

---

# Wasserstein Generative Adversarial Networks

---

Martin Arjovsky<sup>1</sup> Soumith Chintala<sup>2</sup> Léon Bottou<sup>1,2</sup>

## Abstract

We introduce a new algorithm named WGAN, an alternative to traditional GAN training. In this new model, we show that we can improve the stability of learning, get rid of problems like mode collapse, and provide meaningful learning curves useful for debugging and hyperparameter searches. Furthermore, we show that the corresponding optimization problem is sound, and provide extensive theoretical work highlighting the deep connections to different distances between distributions.

## 1. Introduction

The problem this paper is concerned with is that of unsupervised learning. Mainly, what does it mean to learn a probability distribution? The classical answer to this is to learn a probability density. This is often done by defining a parametric family of densities  $(P_\theta)_{\theta \in \mathbb{R}^d}$  and finding the one that maximized the likelihood on our data: if we have real data examples  $\{x^{(i)}\}_{i=1}^m$ , we would solve the problem

$$\max_{\theta \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^m \log P_\theta(x^{(i)})$$

If the real data distribution  $\mathbb{P}_r$  admits a density and  $\mathbb{P}_\theta$  is the distribution of the parametrized density  $P_\theta$ , then, asymptotically, this amounts to minimizing the Kullback-Leibler divergence  $KL(\mathbb{P}_r \parallel \mathbb{P}_\theta)$ .

For this to make sense, we need the model density  $P_\theta$  to exist. This is not the case in the rather common situation where we are dealing with distributions supported by low dimensional manifolds. It is then unlikely that the model manifold and the true distribution's support have a non-negligible intersection (see (Arjovsky & Bottou, 2017)), and this means that the KL distance is not defined (or simply infinite).

---

<sup>1</sup>Courant Institute of Mathematical Sciences, NY <sup>2</sup>Facebook AI Research, NY. Correspondence to: Martin Arjovsky <martin-arjovsky@gmail.com>.

The typical remedy is to add a noise term to the model distribution. This is why virtually all generative models described in the classical machine learning literature include a noise component. In the simplest case, one assumes a Gaussian noise with relatively high bandwidth in order to cover all the examples. It is well known, for instance, that in the case of image generation models, this noise degrades the quality of the samples and makes them blurry. For example, we can see in the recent paper (Wu et al., 2016) that the optimal standard deviation of the noise added to the model when maximizing likelihood is around 0.1 to each pixel in a generated image, when the pixels were already normalized to be in the range  $[0, 1]$ . This is a very high amount of noise, so much that when papers report the samples of their models, they don't add the noise term on which they report likelihood numbers. In other words, the added noise term is clearly incorrect for the problem, but is needed to make the maximum likelihood approach work.



Rather than estimating the density of  $\mathbb{P}_r$ , which may not exist, we can define a random variable  $Z$  with a fixed distribution  $p(z)$  and pass it through a parametric function  $g_\theta : \mathcal{Z} \rightarrow \mathcal{X}$  (typically a neural network of some kind) that directly generates samples following a certain distribution  $\mathbb{P}_\theta$ . By varying  $\theta$ , we can change this distribution and make it close to the real data distribution  $\mathbb{P}_r$ . This is useful in two ways. First of all, unlike densities, this approach can represent distributions confined to a low dimensional manifold. Second, the ability to easily generate samples is often more useful than knowing the numerical value of the density (for example in image superresolution or semantic segmentation when considering the conditional distribution of the output image given the input image). In general, it is computationally difficult to generate samples given an arbitrary high dimensional density (Neal, 2001).

Variational Auto-Encoders (VAEs) (Kingma & Welling, 2013) and Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) are well known examples of this approach. Because VAEs focus on the approximate likelihood of the examples, they share the limitation of the standard models and need to fiddle with additional noise terms. GANs offer much more flexibility in the definition of the objective function, including Jensen-Shannon (Goodfellow et al., 2014), and all  $f$ -divergences (Nowozin et al., 2016) as well as some exotic combinations (Huszar, 2015). On



the other hand, training GANs is well known for being delicate and unstable, for reasons theoretically investigated in (Arjovsky & Bottou, 2017).

In this paper, we direct our attention on the various ways to measure how close the model distribution and the real distribution are, or equivalently, on the various ways to define a distance or divergence  $\rho(\mathbb{P}_\theta, \mathbb{P}_r)$ . The most fundamental difference between such distances is their impact on the convergence of sequences of probability distributions. A sequence of distributions  $(\mathbb{P}_t)_{t \in \mathbb{N}}$  converges if and only if there is a distribution  $\mathbb{P}_\infty$  such that  $\rho(\mathbb{P}_t, \mathbb{P}_\infty)$  tends to zero, something that depends on how exactly the distance  $\rho$  is defined. Informally, a distance  $\rho$  induces a weaker topology when it makes it easier for a sequence of distribution to converge.<sup>1</sup> Section 2 clarifies how popular probability distances differ in that respect.

In order to optimize the parameter  $\theta$ , it is of course desirable to define our model distribution  $\mathbb{P}_\theta$  in a manner that makes the mapping  $\theta \mapsto \mathbb{P}_\theta$  continuous. Continuity means that when a sequence of parameters  $\theta_t$  converges to  $\theta$ , the distributions  $\mathbb{P}_{\theta_t}$  also converge to  $\mathbb{P}_\theta$ . However, it is essential to remember that the notion of the convergence of the distributions  $\mathbb{P}_{\theta_t}$  depends on the way we compute the distance between distributions. The weaker this distance, the easier it is to define a continuous mapping from  $\theta$ -space to  $\mathbb{P}_\theta$ -space, since it's easier for the distributions to converge. The main reason we care about the mapping  $\theta \mapsto \mathbb{P}_\theta$  to be continuous is as follows. If  $\rho$  is our notion of distance between two distributions, we would like to have a loss function  $\theta \mapsto \rho(\mathbb{P}_\theta, \mathbb{P}_r)$  that is continuous, and this is equivalent to having the mapping  $\theta \mapsto \mathbb{P}_\theta$  be continuous when using the distance between distributions  $\rho$ .

The contributions of this paper are:

- In Section 2, we provide a comprehensive theoretical analysis of how the Earth Mover (EM) distance behaves in comparison to popular probability distances and divergences used in the context of learning distributions.
- In Section 3, we define a form of GAN called Wasserstein-GAN that minimizes a reasonable and efficient approximation of the EM distance, and we theoretically show that the corresponding optimization problem is sound.
- In Section 4, we empirically show that WGANs cure the main training problems of GANs. In particular, training WGANs does not require maintaining a careful balance in training of the discriminator and the

generator, does not require a careful design of the network architecture either, and also reduces the mode dropping that is typical in GANs. One of the most compelling practical benefits of WGANs is the ability to continuously estimate the EM distance by training the discriminator to optimality. Because they correlate well with the observed sample quality, plotting these learning curves is very useful for debugging and hyperparameter searches.

## 2. Different Distances

We now introduce our notation. Let  $\mathcal{X}$  be a compact metric set, say the space of images  $[0, 1]^d$ , and let  $\Sigma$  denote the set of all the Borel subsets of  $\mathcal{X}$ . Let  $\text{Prob}(\mathcal{X})$  denote the space of probability measures defined on  $\mathcal{X}$ . We can now define elementary distances and divergences between two distributions  $\mathbb{P}_r, \mathbb{P}_g \in \text{Prob}(\mathcal{X})$ :

- The *Total Variation* (TV) distance

$$\delta(\mathbb{P}_r, \mathbb{P}_g) = \sup_{A \in \Sigma} |\mathbb{P}_r(A) - \mathbb{P}_g(A)|.$$

- The *Kullback-Leibler* (KL) divergence

$$KL(\mathbb{P}_r \| \mathbb{P}_g) = \int \log \left( \frac{P_r(x)}{P_g(x)} \right) P_r(x) d\mu(x),$$

where both  $\mathbb{P}_r$  and  $\mathbb{P}_g$  are assumed to admit densities with respect to a same measure  $\mu$  defined on  $\mathcal{X}$ .<sup>2</sup> The KL divergence is famously **asymmetric** and **possibly infinite** when there are points such that  $P_g(x) = 0$  and  $P_r(x) > 0$ .

- The *Jensen-Shannon* (JS) divergence

$$JS(\mathbb{P}_r, \mathbb{P}_g) = KL(\mathbb{P}_r \| \mathbb{P}_m) + KL(\mathbb{P}_g \| \mathbb{P}_m),$$

where  $\mathbb{P}_m$  is the mixture  $(\mathbb{P}_r + \mathbb{P}_g)/2$ . This divergence is **symmetrical** and always defined because we can choose  $\mu = \mathbb{P}_m$ .

- The *Earth-Mover* (EM) distance or **Wasserstein-1**

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|], \quad (1)$$

where  $\Pi(\mathbb{P}_r, \mathbb{P}_g)$  is the set of all joint distributions  $\gamma(x, y)$  whose marginals are respectively  $\mathbb{P}_r$  and  $\mathbb{P}_g$ . Intuitively,  $\gamma(x, y)$  indicates how much “mass” must be transported from  $x$  to  $y$  in order to transform the distributions  $\mathbb{P}_r$  into the distribution  $\mathbb{P}_g$ . The EM distance then is the “cost” of the optimal transport plan.

<sup>1</sup>More exactly, the topology induced by  $\rho$  is weaker than that induced by  $\rho'$  when the set of convergent sequences under  $\rho$  is a superset of that under  $\rho'$ .

<sup>2</sup>Recall that a probability distribution  $\mathbb{P}_r \in \text{Prob}(\mathcal{X})$  admits a density  $P_r(x)$  with respect to  $\mu$ , that is,  $\forall A \in \Sigma, \mathbb{P}_r(A) = \int_A P_r(x) d\mu(x)$ , if and only if it is absolutely continuous with respect to  $\mu$ , that is,  $\forall A \in \Sigma, \mu(A) = 0 \Rightarrow \mathbb{P}_r(A) = 0$ .

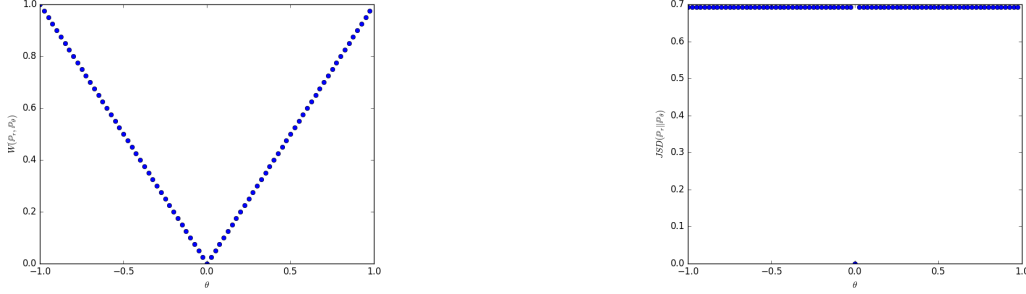


Figure 1: These plots show  $\rho(\mathbb{P}_\theta, \mathbb{P}_0)$  as a function of  $\theta$  when  $\rho$  is the EM distance (left plot) or the JS divergence (right plot). The EM plot is continuous and provides a usable gradient everywhere. The JS plot is not continuous and does not provide a usable gradient.

The following example illustrates how apparently simple sequences of probability distributions converge under the EM distance but do not converge under the other distances and divergences defined above.

**Example 1** (Learning parallel lines). Let  $Z \sim U[0, 1]$  the uniform distribution on the unit interval. Let  $\mathbb{P}_0$  be the distribution of  $(0, Z) \in \mathbb{R}^2$  (a 0 on the x-axis and the random variable  $Z$  on the y-axis), uniform on a straight vertical line passing through the origin. Now let  $g_\theta(z) = (\theta, z)$  with  $\theta$  a single real parameter. It is easy to see that in this case,

- $W(\mathbb{P}_0, \mathbb{P}_\theta) = |\theta|$ ,
- $JS(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} \log 2 & \text{if } \theta \neq 0, \\ 0 & \text{if } \theta = 0, \end{cases}$
- $KL(\mathbb{P}_0 \| \mathbb{P}_\theta) = KL(\mathbb{P}_0 \| \mathbb{P}_\theta) = \begin{cases} +\infty & \text{if } \theta \neq 0, \\ 0 & \text{if } \theta = 0, \end{cases}$
- and  $\delta(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} 1 & \text{if } \theta \neq 0, \\ 0 & \text{if } \theta = 0. \end{cases}$


When  $\theta_t \rightarrow 0$ , the sequence  $(\mathbb{P}_{\theta_t})_{t \in \mathbb{N}}$  converges to  $\mathbb{P}_0$  under the EM distance, but does not converge at all under either the JS, KL, reverse KL, or TV divergences. Figure 1 illustrates this for the case of the EM and JS distances.

Example 1 gives a case where we can learn a probability distribution over a low dimensional manifold by doing gradient descent on the EM distance. This cannot be done with the other distances and divergences because the resulting loss function is not even continuous. Although this simple example features distributions with disjoint supports, the same conclusion holds when the supports have a non empty intersection contained in a set of measure zero. This happens to be the case when two low dimensional manifolds intersect in general position (Arjovsky & Bottou, 2017).

Since the Wasserstein distance is much weaker than the JS

distance,<sup>3</sup> we can now ask whether  $W(\mathbb{P}_r, \mathbb{P}_\theta)$  is a continuous loss function on  $\theta$  under mild assumptions:

**Theorem 1.** Let  $\mathbb{P}_r$  be a fixed distribution over  $\mathcal{X}$ . Let  $Z$  be a random variable (e.g Gaussian) over another space  $\mathcal{Z}$ . Let  $\mathbb{P}_\theta$  denote the distribution of  $g_\theta(Z)$  where  $g : (z, \theta) \in \mathcal{Z} \times \mathbb{R}^d \mapsto g_\theta(z) \in \mathcal{X}$ . Then,

1. If  $g$  is continuous in  $\theta$ , so is  $W(\mathbb{P}_r, \mathbb{P}_\theta)$ . 
2. If  $g$  is locally Lipschitz and satisfies regularity assumption 1, then  $W(\mathbb{P}_r, \mathbb{P}_\theta)$  is continuous everywhere, and differentiable almost everywhere.
3. Statements 1-2 are false for the Jensen-Shannon divergence  $JS(\mathbb{P}_r, \mathbb{P}_\theta)$  and all the KLs.

As a consequence, learning by minimizing the EM distance makes sense (at least in theory) for neural networks:

**Corollary 1.** Let  $g_\theta$  be any feedforward neural network<sup>4</sup> parameterized by  $\theta$ , and  $p(z)$  a prior over  $z$  such that  $\mathbb{E}_{z \sim p(z)}[\|z\|] < \infty$  (e.g. Gaussian, uniform, etc.). Then assumption 1 is satisfied and therefore  $W(\mathbb{P}_r, \mathbb{P}_\theta)$  is continuous everywhere and differentiable almost everywhere.

Both proofs are given in Appendix C.

All this indicates that EM is a much more sensible cost function for our problem than at least the Jensen-Shannon divergence. The following theorem describes the relative strength of the topologies induced by these distances and divergences, with KL the strongest, followed by JS and TV, and EM the weakest.

**Theorem 2.** Let  $\mathbb{P}$  be a distribution on a compact space  $\mathcal{X}$  and  $(\mathbb{P}_n)_{n \in \mathbb{N}}$  be a sequence of distributions on  $\mathcal{X}$ . Then, considering all limits as  $n \rightarrow \infty$ ,

<sup>3</sup>Appendix A explains to the mathematically inclined reader why this happens and how we arrived to the idea that Wasserstein is what we should really be optimizing.

<sup>4</sup>By a feedforward neural network we mean a function composed of affine transformations and componentwise Lipschitz nonlinearities (such as the sigmoid, tanh, elu, softplus, etc). A similar but more technical proof is required for ReLUs.

1. The following statements are equivalent

- $\delta(\mathbb{P}_n, \mathbb{P}) \rightarrow 0$  with  $\delta$  the total variation distance.
- $JS(\mathbb{P}_n, \mathbb{P}) \rightarrow 0$  with  $JS$  the Jensen-Shannon divergence.

2. The following statements are equivalent

- $W(\mathbb{P}_n, \mathbb{P}) \rightarrow 0$ .
- $\mathbb{P}_n \xrightarrow{\mathcal{D}} \mathbb{P}$  where  $\xrightarrow{\mathcal{D}}$  represents convergence in distribution for random variables.

3.  $KL(\mathbb{P}_n \| \mathbb{P}) \rightarrow 0$  or  $KL(\mathbb{P} \| \mathbb{P}_n) \rightarrow 0$  imply the statements in (1).

4. The statements in (1) imply the statements in (2).

*Proof.* See Appendix C  $\square$

This highlights the fact that the KL, JS, and TV distances are not sensible cost functions when learning distributions supported by low dimensional manifolds. However the EM distance is sensible in that setup. This leads us to the next section where we introduce a practical approximation of optimizing the EM distance.

### 3. Wasserstein GAN

Again, Theorem 2 points to the fact that  $W(\mathbb{P}_r, \mathbb{P}_\theta)$  might have nicer properties when optimized than  $JS(\mathbb{P}_r, \mathbb{P}_\theta)$ . However, the infimum in (1) is highly intractable. On the other hand, the Kantorovich-Rubinstein duality (Villani, 2009) tells us that

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r}[f(x)] - \mathbb{E}_{x \sim \mathbb{P}_\theta}[f(x)] \quad (2)$$

where the supremum is over all the 1-Lipschitz functions  $f : \mathcal{X} \rightarrow \mathbb{R}$ . Note that if we replace  $\|f\|_L \leq 1$  for  $\|f\|_L \leq K$  (consider  $K$ -Lipschitz for some constant  $K$ ), then we end up with  $K \cdot W(\mathbb{P}_r, \mathbb{P}_\theta)$ . Therefore, if we have a parameterized family of functions  $\{f_w\}_{w \in \mathcal{W}}$  that are all  $K$ -Lipschitz for some  $K$ , we could consider solving the problem

$$\max_{w \in \mathcal{W}} \mathbb{E}_{x \sim \mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z \sim p(z)}[f_w(g_\theta(z))] \quad (3)$$

and if the supremum in (2) is attained for some  $w \in \mathcal{W}$  (a pretty strong assumption akin to what's assumed when proving consistency of an estimator), this process would yield a calculation of  $W(\mathbb{P}_r, \mathbb{P}_\theta)$  up to a multiplicative constant. Furthermore, we could consider differentiating  $W(\mathbb{P}_r, \mathbb{P}_\theta)$  (again, up to a constant) by back-proping through equation (2) via estimating  $\mathbb{E}_{z \sim p(z)}[\nabla_\theta f_w(g_\theta(z))]$ . While this is all intuition, we now prove that this process is principled under the optimality assumption.

**Algorithm 1** WGAN, our proposed algorithm. All experiments in the paper used the default values  $\alpha = 0.00005$ ,  $c = 0.01$ ,  $m = 64$ ,  $n_{\text{critic}} = 5$ .

**Require:**  $\alpha$ , the learning rate.  $c$ , the clipping parameter.  $m$ , the batch size.  $n_{\text{critic}}$ , the number of iterations of the critic per generator iteration.

**Require:**  $w_0$ , initial critic parameters.  $\theta_0$ , initial generator's parameters.

```

1: while  $\theta$  has not converged do
2:   for  $t = 0, \dots, n_{\text{critic}}$  do
3:     Sample  $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$  a batch from the real data.
4:     Sample  $\{z^{(i)}\}_{i=1}^m \sim p(z)$  a batch of priors.
5:      $g_w \leftarrow \nabla_w [\frac{1}{m} \sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(z^{(i)}))]$ 
6:      $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$ 
7:      $w \leftarrow \text{clip}(w, -c, c)$ 
8:   end for
9:   Sample  $\{z^{(i)}\}_{i=1}^m \sim p(z)$  a batch of prior samples.
10:   $g_\theta \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(z^{(i)}))$ 
11:   $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$ 
12: end while
    
```

**Theorem 3.** Let  $\mathbb{P}_r$  be any distribution. Let  $\mathbb{P}_\theta$  be the distribution of  $g_\theta(Z)$  with  $Z$  a random variable with density  $p$  and  $g_\theta$  a function satisfying assumption 1. Then, there is a solution  $f : \mathcal{X} \rightarrow \mathbb{R}$  to the problem

$$\max_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r}[f(x)] - \mathbb{E}_{x \sim \mathbb{P}_\theta}[f(x)]$$

and we have

$$\nabla_\theta W(\mathbb{P}_r, \mathbb{P}_\theta) = -\mathbb{E}_{z \sim p(z)}[\nabla_\theta f(g_\theta(z))]$$

when both terms are well-defined.

*Proof.* See Appendix C  $\square$

Now comes the question of finding the function  $f$  that solves the maximization problem in equation (2). To roughly approximate this, something that we can do is train a neural network parameterized with weights  $w$  lying in a compact space  $\mathcal{W}$  and then backprop through  $\mathbb{E}_{z \sim p(z)}[\nabla_\theta f_w(g_\theta(z))]$ , as we would do with a typical GAN. Note that the fact that  $\mathcal{W}$  is compact implies that all the functions  $f_w$  will be  $K$ -Lipschitz for some  $K$  that only depends on  $\mathcal{W}$  and not the individual weights, therefore approximating (2) up to an irrelevant scaling factor and the capacity of the ‘critic’  $f_w$ . In order to have parameters  $w$  lie in a compact space, something simple we can do is clamp the weights to a fixed box (say  $\mathcal{W} = [-0.01, 0.01]^l$ ) after each gradient update. The Wasserstein Generative Adversarial Network (WGAN) procedure is described in Algorithm 1.

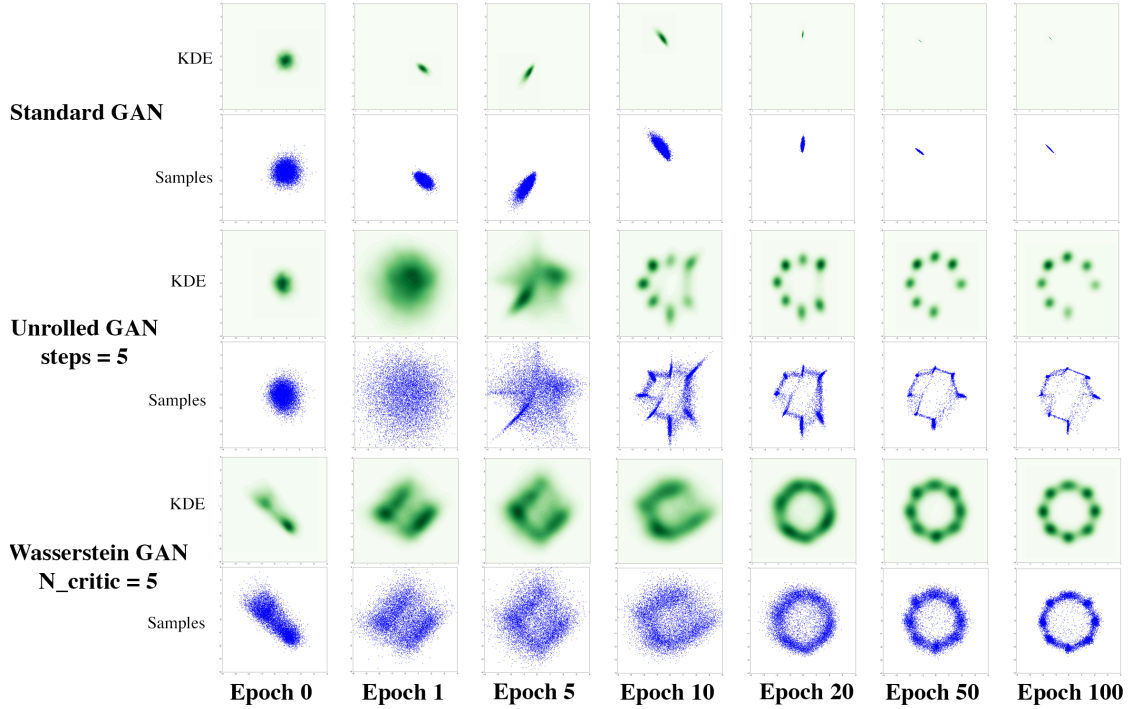


Figure 2: Different methods learning a mixture of 8 gaussians spread in a circle. WGAN is able to learn the distribution without mode collapse. An interesting fact is that the WGAN (much like the Wasserstein distance) seems to capture first the low dimensional structure of the data (the approximate circle) before matching the specific bumps in the density. Green: KDE plots. Blue: samples from the model.

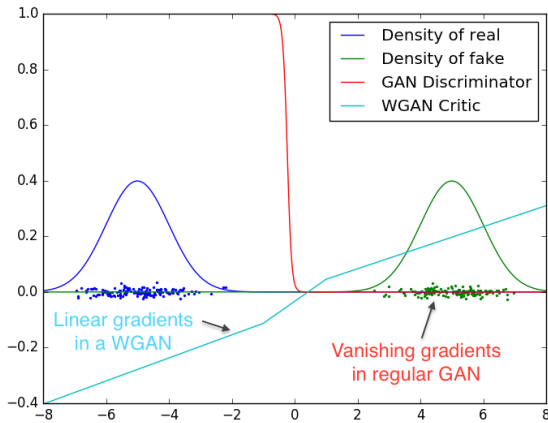


Figure 3: Optimal discriminator and critic when learning to differentiate two Gaussians. As we can see, the traditional GAN discriminator saturates and results in vanishing gradients. Our WGAN critic provides very clean gradients on all parts of the space.

The fact that the EM distance is continuous and differentiable a.e. means that we can (and should) train the critic till optimality. The argument is simple, the more we train the critic, the more reliable gradient of the Wasserstein we get, which is actually useful by the fact that Wasserstein

is differentiable almost everywhere. For the JS, as the discriminator gets better the gradients get more reliable but the true gradient is 0 since the JS is locally saturated and we get vanishing gradients, as can be seen in Figure 1 of this paper and Theorem 2.4 of (Arjovsky & Bottou, 2017). In Figure 3 we show a proof of concept of this, where we train a GAN discriminator and a WGAN critic till optimality. The discriminator learns very quickly to distinguish between fake and real, and as expected provides no reliable gradient information. The critic, however, can't saturate, and converges to a linear function that gives remarkably clean gradients everywhere. The fact that we constrain the weights limits the possible growth of the function to be at most linear in different parts of the space, forcing the optimal critic to have this behaviour.

Perhaps more importantly, the fact that we can train the critic till optimality makes it impossible to collapse modes when we do. This is due to the fact that mode collapse comes from the fact that the optimal generator for a fixed discriminator is a sum of deltas on the points the discriminator assigns the highest values, as observed by (Goodfellow et al., 2014) and highlighted in (Metz et al., 2016).

In the following section we display the practical benefits of our new algorithm, and we provide an in-depth comparison of its behaviour and that of traditional GANs.



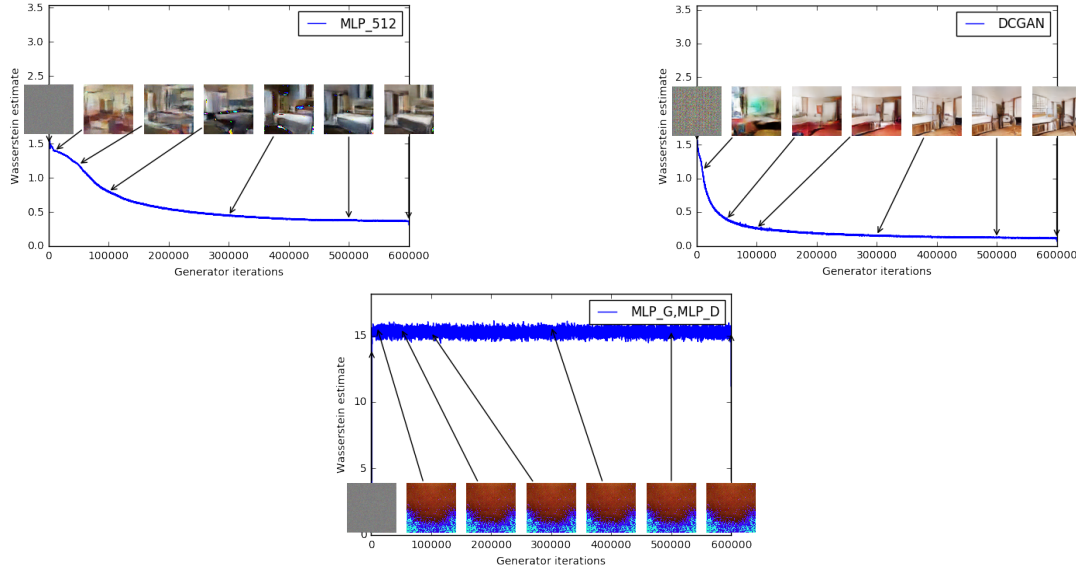


Figure 4: Training curves and samples at different stages of training. We can see a clear correlation between lower error and better sample quality. Upper left: the generator is an MLP with 4 hidden layers and 512 units at each layer. The loss decreases consistently as training progresses and sample quality increases. Upper right: the generator is a standard DCGAN. The loss decreases quickly and sample quality increases as well. In both upper plots the critic is a DCGAN without the sigmoid so losses can be subjected to comparison. Lower half: both the generator and the discriminator are MLPs with substantially high learning rates (so training failed). Loss is constant and samples are constant as well. The training curves were passed through a median filter for visualization purposes.

## 4. Empirical Results

We run experiments on image generation using our Wasserstein-GAN algorithm and show that there are significant practical benefits to using it over the formulation used in standard GANs. We claim two main benefits:

- a meaningful loss metric that correlates with the generator’s convergence and sample quality
- improved stability of the optimization process

### 4.1. Mixtures of Gaussians

In (Metz et al., 2016) the authors presented a simple mixture of Gaussians experiments that served a very specific purpose. In this mixture, the mode collapse problem of GANs is easy to visualize, since a normal GAN would rotate between the different modes of the mixture, and fail to capture the whole distribution. In 2 we show how our WGAN algorithm approximately finds the correct distribution, without any mode collapse.

An interesting thing is that the WGAN first seems to learn to match the low-dimensional structure of the data (the approximate circle), before zooming in on the specific bumps of the true density. Similar to the Wasserstein distance, it looks like WGAN gives more importance to matching the low dimensional supports rather than the specific ratios between the densities.

### 4.2. Experimental Procedure for Image Generation

We run experiments on image generation. The target distribution to learn is the LSUN-Bedrooms dataset (Yu et al., 2015) – a collection of natural images of indoor bedrooms. Our baseline comparison is DCGAN (Radford et al., 2015), a GAN with a convolutional architecture trained with the standard GAN procedure using the  $-\log D$  trick (Goodfellow et al., 2014). The generated samples are 3-channel images of 64x64 pixels in size. We use the hyper-parameters specified in Algorithm 1 for all of our experiments.

### 4.3. Meaningful loss metric

Because the WGAN algorithm attempts to train the critic  $f$  (lines 2–8 in Algorithm 1) relatively well before each generator update (line 10 in Algorithm 1), the loss function at this point is an estimate of the EM distance, up to constant factors related to the way we constrain the Lipschitz constant of  $f$ .

Our first experiment illustrates how this estimate correlates well with the quality of the generated samples. Besides the convolutional DCGAN architecture, we also ran experiments where we replace the generator or both the generator and the critic by 4-layer ReLU-MLP with 512 hidden units.

Figure 4 plots the evolution of the WGAN estimate (3) of the EM distance during WGAN training for all three architectures. The plots clearly show that these curves correlate

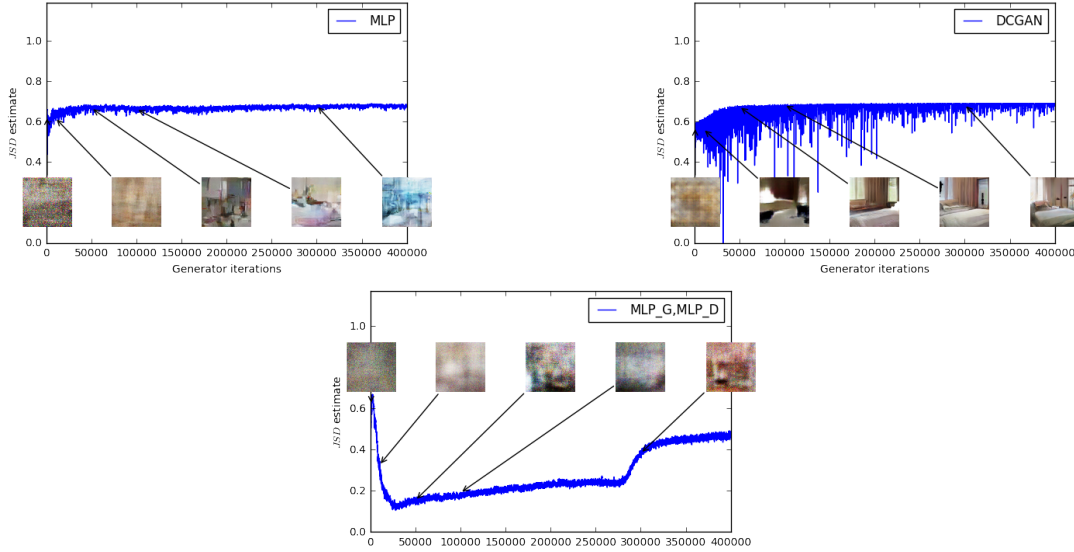


Figure 5: JS estimates for an MLP generator (upper left) and a DCGAN generator (upper right) trained with the standard GAN procedure. Both had a DCGAN discriminator. Both curves have increasing error. Samples get better for the DCGAN but the JS estimate increases or stays constant, pointing towards no significant correlation between sample quality and loss. Bottom: MLP with both generator and discriminator. The curve goes up and down regardless of sample quality. All training curves were passed through the same median filter as in Figure 4.

well with the visual quality of the generated samples.

To our knowledge, this is the first time in GAN literature that such a property is shown, where the loss of the GAN shows properties of convergence. This property is extremely useful when doing research in adversarial networks as one does not need to stare at the generated samples to figure out failure modes and to gain information on which models are doing better over others.

However, we do not claim that this is a new method to quantitatively evaluate generative models yet. The constant scaling factor that depends on the critic’s architecture means it’s hard to compare models with different critics. Even more, in practice the fact that the critic doesn’t have infinite capacity makes it hard to know just how close to the EM distance our estimate really is. This being said, we have successfully used the loss metric to validate our experiments repeatedly and without failure, and we see this as a huge improvement in training GANs which previously had no such facility.

In contrast, Figure 5 plots the evolution of the GAN estimate of the JS distance during GAN training. More precisely, during GAN training, the discriminator is trained to maximize

$$L(D, g_\theta) = \mathbb{E}_{x \sim \mathbb{P}_r} [\log D(x)] + \mathbb{E}_{x \sim \mathbb{P}_\theta} [\log(1 - D(x))]$$

which is a lower bound of  $2JS(\mathbb{P}_r, \mathbb{P}_\theta) - 2 \log 2$ . In the figure, we plot the quantity  $\frac{1}{2}L(D, g_\theta) + \log 2$ , which is a lower bound of the JS distance.

This quantity clearly correlates poorly the sample quality. Note also that the JS estimate usually stays constant or goes up instead of going down. In fact it often remains very close to  $\log 2 \approx 0.69$  which is the highest value taken by the JS distance. In other words, the JS distance saturates, the discriminator has zero loss, and the generated samples are in some cases meaningful (DCGAN generator, top right plot) and in other cases collapse to a single nonsensical image (Goodfellow et al., 2014). This last phenomenon has been theoretically explained in (Arjovsky & Bottou, 2017) and highlighted in (Metz et al., 2016).

When using the  $-\log D$  trick (Goodfellow et al., 2014), the discriminator loss and the generator loss are different. Figure 9 in Appendix F reports the same plots for GAN training, but using the generator loss instead of the discriminator loss. This does not change the conclusions.

Finally, as a negative result, we report that WGAN training becomes unstable at times when one uses a momentum based optimizer such as Adam (Kingma & Ba, 2014) (with  $\beta_1 > 0$ ) on the critic, or when one uses high learning rates. Since the loss for the critic is nonstationary, momentum based methods seemed to perform worse. We identified momentum as a potential cause because, as the loss blew up and samples got worse, the cosine between the Adam step and the gradient usually turned negative. The only places where this cosine was negative was in these situations of instability. We therefore switched to RMSProp (Tieleman & Hinton, 2012) which is known to perform well even on very nonstationary problems (Mnih et al., 2016).



Figure 6: Algorithms trained with a DCGAN generator. Left: WGAN algorithm. Right: standard GAN formulation. Both algorithms produce high quality samples.

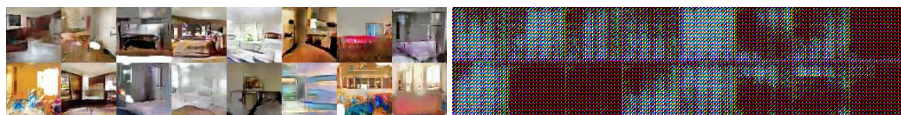


Figure 7: Algorithms trained with a generator without batch normalization and constant number of filters at every layer (as opposed to duplicating them every time as in (Radford et al., 2015)). Aside from taking out batch normalization, the number of parameters is therefore reduced by a bit more than an order of magnitude. Left: WGAN algorithm. Right: standard GAN formulation. As we can see the standard GAN failed to learn while the WGAN still was able to produce samples.

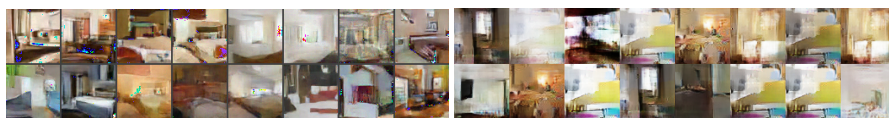


Figure 8: Algorithms trained with an MLP generator with 4 layers and 512 units with ReLU nonlinearities. The number of parameters is similar to that of a DCGAN, but it lacks a strong inductive bias for image generation. Left: WGAN algorithm. Right: standard GAN formulation. The WGAN method still was able to produce samples, lower quality than the DCGAN, and of higher quality than the MLP of the standard GAN. Note the significant degree of mode collapse in the GAN MLP.

#### 4.4. Improved stability

One of the benefits of WGAN is that it allows us to train the critic till optimality. When the critic is trained to completion, it simply provides a loss to the generator that we can train as any other neural network. This tells us that we no longer need to balance generator and discriminator’s capacity properly. The better the critic, the higher quality the gradients we use to train the generator.

We observe that WGANs are more robust than GANs when one varies the architectural choices for the generator in certain ways. We illustrate this by running experiments on three generator architectures: (1) a convolutional DCGAN generator, (2) a convolutional DCGAN generator without batch normalization and with a constant number of filters (the capacity of the generator is drastically smaller than that of the discriminator), and (3) a 4-layer ReLU-MLP with 512 hidden units. The last two are known to perform very poorly with GANs. We keep the convolutional DCGAN architecture for the WGAN critic or the GAN discriminator.

Figures 6, 7, and 8 show samples generated for these three architectures using both the WGAN and GAN algorithms. We refer the reader to Appendix H for full sheets of generated samples. Samples were not cherry-picked.

**In no experiment did we see evidence of mode collapse for the WGAN algorithm.**

## 5. Related Work

We refer the reader to Appendix D for the connections to the different integral probability metrics (Müller, 1997).

The recent work of (Montavon et al., 2016) has explored the use of Wasserstein distances in the context of learning for Restricted Boltzmann Machines for discrete spaces. Even though the motivations at a first glance might seem quite different, at the core of it both our works want to compare distributions in a way that leverages the geometry of the underlying space, which Wasserstein allows us to do.

Finally, the work of (Genevay et al., 2016) shows new algorithms for calculating Wasserstein distances between different distributions. We believe this direction is quite important, and perhaps could lead to new ways to evaluate generative models.

## 6. Conclusion

We introduced an algorithm that we deemed WGAN, an alternative to traditional GAN training. In this new model, we showed that we can improve the stability of learning, get rid of problems like mode collapse, and provide meaningful learning curves useful for debugging and hyperparameter searches. Furthermore, we showed that the corresponding optimization problem is sound, and provided extensive theoretical work highlighting the deep connections to other distances between distributions.



## Acknowledgments

We would like to thank Mohamed Ishmael Belghazi, Emily Denton, Ian Goodfellow, Ishaan Gulrajani, Alex Lamb, David Lopez-Paz, Eric Martin, musyoku, Maxime Oquab, Aditya Ramesh, Ronan Riochet, Uri Shalit, Pablo Sprechmann, Arthur Szlam, Ruohan Wang, for helpful comments and advice.

## References

- Arjovsky, Martin and Bottou, Léon. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017.
- Dziugaite, Gintare Karolina, Roy, Daniel M., and Ghahramani, Zoubin. Training generative neural networks via maximum mean discrepancy optimization. *CoRR*, abs/1505.03906, 2015.
- Genevay, Aude, Cuturi, Marco, Peyré, Gabriel, and Bach, Francis. Stochastic optimization for large-scale optimal transport. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 3440–3448. Curran Associates, Inc., 2016.
- Goodfellow, Ian J., Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014.
- Gretton, Arthur, Borgwardt, Karsten M., Rasch, Malte J., Schölkopf, Bernhard, and Smola, Alexander. A kernel two-sample test. *J. Mach. Learn. Res.*, 13:723–773, 2012. ISSN 1532-4435.
- Huszar, Ferenc. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *CoRR*, abs/1511.05101, 2015.
- Kakutani, Shizuo. Concrete representation of abstract (m)-spaces (a characterization of the space of continuous functions). *Annals of Mathematics*, 42(4):994–1024, 1941.
- Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- Kingma, Diederik P. and Welling, Max. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.
- Li, Yujia, Swersky, Kevin, and Zemel, Rich. Generative moment matching networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 1718–1727. JMLR Workshop and Conference Proceedings, 2015.
- Metz, Luke, Poole, Ben, Pfau, David, and Sohl-Dickstein, Jascha. Unrolled generative adversarial networks. *Corr*, abs/1611.02163, 2016.
- Milgrom, Paul and Segal, Ilya. Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2):583–601, 2002. ISSN 1468-0262.
- Mnih, Volodymyr, Badia, Adrià Puigdomènech, Mirza, Mehdi, Graves, Alex, Lillicrap, Timothy P., Harley, Tim, Silver, David, and Kavukcuoglu, Koray. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pp. 1928–1937, 2016.
- Montavon, Grégoire, Müller, Klaus-Robert, and Cuturi, Marco. Wasserstein training of restricted boltzmann machines. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 3718–3726. Curran Associates, Inc., 2016.
- Müller, Alfred. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997.
- Neal, Radford M. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, April 2001. ISSN 0960-3174.
- Nowozin, Sebastian, Cseke, Botond, and Tomioka, Ryota. f-gan: Training generative neural samplers using variational divergence minimization. pp. 271–279, 2016.
- Radford, Alec, Metz, Luke, and Chintala, Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.
- Ramdas, Aaditya, Reddi, Sashank J., Poczos, Barnabas, Singh, Aarti, and Wasserman, Larry. On the high-dimensional power of linear-time kernel two-sample testing under mean-difference alternatives. *Corr*, abs/1411.6314, 2014.
- Sutherland, Dougal J, Tung, Hsiao-Yu, Strathmann, Heiko, De, Soumyajit, Ramdas, Aaditya, Smola, Alex, and Gretton, Arthur. Generative models and model criticism via optimized maximum mean discrepancy. In *International Conference on Learning Representations*, 2017.

Tieleman, T. and Hinton, G. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

Villani, Cédric. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer, Berlin, 2009.

Wu, Yuhuai, Burda, Yuri, Salakhutdinov, Ruslan, and Grosse, Roger B. On the quantitative analysis of decoder-based generative models. *CoRR*, abs/1611.04273, 2016.

Yu, Fisher, Zhang, Yinda, Song, Shuran, Seff, Ari, and Xiao, Jianxiong. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *Corr*, abs/1506.03365, 2015.

Zhao, Junbo, Mathieu, Michael, and LeCun, Yann. Energy-based generative adversarial network. *Corr*, abs/1609.03126, 2016.