Boundary-Seeking Generative Adversarial Networks (BGANs)

Hjelm, R. Devon, et al.

Presenting: Yevgeny Tkach

2019 Spring @
https://qdata.github.io/deep2Read/

Executive Summary

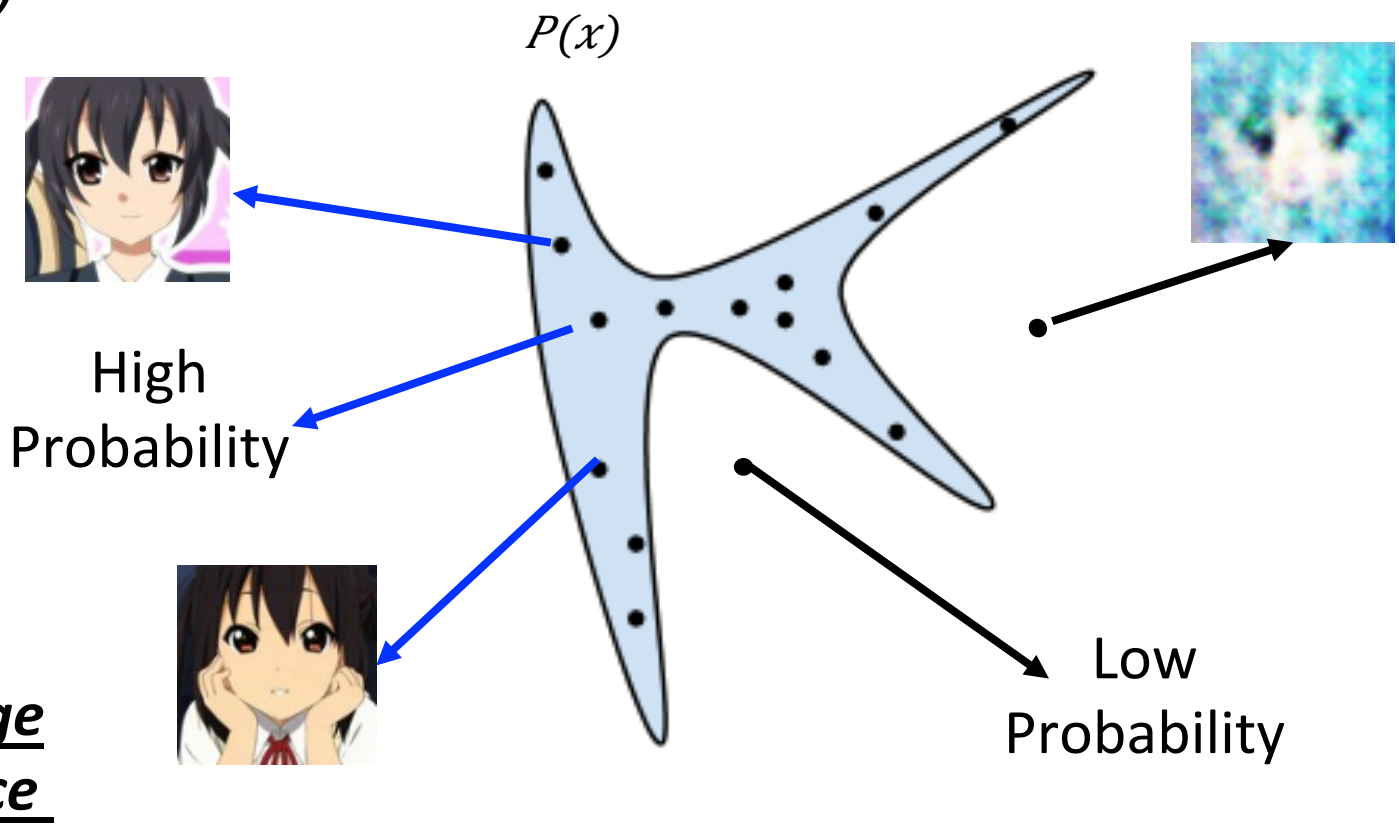
- BGAN is framework that allows GAN to generate both discrete and continuous data
- Discriminator is trained by maximizing the f-divergence between the data and generated distributions
- Generator is trained to minimize the f-divergence between the generated distribution and a self-normalized importance sampling (SIS) estimation of the data distribution
- Experiments show state of the art results in training GANs on discrete data generation and high stability in training GANs with continuous data.

Outline

- GAN − Basic Idea
- f - GAN Introduction
- Importance Sampling – Detour
- BGAN

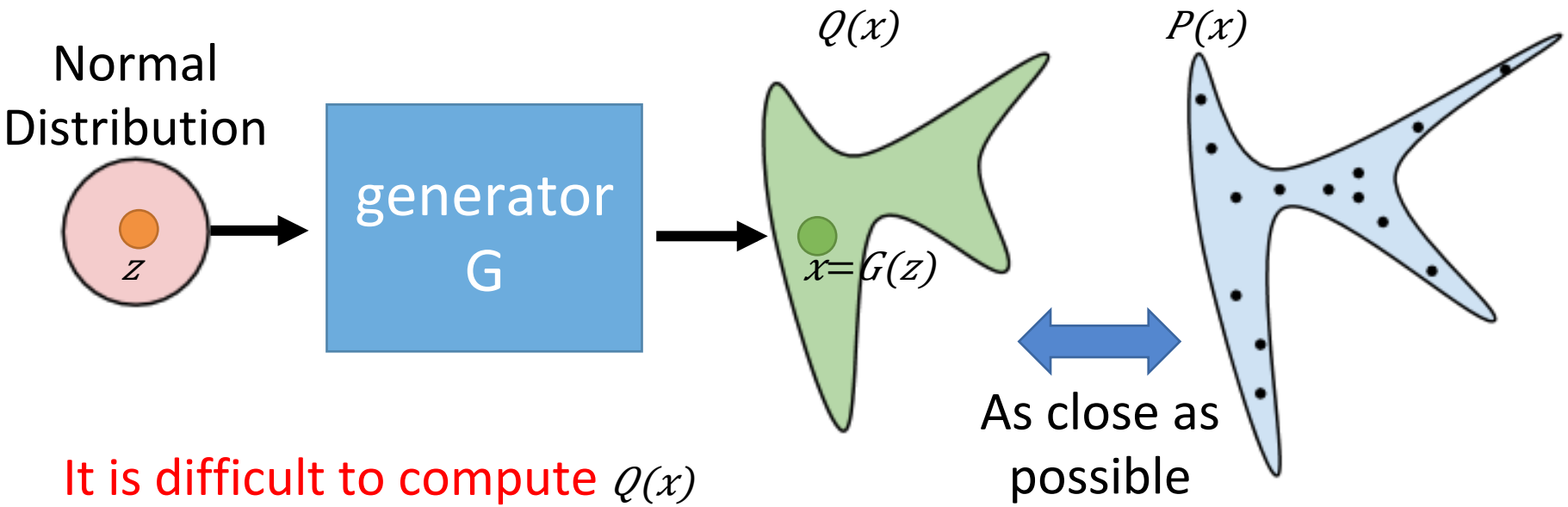
Basic Idea of GAN

- The data we want to generate has a distribution $P(x)$



Basic Idea of GAN

- A generator G is a network. The network defines a probability distribution.

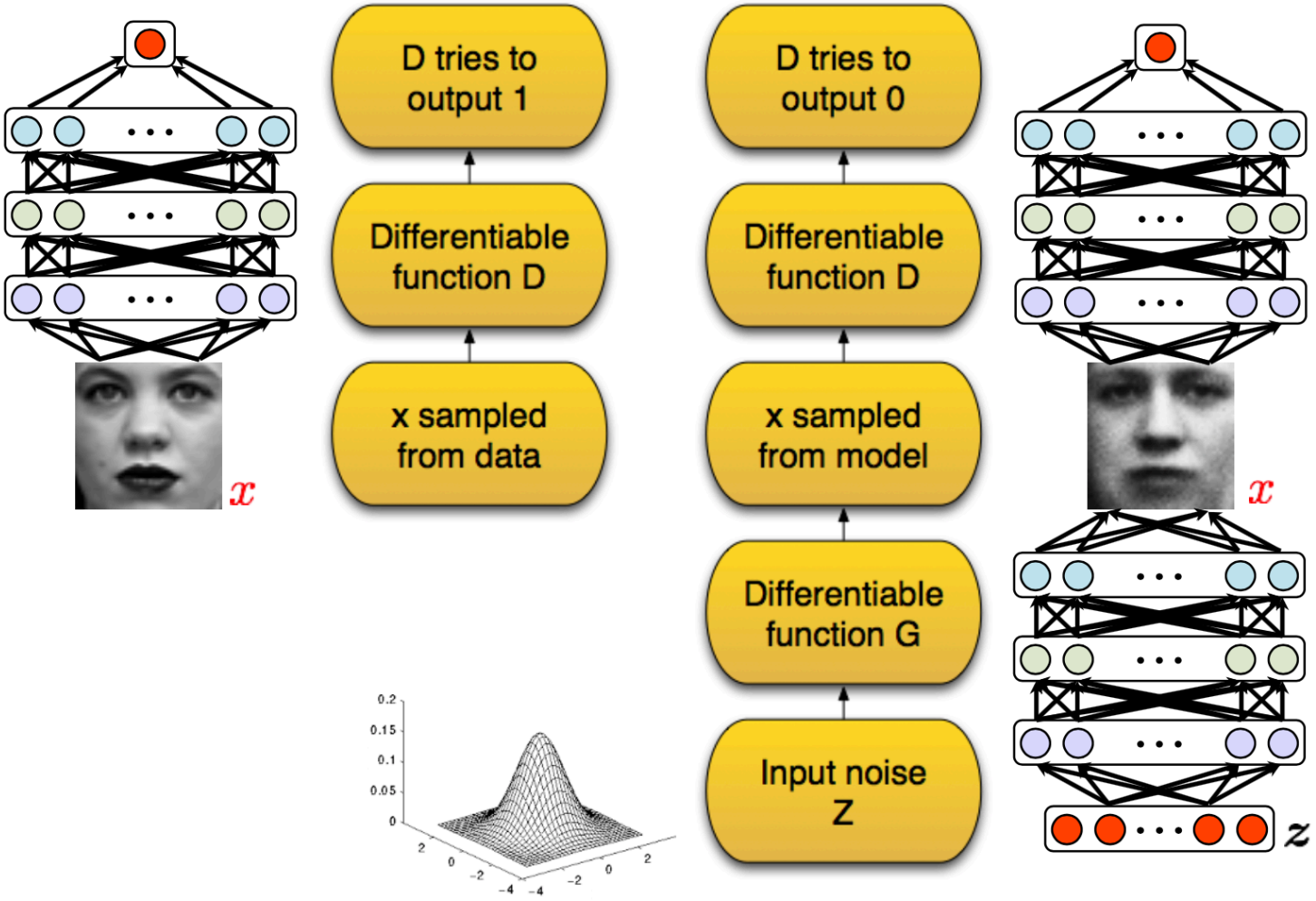


It is difficult to compute $Q(x)$

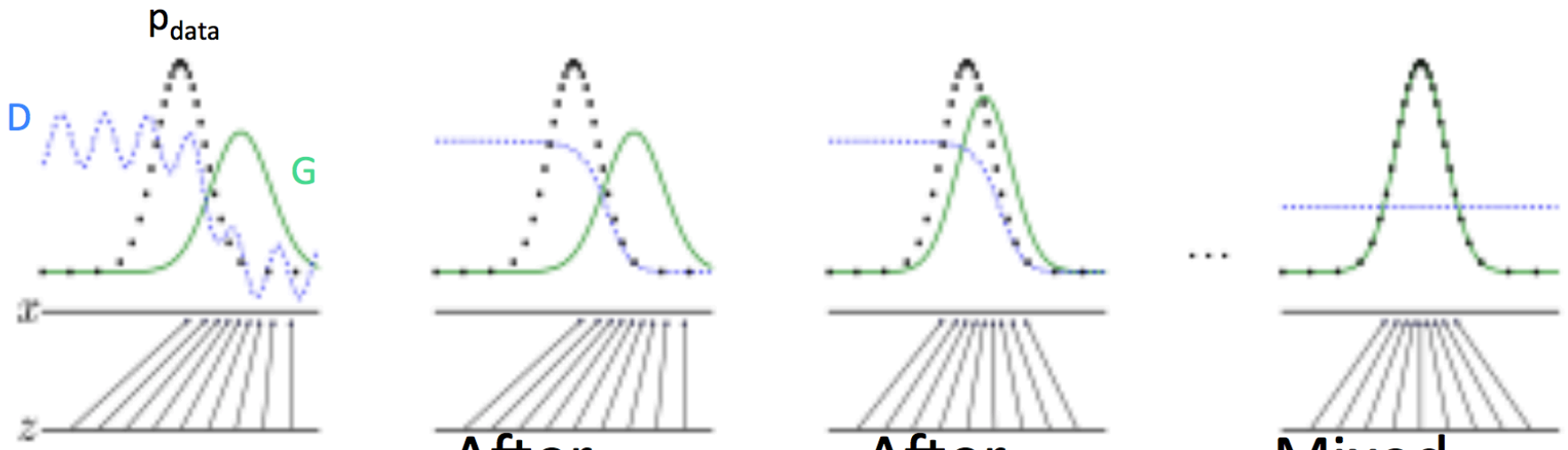
We can only sample from the distribution.

https://blog.openai.com/generative-models/

Basic Idea of GAN



GAN Intuition



Poorly fit model

After updating D

After updating G

Mixed strategy equilibrium

GAN Formally

- Value Function:

$$V(\mathbb{P}, G_\theta, D_\phi) = E_{x\sim P}[\log D(x)] + E_{x\sim Q}[\log(1-D(x))]$$
$$= E_{x\sim P}[\log D(x)] + E_{z\sim h(z)}[\log(1-D(G(z)))]$$

- Monte-Carlo Approximation:

$$V(\mathbb{P}, G_\theta, D_\phi) = 1/m \sum_{i=1}^{m} \log D(x^i) + 1/m \sum_{i=1}^{m} \log(1-D(G(z^i)))$$

- Discriminator target:

$$\max_\phi V(\mathbb{P}, G_\theta, D_\phi)$$

- Generator target:

$$\min_\theta \max_\phi V(\mathbb{P}, G_\theta, D_\phi)$$

Algorithm Initialize ϕ_d for D and θ_g for G

- In each training iteration:

Learning D — Repeat k times

- Sample m examples $\{x^1, x^2, ..., x^m\}$ from data distribution $P(x)$
- Sample m noise samples $\{z^1, z^2, ..., z^m\}$ from the prior $h(z)$
- Obtaining generated data $\{x^1, x^2, ..., x^m\}$, $x^i = G(z^i)$
- Update discriminator parameters θ_d to maximize
  - $V = 1/m \sum_{i=1}^{m} \log D(x^i) + 1/m \sum_{i=1}^{m} \log(1 - D(x^i))$
  - $\phi_d \leftarrow \phi_d + \eta \nabla V(\phi_d)$

Learning G — Only Once

- Sample another m noise samples $\{z^1, z^2, ..., z^m\}$ from the prior $P_{prior}(z)$
- Update generator parameters θ_g to minimize
  - $V = 1/m \sum_{i=1}^{m} \log D(x^i) + 1/m \sum_{i=1}^{m} \log(1 - D(G(z^i)))$
  - $\theta_g \leftarrow \theta_g - \eta \nabla V(\theta_g)$

f - GAN Introduction

- Sebastian Nowozin, Botond Cseke, Ryota Tomioka, "***f-GAN***: Training Generative Neural Samplers using Variational Divergence Minimization", NIPS, 2016

- One sentence: you can use any f-divergence

f-divergence

P and Q are two distributions. $p(x)$ and $q(x)$ are the density functions respectively.

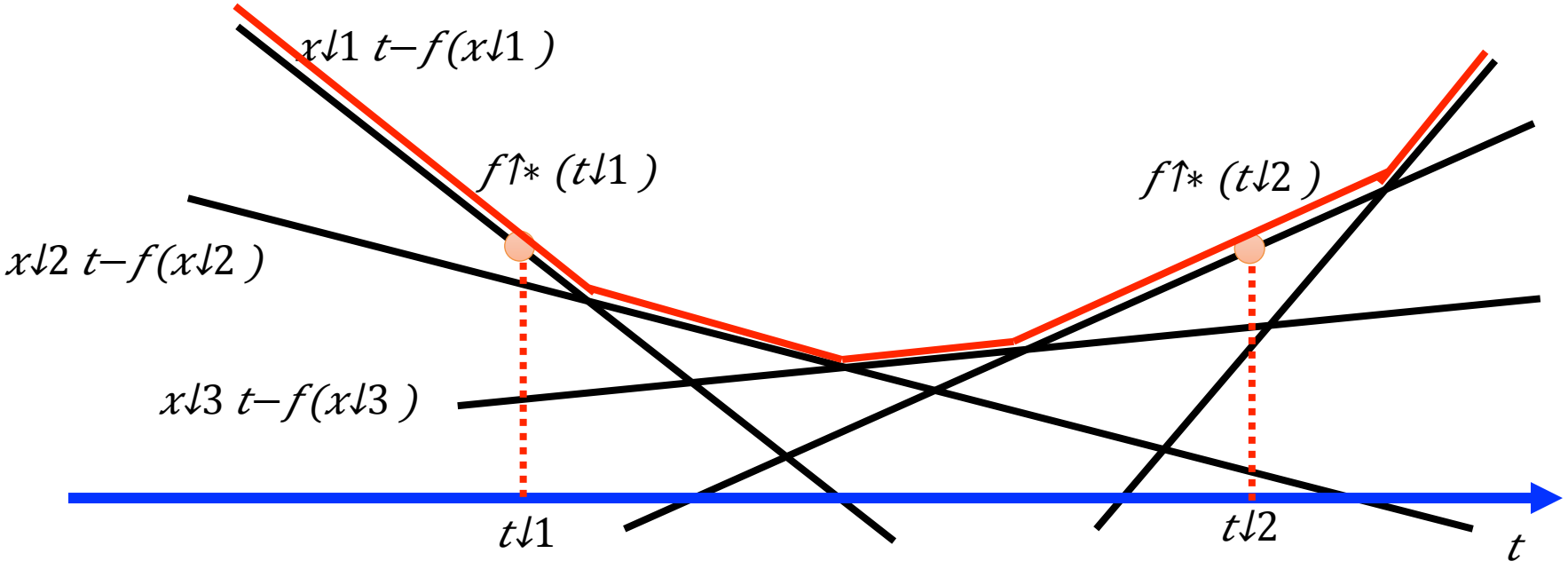
$$D_f(P||Q) = \int_x q(x) f\left(\frac{p(x)}{q(x)}\right) dx$$

f is convex

$f(1) = 0$

- Every convex function f has a conjugate function f*

$$f^*(t) = \max_{x \in dom(f)} \{xt - f(x)\} \quad \longleftrightarrow \quad f(x) = \max_{t \in dom(f^*)} \{xt - f^*(t)\}$$



Connection with GAN

$$f^*(t)=\max_{x\in dom(f)}\{xt-f(x)\} \quad \longleftrightarrow \quad f(x)=\max_{t\in dom(f^*)}\{xt-f^*(t)\}$$

$$D_f(P||Q)=\int_x q(x) f(p(x)/q(x))dx$$

$p(x)/q(x)$ $\qquad$ $p(x)/q(x)$

$$=\int_x q(x)(\max_{t\in dom(f^*)}\{p(x)/q(x)\ t-f^*(t)\})dx$$

D is a function whose input is x, and output is t

$$\geq \max_{D\in\mathcal{D}} \int_x q(x)(p(x)/q(x)\ D(x)-f^*(D(x)))dx$$

$$=\max_{D\in\mathcal{D}} \int_x p(x)D(x)dx-\int_x q(x)f^*(D(x))dx$$

Connection with GAN

$$\geq \max_{T} D \{ \int_x p(x) D(x) dx - \int_x q(x) f^*(D(x)) dx \}$$

$$D_f(P||Q)$$

$$= \max_{T} D \{ E_{x \sim P}[D(x)] - E_{x \sim Q}[f^*(D(x))] \}$$

Samples from P Samples from Q

$$D_f(P||Q) \geq \max_{T} D \{ E_{x \sim P}[v \circ D(x)] - E_{x \sim Q}[f^*(v \circ D(x))] \}$$

$$G^* = arg\min_{T} G \square D_f(P||Q)$$
$$= arg\min_{T} G \square \max_{T} D \square \{ E_{x \sim P}[v \circ D(x)] - E_{z \sim h(z)}[f^*(v \circ D(G(z)))] \}$$

GAN value function:

$$V(\mathbb{P}, G_\theta, D_\phi) = E_{x \sim P}[\log D(x)] + E_{z \sim h(z)}[\log(1 - D(G(z)))]$$

Importance Sampling – Detour

$$E_{x\sim P}[f(x)] = \int f(x)p(x)dx$$
$$= \int f(x)p(x)/q(x)\, q(x)dx$$
$$= \int f(x)w(x)q(x)dx$$
$$= E_{x\sim Q}[f(x)w(x)]$$
$$= E_{x\sim Q}[f(x)w(x)]/E_{x\sim Q}[w(x)]$$
$$w(x) = p(x)/q(x)$$

In case p or q are scaled density functions

$w(x)$ - Importance Weights

Boundary Seeking GAN - BGAN

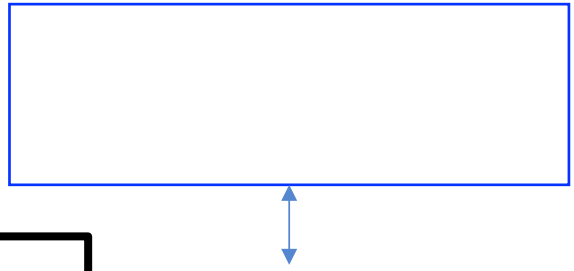
Theorem 1: P *and* Q as in f-GAN, and $D^{*} \in D$ satisfying:

$$D_f(P||Q) = \max_{D} \{E_{x\sim P}[D(x)] - E_{x\sim Q}[f^{*}(D(x))]\}$$

Then: $p(x) = (\partial f^{*}/\partial D)(D^{*}(x))q(x)$

Proof:

$$D_f(P||Q) = E_{x\sim Q}[f(p(x)/q(x))] = E_{x\sim Q}[\sup_{t} \{tp(x)/q(x) - f^{*}(t)\}]$$



p re-written in terms of q and a scaling factor

w(x) $= (\partial f^{*}/\partial D)(D^{*}(x))$ – Importance weights

$$p(x)/q(x) = \partial f^{*}(t)/\partial t$$

Boundary Seeking GAN - BGAN

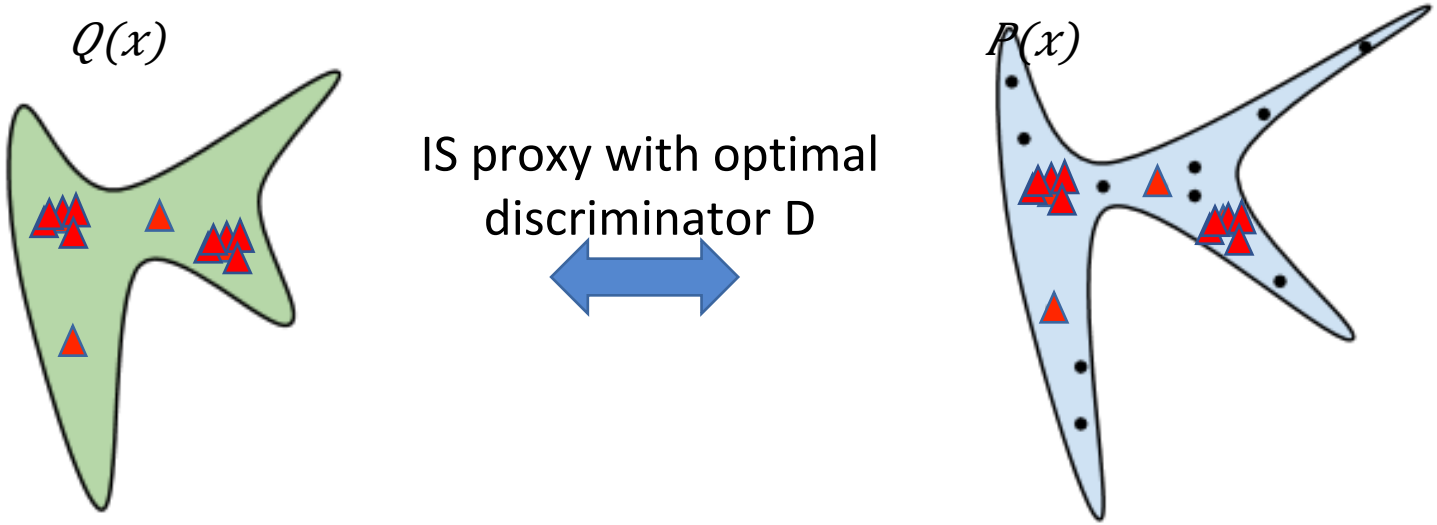
BGAN suggests to use the **divergence** between $q(x)$ and the self normalized importance sampling (IS) estimation of $p(x)$:

$p(x) = w(x) / \beta \, q(x)$

Where:

$\beta = E{\downarrow}x \sim Q \, [w(x)]$

BGAN – IS intuition



- Divergence between ▲ should have lower variance than if taking arbitrary samples from $P(x)$
- Since $G(z)$ defines a distribution that x is sampled from - the variance can be further decreased by taking multiple samples from the same z

BGAN – reduced variance

We can restate everything in terms of conditional distributions:

- $q(x) = \int_{Z\uparrow} g(x|z)h(z)dz$
- $g(x|z): Z \rightarrow [0,1]\uparrow d$ - multivariate Bernoulli distribution
- $\alpha(z) = E\downarrow x \sim g(x|z)\,[w(x)]$ - similar to β
- $p\,(x|z) = w(x)/\alpha(z)\; g(x|z)$
- $D\downarrow KL\,(p\,(x)||q(x)) = E\downarrow h(z)\,[D\downarrow KL\,(p\,(x|z)||q(x|z))]$
- $\nabla E\downarrow h(z)\,[D\downarrow KL\,(p\,(x|z)||q(x|z))]$ approximates with two MC

BGAN - Algorithm

Algorithm 1 . Discrete Boundary Seeking GANs

$(\theta, \phi) \leftarrow$ initialize the parameters of the generator and statistic network

repeat

$\hat{x}^{(n)} \sim \mathbb{P}$ ▷ Draw N samples from the empirical distribution

$z^{(n)} \sim h(z)$ ▷ Draw N samples from the prior distribution

$x^{(m|n)} \sim g_\theta(x \mid z^{(n)})$ ▷ Draw M samples from each conditional $g_\theta(x \mid z^{(m)})$ (drawn independently if $\mathbb{P}$ and $\mathbb{Q}_\theta$ are multi-variate)

$w(x^{(m|n)}) \leftarrow (\partial f^\star / \partial T) \circ (\nu \circ F_\phi(x^{(m|n)}))$

$\tilde{w}(x^{(m|n)}) \leftarrow w(x^{(m|n)}) / \sum_{m'} w(x^{(m'|n)})$ ▷ Compute the un-normalized and normalized importance weights (applied uniformly if $\mathbb{P}$ and $\mathbb{Q}_\theta$ are multi-variate)

$\mathcal{V}(\mathbb{P}, \mathbb{Q}_\theta, T_\phi) \leftarrow \frac{1}{N} \sum_n F_\phi(\hat{x}^{(n)}) - \frac{1}{N} \sum_n \frac{1}{M} \sum_m w(x^{(m|n)})$ ▷ Estimate the variational lower-bound

$\phi \leftarrow \phi + \gamma_d \nabla_\phi \mathcal{V}(\mathbb{P}, \mathbb{Q}_\theta, T_\phi)$ ▷ Optimize the discriminator parameters

$\theta \leftarrow \theta + \gamma_g \frac{1}{N} \sum_{n,m} \tilde{w}(x^{(m|n)}) \nabla_\theta \log g_\theta(x^{(m|n)} \mid z)$ ▷ Optimize the generator parameters

until convergence

Boundary Seeking GAN - BGAN

$$D_f(P_{data}||P_G) \geq \max_D \{E_{x\sim P_{data}}[v \circ D(x)] - E_{x\sim P_G}[f^*(v \circ D(x))]\}$$

$$p(x) = w(x)/\beta\, q(x) \qquad w(x) = (\partial f^*/\partial D)\left(D^*(x)\right)$$

Table 1: Important weights and nonlinearities that ensure

Importance weights for f-divergences		
f-divergence	$\nu(y)$	$w(x) = (\partial f^\star/\partial T)(T(x))$
GAN	$-\log\left(1+e^{-y}\right)$	$-\frac{1}{1-e^{-T_\phi}} = e^{F_\phi(x)}$
Jensen-Shannon	$\log 2 - \log\left(1+e^{-y}\right)$	$-\frac{1}{2-e^{-T_\phi}} = e^{F_\phi(x)}$
KL	$y+1$	$e^{(T_\phi(x)-1)} = e^{F_\phi(x)}$
Reverse KL	$-e^{-y}$	$-\frac{1}{T_\phi(x)} = e^{F_\phi(x)}$
Squared-Hellinger	$1-e^{-v/2}$	$\frac{1}{(1-T_\phi(x))^2} = e^{F_\phi(x)}$

BGAN – Experiments

Train Measure	Eval Measure (lower is better)		
	JS	reverse KL	Wasserstein
BGAN - JS	0.37 ($\pm$0.02)	0.16 ($\pm$0.01)	0.40 ($\pm$0.03)
BGAN - reverse KL	0.44 ($\pm$0.02)	0.44 ($\pm$0.03)	0.45 ($\pm$0.04)
WGAN-GP (samples)	0.45 ($\pm$0.03)	1.32 ($\pm$0.06)	0.87 ($\pm$0.18)
WGAN-GP (softmax)	-	-	0.54 ($\pm$0.12)

BGAN – Experiments

And it 's miant a quert could he
" We pait of condels of money wi
Lankard Avaloma was Mr. Palin ,
Thene says the sounded Sunday in
About dose and warthestrinds fro

He weirst placed produces hopesi
Sance Jory Chorotic , Sen doesin
What was like one of the July 2
The BBC nothing overton and slea
College is out in contesting rev

BGAN – Continuous case

Recall:

$$G^{*} = arg\min_{G} D_{f}(P_{data} \| P_{G})$$

$$D_{f}(P\|Q) = E_{x\sim Q}\left[f\left(\frac{p(x)}{q(x)}\right)\right] = E_{x\sim Q}\left[\sup_{t}\{t\,p(x)/q(x) - f^{*}(t)\}\right]$$

$$\Updownarrow$$

$$p(x)/q(x) = \left(\partial f^{*}/\partial D\right)(D^{*}(x)) = w(x)$$

$$\Updownarrow$$

$$G^{*} = arg\min_{G}(\log w(G(z)))^{2}$$

Max when $\nabla\{t\,p(x)/q(x) - f^{*}(t)\}=0$

$$\Updownarrow$$

$$G^{*} = arg\min_{G} D(G(z))^{2}$$

$p(x)=q(x)$ when $w(x)=1$

BGAN – Continuous case

f – GAN:

$$G^{*} = arg\min_{G} \{E_{x\sim P}[v \circ D(x)] - E_{z\sim h(z)}[f^{*}(v \circ D(G(z)))]\}$$

GAN (Proxy GAN):

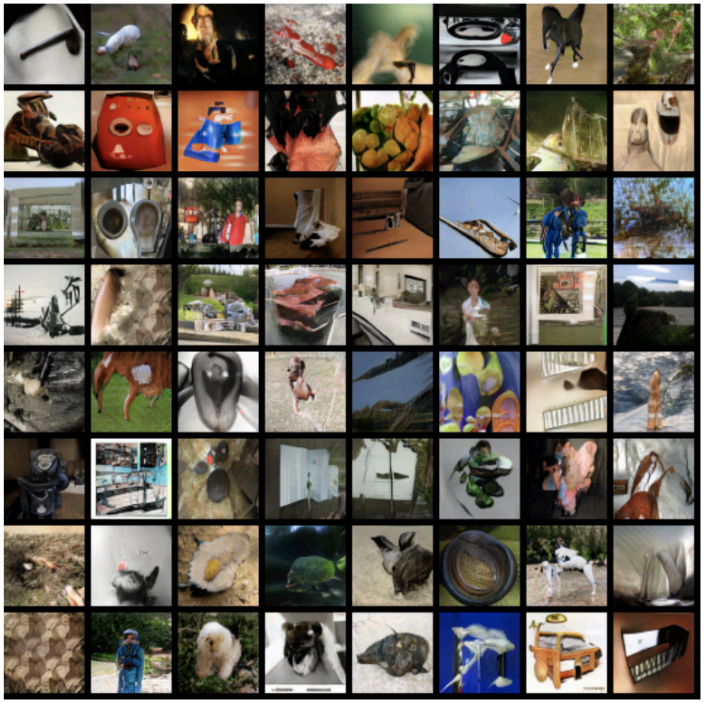
$$G^{*} = arg\min_{G} \{E_{x\sim P}[\log D(x)] + E_{z\sim h(z)}[\log(1 - D(G(z)))]\}$$

BGAN:

$$G^{*} = arg\min_{G} D(G(z))^{2} \iff w(x) = 1 \iff p(x) = q(x)$$

BGAN – Continuous Experiments

CelebA



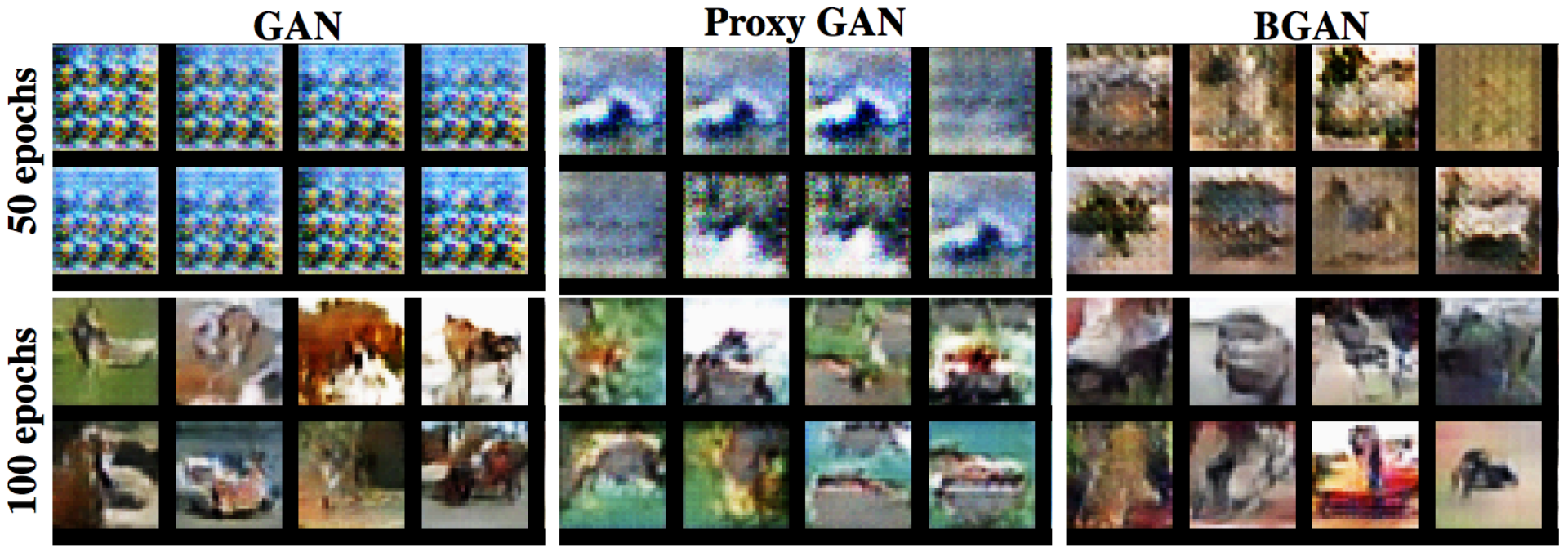
Imagenet

LSUN

Figure 3: Highly realistic samples from a generator trained with BGAN on the CelebA and LSUN datasets. These models were trained using a deep ResNet architecture with gradient norm regularization (Roth et al., 2017). The Imagenet model was trained on the full 1000 label dataset without conditioning.

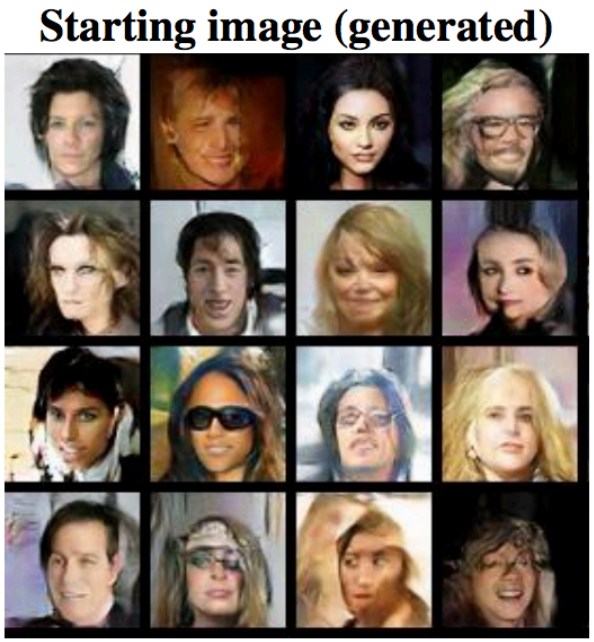
BGAN – Continuous Experiments

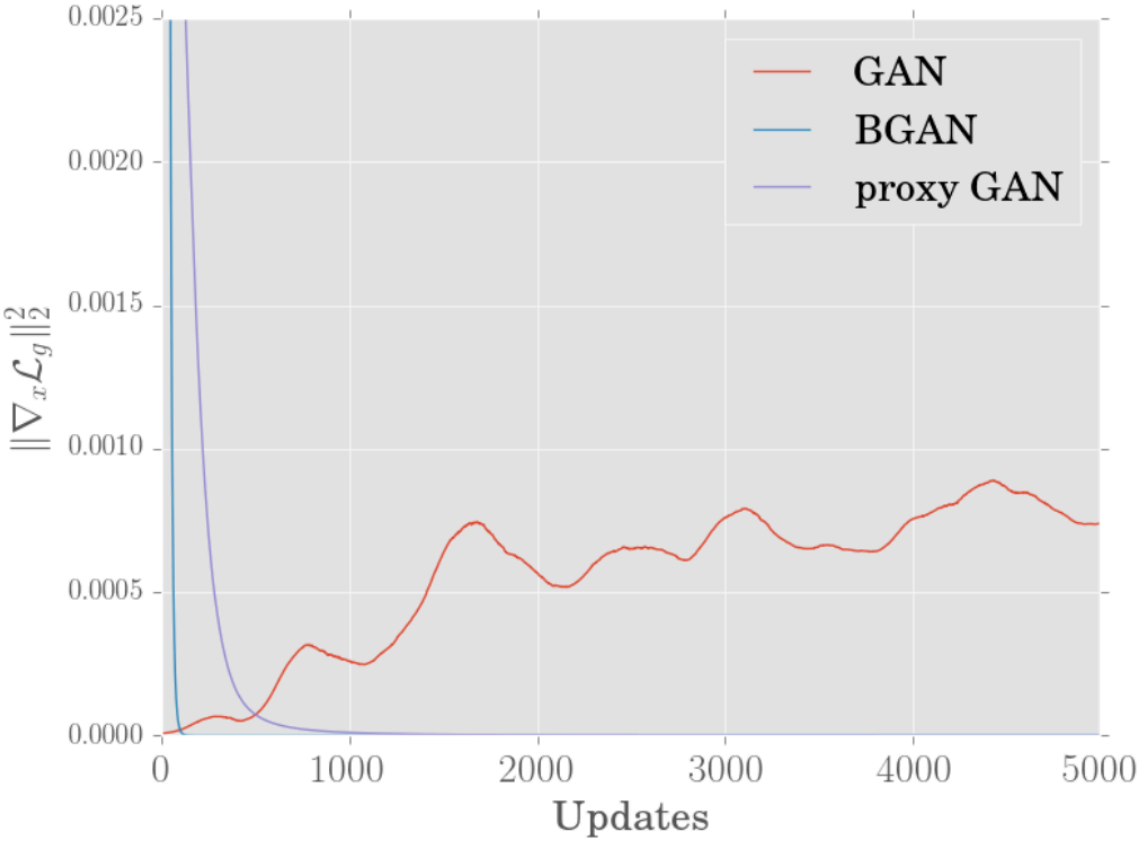
- Generator trained for 5 steps for every 1 step of the discriminator



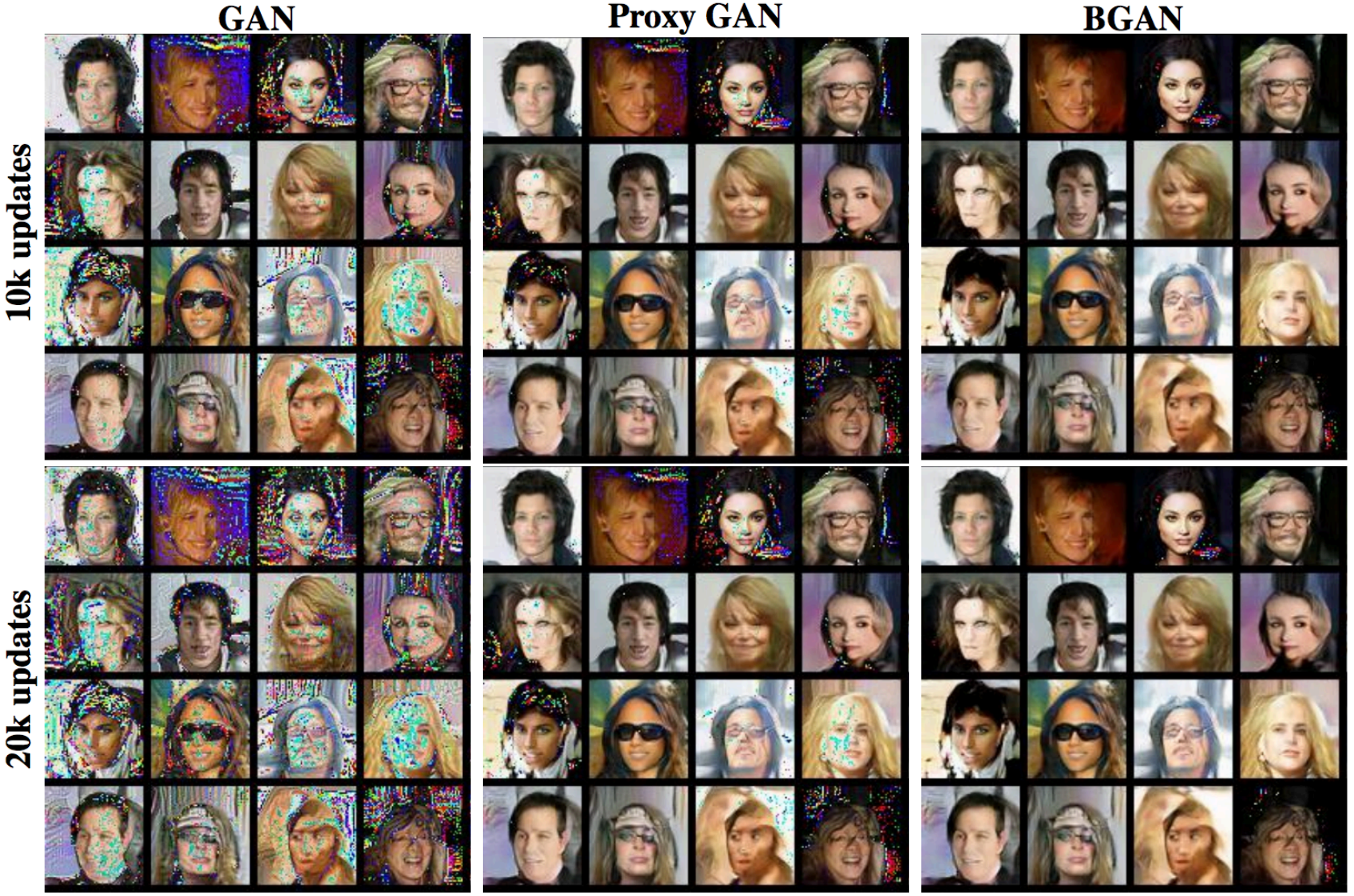
BGAN – Continuous Experiments

- Train a DCGAN using the proxy loss.
- Train the discriminator for 1000 more steps
- Perform gradient descent directly on the pixels





BGAN – Continuous Experiments



Discussion