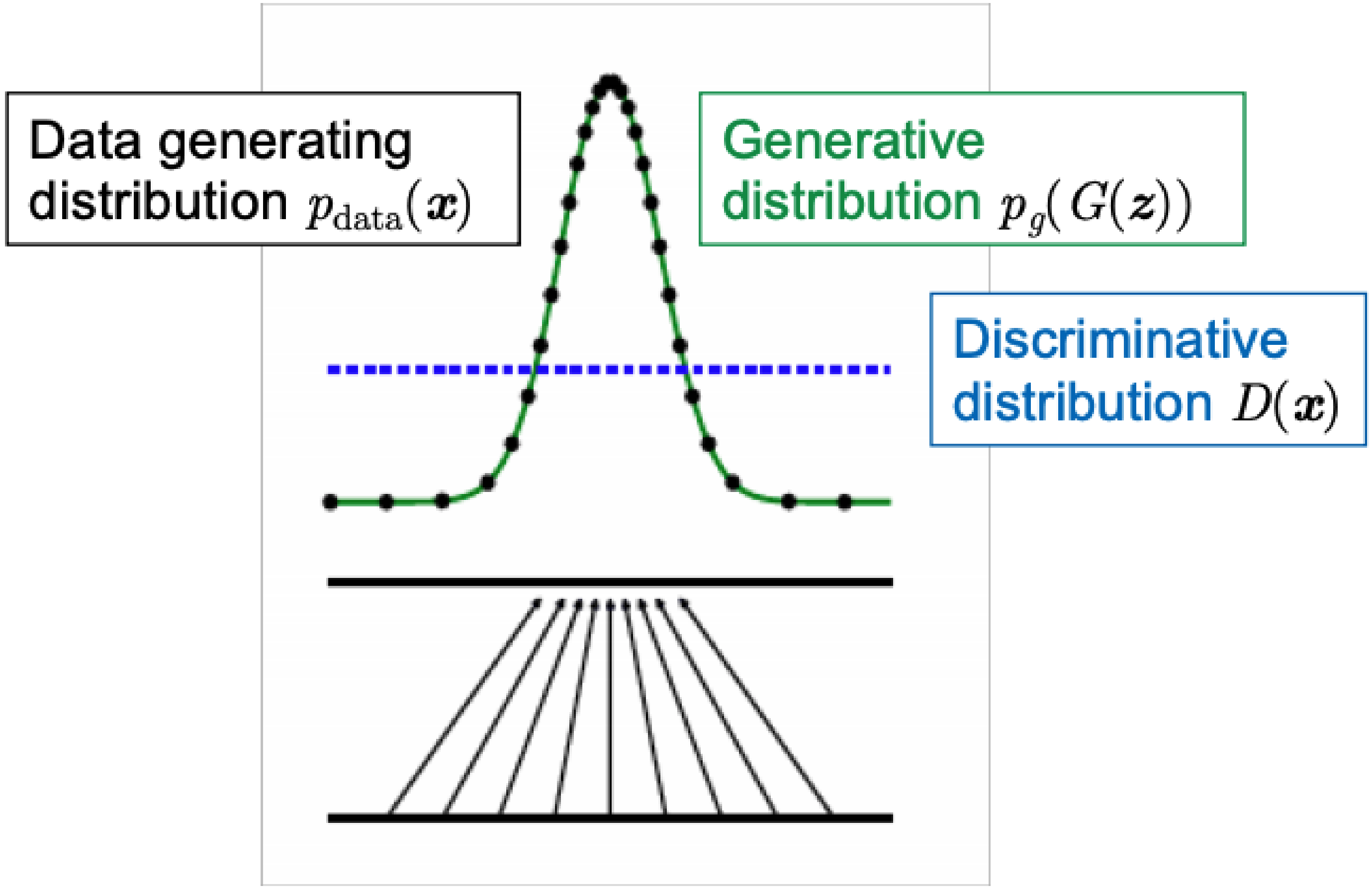
https://cedar.buffalo.edu/~srihari/CSE676/22.2-GAN%20Theory.pdf

GAN Training - Intuition



https://cedar.buffalo.edu/~srihari/CSE676/22.2-GAN%20Theory.pdf

GAN Training - Intuition



https://cedar.buffalo.edu/~srihari/CSE676/22.2-GAN%20Theory.pdf

Value function

E_x is the expected value over all real data instances

E_z is the expected value over all generated fake instances G(z)

G(z) is the generator's output when given noise z

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

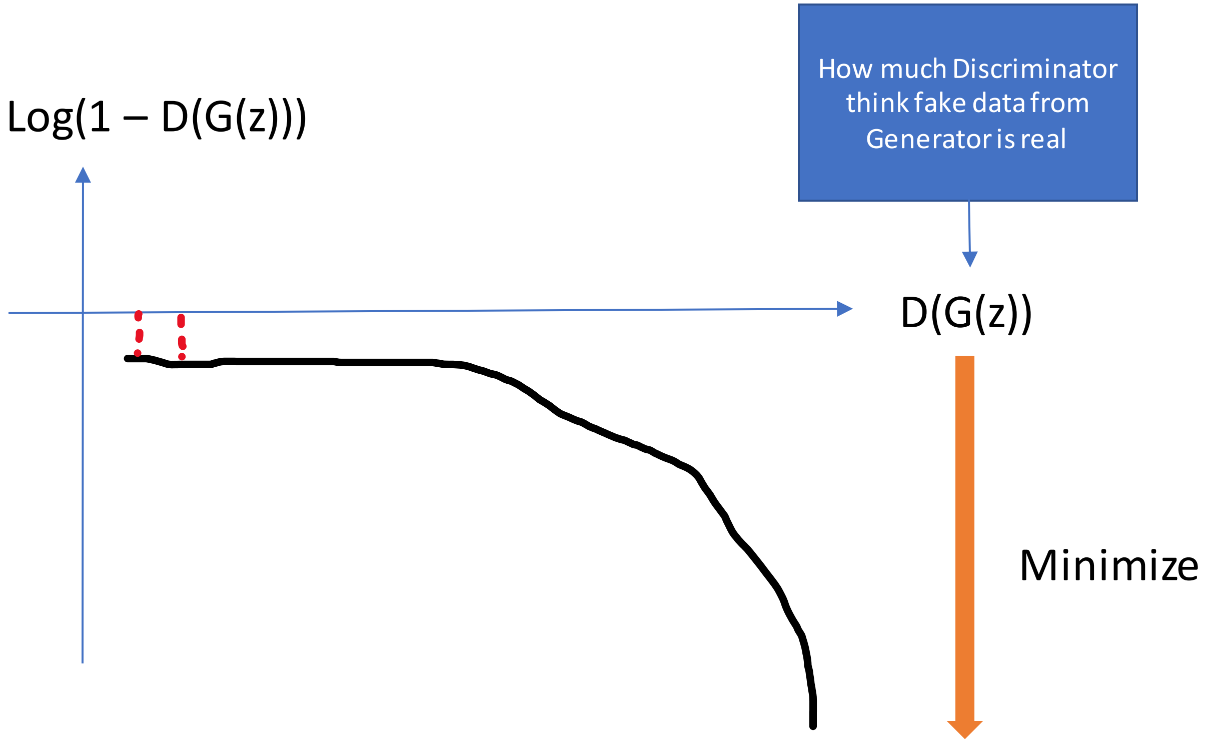
D(x) is the discriminator's estimate of the probability that real data instance x is real

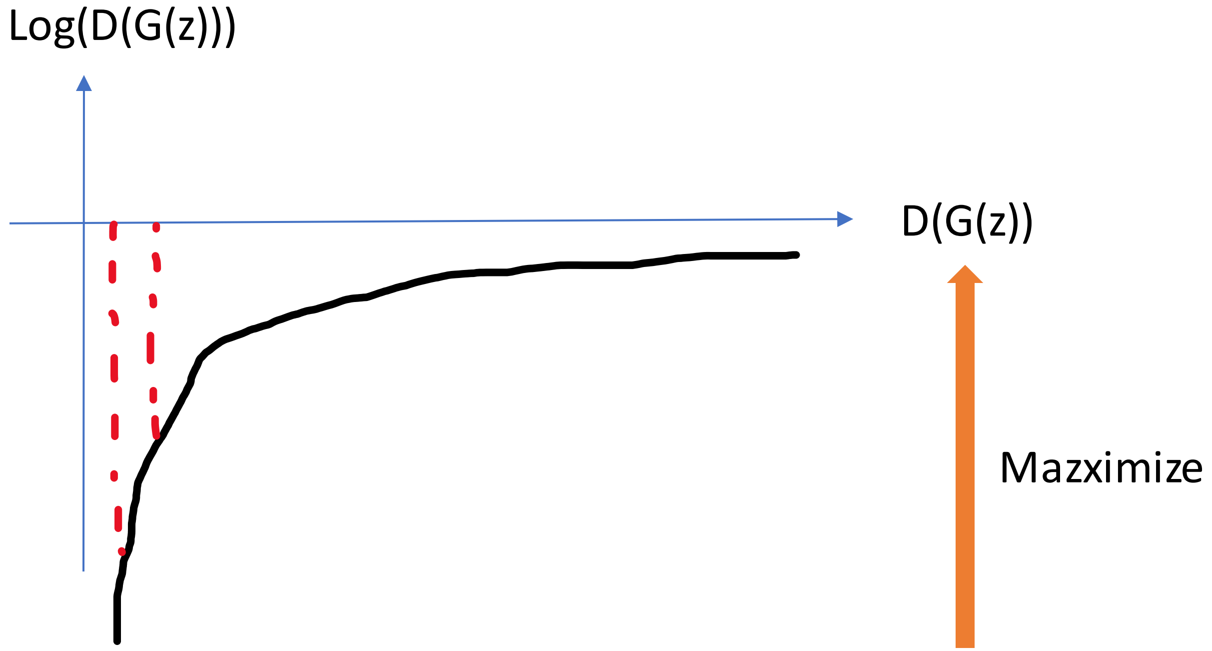
D(G(z)) is the discriminator's estimate of the probability that a fake instance is real

- The Generator tries to minimize this function while the Discriminator tries to maximize it.
- The Generator can't directly affect the **log(D(x))** term in the function, so it minimizes the equivalent **log(1 - D(G(z)))**.

Alternate gradient updates

In practice, equation 1 may not provide sufficient gradient for G to learn well. Early in learning, when G is poor, D can reject samples with high confidence because they are clearly different from the training data. In this case, $\log(1 - D(G(\boldsymbol{z})))$ saturates. Rather than training G to minimize $\log(1 - D(G(\boldsymbol{z})))$ we can train G to maximize $\log D(G(\boldsymbol{z}))$. This objective function results in the same fixed point of the dynamics of G and D but provides much stronger gradients early in learning.





GAN – Proof of optimality

We first consider the optimal discriminator D for any given generator G.

Proposition 1. *For G fixed, the optimal discriminator D is*

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})} \tag{2}$$

Proof. The training criterion for the discriminator D, given any generator G, is to maximize the quantity $V(G, D)$

$$\begin{aligned} V(G, D) &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_z p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz \\ &= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx \end{aligned} \tag{3}$$

For any $(a, b) \in \mathbb{R}^2 \setminus \{0, 0\}$, the function $y \to a \log(y) + b \log(1 - y)$ achieves its maximum in $[0, 1]$ at $\frac{a}{a+b}$. The discriminator does not need to be defined outside of $Supp(p_{\text{data}}) \cup Supp(p_g)$, concluding the proof. $\square$

GAN – Proof of optimality

Proposition 2. *If G and D have enough capacity, and at each step of Algorithm 1, the discriminator is allowed to reach its optimum given G, and p_g is updated so as to improve the criterion*

$$\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D^*_G(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - D^*_G(\boldsymbol{x}))]$$

then p_g converges to p_{data}

Proof. Consider $V(G, D) = U(p_g, D)$ as a function of p_g as done in the above criterion. Note that $U(p_g, D)$ is convex in p_g. The subderivatives of a supremum of convex functions include the derivative of the function at the point where the maximum is attained. In other words, if $f(x) = \sup_{\alpha\in\mathcal{A}} f_\alpha(x)$ and $f_\alpha(x)$ is convex in x for every α, then $\partial f_\beta(x) \in \partial f$ if $\beta = \arg\sup_{\alpha\in\mathcal{A}} f_\alpha(x)$. This is equivalent to computing a gradient descent update for p_g at the optimal D given the corresponding G. $\sup_D U(p_g, D)$ is convex in p_g with a unique global optima as proven in Thm 1, therefore with sufficiently small updates of p_g, p_g converges to p_x, concluding the proof. $\square$

GAN - Experiments

Figure 2: Visualization of samples from the model. Rightmost column shows the nearest training example of the neighboring sample, in order to demonstrate that the model has not memorized the training set. Samples are fair random draws, not cherry-picked. Unlike most other visualizations of deep generative models, these images show actual samples from the model distributions, not conditional means given samples of hidden units. Moreover, these samples are uncorrelated because the sampling process does not depend on Markov chain mixing. a) MNIST b) TFD c) CIFAR-10 (fully connected model) d) CIFAR-10 (convolutional discriminator and "deconvolutional" generator)

GAN -Advantages and disadvantages

Advantages

- Only backprop is used to obtain gradients
- Generator network not being updated directly with data examples, but only with gradients flowing through the discriminator => computational advantage
- GAN can represent very sharp, even degenerate distributions

Disadvantages

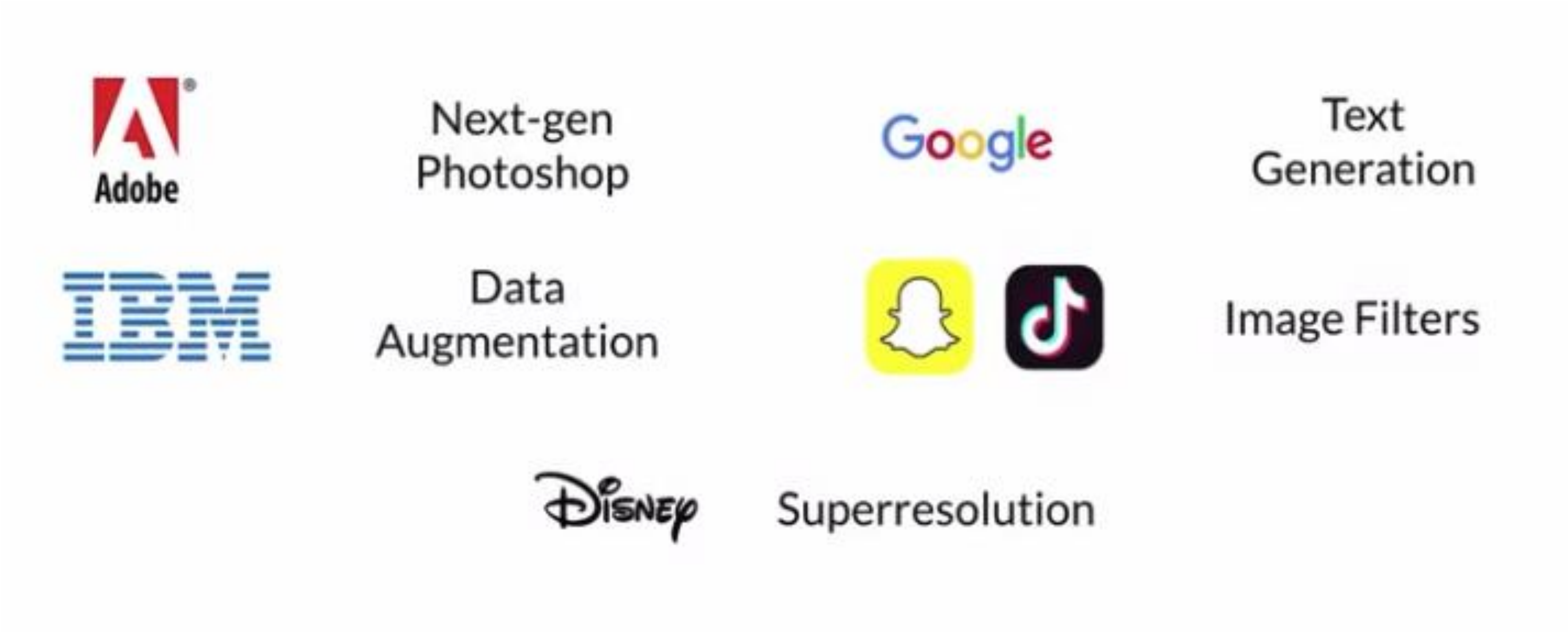
- D must be synchronized well with G during training

GAN evolutions and applications

FaceApp

https://www.thispersondoesnotexist.com/

Companies using GAN



Sharon Zhou "GAN for good" Coursera

References

- https://www.cs.toronto.edu/~duvenaud/courses/csc2541/slides/gan-foundations.pdf
- https://cedar.buffalo.edu/~srihari/CSE676/22.2-GAN%20Theory.pdf
- https://www.tensorflow.org/tutorials/generative/dcgan
- https://www.coursera.org/learn/build-basic-generative-adversarial-networks-gans/home/week/1
- https://www.youtube.com/watch?v=8L11aMN5KY8&ab_channel=LuisSerrano
- https://medium.com/datadriveninvestor/deep-learning-generative-adversarial-network-gan-34abb43c0644

Thank you