PMI Lab
Pattern Recognition and
Machine Intelligence

Hankuk University
of Foreign Studies

Generative Adversarial Networks (GANs)

Prof. Jae Young Choi

Pattern Recognition and Machine Intelligence (PMI) Lab.
Hankuk University of Foreign Studies

Introduction and Background

Generative Model (1/3)

- Generative adversarial networks are an example of **generative models**

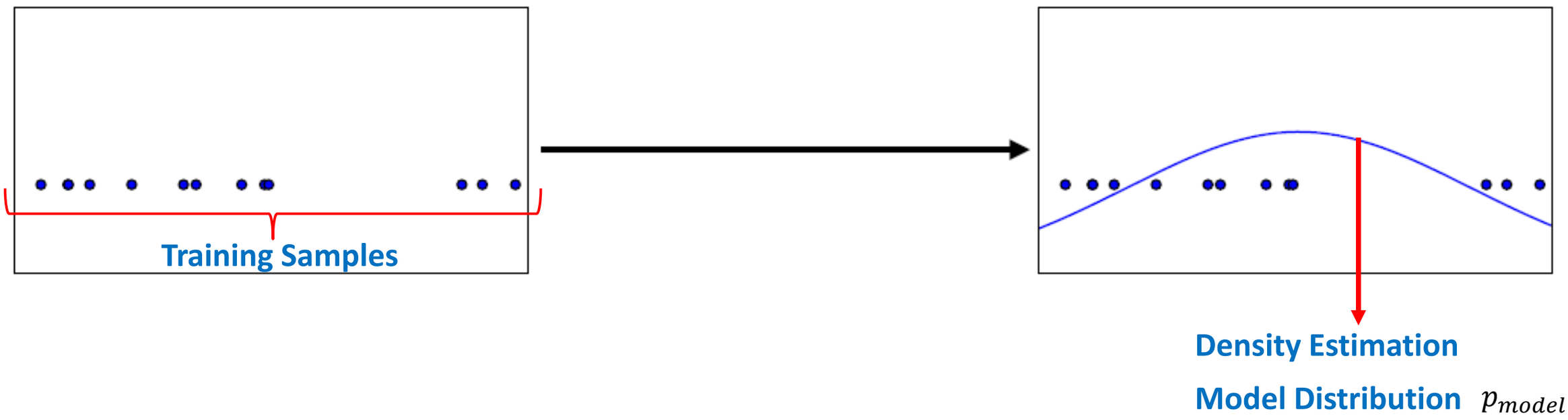
- This refers to any model that takes a training set, consisting of samples drawn from a distribution p_{data} and learns to represent an estimate of distribution

→ **Density Estimation**

- GANs focus primarily on **sample generation**

Generative Model (2/3)

❖ Example of Density Estimation



- A training set of examples drawn from an unknown data-generating distribution p_{data}
- Estimate of distribution p_{data}
- The estimate p_{model} can be evaluated for a particular value of x to obtain an estimate $p_{model}(x)$ of the true density $p_{model}(x)$.

Generative Model (3/3)

- Example of Sample Generation

Training Examples **Model Samples**

- Generation of samples form the model distribution
- Illustration of samples from the ImageNet dataset
- Generative model would be able to train on examples on the left and then create examples from the same distribution on the right

Why study generative models?

- Excellent test of our ability to use high-dimensional, complicated probability distributions
- Missing data
  - Semi-supervised learning
  - Realistic generation tasks
- Sampling (or generation) is straightforward
- Training doesn't involve maximum likelihood estimation
- Robust to overfitting since the generator never sees the training data
- GANs are good at capturing the modes of distribution

Discriminative vs. Generative Models

❖ Discriminative models

- A discriminative model learns the conditional probability distribution p(y|x) which could be interpreted as the probability of y given x

- Classifier learns by observing data. It makes fewer assumptions on the distributions, but depends heavily on the quality of the data. The distribution p(y|x) simply classifies a given example x directly int a label y.

Discriminative vs. Generative Models

❖ Generative models (1/2)

- Whereas a generative model learns the joint probability distribution p(x,y), where x is the input data and y is the label that you want to classify

- A generative model can generate more samples by itself artificially, based on assumptions about the distribution of data

- For example, in the Naïve Bayes' model, we can learn p(x) from data, also p(y), the prior class probabilities, and we can also learn p(x|y) from the data using say maximum likelihood

$$\arg\max p(x|y)p(y)$$

Discriminative vs. Generative Models

❖ Generative models (2/2)

$$\arg\max p(x|y)p(y)$$

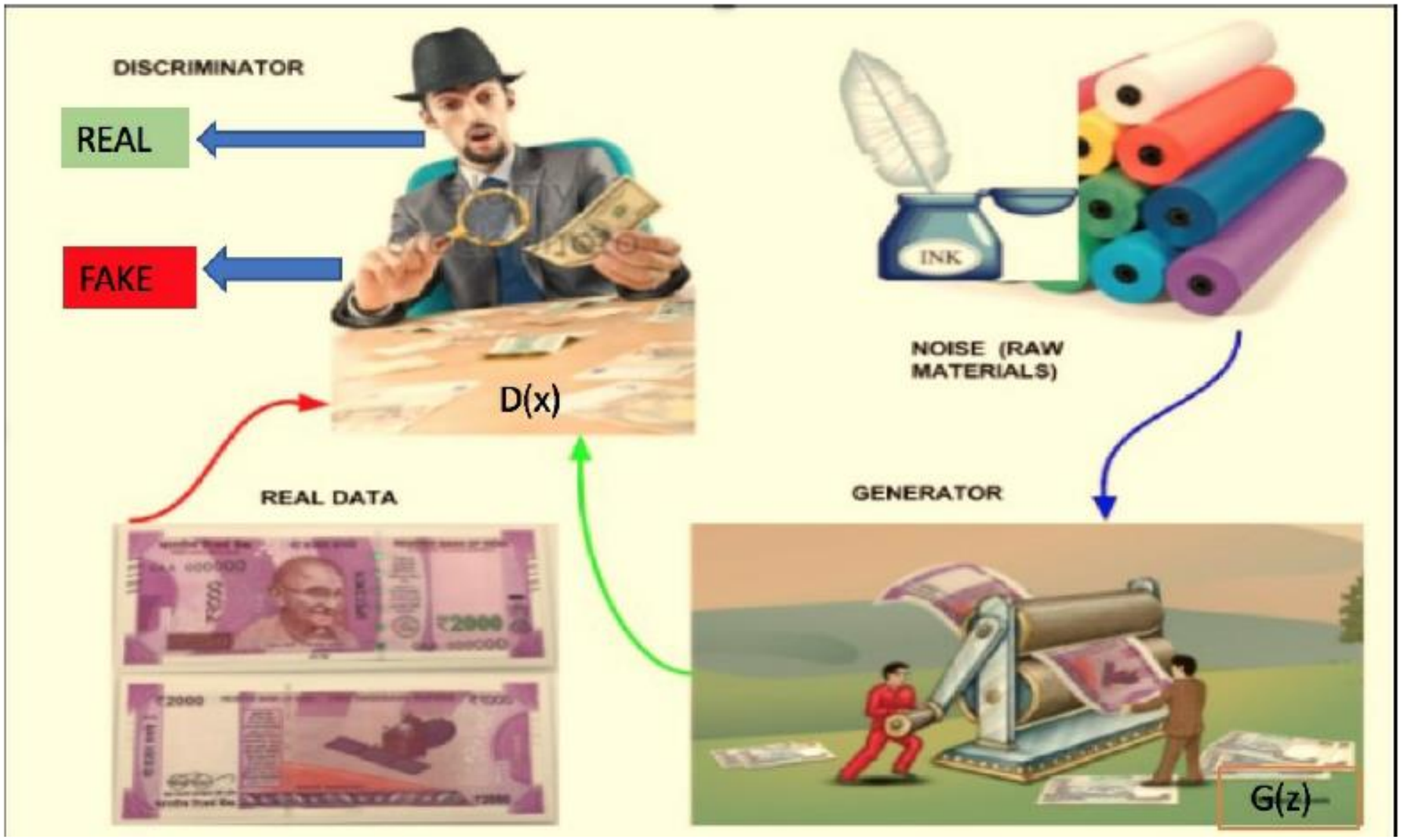
- This is the equation we use in generative models, as $p(x, y) = p(x|y)p(y)$, which explicitly models the actual distribution of each class

- In practice, the discriminative models generally outperform generative models in classification tasks, but the generative model shines over discriminative models in generation task

Adversarial Process

- This can be modeled as a **minimax game** in game theory. This phenomenon is called **adversarial process**
- GAN, introduced by Ian Goodfellow in 2014 at arXiv: 1406.2661, is a special case of an adversarial process where **two neural networks compete against each other**
- The first network generates data and the second network tries to find the difference between the real data and the fake data generated by the first network
- The second network will output a scalar [0,1], which represents a probability of real data

Adversarial Process: Real World Analogy



Police
Discriminator D(x)

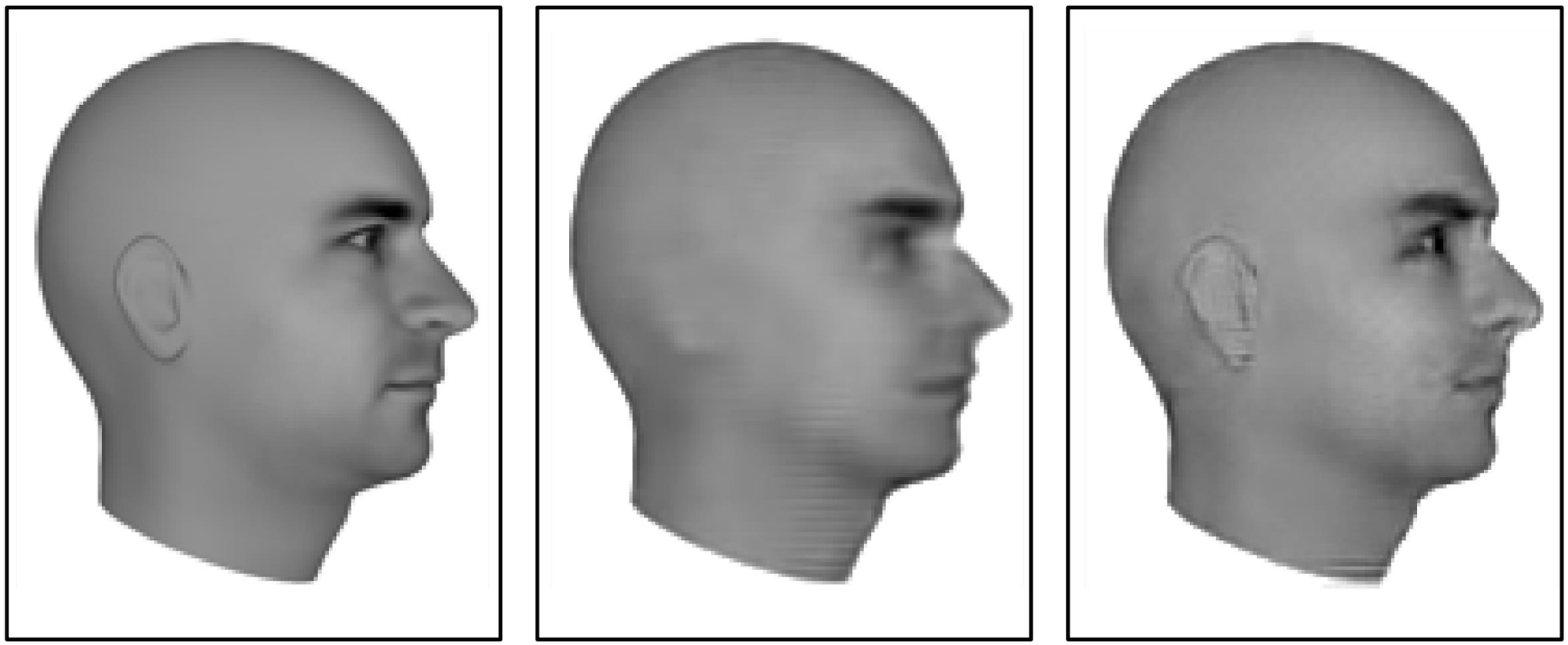
Counterfeiter
Generator G(z)

- To become a successful money counterfeiter, the criminal needs to fool the police so that the police can't tell the difference between fake and real money
- As a paragon of justice, the police want to defect fake money as effectively as possible

Applications

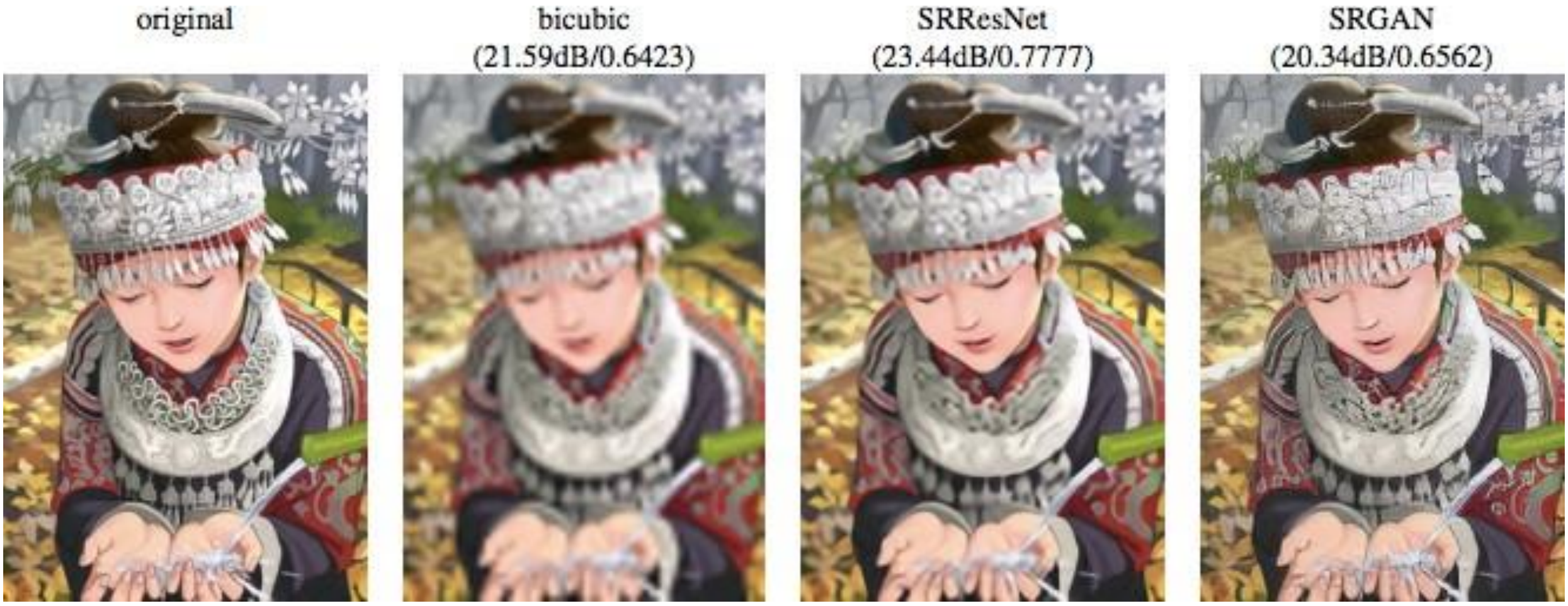
Next Video Frame Prediction

Ground Truth | MSE | Adversarial



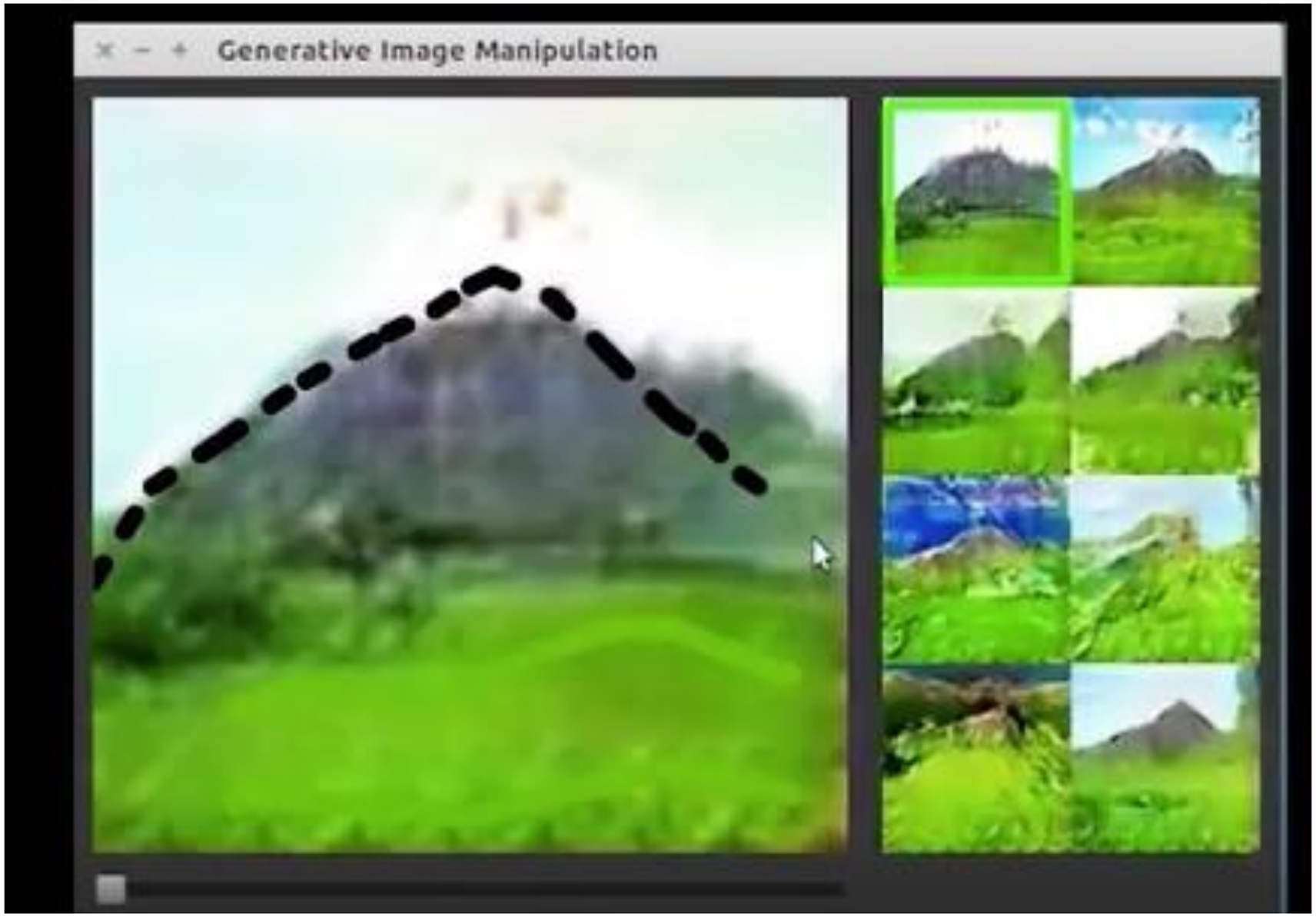
(Ledig et al 2016)

Single Image Super-Resolution



(Ledig et al 2016)

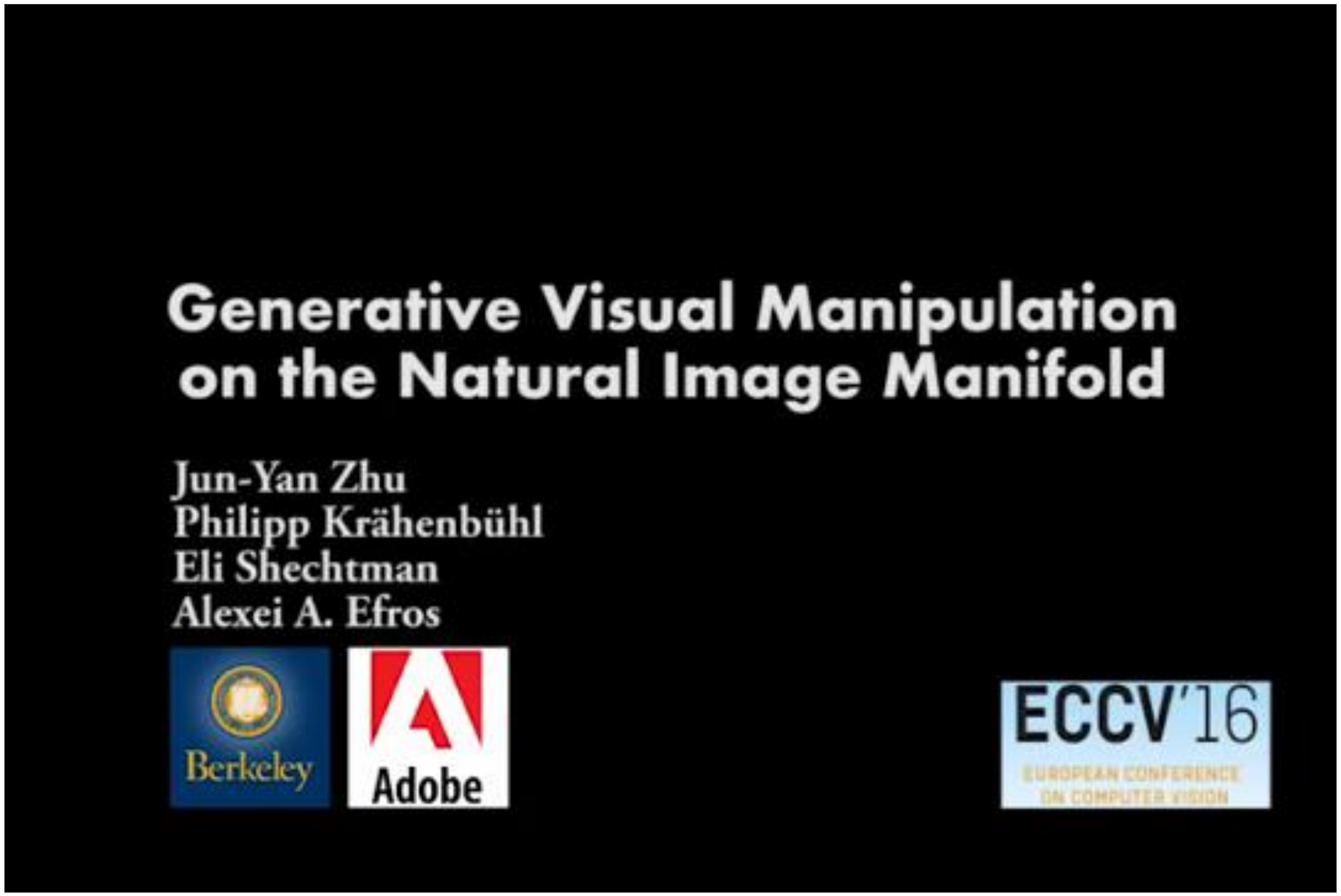
iGAN



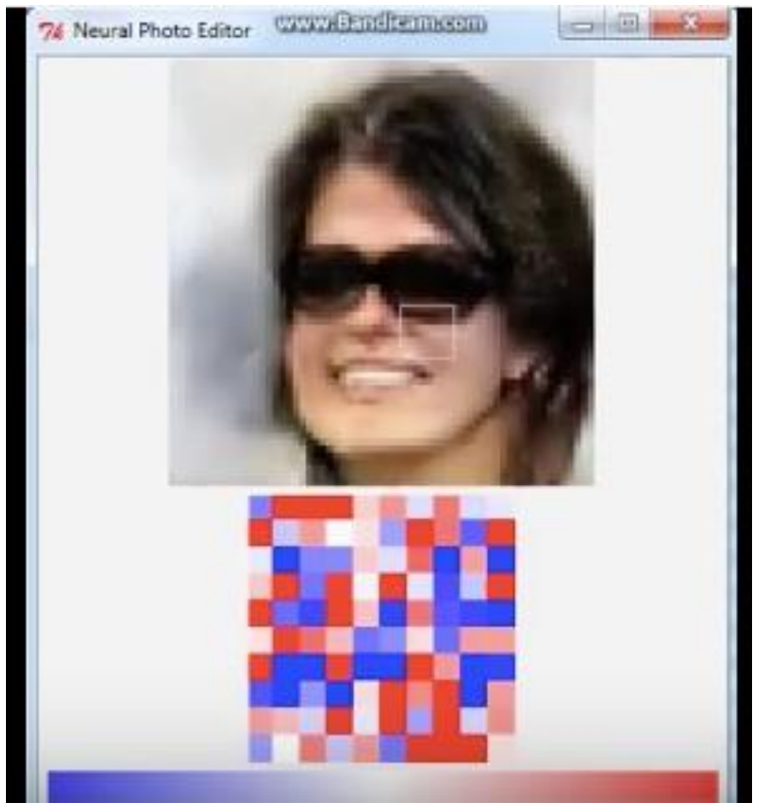
youtube

(Zhu et al 2016)

iGAN


Generative Visual Manipulation
on the Natural Image Manifold
Jun-Yan Zhu
Philipp Krähenbühl
Eli Shechtman
Alexei A. Efros
Berkeley
Adobe
ECCV'16
EUROPEAN CONFERENCE
ON COMPUTER VISION

Introspective Adversarial Networks



youtube

(Brock et al 2016)

Introspective Adversarial Networks

[redacted]

Neural Photo Editing

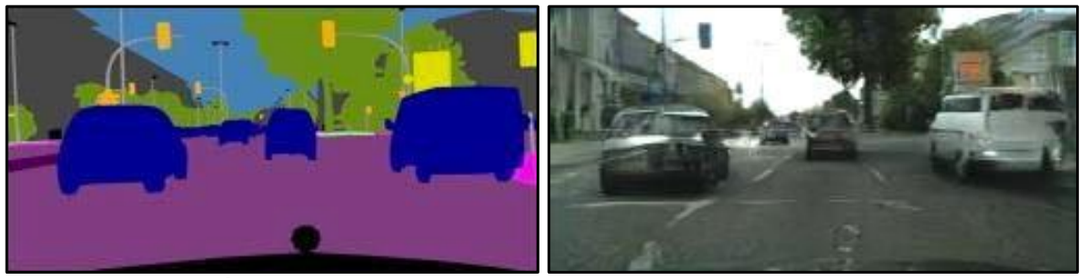
Andrew Brock

EDINBURGH CENTRE FOR ROBOTICS
Innovation Ready

[redacted]

Image to Image Translation

Labels to Street Scene



input output

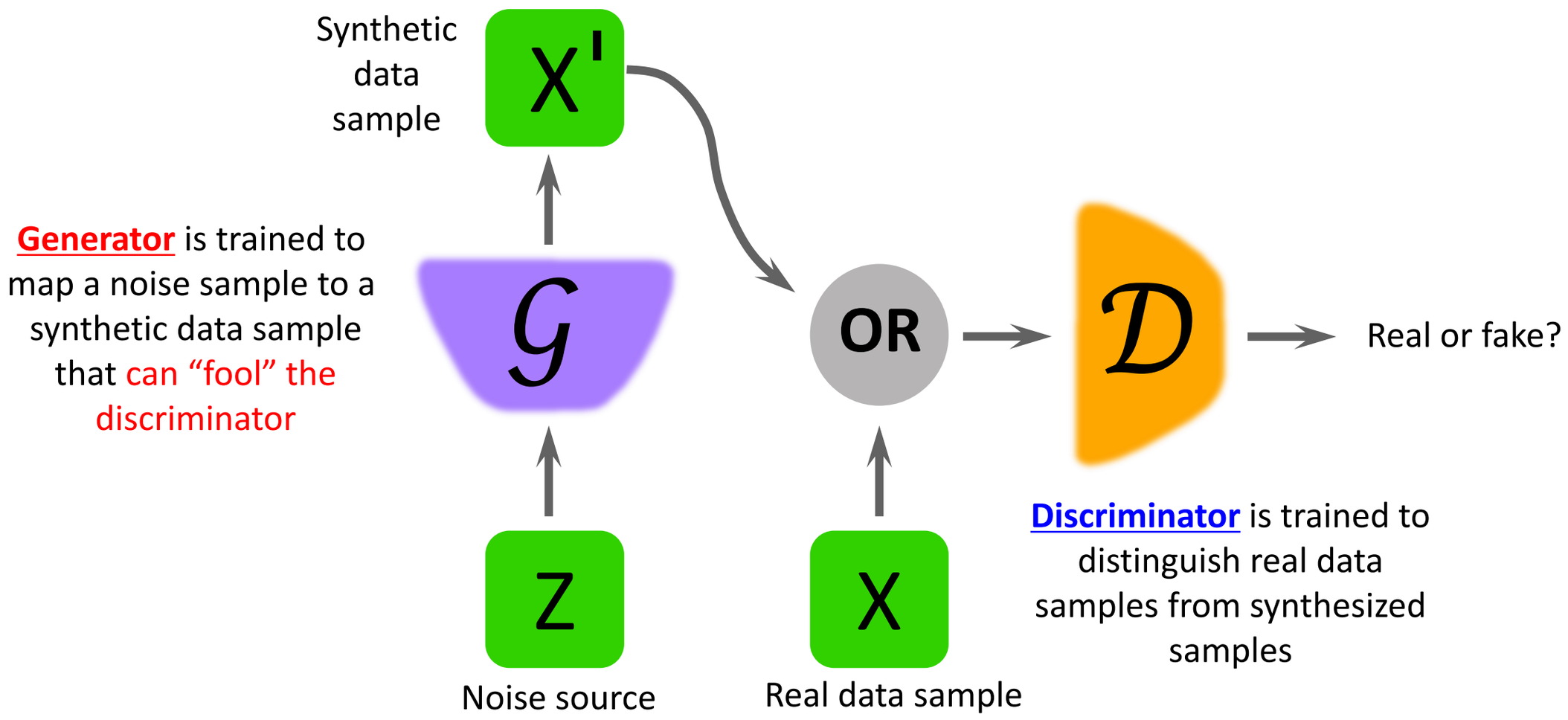
Aerial to Map

input output

(Isola et al 2016)

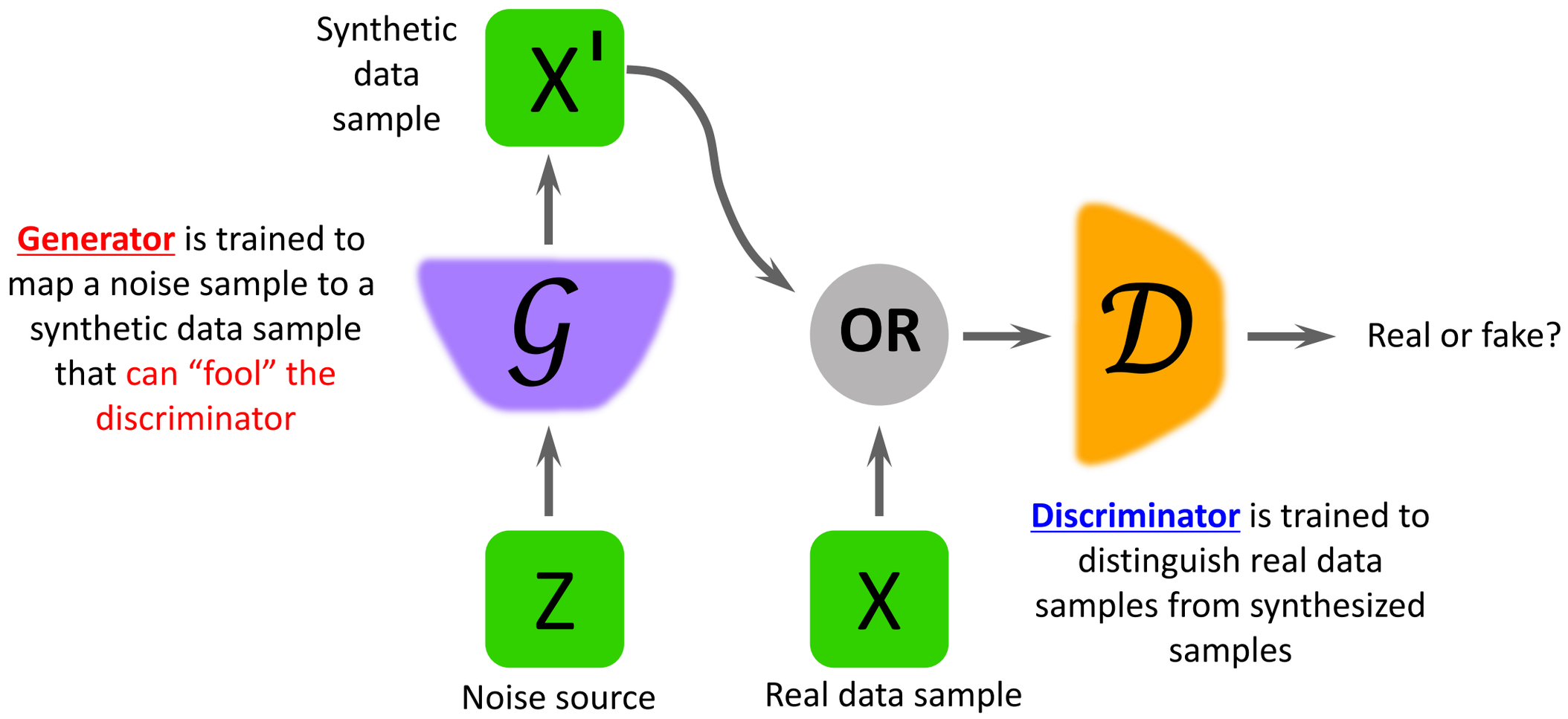
Generative Adversarial Networks (GANs)

Overview of GAN (1/5)



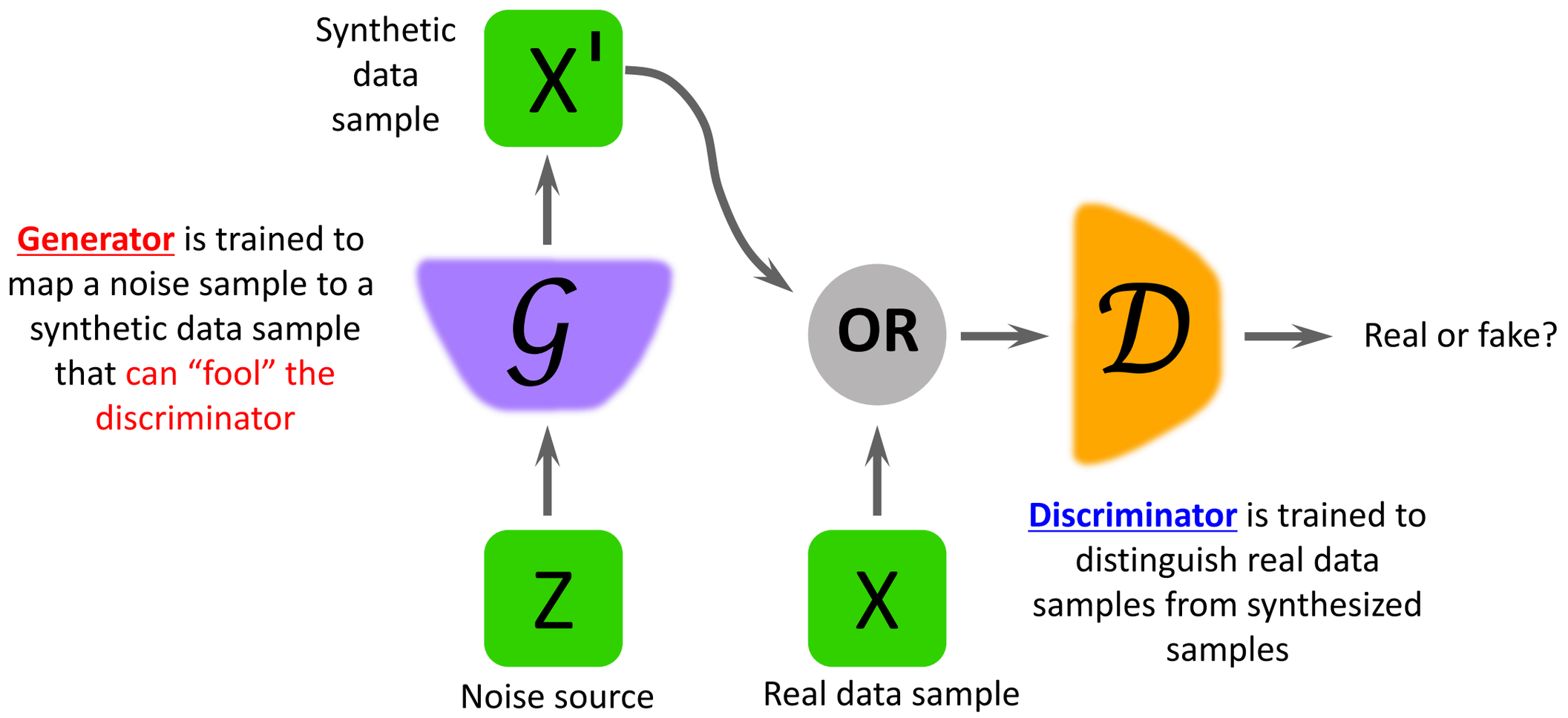
- The forger, known in the GAN literature as the generator, $\mathcal{G}$, creates forgeries, with the aim of making realistic images
- The expert, known as the discriminator, $\mathcal{D}$, receives both forgeries and real images, and aims to tell them apart
- Both are trained simultaneously, and in competition with each other

Overview of GAN (2/5)



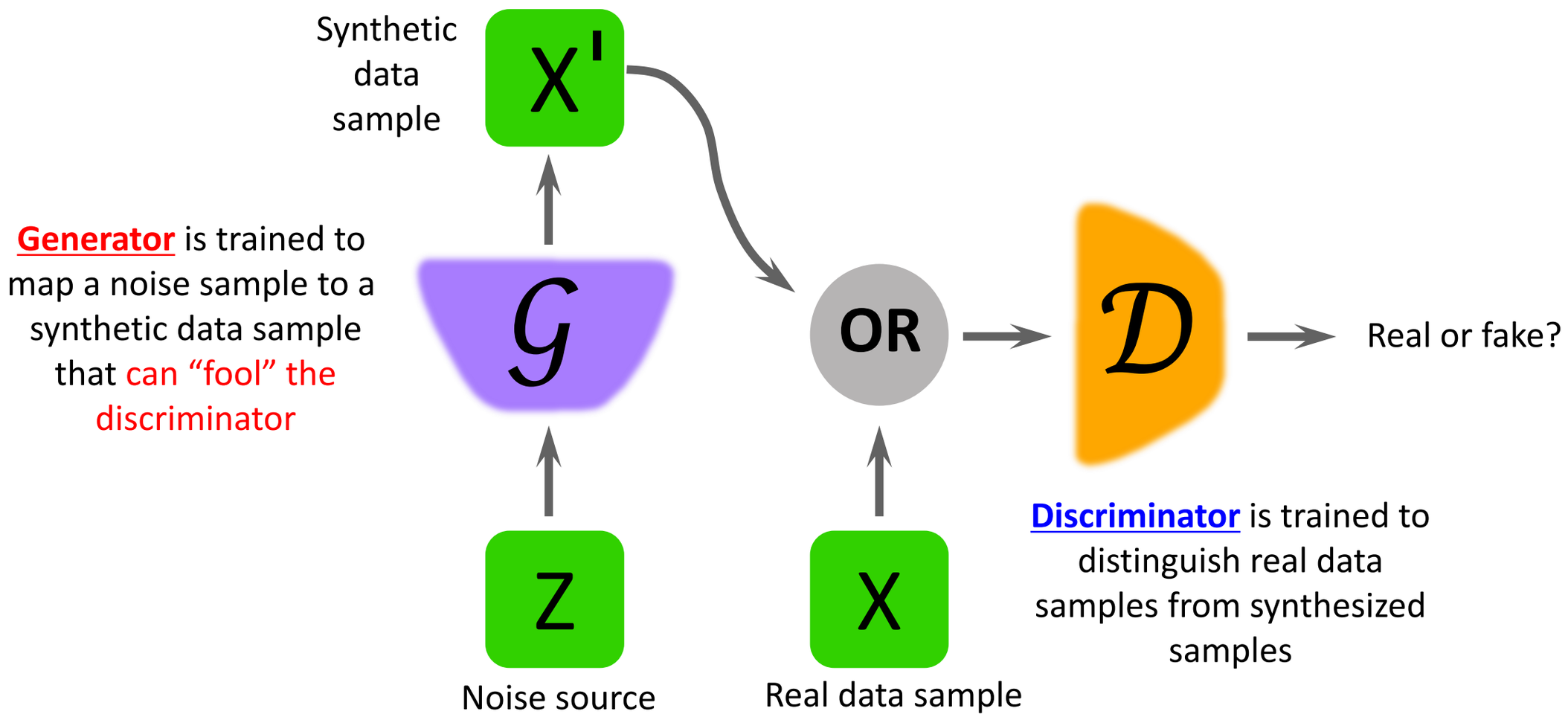
- Crucially, Generator has no direct access to real images – the only way it learns is through its interaction with the discriminator
- Discriminator has access to both the synthetic samples and samples drawn from the stack of real images

Overview of GAN (3/5)



We express generator formally as $\mathcal{G}: \mathcal{G}(\mathrm{z}) \rightarrow R^{|\mathrm{x}|}$, where $\mathrm{z} \in R^{|\mathrm{z}|}$ is a sample from the latent space, $\mathrm{x} \in R^{|\mathrm{x}|}$ is an image and $|\cdot|$ denotes the number of dimensions

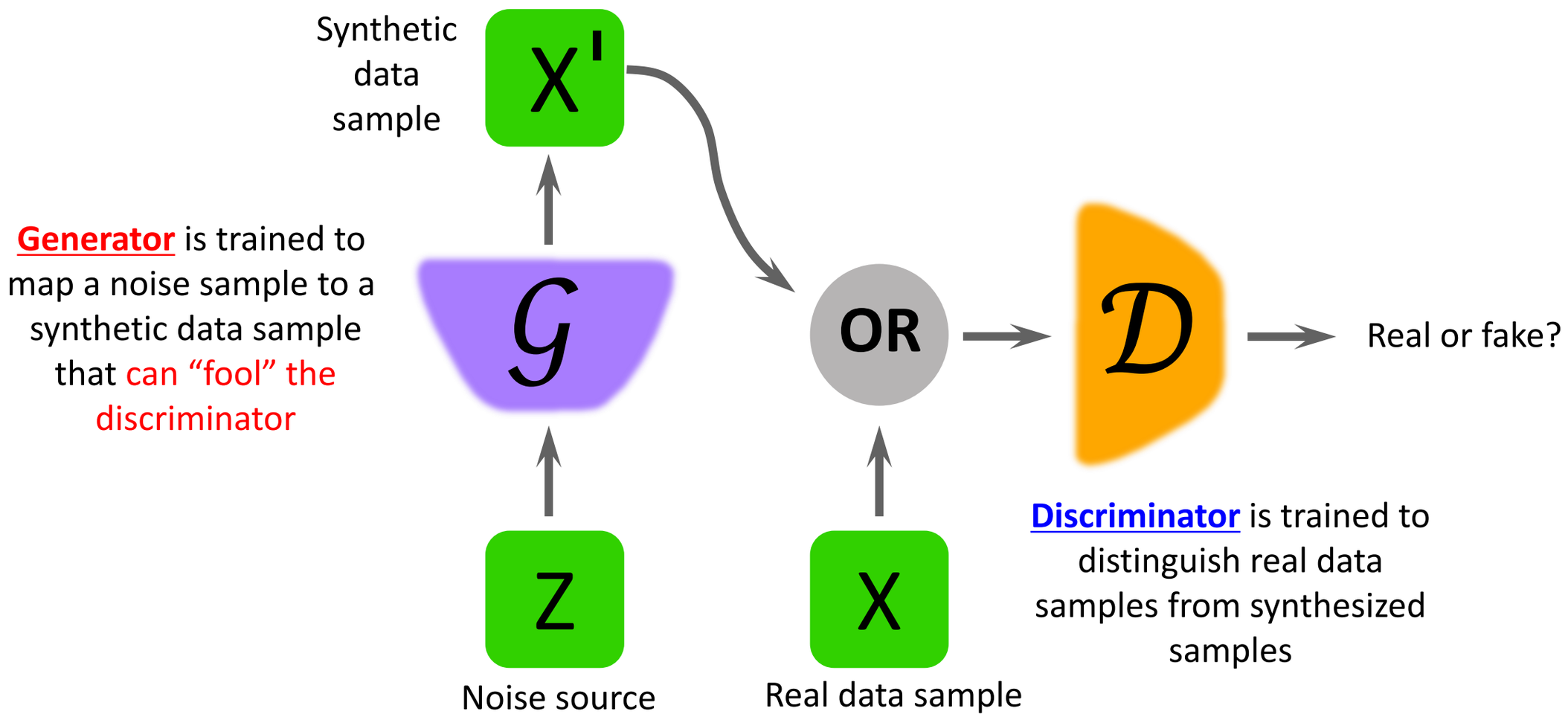
Overview of GAN (4/5)



- Discriminator network, $\mathcal{D}$, is characterized as a function that maps from image data to a probability that the image is from the real data distribution, rather than the generator distribution

- $\mathcal{D}: \mathcal{D}(\mathrm{x}) \rightarrow (0,1)$

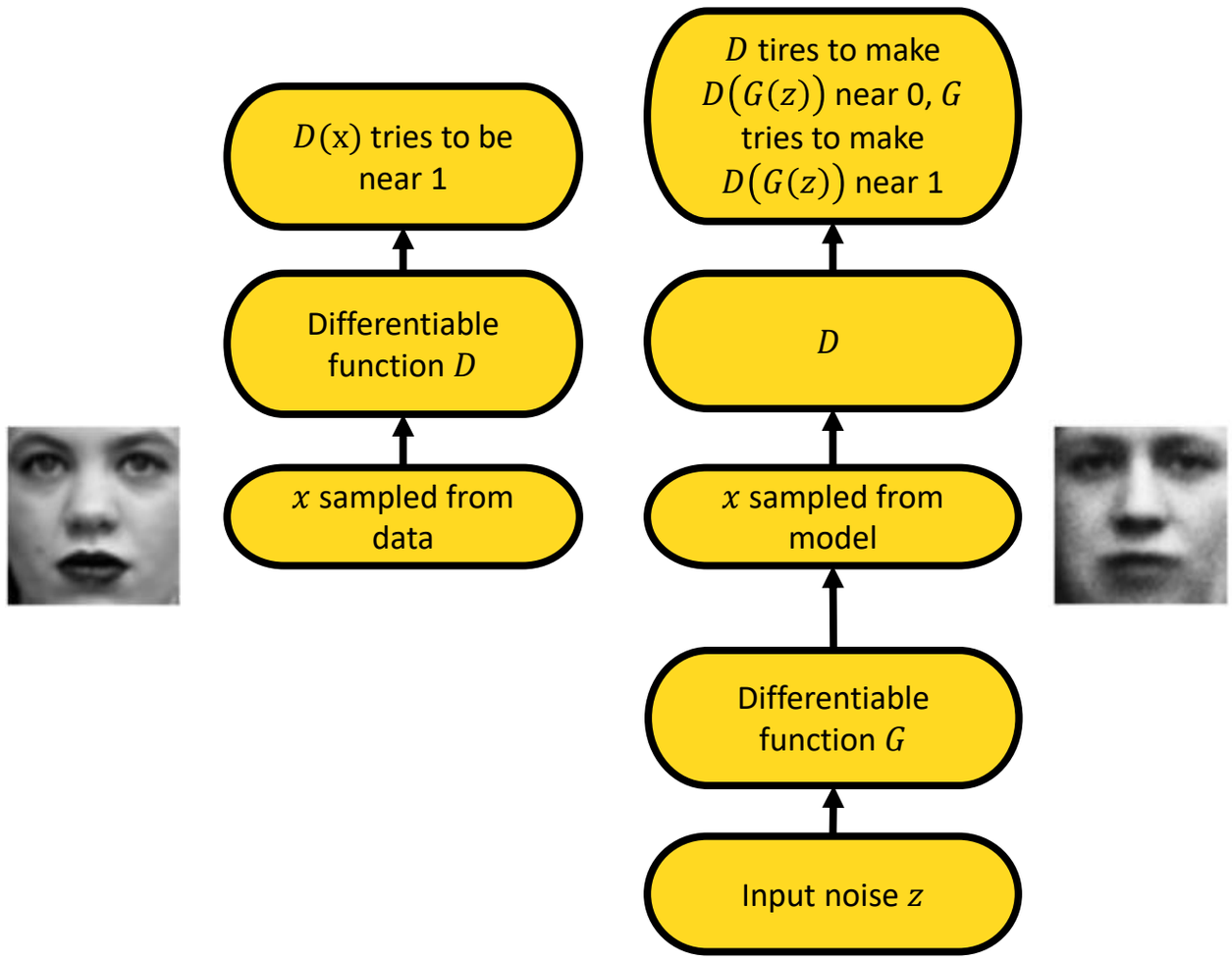
Overview of GAN (5/5)



- For a fixed generator, $\mathcal{G}$, the discriminator, $\mathcal{D}$, is trained to classify images as either being from the training data (real, close to 1) or from a fixed generator (fake, close to 0)
- If generator distribution is able to match real data distribution perfectly then the discriminator will be maximally confused, predicting 0.5 for all inputs

GAN Framework (1/2)

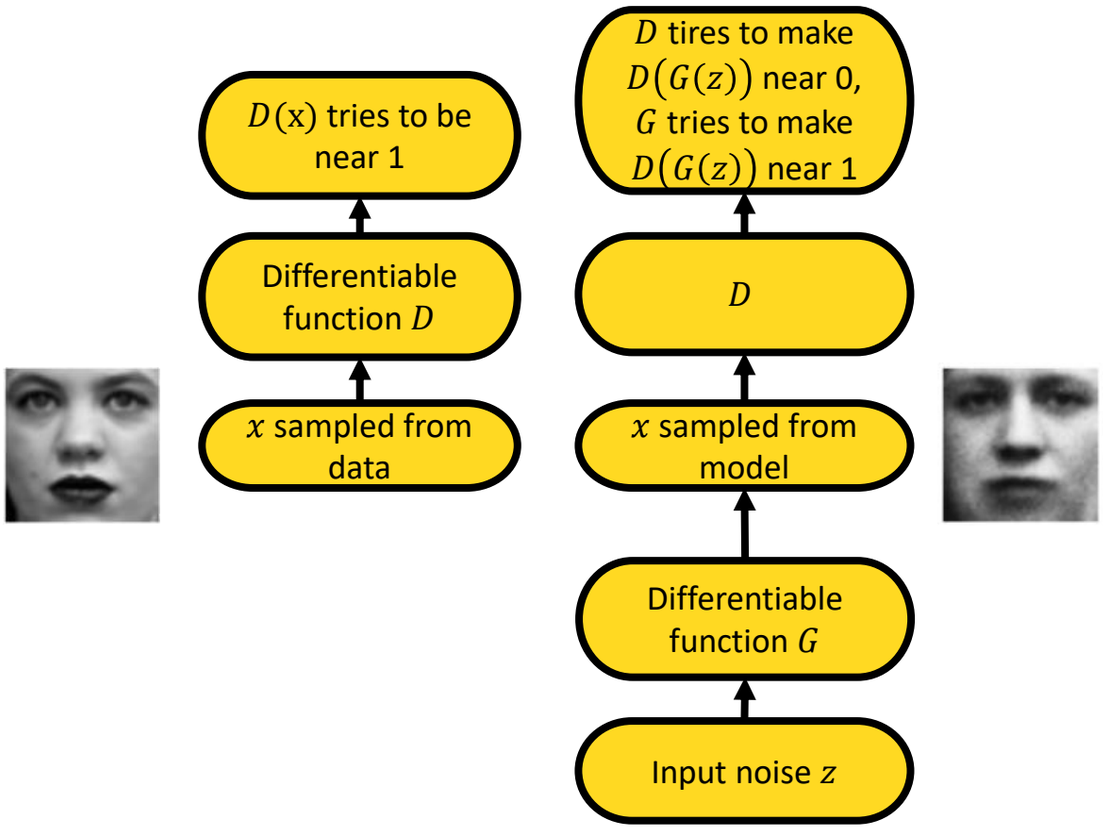
$$V(\mathcal{G},\mathcal{D}) = \mathbb{E}_{p_{data}(\mathrm{x})} \log \mathcal{D}(\mathrm{x}) + \mathbb{E}_{p_g(\mathrm{x})} \log\big(1 - \mathcal{D}(x)\big)$$



- The GAN framework pits two adversaries against each other in a game
- Each player is represented by a differentiable function controlled by a set of parameters
- The goal of discriminator(D) is to output the probability that its input is real rather than fake

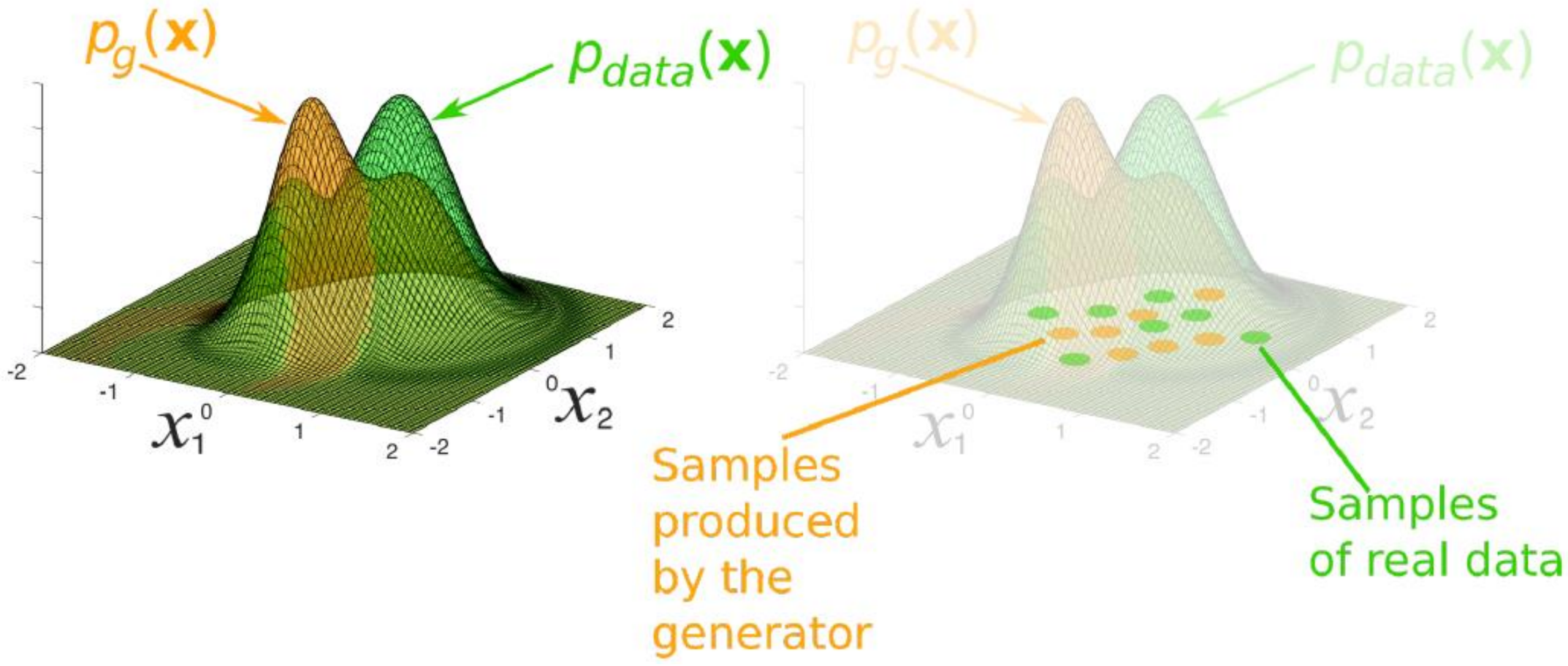
GAN Framework (2/2)

$$V(\mathcal{G},\mathcal{D}) = \mathbb{E}_{p_{data}(\mathrm{x})} \log \mathcal{D}(\mathrm{x}) + \mathbb{E}_{p_g(\mathrm{x})} \log\big(1 - \mathcal{D}(x)\big)$$



- The goal of the discriminator is for $D(x)$ to be near 1 while inputs z to the generator are randomly sampled from the model's prior over the latent variables
- The discriminator then receives input $G(z)$, a fake sample created by the generator
- The discriminator strives to make $D\big(G(z)\big)$ approach 0 while the generative strives to make the same quantity approach 1
- If both models have sufficient capacity, then the Nash equilibrium of this game corresponds to the $G(z)$ being drawn from the same distribution as the training data, and $D(x) = \frac{1}{2}$ for all x

Nash equilibrium: Capturing Data Distribution



- During GAN training, the generator is encouraged to produce a distribution of samples, $p_g(\mathrm{x})$ to match that of real data, $p_{data}(\mathrm{x})$

- For an appropriately parametrized and trained GAN, these distributions will be nearly identical

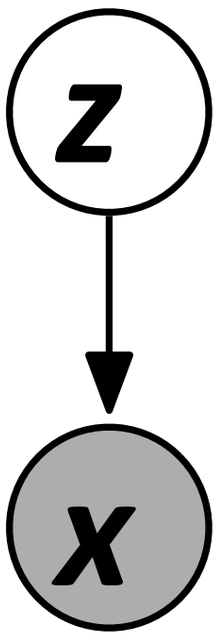
Training GANs

- Training of GANs involves both **finding the parameters of a discriminator** that maximize its classification accuracy

- At the same time, finding **the parameters of a generator which maximally confuse the discriminator**

Training GANs

❖ Generator Network



$$\boldsymbol{x} = G(\boldsymbol{z}; \boldsymbol{\theta}^{(G)})$$

- Must be differentiable
- No invertibility requirement
- Trainable for any size of *z*
- Can make *x* conditionally Gaussian given *z* but need not do so

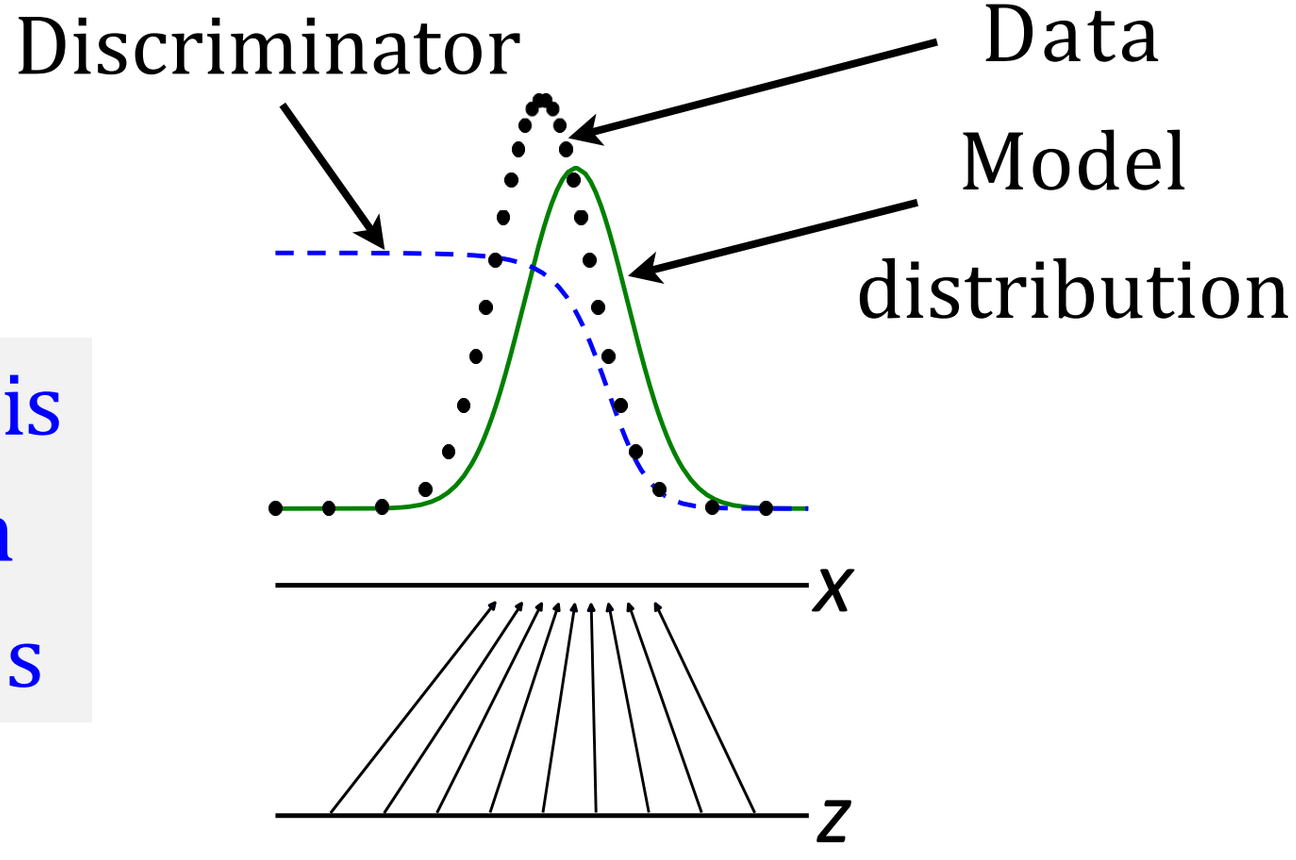
Training GANs

❖ Discriminator Strategy

Optimal $D(\boldsymbol{x})$ for any $p_{\text{data}}(\boldsymbol{x})$ and $p_{\text{model}}(\boldsymbol{x})$ is always

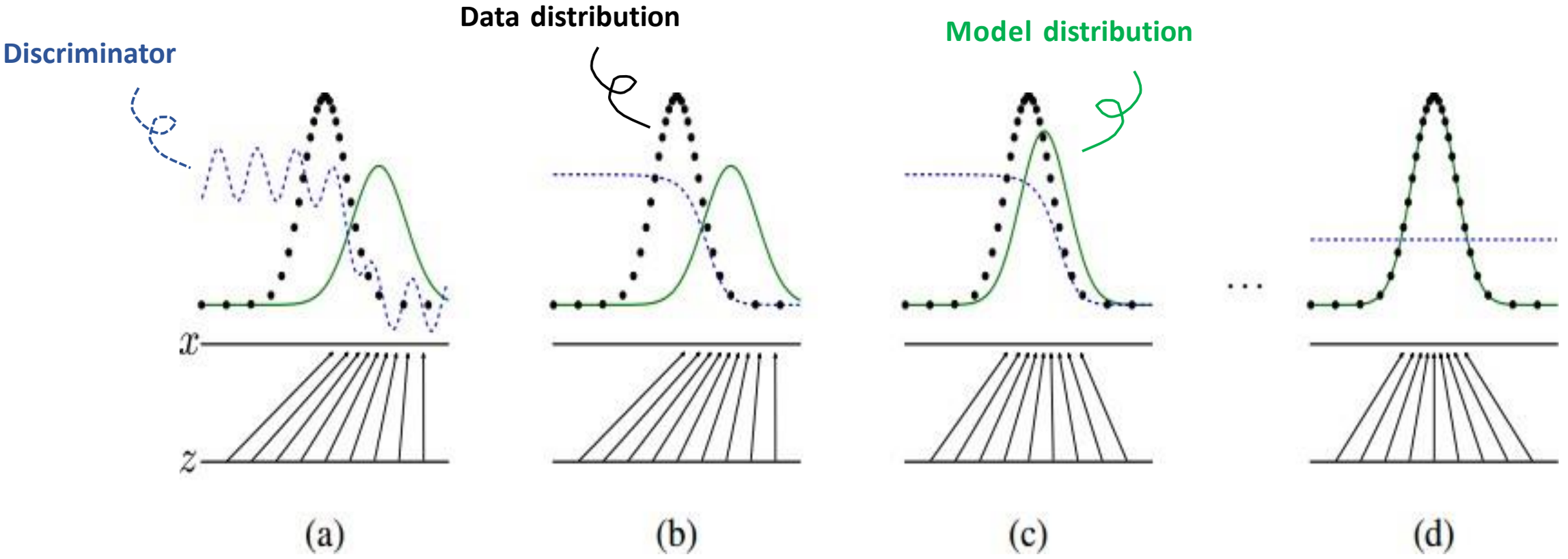
$$D(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)}$$

Estimating this ratio is the key approximation mechanism used by GANs



Training GANs

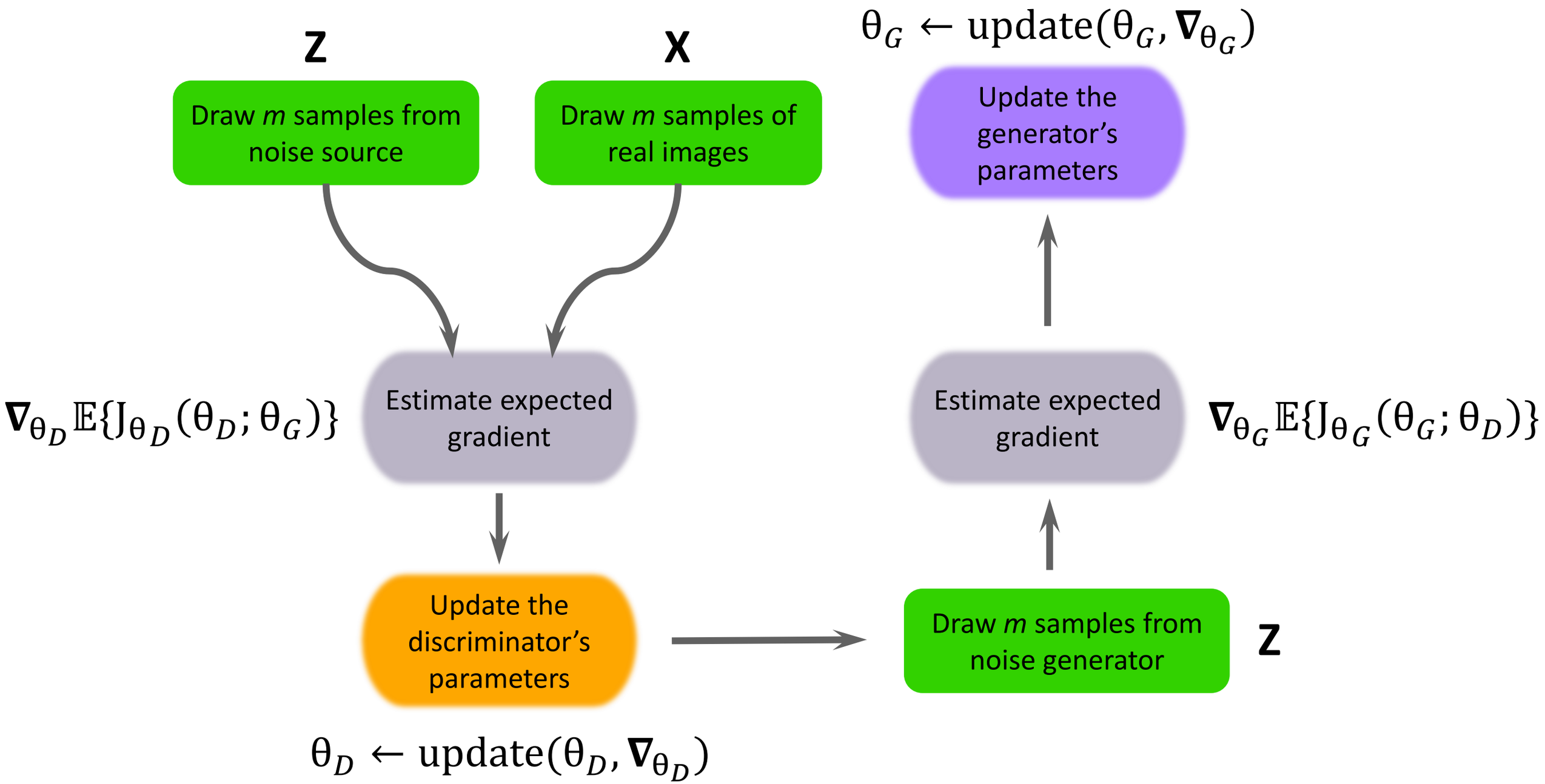
❖ Discriminator Strategy (*Cont'd*)



- GANs are trained by simultaneously updating **discriminative distribution** (*D*, blue, dashed line) so that it discriminates between samples from the data generating distribution (black, dotted line) from those of **generative (model) distribution** (green, solid line).
- The lower horizontal line is the domain from which latent variable z is uniformly sampled
- The horizontal line above is part of the domain of sample x

Training GANs

- Main loop of GAN Training



- Novel data samples, **x**', is drawn by passing random samples, **z**
- The gradient of the discriminator is updated before updating the generator

Training GANs

❖ Cost Function

$$\max_{\mathcal{D}} \min_{\mathcal{G}} V(\mathcal{G}, \mathcal{D})$$

$$V(\mathcal{G}, \mathcal{D}) = \mathbb{E}_{p_{data}(\mathrm{x})} \log \mathcal{D}(\mathrm{x}) + \mathbb{E}_{p_g(\mathrm{x})} \log\big(1 - \mathcal{D}(\mathrm{x})\big)$$

- During training, the parameters of one model are updated, while the parameters of the other are fixed
- For a fixed generator there is a unique optimal discriminator,

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

- Generator, $\mathcal{G}$, is optimal when $p_g(\mathrm{x}) = p_{data}(\mathrm{x})$, which is equivalent to the optimal discriminator predicting 0.5 for all samples drawn from x
- In other words, the generator is optimal when discriminator, $\mathcal{D}$, is maximally confused and cannot distinguish real samples from fake ones

Training GANs

❖ Heuristic, non-saturating game

➢ Motivation

- The cost used for generator in the minimax game is useful for theoretical analysis, but does not perform especially well in practice
- When discriminator successfully rejects generator samples with high confidence, the generator's gradient vanishes
- Generator can still learn even when discriminator successfully rejects all generator samples

➢ Cost for non-saturating game

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{x\sim p_{data}} \log D(x) - \frac{1}{2}\mathbb{E}_z \log\Big(1 - D\big(G(z)\big)\Big)$$

$$J^{(G)} = -\frac{1}{2}\mathbb{E}_z \log D\big(G(z)\big)$$

Encouraging "Generator" to minimize this cost

Training GANs

❖ Training GAN describes a game rather than optimization problem

- Both players have cost functions that are defined in terms of both player's parameters
- Discriminator wishes to minimize $J^{(D)}\left(\theta^{(D)}, \theta^{(G)}\right)$ and must do so while controlling only $\theta^{(D)}$
- Generator wishes to minimize $J^{(G)}\left(\theta^{(D)}, \theta^{(G)}\right)$ and must do so while controlling only $\theta^{(G)}$

Because each player's cost depends on the other player's parameters, but each player cannot control the other player's parameters, this scenario is most straightforward to describe as a **game rather than as an optimization problem**

Training GANs

❖ The solution to a game is Nash equilibrium

- The solution to an **optimization problem** is a (local) minimum, a point in parameter space where all neighboring points have greater or equal cost

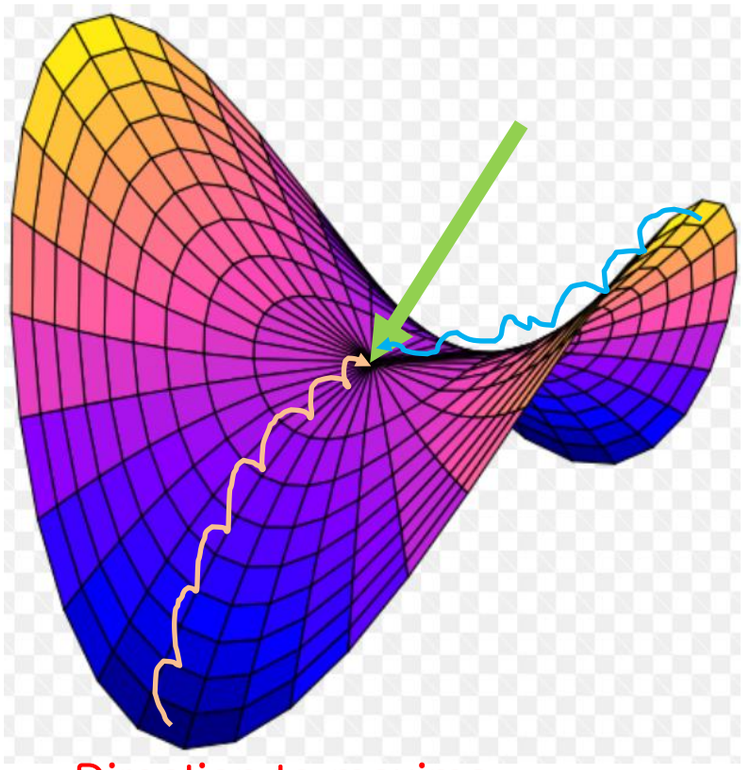
- **The solution to a game is a Nash equilibrium**

- Nash equilibrium is a tuple $(\theta^{(D)}, \theta^{(G)})$ that is a local minimum of $J^{(D)}$ with respect to $\theta^{(D)}$ and a local minimum of $J^{(G)}$ with respect to $\theta^{(G)}$

Research Issues

❖ Non-convergence

- GANs require finding **the equilibrium to a game** with two players
- Optimization algorithms often approach a **saddle point** or local minimum rather than a global minimum



Research Issues

❖ Non-convergence (*Cont'd*)

- For some games, simultaneous gradient descent does converge, and for others, it does not
- For GANs, there is no theoretical prediction as to whether simultaneous gradient descent should converge or not
- Developing algorithms guaranteed to converge remain important open research problem

Research Issues

❖ Non-convergence (*Cont'd*)

- For some games, simultaneous gradient descent does converge, and for others, it does not
- For GANs, there is no theoretical prediction as to whether simultaneous gradient descent should converge or not
- Developing algorithms guaranteed to converge remain important open research problem

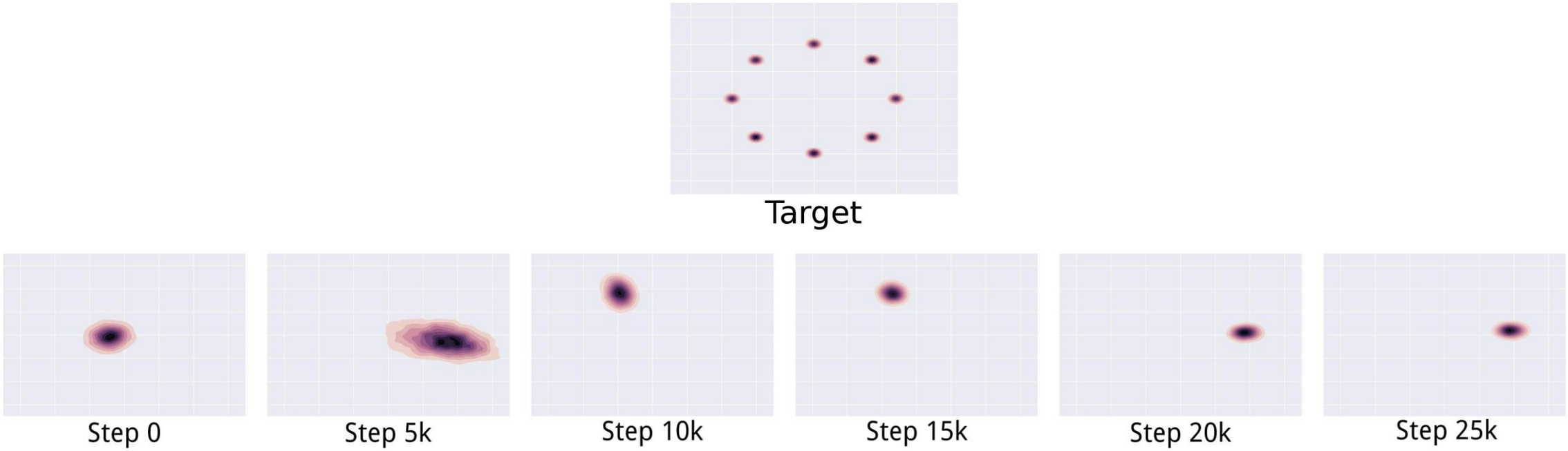
Research Issues

❖ Mode Collapse

- Problem that occurs when the generator learns to map several different input **z** values to the same output point
- **Partial mode collapse** : Generator makes multiple images that contain the same color or texture themes, or multiple images containing different views of the same object

Research Issues

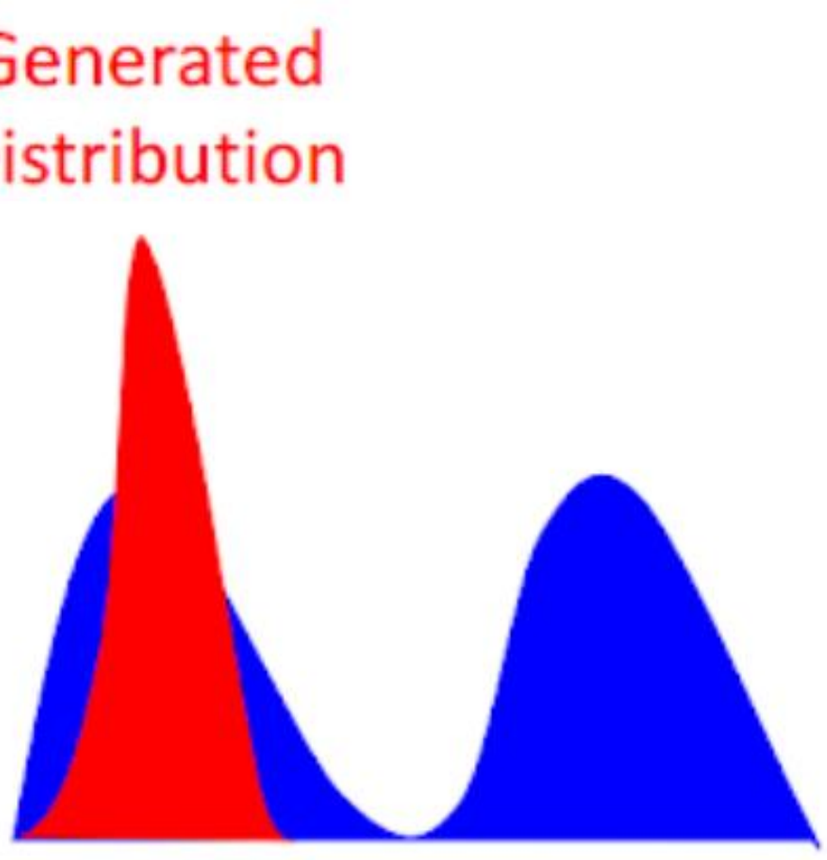
❖ Mode Collapse (*Cont'd*)



Rather than converging to a distribution containing all of the modes in the training set, generator only ever produces a single mode at a time, cycling between different modes as the discriminator learns to reject each one

Research Issues

❖ Mode Collapse (*Cont'd*)



Conclusion

- GANs are generative models that use supervised learning to approximate an intractable cost function
- GANs can simulatemany cost functions, including the one used for maximum likelihood
- Finding Nash equilibria in high-dimensional, continuous, non-convex games is an important open research problem
- GANs are a key ingredient of PPGNs, which are able to generate compelling high resolution samples from diverse image classes