

Max-Sliced Wasserstein Distance and its use for GANs

Ishan Deshpande, Yuan-Ting Hu, Ruoyu Sun, Ayis Pyrros[†], Nasir Siddiqui[†],
Sanmi Koyejo, Zhizhen Zhao, David Forsyth, Alexander Schwing
University of Illinois at Urbana-Champaign [†]Dupage Medical Group

ishan.sd@gmail.com, {ythu2, ruoyus}@illinois.edu, ayis@ayis.org, nsiddiqui@gmail.com,
{sanmi, zhizhenz, daf, aschwing}@illinois.edu

Abstract

Generative adversarial nets (GANs) and variational auto-encoders have significantly improved our distribution modeling capabilities, showing promise for dataset augmentation, image-to-image translation and feature learning. However, to model high-dimensional distributions, sequential training and stacked architectures are common, increasing the number of tunable hyper-parameters as well as the training time. Nonetheless, the sample complexity of the distance metrics remains one of the factors affecting GAN training. We first show that the recently proposed sliced Wasserstein distance has compelling sample complexity properties when compared to the Wasserstein distance. To further improve the sliced Wasserstein distance we then analyze its ‘projection complexity’ and develop the max-sliced Wasserstein distance which enjoys compelling sample complexity while reducing projection complexity, albeit necessitating a max estimation. We finally illustrate that the proposed distance trains GANs on high-dimensional images up to a resolution of 256x256 easily.

1. Introduction

Generative modeling capabilities have improved tremendously in the last few years, especially since the advent of deep learning-based models like generative adversarial nets (GANs) [11] and variational auto-encoders (VAEs) [17]. Instead of sampling from a high-dimensional distribution, GANs and VAEs transform a sample obtained from a simple distribution using deep nets. These models have found use in dataset augmentation [31], image-to-image translation [15, 37, 21, 14, 24, 29, 35, 38], and even feature learning for inference related tasks [9].

GANs and many of their variants formulate generative modeling as a two player game. A ‘generator’ creates samples that resemble the ground truth data. A ‘discriminator’ tries to distinguish between ‘artificial’ and ‘real’ samples.

Both, the generator and discriminator, are parametrized using deep nets and trained via stochastic gradient descent. In its original formulation [11], a GAN minimizes the Jensen-Shannon divergence between the data distribution and the probability distribution induced in the data space by the generator. Many other variants have been proposed, which use either some divergence or the integral probability metric to measure the distance between the distributions [2, 22, 12, 20, 8, 7, 27, 4, 26, 23, 13, 30]. When carefully trained, GANs are able to produce high quality samples [28, 16, 25, 16, 25]. Training GANs is, however, difficult – especially on high dimensional datasets.

The scaling difficulty of GANs may be related to one fundamental theoretical issue: the sample complexity. It is shown in [3] that KL-divergence, Jensen-Shannon and Wasserstein distance do not generalize, in the sense that the population distance cannot be approximated by an empirical distance when there are only a polynomial number of samples. To improve generalization, one popular method is to limit the discriminator class [3, 10] and interpret the training process as minimizing a neural-net distance [3].

In this work, we promote a different path that resolves the sample complexity issue. A fundamental reason for the exponential sample complexity of the Wasserstein distance is the sparsity of points in a high dimensional space. Even if two collections of points are randomly drawn from the same ball, these two collections are far away from each other. Our intuition is that projection onto a low-dimensional subspace, such as a line, mitigates the artificial distance effect in high dimensions and the distance of the projected samples reflects the true distance.

We first apply this intuition to analyze the recently proposed sliced Wasserstein distance GAN, which is based on the average Wasserstein distance of the projected versions of two distributions along a few randomly picked directions [8, 20, 34]. We prove that the sliced Wasserstein distance is generalizable for Gaussian distributions (*i.e.*, it has polynomial sample complexity), while Wasserstein distance is not, thus partially explaining why [8, 20, 34] may exhibit

better behavior than the Wasserstein distance [2].

One drawback of the sliced Wasserstein distance is that it requires a large number of projection directions, since random directions lose a lot of information. To address this concern, we propose to project onto the “best direction,” along which the projected distance is maximized. We call the corresponding metric the “max-sliced Wasserstein distance,” and prove that it is also generalizable for Gaussian distributions.

Using this new metric, we are able to train GANs to generate high resolution images from the CelebA-HQ [16] and LSUN Bedrooms [36] datasets. We also achieve improved performance in other distribution matching tasks like unpaired word translation [6].

The main contributions of this paper are the following:

- We analyze in Sec. 3.1 the sample complexity of the Wasserstein and sliced Wasserstein distances. We show that for a certain class of distributions the Wasserstein distance has an exponential sample complexity, while the sliced Wasserstein distance [8, 34] has a polynomial sample complexity.
- We then study in Sec. 3.2 the projection complexity of the sliced Wasserstein distance, *i.e.*, how the number of random projection directions affects estimation.
- We introduce the max-sliced Wasserstein distance in Sec. 3.3 to address the projection complexity issue.
- We then employ the max-sliced Wasserstein distance to train GANs in Sec. 4, demonstrating significant reduction in the number of projection directions required for the sliced-Wasserstein GAN.

2. Background

Generative modeling is the task of learning a probability distribution from a given dataset $\mathcal{D} = \{(x)\}$ of samples $x \sim \mathbb{P}_d$ drawn from an unknown data distribution \mathbb{P}_d . While this has traditionally been seen through the lens of likelihood-maximization, GANs pose generative modeling as a distance minimization problem. More specifically, these approaches recommend learning the data distribution \mathbb{P}_d by finding a distribution \mathbb{P}_g that solves:

$$\operatorname{argmin}_{\mathbb{P}_g} D(\mathbb{P}_g, \mathbb{P}_d), \quad (1)$$

where $D(\cdot, \cdot)$ is some distance or divergence between distributions. Arjovsky *et al.* [1] proposed using the Wasserstein distance in the context of GAN formulations. The Wasserstein-p distance between distributions \mathbb{P}_g and \mathbb{P}_d is defined as:

$$W_p(\mathbb{P}_g, \mathbb{P}_d) = \inf_{\gamma \in \Pi(\mathbb{P}_g, \mathbb{P}_d)} (\mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|^p])^{\frac{1}{p}}, \quad (2)$$

where $\Pi(\mathbb{P}_g, \mathbb{P}_d)$ is the set of all possible joint distributions on (x, y) with marginals \mathbb{P}_g and \mathbb{P}_d .

Estimating the Wasserstein distance is, however, not straightforward. Arjovsky *et al.* [2] used the Kantorovich-Rubinstein duality to the Wasserstein-1 distance, which states that:

$$W(\mathbb{P}_g, \mathbb{P}_d) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_g} [f(x)] - \mathbb{E}_{x \sim \mathbb{P}_d} [f(x)], \quad (3)$$

where the supremum is over all 1-Lipschitz functions $f : \mathcal{X} \rightarrow \mathbb{R}$. The function f is commonly represented via a deep net and various ways have been suggested to enforce the Lipschitz constraint, *e.g.*, [12].

While the Wasserstein distance based approaches have been successful in several complex generative tasks, they suffer from instability arising from incorrect estimation. The cause behind this was noted in [33], where it was shown that estimates of the Wasserstein distance suffer from the ‘curse of dimensionality.’ To tackle the instability and complexity, a sliced version of the Wasserstein-2 distance was employed by [8, 20, 18, 34], which only requires estimating distances of 1-d distributions and is, therefore, more efficient. The “sliced Wasserstein-p distance” [5] between distributions \mathbb{P}_d and \mathbb{P}_g is defined as

$$\tilde{W}_p(\mathbb{P}_d, \mathbb{P}_g) = \left[\int_{\omega \in \Omega} W_p^p(\mathbb{P}_d^\omega, \mathbb{P}_g^\omega) d\omega \right]^{\frac{1}{p}}, \quad (4)$$

where $\mathbb{P}_g^\omega, \mathbb{P}_d^\omega$ denote the projection (*i.e.*, marginal) of $\mathbb{P}_g, \mathbb{P}_d$ onto the direction ω , and Ω is the set of all possible directions on the unit sphere. Kolouri *et al.* [19] have shown that the sliced Wasserstein distance satisfies the properties of non-negativity, identity of indiscernibles, symmetry, and subadditivity. Hence, it is a true metric.

In practice, Deshpande *et al.* [8] approximate the sliced Wasserstein-2 distance between the distributions by using samples $\mathcal{D} \sim \mathbb{P}_d, \mathcal{F} \sim \mathbb{P}_g$, and a finite number of random Gaussian directions, replacing the integration over Ω with a summation over a randomly chosen set of unit vectors $\hat{\Omega} \propto \mathcal{N}(0, I)$, where ‘ \propto ’ is used to indicate normalization to unit length. With \mathbb{P}_g (and hence, \mathcal{F}) being implicitly parametrized by θ_g , [8] uses the following program for generative modeling:

$$\min_{\theta_g} \frac{1}{|\hat{\Omega}|} \sum_{\omega \in \hat{\Omega}} W_2^2(\mathcal{D}^\omega, \mathcal{F}^\omega). \quad (5)$$

The Wasserstein-2 distance between the projected samples \mathcal{D}^ω and \mathcal{F}^ω can be computed by finding the optimal transport map. For 1-d distributions, this can be done through sorting [32], *i.e.*,

$$W_2^2(\mathcal{D}^\omega, \mathcal{F}^\omega) = \frac{1}{|\mathcal{D}|} \sum_i \|\mathcal{D}_{\pi_D(i)}^\omega - \mathcal{F}_{\pi_F(i)}^\omega\|_2^2, \quad (6)$$

where $\pi_{\mathcal{D}}$ and $\pi_{\mathcal{F}}$ are permutations that sort the projected sample sets \mathcal{D}^ω and \mathcal{F}^ω respectively, i.e., $\mathcal{D}_{\pi_{\mathcal{D}}(1)}^\omega \leq \mathcal{D}_{\pi_{\mathcal{D}}(2)}^\omega \leq \dots \leq \mathcal{D}_{\pi_{\mathcal{D}}(|\mathcal{D}|)}^\omega$.

The program in Eq. (5), when coupled with a discriminator, was shown to work well on high-dimensional datasets. Instead of working directly with sets \mathcal{D} and \mathcal{F} , it was proposed that we transform them to an adversarially learnt feature space, say $h_{\mathcal{D}}$ and $h_{\mathcal{F}}$ respectively, where h is implicitly parameterized by θ_d , e.g., by using a deep net. The generator, parametrized by θ_g , minimizes

$$\min_{\theta_g} \frac{1}{|\hat{\Omega}|} \sum_{\omega \in \hat{\Omega}} W_2^2(h_{\mathcal{D}}^\omega, h_{\mathcal{F}}^\omega). \quad (7)$$

The adversarial feature space h is learnt via a discriminator which classifies real and fake data. This discriminator can be written as $\omega_d^T h$, where ω_d is a logistic layer and the parameters are learnt using

$$\hat{\theta}_d, \hat{\omega}_d = \operatorname{argmax}_{\theta_d, \omega_d} \sum_{x \in \mathcal{D}} \ln(\sigma(\omega_d^T h_x)) + \sum_{\hat{x} \in \mathcal{F}} \ln(1 - \sigma(\omega_d^T h_{\hat{x}})). \quad (8)$$

3. Analysis and Max-Sliced Distance

In this section we provide the first analysis of the sample-complexity benefits of the sliced Wasserstein distance compared to the Wasserstein distance. We discuss how ‘projection complexity’ is a shortcoming of the sliced Wasserstein distance and present as a fix the max-sliced Wasserstein distance, which – as we will show – enjoys the same beneficial sample-complexity as the slice Wasserstein distance, albeit necessitating estimation of a maximum. We will then show how those results are used for training GANs.

3.1. Sample complexity of the Wasserstein and sliced Wasserstein distances

We first show the benefits of using the sliced Wasserstein distance over the Wasserstein distance. Specifically, we show that, in certain cases, estimation of the sliced Wasserstein distance has polynomial complexity, while the Wasserstein distance does not. To make this notion concrete, we introduce ‘generalizability’ of a distance:

Definition 1 Consider a family of distributions \mathcal{P} over \mathbb{R}^d . A distance $\operatorname{dist}(\cdot, \cdot)$ is said to be \mathcal{P} -generalizable if there exists a polynomial g such that for any two distributions $\mu, \nu \in \mathcal{P}$, and their empirical ensembles $\hat{\mu}, \hat{\nu}$ with size $n = g(d, 1/\epsilon), \epsilon > 0$, the following holds:

$$|\operatorname{dist}(\mu, \nu) - \operatorname{dist}(\hat{\mu}, \hat{\nu})| \leq \epsilon \text{ w.p. } \geq 1 - \text{polynomial}(-n).$$

With this definition, we can prove the following result:

Claim 1 Consider the family of Gaussian distributions

$$\mathcal{P} = \{\mathcal{N}(a, I) \mid a \in \mathbb{R}^d\}.$$

The sliced Wasserstein-2 distance \tilde{W}_2 defined in Eq. (4) is \mathcal{P} -generalizable whereas the Wasserstein-2 distance W_2 defined in Eq. (2) is not.

Proof. See the supplementary material. ■

Claim 1 implies that for GAN training, under certain conditions, it is better to use the sliced Wasserstein distance as we can get a more accurate training signal with a fixed computational budget. This will result in a more stable discriminator.

Even though the sliced Wasserstein distance enjoys better sample complexity, it has limitations when a finite number of random projection directions is used. We refer to this property as ‘projection complexity’ and illustrate it in the following section. We then present our proposed method to help alleviate this problem.

3.2. Projection complexity of the Sliced Wasserstein Distance

We begin with a simple example to demonstrate the limitations of using \tilde{W}_2 defined in Eq. (4) for learning distributions through gradient descent. To analyze the ‘projection complexity’ of \tilde{W}_2 we use infinitely many samples, but we use only finitely many directions $\omega \in \hat{\Omega}$.

Concretely, consider two d -dimensional Gaussians μ, ν with identity covariance. Let $\mu = \mathcal{N}(0, I) = \mathbb{P}_d$ be the data distribution and let $\nu = \mathcal{N}(\beta \hat{e}, I) = \mathbb{P}_g$ be the induced generator distribution, parametrized only by its mean β , while \hat{e} is a fixed unit vector. Using gradient descent on the estimated sliced Wasserstein distance between μ and ν , we aim to learn β so that $\mu = \nu$. Thus, the updates for β are

$$\beta \leftarrow \beta - \alpha \nabla_{\beta} \tilde{W}_2(\mu, \nu), \quad (9)$$

where α is the learning rate.

The sliced Wasserstein distance \tilde{W}_2 is calculated by projecting the distributions (since we use infinitely many samples) onto random directions and comparing the projections, i.e., marginals. Therefore, the estimated distance is

$$\tilde{W}_2(\mu, \nu) = \frac{1}{|\hat{\Omega}|} \sum_{\omega \in \hat{\Omega}} W_2(\mu^\omega, \nu^\omega), \quad (10)$$

where $W_2(\mu^\omega, \nu^\omega)$ is the Wasserstein distance between marginal distributions μ^ω, ν^ω . Note that each ω is normalized to unit norm.

Intuitively, projection of the Gaussians μ, ν onto any direction other than \hat{e} makes them appear closer than they actually are – making the learning process slower. For any

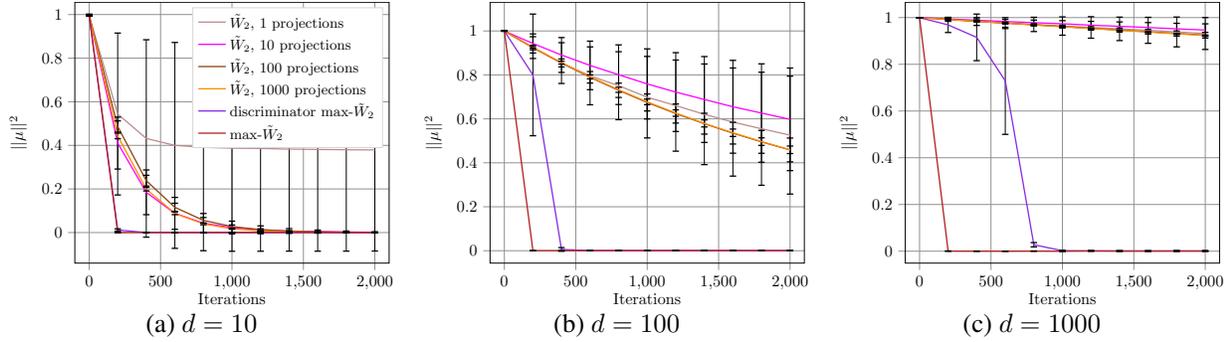


Figure 1: Convergence of the mean for different sampling strategies for learning the mean of a d -dimensional Gaussian using the sliced Wasserstein distance and the max-sliced Wasserstein distance. Numbers in the legend denote the number of projection directions used.

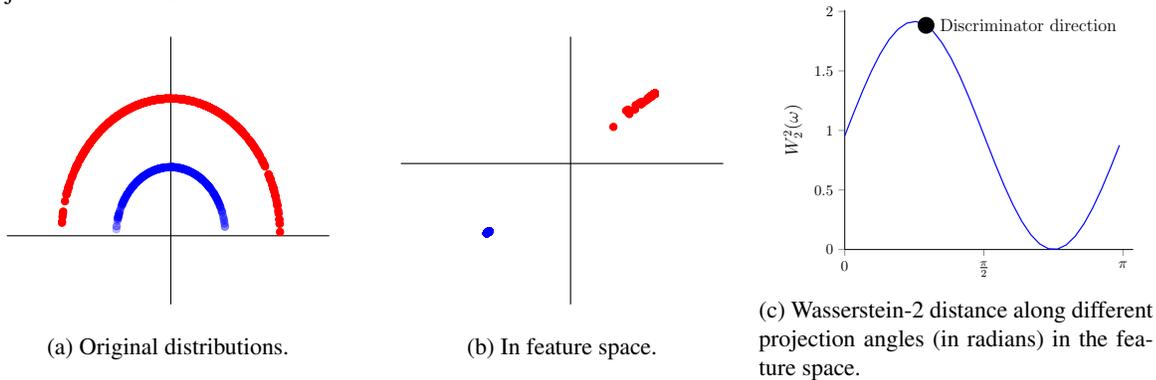


Figure 2: The discriminator is able to identify important projection directions. The discriminator transforms the distributions in Fig. 2a to Fig. 2b. In this new space, the discriminator’s direction is aligned with the one along which the distributions are the most dissimilar as shown in Fig. 2c.

given ω , it is easy to see that $W_2(\mu^\omega, \nu^\omega) = \beta |\hat{e}^T \omega|$. Therefore, the update equation for β is

$$\beta \rightarrow \beta - \alpha \frac{1}{|\hat{\Omega}|} \sum_{\omega \in \hat{\Omega}} |\hat{e}^T \omega|. \quad (11)$$

The updates to β are particularly small for high dimensional distributions, since any random unit-norm direction ω is orthogonal to \hat{e} with high probability. Therefore, $\beta \rightarrow 0$ very slowly. We verify this effect empirically in Fig. 1, experimenting with different numbers of random projections and find that using the sliced Wasserstein distance results in very slow convergence. This problem is further aggravated when the dimensions of the distributions increase.

It is intuitively obvious that the aforementioned problem can easily be solved by choosing \hat{e} as the projection direction. This results in larger updates and, consequently, faster convergence. This intuition is also verified empirically. We repeat the same experiment of learning β , but this time we use only one projection direction $\omega = \hat{e}$. This is labelled as max- \tilde{W}_2 in Fig. 1. By simply using the important projection direction, we achieve fast convergence of the mean.

Considering this example, it is evident that some projec-

tion directions are more meaningful than others. Therefore, GAN training should benefit from including such directions when comparing distributions. This observation motivates the max-sliced Wasserstein distance which we discuss next.

3.3. Max sliced Wasserstein distance

In this section we introduce the max-sliced Wasserstein distance and illustrate that it fixes the ‘projection complexity’ concern. We also prove that the max-sliced Wasserstein distance enjoys the same sample-complexity as the sliced Wasserstein distance, *i.e.*, we are not trading one benefit for another.

As noted in Sec. 3.2, it is useful to include the most meaningful projection direction. Formally, for the aforementioned example of $\mu = \mathcal{N}(0, I), \nu = \mathcal{N}(\beta \hat{e}, I)$, we want to use the direction ω^* that satisfies

$$\omega^* = \operatorname{argmax}_{\omega \in \Omega} |\hat{e}^T \omega|. \quad (12)$$

Comparing distributions along such a direction ω^* can, in fact, be shown to be a proper distance. We call it the ‘max-sliced Wasserstein distance’ and define it as follows:

Algorithm 1: Training the improved Sliced Wasserstein Generator

Given : Generator parameters θ_g , Discriminator parameters θ_d, ω_d , sample size n , learning rate α

```
1 while  $\theta_g$  not converged do
2   for  $i \leftarrow 0$  to  $k$  do
3     Sample data  $\{\mathcal{D}^i\}_{i=1}^n \sim \mathbb{P}_d$ , generated samples  $\{\mathcal{F}_{\theta_g}^i\}_{i=1}^n \sim \mathbb{P}_g$ ;
4     compute surrogate loss  $s(\omega^T h_{\mathcal{D}}, \omega^T h_{\mathcal{F}(\theta_g)})$ 
5     | return  $L \leftarrow s(\omega^T h_{\mathcal{D}}, \omega^T h_{\mathcal{F}(\theta_g)})$ ;
6      $(\hat{\omega}, \hat{\theta}_d) \leftarrow (\hat{\omega}, \hat{\theta}_d) - \alpha \nabla_{\omega, \theta_d} L$ ;
7   end
8   compute max-sliced Wasserstein Distance  $\max\text{-}\tilde{W}_2(\hat{\omega}^T h_{\mathcal{D}}, \hat{\omega}^T h_{\mathcal{F}(\theta_g)})$ 
9   | Sample data  $\{\mathcal{D}^i\}_{i=1}^n \sim \mathbb{P}_d$ , generated samples  $\{\mathcal{F}_{\theta_g}^i\}_{i=1}^n \sim \mathbb{P}_g$ ;
10  | sort  $\hat{\omega}^T h_{\mathcal{D}}$  and  $\hat{\omega}^T h_{\mathcal{F}(\theta_g)}$  to obtain permutations  $\pi_{\mathcal{D}}, \pi_{\mathcal{F}}$ ;
11  | return  $L = \sum_i \|\hat{\omega}^T h_{\mathcal{D}_{\pi_{\mathcal{D}}(i)}} - \hat{\omega}^T h_{\mathcal{F}_{\pi_{\mathcal{F}}(i)}(\theta_g)}\|_2^2$ ;
12   $\theta_g \leftarrow \theta_g - \alpha \nabla_{\theta_g} L$ ;
13 end
```

Definition 2 Let Ω be the set of all directions on the unit sphere. Then, the max-sliced Wasserstein-2 distance between distributions μ and ν is defined as:

$$\max\text{-}\tilde{W}_2(\mu, \nu) = \left[\max_{\omega \in \Omega} W_2^2(\mu^\omega, \nu^\omega) \right]^{\frac{1}{2}}. \quad (13)$$

As illustrated in the following claim, it can be shown easily that $\max\text{-}\tilde{W}_2(\cdot, \cdot)$ is a valid distance.

Claim 2 The max-sliced Wasserstein-2 distance defined in Eq. (13) is a well defined distance between distributions.

Proof. See supplementary material. ■

We can also show that the max-sliced Wasserstein distance has polynomial sample complexity:

Claim 3 Consider the family of Gaussian distributions

$$\mathcal{P} = \{\mathcal{N}(a, I) \mid a \in \mathbb{R}^d\}.$$

The max-sliced Wasserstein-2 ($\max\text{-}\tilde{W}_2$) distance is \mathcal{P} -generalizable.

Proof. See the supplementary material. ■

Since it is a valid metric, we can directly use the max-sliced Wasserstein distance for learning distributions.

By definition, the max-sliced Wasserstein distance overcomes the limitation discussed in Sec. 3.2. However, we note that the use of a max-estimator is necessary, which is harder than estimation of a conventional random variable. In the following section, we discuss how the max-sliced Wasserstein distance can be estimated and used in a GAN-like setting.

3.4. max-sliced GAN

In this section, we discuss our approach that uses the max-sliced Wasserstein distance to train a GAN. We also

discuss how we approximate the max-sliced Wasserstein distance in practice. Since we use $\max\text{-}\tilde{W}_2$, we are able to achieve significant savings in terms of the number of projection directions needed as compared to [8].

Intuitively, we want to project data into a space where real samples can easily be differentiated from artificially generated points. To this end, we work with an adversarially learnt feature space, *i.e.*, we use the penultimate layer of a discriminator network. In this feature space, we minimize the max-sliced Wasserstein distance $\max\text{-}\tilde{W}_2$. As will be discussed later in this section, finding the actual max is hard and therefore we resort to approximating it.

Let \mathbb{P}_d again denote the data distribution and let \mathbb{P}_g refer to the induced generator distribution. Further, let the discriminator be represented as $\omega_d^T h(\cdot)$, where ω denotes the weights of a fully connected layer and h represents the feature space we are interested in. Further, let $h_{\mathcal{D}}$ and $h_{\mathcal{F}}$ represent the two empirical distributions in this feature space. Then, we would like to solve

$$\max\text{-}\tilde{W}_2(h_{\mathcal{D}}, h_{\mathcal{F}}) = \max_{\omega \in \Omega} W_2(h_{\mathcal{D}}^\omega, h_{\mathcal{F}}^\omega), \quad (14)$$

where Ω is the set of all normalized directions. There is no easy way in general to solve

$$\omega^* = \operatorname{argmax}_{\omega \in \Omega} W_2(h_{\mathcal{D}}^\omega, h_{\mathcal{F}}^\omega), \quad (15)$$

even if the parameters θ_d of the feature transform h are fixed. This is because computation of the Wasserstein distance $W_2(h_{\mathcal{D}}^\omega, h_{\mathcal{F}}^\omega)$ in the 1-dimensional case requires sorting, *i.e.*, solving of a minimization problem. Hence the program given in Eq. (15) is a saddlepoint objective, for which both maximization and minimization can be solved exactly when assuming the parameters of the other program to be fixed.

	en-es	es-en	en-fr	fr-en	en-de	de-en	en-ru	ru-en	en-zh	zh-en
[6] - NN	79.1	78.1	78.1	78.2	71.3	69.6	37.3	54.3	30.9	21.9
[6] - CSLS	81.7	83.3	82.3	82.1	74.0	72.2	44.0	59.1	32.5	31.4
Max-sliced WGAN - NN	79.6	79.1	78.2	78.5	71.9	69.6	38.4	58.7	34.9	25.1
Max-sliced WGAN - CSLS	82.0	84.1	82.5	82.3	74.8	73.1	44.6	61.7	35.3	31.9

Table 1: Unsupervised word translation. We show the retrieval precision P@1 on 5 pairs of languages on MUSE bilingual dictionaries [6]: English (‘en’), French (‘fr’), German (‘de’), Russian (‘ru’) and Chinese (‘zh’).

If we want to jointly find the parameters θ_d of the feature transform h and the projection direction ω , *i.e.*, if we want to solve

$$\omega^*, \theta_d^* = \operatorname{argmax}_{\omega \in \Omega, \theta_d} W_2(h_{\mathcal{D}}^\omega, h_{\mathcal{F}}^\omega), \quad (16)$$

using gradient descent based methods, we also need to pay attention to bounded-ness of the objective. Using regularization often proves tricky and may require separate tuning for each use case.

To circumvent those difficulties when jointly searching for ω^* and θ_d^* , we use a surrogate function s and write the objective for the discriminator as follows:

$$\hat{\omega}, \hat{\theta}_d = \operatorname{argmax}_{\omega \in \Omega, \theta_d} s(\omega^T h_{\mathcal{D}}, \omega^T h_{\mathcal{F}}). \quad (17)$$

Intuitively, and in spirit similar to $\max\tilde{W}_2$, we want the surrogate function s to transform the data via h into a space where $h_{\mathcal{D}}$ and $h_{\mathcal{F}}$ are easy to differentiate. Moreover, we want ω to be the direction which best separates the transformed real and generated data. A variety of surrogate functions such as the log-loss as specified in Eq. (8), the hinge-loss, or a moment separator with

$$s(\omega^T h_{\mathcal{D}}, \omega^T h_{\mathcal{F}}) = \sum_{x \in \mathcal{D}} \omega^T h_x - \sum_{\hat{x} \in \mathcal{F}} \omega^T h_{\hat{x}} \quad (18)$$

come to mind immediately.

For instance, in case of a log-loss, $\omega^T h$ learns to classify real and fake samples, essentially performing linear logistic regression using ω on a learned feature representation h . If trained to optimality, the two distributions are well separated in the discriminator’s feature space h . An example is given in Fig. 2. The discriminator takes two distributions, shown in Fig. 2a and is trained to classify them. In doing so the discriminator transforms them to the feature space shown in Fig. 2b. In this simple example, we can plot the Wasserstein distance along the different projection directions. This is visualized in Fig. 2c. The discriminator’s final layer can be considered as a projection direction. This direction is very close to the maximizer of the projected Wasserstein distance in the feature space.

Additionally, in this case, ω^* can be approximated with $\hat{\omega}$ – because the discriminator, trained for classification, essentially separates the distributions along $\hat{\omega}$. If we compute the Wasserstein-2 distance for projections onto different angles (as in Fig. 2c), we see that the maximum distance is

achieved close to the projection direction from the discriminator, *i.e.*, $\hat{\omega}$. We next assess: ‘how close?’

While log-loss and all other functions seem intuitive, we provide for the special case of the moment separator given in Eq. (18) and an identity transform h the maximal sub-optimality in terms of the max-sliced Wasserstein distance:

Claim 4 For the surrogate function s given in Eq. (18), h the identity, and $\hat{\omega}$ computed as specified in Eq. (17), we obtain

$$\alpha(\mathcal{D}, \mathcal{F}) \leq W_2^2(\mathcal{D}^{\hat{\omega}}, \mathcal{F}^{\hat{\omega}}) \leq V^* = \max\tilde{W}_2(\mathcal{D}, \mathcal{F})^2,$$

for a lower bound $\alpha(\mathcal{D}, \mathcal{F}) = \|m\|_2^2$, where $m = \sum_i \mathcal{D}_i - \sum_i \mathcal{F}_i$ is the difference of dataset means.

Proof. See the supplementary material. ■

To summarize, training the discriminator for classification provides a rich feature space which can be utilized for faster training. We note that the discriminator might be trained to obtain such features in a more explicit manner, but we leave this to future research.

3.5. max-sliced GAN Algorithm

We summarize the resulting training process in Alg. 1. It proceeds as follows: In every iteration, we draw a set of samples \mathcal{D} and \mathcal{F} from the true and fake distributions. We optimize the parameters θ_d and ω of the feature transform h for k iterations (k is a hyper-parameter) to maximize a surrogate loss function $s(\omega^T h_{\mathcal{D}}, \omega^T h_{\mathcal{F}})$. Then we compute the Wasserstein-2 distance between the output distributions of the discriminator, *i.e.*, $W_2(\hat{\omega}^T h_{\mathcal{D}}, \hat{\omega}^T h_{\mathcal{F}})$. The generator is trained to minimize this distance. In our experiments, we choose h to be the binary classification loss.

4. Experiments

In this section, we present results to demonstrate the effectiveness of the max-sliced Wasserstein distance and the computational benefits it offers over the sliced Wasserstein distance. We show quantitative results on unpaired word translation [6], and qualitative and quantitative results on image generation tasks using the CelebA-HQ [16] and the LSUN Bedrooms [36] datasets.



Figure 3: Generated samples (256×256) from CelebA-HQ.

4.1. Word Translation without Parallel Data

We evaluate the effectiveness of the max-sliced GAN on unsupervised word translation tasks, *i.e.*, without paired/parallel data [6]. This allows us to quantitatively compare different methods.

The setting of this experiment is as follows. We are given embeddings of words from two languages, say $X, Y \in \mathbb{R}^d$. We want to learn an orthogonal transformation W^* that maps the source embeddings X to Y , *i.e.*:

$$W^* = \underset{W \in \mathbb{R}^{d \times d}, \text{orthogonal}}{\operatorname{argmin}} \|WX - Y\|_F. \quad (19)$$

The current state-of-the-art [6] employs a GAN-like [11] adversary to learn the transformation. Therefore, the transformation is learned by minimizing the Jensen-Shannon divergence between WX and Y . We instead minimize the max-sliced Wasserstein distance to learn W .

We follow the training method and evaluation in [6] and

report the word translation precision by computing the retrieval precision@ k for $k = 1$ on the MUSE bilingual dictionaries [6]. During testing, 1,500 queries are tested and 200k words of the target language are taken into account. We compare our method with [6] and present results for 5 pairs of languages in Tab. 1. In Tab. 1 ‘NN’ represents use of nearest neighbors to build the dictionary after training the transformation W , and ‘CSLS’ stands for use of cross-domain similarity local scaling [6]. Our method with CSLS outperforms the baseline in all tested language pairs. This demonstrates the competitiveness of our method with current established GAN frameworks.

4.2. Image Generation

In this section, we present results on the task of image generation. Using the max-sliced Wasserstein distance, we train a GAN on the CelebA [16] and LSUN Bedrooms [36] datasets for images of resolution 256×256 . We compare with the sliced Wasserstein GAN [8].

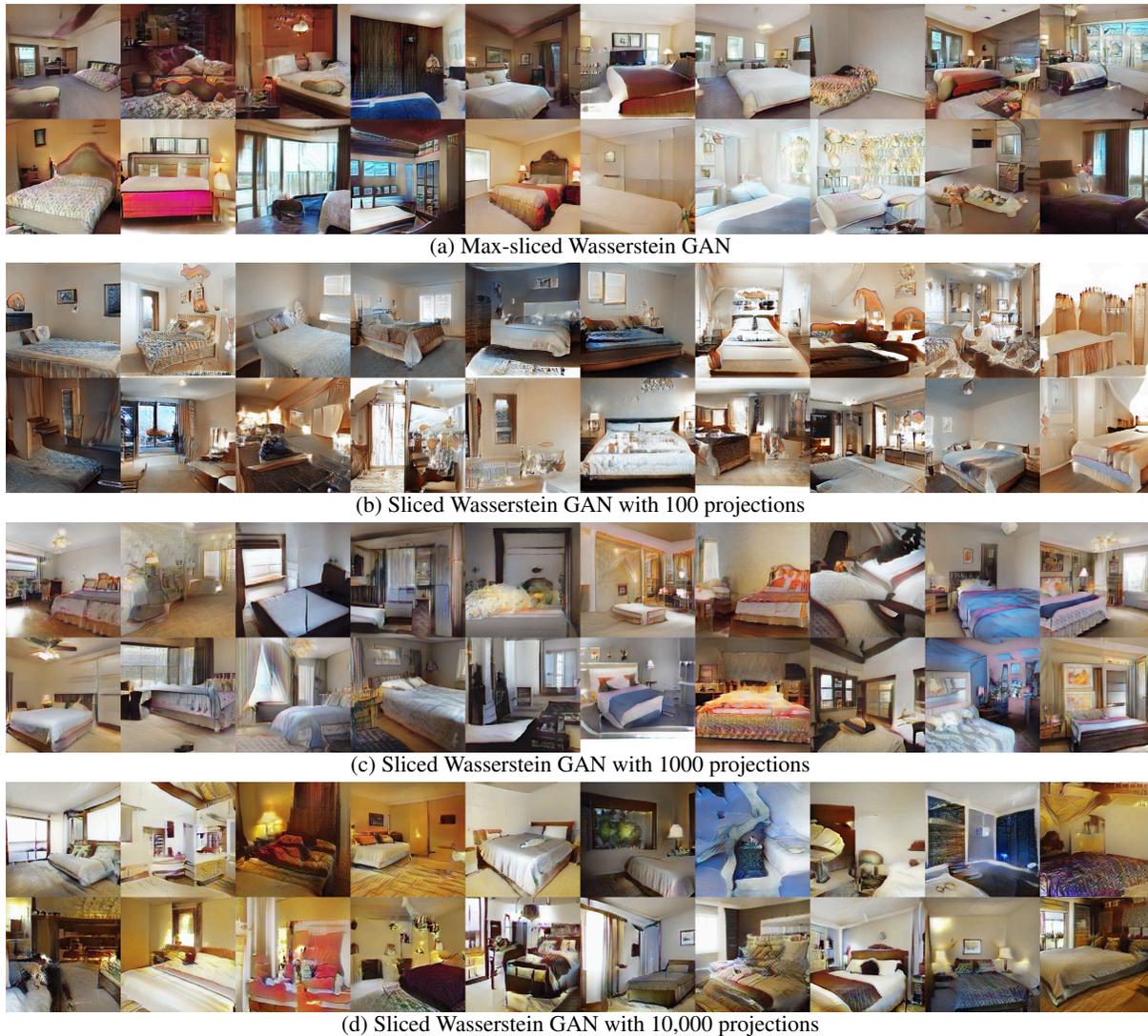


Figure 4: Generated samples (256×256) from LSUN Bedrooms.

Samples generated by each trained model are presented in Fig. 3 and Fig. 4. The results of the max-sliced Wasserstein GAN are shown in Fig. 3a and Fig. 4a. We train the sliced Wasserstein GAN with 100, 1000, and 10000 random projections. Results of each of these are respectively shown in Fig. 3b, Fig. 3c, and Fig. 3d for CelebA-HQ, and in Fig. 4b, Fig. 4c, and Fig. 4d for LSUN. The max-sliced Wasserstein GAN using just one projection direction is able to produce results which are either comparable or better than the sliced Wasserstein GAN even when using 10000 projections. This significantly reduces the computational complexity and also the memory footprint of the model.

We used a simple extension of the popular DCGAN architecture for the generator and discriminator. Two extra strided (transpose) convolutional layers are added to the generator and the discriminator to scale to 256×256 . We do not use any special normalization/ initialization to train the models. Specific details are given in the supplementary.

5. Conclusion

In this paper, we analyzed the Wasserstein and sliced Wasserstein distance and developed a simple yet effective training strategy for generative adversarial nets based on the max-sliced Wasserstein distance. We showed that this distance enjoys a better sample complexity than the Wasserstein distance, and a better projection complexity than the sliced Wasserstein distance. We developed a method to approximate it using a surrogate loss, and also analyzed the approximation error for one such surrogate. Empirically, we showed that the discussed approach is able to learn high dimensional distributions. The method requires orders of magnitude fewer projection directions than the sliced Wasserstein GAN even though both work in a similar distance space.

Acknowledgments: This work is supported in part by NSF under Grant No. 1718221, Samsung, and 3M. We thank NVIDIA for providing GPUs used for this work.

References

- [1] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017. 2
- [2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. In *ICML*, 2017. 1, 2
- [3] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *ICML*, 2017. 1
- [4] D. Berthelot, T. Schumm, and L. Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017. 1
- [5] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 2015. 2
- [6] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jegou. Word translation without parallel data. In *ICLR*, 2018. 2, 6, 7
- [7] R. W. A. Cully, H. J. Chang, and Y. Demiris. Magan: Margin adaptation for generative adversarial networks. *arXiv preprint arXiv:1704.03817*, 2017. 1
- [8] I. Deshpande, Z. Zhang, and A. Schwing. Generative modeling using the sliced wasserstein distance. In *CVPR*, 2018. 1, 2, 5, 7
- [9] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. In *ICLR*, 2017. 1
- [10] S. Feizi, C. Suh, F. Xia, and D. Tse. Understanding gans: the lqg setting. *arXiv preprint arXiv:1710.10793*, 2017. 1
- [11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 1, 7
- [12] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. In *NIPS*, 2017. 1, 2
- [13] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017. 1
- [14] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal Unsupervised Image-to-Image Translation. In *Proc. ECCV*, 2018. 1
- [15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 1
- [16] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2017. 1, 2, 6, 7
- [17] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1
- [18] S. Kolouri, C. E. Martin, and G. K. Rohde. Sliced-wasserstein autoencoder: An embarrassingly simple generative model. *arXiv preprint arXiv:1804.01947*, 2018. 2
- [19] S. Kolouri, S. R. Park, and G. K. Rohde. The radon cumulative distribution transform and its application to image classification. *IEEE transactions on image processing*, 2016. 2
- [20] S. Kolouri, G. K. Rohde, and H. Hoffman. Sliced wasserstein distance for learning gaussian mixture models. In *CVPR*, 2018. 1, 2
- [21] H. Y. Lee, H. Y. Tseng, J. B. Huang, M. K. Singh, and M. H. Yang. Diverse image-to-image translation via disentangled representation. In *Proc. ECCV*, 2018. 1
- [22] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos. Mmd gan: Towards deeper understanding of moment matching network. In *NIPS*, 2017. 1
- [23] Z. Lin, A. Khetan, G. Fanti, and S. Oh. Pacgan: The power of two samples in generative adversarial networks. In *NIPS*, 2018. 1
- [24] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised Image-to-Image Translation Networks. In *Proc. NIPS*, 2017. 1
- [25] L. Mescheder, A. Geiger, and S. Nowozin. Which training methods for gans do actually converge? In *ICML*, 2018. 1
- [26] Y. Mroueh and T. Sercu. Fisher gan. In *NIPS*, 2017. 1
- [27] Y. Mroueh, T. Sercu, and V. Goel. Mcgan: Mean and covariance feature matching gan. *arXiv preprint arXiv:1702.08398*, 2017. 1
- [28] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 1
- [29] A. Royer, K. Bousmalis, S. Gouws, F. Bertsch, I. Moressi, F. Cole, and K. Murphy. Xgan: Unsupervised image-to-image translation for many-to-many mappings. In *arXiv:1711.05139*, 2017. 1
- [30] T. Salimans, H. Zhang, A. Radford, and D. Metaxas. Improving gans using optimal transport. In *ICLR*, 2018. 1
- [31] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, 2017. 1
- [32] C. Villani. *Optimal transport: old and new*. Springer Science & Business Media, 2008. 2
- [33] J. Weed and F. Bach. Sharp asymptotic and finite-sample rates of convergence of empirical measures in wasserstein distance. *arXiv preprint arXiv:1707.00087*, 2017. 2
- [34] J. Wu, Z. Huang, W. Li, J. Thoma, and L. Van Gool. Sliced wasserstein generative models. *arXiv preprint arXiv:1706.02631*, 2017. 1, 2
- [35] Z. Yi, H. Zhang, P. Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proc. ICCV*, 2017. 1
- [36] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 2, 6, 7
- [37] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 1
- [38] J. Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *Proc. NIPS*, 2017. 1