Generative Adversarial Networks

From dawn to Cramér GANs

Matteo Barbetti

03.02.2020

Generative Adversarial Nets
Introduction

GANs[1] are a powerful class of generative models based on simultaneous training of two neural networks:

- **Generator network** (G) that produces synthetic data given some noise source;
- **Discriminator network** (D) that distinguishes generator's output from true data[2].

We want that D to optimally discriminate on the origin of the two samples. Simultaneously the training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a **minimax two-player game**[1].

[1] I.J. Goodfellow *et al*.. "Generative Adversarial Nets". arXiv:1406.2661.
[2] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin & A. Courville. "Improved Training of Wasserstein GANs". arXiv:1704.00028.

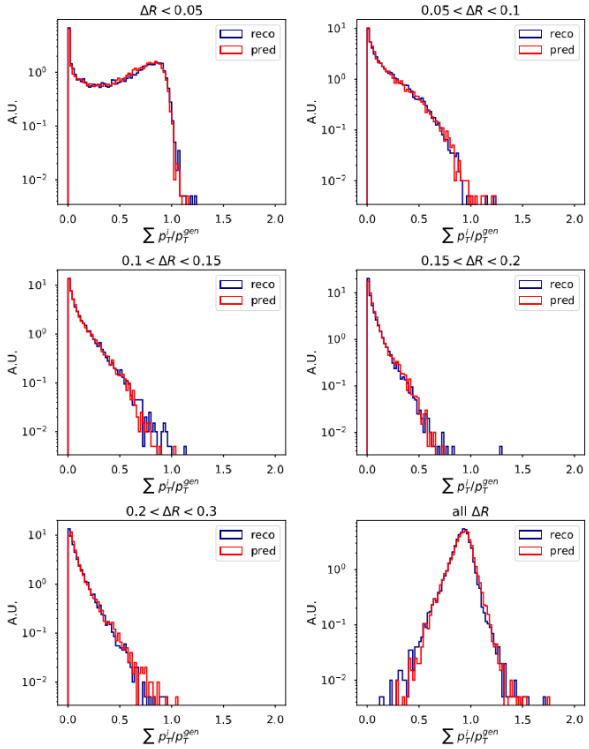
Generative Adversarial Nets
Application in Computer Vision

GANs are widely used as **generative image model** thanks to its capacity in reproducing highly faithful and diverse images with models learned directly from data[3].

[3] A. Brock, J. Donahue & K. Simonyan. "Large Scale GAN Training for High Fidelity Natural Image Synthesis". arXiv:1809.11096.

Generative Adversarial Nets

Application in Physics

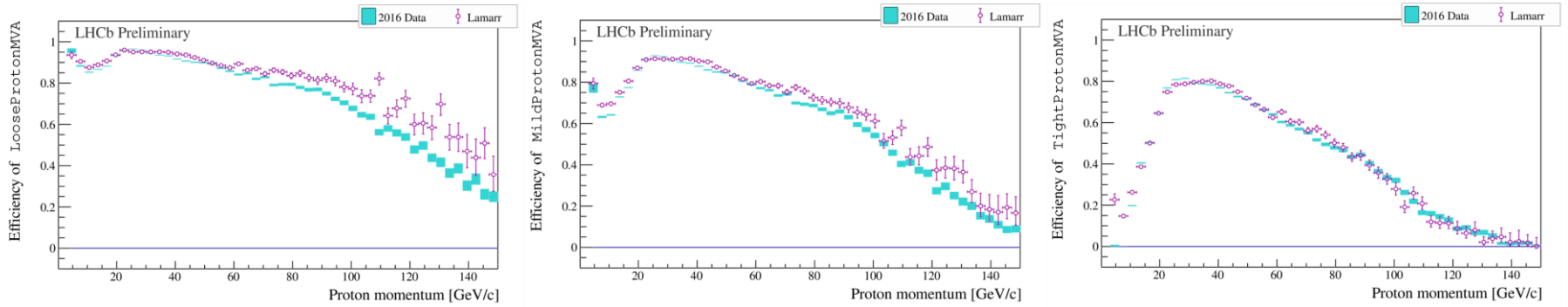


The **extreme scalability** of deep learning based models makes them perfect for application in Physics.

GANs natural propensity for image generation makes you immediately think of calorimeter response[4] or hadronic jet reconstruction[5], but there is no shortage of application in **other science areas**, such as Astrophysics[6, 7], Condensed Matter Physics[8] or Oncology[9].

[4] M. Paganini, L. de Oliveira & B. Nachman. "CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks". arXiv:1712.10321.
[5] P. Musella & F. Pandolfi. "Fast and accurate simulation of particle detectors using generative adversarial networks". arXiv:1805.00850.
[6] K. Schawinski, Ce Zhang, H. Zhang, L. Fowler & G.K. Santhanam. "Generative Adversarial Networks recover features in astrophysical images of galaxies beyond the deconvolution limit". arXiv:1702.00403.
[7] M. Erdmann, L. Geiger, J. Glombitza & D. Schmidt. "Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks". arXiv:1802.03325.
[8] L. Mosser, O. Dubrule & M.J. Blunt. "Reconstruction of three-dimensional porous media using generative adversarial neural networks". arXiv:1704.03225.
[9] A. Kadurin, A. Aliper, A. Kazennov, P. Mamoshina, Q. Vanhaelen, K. Khrabrov & A. Zhavoronkov. "The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology". Oncotarget.14073.

Generative Adversarial Nets
Application to particle identification



Going back to Particle Physics application, we are currently working on the development of an **ultra-fast simulation** for PID system in LHCb using GANs to generate high-level reconstructed observables[10, 11].

Images above show the performance of the **Lamarr Prototype**, an ultra-fast simulation option developed for the LHCb Experiment[12].

[10] G. Sassoli & L. Anderlini. "Generative Adversarial Networks for Fast Simulation of MuonID". Machine Learning @ INFN Firenze.
[11] A. Maevskiy, D. Derkach, N. Kazeev, A. Ustyuzhanin, M. Artemev & L. Anderlini. "Fast Data-Driven Simulation of Cherenkov Detectors Using Generative Adversarial Networks". arXiv:1905.11825.
[12] LHCb Collaboration. "Performance of the Lamarr Prototype: the ultra-fast simulation option integrated in the LHCb simulation framework". LHCB-FIGURE-2019-017.

Generative Adversarial Nets
Minimax two-player game

Defining the function V(D,G) as follows

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

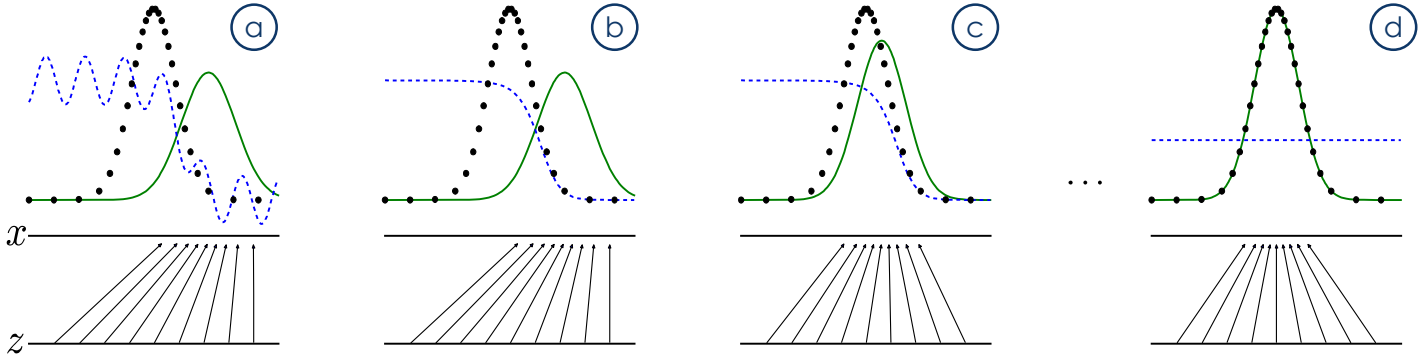
the **minimax game** can be written in this form:

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

A unique solution exists, with G recovering the training data distribution and D equal to ½ everywhere[1].

[1] I. J. Goodfellow *et al*.. "Generative Adversarial Nets". arXiv:1406.2661.

Generative Adversarial Nets
Pedagogical explanation



a) Minimax game near convergence: P_g is similar to P_r and D is a partially accurate classifier.
b) D is trained to discriminate samples from data, converging to optimality.
c) After an update of G, gradient of D has guided G(z) to flow to region that are more likely to be classified as data.
d) After several steps of training, they will reach a point at witch both cannot improve because the discriminator is unable to differentiate between the two distributions[1].

[1] I. J. Goodfellow *et al*.. "Generative Adversarial Nets". arXiv:1406.2661.

Generative Adversarial Nets
Jensen-Shannon divergence

Solving the minimax game corresponds to minimize the **Jensen-Shannon divergence** between the real data distribution P_r and the generator's distribution P_g.

By **varying NN parameters θ**, we can change the map G_θ to data space and make P_g close to the real data distribution. It corresponds to minimize JS divergence that goes to zero for equal distributions.

GANs take a radically different approach compared to other deep generative model not requiring **inference** or **explicit calculation** of the data likelihood[13].

[13] L. Metz, B. Poole, D. Pfau & J. Sohl-Dickstein. "Unrolled Generative Adversarial Networks". arXiv:1611.02163.

Training GANs
Problems

GANs suffer from many issues, particularly during training:

- generator **collapsing** to produce only a single sample or a small family of very similar samples;
- generator and discriminator **oscillating** during training rather than converging to a fixed point;
- if **imbalance** between the two agents occurs, the system doesn't learn[13].

In theory, although minimax game corresponds to minimize JS divergence when the discriminator is optimal, training it till optimality and then doing gradient steps on θ **doesn't work**! In practise, as the discriminator gets better, the updates to the generator gets consistently worse[14].

[13] L. Metz, B. Poole, D. Pfau & J. Sohl-Dickstein. "Unrolled Generative Adversarial Networks". arXiv:1611.02163.
[14] M. Arjovsky & L. Bottou. "Towards Principled Methods for Training Generative Adversarial Networks". arXiv:1701.04862.

Training GANs
Vanishing gradient

Typically, the divergences which GANs minimize are **not continuous** with respect to generator's parameters θ[2]. This allows the existence of the perfect discriminator D* for which the **gradient on the generator vanishes**. If we consider an approximation D that distances ε from D*, we can prove what follows:

$$\lim_{\|D-D^*\|\to 0} \nabla_\theta \mathbb{E}_{z\sim P_z}[\log(1-D(G_\theta(z)))] = 0$$

As our discriminator gets better, the gradient of the generator vanishes. In other words, either our updates to the discriminator will be inaccurate, or they will vanish[14].

[2] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin & A. Courville. "Improved Training of Wasserstein GANs". arXiv:1704.00028.
[14] M. Arjovsky & L. Bottou. "Towards Principled Methods for Training Generative Adversarial Networks". arXiv:1701.04862.

Training GANs
Noise insertion

There is something we can do to break our gradient problem: adding **continuous noise** to both discriminator and generator. This move allows to learn thanks to **non-zero gradient** of the generator. However, it's now proportional to the gradient of *noisy* JS divergence:

$$\mathbb{E}_{z\sim P_z,\varepsilon'}\left[\nabla_\theta \log(1 - D^*_\varepsilon(G_\theta(z) + \varepsilon'))\right] = 2 \cdot \nabla_\theta JS\left(P_{r+\varepsilon} \| P_{g+\varepsilon}\right)$$

This variant of JS divergence measures a similarity between the two **noisy distribution** and isn't an intrinsic measure of P_r and P_g. Luckily, using **Wasserstein metric** we can solve this problem[14].

[14] M. Arjovsky & L. Bottou. "Towards Principled Methods for Training Generative Adversarial Networks". arXiv:1701.04862.

Wasserstein GANs
Earth-Mover distance

The **Earth-Mover distance** induces the Wasserstein metric:

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbb{E}_{(x,y)\sim\gamma} \left[\|x - y\| \right]$$

where with γ(x,y) we indicate every joint distribution whose marginals are respectively P_r and P_g. The EM distance is the *cost* of the **optimal transport plan** from x to y.

For EM distance, we can demonstrate that

- If G_θ is continuous in θ, so is $W(P_r,P_\theta)$;
- If G_θ is locally Lipschitz and continuous, the $W(P_r,P_\theta)$ is **continuous** e.w., and **differentiable** almost e.w.;
- JS and KL divergences don't have these properties[15].

[15] M. Arjovsky, S. Chintala & L. Bottou. "Wasserstein Generative Adversarial Networks". arXiv:1701.07875.

Wasserstein GANs
Wasserstein loss

The **Earth-Mover distance** can be defined also as:

$$W(P_r, P_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim P_r}[f(x)] - \mathbb{E}_{x \sim P_g}[f(x)]$$

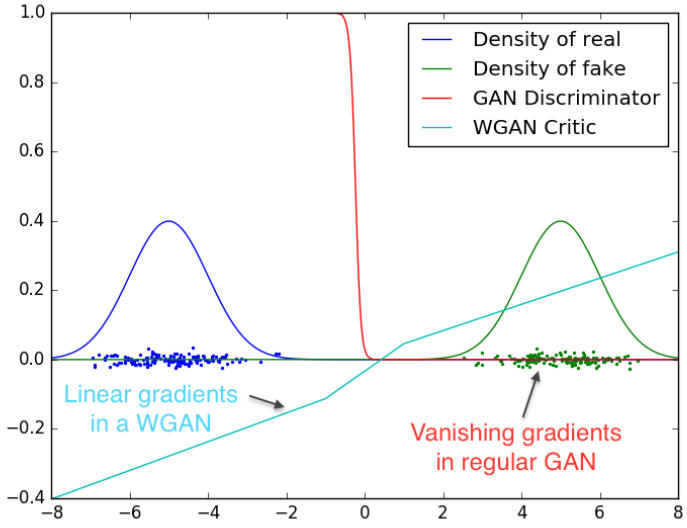
where the supremum is over all the 1-Lipschitz functions f. Considering the **K-Lipschitz family** $\{f_w\}$, then we end up with K-times EM distance.

In WGAN context, the discriminative model corresponds to finding the function f that **maximize** the previous relation. Simultaneously, we want to **minimize** the EM distance with respect to θ for the distributions convergence[15].

[15] M. Arjovsky, S. Chintala & L. Bottou. "Wasserstein Generative Adversarial Networks". arXiv:1701.07875.

Wasserstein GANs
Critic function

Typically, WGAN solves the minimax game with the **critic function** (f_w) that can approximate the problem up to a scaling factor.



The fact that the EM distance is continuous e.w. and differentiable almost e.w. means that we can train the critic **till optimality**.

In the figure, we can see the original GAN discriminator saturates and results in vanishing gradients. The critic, however, **can't saturate** (K-Lipschitz), and converges to a *linear* function[15].

[15] M. Arjovsky, S. Chintala & L. Bottou. "Wasserstein Generative Adversarial Networks". arXiv:1701.07875.

Cramér GANs
Unbiased sample gradients

Most of loss functions used in machine learning are **distances d**, as in the case of Wasserstein metric. A crucial characteristic of this kind of loss is the **unbiased sample gradients** (U) notion owning[16]:

$$\mathbb{E}_{x_m \sim P}\left[\nabla_\theta d\left(\hat{P}_m, Q_\theta\right)\right] = \nabla_\theta d(P, Q_\theta)$$

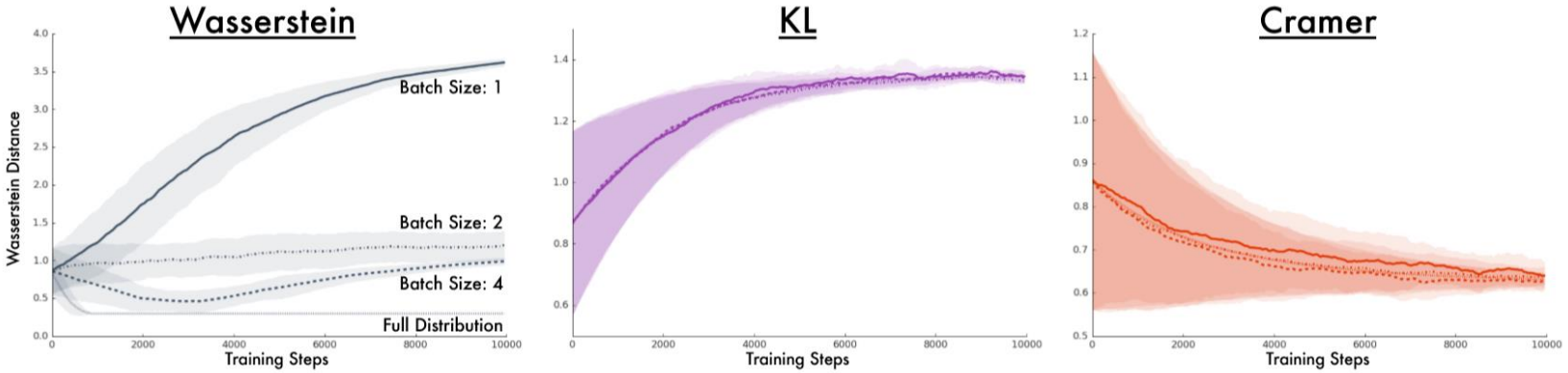
Wasserstein metric is an **ideal divergence**[16], but it doesn't have (U). So, we need a distance that not only has the same appealing properties of Wasserstein metric but also provides us with (U): the **Cramér distance**.

$$l_2(P, Q) = \sup_{f \in \mathbb{F}_2} \left| \mathbb{E}_{x \sim P}[f(x)] - \mathbb{E}_{x \sim Q}[f(x)] \right|$$

[16] M. G. Bellemare, I. Danihelka, W. Dabney, S. Mohamed, B. Lakshminarayanan, S. Hoyer & R. Munos. "The Cramer Distance as a Solution to Biased Wasserstein Gradients". arXiv:1705.10743.

Cramér GANs

Bias in sample gradient estimates



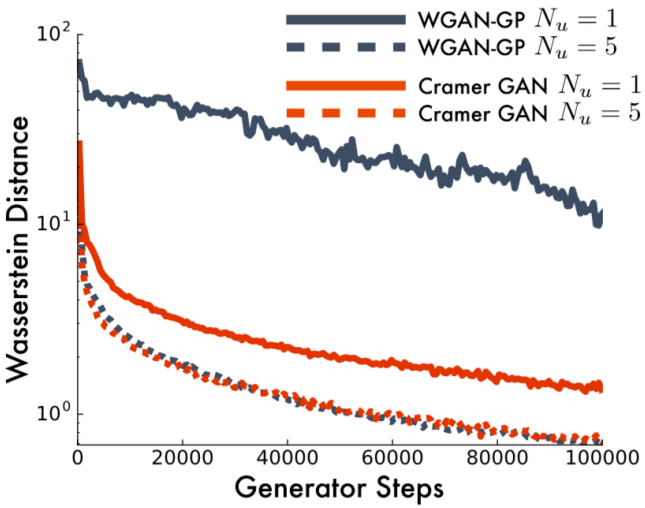
If a divergence doesn't possess (U) then minimizing it with stochastic gradient descent may **not converge**, or it may converge to the **wrong minimum**[16].

Images above show the learning curves of GANs training with (U)-losses (KL and Cramér distances) and with Wasserstein metric. For this one you can see how the **batch size** choice affects the minimum search[16].

[16] M. G. Bellemare, I. Danihelka, W. Dabney, S. Mohamed, B. Lakshminarayanan, S. Hoyer & R. Munos. "The Cramer Distance as a Solution to Biased Wasserstein Gradients". arXiv:1705.10743.

Cramér GANs
Stability and diversity

The **energy distance ε** is a natural extension of the Cramér distance to the multivariate case[16].



Starting from ε, we can define a loss function that reproduces the **minimax two-player game** thanks to an imperfect critic function similar to the Wasserstein one.

The Cramér GAN leads to **more stable learning** and **increased diversity** in the generated samples[16].

[16] M. G. Bellemare, I. Danihelka, W. Dabney, S. Mohamed, B. Lakshminarayanan, S. Hoyer & R. Munos. "The Cramer Distance as a Solution to Biased Wasserstein Gradients". arXiv:1705.10743.

Conclusion

- GANs offer a generative model based on a **minimax game** not requiring inference or likelihood calculation.
- Training GANs is very hard because of **mode collapse** and **instability** caused by disjoint supports.
- **Wasserstein metric** produces a continuous loss function even though disjoint supports.
- WGANs solve the zero-gradient problem substituting discriminator with the **critic function** that can't saturate.
- Wasserstein metric is an **ideal divergence** but it doesn't have **unbiased sample gradients**.
- **Cramér distance** is an ideal divergence with unbiased sample gradients.
- Cramér GANs offer a **stable-training** solution to reproduce **high-dimensional** spaces.

Backup

Generative Adversarial Nets
Mathematical notation

For the **variables** we have:

- $\mathcal{X}$ - *real data space*
- $x \sim P_r$ - *real data density*
- $\mathcal{Z}$ - *latent space*
- $z \sim P_z$ - *latent variable density*
- $G_\theta : \mathcal{Z} \mapsto \mathcal{X}$ - *map to data space*
- $G_\theta(z) \sim P_g$ - *generated data density*

For the **models** we have:

- $D(x)$ - *probability that sample came from data*
- $\min_G \log(1 - D(G(z)))$ - *maximize discriminator mistake*[1]

[1] I. J. Goodfellow *et al*.. "Generative Adversarial Nets". arXiv:1406.2661.

Generative Adversarial Nets
Optimal discriminator

Solving the minimax game with respect to D, we obtain

$$\max_D V(D,G) = V(D^*,G)$$

where D* indicates the **optimal discriminator**:

$$D^*(x) = \frac{p_r(x)}{p_r(x) + p_g(x)}$$

It's easily to demonstrate that V(D*,G) is related to the **Jensen-Shannon divergence**:

$$\boxed{V(D^*,G) = -\log 4 + 2 \cdot JS(P_r \| P_g)}$$

Generative Adversarial Nets
Proof optimal discriminator

Recalling the definition of V(D,G)

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

we have

$$V(D,G) = \int_x p_r(x)\log(D(x))\,dx + \int_z p_z(z)\log(1-D(G(z)))\,dz$$

$$= \int_x [p_r(x)\log(D(x)) + p_g(x)\log(1-D(x))]\,dx = \int_x v(D,G)\,dx$$

Obviously it follows that $\max_D V(D,G) = \max_D v(D,G)$.

It's easy to see that it occurs for D*:

$$\boxed{D^*(x) = \frac{p_r(x)}{p_r(x) + p_g(x)}}$$

Generative Adversarial Nets
Proof Jensen-Shannon divergence

Subsisting D^* into the definition of V(D,G) we obtain

$$\begin{aligned} V(D^*, G) &= \mathbb{E}_{x\sim P_r}[\log D^*(x)] + \mathbb{E}_{z\sim P_z}[\log(1 - D^*(G(z)))] \\ &= \mathbb{E}_{x\sim P_r}[\log D^*(x)] + \mathbb{E}_{x\sim P_g}[\log(1 - D^*(x))] \\ &= \mathbb{E}_{x\sim P_r}\left[\log \frac{p_r(x)}{p_r(x) + p_g(x)}\right] + \mathbb{E}_{z\sim P_z}\left[\log \frac{p_g(x)}{p_r(x) + p_g(x)}\right] \\ &= -2\log 2 + KL(P_r \| P_A) + KL(P_g \| P_A) \end{aligned}$$

where P_A is a sort of *average* distribution:

$$P_A = \frac{P_r + P_g}{2}$$

Recalling the definition of Jensen-Shannon divergence

$$\boxed{V(D^*, G) = -\log 4 + 2 \cdot JS(P_r \| P_g)}$$

Training GANs
Perfect discriminator

Empirically, if we train D till convergence, the JS divergence between P_r and P_g is **maxed out**. The only way this can happen is if the supports of distributions are **disjoint** or lie in **low dimensional** manifolds. In these hypothesis we can demonstrate that a perfect discriminator always exists.

PERFECT DISCRIMINATOR

$$D : \mathcal{X} \rightarrow [0, 1]$$
$$P_r[D(x) = 1] = 1$$
$$P_g[D(x) = 0] = 1$$

A perfect discriminator has **zero gradient** almost everywhere on the union of sets containing P_r and P_g supports[14].

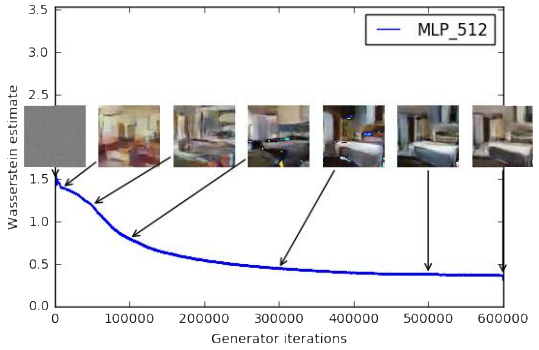
[14] M. Arjovsky & L. Bottou. "Towards Principled Methods for Training Generative Adversarial Networks". arXiv:1701.04862.

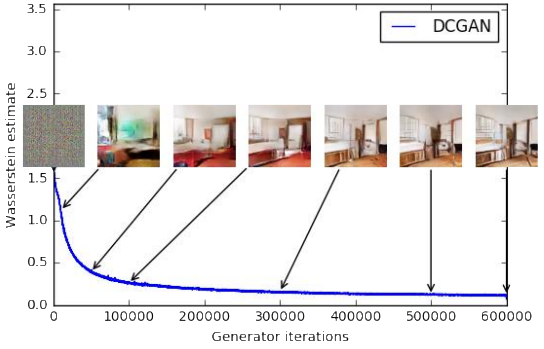
Wasserstein GANs
Meaningful loss metric

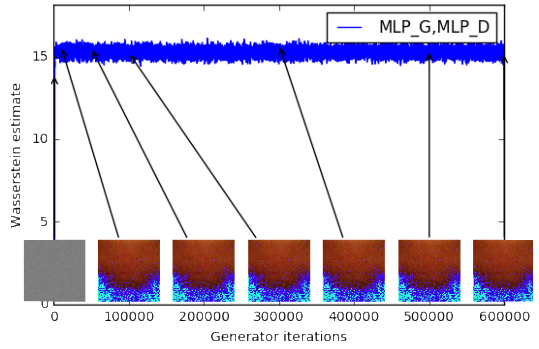
The figures represents the first example, in GAN literature, where the loss of the GAN shows properties of **convergence** in training curves.

Top-down figures:

- The generator is a **MLP** with 4 hidden layers and 512 units at each layer. The loss decreases consistently as training progresses and sample quality increases.
- The generator is a standard **DCGAN**. The loss decreases quickly and sample quality increases as well.
- Both the generator and the discriminator are **MLPs** with high learning rates (training failed). The loss is constant and samples are constant as well[15].

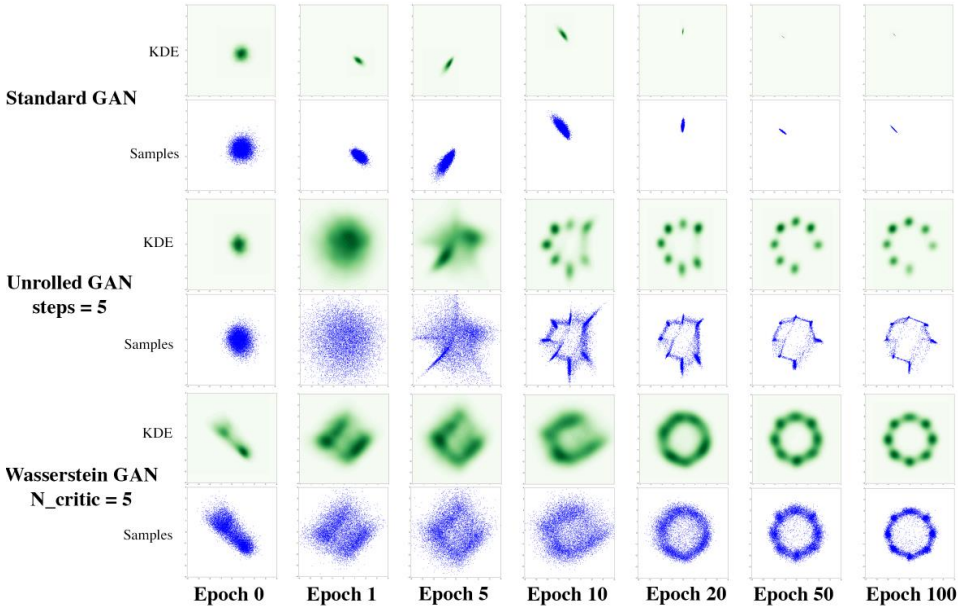






[15] M. Arjovsky, S. Chintala & L. Bottou. "Wasserstein Generative Adversarial Networks". arXiv:1701.07875.

Wasserstein GANs
Experiment



Consider a 2D mixture of 8 Gaussians arranged in a circle. Looking to WGAN output, we can note how it tends to learn to match **low-dimensional** structure of the data, before zooming in on specific bumps of the true density[15].

[15] M. Arjovsky, S. Chintala & L. Bottou. "Wasserstein Generative Adversarial Networks". arXiv:1701.07875.

Cramér GANs
Ideal divergence

Consider a divergence d, and for two random variables (X,Y) with distribution (P,Q) write d(P,Q) = d(X,Y). So, we can say that d is an **ideal divergence**[16] if

1) d is **scale sensitive**: $$d(cX, cY) \leq |c|^{\beta} d(X, Y)$$

2) d is **sum invariant**: $$d(A + X, A + Y) \leq d(X, Y)$$

As we have seen, another useful property for loss function is the **unbiased sample gradients** [16] :

$$\mathbb{E}_{x_m \sim P}\left[\nabla_\theta d\left(\hat{P}_m, Q_\theta\right)\right] = \nabla_\theta d(P, Q_\theta)$$

where with F_2 we indicate every function absolutely continuous with gradient norm less than one.

[16] M. G. Bellemare, I. Danihelka, W. Dabney, S. Mohamed, B. Lakshminarayanan, S. Hoyer & R. Munos. "The Cramer Distance as a Solution to Biased Wasserstein Gradients". arXiv:1705.10743.