Generative Adversarial Nets

2014 NIPS Workshop on Perturbations, Optimization, and Statistics --- Ian Goodfellow

Generative Adversarial Nets

Ian Goodfellow

Jean Pouget-Abadie

Mehdi Mirza

Yoshua Bengio

Mini-Max Two-Player Game

Strategy: specification of which moves you make in which circumstances

Equilibrium: each player's strategy is the best possible for their opponent's strategy.

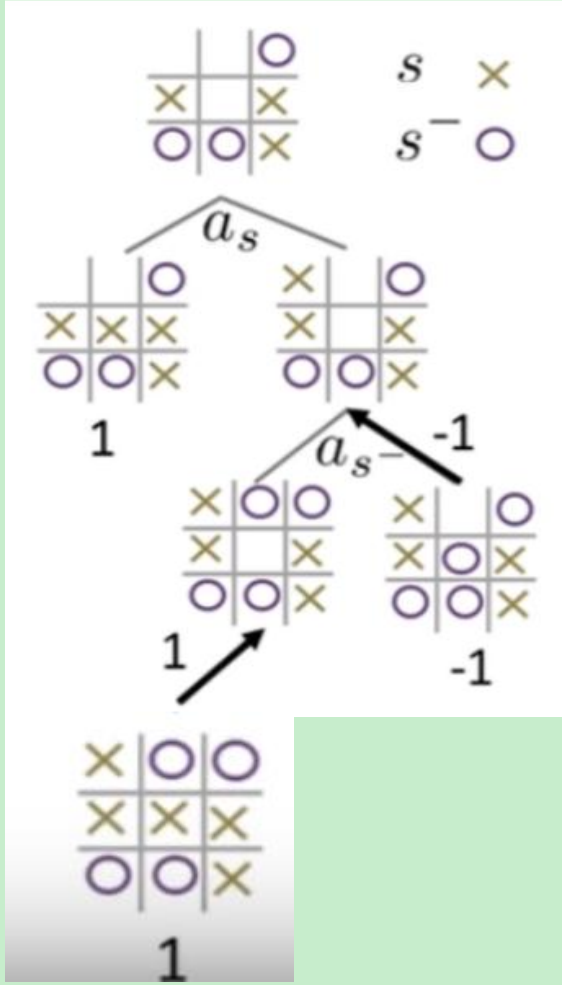
Example: Rock-paper-scissors:
- *Mixed strategy equilibrium*
- Choose you action at random

		Your opponent		
		Rock	Paper	Scissors
You	Rock	0	-1	1
	Paper	1	0	-1
	Scissors	-1	1	0

Mini-Max Two-Player Game

$$v_s = \max_{a_s} \min_{a_{s^-}} v_s(a_s, a_{s^-})$$

- s : the current player
- s^- : the opponent
- a_s : the action taken by current player
- a_{s^-} : the action taken by opponent
- v_s : the value function of current player



- Can we design a game with a mixed-strategy equilibrium that forces one player to learn to generate from the data distribution?

Adversarial nets framework

- A game between two players:
  1. Discriminator D
  2. Generator G
- D tries to discriminate between:
  - A sample from the data distribution.
  - And a sample from the generator G.
- G tries to “trick” D by generating samples that are hard for D to distinguish from data.

2 Introduction

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models:

a **generative model *G*** Generate new samples that are as similar as the data

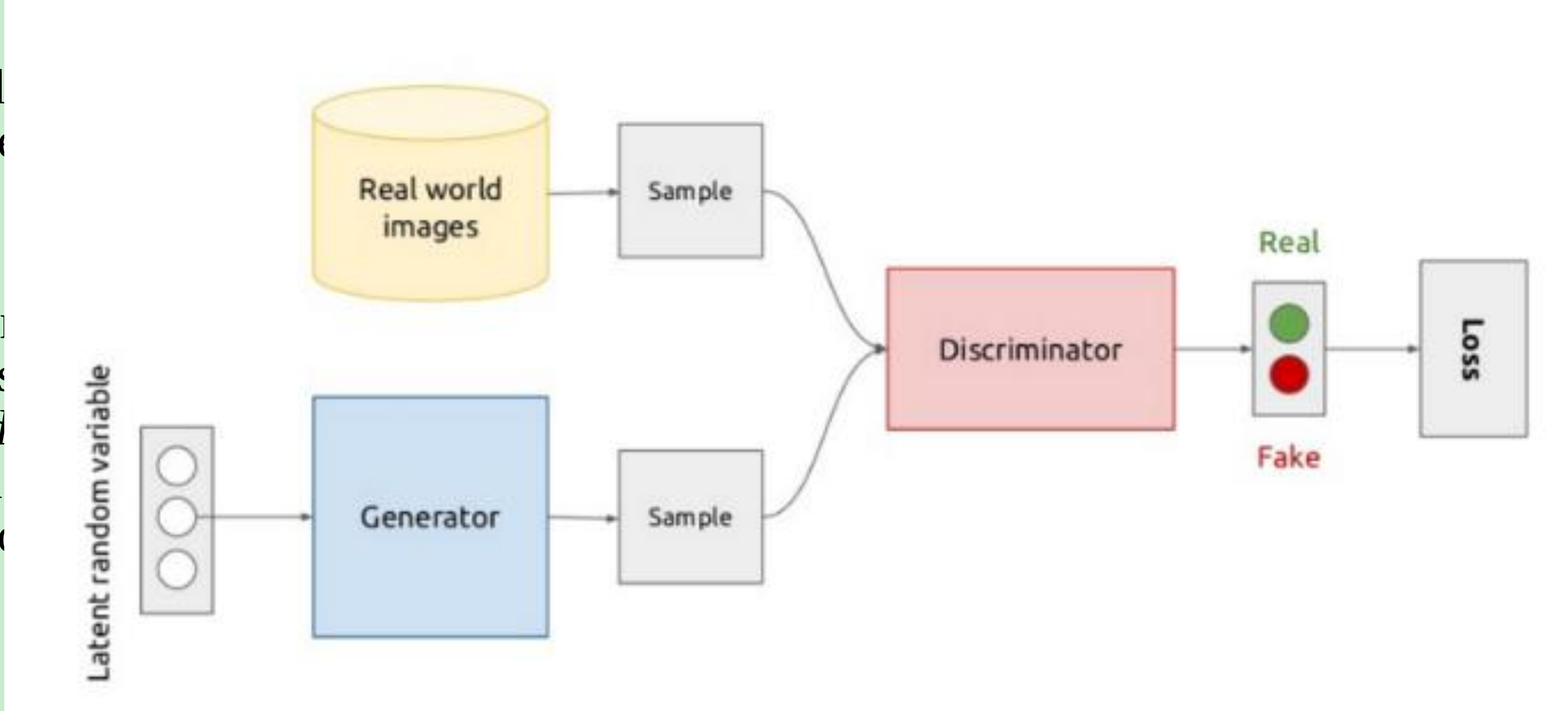
a **discriminative model *D*** that estimates the probability that a sample came from the training data rather than *G*.

In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to 1/2 everywhere

data distribution. The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistiguishable from the genuine articles.

2 Introduction

In … st an
ad … om
the



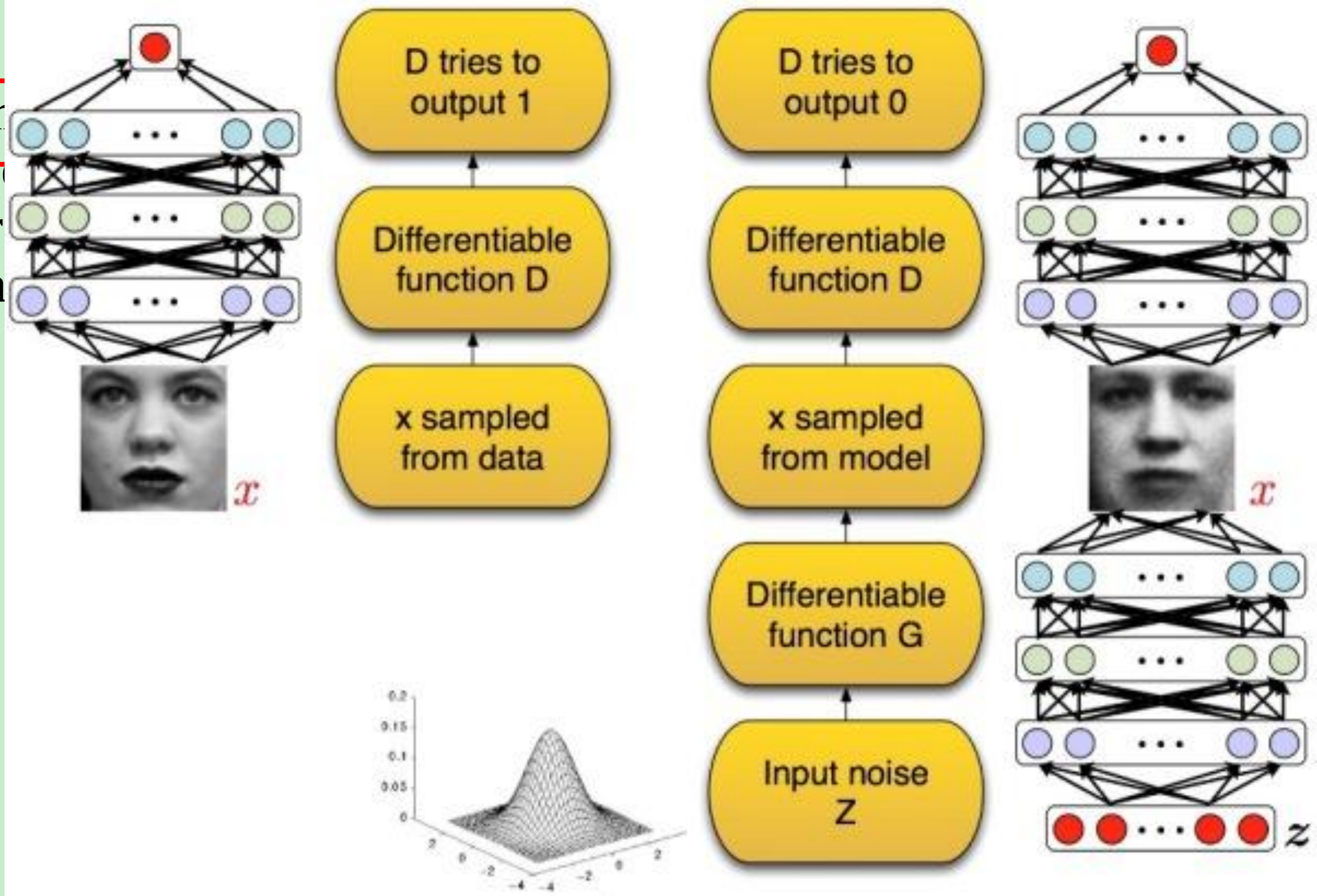
In
sa … he
dis … as
ad
su … ve
mo

3 Adversarial Net

To learn the generator's distributio
$p_z(z)$, then represent a mapping to
by a multilayer perceptron with par
$D(\mathbf{x};\theta_d)$ that outputs a **single scala**
data rather than p_g .

1) We train D to maximize the probability of assigning the correct label to both training examples and samples from G.

2) We simultaneously train G to minimize log(1-D(G(z))).

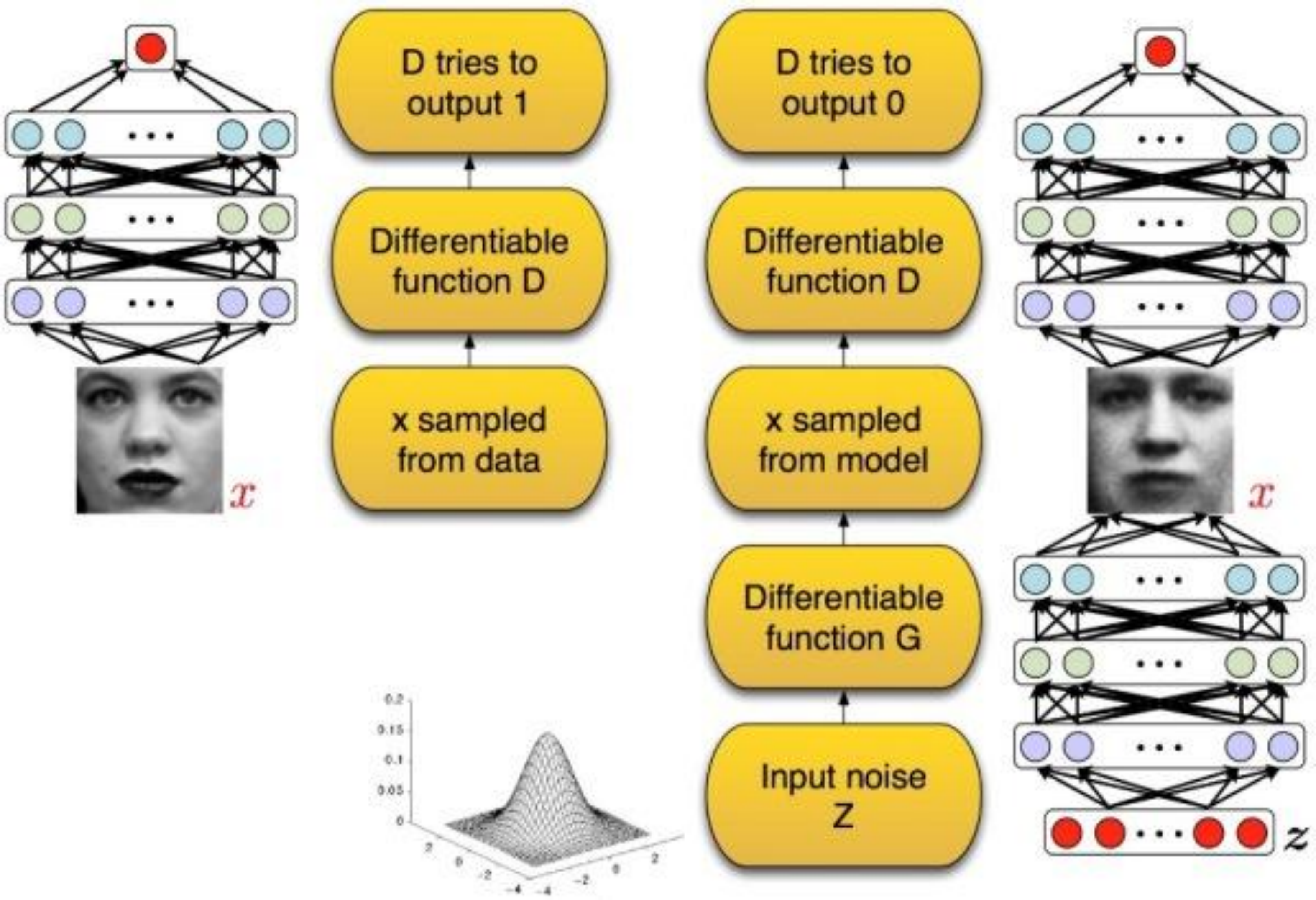


3 Adversarial Net

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

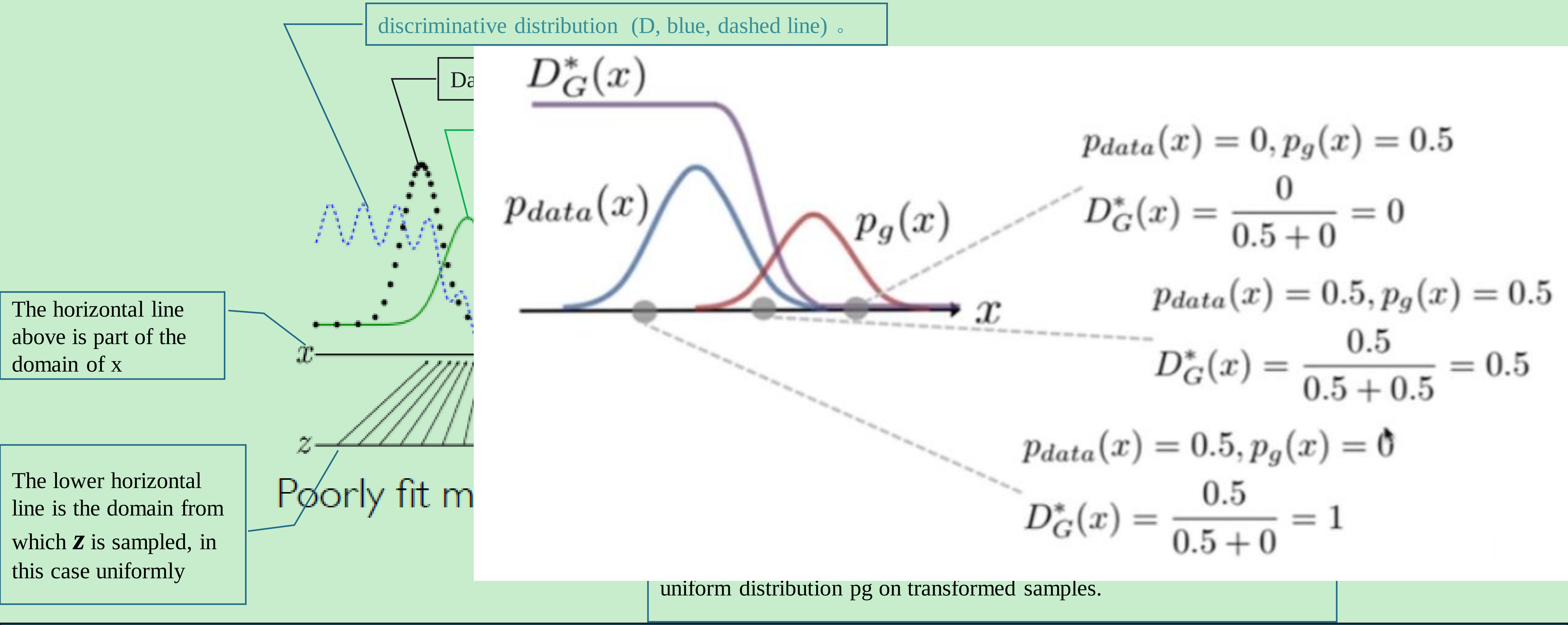
Until recently, mo metric
specification of a ximiz-
ing the log likelih zmann
machine [25]. Su equire
numerous approxi pment
of "generative ma o gen-
erate samples fron ple of
a generative mach us ap-
proximations requ achine
by eliminating the

Our work backpro hat



3 Adversarial Net

See Figure 1 for a less formal, more pedagogical explanation of the approach.



3 Adversarial Net

See Figure 1 for a less formal, more pedagogical explanation of the approach.

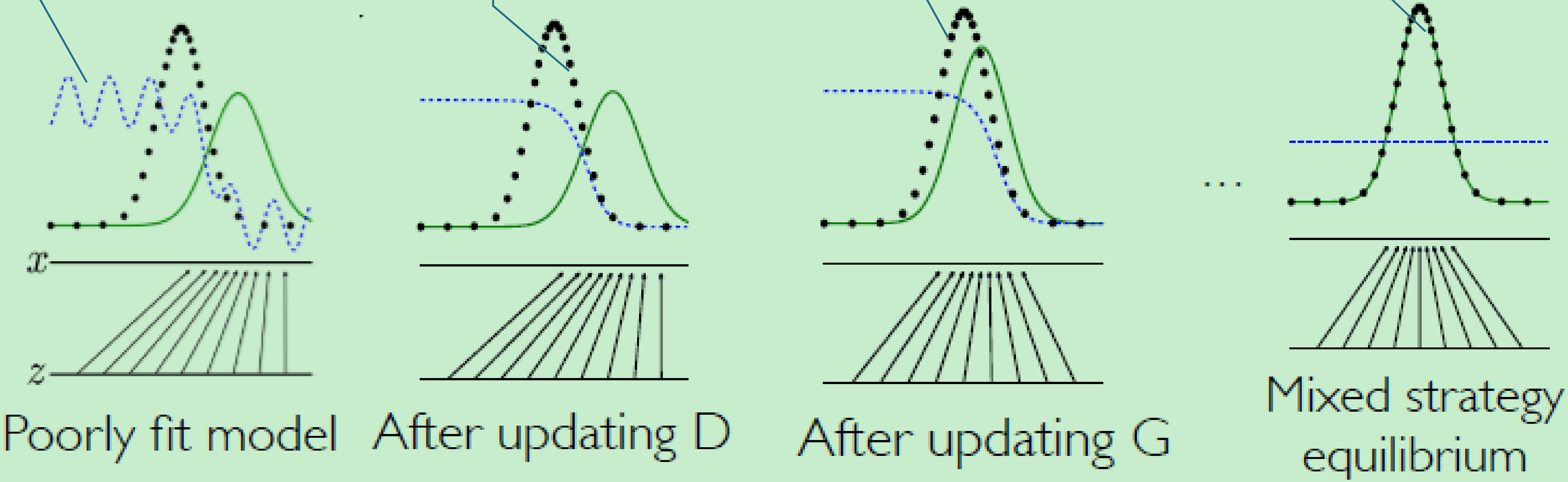
Consider an adversarial pair near convergence : pg is similar to pdata andDis a partially accurate classifier.

In the inner loop of the algorithm D is trained to discriminate samples from data, converging to

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

After an update to G, gradient of D has guided G(z) to flow to regions that are more likely to be classified as data

fter several steps of training, if GandDhave enough capacity, they will reach a point at which both cannot improve because pg = pdata. The discriminator is unable to differentiate between the two distributions, i.e. D(x) = 1/2.



3 Adversarial Net

we alternate be… …esults in D
being maintain… …he procedure
is formally pres…

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

---

for number of training iterations **do**

  for k steps **do**

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

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

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

4 Theoretical Results

The generator G implicitly defines a probability distribution p_g as the distribution of the samples $G(z)$ obtained when $z \sim p_z$.

theoretical properties
(assuming infinite data ,infinite model capacity direct updating of generator's distribution)

4.1. Unique Global optimum.
4.2. Optimum corresponds to data distribution.
4.3. Convergence to optimum guaranteed.

4 Theoretical Results

4.1 **Global Optimality of** $p_g = p_{data}$

We first consider the optimal discriminator D f

Proposition 1. For G fixed, the optimal discri

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

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

$$= \int_x p_{data}(x) log(D(x))dx + \int_z p_z(z) log(1 - D(G(z)))dz$$

$$x = G(z) \Rightarrow z = G^{-1}(x) \Rightarrow dz = (G^{-1})'(x)dx$$

$$\Rightarrow p_g(x) = p_z(G^{-1}(x))(G^{-1})'(x)$$

$$= \int_x p_{data}(x) log(D(x))dx + \int_x p_z(G^{-1}(x)) log(1 - D(x))(G^{-1})'(x)dx$$

$$= \int_x p_{data}(x) log(D(x))dx + \int_x p_g(x) log(1 - D(x))dx$$

$$= \int_x p_{data}(x) log(D(x)) + p_g(x) log(1 - D(x))dx$$

Proof. The training criterion for the discriminator D,
quantity $V(G,D)$

$$D^*(G) = \underset{D}{MAX}\ V(G,D) = \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_z p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz$$

$$= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D$$

$p_g = p_{data}$

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} = \frac{1}{2}$$

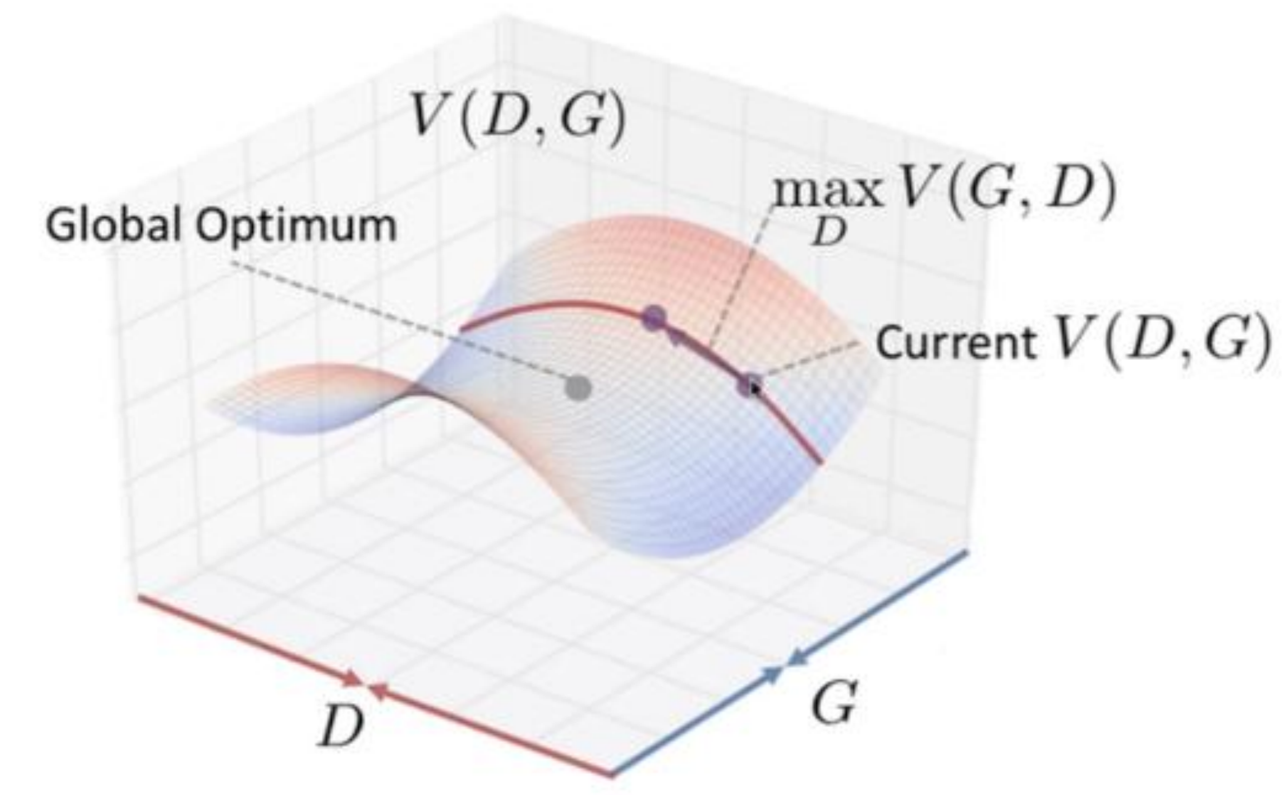
For any $(a,b) \in \mathbb{R}^2 \setminus \{0,0\}$, the function $y \to a\log(y) + b\log(1$ -
$[0,1]$ at $\frac{a}{a+b}$. The discriminator does not need to be defined outside of $Supp(p_{\text{data}}) \cup Supp(p_g)$,
concluding the proof. □

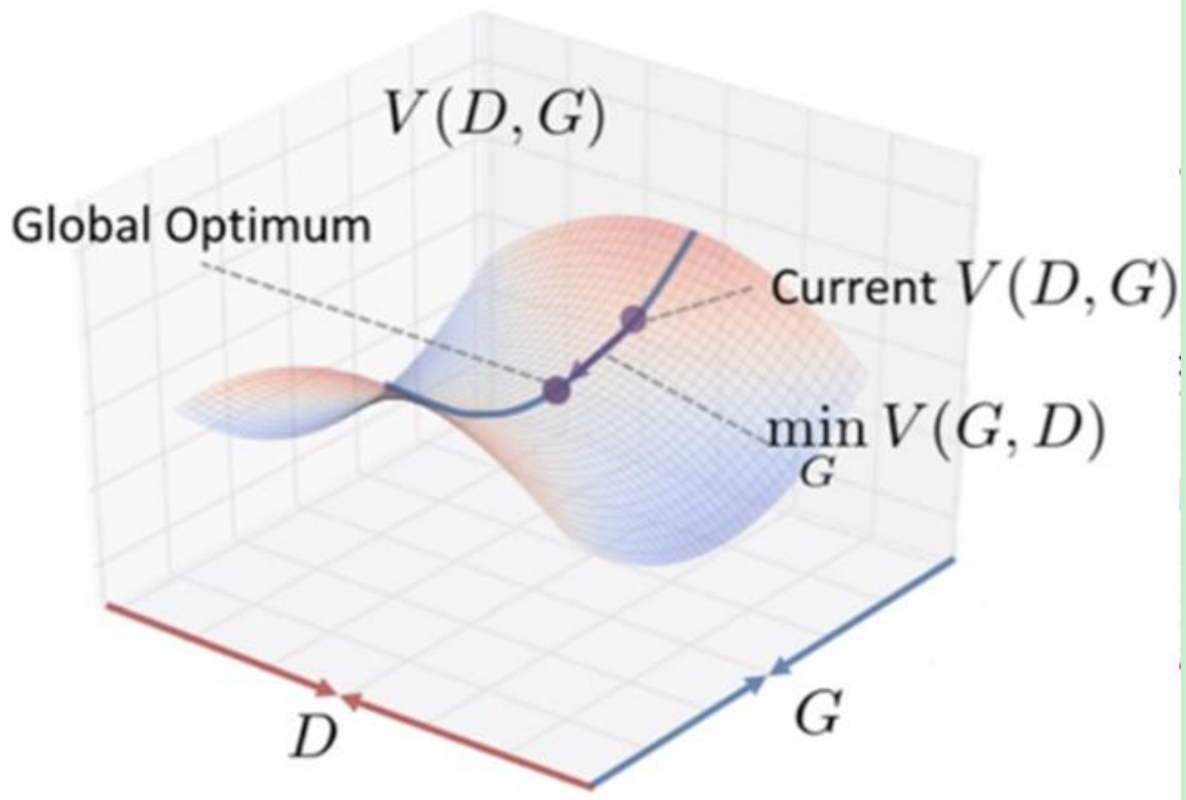
4 Theoretical Results

Theorem
only if p

$$
\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \max_D \int_x p_{data}(x) log(D(x)) + p_g(x) log(1 - D(x)) dx \\
&= \int_x p_{data}(x) log(D^*_G(x)) + p_g(x) log(1 - D^*_G(x)) dx \\
&= \int_x p_{data}(x) log\left(\frac{p_{data}(x)}{p_{data}(x) + p_g(x)}\right) + p_g(x) log\left(\frac{p_g(x)}{p_{data}(x) + p_g(x)}\right) dx \\
&= \int_x p_{data}(x) log\left(\frac{p_{data}(x)}{\frac{p_{data}(x)+p_g(x)}{2}}\right) + p_g(x) log\left(\frac{p_g(x)}{\frac{p_{data}(x)+p_g(x)}{2}}\right) dx - log(4) \\
&= KL\left[p_{data}(x) \middle|\middle| \frac{p_{data}(x) + p_g(x)}{2}\right] + KL\left[p_g(x) \middle|\middle| \frac{p_{data}(x) + p_g(x)}{2}\right] - log(4)
\end{aligned}
$$

4 Theoretical Results





4 Theoretical Results

4.2 Convergence of Algorithm 1

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}

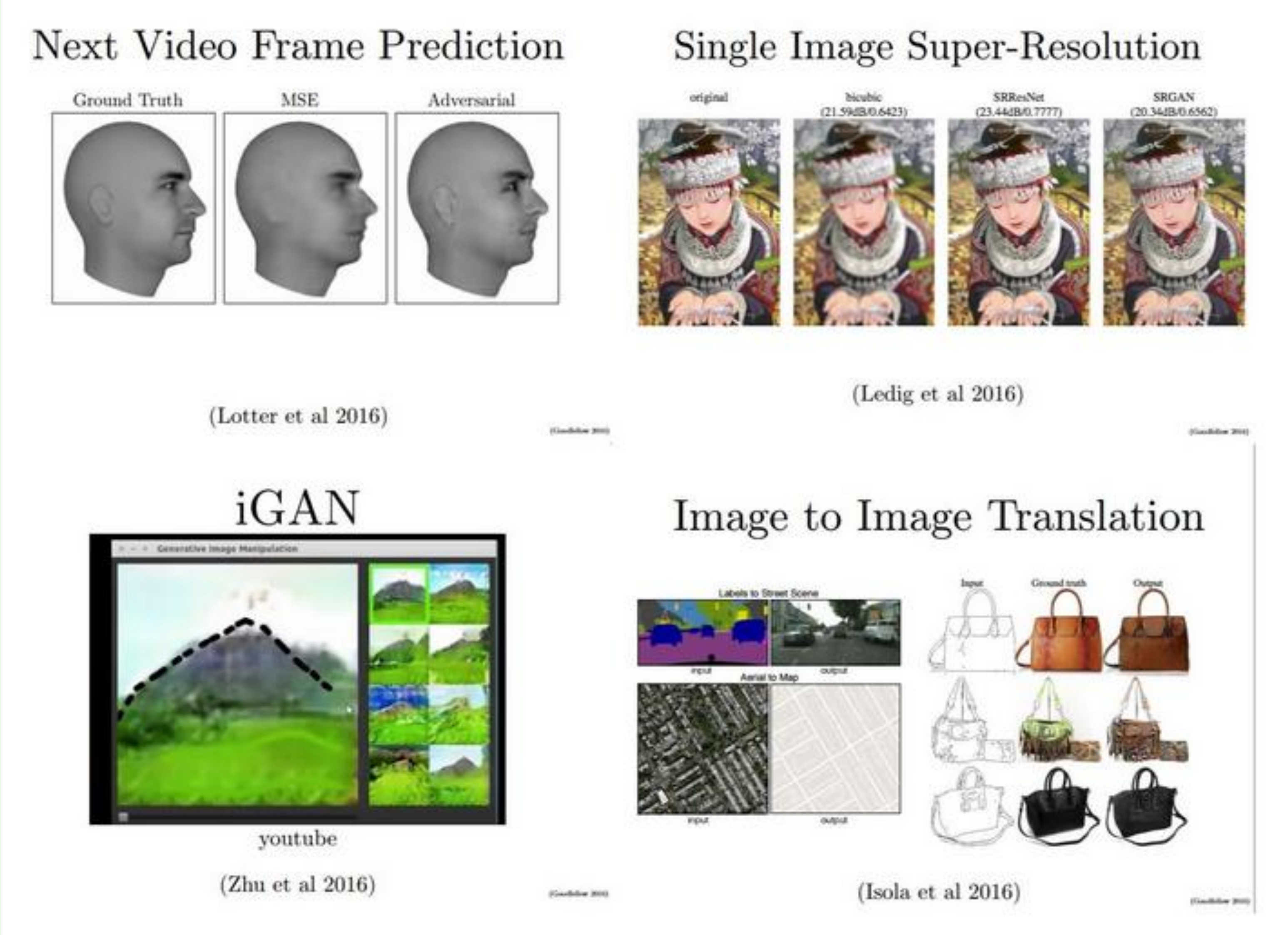
Proof. Consider $V(G, D) = U(p_g, D)$ as a function of p_g as done in the above criterion. Note that $U(p_g, D)$ is convex in p_g. The subderivatives of a supremum of convex functions include the derivative of the function at the point where the maximum is attained. In other words, if $f(x) = \sup_{\alpha\in\mathcal{A}} f_\alpha(x)$ 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. $\square$

4 Theoretical Results

Why study generative models?

- Excellent test of our ability to use high-dimensional, complicated probability distributions
- Simulate possible futures for planning or simulated RL
- Missing data
    - Semi-supervised learning
- Multi-modal outputs
- Realistic generation tasks

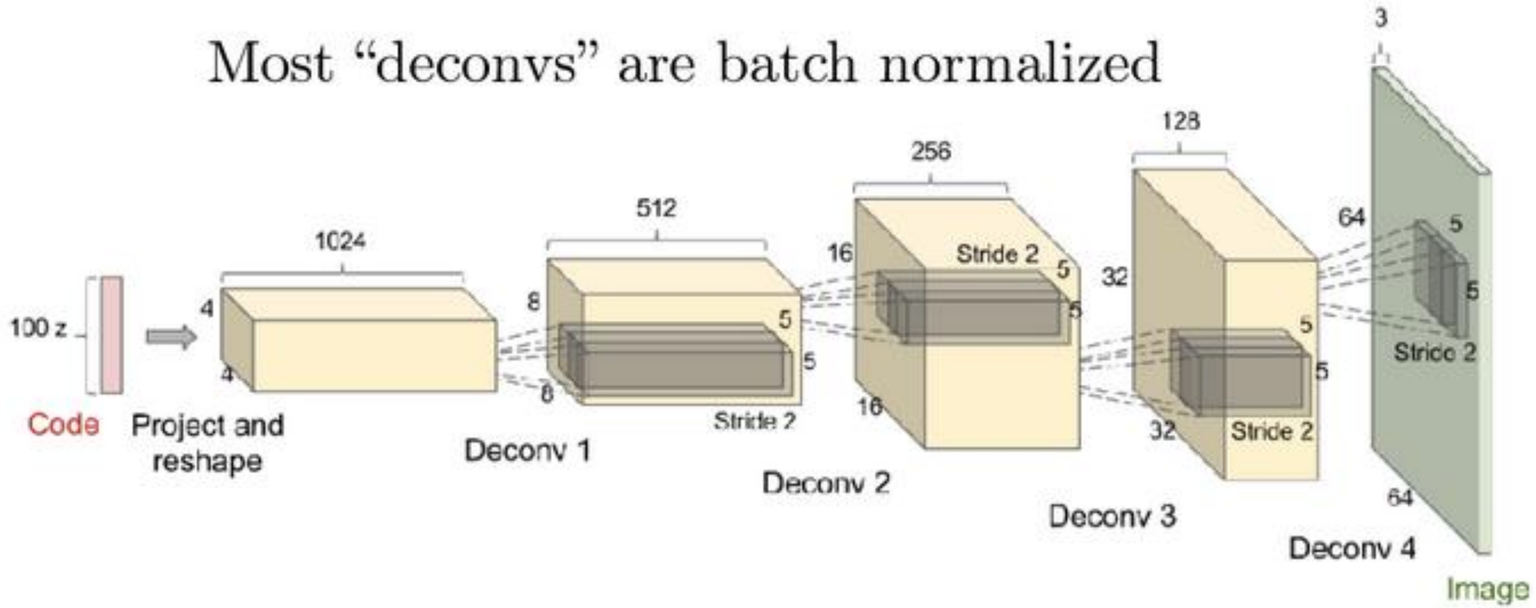
4 Theoretical Results



4 Theoretical Results

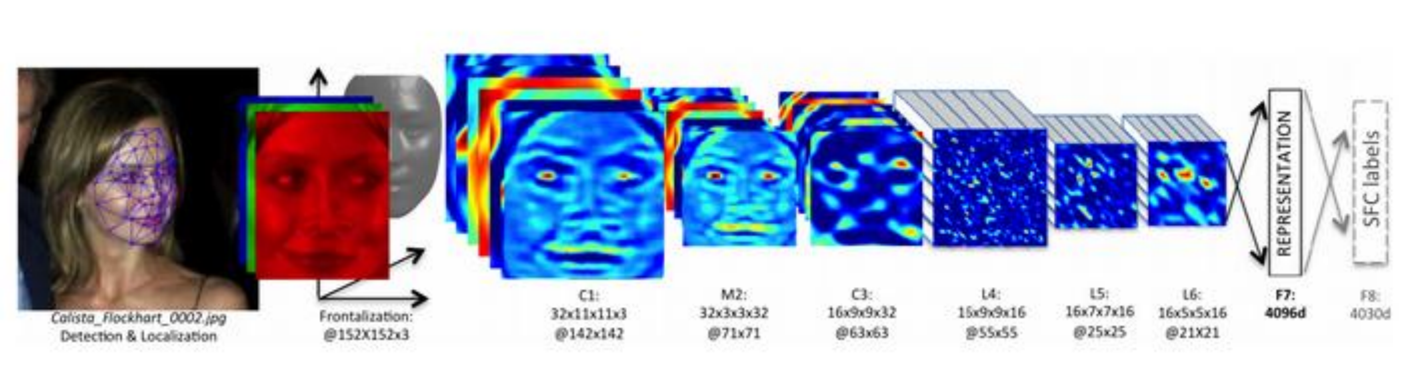
DCGAN Architecture

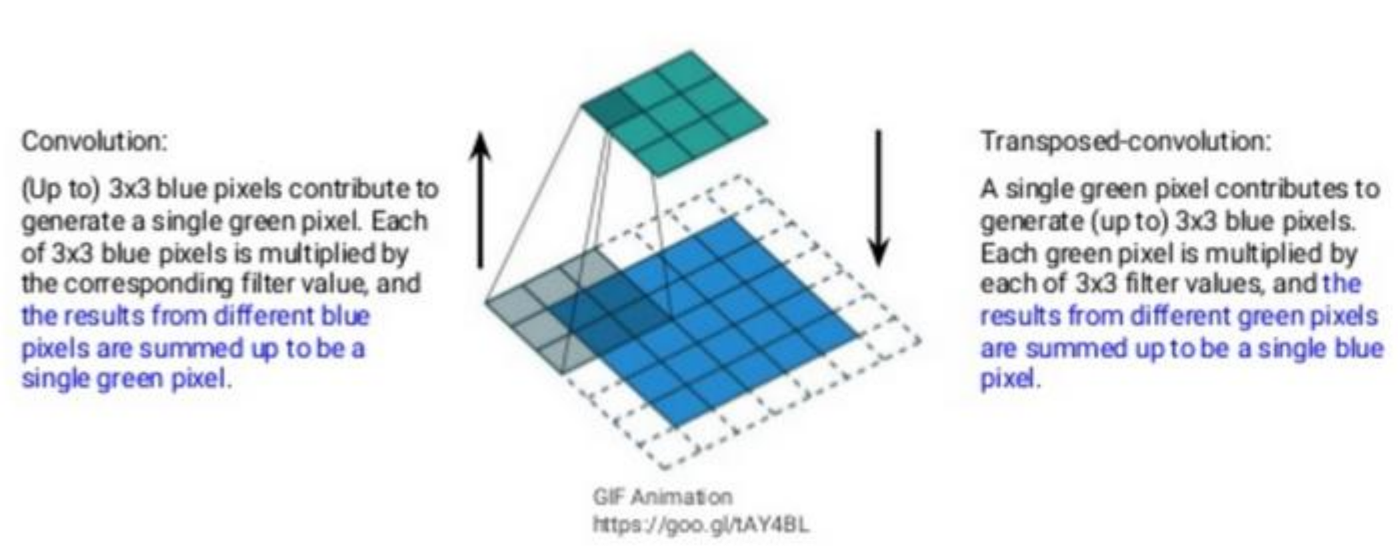
Most "deconvs" are batch normalized



(Radford et al 2015)

4 Theoretical Results





Thank

You