DeepMind

How to build your GAN loss from distributional divergences and distances

Mihaela Rosca

Staff Research Engineer at DeepMInd
PhD student at UCL

Prob AI 2021

Generative models

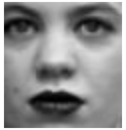
Aim: learn a probabilistic model from data.

Generative adversarial networks

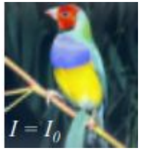
Want to learn more?

Goodfellow, et al. **Generative adversarial networks..** Neural Information Processing Systems (2014)

Learning an implicit model through a two player game.



Goodfellow, et al. **Generative adversarial networks.** NIPS **(2014)**



Denton, et al. **Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks.** NIPS **(2015)**

Radford et al. **Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks** ICLR **(2015)**

Miyato et al. **Spectral normalization for Generative Adversarial Networks** ICLR **(2018)**

Karras et al. **Large Scale GAN Training for High Fidelity Natural Image Synthesis** ICLR **(2018)**

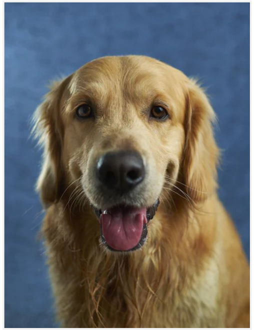
Brock et al. **Large Scale GAN Training for High Fidelity Natural Image Synthesis** ICLR **(2019)**

Karras et al. **A Style-Based Generator Architecture for Generative Adversarial Networks** CVPR **(2019)**

Generative adversarial networks

Discriminator

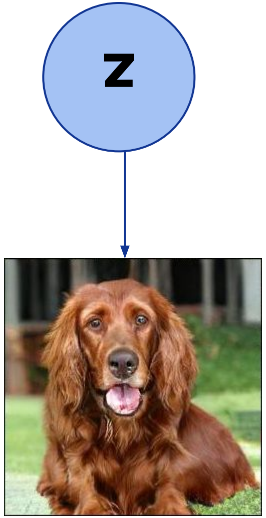
Learns to distinguish between real and generated data.



vs

Generator

Learns to generate data to “fool” the discriminator.

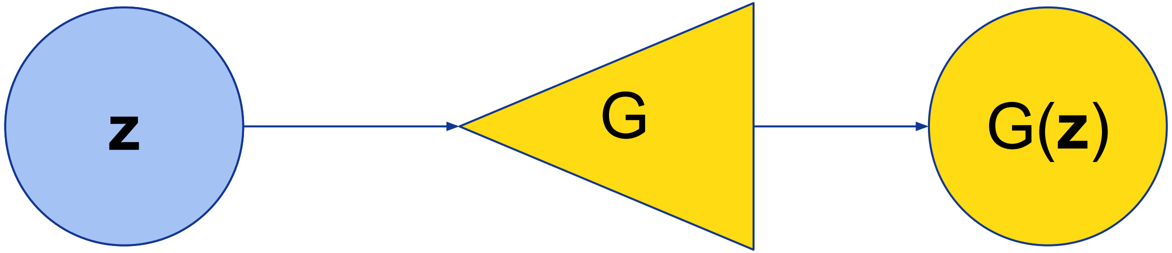


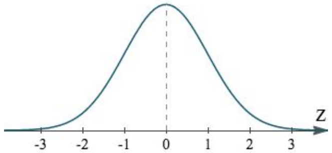
Generator

latent (“noise”) vector
$\mathbf{z} \sim P(\mathbf{z})$

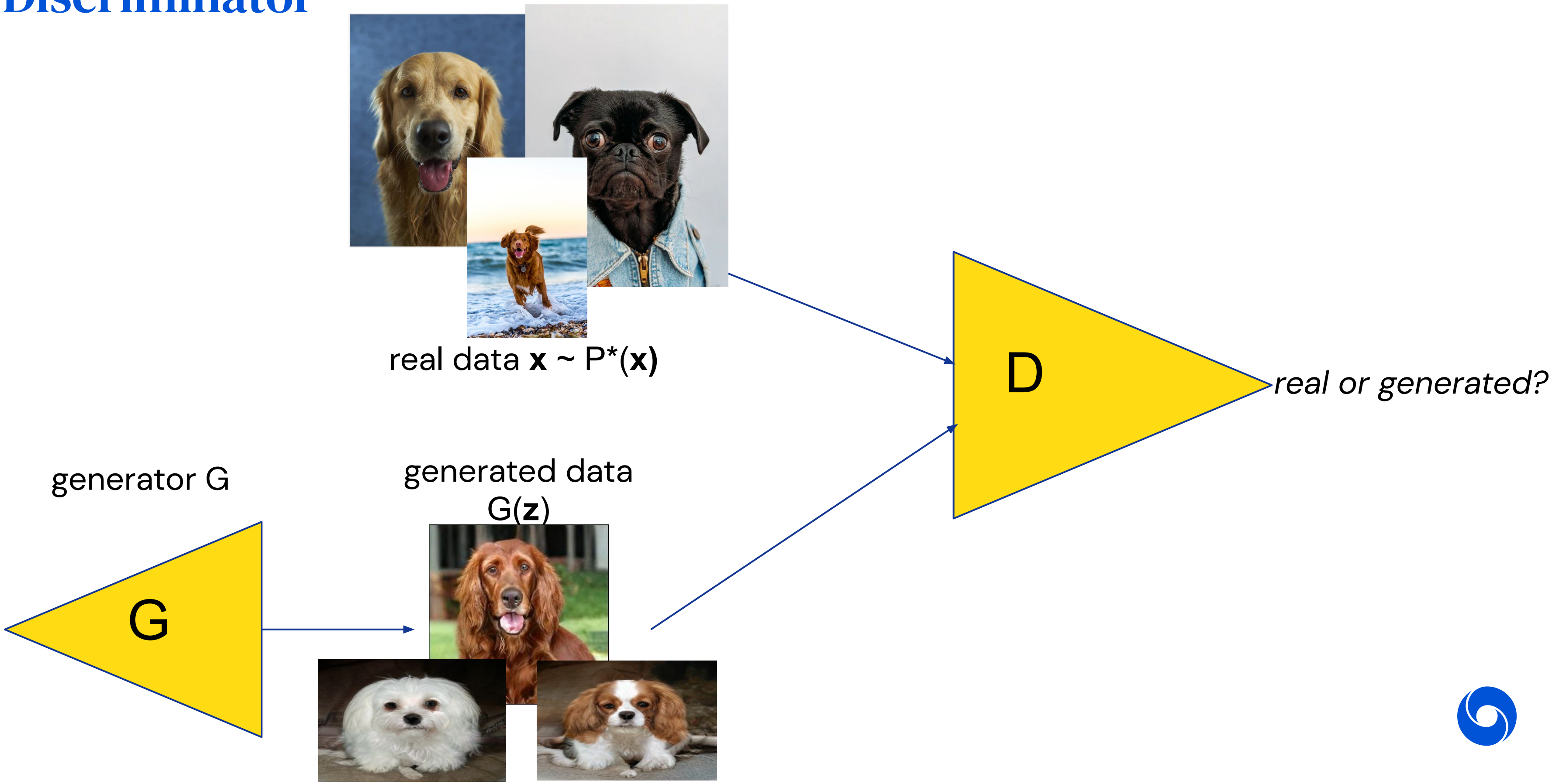
generator G:
a deep neural network

generated data
$G(\mathbf{z})$



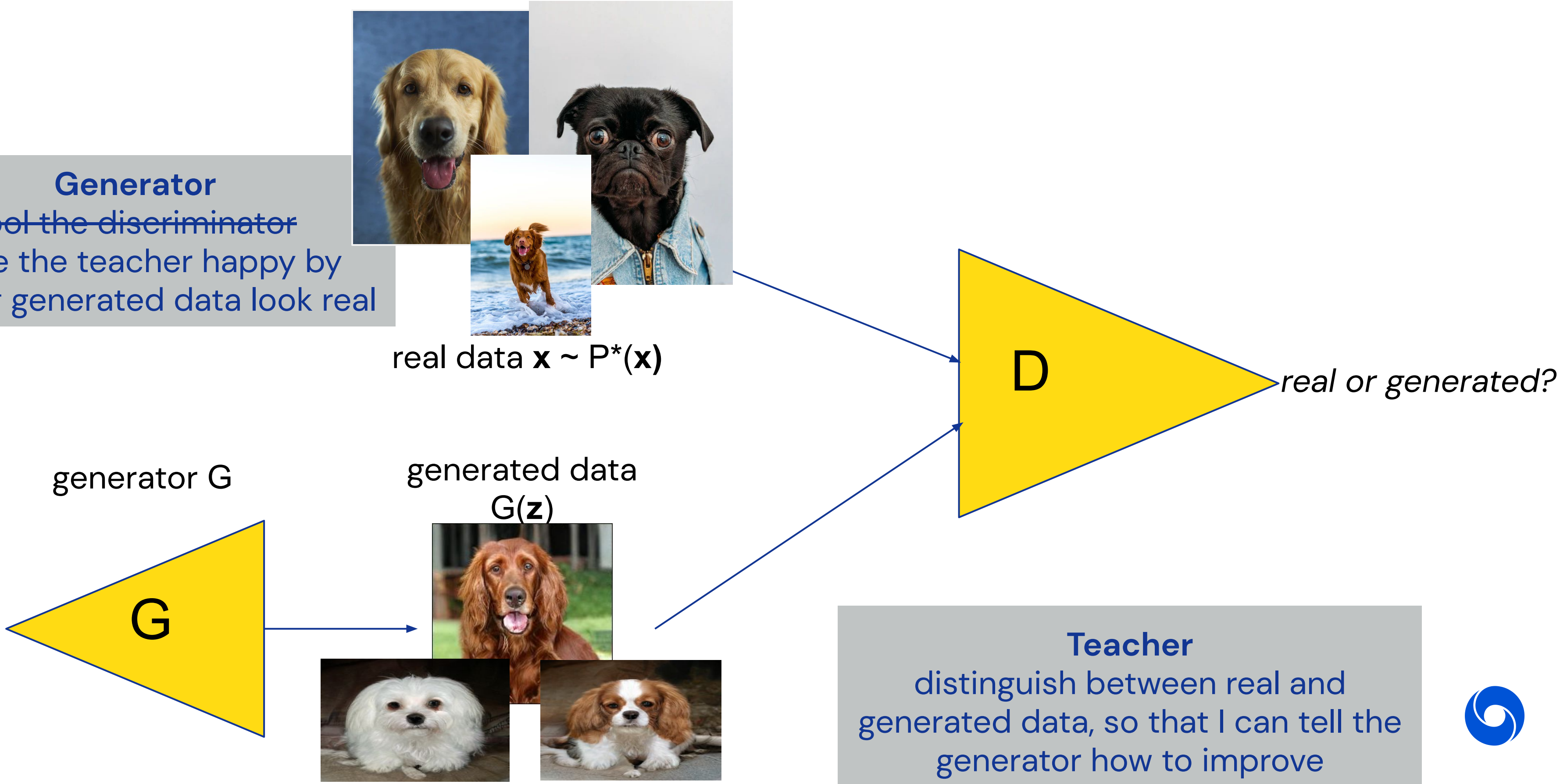


Discriminator



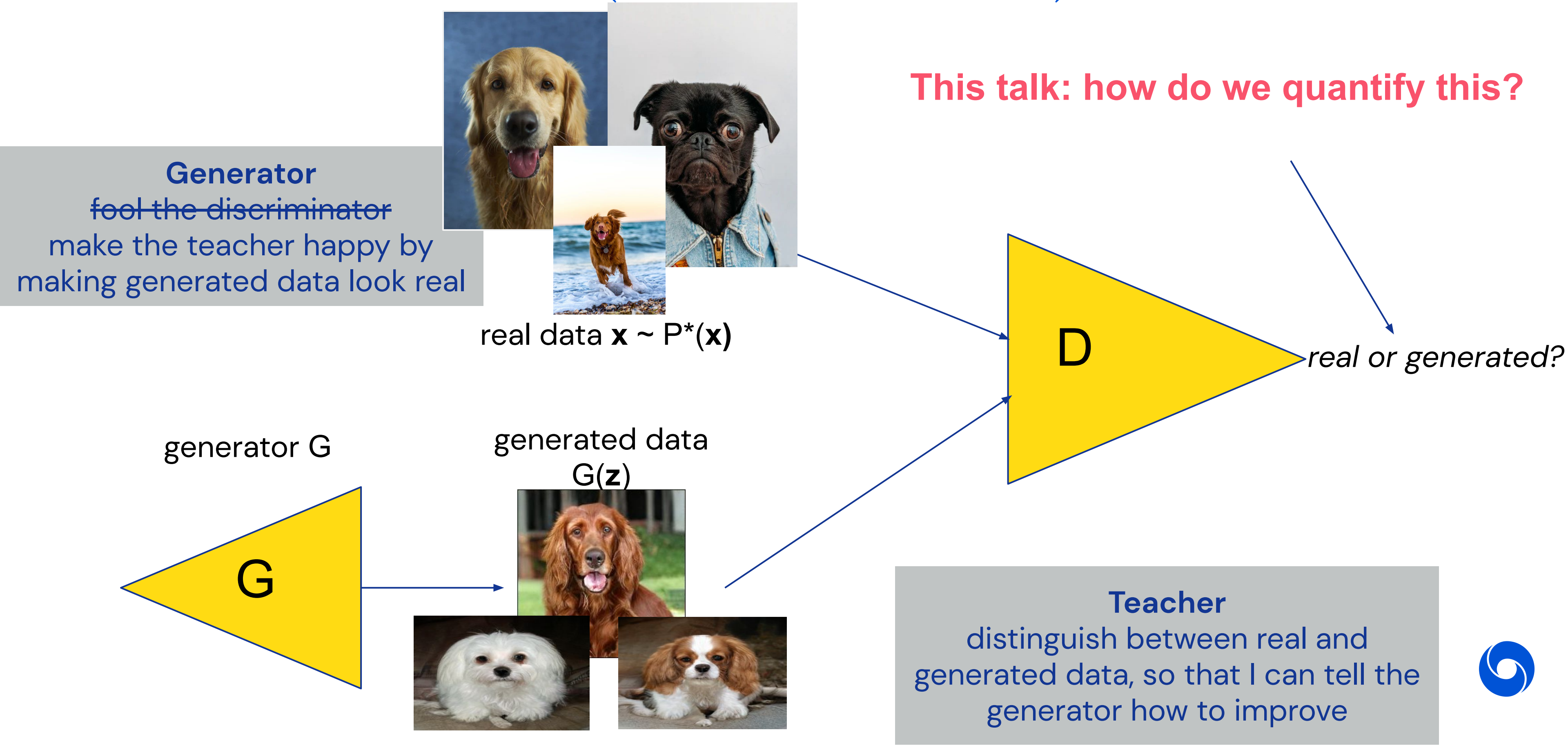
~~Discriminator~~ Teacher

Generator
~~fool the discriminator~~
make the teacher happy by making generated data look real



Teacher
distinguish between real and generated data, so that I can tell the generator how to improve

~~Discriminator~~ Teacher (less adversarial view)



Original GAN

Original Generative Adversarial Network

Want to learn more?

Goodfellow, et al. **Generative adversarial networks..** Neural Information Processing Systems (**2014**)

$$\min_G \max_D V(D, G) = \underbrace{\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]}_{\text{log-probability that D correctly predicts real data } \mathbf{x} \text{ are real}} + \underbrace{\mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}_{\text{log-probability that D correctly predicts generated data G(}\mathbf{z}\text{) are generated}}$$

log-probability that D correctly predicts real data **x** are real

log-probability that D correctly predicts generated data G(**z**) are generated

Generative Adversarial Networks

Want to learn more?

Goodfellow, et al. **Generative adversarial networks..** Neural Information Processing Systems (**2014**)

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

log-probability that D correctly predicts real data **x** are real

log-probability that D correctly predicts generated data G(**z**) are generated

Discriminator's (D) goal: **maximize** prediction accuracy

Generator's (G) goal: **minimize** D's prediction accuracy.

Generative adversarial networks

Want to learn more?

Goodfellow, et al. **Generative adversarial networks..** Neural Information Processing Systems (**2014**)

for number of training iterations **do**

for k steps **do**

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

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

end for

Algorithm from Goodfellow et al. (2014)

Generative Adversarial Networks as zero sum game

$$\min_G \max_D V(D, G)$$

- Bi level optimization of the same loss function.
- Connection to game theory literature.
    - Nash equilibria
    - Strategies
    - Fictitious play

Generative models as divergence or distance minimization

- Generative models **often to minimize a divergence or distance**.
- Most common: Maximum likelihood (KL divergence).

Why divergence/distance minimization?

$$D(p^*, p) \geq 0$$

$$D(p^* || p) = 0 \implies p = p^*$$

Are GANs doing divergence minimization?

Want to learn more?

Goodfellow, et al. **Generative adversarial networks..** Neural Information Processing Systems (2014)

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

If the discriminator (D) is optimal:
the generator is minimizing the Jensen Shannon divergence between the true and generated distributions.

Are GANs doing divergence minimization?

Want to learn more?

Goodfellow, et al. **Generative adversarial networks..** Neural Information Processing Systems (2014)

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

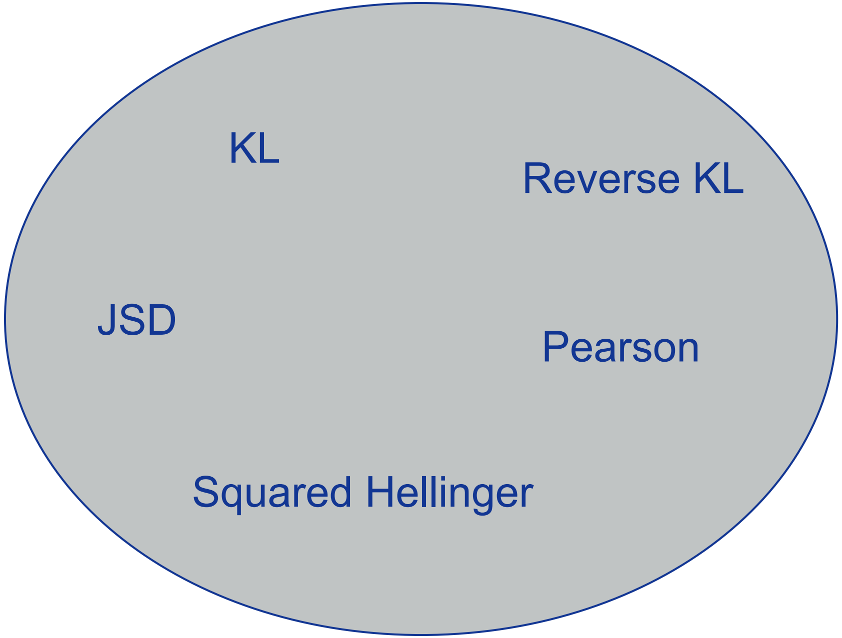
If the discriminator (D) is optimal:
the generator is minimizing the Jensen Shannon divergence between the true and generated distributions.

Connection to optimality:

$$JSD(p^*||p) = 0 \implies p = p^*$$

From f-divergences

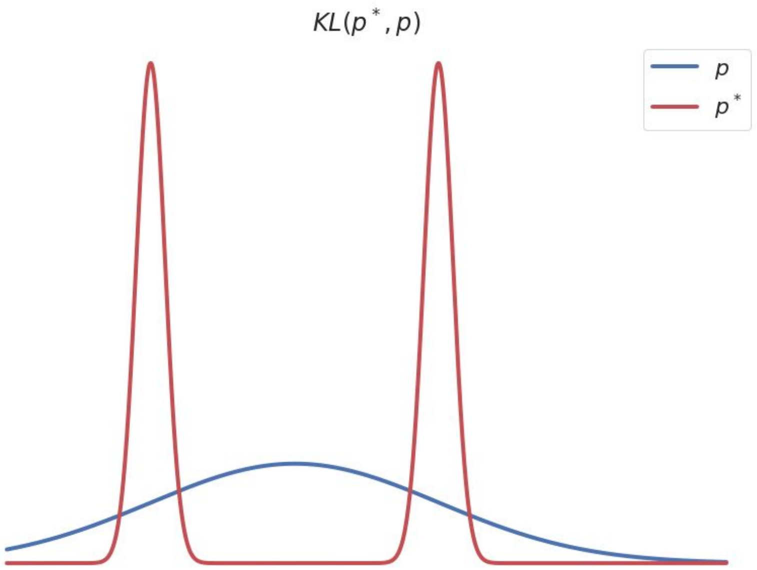
f-divergences

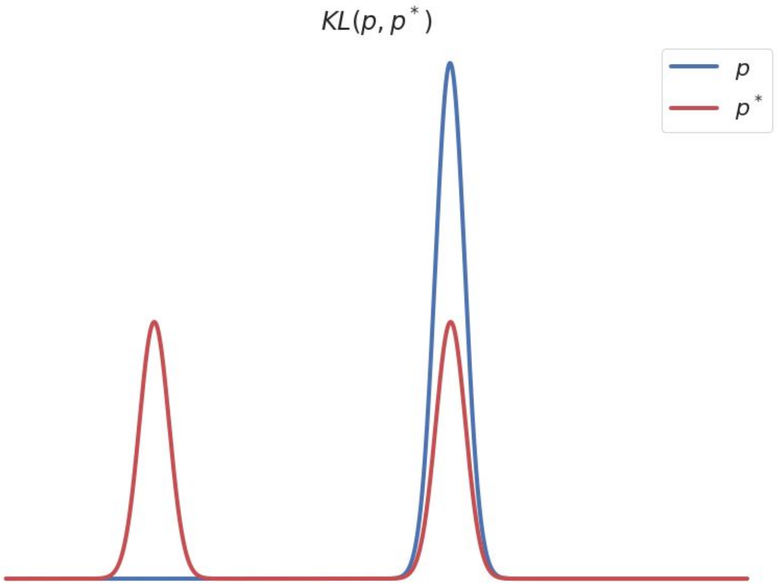


Effects of the choice of divergence

Want to learn more?

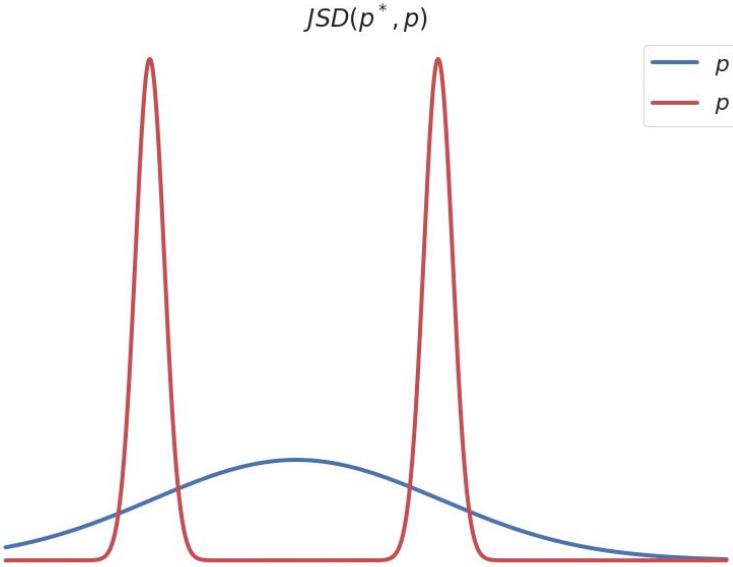
Goodfellow, et al. **NIPS 2016 Tutorial: Generative Adversarial Networks** Arxiv (**2016**)

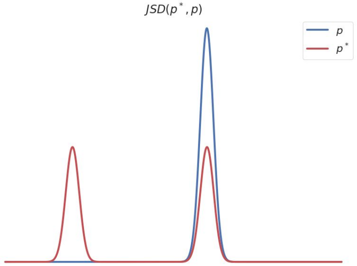




Jensen Shannon divergence

$$\mathrm{JSD}(p, p^*) = \frac{1}{2}\mathrm{KL}(p, \frac{p+p^*}{2}) + \frac{1}{2}\mathrm{KL}(p^*, \frac{p+p^*}{2})$$





f-divergences

Want to learn more?

Nowozin, et al **f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization.** Neural Information Processing Systems **(2016)**

$$D_f(p^*||p) = \int p(x) f\left(\frac{p^*(x)}{p(x)}\right) dx$$

f convex, semi continuous and f(1) = 0.

Examples of f-divergences

Want to learn more?

Nowozin, et al **f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization.** Neural Information Processing Systems (2016)

Name	$D_f(P\|Q)$	$f(u)$
Kullback-Leibler	$\int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x$	$u \log u$
Reverse KL	$\int q(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$	$-\log u$
Pearson χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)} \, \mathrm{d}x$	$(u-1)^2$
Squared Hellinger	$\int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 \, \mathrm{d}x$	$(\sqrt{u}-1)^2$
Jensen-Shannon	$\frac{1}{2} \int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} \, \mathrm{d}x$	$-(u+1) \log \frac{1+u}{2} + u \log u$
GAN	$\int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} \, \mathrm{d}x - \log(4)$	$u \log u - (u+1) \log(u+1)$

Challenge with f-divergences

unknown!

$$D_f(p^*||p) = \int p(x) f\left(\frac{p^*(x)}{p(x)}\right) dx$$

KL divergence

$$\mathrm{KL}(p^*(\mathbf{x})||p_\theta(\mathbf{x})) = \int p^*(\mathbf{x}) \log \frac{p^*(\mathbf{x})}{p_\theta(\mathbf{x})} d\mathbf{x}$$

$$= C - \int p^*(\mathbf{x}) \log p_\theta(\mathbf{x}) d\mathbf{x}$$

JSD - cannot do the same trick

$$\mathrm{JSD}(p, p^*) = \frac{1}{2}\mathrm{KL}(p, \frac{p + p^*}{2}) + \frac{1}{2}\mathrm{KL}(p^*, \frac{p + p^*}{2})$$

requires knowledge of mixture, unknown!

Variational bound on *f*-divergences

Want to learn more?

Nowozin, et al **f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization.** Neural Information Processing Systems **(2016)**

$$D_f(p^*, p) = \int p(x) f\left(\frac{p^*(x)}{p(x)}\right) dx$$

f convex:

$$f(x) = \sup_t tx - f^\dagger(t)$$

Variational bound on *f*-divergences

Want to learn more?

Nowozin, et al **f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization.** Neural Information Processing Systems **(2016)**

$$
\begin{aligned}
D_f(p^*, p) &= \int p(x) f\left(\frac{p^*(x)}{p(x)}\right) dx \\
&= \int p(x) \sup_t \left[t \frac{p^*(x)}{p(x)} - f^\dagger(t)\right] dx \\
&= \int \sup_{t(x)} p(x) \left[t(x) \frac{p^*(x)}{p(x)} - f^\dagger(t(x))\right] dx
\end{aligned}
$$

Variational bound on *f*-divergences

Want to learn more?

Nowozin, et al **f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization.** Neural Information Processing Systems **(2016)**

$$\begin{aligned}
D_f(p^*, p) &= \int p(x) f\left(\frac{p^*(x)}{p(x)}\right) dx \\
&= \int p(x) \sup_t \left[t \frac{p^*(x)}{p(x)} - f^\dagger(t)\right] dx \\
&= \int \sup_{t(x)} p(x) \left[t(x) \frac{p^*(x)}{p(x)} - f^\dagger(t(x))\right] dx \\
&= \int \sup_{t(x)} t(x) p^*(x) - p(x) f^\dagger(t(x)) dx \\
&= \sup_{t(x)} \int t(x) p^*(x) - p(x) f^\dagger(t(x)) dx \\
&= \sup_{t(x)} \mathbf{E}_{p^*(x)} t(x) - \mathbf{E}_{p(x)} f^\dagger(t(x))
\end{aligned}$$

Variational bound on *f*-divergences

Want to learn more?

Nowozin, et al **f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization.** Neural Information Processing Systems **(2016)**

$$
\begin{aligned}
D_f(p^*, p) &= \sup_{t:\mathcal{X}\to\mathbb{R}} \mathbb{E}_{p^*(x)} t(x) - \mathbb{E}_{p(x)} f^\dagger(t(x)) dx \\
&\geq \sup_{t\in\mathcal{T}} \mathbb{E}_{p^*(x)} t(x) - \mathbb{E}_{p(x)} f^\dagger(t(x)) dx
\end{aligned}
$$

Variational bounds on f-divergences

Want to learn more?

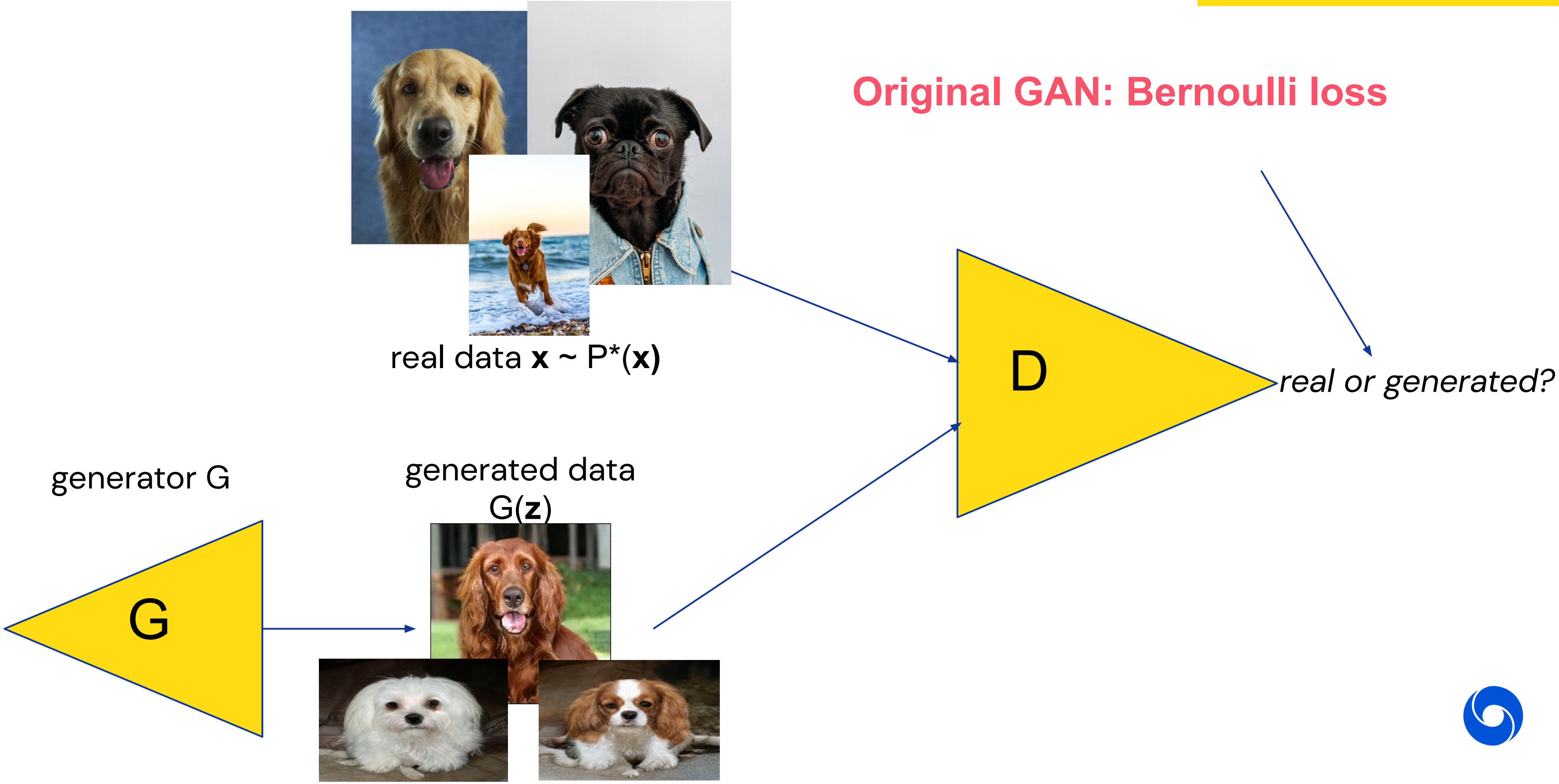
Nowozin, et al **f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization.** Neural Information Processing Systems (**2016**)

Name	$D_f(P\|Q)$	Generator $f(u)$	$T^*(x)$
Kullback-Leibler	$\int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x$	$u \log u$	$1 + \log \frac{p(x)}{q(x)}$
Reverse KL	$\int q(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$	$-\log u$	$-\frac{q(x)}{p(x)}$
Pearson χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)} \, \mathrm{d}x$	$(u-1)^2$	$2(\frac{p(x)}{q(x)} - 1)$
Squared Hellinger	$\int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 \mathrm{d}x$	$(\sqrt{u}-1)^2$	$(\sqrt{\frac{p(x)}{q(x)}} - 1) \cdot \sqrt{\frac{q(x)}{p(x)}}$
Jensen-Shannon	$\frac{1}{2} \int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} \, \mathrm{d}x$	$-(u+1)\log \frac{1+u}{2} + u \log u$	$\log \frac{2p(x)}{p(x)+q(x)}$
GAN	$\int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} \, \mathrm{d}x - \log(4)$	$u \log u - (u+1)\log(u+1)$	$\log \frac{p(x)}{p(x)+q(x)}$

Connection to proper scoring rules

Want to learn more?

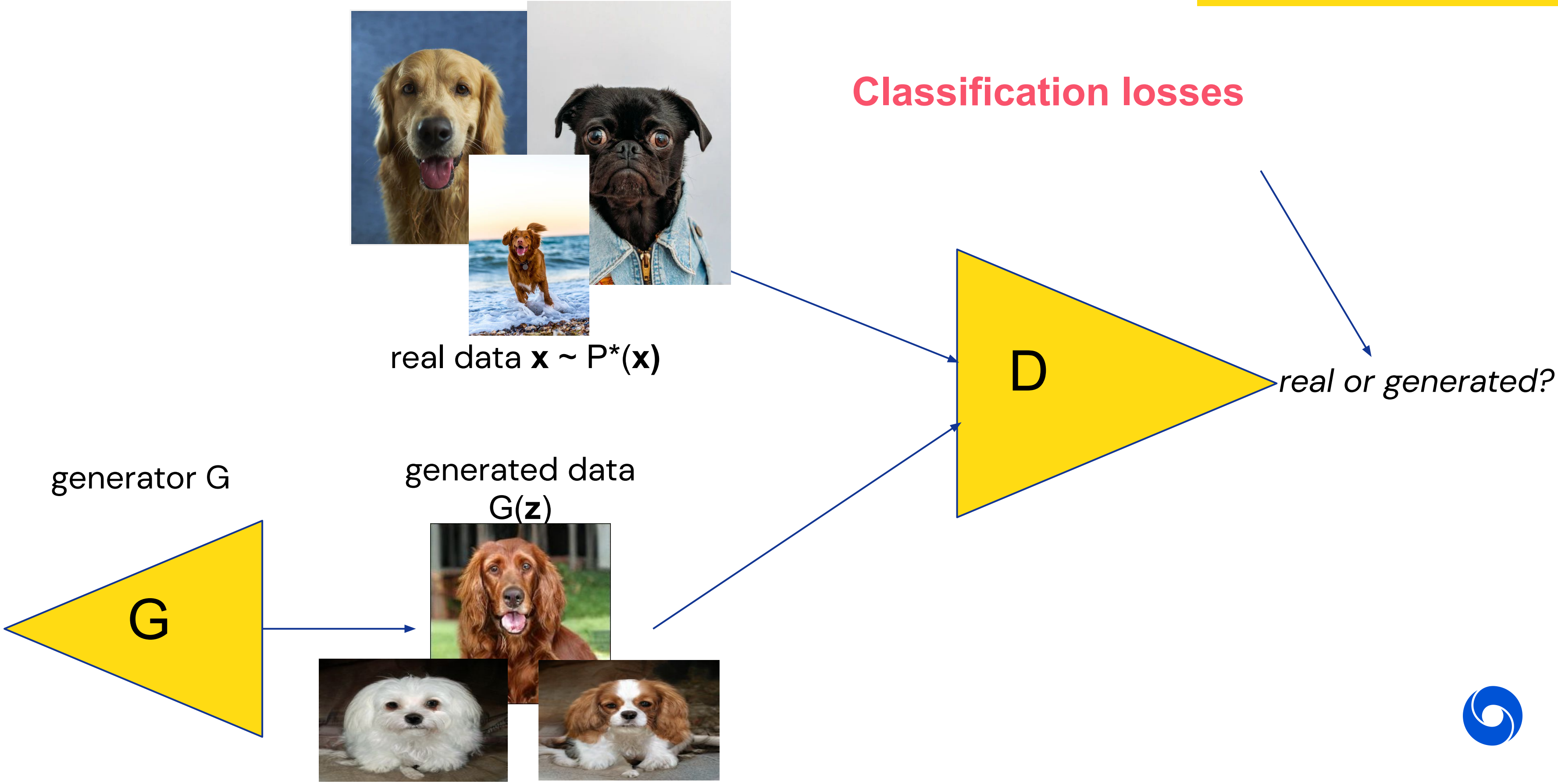
Mohamed, et al **Learning in implicit generative models** arxiv (2016)



Connection to proper scoring rules

Want to learn more?

Mohamed, et al **Learning in implicit generative models** arxiv (2016)



Connection to proper scoring rules

Want to learn more?
Mohamed, et al **Learning in implicit generative models** arxiv (2016)

Loss	Objective Function
Bernoulli loss	$\mathbb{E}_{p^*(x)}[\log \mathcal{D}] + \mathbb{E}_{p(x)}[\log(1-\mathcal{D})]$
Brier score	$\mathbb{E}_{p^*(x)}[-(1-\mathcal{D})^2] + \mathbb{E}_{p(x)}[-\mathcal{D}^2]$
Exponential loss	$\mathbb{E}_{p^*(x)}\left[\left(-\frac{1-\mathcal{D}}{\mathcal{D}}\right)^{\frac{1}{2}}\right] + \mathbb{E}_{p(x)}\left[\left(-\frac{\mathcal{D}}{1-\mathcal{D}}\right)^{\frac{1}{2}}\right]$
Misclassification	$\mathbb{E}_{p^*(x)}[-\mathbb{I}[\mathcal{D} \leq 0.5]] + \mathbb{E}_{p(x)}[-\mathbb{I}[\mathcal{D} > 0.5]]$
Hinge loss	$\mathbb{E}_{p^*(x)}\left[-\max\left(0, 1 - \log\frac{\mathcal{D}}{1-\mathcal{D}}\right)\right] + \mathbb{E}_{p(x)}\left[-\max\left(0, 1 + \log\frac{\mathcal{D}}{1-\mathcal{D}}\right)\right]$
Spherical	$\mathbb{E}_{p^*(x)}[\alpha\mathcal{D}] + \mathbb{E}_{p(x)}[\alpha(1-\mathcal{D})]; \quad \alpha = (1 - 2\mathcal{D} + 2\mathcal{D}^2)^{-\frac{1}{2}}$

Proper scoring rules

Proper scoring rules are loss functions used for binary classification problems, which ensure that an optimal classifier can be used to learn the density ratio between the two distributions.

$$\frac{p^*(x)}{p(x)} = \frac{\mathcal{D}(x)}{1 - \mathcal{D}(x)}$$

Proper scoring rule and f-divergence connection

For each f-divergence, there is a corresponding scoring rule which when maximised a bound on an f-divergence is obtained.

This fundamentally connects f-divergences and binary classification.

Connection to density ratios

Want to learn more?

Mohamed, et al **Learning in implicit generative models** arxiv (2016)

$$\frac{p^*(x)}{p(x)} = \frac{\mathcal{D}(x)}{1 - \mathcal{D}(x)}$$

Useful trick: density ratios can be estimated *only from samples* using a binary classifier.

So far… evaluating *f*-divergences

$$D_f(p^*||p) = \int p(x) f\left(\frac{p^*(x)}{p(x)}\right) dx$$

$$\sup_{t \in \mathcal{T}} \mathbb{E}_{p^*(x)} t(x) - \mathbb{E}_{p(x)} f^\dagger(t(x)) dx$$

learning a *discriminator* to distinguish between samples from two distributions

Back to learning generative models

We want to find a model distribution *p* which minimises: $D_f(p^*||p)$

Back to learning generative models

We want to find a model distribution *p* which minimises: $D_f(p^*||p)$

We replace the intractable divergence with the bound.

From *f*-divergences to *f*-GAN

Want to learn more?

Nowozin, et al **f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization.** Neural Information Processing Systems **(2016)**

evaluation

$$D_f(p^*||p) = \int p(x) f\left(\frac{p^*(x)}{p(x)}\right) dx$$

↓

evaluation

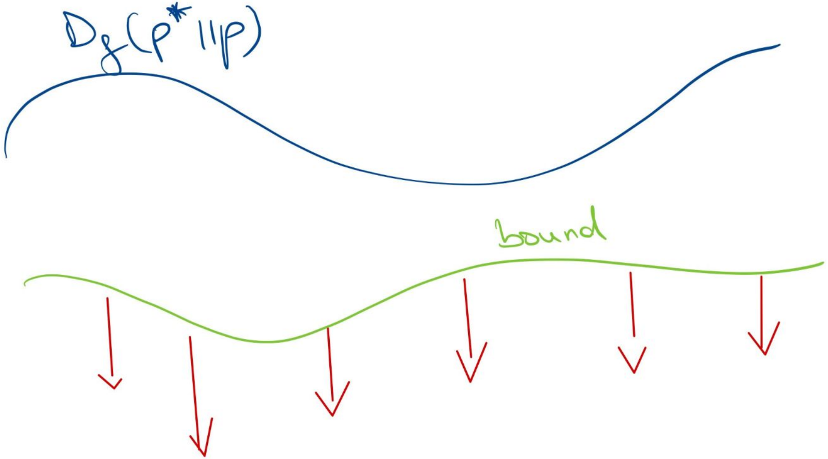
$$\sup_{t \in \mathcal{T}} \mathbb{E}_{p^*(x)} t(x) - \mathbb{E}_{p(x)} f^\dagger(t(x)) dx$$

↓

learning

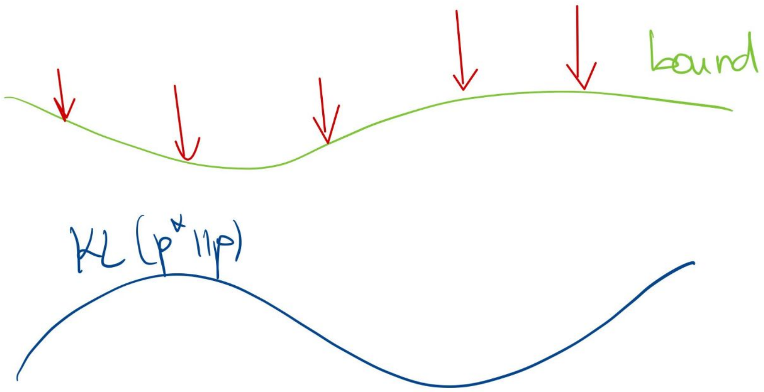
$$\min_G \max_D \mathbb{E}_{p^*(x)} D(x) - \mathbb{E}_{p(z)} f^\dagger(D(G(z)))$$

Challenge: minising a lower bound

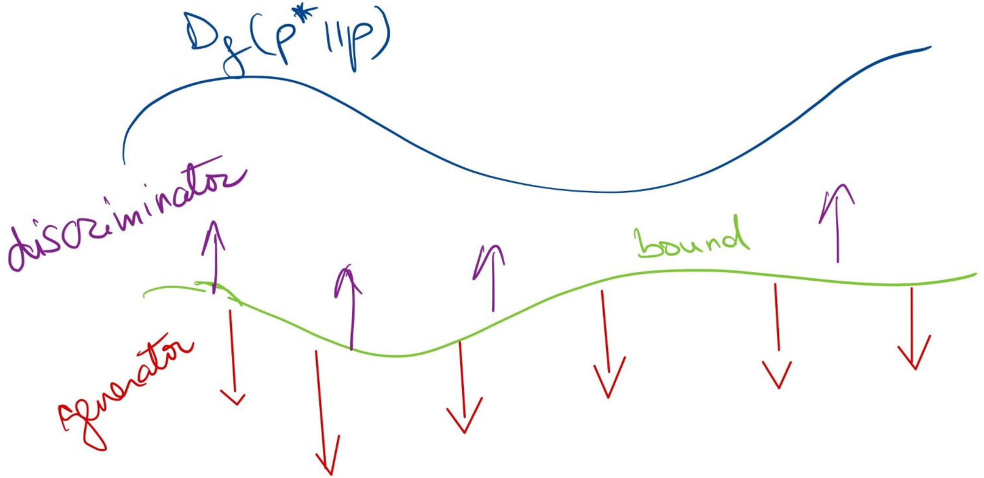


Contrast with VAEs

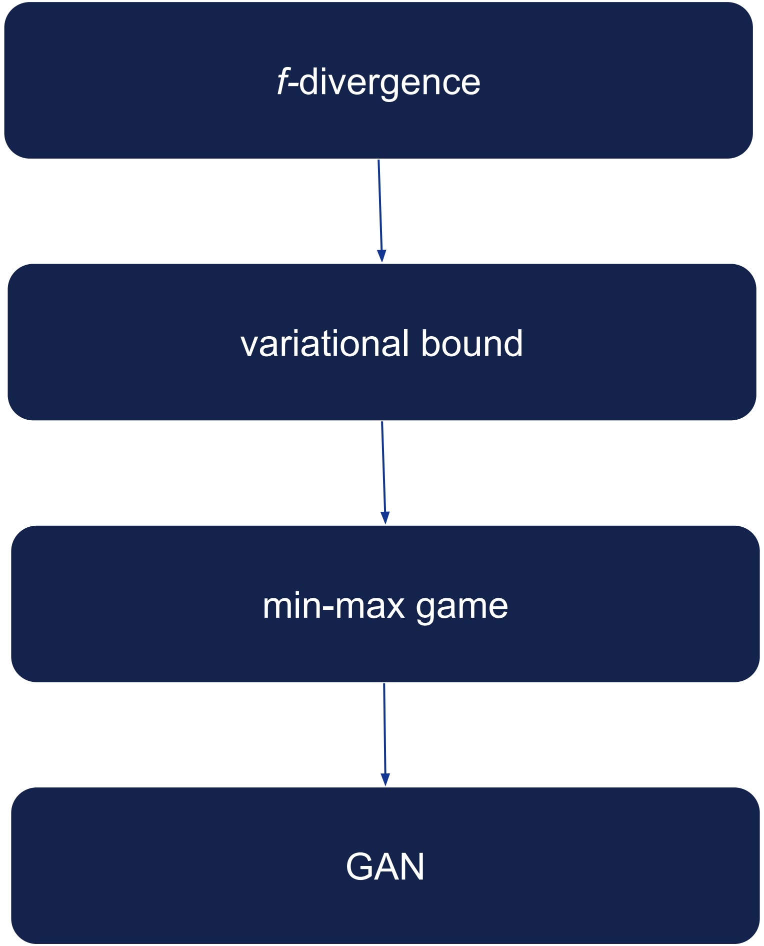
$$KL\left[p^*||p\right] = C - \mathbb{E}_{p*(\mathbf{x})} \log p(\mathbf{x}) =$$
$$\leq C - \mathbb{E}_{p*(\mathbf{x})} \left[\mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}) - KL[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]\right]$$



Still works well in practice!



Recipe so far



From Integral Probability Metrics

Integral probability metrics are distances, not divergences

Divergence

$$D(p^*, p) \geq 0$$

$$D(p^* || p) = 0 \implies p = p^*$$

Distance

$$D(p^*, p) \geq 0$$

$$D(p^* || p) = 0 \implies p = p^*$$

$$D(p^*, p) = D(p, p^*)$$

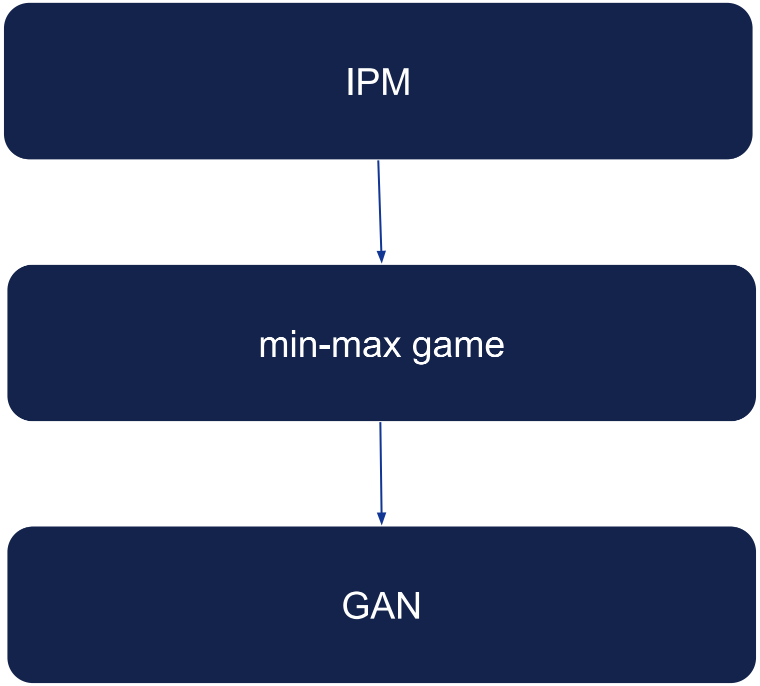
$$D(p^*, p) \leq D(p, q) + D(p^*, q)$$

Integral Probability Metrics

$$D(p^*, p) = \sup_{f \in \mathcal{F}} \left| \mathbb{E}_{p^*(x)} f(x) - \mathbb{E}_{p(x)} f(x) \right|$$

Different IPM instatiations given by different family of functions.

Integral Probability Metrics



Wasserstein Distance

Want to learn more?

Gretton, et al **Interpretable comparison of distributions and models** Neural Information Processing Systems Tutorial **(2019)**

$$W(p^*, p) = \sup_{||f||_L \leq 1} \mathbb{E}_{p^*(x)} f(x) - \mathbb{E}_{p(x)} f(x)$$

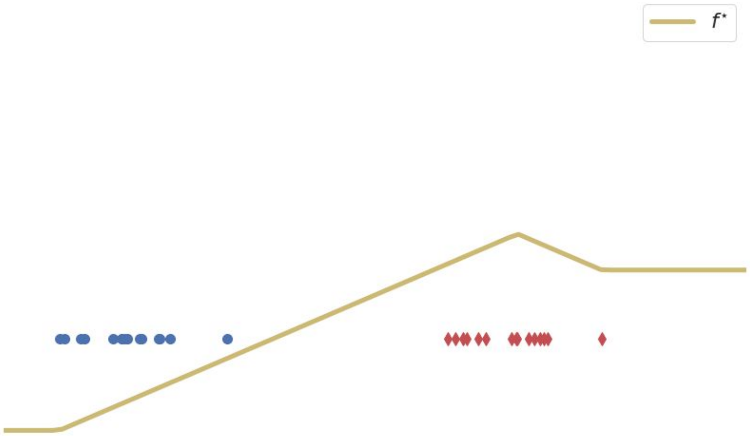
$$|f(x) - f(y)| \leq |x - y|$$

Wasserstein Distance

Want to learn more?

Gretton, et al **Interpretable comparison of distributions and models** Neural Information Processing Systems Tutorial **(2019)**

$$W(p^*, p) = \sup_{||f||_L \leq 1} \mathbb{E}_{p^*(x)} f(x) - \mathbb{E}_{p(x)} f(x)$$



Wasserstein Distance

Want to learn more?
Gretton, et al **Interpretable comparison of distributions and models** Neural Information Processing Systems Tutorial **(2019)**

Estimation

$$W(p^*, p) = \sup_{||f||_L \leq 1} \mathbb{E}_{p^*(x)} f(x) - \mathbb{E}_{p(x)} f(x)$$

Learning

$$\min_G W(p, p^*) = \min_G \sup_{||f||_L \leq 1} \mathbb{E}_{p^*(x)} f(x) - \mathbb{E}_{p(z)} f(G(z))$$

Wasserstein Distance

Want to learn more?

Arjovsky, et al **Wasserstein GAN.** International Conference on Machine Learning (2017)

Wasserstein Distance

$$\mathrm{W}(p, p^*) = \sup_{||f||_L \leq 1} \mathbb{E}_{p(x)} f(x) - \mathbb{E}_{p^*(x)} f(x)$$

Wasserstein GAN

$$\min_{G} \max_{||D||_L \leq 1} \mathbb{E}_{p^*(x)} D(x) - \mathbb{E}_{p(z)} D(G(z))$$

Try to make D is 1-Lipschitz via gradient penalties, spectral normalization, weight clipping.

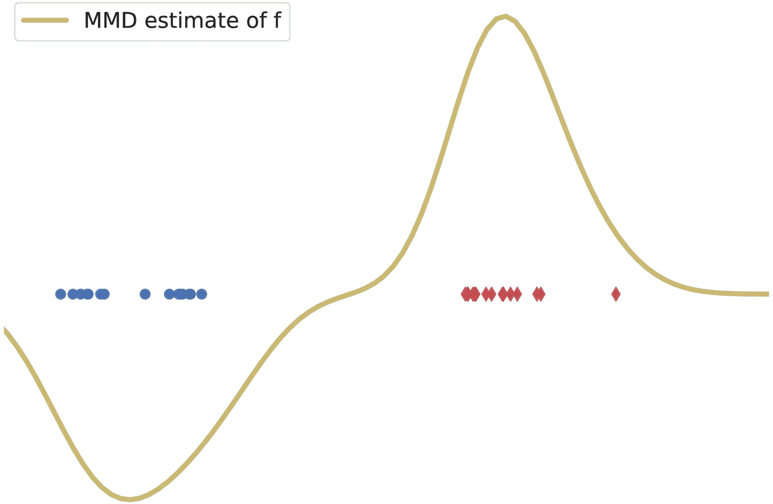
MMD

Want to learn more?

Gretton, et al **Interpretable comparison of distributions and models** Neural Information Processing Systems Tutorial **(2019)**

$$\text{MMD}(p^*, p) = \sup_{||f||_{\mathcal{H}} \leq 1} \mathbb{E}_{p^*(x)} f(x) - \mathbb{E}_{p(x)} f(x)$$

$\mathcal{H}$ *is a RKHS.*



MMD

Want to learn more?
Li, et al **MMD GAN: Towards Deeper Understanding of Moment Matching Network.** Neural Information Processing Systems (2017)

MMD

$$\mathrm{MMD}(p^*, p) = \sup_{||f||_{\mathcal{H}} \leq 1} \mathbb{E}_{p^*(x)} f(x) - \mathbb{E}_{p(x)} f(x)$$

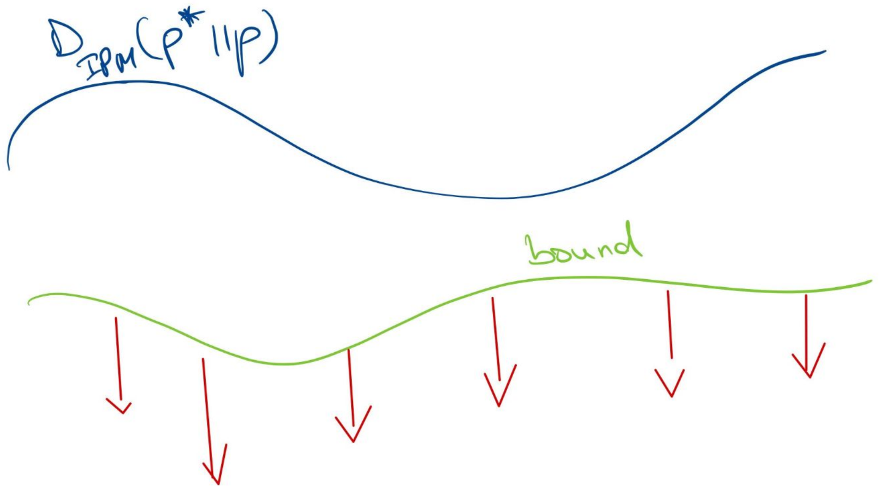
$\mathcal{H}$ *is a RKHS.*

MMD-GAN

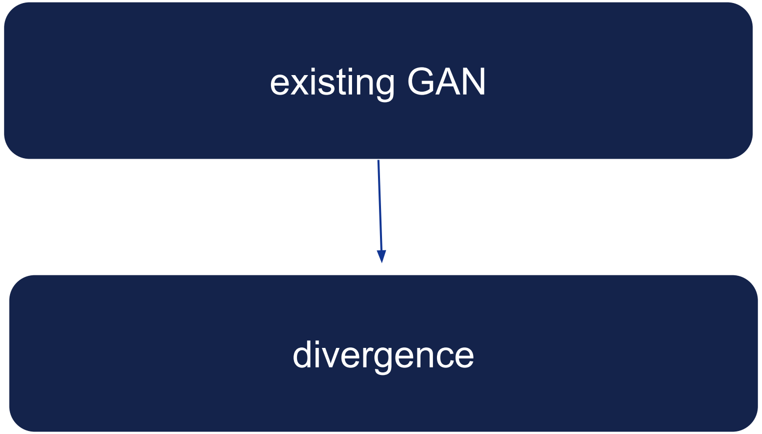
$$\min_{G} \max_{||D||_{\mathcal{H}} \leq 1} \mathbb{E}_{p^*(x)} D(x) - \mathbb{E}_{p(z)} D(G(z))$$

Choose kernel with learned features (via D) $K_\phi(x, x') = K(\phi(x), \phi(x'))$

Still minising a lower bound



Finding the divergence from the GAN


existing GAN
divergence

On Relativistic *f*-Divergences

Want to learn more?

Jolicoeur-Martineau, et al **On Relativistic f-Divergences**

ICML (**2020**)

Relativistic GAN: intuitive introduction of a new approach to train GANs.

$$\max_D \mathbb{E}_{x \sim p^*, y \sim p} f\left(D(x) - D(y)\right)$$

$$\max_G \mathbb{E}_{x \sim p^*, z \sim p_z} f\left(D(G(z)) - D(x)\right)$$

On Relativistic *f*-Divergences

Want to learn more?

Jolicoeur-Martineau, et al **On Relativistic f-Divergences**

ICML (2020)

Relativistic GAN: intuitive introduction of a new approach to train GANs.

$$\max_D \mathbb{E}_{x \sim p^*, y \sim p} f\left(D(x) - D(y)\right)$$

$$\max_G \mathbb{E}_{x \sim p^*, z \sim p_z} f\left(D(G(z)) - D(x)\right)$$

On relativistic f-divergences: proves that the above objective corresponds to a divergence (and thus obtains all theoretical guarantees that come from that).

$$D_f^{rel} = \sup_D 2\mathbb{E}_{x \sim p^*, y \sim p} f\left(D(x)) - D(y)\right)$$

Non-saturating GAN training as divergence minimization

Want to learn more?

Shannon, et al **Non-saturating GAN training as divergence minimization** arxiv (2020)

Non saturating GANs have been introduced by the original GAN paper and mainly used in practice because of better performance in practice.

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

$$\min_G \mathbb{E}_{z \sim p_z} - \log D(G(z))$$

Recently, it has been shown that the non-saturating GAN corresponds to another f-divergence.

You can create GAN training criteria inspired by multiple divergences & distances.

Why train a GAN instead of doing divergence minimization?

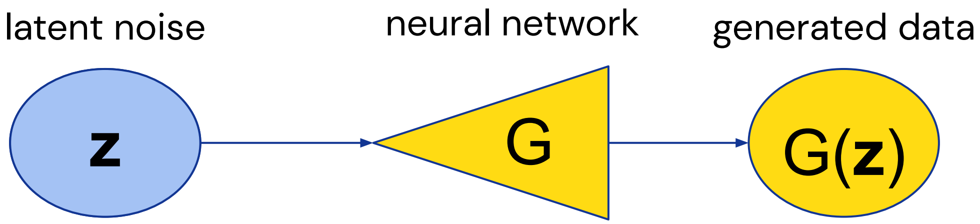
- Model type
- Computational Intractability
- Smooth learning signal
- Learned “divergence“

Implicit models and KL divergence

Model type

$$\mathrm{KL}(p^*(\mathbf{x})||p(\mathbf{x})) = \int p^*(\mathbf{x}) \log \frac{p^*(\mathbf{x})}{p(\mathbf{x})} dx$$

For implicit models, we do not have access to the explicit distribution p(x).

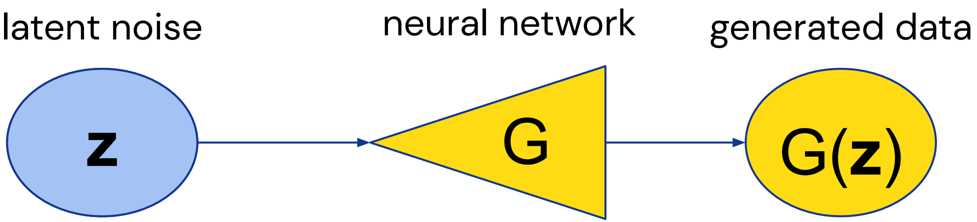


Implicit models and KL divergence

Model type

$$\mathrm{KL}(p^*(\mathbf{x})||p(\mathbf{x})) = \int p^*(\mathbf{x}) \log \frac{p^*(\mathbf{x})}{p(\mathbf{x})} dx$$

For implicit models, we do not have access to the explicit distribution p(x).



f-GAN

$$\min_G \max_D \mathbb{E}_{p^*(x)} D(x) - \mathbb{E}_{p(z)} f^\dagger(D(G(z)))$$

Wasserstein distance & computational intractability

Computational intractability

$$W(p, p^*) = \sup_{||f||_L \leq 1} \mathbb{E}_{p(x)} f(x) - \mathbb{E}_{p^*(x)} f(x)$$

Computationally intractable for complex cases.

Wasserstein GAN

$$\min_G \max_{||D||_L \leq 1} \mathbb{E}_{p^*(x)} D(x) - \mathbb{E}_{p(z)} D(G(z))$$

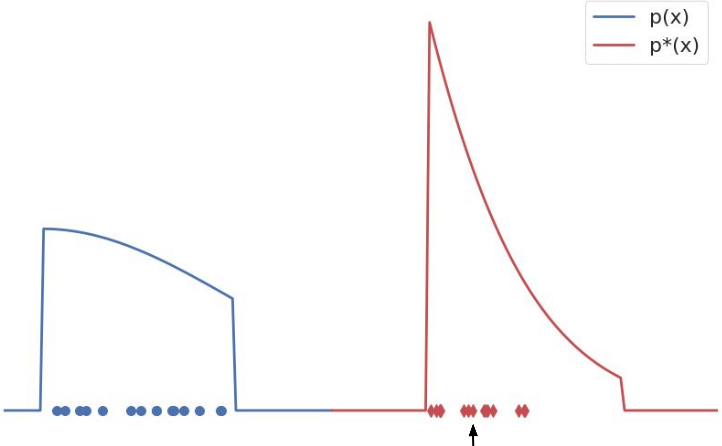
Smooth learning signal

Want to learn more
Gretton, et al **Interpretable comparison of distributions and models** Neural Information Processing Systems Tutorial **(2019)**

No learning signal from KL/JSD divergence if non-overlapping support between the data and the model.

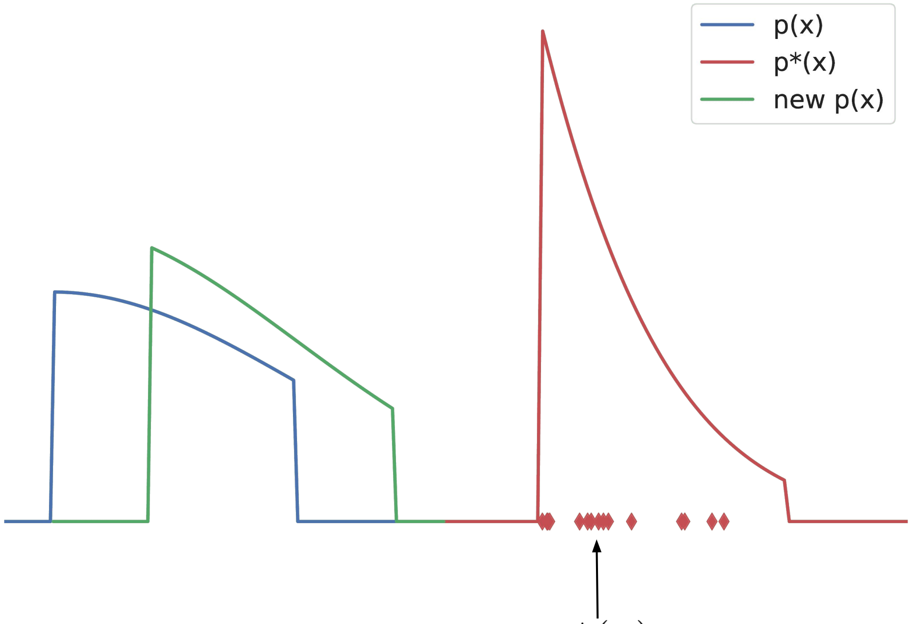
$$\mathrm{KL}(p^*(\mathbf{x})||p(\mathbf{x})) = \infty$$

$$\mathrm{JSD}(p^*(\mathbf{x})||p(\mathbf{x})) = \log 2$$



$$\frac{p^*(\mathbf{x})}{p(\mathbf{x})} = \infty$$

Smooth learning signal



$$\frac{p^*(\mathbf{x})}{p(\mathbf{x})} = \infty$$

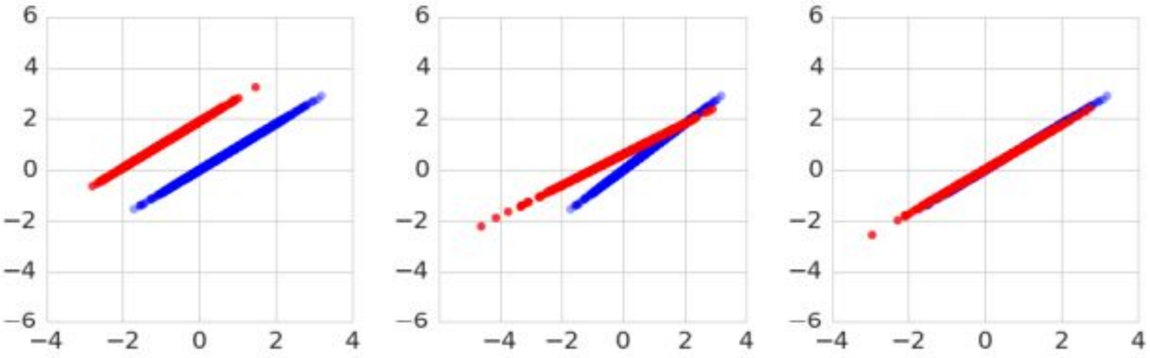
The density ratio jumps to infinity at the data distribution.

Smooth learning signal

Want to learn more?

Fedus, et al **Many paths to equilibrium.**
International Conference for learning representations **(2018)**

But GANs still learn!



(a) Step 0 (b) Step 5000 (c) Step 12500

Red = data
Blue = model (changes in training)

Want to learn more?

Gretton, et al **Interpretable comparison of distributions and models** Neural Information Processing Systems Tutorial **(2019)**

true ratio

$$KL[p^*(x)||p(x)] = \int p^*(x) \log\left(\frac{p^*(x)}{p(x)}\right) dx \geq \sup_{D \in \mathcal{F}} \left(\mathbb{E}_{p^*(x)} D(x) - \mathbb{E}_{p(x)} e^{D(x)}\right)$$

ratio approximation used in GAN training

Want to learn more?

Gretton, et al **Interpretable comparison of distributions and models** Neural Information Processing Systems Tutorial (**2019**)

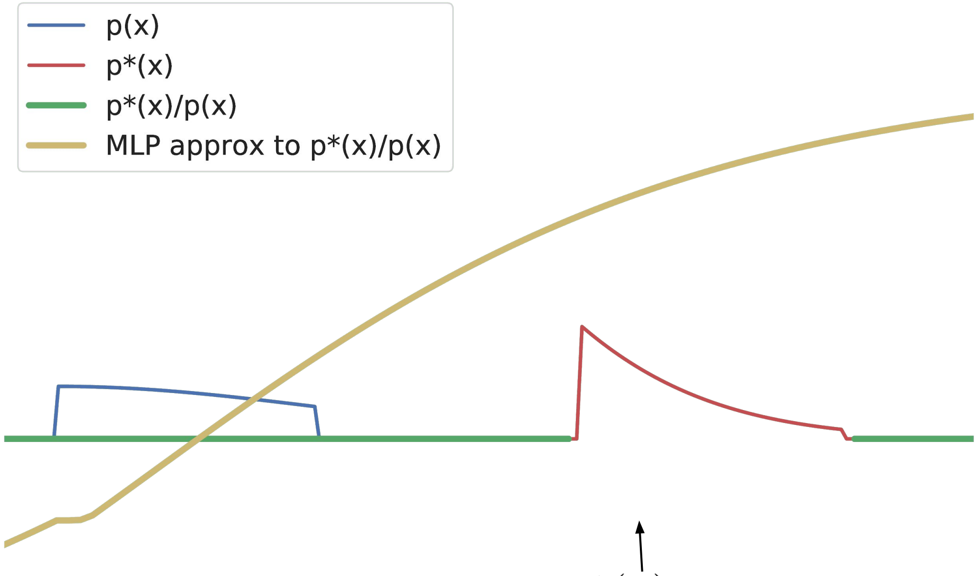
true ratio

$$KL[p^*(x)||p(x)] = \int p^*(x) \log\left(\frac{p^*(x)}{p(x)}\right) dx \geq \sup_{D \in \mathcal{F}} \left(\mathbb{E}_{p^*(x)} D(x) - \mathbb{E}_{p(x)} e^{D(x)}\right)$$

ratio approximation used in GAN training

$\mathcal{F}$ is the family of functions used to approximate the ratio (deep neural networks, RKHS).

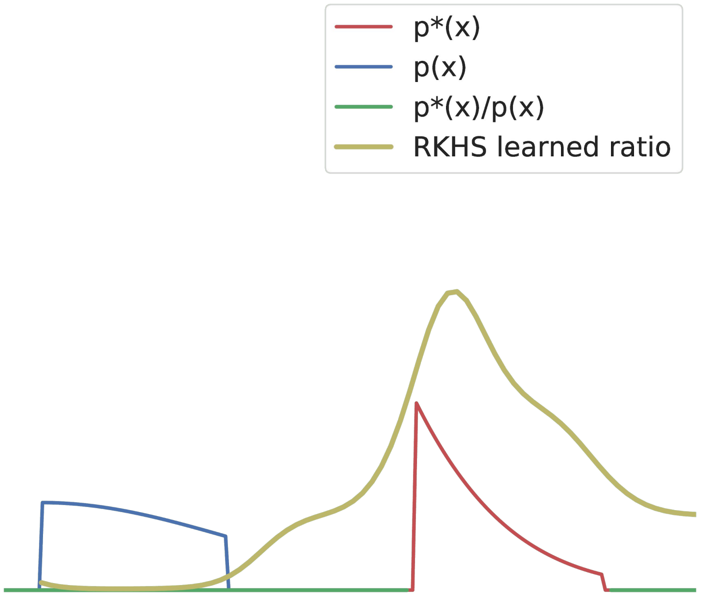
Smooth learning signal



$$\frac{p^*(\mathbf{x})}{p(\mathbf{x})} = \infty$$

Smooth approximation of the density ratio does not go to infinity.

Smooth learning signal



$$\frac{p^*(\mathbf{x})}{p(\mathbf{x})} = \infty$$

Smooth approximation of the density ratio does not go to infinity.

D is smooth approximation to the decision boundary of the underlying divergence:

- GANs do not do divergence minimization in practice
- GANs do not fail in cases where the underlying divergence would

Discriminators as learned "distances"

Want to learn more?

Arora, et al **Generalization and Equilibrium in Generative Adversarial Nets.** International Conference for machine learning (2017)

We can think of D (the teacher) as learning a "distance" between the data and model distribution that can provide useful gradients to the model.

Discriminators as “learned” distances

$$\min_G \boxed{\max_D V(D, G)}$$

D provides a learned distance between the data and sample distributions, using **learned neural network features.**

GANs (learned "distance") or divergence minimization?

GANs

- good samples
- learned loss function

- hard to analyze dynamics (game theory)
- (in practice) no optimal convergence guarantees

Divergence minimization

- optimal convergence guarantees
- easy to analyze loss properties

- harder to get good samples
- loss functions don't correlate with human evaluation

In practice, GANs do not do divergence minimization.
The discriminator can be seen as a learned "distance".

Which GAN should I use?

Empirically, it has been observed that the underlying loss matters less than neural architectures, training regime, data.

This talk focused on obtaining GAN losses from distributional distances and divergences. There are other ways to change GAN losses, through regularisation or other approaches, including:

- Gradient penalties wrt to inputs
  - *Improved training for Wasserstein GAN*, Gulrajani et al, Neurips, 2017
  - *Which methods of GANs actually converge?* Mescheder et al, ICML 2018
- Gradient regularization wrt to parameters
  - *The numerics of GANs*, Mescheder et al, Neurips, 2017
  - *The Mechanics of n-Player Differentiable Games,* Balduzzi et al, ICML 2018
- Entropy regularization
  - *Prescribed Generative Adversarial Networks*, Dieng et al, 2019
- and many others...

Architectures and model regularisation are a core ingredient of GAN training:

- Self attention
  - Self-Attention Generative Adversarial Networks, Zhang et al, ICML 2019
- Discriminator regularisation
  - *Spectral Normalization for Generative Adversarial Networks*, Miyato et al, ICLR 2018
- BatchNormalisation is often used for the generator.

Evaluating GANs:

- Inception Score
  - *Improved Techniques for Training GANs*, Salimans et al, Neurips 2016
- Frechet Inception Distance
  - *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium,* Heusel et al, Neurips 2017
- Kernel Inception Distance
  - *Demystifying MMD GANs*, Binkowski et al, ICLR 2018
- Precision and recall metrics
  - Improved Precision and Recall Metric for Assessing Generative Models, s Kynkäänniemi et al, Neurips 2019
- Training classifiers with data generated from GANs
  - *Classification Accuracy Score for Conditional Generative Models*, Ravuri et al, Neurips 2019

And much more…

You can find more related work at conectedpapers.com

Thank you