# Implicit generative models

*Egor Zakharov*

*Skolkovo Institute of Science and Technology*

DeepBayes

# Outline

- Adversarial objective discussion
- Training tricks for GANs
- Applications

# Problem formulation

# Problem formulation

$$x \sim p(x)$$

# Problem formulation

$$x \sim p(x)$$

Complicated multidimensional distribution, ex. images

# Problem formulation

$$x \sim p(x), \quad z \sim p(z) = \mathcal{N}(0, \mathcal{I})$$

Complicated multidimensional
distribution, ex. images

# Problem formulation

$$x \sim p(x), \quad z \sim p(z) = \mathcal{N}(0, \mathcal{I})$$

Complicated multidimensional distribution, ex. images

Simple distribution we can sample from (ex. another dataset)

# Problem formulation

$$x \sim p(x), \quad z \sim p(z) = \mathcal{N}(0, \mathcal{I}), \quad G_\theta(z) \sim q(x)$$

Complicated multidimensional distribution, ex. images

Simple distribution we can sample from (ex. another dataset)

# Problem formulation

$$x \sim p(x), \quad z \sim p(z) = \mathcal{N}(0, \mathcal{I}), \quad G_\theta(z) \sim q(x)$$

Complicated multidimensional distribution, ex. images

Simple distribution we can sample from (ex. another dataset)

$G_\theta$ is a mapping with the same codomain as $p(x)$

# Problem formulation

$$x \sim p(x), \quad z \sim p(z) = \mathcal{N}(0, \mathcal{I}), \quad G_\theta(z) \sim q(x)$$

$$\boxed{\mathcal{D}(P||Q) = \max_f \ \mathbb{E}_{x\sim p(x)}[\log f(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - f(G_\theta(z)))]}$$

# Problem formulation

$$x \sim p(x), \quad z \sim p(z) = \mathcal{N}(0, \mathcal{I}), \quad G_\theta(z) \sim q(x)$$

$$\boxed{\mathcal{D}(P||Q) = \max_f \; \mathbb{E}_{x \sim p(x)}[\log f(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - f(G_\theta(z)))]}$$

Divergence between distributions of "real" data and generated samples

# Problem formulation

$$x \sim p(x), \quad z \sim p(z) = \mathcal{N}(0, \mathcal{I}), \quad G_\theta(z) \sim q(x)$$

$$\boxed{\mathcal{D}(P||Q) = \max_f \; \underbrace{\mathbb{E}_{x\sim p(x)}[\log f(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - f(G_\theta(z)))]}}$$

Divergence between distributions of “real” data and generated samples

$-$Binary cross entropy for classifier
$f$ : probability of data being “real”

# Problem formulation

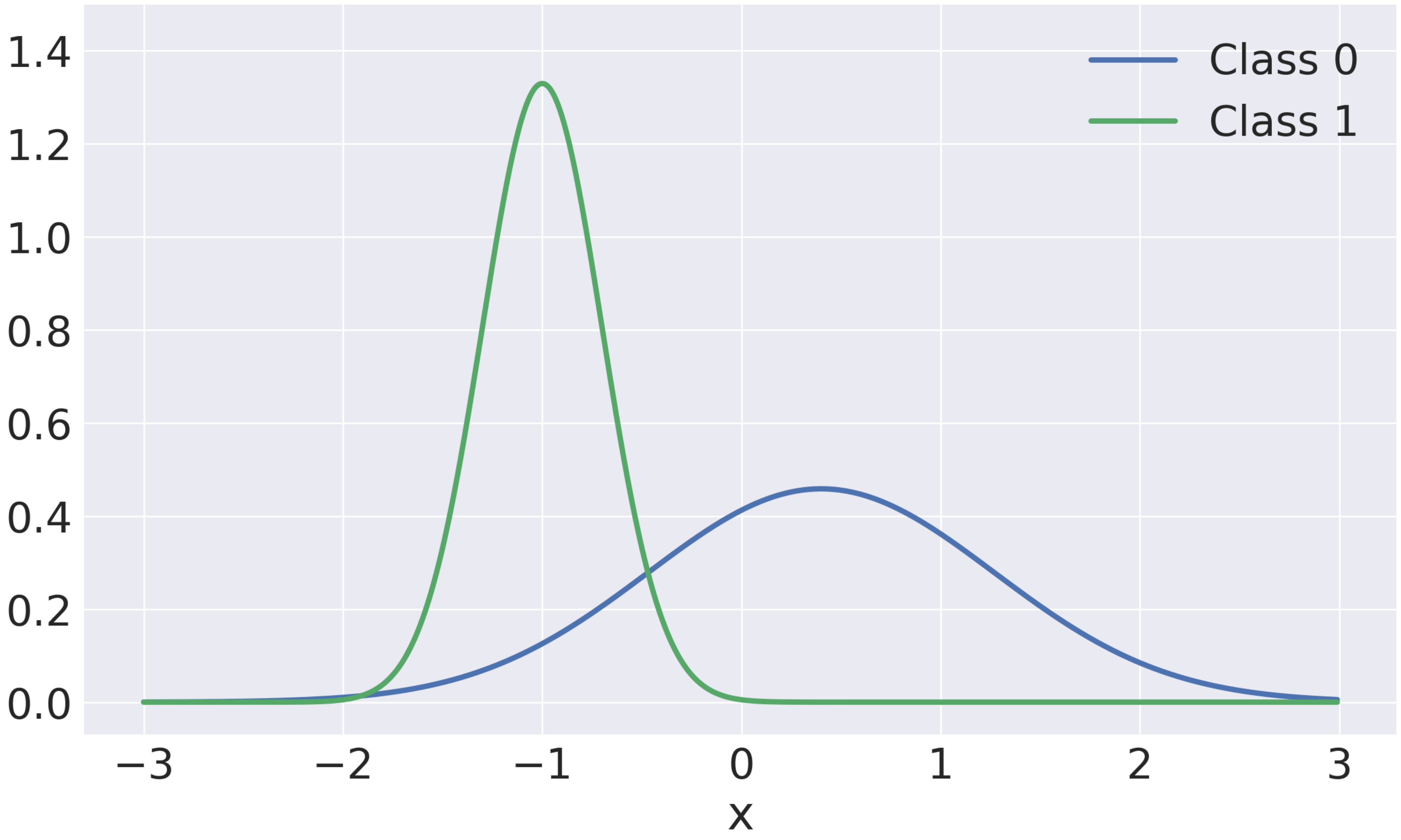
$$x \sim p(x), \quad z \sim p(z) = \mathcal{N}(0, \mathcal{I}), \quad G_\theta(z) \sim q(x)$$

$$\mathcal{D}(P||Q) = \max_f \ \mathbb{E}_{x \sim p(x)}[\log f(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - f(G_\theta(z)))]$$

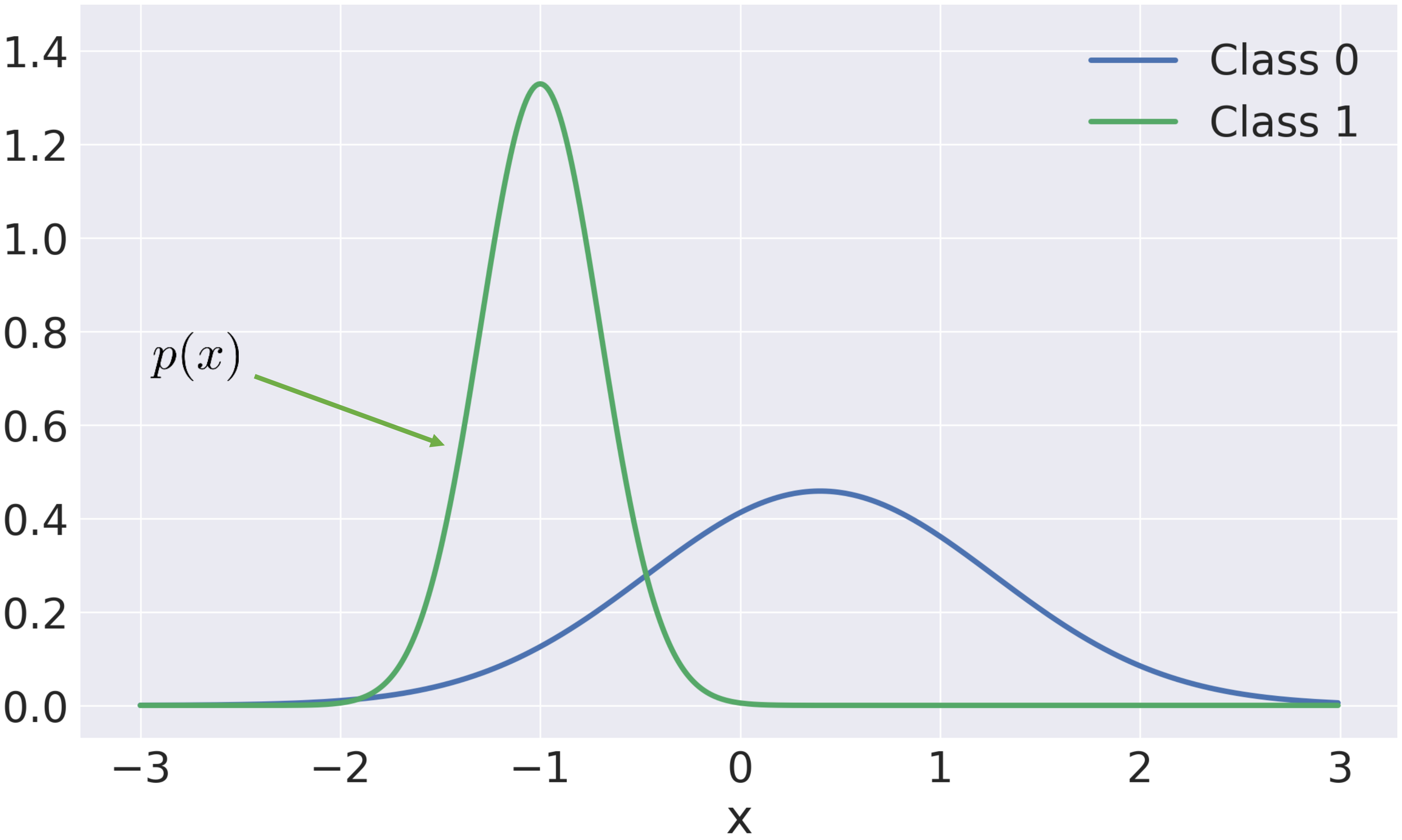
Optimization problem for $G_\theta$ :

$$\boxed{\begin{aligned} &\min_\theta \ \mathcal{D}(P||Q) \\ &\text{s.t. } f^*(x) = \arg\max_f \ \mathcal{D}(P||Q) = \frac{p(x)}{p(x) + q(x)} \end{aligned}}$$

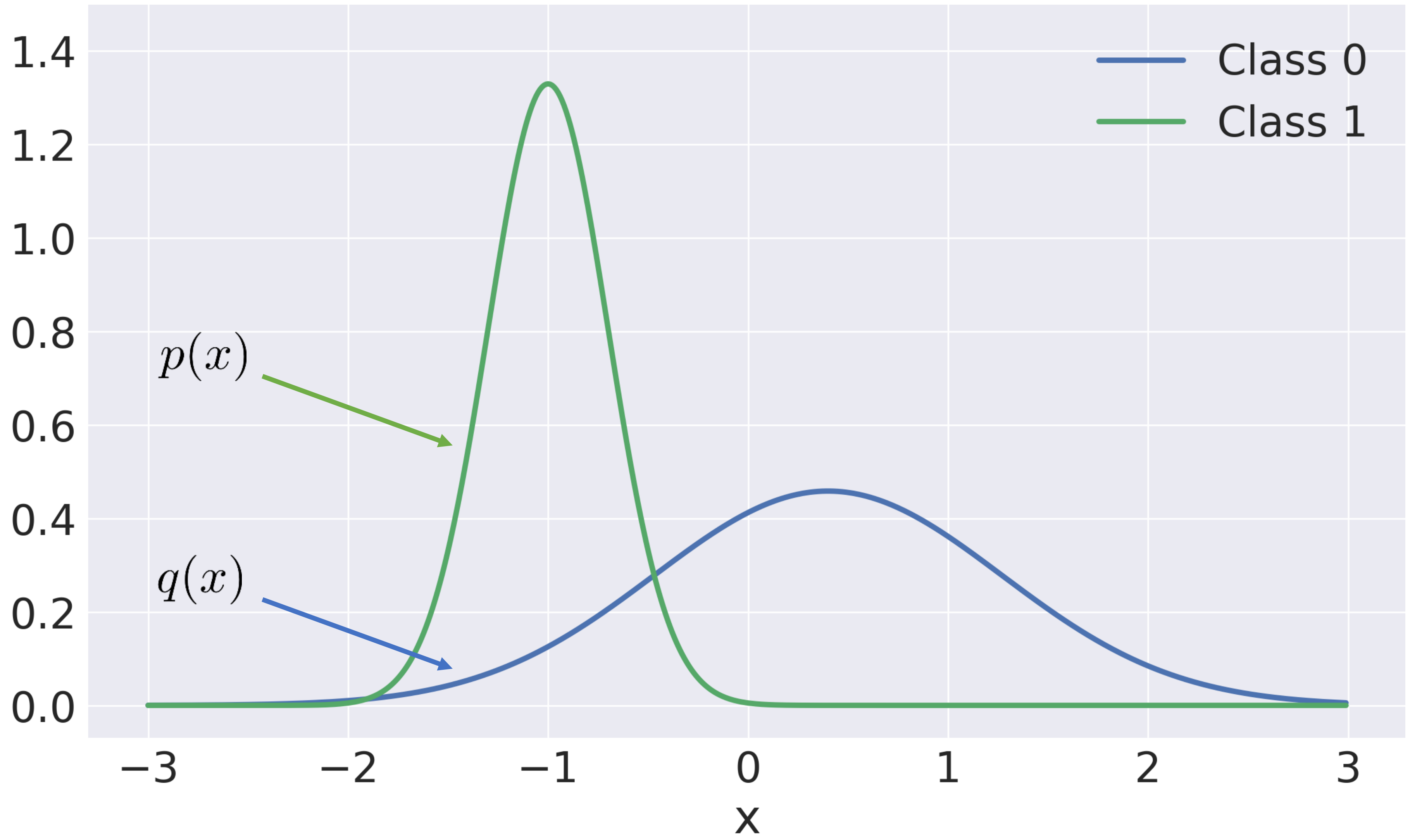
# Example



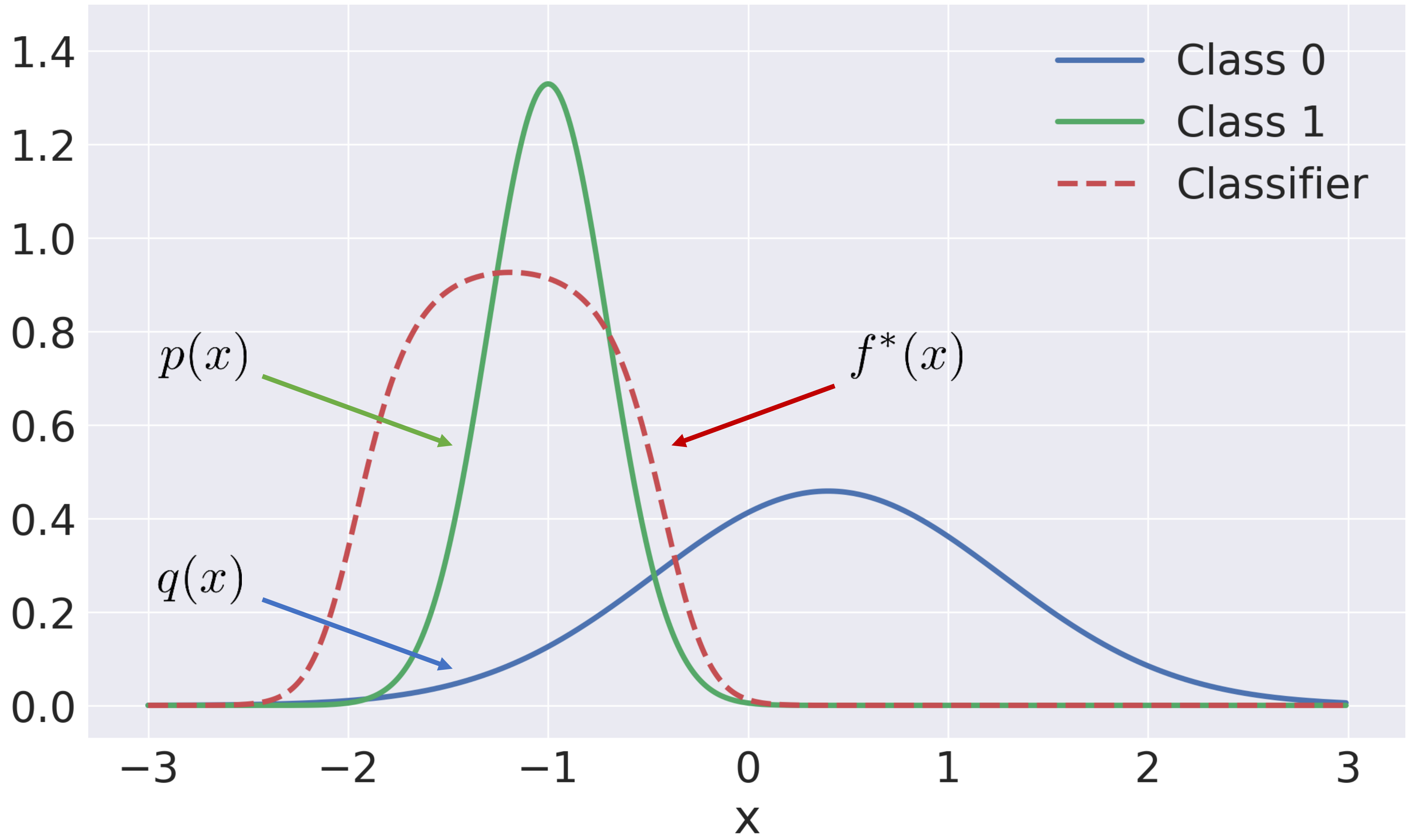
# Example



# Example

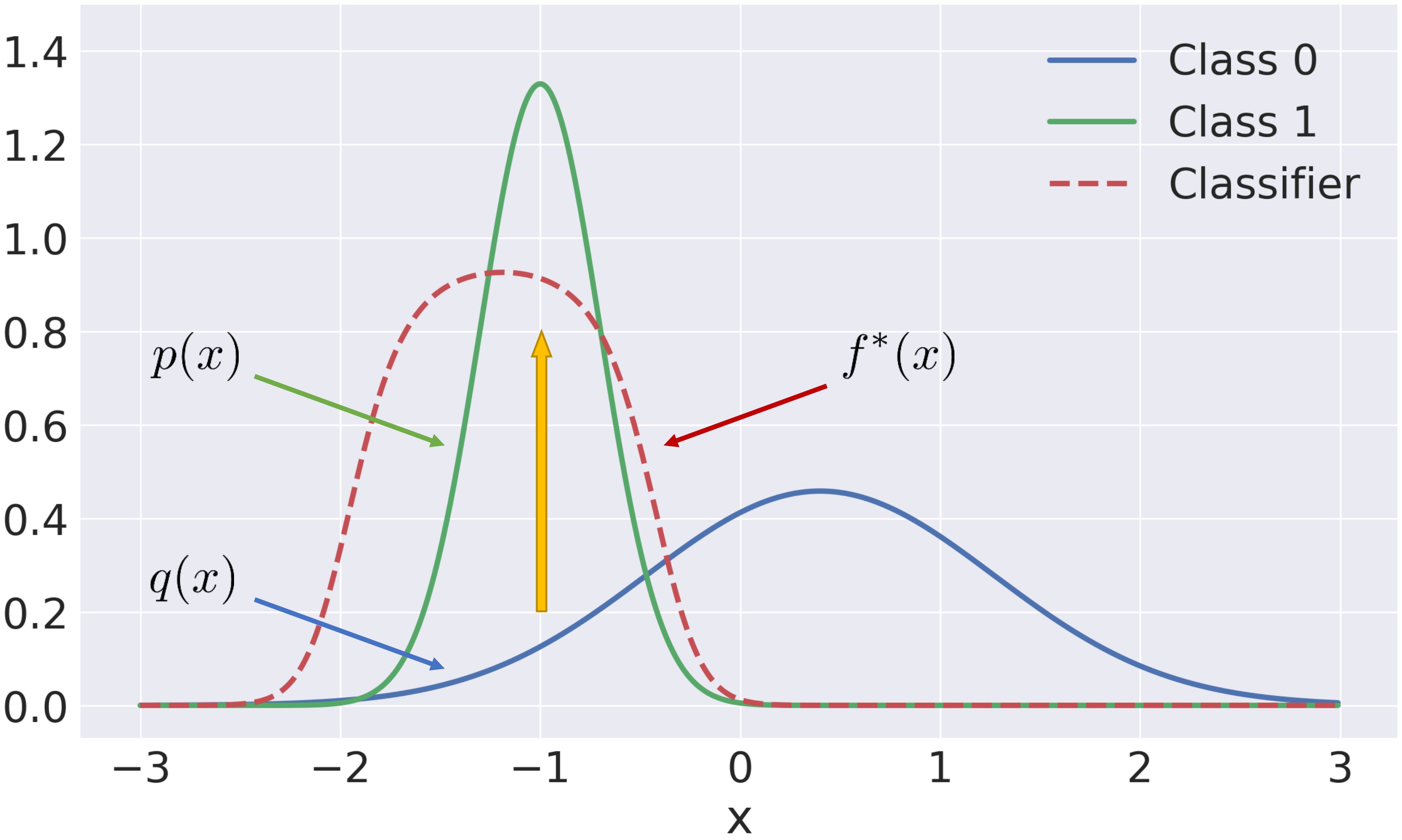


# Example



Class 0
Class 1
Classifier
$p(x)$
$f^*(x)$
$q(x)$
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3
x

# Example



Class 0
Class 1
Classifier
$p(x)$
$q(x)$
$f^*(x)$
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3
x

# Optimization of divergence

$$\boxed{\begin{aligned}&\min_{\theta}\ \mathbb{E}_{x\sim p(x)}[\log f^*(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - f^*(G_\theta(z)))]\\ &\text{s.t. } f^*(x) \ = \ \frac{p(x)}{p(x)+q(x)}\end{aligned}}$$

# Optimization of divergence

$$\boxed{\begin{aligned} &\min_\theta \; \mathbb{E}_{x\sim p(x)}[\log f^*(x)] + \mathbb{E}_{z\sim p(z)}[\log(1-f^*(G_\theta(z)))] \\ &\text{s.t. } f^*(x) = \frac{p(x)}{p(x)+q(x)} \end{aligned}}$$

$\mathcal{D}(P||Q)$

# Optimization of divergence

$$\boxed{\begin{aligned}&\min_{\theta}\ \mathbb{E}_{x\sim p(x)}[\log f^*(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - f^*(G_\theta(z)))] \\ &\text{s.t. } f^*(x) \ = \ \frac{p(x)}{p(x)+q(x)}\end{aligned}}$$

$$\mathcal{D}(P||Q)$$

$$\frac{\partial}{\partial\theta}\mathcal{D}(P||Q)$$

# Optimization of divergence

$$\boxed{\begin{aligned} &\min_{\theta}\ \mathbb{E}_{x\sim p(x)}[\log f^*(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - f^*(G_\theta(z)))] \\ &\text{s.t. } f^*(x) \ = \ \frac{p(x)}{p(x)+q(x)} \end{aligned}}$$

$$\mathcal{D}(P||Q)$$

$$\frac{\partial}{\partial\theta}\mathcal{D}(P||Q) \ = \ \mathbb{E}_{z\sim p(z)}\frac{\partial}{\partial\theta}[\log(1 - f^*(G_\theta(z)))]$$

# Optimization of divergence

$$\min_{\theta}\ \mathbb{E}_{x\sim p(x)}[\log f^*(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - f^*(G_\theta(z)))]$$

$$\text{s.t. } f^*(x) \;=\; \frac{p(x)}{p(x)+q(x)}$$

$$\mathcal{D}(P||Q)$$

$$\frac{\partial}{\partial\theta}\mathcal{D}(P||Q) \;=\; \mathbb{E}_{z\sim p(z)}\frac{\partial}{\partial\theta}[\log(1 - f^*(G_\theta(z)))]$$

$$\frac{\partial}{\partial\theta}\mathcal{D}(P||Q) \;\propto\; \left\{\text{chain rule, } \; x = G_\theta(z)\right\} \;\propto\; \mathbb{E}_{z\sim p(z)}\left(\frac{1}{1-f^*}\frac{\partial f^*}{\partial x}\frac{\partial x}{\partial \theta}\right)\Bigg|_z$$

# Optimization of divergence

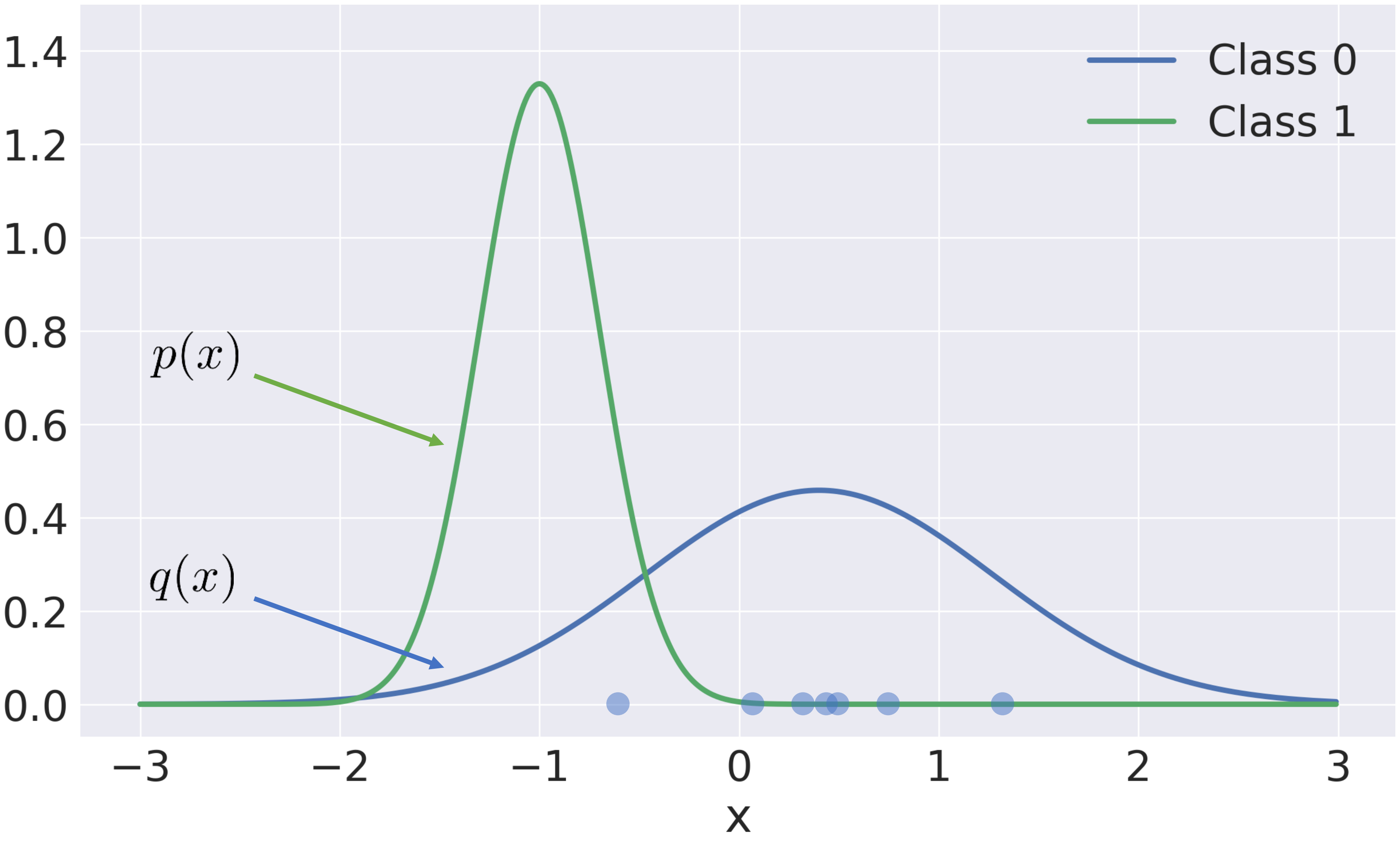
$$\min_{\theta} \; \underbrace{\mathbb{E}_{x\sim p(x)}[\log f^*(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - f^*(G_\theta(z)))]}_{\mathcal{D}(P||Q)}$$

$$\text{s.t. } f^*(x) \;=\; \frac{p(x)}{p(x)+q(x)}$$

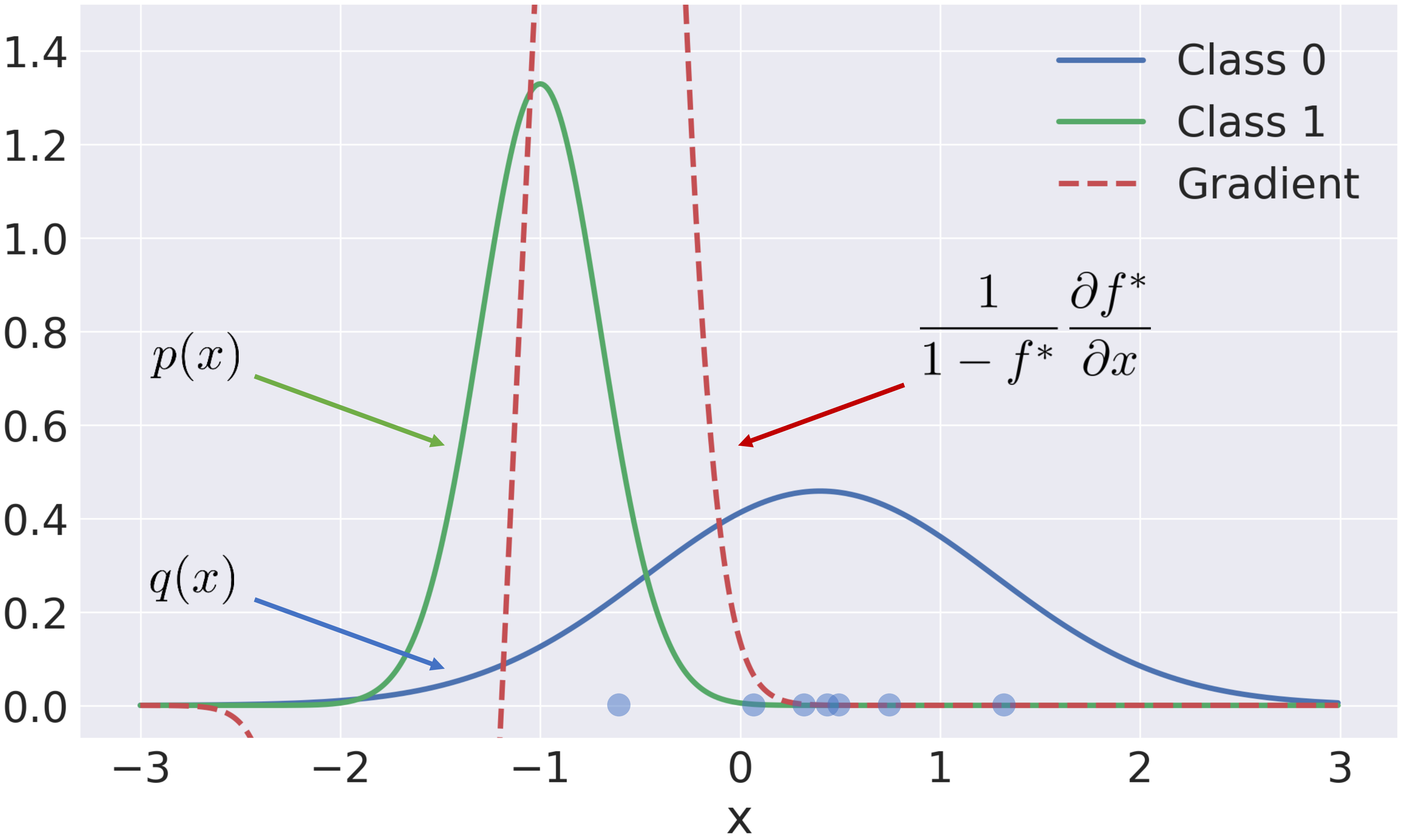
$$\frac{\partial}{\partial\theta}\mathcal{D}(P||Q) \;=\; \mathbb{E}_{z\sim p(z)}\frac{\partial}{\partial\theta}[\log(1 - f^*(G_\theta(z)))]$$

$$\frac{\partial}{\partial\theta}\mathcal{D}(P||Q) \;\propto\; \left\{\text{chain rule, } \; x = G_\theta(z)\right\} \;\propto\; \mathbb{E}_{z\sim p(z)}\left(\boxed{\frac{1}{1-f^*}\frac{\partial f^*}{\partial x}}\frac{\partial x}{\partial\theta}\right)\Bigg|_z$$

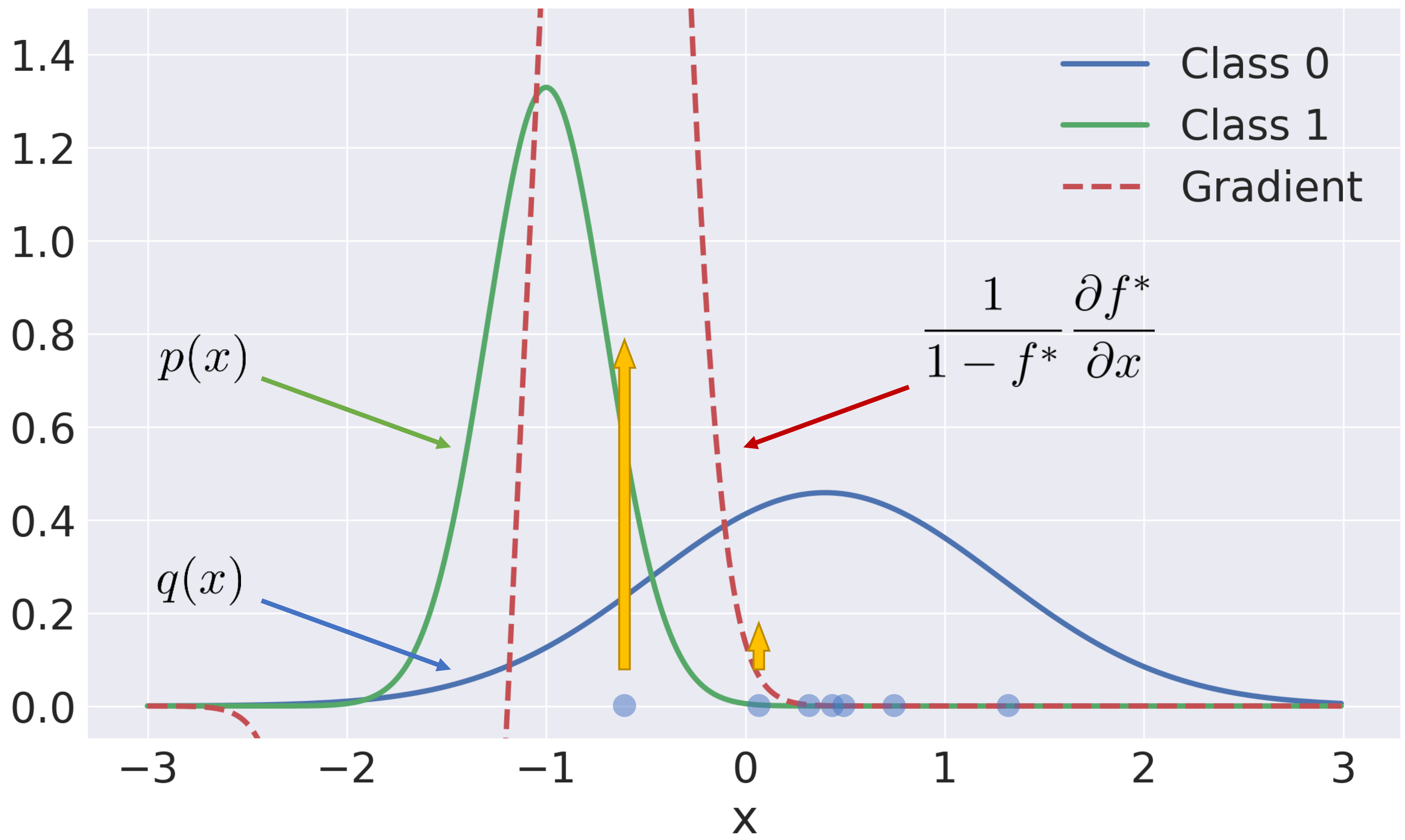
# Optimization of divergence



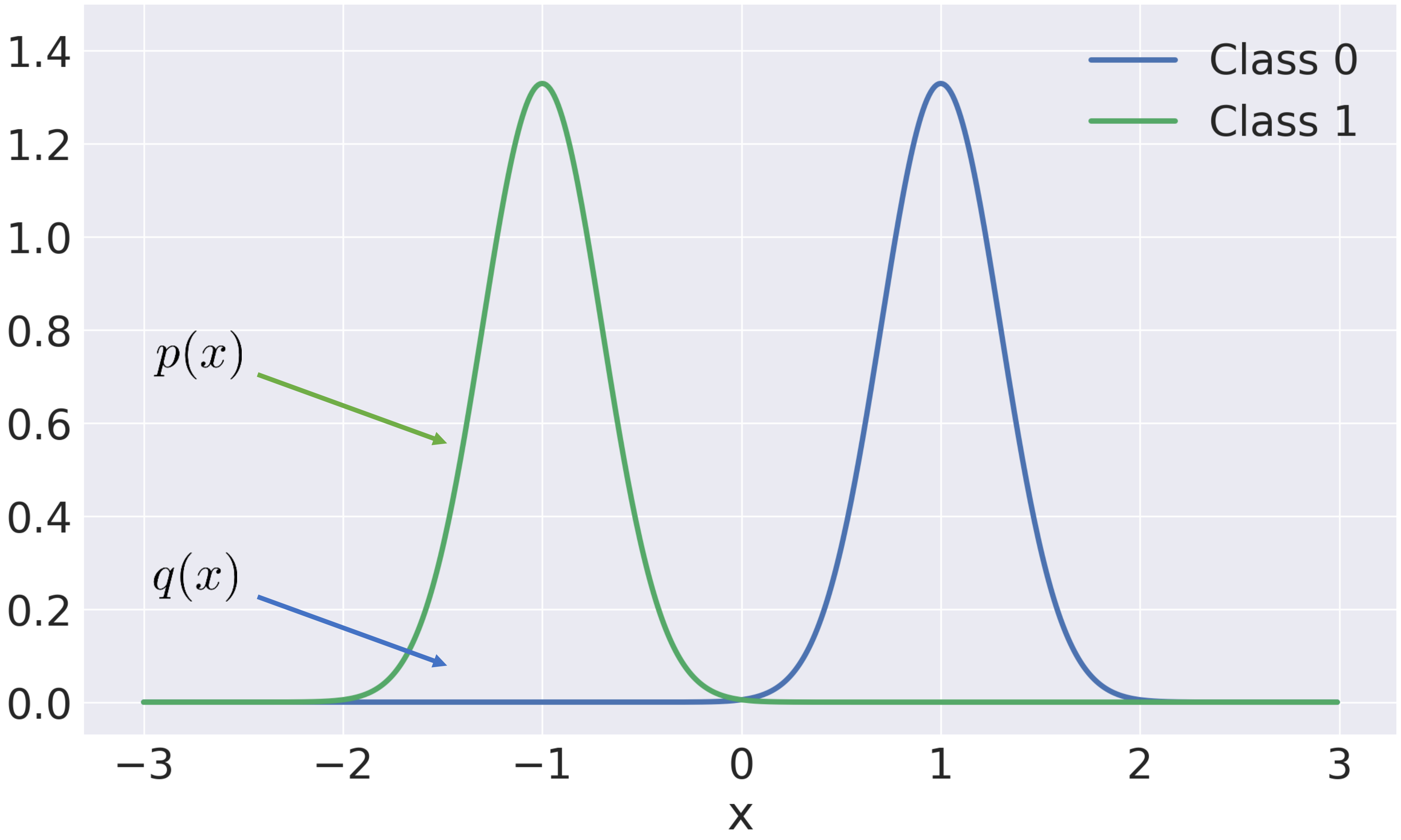
# Optimization of divergence



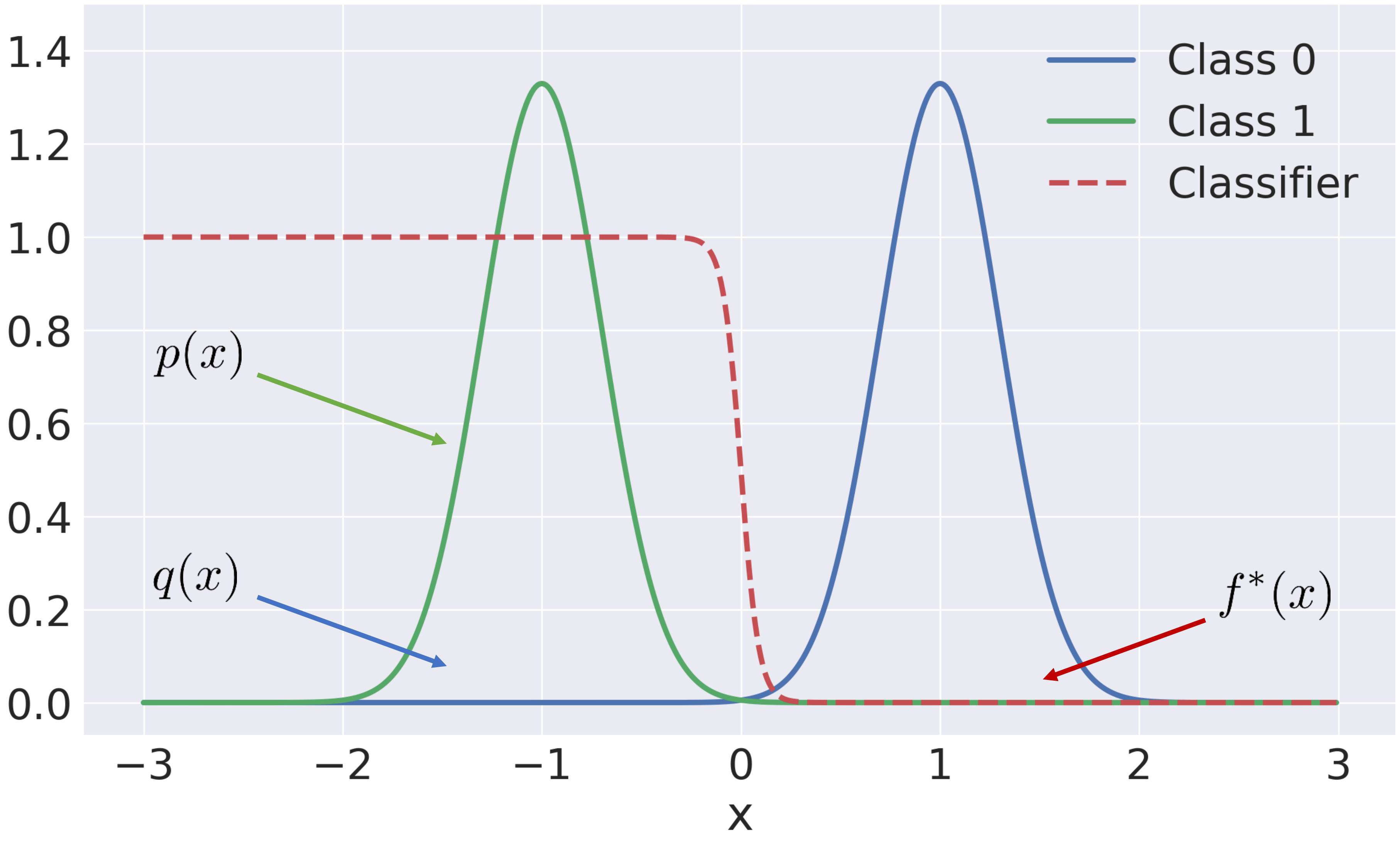
# Optimization of divergence



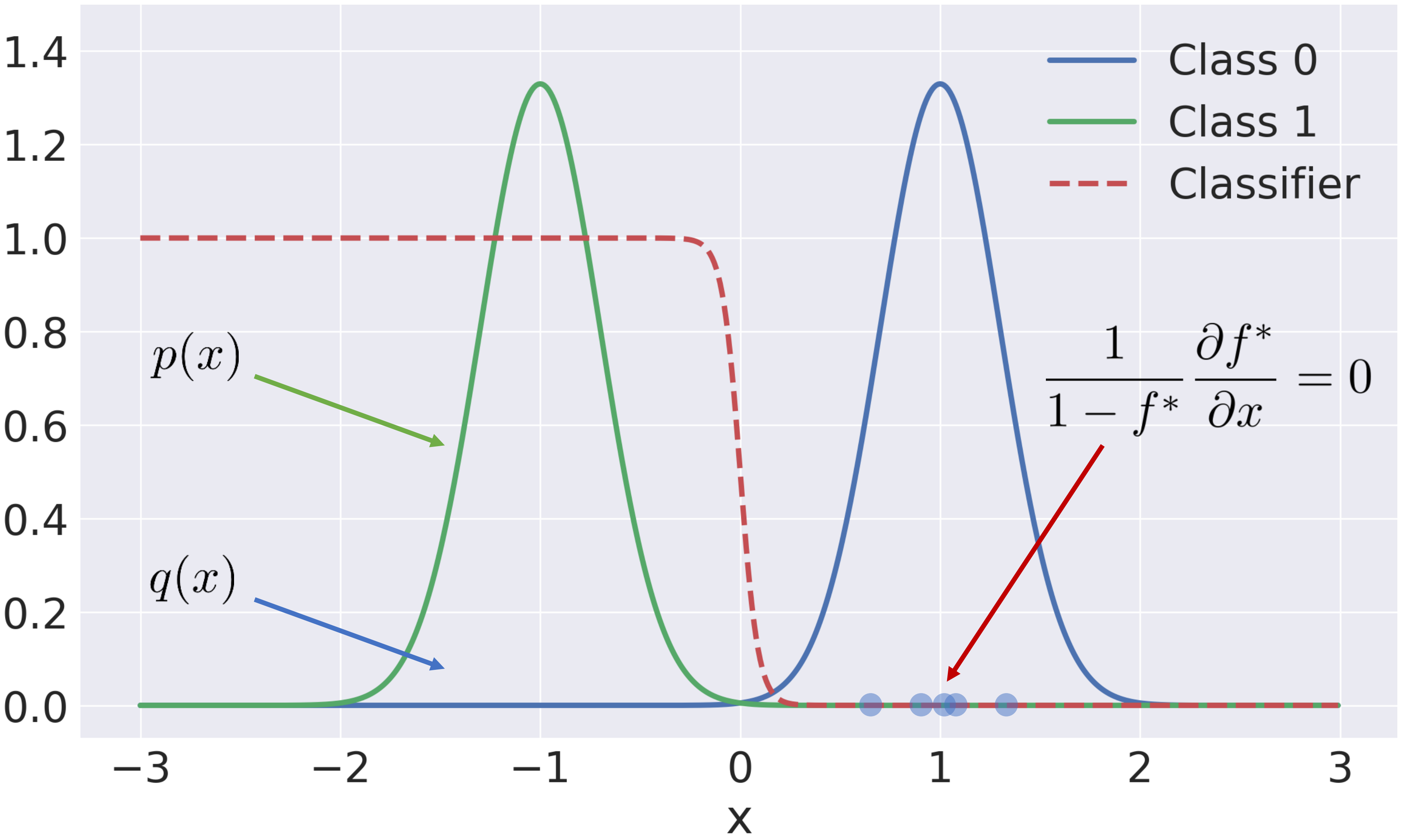
# Failure case



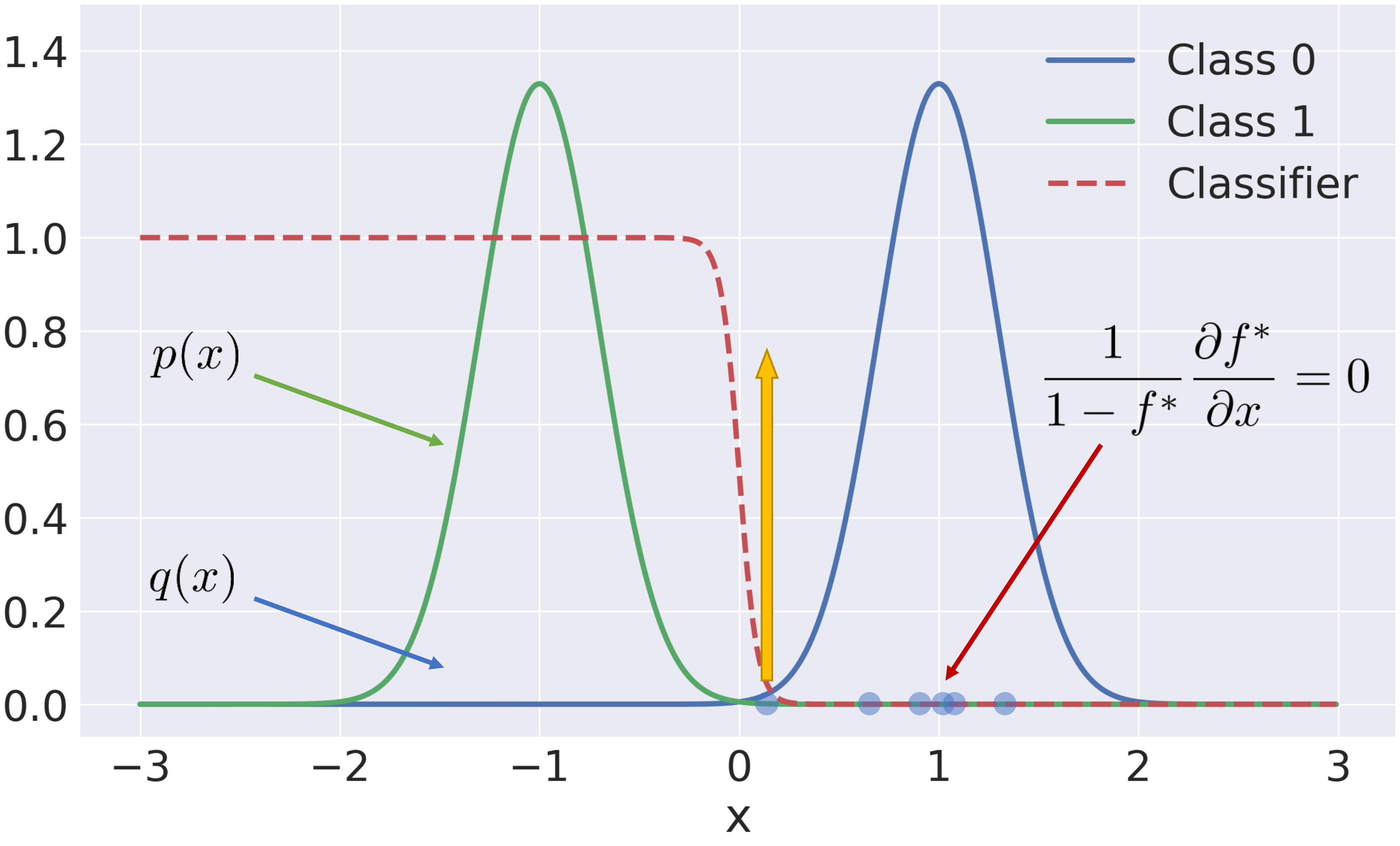
## Failure case



# Failure case



# Failure case



# Example

Let's consider a toy example

$$x \sim p(x), \quad z \sim p(z) = \mathcal{N}(0, \mathcal{I})$$

# Example

Let's consider a toy example

$$x \sim p(x), \quad z \sim p(z) = \mathcal{N}(0, \mathcal{I})$$

Mixture of gaussians

# Example

Let's consider a toy example

$$x \sim p(x), \quad z \sim p(z) = \mathcal{N}(0, \mathcal{I})$$

Mixture of gaussians

$$f = D_\phi$$

$$\phi^* = \arg\max_\phi \; \mathbb{E}_{x \sim p(x)}[\log D_\phi(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]$$

# Example

Let's consider a toy example

$$x \sim p(x), \quad z \sim p(z) = \mathcal{N}(0, \mathcal{I})$$

Mixture of gaussians

$$f = D_\phi$$

$$\phi^* = \arg\max_\phi \; \mathbb{E}_{x\sim p(x)}[\log D_\phi(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]$$

We will use gradient ascend until convergence

# Example

# Alternating gradients

$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim p(x)}[\log D_{\phi}(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D_{\phi}(G_{\theta}(z)))]$$

# Alternating gradients

$$\min_{\theta} \max_{\phi} \underbrace{\mathbb{E}_{x \sim p(x)}[\log D_{\phi}(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D_{\phi}(G_{\theta}(z)))]}_{\mathcal{L}(\theta, \phi)}$$

# Alternating gradients

$$\min_{\theta} \max_{\phi} \underbrace{\mathbb{E}_{x\sim p(x)}[\log D_\phi(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]}_{\mathcal{L}(\theta,\phi)}$$

1. $\phi^* \;=\; \arg\max_\phi \mathcal{L}(\theta^{\text{old}}, \phi)$
2. $\theta^{\text{new}} =\; \theta^{\text{old}} - \alpha \nabla_\theta \mathcal{L}(\theta^{\text{old}}, \phi^*)$
3. $\theta^{\text{old}} \;=\; \theta^{\text{new}}$
4. Go to step 1

# Alternating gradients

$$\min_{\theta} \max_{\phi} \underbrace{\mathbb{E}_{x \sim p(x)}[\log D_\phi(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]}_{\mathcal{L}(\theta, \phi)}$$

1. $\phi^* \ = \ \arg\max_\phi \mathcal{L}(\theta^{\text{old}}, \phi)$
2. $\theta^{\text{new}} = \ \theta^{\text{old}} - \alpha \nabla_\theta \mathcal{L}(\theta^{\text{old}}, \phi^*)$
3. $\theta^{\text{old}} \ = \ \theta^{\text{new}}$
4. Go to step 1

$\mathcal{D}(P||Q)$

Minimization of divergence at step 2

# Alternating gradients

$$\min_{\theta} \max_{\phi} \underbrace{\mathbb{E}_{x \sim p(x)}[\log D_\phi(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]}_{\mathcal{L}(\theta, \phi)}$$

1. $\phi^* \ = \ \arg\max_\phi \mathcal{L}(\theta^{\text{old}}, \phi)$
2. $\theta^{\text{new}} = \ \theta^{\text{old}} - \alpha \nabla_\theta \mathcal{L}(\theta^{\text{old}}, \phi^*)$
3. $\theta^{\text{old}} \ = \ \theta^{\text{new}}$
4. Go to step 1

$\mathcal{D}(P||Q)$

Minimization of divergence at step 2

1. $\phi^{\text{new}} \ = \ \phi^{\text{old}} + \alpha \nabla_\phi \mathcal{L}(\theta^{\text{old}}, \phi^{\text{old}})$
2. $\theta^{\text{new}} \ = \ \theta^{\text{old}} - \alpha \nabla_\theta \mathcal{L}(\theta^{\text{old}}, \phi^{\text{new}})$
3. $\theta^{\text{old}} = \theta^{\text{new}}, \quad \phi^{\text{old}} = \phi^{\text{new}}$
4. Go to step 1

# Alternating gradients

$$\min_\theta \max_\phi \underbrace{\mathbb{E}_{x\sim p(x)}[\log D_\phi(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]}_{\mathcal{L}(\theta,\phi)}$$

1. $\phi^* = \arg\max_\phi \mathcal{L}(\theta^{\text{old}}, \phi)$
2. $\theta^{\text{new}} = \theta^{\text{old}} - \alpha \nabla_\theta \mathcal{L}(\theta^{\text{old}}, \phi^*)$
3. $\theta^{\text{old}} = \theta^{\text{new}}$
4. Go to step 1

$\mathcal{D}(P||Q)$

Minimization of divergence at step 2

1. $\phi^{\text{new}} = \phi^{\text{old}} + \alpha \nabla_\phi \mathcal{L}(\theta^{\text{old}}, \phi^{\text{old}})$
2. $\theta^{\text{new}} = \theta^{\text{old}} - \alpha \nabla_\theta \mathcal{L}(\theta^{\text{old}}, \phi^{\text{new}})$
3. $\theta^{\text{old}} = \theta^{\text{new}}, \quad \phi^{\text{old}} = \phi^{\text{new}}$
4. Go to step 1

Minimization of lower bound for divergence

# Example

# Summary

- Minimization of divergence does not work well with neural networks because of extremely slow convergence speed

# Summary

- Minimization of divergence does not work well with neural networks because of extremely slow convergence speed

- GANs are inspired by the optimization of divergence, but we minimize a very loose lower bound to it

# Summary

- Minimization of divergence does not work well with neural networks because of extremely slow convergence speed

- GANs are inspired by the optimization of divergence, but we minimize a very loose lower bound to it

- They are optimized using alternating gradient descend, convergence to saddle point is not guaranteed

# Use prior knowledge

1. Explicitly match the support of your real distribution

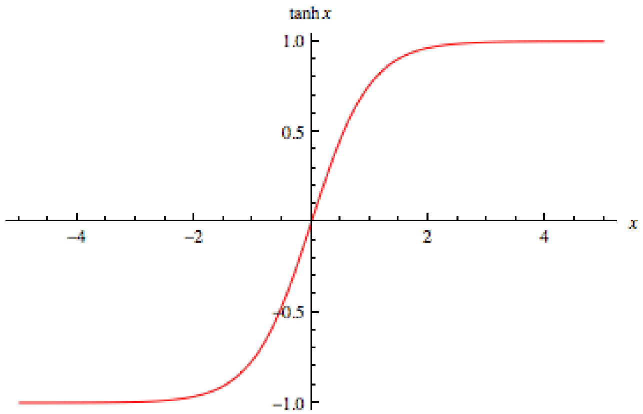
   Put appropriate nonlinearity at the end of the generator

# Use prior knowledge

1. Explicitly match the support of your real distribution

   Put appropriate nonlinearity at the end of the generator

   Ex. Images: hyperbolic tangent

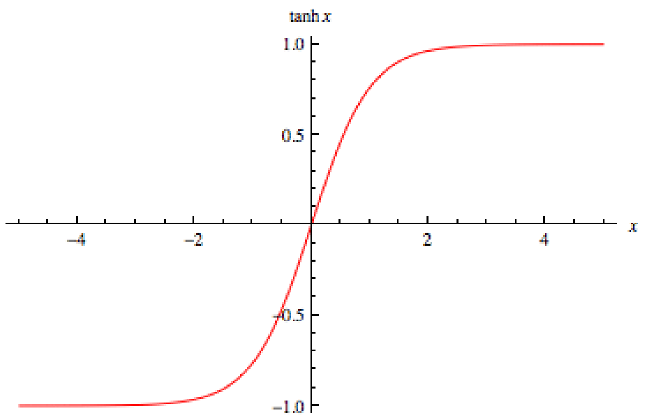


# Use prior knowledge

1. Explicitly match the support of your real distribution

   Put appropriate nonlinearity at the end of the generator

   Ex. Images: hyperbolic tangent




2. Avoid sparse gradients in discriminator

   Replace ReLUs with LeakyReLUs and MaxPool with AvgPool

   Has to do with intrinsically smooth domains (1d signals, images)

# Use prior knowledge

https://distill.pub/2016/deconv-checkerboard/

# Question

$$\mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{x \sim p(x)}[\log D_\phi(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]$$

# Question

$$\mathcal{L}(\theta, \phi) \;\; = \;\; \mathbb{E}_{x \sim p(x)}[\log D_\phi(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]$$

$$\nabla_\phi \mathcal{L}(\theta, \phi) \;\; = \;\; \mathbb{E}_{x \sim p(x)}\left[\frac{1}{D_\phi(x)} \frac{\partial D_\phi(x)}{\partial \phi}\right] + \mathbb{E}_{z \sim p(z)}\left[\frac{1}{1 - D_\phi(G_\theta(z))} \frac{\partial D_\phi(G_\theta(z))}{\partial \phi}\right]$$

# Question

$$\mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{x\sim p(x)}[\log D_\phi(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]$$

$$\nabla_\phi \mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{x\sim p(x)}\left[\frac{1}{D_\phi\underbrace{(x)}_{\hat{x}}} \frac{\partial D_\phi\overbrace{(x)}^{\hat{x}}}{\partial \phi}\right] + \mathbb{E}_{z\sim p(z)}\left[\frac{1}{1 - D_\phi\underbrace{(G_\theta(z))}_{\hat{x}}} \frac{\partial D_\phi\overbrace{(G_\theta(z))}^{\hat{x}}}{\partial \phi}\right]$$

# Question

$$\mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{x\sim p(x)}[\log D_\phi(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]$$

$$\nabla_\phi \mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{x\sim p(x)}\left[\frac{1}{D_\phi(x)} \frac{\partial D_\phi(x)}{\partial \phi}\right] + \mathbb{E}_{z\sim p(z)}\left[\frac{1}{1 - D_\phi(G_\theta(z))} \frac{\partial D_\phi(G_\theta(z))}{\partial \phi}\right]$$

$$p(\hat{x}, y) = p(\hat{x}|y)p(y) = p(x)p(y=1) + q(x)p(y=0)$$

# Question

$$\mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{x\sim p(x)}[\log D_\phi(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]$$

$$\nabla_\phi \mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{x\sim p(x)}\left[\frac{1}{D_\phi(x)} \frac{\partial D_\phi(x)}{\partial \phi}\right] + \mathbb{E}_{z\sim p(z)}\left[\frac{1}{1 - D_\phi(G_\theta(z))} \frac{\partial D_\phi(G_\theta(z))}{\partial \phi}\right]$$

$$p(\hat{x}, y) = p(\hat{x}|y)p(y) = p(x)p(y=1) + q(x)p(y=0)$$

$$\boxed{\begin{aligned} \nabla_\phi \mathcal{L}(\theta, \phi) \;&=\; \mathbb{E}_{\tilde{x}, y \sim p(\tilde{x}, y)}\Big[\mathbb{I}[y=1] \;\dots\; + \mathbb{I}[y=0] \;\dots\; \Big] \\ \nabla_\theta \mathcal{L}(\theta, \phi) \;&=\; \mathbb{E}_{z\sim p(z)}\left[\frac{1}{1 - D_\phi(G_\theta(z)))} \frac{\partial D_\phi(G_\theta(z))}{\partial G_\theta(z)} \frac{\partial G_\theta(z)}{\partial \theta}\right] \end{aligned}}$$

## Question

$$\mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{x \sim p(x)}[\log D_\phi(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]$$

$$\nabla_\phi \mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{x \sim p(x)}\left[\frac{1}{D_\phi(x)} \frac{\partial D_\phi(x)}{\partial \phi}\right] + \mathbb{E}_{z \sim p(z)}\left[\frac{1}{1 - D_\phi(G_\theta(z))} \frac{\partial D_\phi(G_\theta(z))}{\partial \phi}\right]$$

$$p(\hat{x}, y) = p(\hat{x}|y)p(y) = p(x)p(y=1) + q(x)p(y=0)$$

$$\boxed{\begin{aligned}\nabla_\phi \mathcal{L}(\theta, \phi) \;&=\; \mathbb{E}_{\tilde{x}, y \sim p(\tilde{x}, y)}\left[\mathbb{I}[y=1] \;\ldots\; + \mathbb{I}[y=0] \;\ldots\;\right] \\ \nabla_\theta \mathcal{L}(\theta, \phi) \;&=\; \mathbb{E}_{z \sim p(z)}\left[\frac{1}{1 - D_\phi(G_\theta(z)))} \frac{\partial D_\phi(G_\theta(z))}{\partial G_\theta(z)} \frac{\partial G_\theta(z)}{\partial \theta}\right]\end{aligned}}$$

Is this procedure correct for any D?

# Question

$$\mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{x\sim p(x)}[\log D_\phi(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]$$

$$\nabla_\phi \mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{x\sim p(x)}\left[\frac{1}{D_\phi(x)} \frac{\partial D_\phi(x)}{\partial \phi}\right] + \mathbb{E}_{z\sim p(z)}\left[\frac{1}{1 - D_\phi(G_\theta(z))} \frac{\partial D_\phi(G_\theta(z))}{\partial \phi}\right]$$

$$p(\hat{x}, y) = p(\hat{x}|y)p(y) = p(x)p(y=1) + q(x)p(y=0)$$

$$\nabla_\phi \mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{\tilde{x}, y \sim p(\tilde{x}, y)}\Big[\mathbb{I}[y=1] \;\ldots\; + \mathbb{I}[y=0] \;\ldots\;\Big]$$

$$\nabla_\theta \mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{z\sim p(z)}\left[\frac{1}{1 - D_\phi(G_\theta(z)))} \frac{\partial D_\phi(G_\theta(z))}{\partial G_\theta(z)} \frac{\partial G_\theta(z)}{\partial \theta}\right]$$

Is this procedure correct for any D?

Answer: no

Example?

# Question

$$\mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{x \sim p(x)}[\log D_\phi(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]$$

$$\nabla_\phi \mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{x \sim p(x)}\left[\frac{1}{D_\phi(x)} \frac{\partial D_\phi(x)}{\partial \phi}\right] + \mathbb{E}_{z \sim p(z)}\left[\frac{1}{1 - D_\phi(G_\theta(z))} \frac{\partial D_\phi(G_\theta(z))}{\partial \phi}\right]$$

$$p(\hat{x}, y) = p(\hat{x}|y)p(y) = p(x)p(y=1) + q(x)p(y=0)$$

$$\nabla_\phi \mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{\tilde{x}, y \sim p(\tilde{x}, y)}\Big[\mathbb{I}[y=1] \;\ldots\; + \mathbb{I}[y=0] \;\ldots\;\Big]$$

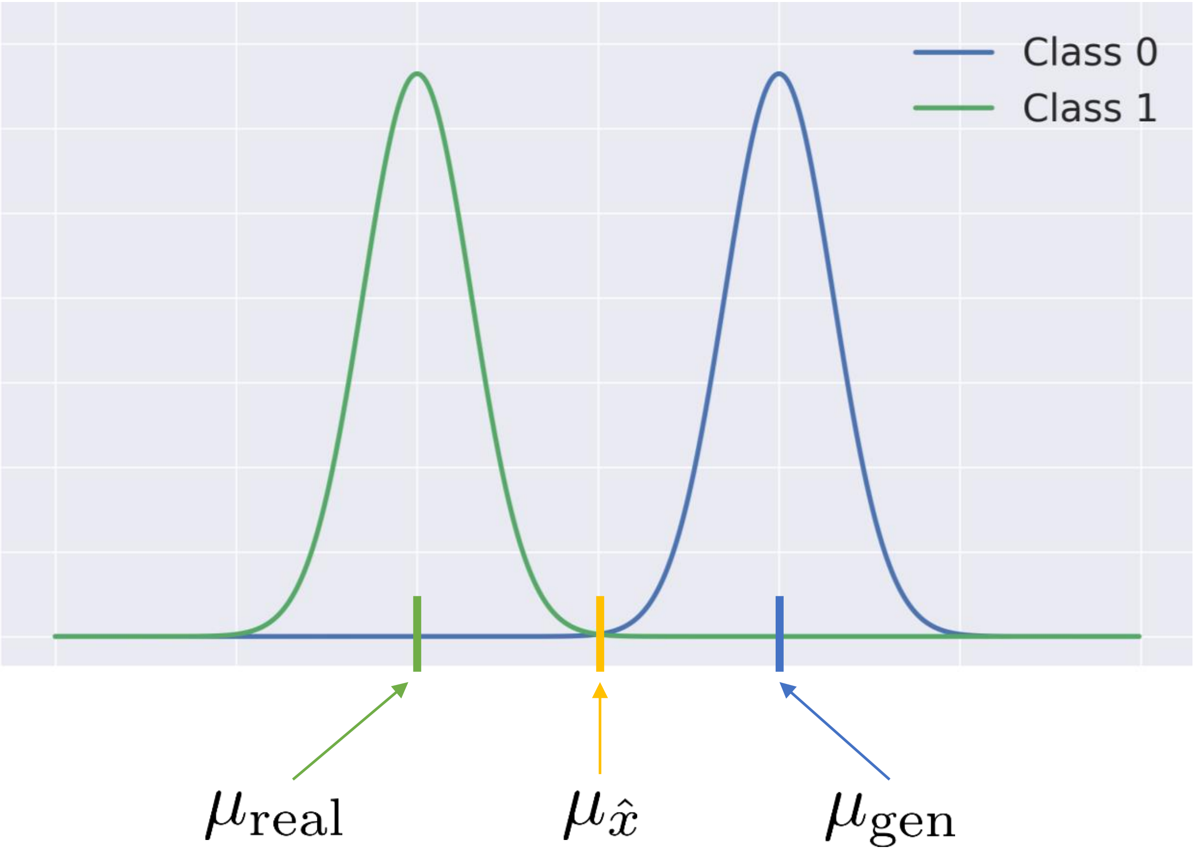
$$\nabla_\theta \mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{z \sim p(z)}\left[\frac{1}{1 - D_\phi(G_\theta(z)))} \frac{\partial D_\phi(G_\theta(z))}{\partial G_\theta(z)} \frac{\partial G_\theta(z)}{\partial \theta}\right]$$

Is this procedure correct for any D?

Answer: no
Ex.: BatchNorm

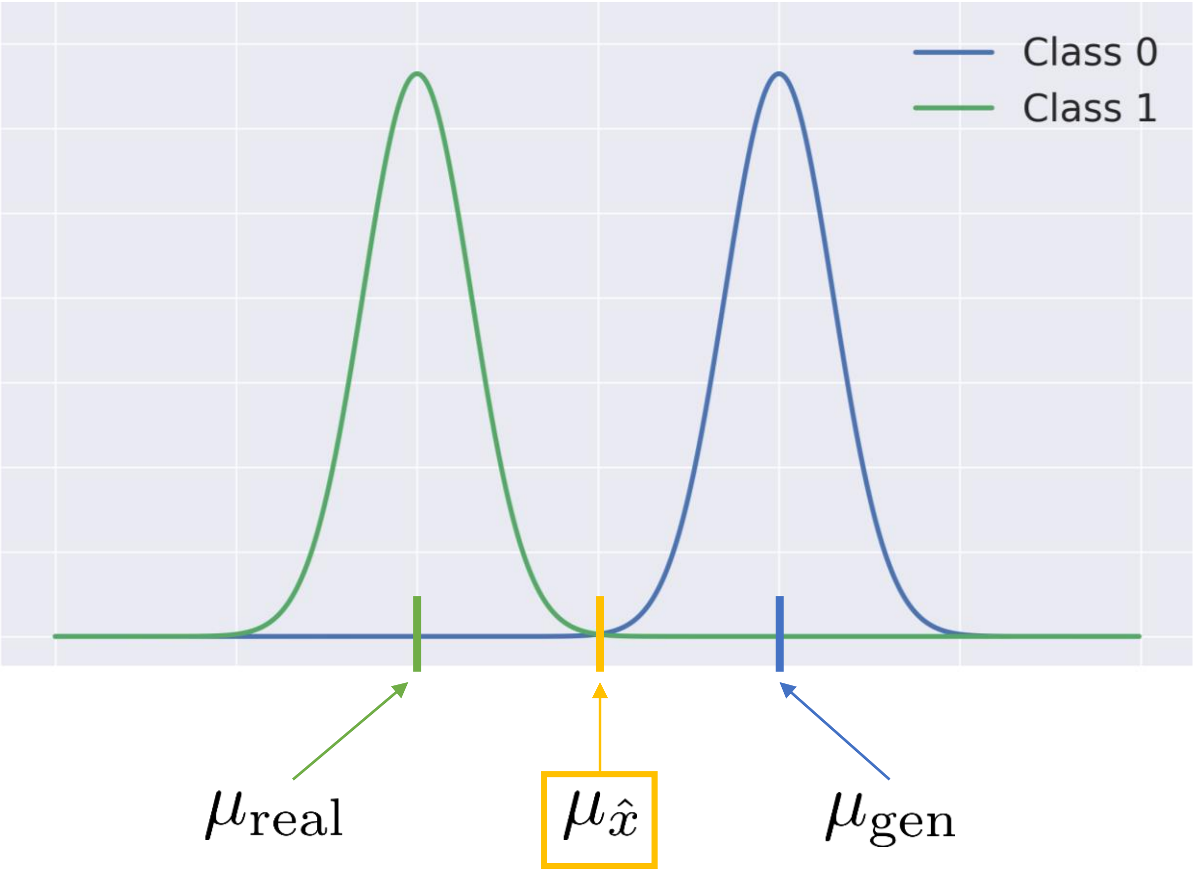
# Merged batches for real and generated data

Distribution of activations for real and generated data



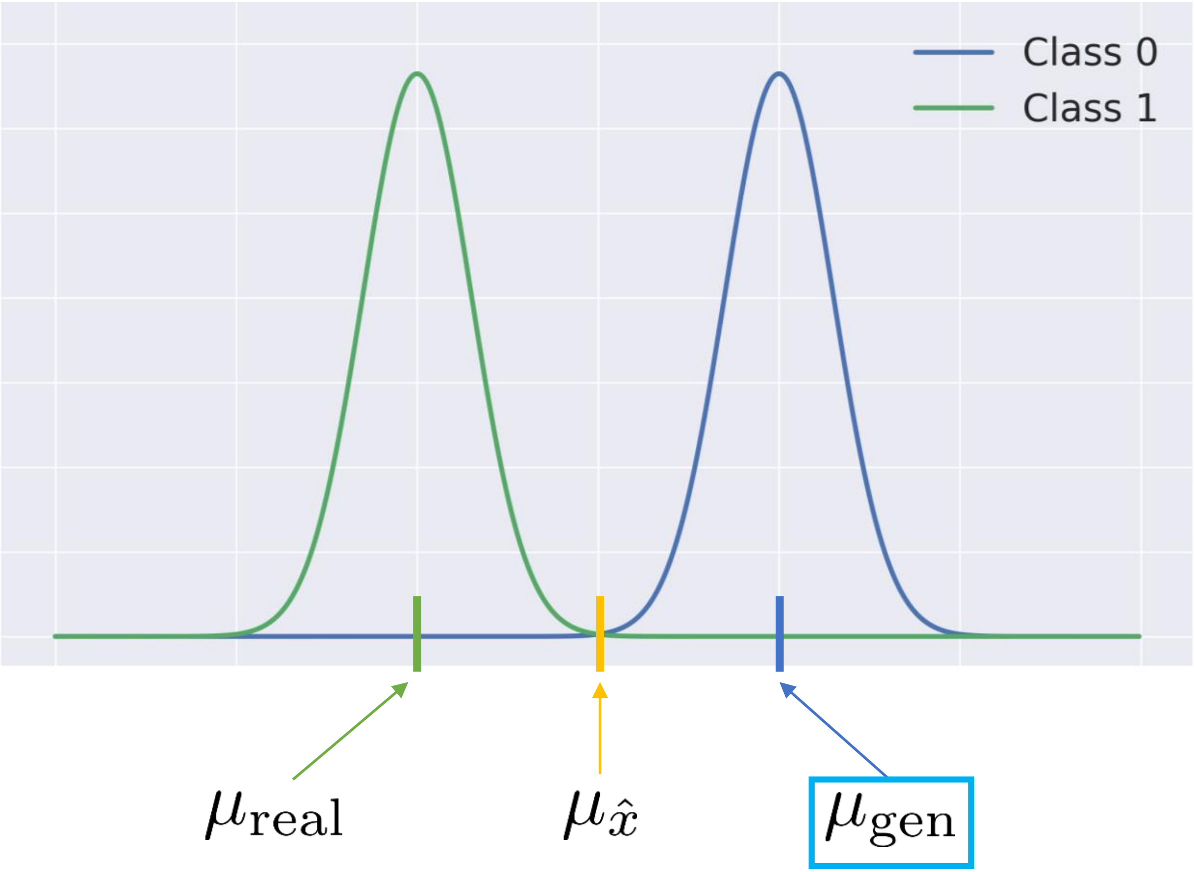
# Merged batches for real and generated data

$$\nabla_\phi \mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_{\hat{x},y}\left[\mathbb{I}[y=1]\frac{1}{D_\phi^{\mu_{\hat{x}}}}\frac{\partial D_\phi^{\mu_{\hat{x}}}}{\partial \phi} + \mathbb{I}[y=0]\frac{1}{1-D_\phi^{\mu_{\hat{x}}}}\frac{\partial D_\phi^{\mu_{\hat{x}}}}{\partial \phi}\right]$$



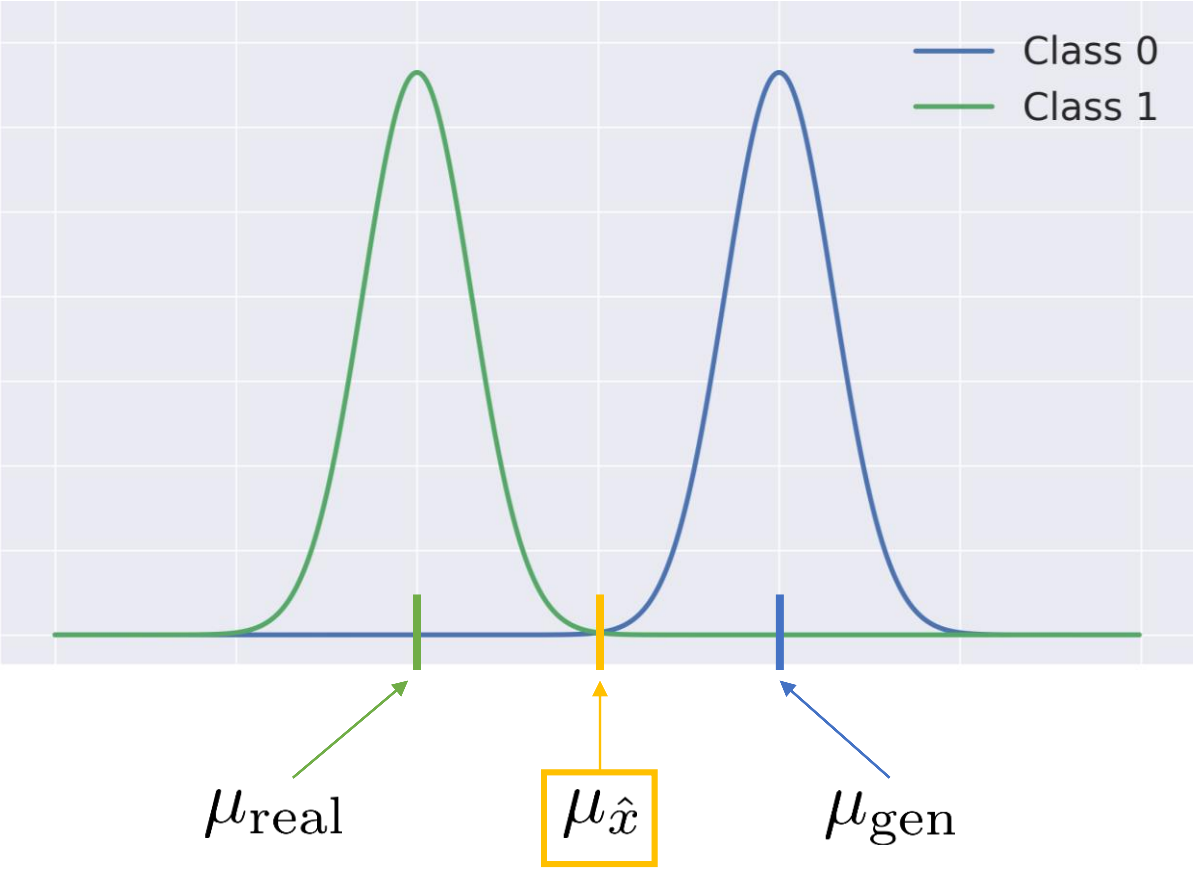
# Merged batches for real and generated data

$$\nabla_\theta \mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_z\left[\frac{1}{1 - D_\phi^{\mu_{\text{gen}}}} \frac{\partial D_\phi^{\mu_{\text{gen}}}}{\partial G_\theta} \frac{\partial G_\theta}{\partial \theta}\right]$$



# Merged batches for real and generated data

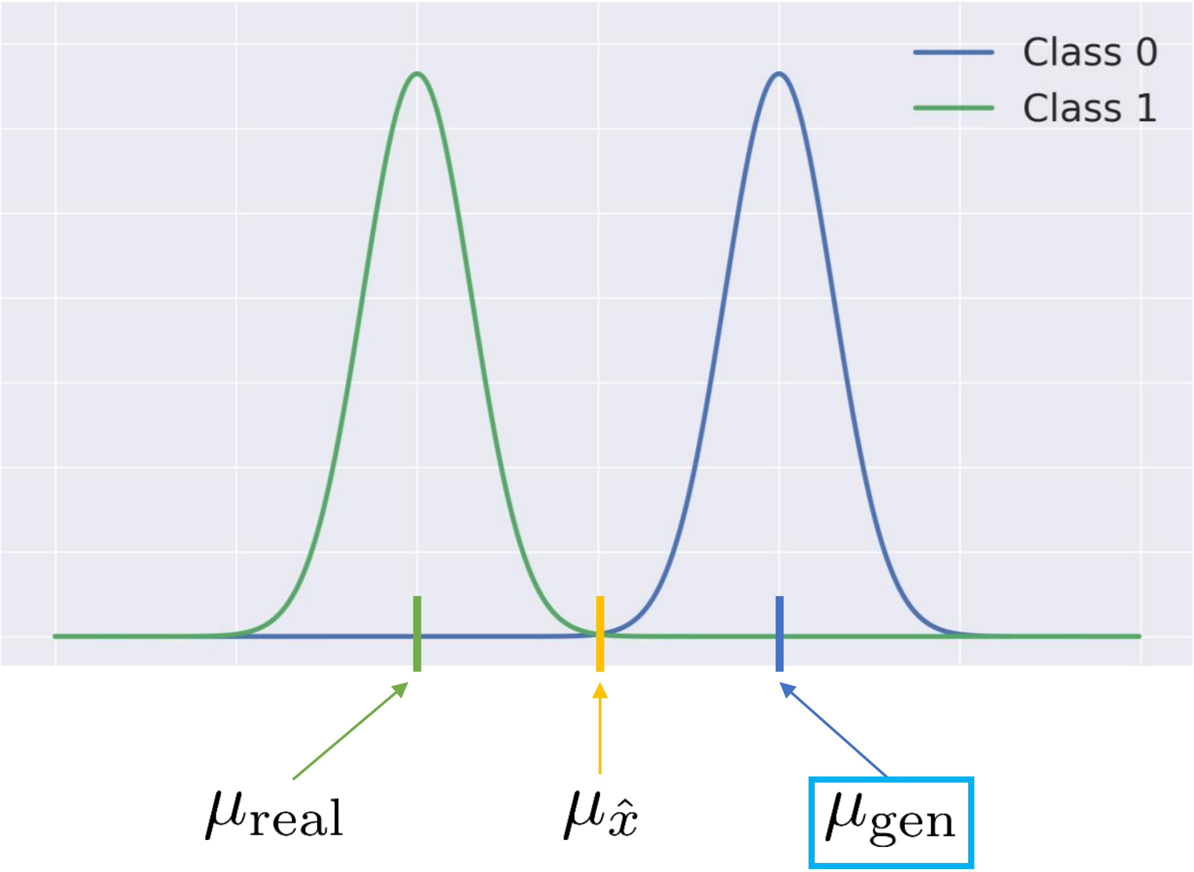
$$\nabla_\phi \mathcal{L}(\theta, \phi) = \mathbb{E}_{\hat{x},y}\left[\mathbb{I}[y=1]\frac{1}{D_\phi^{\mu_{\hat{x}}}}\frac{\partial D_\phi^{\mu_{\hat{x}}}}{\partial \phi} + \mathbb{I}[y=0]\boxed{\frac{1}{1-D_\phi^{\mu_{\hat{x}}}}\frac{\partial D_\phi^{\mu_{\hat{x}}}}{\partial \phi}}\right]$$




We optimize slightly different objectives during discriminator backward pass

# Merged batches for real and generated data

$$\nabla_\theta \mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_z\left[\boxed{\frac{1}{1 - D_\phi^{\mu_{\text{gen}}}} \frac{\partial D_\phi^{\mu_{\text{gen}}}}{\partial G_\theta}} \frac{\partial G_\theta}{\partial \theta}\right]$$

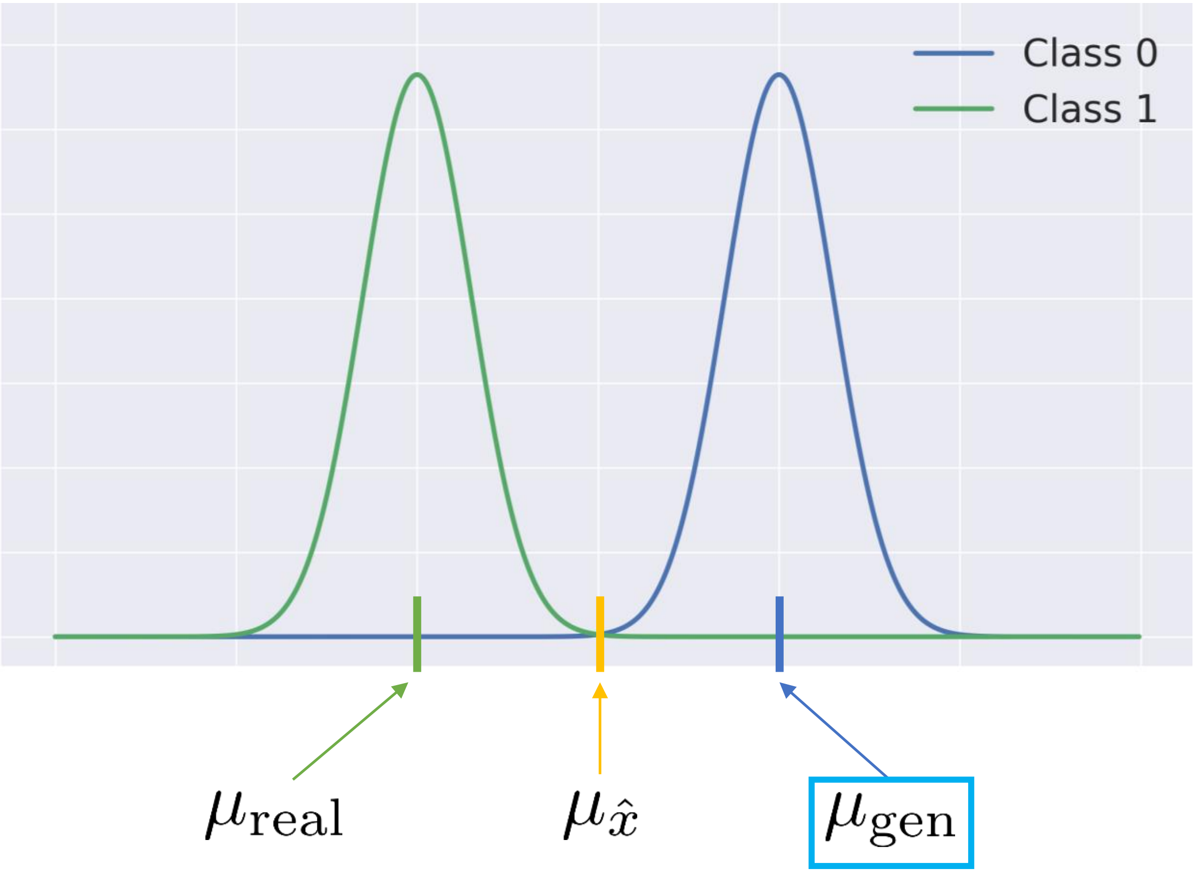



We optimize slightly different objectives during discriminator backward pass

And generator backward pass

# Merged batches for real and generated data

$$\nabla_\theta \mathcal{L}(\theta, \phi) = \mathbb{E}_z \left[ \boxed{\frac{1}{1 - D_\phi^{\mu_{\text{gen}}}} \frac{\partial D_\phi^{\mu_{\text{gen}}}}{\partial G_\theta}} \frac{\partial G_\theta}{\partial \theta} \right]$$



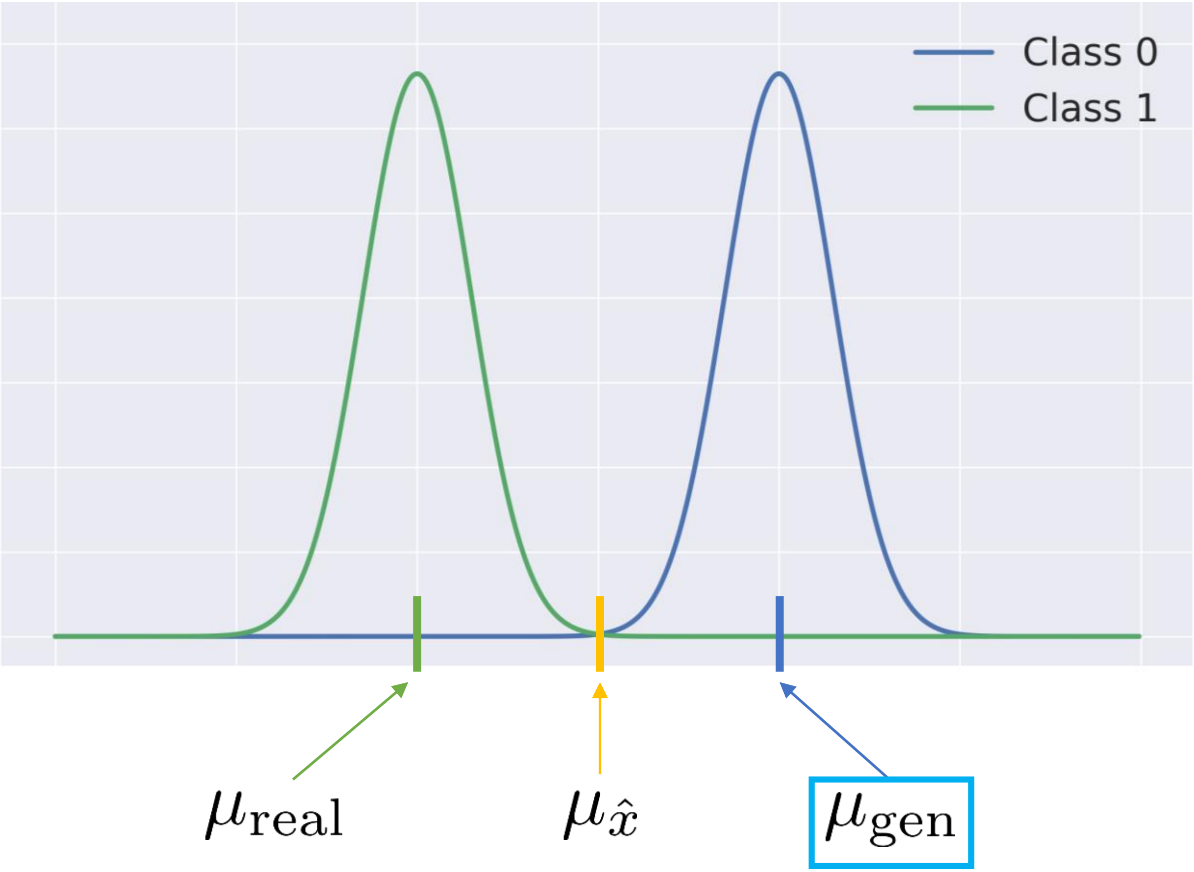

We optimize slightly different objectives during discriminator backward pass

And generator backward pass

$$\mathbb{E}(\mu_{\hat{x}} - \mu_{\text{gen}})^2 = (\mathbb{E}[\mu_{\text{gen}} - \mu_{\hat{x}}])^2 + \sigma^2_{\mu_{\text{gen}}} + \sigma^2_{\mu_{\hat{x}}}$$

# Merged batches for real and generated data

$$\nabla_\theta \mathcal{L}(\theta, \phi) \;=\; \mathbb{E}_z\left[\boxed{\frac{1}{1 - D_\phi^{\mu_{\text{gen}}}} \frac{\partial D_\phi^{\mu_{\text{gen}}}}{\partial G_\theta}} \frac{\partial G_\theta}{\partial \theta}\right]$$




We optimize slightly different objectives during discriminator backward pass

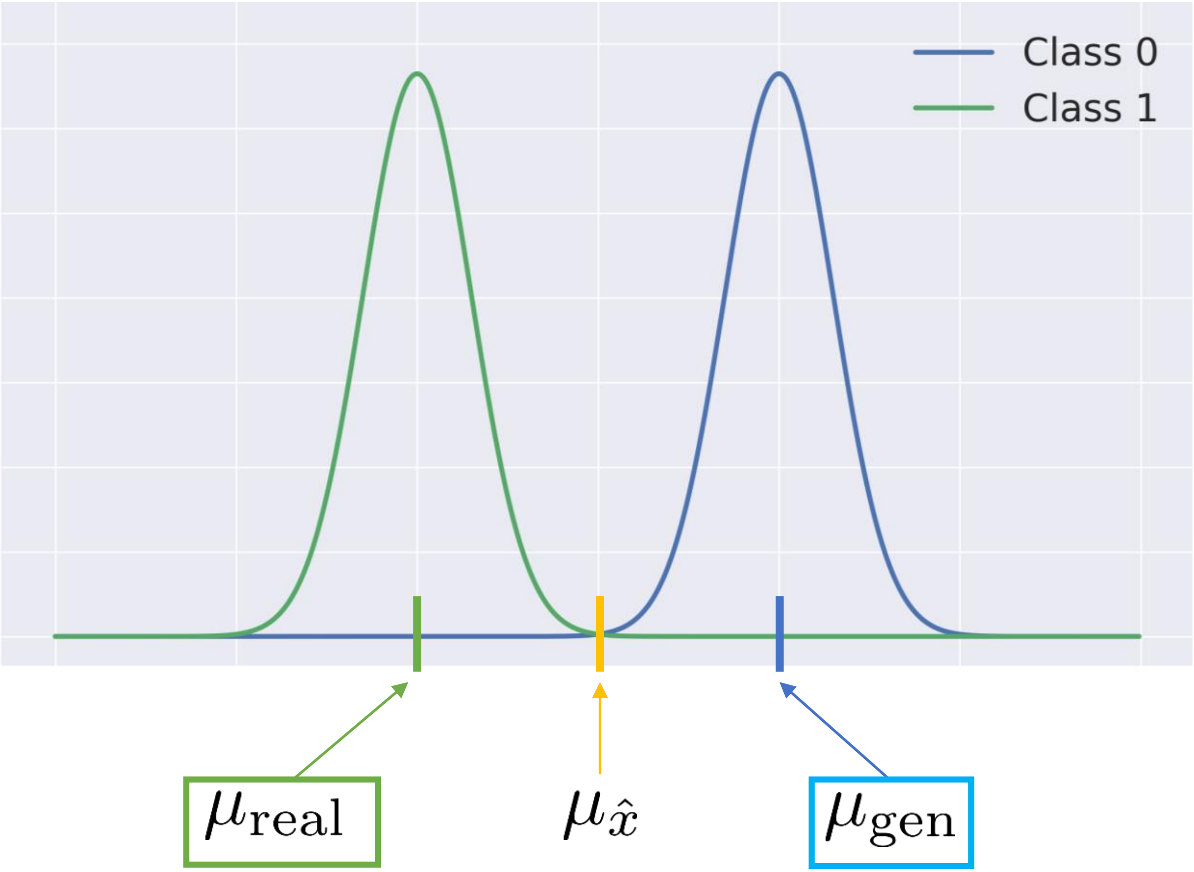
And generator backward pass

$$\mathbb{E}(\mu_{\hat{x}} - \mu_{\text{gen}})^2 = \underbrace{(\mathbb{E}[\mu_{\text{gen}} - \mu_{\hat{x}}])^2} + \sigma^2_{\mu_{\text{gen}}} + \sigma^2_{\mu_{\hat{x}}}$$

True when bias term is dominating (depends on the objective and architecture!)

# Merged batches for real and generated data

$$\nabla_\phi \mathcal{L}(\theta,\phi) = \mathbb{E}_{x\sim p(x)}\boxed{\left[\frac{1}{D_\phi^{\mu_{\text{real}}}(x)}\frac{\partial D_\phi^{\mu_{\text{real}}}(x)}{\partial \phi}\right]} + \mathbb{E}_{z\sim p(z)}\boxed{\left[\frac{1}{1-D_\phi^{\mu_{\text{gen}}}(G_\theta(z))}\frac{\partial D_\phi^{\mu_{\text{gen}}}(G_\theta(z))}{\partial \phi}\right]}$$




Recommended: separate batches of real and generated data

# GANs as a trainable objective

Another way to apply GANs:

- Pick a well defined problem where regular objectives can be used
- Use GAN as another unsupervised objectiv

# GANs as a trainable objective

Another way to apply GANs:

- Pick a well defined problem where regular objectives can be used
- Use GAN as another unsupervised objective

Example: super-resolution

- Given low resolution image, obtain its higher resolution version
- Objective:

$$\min_{\theta} ||G_{\theta}(x_{\mathrm{LR}}) - x_{\mathrm{HR}}||$$

# Super-resolution

Groundtruth

https://adeshpande3.github.io/Deep-Learning-Research-Review-Week-1-Generative-Adversarial-Nets

# Super-resolution

Groundtruth MSE

https://adeshpande3.github.io/Deep-Learning-Research-Review-Week-1-Generative-Adversarial-Nets

# Super-resolution

Groundtruth MSE MSE + GAN

https://adeshpande3.github.io/Deep-Learning-Research-Review-Week-1-Generative-Adversarial-Nets

# Summary

- In order to make optimization process converge close to the saddle point, multiple hyperparameters need to be tuned

# Summary

- In order to make optimization process converge close to the saddle point, multiple hyperparameters need to be tuned

- Git clone is the best way to tune a lot of them for free

# Summary

- In order to make optimization process converge close to the saddle point, multiple hyperparameters need to be tuned
- Git clone is the best way to tune a lot of them for free
- Use separate batches for real and fake data when using BatchNorm

# Summary

- In order to make optimization process converge close to the saddle point, multiple hyperparameters need to be tuned

- Git clone is the best way to tune a lot of them for free

- Use separate batches for real and fake data when using BatchNorm

- Combine GAN with other supervised objectives

# Applications


StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks

# Applications

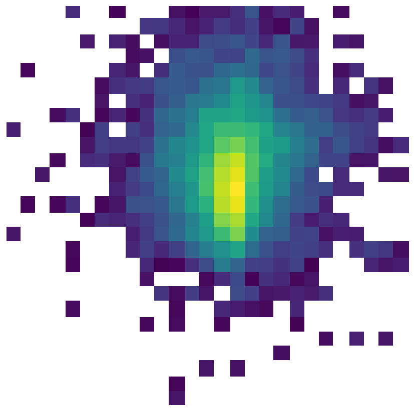
Problem: given initial data for the particle, produce high dimensional response of the detector

# Applications

Problem: given initial data for the particle, produce high dimensional response of the detector

$$p_x, p_y, p_z, \ldots \qquad \xrightarrow{\quad G \quad}$$

$$z - \text{latent variables}$$

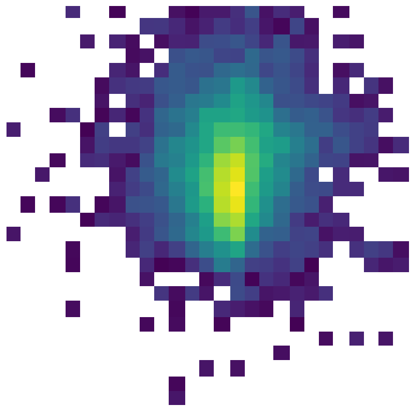


# Applications

Problem: given initial data for the particle, produce high dimensional response of the detector

$$p_x, p_y, p_z, \dots \xrightarrow{\quad G \quad}$$

$$z - \text{latent variables}$$



Physical model

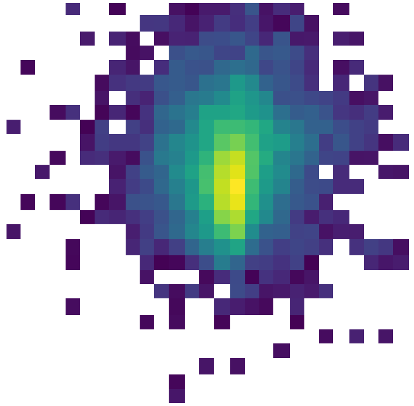
- Real physical equations are simulated
- Sampling is slow
- Accurate results

# Applications

Problem: given initial data for the particle, produce high dimensional response of the detector

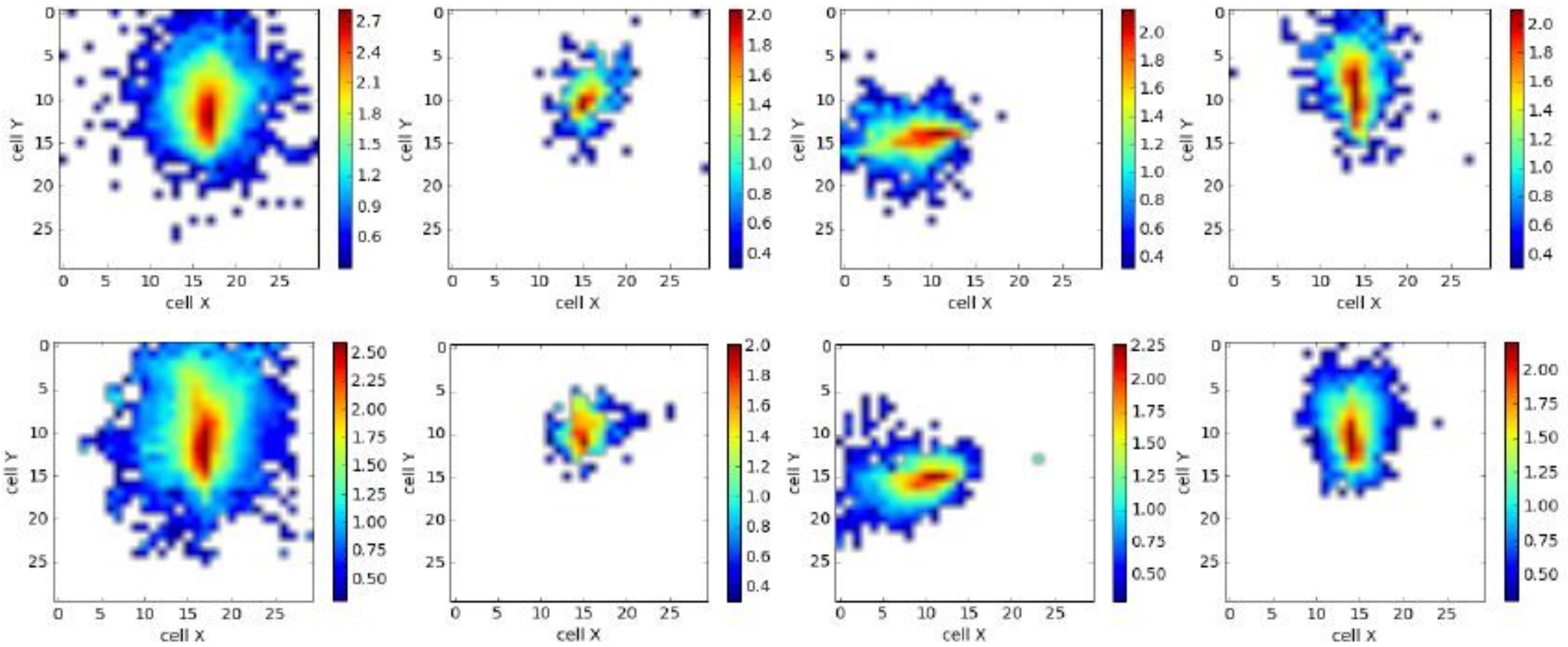
$$p_x, p_y, p_z, \dots$$
$$z - \text{latent variables}$$

$$\xrightarrow{G}$$



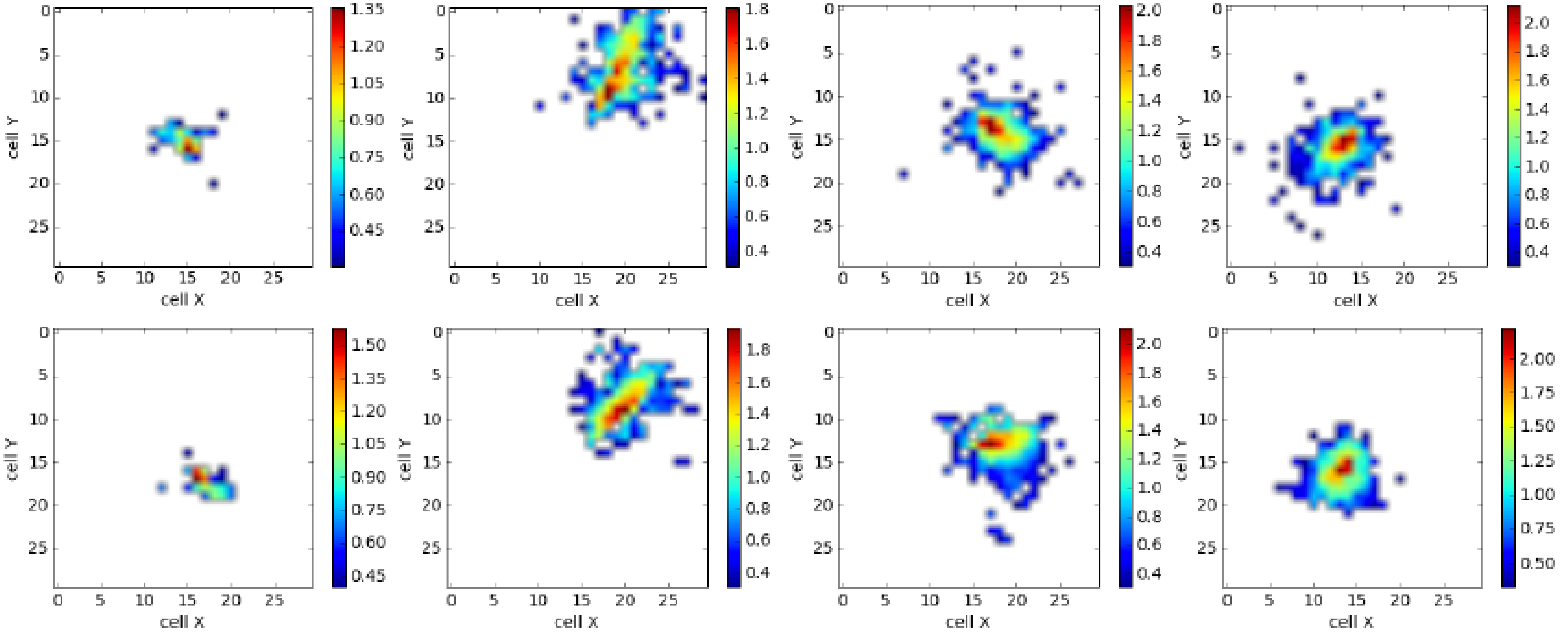
| Physical model | GANs |
|---|---|
| • Real physical equations are simulated | • Sampling function is approximated |
| • Sampling is slow | • Training takes days, sampling is free |
| • Accurate results | • Approximation quality needs to be verified |

# Applications



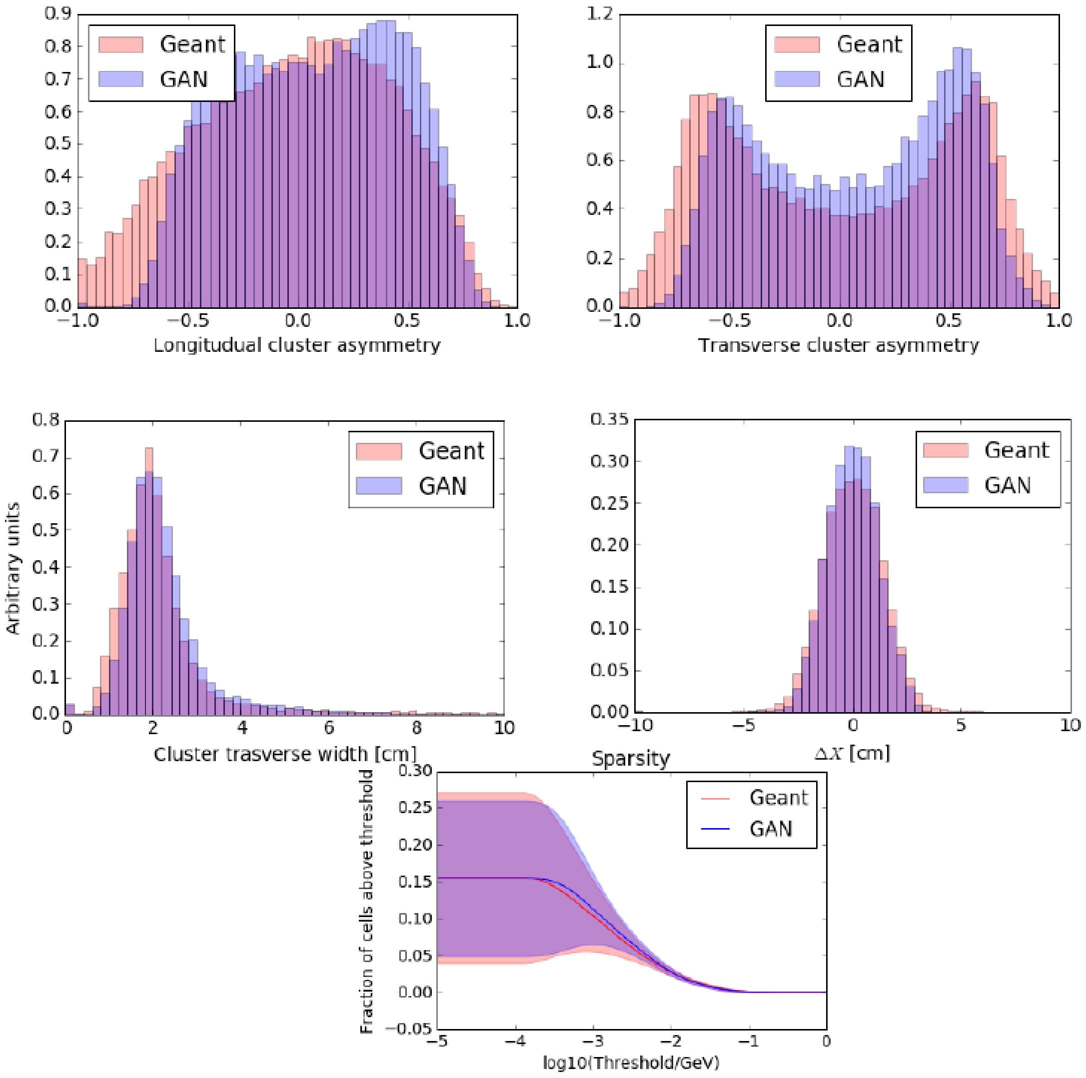

First row: simulated data, second row: data generated using GAN

# Applications




First row: simulated data, second row: data generated using GAN

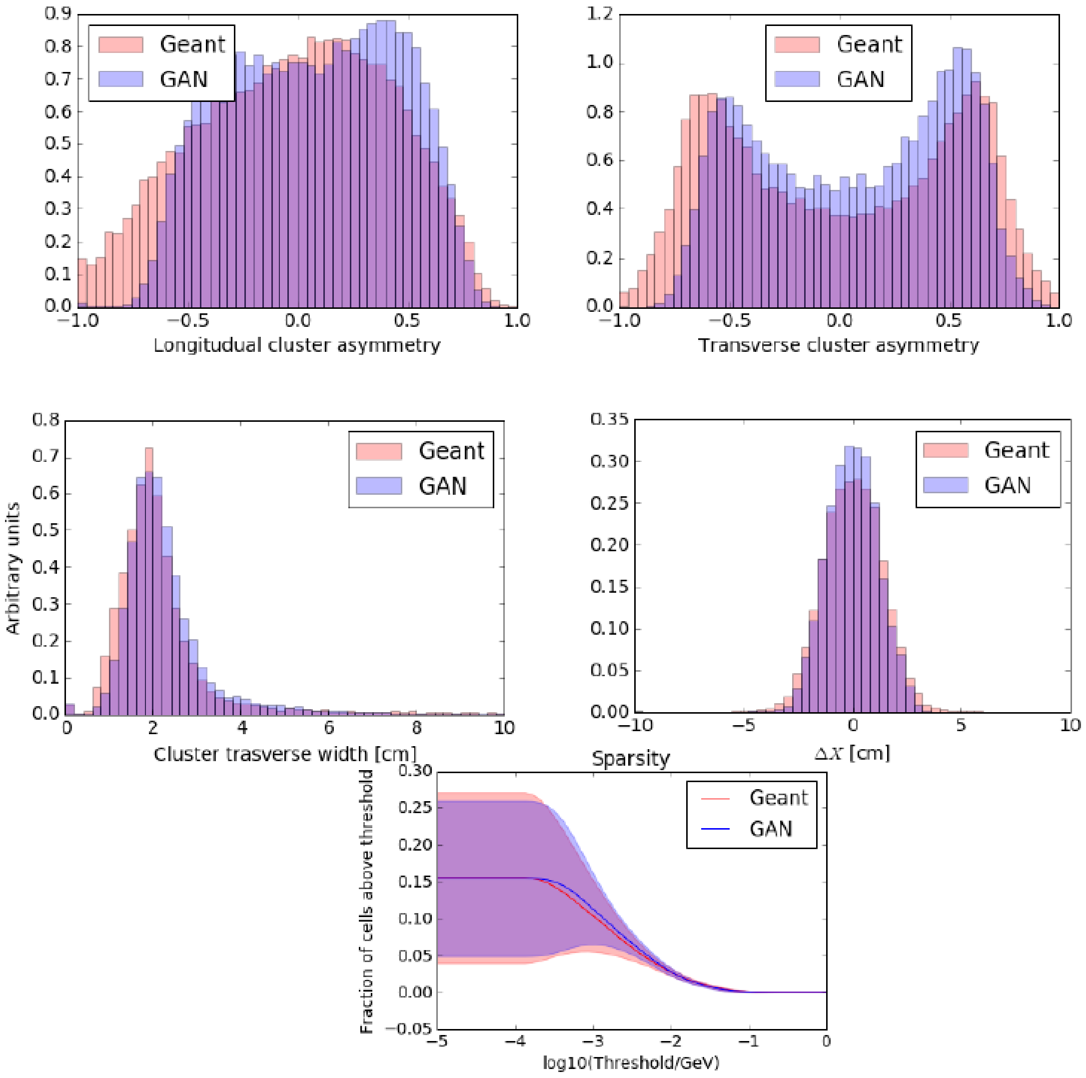
# Applications




<u>Theorem (informal)</u>

Two distributions match if and only if all their statistics match

# Applications

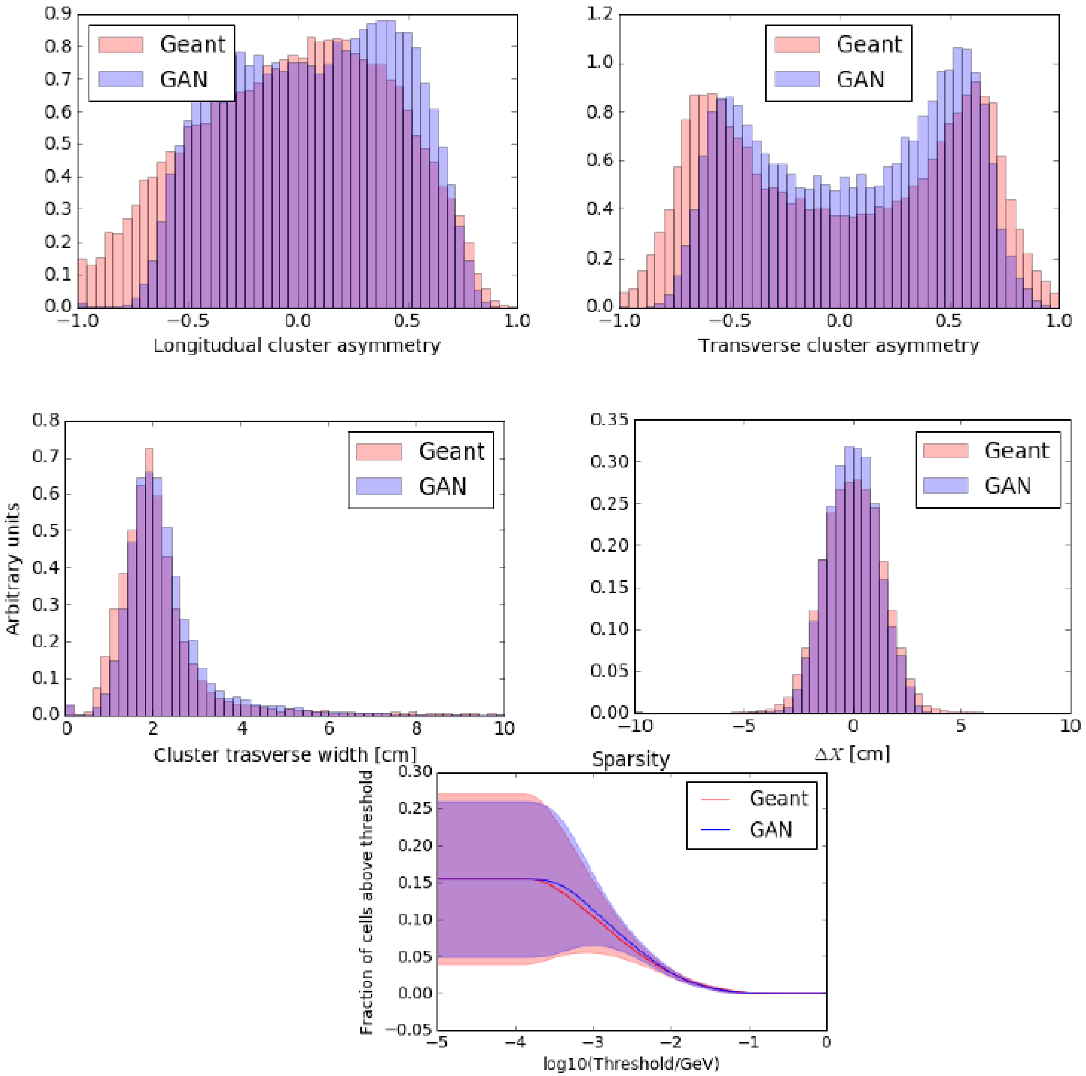



__Theorem (informal)__

Two distributions match if and only if all their statistics match

Validation method: compare statistics between real data from test set (unseen during training) and generated data

# Applications




<u>Theorem (informal)</u>

Two distributions match if and only if all their statistics match

Validation method: compare statistics between real data from test set (unseen during training) and generated data

If statistics match, this is an indication that p(x) matches q(x)

# Summary

# Summary

- GANs cannot be explained only by optimization of divergence

# Summary

- GANs cannot be explained only by optimization of divergence

- Training requires finetuning of many hyperparameters, use other peoples' work instead of doing it yourself

# Summary

- GANs cannot be explained only by optimization of divergence
- Training requires finetuning of many hyperparameters, use other peoples' work instead of doing it yourself
- Produce solid practical results as an auxiliary objective for the existing problem

# Summary

- GANs cannot be explained only by optimization of divergence
- Training requires finetuning of many hyperparameters, use other peoples' work instead of doing it yourself
- Produce solid practical results as an auxiliary objective for the existing problem
- Good application of GANs as a generative model is still work in progress