Distributional Encoder: A Framework for Learning the Wasserstein space of Probability Measures

Anonymous Authors[1]

Abstract

Understanding the space of probability measures on a metric space equipped with a Wasserstein distance is one of the fundamental questions in mathematical analysis. The Wasserstein metric has received a lot of attention in the machine learning community especially for its principled way of comparing distributions. In this work, we present the Distributional Encoder, a neural network framework that learns to encode samples from probability measures into a low dimensional Euclidean space. We show that using this space we can compute Wasserstein distances on seen and unseen measures, compute barycenters, as well as generate distributions from interpolations in this space. We also introduce a dynamic optimal transport variant of our method, based on conditional GANs, that allows learning of the distributional encoding, as well as generating from it, without the need for precomputed Wasserstein distances. We show that our framework can learn meaningful embeddings that can be used for downstream tasks.

1. Introduction

The Wasserstein distance is a distance function between probability measures on a metric space $\mathcal{X}$. It is a natural way to compare the probability distributions of two variables X and Y, where one variable is derived from the other by small, non-uniform perturbations, while strongly reflecting the metric of the underlying space $\mathcal{X}$. It can also be used to compare discrete distributions. The Wasserstein distance enjoys a number of useful properties, which likely contributes to its wide-spread interest amongst mathematicians and computer scientists (Bobkov & Ledoux, 2019; Ambrosio et al., 2005; Bigot et al., 2013; Canas & Rosasco, 2012; del Barrio et al., 1999; Givens & Shortt, 1984; Villani, 2003; 2008; Arjovsky et al., 2017). However, despite it's broad use, the Wasserstein distance has several problems. For one, it is computationally expensive. Second, the Wasserstein distance is not Hadamard differentiable, which can present serious challenges when trying to use it in machine learning. Third, the distance is not robust. To alleviate these problems, one can use various regularized entropies to compute an approximation of this Wasserstein distance. Such an approach is more tractable and also enjoys several nice properties (Altschuler et al., 2018; Cuturi, 2013; Peyré & Cuturi, 2020).

Here, we are interested in learning about the Wasserstein space of order p, i.e. an infinite dimensional space of all probability measures with up to p-th order finite moments on a complete and separable metric space $\mathcal{X}$. More specifically, we asked, (**1**) can we propose a neural network that correctly computes the Wasserstein distance between 2 measures, even if both of them are not in our training examples? (**2**) What properties of the measures does such a network learn?

In this article, we present the **Distributional Encoder**, a neural network framework for encoding probability measures in a low dimensional Euclidean space. To do this we use permutation invariant functions, to encode samples from distributions into a latent space that we constrain with a Wasserstein loss.

We show that we can learn a space where the Euclidean distance between the encoded samples matches their true Wasserstein distances. We do extensive experiments to show that the space learns various properties of a Wasserstein space. Finally, we show a downstream application of such a trained encoder on a point cloud classification task.

While our approach uses precomputed Wasserstein distances, we also propose an alternative way to train our model that is free of such (slow) distance function computations. This variant is based on dynamic optimal transport (OT) and uses a conditional GAN to define a loss on our learned encodings. We show that the Euclidean distances learnt in this space correlate strongly with the actual Wasserstein distances without requiring explicit computation of Wasser-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

stein distances. Another advantage of this approach is that we can now also easily sample from our encoding space, such as generating barycenters from interpolations between distributions.

2. Related Work

There has been a lot of work in understanding the space of Gaussian processes (Mallasto & Feragen, 2017; Takatsu, 2011) but our work is more similar to (Courty et al., 2017; Frogner et al., 2019), which attempts to understand Wasserstein spaces with neural networks. Like (Courty et al., 2017), we use a Siamese network to compare and contrast various densities but the questions we address in this article and our motivations are different than that of (Courty et al., 2017). Furthermore, we try to approximate the Wasserstein space by learning a mapping from the space to a low dimensional Euclidean space, unlike (Frogner et al., 2019), where they learn a mapping from an Euclidean space to the Wasserstein space.

Wasserstein metrics have also an important role in the generative networks. There has been a lot of work connecting Generative Adversarial Networks (GANs) and Optimal Transport (OT) for example in (Arjovsky et al., 2017; Salimans et al., 2018; Tanaka, 2019; Avraham et al., 2019) where the authors have used Wasserstein metrics to match the target and source distributions by the discriminator networks. Further works have shown how OT can improve the stability of a GAN training and robustness of the GAN (Balaji et al., 2020; Sanjabi et al., 2018; Cao et al., 2020; Chu et al., 2020). Recent works have also shown that GANs can effectively compute the transport maps (when they exist) (Seguy et al., 2018; Jacob et al., 2019). The main difference here is that we use the W_1 metric and we do not assume the existence of the Monge map since we do not explicitly compute the transport plan.

Recent works on normalizing flows (Tong et al., 2020; Finlay et al., 2020) have also used optimal transport plans as a way to move from target to source distributions. These maps are desirable since they can be naturally interpreted as minimizing the kinetic energy.

Even though the Wasserstein metric is a natural way to compare density distributions, it is not often used in practice for real world datasets like dense 3D point clouds due to its high computational costs. Yet many of the recent works (Fan et al., 2017; Achlioptas et al., 2018; Li et al., 2018) use W_1 metric, also known as Earth Mover's Distance (EMD), as evaluation distance or loss function to identify subtle differences between generated and ground truth point clouds.

In (Kawano et al., 2020) the authors used PointNet (Qi et al., 2016) to embed point clouds with Wasserstein distances. Unlike their work, we focus on embedding probability distribution measures in general and also provide a Wasserstein distance function free approach, which greatly increases performance. Furthermore, we use the DeepSets encoding architecture (Zaheer et al., 2017) which is generally applicable for any distribution, not just point clouds.

3. Wasserstein Metrics and Optimal Transport

There is a vast body of literature on Wasserstein metrics and their uses in various fields of Machine Learning. For the convenience of the reader, we summarize some key well known results that are used in the article.

Let $\mathcal{X}$ be a complete and separable metric space. For simplicity, we take $\mathcal{X}$ to be $\mathbb{R}^n$ or a compact subset of $\mathbb{R}^n$. Let $\mathbb{P}(\mathcal{X})$ be the space of all probability measures on $\mathcal{X}$. One can endow the space $\mathbb{P}(\mathcal{X})$ with a family of metrics called the Wasserstein metrics W_p.

$$W_p(\mu,\nu) = \inf_{\substack{X\sim\mu \\ Y\sim\nu}} \mathbb{E}(|X-Y|^p)^{1/p}, \quad p \geq 1 \tag{1}$$

We use the notations $W_p(X,Y)$ and $W_p(\mu,\nu)$ interchangeably whenever $X \sim \mu$ and $Y \sim \nu$. We also assume that $\mathbb{E}(|X|^p)$ (and $\mathbb{E}(|Y|^p)$) is finite. The optimization problem defining the distance (Equation 1) is popularly known as optimal transport or the Monge–Kantorovich problem. The Kantorovich formulation (Kantorovich, 2006) of the transportation problem is:

$$\mathrm{OT}(\mu,\nu) := \min_{\pi\in\Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} c(x,y)d\pi(x,y) \tag{2}$$

where $c(\cdot,\cdot) : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ is a cost function and the set of couplings $\Pi(\mu,\nu)$ consists of joint probability distributions over the product space $\mathcal{X}\times\mathcal{X}$ with marginals μ and ν,

$$\Pi(\mu,\nu) := \{\pi \in \mathbb{P}(\mathcal{X}\times\mathcal{X}) : P_{1\#}\pi = \mu, P_{2\#}\pi = \nu\}. \tag{3}$$

where P_i are the projection maps from $\mathcal{X}\times\mathcal{X}$ to ith factor of $\mathcal{X}$ and $P_{i\#}\pi$ is the pushforward of the measure π onto $\mathcal{X}$. The cost function generally reflects the metric of the space $\mathcal{X}$ and in our case is just $c(x,y) := ||x-y||_p^{1/p}$ for some $p \geq 1$. For the Monge formulation of the transport problem, we refer the reader to (Villani, 2003; 2008).

Most of the following properties regarding the space $\mathbb{P}(\mathcal{X})$ and W_p are well-known but we summarize them for the convenience of the reader (Santambrogio, 2015; Panaretos & Zemel, 2019).

Theorem 3.1. 1. $\mathbb{P}(\mathcal{X})$ equipped with W_p is a complete and separable metric space (Polish space).

2. If X and Y are degenerate at $x, y \in \mathcal{X}$, then $W_p(X,Y) = ||x-y||_p$.

3. (Scaling property) For any $a \in \mathbb{R}$, $W_p(aX, aY) = |a| W_p(X, Y)$.
4. (Translation invariance property) For any $\alpha \in \mathcal{X}$, $W_p(X + \alpha, Y + \alpha) = W_p(X, Y)$
5. $\mathbb{P}(\mathcal{X})$ is a geodesic metric space, when $\mathcal{X} \subset \mathbb{R}^n$ is a convex and compact set.

Proof. See Section 2 in (Panaretos & Zemel, 2019). The last fact is proved in (Santambrogio, 2015; Villani, 2008). □

However let us mention some caveats to the last fact in the earlier theorem. The geodesic connecting the measures μ and ν are not unique. The uniqueness of Kantorovich potentials (dual to the Kantorovich transport problem) (Villani, 2008) can not generally be guaranteed except when $p = 2$ and some additional constraints about the support of the measures (Santambrogio, 2015).

Theorem 3.2. (Topology generated by W_p)

1. If $\mathcal{X} \subset \mathbb{R}^n$ is compact and $p \in [1, \infty)$, in the space $\mathbb{P}(\mathcal{X})$, we have $\mu_k \to \mu$ iff $W_p(\mu_k, \mu) \to 0$.
2. If $\mathcal{X} = \mathbb{R}^n$, then $W_p(\mu_k, \mu) \to 0$ iff $\mu_k \to \mu$ and $\int |x|^p d\mu_k \to \int |x|^p d\mu$

Proof. See proofs associated with Theorem 5.10 and Theorem 5.11 in (Santambrogio, 2015). □

Furthermore the geodesics in $\mathbb{P}(\mathcal{X})$ are well-understood. Given probability measures μ and ν, one can define a path $\gamma(t)$ joining μ and ν, i.e. for $0 \leq t \leq 1$, $\gamma(0) = \mu$, $\gamma(1) = \nu$ and $\gamma(t) \in \mathbb{P}(\mathcal{X})\ \forall\ t$. Theorem 3.1 (5) implies that there is a path $\gamma(t)$, whose length between 0 and 1 is $W_p(\mu, \nu)$. In fact,

$$W_p(\gamma(t), \gamma(s)) = |s - t| W_p(\mu, \nu)\ \forall s, t \in [0, 1] \tag{4}$$

Such curves $\gamma(t)$ are called constant speed geodesics. In fact one has an explicit formula for such a curve $\gamma(t)$ (Ambrosio et al., 2005).

$$\gamma(t) = ((1 - t)P_1 + tP_2)_{\#}\pi \tag{5}$$

where P_i are defined as in Equation 3 and π is the optimal transport plan minimizing Equation 1. When π is given by a transport map T á la Monge (Villani, 2003; 2008), $\gamma(t)$ can be succinctly written as

$$\gamma(t) = ((1 - t)Id + tT)_{\#}\mu \tag{6}$$

This above formula was first introduced (McCann, 1997) and is called the displacement interpolation. Moreover γ satisfies the following equation,

$$\nabla_{\dot{\gamma}}\dot{\gamma} = 0 \tag{7}$$

which can be used a regularizer for an appropriate loss function to enforce the geodesic condition.

The measures μ and ν are rarely known in practice. Instead, one has access to finite samples $\{x_i\} = X \sim \mu$ and $\{y_j\} = Y \sim \nu$. We then construct discrete measures $\mu := \sum_{i=1}^{n} a_i \delta_{x_i}$ and $\nu := \sum_{j=1}^{m} b_j \delta_{y_j}$ where a, b are vectors in the probability simplex, and the pairwise costs can be compactly represented as an $n \times m$ matrix C, i. e., $c_{ij} := c(x_i, y_j)$ where c is the metric of the underlying space $\mathcal{X}$.

$$W_p(\mu, \nu) = \min_{P \in U(\mu,\nu)} \sum_{i,j} P_{ij} C_{ij} \tag{8}$$

where $U(\mu, \nu) = \{P \in \mathbb{R}_+^{n \times m} : P\mathbf{1}_m = a, P^t\mathbf{1}_n = b\}$. Since the marginals here are fixed to be the laws of X and Y, the problem is to find a copula (Sklar, 1959) that couples X and Y together as "tightly" as possible in an L^p-sense, on average. However solving the above problem scales cubically on the sample sizes and is extremely difficult in practice. Adding an entropy regularization, leads to a problem that can be solved much more efficiently (Altschuler et al., 2018; Cuturi, 2013; Peyré & Cuturi, 2020). The entropy regularized version of this problem reads:

$$SD_p^{\lambda}(\mu, \nu) := \min_{P \in U(\mu,\nu)} \sum_{i,j} P_{ij} C_{ij} + \frac{1}{\lambda} \sum_{i,j} P_{ij} \log P_{ij} \tag{9}$$

Due to the strong convexity introduced by the regularizer, the above problem now has a unique solution and can be efficiently solved by the Sinkhorn algorithm. In this article, we use the Sinkhorn distance SD_p^{λ} and their computation, as in (Cuturi, 2013). The Sinkhorn distance however is not a true metric (Cuturi, 2013) and fails to satisfy $SD_p^{\lambda}(X, X) = 0$. Moreover, the Sinkhorn distance requires discretizing the space, which alters the true metric.

4. Neural Networks to learn the Wasserstein Space

In this section, we will describe two approaches to learn the Wasserstein space. The first approach is a direct approach to train a Neural Network that learns an encoding such that the Euclidean distance between encoded samples is equal to the Sinkhorn distance between the samples. The second approach is to train a CGAN where the condition is trained end-to-end with the GAN along with a kinetic energy regularization that ensures a dynamic optimal transport between encoded distributions.

4.1. Learning Sinkhorn Distances with DeepSets

We draw random samples with replacement of size N from various distributions in $\mathbb{P}(\mathcal{X})$. We use the DeepSets architecture (Zaheer et al., 2017) to encode this set of N elements as

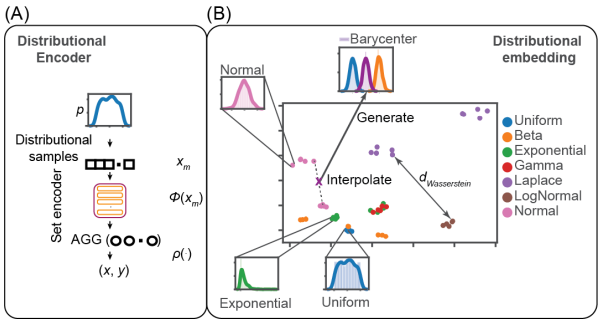

Figure 1. (**A**) Diagram representation of the **Distributional Encoder**. Our framework uses permutation invariant networks to map samples from distributions to a low dimensional Euclidean space. (**B**) Low-dimensional embedding of encoded distributions with interpolations representing barycenters between endpoints. The Distributional Encoder enables encoding of as well as generating from barycenters.

we want an encoding that is invariant of the permutations of the samples. More precisely, if $X \sim \mu$ and $Y \sim \nu$, ($\mu = \nu$ is allowed, but X and Y are drawn independently) and by abuse of notation, we denote the set of samples drawn from μ as X (similarly Y), we train the encoder H_θ such that,

$$||H_\theta(X) - H_\theta(Y)|| = SD_p^\lambda(\mu, \nu) \tag{10}$$

Thus, the loss function becomes,

$$L_{wass} = \frac{1}{\binom{m}{2}} \sum (||H_\theta(X) - H_\theta(Y)|| - SD_p^\lambda(\mu, \nu))^2 \tag{11}$$

where m is the size of the mini-batch and we pick 2 sets at random from the mini-batch to compare distances. Note that we do not expect the samples to be distinct thus they are not technically sets but rather a collection and we do not make that distinction anywhere in the paper. All we need is a representation that is invariant under permutation. However, during this process, we make sure that at no point both sets of samples are the same to circumvent the issue that the Sinkhorn distance is not a true metric. One can think of our network as a Siamese Network (Koch et al., 2015) with a DeepSets backbone which allows us to compare and contrast samples drawn from same or different distributions. Our work can be thought as *next-generation functional data analysis* (Wang et al., 2015) (Section 6). More details about the network architecture can be found in the supplementary materials.

Regularizers to ensure the Scaling and the Translation Invariance Property

If $X' = X + \alpha$, i.e. by abuse of notation, X' is the set of samples X after translation α (similarly Y' is a set of samples Y after translation α). To ensure the properties of W_p are reflected in our computed Euclidean distance, we demand that,

1. $||H_\theta(X') - H_\theta(Y')|| = ||H_\theta(X) - H_\theta(Y)||$
2. $||H_\theta(aX) - H_\theta(aY)|| = |a|||H_\theta(X) - H_\theta(Y)||$.

These constraints comprise the loss function

$$\mathcal{L} := L_{Wass} + \frac{1}{\binom{m}{2}} \sum ((||H_\theta(X') - H_\theta(Y')|| - ||H_\theta(X) - H_\theta(Y)||)^2 + (||H_\theta(aX) - H_\theta(aY)|| - |a|||H_\theta(X) - H_\theta(Y)||))^2$$

4.2. Distance function free Wasserstein encodings via cGAN and Dynamical OT

The above approach requires computing Wasserstein distances, which are expensive. Various entropic regularizations have been proposed to fix this issue. However, these methods are still slow and scale poorly with dimension. To overcome these issues, we propose an approach for training the Distributional Encoder without the need for Sinkhorn algorithm computation. To do this, we use a conditional GAN (Mirza & Osindero, 2014) combined with a dynamic OT loss to ensure that the latent space learns an embedding that satisfies the properties of the Wasserstein space. In addition, a GAN will allow us to generate distributions from the latent space.

In our approach, the GAN conditioning y, given by $H(X)$, is obtained by passing the set X sampled from a target distribution through a permutation invariant function (e.g. DeepSets) H_θ, which we learn. Using a function to learn y is not new and has been used in self-supervised tasks (Tran et al., 2019) and in learning conditions in point clouds (Li et al., 2018). We train the permutation invariant function together with the generator in an end-to-end manner, and as such the model learns an encoding that is maximally informative as a condition to the generator. Thus, this framework allows us to encode a distribution and generate samples from it. The conditioning is appended to the inputs of both the generator G and the discriminator D. The objective function for the above training scheme reads:

$$\min_{G,H} \max_{D,H} V(D, G, H) = \mathbb{E}_{x \in X \sim p_{\text{data}}(x)}[logD(x|H(X))] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z|H(X))))] \tag{12}$$

where x is an element from the sampled set X, z is a randomly sampled noise vector and $H(X)$ is the learned condition. In this min-max formulation, the training of G and D is done alternately. The encoder H is updated with both G and D but the gradients for H are allowed to flow only

through the respective side's conditioning. During the training, the generator tries to generate samples conditioned on the distributional encoding and the discriminator tries to classify the samples as real or fake.

Dynamic Optimal Transport based regularization

While the above described model encodes distributions in a latent space, there are no specific guarantees on the properties of this space, such as on the distances between encoded distributions. To provide Wasserstein properties to this space, similar to our original approach but without using specific distance functions, we make use of a dynamic optimal transport regularization. This regularization is based on a kinetic energy loss, similar to (Finlay et al., 2020; Tong et al., 2020). In these works, that are based on Continuous Normalizing Flows, the authors implement a kinetic energy loss motivated by the Benamou-Brenier formulation of the OT problem. This energy is given by the integral of a vector field which effectively measures path lengths, and thus minimizing it provides an optimal transport. In our approach, we do not have the vector field, rather the trajectory itself, where each z value maps to a trajectory. Thus, we do not need to integrate but instead sample random values of t (thought of random points on the trajectory where the initial point is at time $t = 0$ and the end of the trajectory is at time $t = 1$), compute the derivative of the path at those random values of t, and then minimize the squared norm. We reason that with sufficient samples of paths and random t values, we achieve the same effect as in the previously mentioned papers, i.e. minimizing the squared norm of the sum of the derivatives along the path.

Specifically, we take the following approach:

During training, we randomly sample N pairs of distributions from each minibatch. For each pair (i, j), we randomly sample the noise vector z which defines a trajectory in the generated space. We define a corresponding trajectory in the embedding space whose end points (and starting points resp.) are given by the embeddings of the two distributions say H_i (and H_j resp.). We randomly sample a $t \in (0, 1)$ and find a point along the trajectory using a convex combination of these embeddings given by

$$H_t^{ij} = t * H_i + (1 - t) * H_j \tag{13}$$

The generated output $g_{z,t}^{ij}$ from this interpolated encoding is given by

$$g_{z,t}^{ij} = G(z|H_t^{ij}) \tag{14}$$

We calculate the total OT loss by summing the norm of the gradient at such points for all trajectories as follows:

$$L_{OT} = \sum_{(i,j)} \sum_{z} \sum_{t} ||g_{z,t+\epsilon}^{ij} - g_{z,t}^{ij}||/\epsilon \tag{15}$$

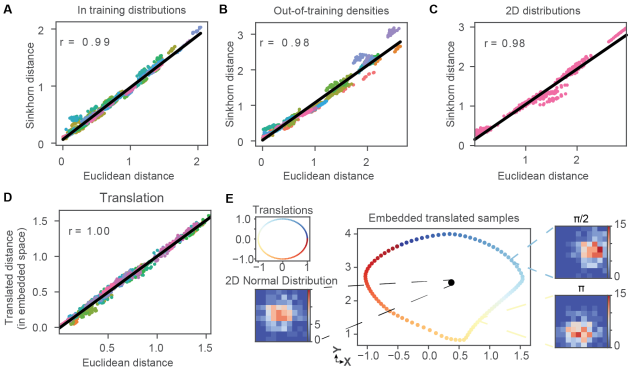

Figure 2. The Distributional Encoder trained with the Sinkhorn loss learns Wasserstein distances: (**A**–**C**) Pearson's r correlation coefficient for association between embedded and Sinkhorn distances (color code in the Supplementary Materials). (**D**) Correlation after translations. (**E**) Samples from a multivariate normal distribution translated around a circular path.

Note that $||g_{z,t+\epsilon}^{ij} - g_{z,t}^{ij}||/\epsilon$ can be thought as estimating the norm of the time derivative of the trajectory. Thus, minimizing this loss will enforce the generator to generate trajectories between two distributions with minimal total path length in the generated space, which ensures dynamic optimal transport. The OT regularized objective function reads:

$$\min_{G,H} \max_{D,H} V(D, G, H) = \mathbb{E}_{x \in X \sim p_{\text{data}}(x)}[logD(x|H(X))] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z|H(X))))] + \lambda L_{OT} \tag{16}$$

With this regularization the network learns an embedding space that reflects Wasserstein distances between distributions.

5. Experiments

The experiments are categorized into 2 parts: toy experiments and point cloud experiments. Detailed ablation studies for the experiments can be found in the Supplementary materials. In all the experiments the entropic regularization in Equation 9, $\lambda = 10$.

5.1. Toy experiments

In this section we will describe toy examples to understand the behavior of the Distributional Encoder and the interesting properties of the space it learns. Our datasets are the following: **(1)** Random samples of size 500 drawn independently about 50 times from uniform, Normal, Beta, Gamma, Exponential, Laplace, Log Normal and mixtures of Gaussian distributions with varying parameters. **(2)** Random samples of size 300 drawn independently about 100 times from $2D$ Normal distributions with various μ, Σ. We

use the W_1 metric for this experiment and the output dimension of our model is 2. Fig 1 (**B**) shows the embedding our datasets by our model. In Fig 2, we show how well the neural network approximates the Sinkhorn distances from samples drawn from our test densities.

Generalizing to out-of-sample-densities

We also show that our model can generalize well to densities that are out of our training set. These densities are primarily constructed from the training densities but by changing the parameters (Fig 2 **B,C**). But even more interestingly, we found that our model can correctly measure the distance between 2 Dirac measures and distance between 2 Binomial densities, even though we did not use any discrete measures in our training.

Translating and Scaling samples

Given 2 samples $X \sim \mu, Y \sim \nu$, we can translate them around by a random vector a, to create new samples $X' := X+a, Y' := Y+a$, under property 4 in Theorem 3.1, $H_\theta(X), H_\theta(Y), H_\theta(X'), H_\theta(Y')$ would form a parallelogram. Fig 2(**D**) shows the exact relationship between the distances of encoded translated samples and the encoded samples. Furthermore, we took samples from a $2D$ Normal Distribution $N(\mu, \Sigma)$ and rotated it around by using a circle, i.e. created new samples via $X' := X + (\cos(\phi), \sin(\phi))$ and we found that the encoded translated samples also formed a circular pattern around the original encoded sample. Moreover, we also get a correlation score of .99 between the scaled the Euclidean distance between the encoded samples X and Y, and the encodings of the scaled samples αX and αY. Thus our simple examples show that our metric preserves the translation invariance property, scaling property, and some geometry of the space (Figure 2E).

Learning statistical properties of the measures

Surprisingly, for encoded $1D$-distributions, we found a strong correlation between means and variances of the distributions and the x-coordinate and y-coordinate, respectively, of the encoded points (Fig 3 **A,B**). This explains why the encoded Dirac distribution at 0 and Normal distribution with mean 0 and standard deviation σ has x-coordinates close to each other. Perhaps, it is unsurprising that the Distributional Encoder learns to represent the moments of the encoded distributions. However, we did not find meaningful encodings of higher moments beyond the variance. In future work we will explore learning disentangled representations where we enforce the dimensions to correspond to (higher) distributional moments.

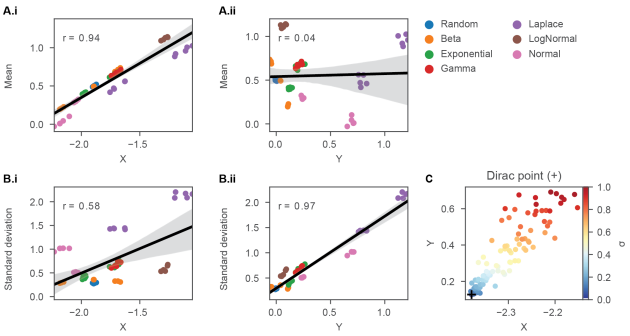

Figure 3. Properties of the Distributional Encoder trained with the Sinkhorn loss: Person's r comparing embedding axes to means (**A (i, ii)**) and standard deviations (**B (i, ii)**). (**C**) Convergence of samples from Normal Distributions with a fixed mean and standard deviations converging to 0, to the Dirac distribution encoding.

Respecting the topology of the space

We know that the Dirac delta measure is the limit of Gaussian measures under the weak convergence of measures. Choosing samples drawn from $N(0, 1/n)$ we can see that our encoded points converge to the point encoded by the Dirac measure (Fig 3**C**). This gives us an empirical evidence that our neural network may be continuous with respect to the Wasserstein metric.

Wasserstein barycenters

Given two densities μ_1, μ_2, if $\hat{\mu}$ is their Wasserstein barycenter (Anderes et al., 2015; Agueh & Carlier, 2011; Zemel & Panaretos, 2017; Karcher, 1977; 2014), our aim is to show that $H_\theta(\hat{\mu})$ can be approximated by the midpoint of the line joining $H_\theta(\mu_1)$ and $H_\theta(\mu_2)$. Fig 4**D** shows the following examples of this claim: **1)** Samples drawn from $N(0, .1)$ and $N(1, .1)$. **2)** Dirac at 0 and 1. **3)** Uniform distribution in $[0, .1]$ and in $[.8, .1]$. We also note that the

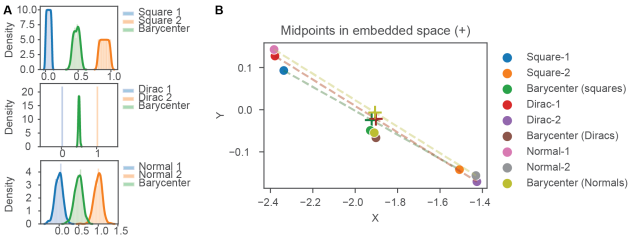

Figure 4. Encoding barycenters with the Distributional Encoder trained with the Sinkhorn loss: **(A)** Barycenters of distributions, **(B)** Midpoints drawn between lines connecting the encoded samples (right).

measures used above were not observed during training. Finally observe that the figure also shows the correlation between x-coordinates and means of the chosen measures. Finally, the experiment also show that we can approximate

the Wasserstein geodesic by straight lines in our encoded space.

cGAN and dynamical OT

We also train the cGAN framework to encode the $1D$ distributions as described above. We use the Deepsets encoder with output dimension 2. Latent noise dimension used is 20. For calculating OT loss, we use $N = 100$ and $\epsilon = 0.01$. The regularization value λ used was 10.

Interestingly, we observed that on training data, our model, even without OT regularization, learns distances that are strongly correlated to true Wasserstein distances calculated using the PythonOT package (Flamary & Courty, 2017) (see Fig. 5). We hypothesize that this is due to the nature of neural networks, which want to learn smooth functions, and implicitly find low energy solutions. This may result in functions that provide shorter paths in the conditions space, resembling optimal transport. However, OT loss significantly improved correlation with ground truth distances as well as allows us to guarantee OT, in any situation.

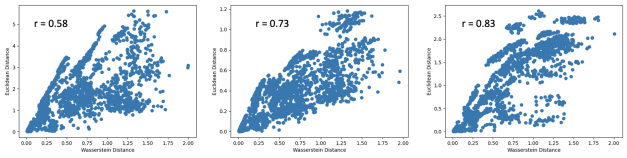

Figure 5. The Distributional Encoder learns Wasserstein distances without the explicit use of distance functions. X-axes show ground truth Wasserstein distances, Y-axes show euclidean distances in the encoded space. Pearson correlation values are shown within each plot. **(A)** Random network weights ($r = 0.58$) **(B)** Trained network without OT loss ($r = 0.73$) **(C)** Trained network with OT loss regularization ($r = 0.83$).

We have also analyzed the generative properties of the cGAN model on both parametric distributions and MNIST data. For visualizations of the generated outputs, please refer to the supplementary material.

Runtime Comparison between model variants

We also compare the runtime of the cGAN based training with Sinkhorn loss based training on the point cloud dataset described in the next section. We observed the convergence time for the sinkhorn training to be around 1000 seconds, while for cGAN training it was around 200 seconds. This gap can be attributed to the high computational cost associated with the Sinkhorn distance function.

5.2. Point Cloud Experiments

ModelNet10 is a shape classification benchmark dataset which is a subset of ModelNet40 containing 4899 pre-aligned shapes from 10 categories. There are $3,991$ (80%) shapes for training and 908 (20%) shapes for testing. We use ModelNet10 in this work and use PyTorch Geometric (Fey & Lenssen, 2019) for loading the data. For each shape, we sample point clouds of size 100. We also use zero centering and $[-1, 1]$ scaling for normalizing the point clouds before feeding them to our network.

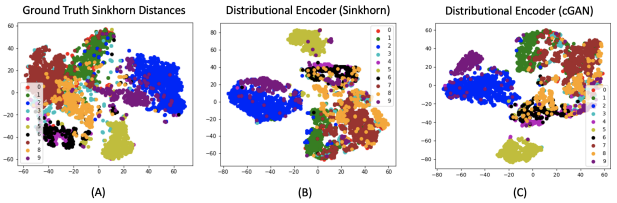

Figure 6. Embeddings learned by the Distributional Encoder closely resemble ground truth Sinkhorn distance embeddings. Shown are embeddings using t-SNE on ModelNet10 data. **(A)** Ground truth Sinkhorn distances **(B)** Embedding learned by the Distributional Encoder using Sinkhorn loss **(C)** Embedding learned by the Distributional Encoder using cGAN (dynamical OT) loss.

- Pretraining: For pre-training the DeepSets encoder using the Sinkhorn loss, we use the vanilla DeepSets architecture as used in (Zaheer et al., 2017) consisting of 3 permutation equivariant layers with 256 channels followed by max pooling. The pooled representation is fed to a fully connected layer with 256 units. We remove the last classification layer for pretraining. We compare the t-SNE embeddings (van der Maaten & Hinton, 2008) obtained using the ground truth Sinkhorn distances and embeddings obtained using the Euclidean distances in the learned space and observe in Fig. 6 that the embeddings look very similar. This confirms that it is possible to embed Wasserstein distances in the Euclidean space even for real world datasets like point clouds. We also show the embeddings learned by cGAN encoder and we can observe that it also learns very similar embeddings without explicit supervision of sinkhorn distances.

- Classification: After pre-training, we train a softmax layer on the top of this network for the downstream classification task after freezing the pre-trained network. Finally we fine-tune the entire network on the same task after unfreezing the previous layers. We compare our approach with the same network trained entirely from scratch on the classification task.

- Label Efficiency: Since the pretraining does not require any labels, we expect our model to be label efficient. To show this, we experiment with different percentage of labels for training on the downstream classification task in Table 1. We found that the pre-trained network

performs on par with the deepsets model trained from scratch and outperforms it in the limited labeled data regime when a significant portion of data (90%) is unlabeled.

Table 1. Classification Accuracy on ModelNet10 with different percentage of labels used for training

Method	100%	50%	10%
Training from scratch	87.22	78.41	61.01
Pre-training + fine-tuning	84.36	76.87	64.32

6. Conclusion

In this work, we presented the Distributional Encoder, a framework for encoding samples from distributions into a metric space with Wasserstein properties. The method is based on permutation invariant functions, such as DeepSets, and a Wasserstein loss on the encoding space. This method can compute the Sinkhorn distances between samples from known and unknown measures. Interestingly, for $1D$ measures, we found strong correlation between the encoded vectors and moments of the distributions (mean and variance). We also showed that the Wasserstein geodesics correspond to straight lines in our encoded space, thus allowing accurate encoding of barycenters. Further, we presented an alternative training method, by using dynamical optimal transport via a conditional GAN, that does not require explicit distance computation. We showed that the Euclidean distances in the obtained space have high correlation with Wasserstein distances. Not only does this method provide much faster training, but also enables convenient generating from the latent space. Finally, we show that our method learns meaningful embeddings that can be used for downstream classification tasks.

References

Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L. Learning Representations and Generative Models for 3D Point Clouds. In *International conference on machine learning*, pp. 40–49. PMLR, 2018.

Agueh, M. and Carlier, G. Barycenters in the Wasserstein Space. *SIAM Journal on Mathematical Analysis*, 43(2): 904–924, 2011. doi: 10.1137/100805741. URL https://doi.org/10.1137/100805741.

Altschuler, J., Weed, J., and Rigollet, P. Near-linear time approximation algorithms for Optimal Transport via Sinkhorn iteration, 2018.

Alvarez-Melis, D. and Fusi, N. Geometric Dataset Distances via Optimal Transport. ArXiv, February 2020.

Ambrosio, L., Gigli, N., and Savare, G. *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Birkhauser, 01 2005. doi: 10.1007/978-3-7643-8722-8.

Anderes, E., Borgwardt, S., and Miller, J. Discrete Wasserstein Barycenters: Optimal Transport for Discrete Data, 2015.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein GAN, 2017.

Avraham, G., Zuo, Y., and Drummond, T. Parallel Optimal Transport GAN. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4406–4415, 2019. doi: 10.1109/CVPR.2019.00454.

Balaji, Y., Chellappa, R., and Feizi, S. Robust Optimal Transport with Applications in Generative Modeling and Domain Adaptation, 2020.

Bigot, J., Gouet, R., Klein, T., and López, A. Geodesic PCA in the Wasserstein space, 2013.

Bobkov, S. and Ledoux, M. One-dimensional empirical measures, order statistics, and Kantorovich transport distances. *Memoirs of the American Mathematical Society*, 261:0–0, 2019.

Bonneel, N., Peyré, G., and Cuturi, M. Wasserstein Barycentric Coordinates: Histogram Regression Using Optimal Transport. *ACM Transactions on Graphics*, 35(4):71:1–71:10, April 2016. doi: 10.1145/2897824.2925918. URL https://hal.archives-ouvertes.fr/hal-01303148.

Bréchet, P., Wu, T., Möllenhoff, T., and Cremers, D. Informative GANs via Structured Regularization of Optimal Transport, 2019.

Canas, G. D. and Rosasco, L. Learning Probability Measures with respect to Optimal Transport Metrics, 2012.

Cao, H., Guo, X., and Laurière, M. Connecting GANs, MFGs, and OT, 2020.

Chu, C., Minami, K., and Fukumizu, K. Smoothness and Stability in GANs, 2020.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs), 2016.

Courty, N., Flamary, R., and Ducoffe, M. Learning Wasserstein Embeddings, 2017.

Cuturi, M. Sinkhorn Distances: Lightspeed Computation of Optimal Transportation Distances, 2013.

Cuturi, M. and Doucet, A. Fast Computation of Wasserstein Barycenters, 2014.

de Goes, F., Breeden, K., Ostromoukhov, V., and Desbrun, M. Blue Noise through Optimal Transport. *ACM Trans. Graph.*, 31(6), November 2012. ISSN 0730-0301. doi: 10.1145/2366145.2366190. URL https://doi.org/10.1145/2366145.2366190.

del Barrio, E., Giné, E., and Matrán, C. Central Limit Theorems for the Wasserstein Distance Between the Empirical and the True Distributions. *Ann. Probab.*, 27(2):1009–1071, 04 1999. doi: 10.1214/aop/1022677394. URL https://doi.org/10.1214/aop/1022677394.

Fan, H., Su, H., and Guibas, L. J. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 605–613, 2017.

Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Feydy, J., Séjourné, T., Vialard, F.-X., ichi Amari, S., Trouvé, A., and Peyré, G. Interpolating between Optimal Transport and MMD using Sinkhorn Divergences, 2018.

Finlay, C., Jacobsen, J.-H., Nurbekyan, L., and Oberman, A. M. How to train your neural ODE: the world of Jacobian and kinetic regularization, 2020.

Flamary, R. and Courty, N. Pot python optimal transport library, 2017. URL https://pythonot.github.io/.

Frogner, C., Zhang, C., Mobahi, H., Araya-Polo, M., and Poggio, T. Learning with a Wasserstein Loss, 2015.

Frogner, C., Mirzazadeh, F., and Solomon, J. Learning Embeddings into Entropic Wasserstein Spaces, 2019.

Gadelha, M., Chowdhury, A. R., Sharma, G., Kalogerakis, E., Cao, L., Learned-Miller, E., Wang, R., and Maji, S. Label-Efficient Learning on Point Clouds using Approximate Convex Decompositions. In *European Conference on Computer Vision 2020*, 2020.

Genevay, A. Entropy-Regularized Optimal Transport for Machine Learning. PhD Thesis, 2019.

Givens, C. R. and Shortt, R. M. A class of Wasserstein metrics for probability distributions. *Michigan Math. J.*, 31(2):231–240, 1984. doi: 10.1307/mmj/1029003026. URL https://doi.org/10.1307/mmj/1029003026.

Harutyunyan, H., Khachatrian, H., Kale, D. C., Ver Steeg, G., and Galstyan, A. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(1):96, 2019. ISSN 2052-4463. doi: 10.1038/s41597-019-0103-9. URL https://doi.org/10.1038/s41597-019-0103-9.

Jacob, L., She, J., Almahairi, A., Rajeswar, S., and Courville, A. W2GAN: Recovering an Optimal Transport Map with a GAN, 2019. URL https://openreview.net/forum?id=BJx9f305t7.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. Mimic-III, a freely accessible critical care database. *Scientific data*, 3:160035, 2016a.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., and Mark, R. G. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3(1):160035, May 2016b. ISSN 2052-4463. doi: 10.1038/sdata.2016.35. URL https://doi.org/10.1038/sdata.2016.35.

Kantorovich, L. On the Translocation of Masses. *Journal of Mathematical Sciences*, 133, 03 2006. doi: 10.1007/s10958-006-0049-2.

Karcher, H. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 30(5):509–541, 1977. doi: 10.1002/cpa.3160300502. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160300502.

Karcher, H. Riemannian Center of Mass and so called Karcher mean, 2014.

Kawano, K., Koide, S., and Kutsuna, T. Learning Wasserstein Isometric Embedding for Point Clouds. In *2020 International Conference on 3D Vision (3DV)*, pp. 473–482, 2020. doi: 10.1109/3DV50981.2020.00057.

Kloeckner, B. R. A Geometric Study of Waserstein Spaces: Ultrametrics. *Mathematika*, 61(1):162–178, May 2014. ISSN 2041-7942. doi: 10.1112/s0025579314000059. URL http://dx.doi.org/10.1112/S0025579314000059.

Koch, G., Zemel, R., and Salakhutdinov, R. Siamese Neural Networks for One-shot Image Recognition, 2015.

Li, C.-L., Zaheer, M., Zhang, Y., Poczos, B., and Salakhutdinov, R. Point cloud GAN, 2018.

Mallasto, A. and Feragen, A. Learning from Uncertain Curves: The 2-Wasserstein Metric for Gaussian Processes. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 5665–5674, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

McCann, R. J. A convexity principle for interacting gases. *Advances in Mathematics*, 128(1):153 – 179, 1997. ISSN 0001-8708. doi: https://doi.org/10.1006/aima.1997.1634. URL http://www.sciencedirect.com/science/article/pii/S0001870897916340.

Mirza, M. and Osindero, S. Conditional Generative Adversarial Nets. *CoRR*, abs/1411.1784, 2014. URL http://arxiv.org/abs/1411.1784.

Mroueh, Y., Sercu, T., and Raj, A. Sobolev Descent. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pp. 2976–2985. PMLR, 16–18 Apr 2019. URL http://proceedings.mlr.press/v89/mroueh19a.html.

Neumayer, S. and Steidl, G. From Optimal Transport to Discrepancy, 2020.

Panaretos, V. and Zemel, Y. *An Invitation to Statistics in Wasserstein Space*. Springer Briefs in Probability and Mathematical Statistics, 01 2020. ISBN 978-3-030-38437-1. doi: 10.1007/978-3-030-38438-8.

Panaretos, V. M. and Zemel, Y. Statistical Aspects of Wasserstein Distances. *Annual Review of Statistics and Its Application*, 6(1):405–431, 2019. doi: 10.1146/annurev-statistics-030718-104938. URL https://doi.org/10.1146/annurev-statistics-030718-104938.

Peyré, G. and Cuturi, M. Computational Optimal Transport, 2020.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *CoRR*, abs/1612.00593, 2016. URL http://arxiv.org/abs/1612.00593.

Salimans, T., Zhang, H., Radford, A., and Metaxas, D. Improving GANs using Optimal Transport, 2018.

Sanjabi, M., Ba, J., Razaviyayn, M., and Lee, J. D. On the Convergence and Robustness of Training GANs with Regularized Optimal Transport, 2018.

Santambrogio, F. *Wasserstein distances and curves in the Wasserstein spaces*, pp. 177–218. Springer International Publishing, Cham, 2015. ISBN 978-3-319-20828-2. doi: 10.1007/978-3-319-20828-2_5. URL https://doi.org/10.1007/978-3-319-20828-2_5.

Sauder, J. and Sievers, B. Self-supervised Deep Learning on Point Clouds by Reconstructing Space, 2019.

Seguy, V., Damodaran, B. B., Flamary, R., Courty, N., Rolet, A., and Blondel, M. Large-scale Optimal Transport and Mapping Estimation, 2018.

Shirdhonkar, S. and Jacobs, D. W. Approximate Earth Mover's Distance in linear time. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

Sklar, M. Fonctions de repartition á n dimensions et leurs marges. *Publications de l'Institut Statistique de l'Université de Paris*, pp. 229–231, 1959.

Takatsu, A. Wasserstein geometry of Gaussian measures. *Osaka J. Math.*, 48(4):1005–1026, 12 2011. URL https://projecteuclid.org:443/euclid.ojm/1326291215.

Tanaka, A. Discriminator Optimal Transport, 2019.

Tong, A., Huang, J., Wolf, G., van Dijk, D., and Krishnaswamy, S. TrajectoryNet:A Dynamic Optimal Transport Network for Modeling Cellular Dynamics, 2020.

Tran, N.-T., Tran, V.-H., Nguyen, N.-B., Yang, L., and Cheung, N.-M. Self-supervised GAN: Analysis and Improvement with Multi-class Minimax Game. In *NeurIPS*, December 2019.

van der Maaten, L. and Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9: 2579–2605, 2008. URL http://www.jmlr.org/papers/v9/vandermaaten08a.html.

Villani, C. Topics in Optimal Transportation, 2003.

Villani, C. Optimal Transport: Old and New, 2008.

Wang, J.-L., Chiou, J.-M., and Mueller, H.-G. Review of Functional Data Analysis, 2015.

Wasserman, L. Topological Data Analysis, 2016.

Wu, Z., Song, S., Khosla, A., Tang, X., and Xiao, J. 3D ShapeNets : A Deep Representation for Volumetric Shapes. *CoRR*, abs/1406.5670, 2014. URL http://arxiv.org/abs/1406.5670.

Yang, Y., Feng, C., Shen, Y., and Tian, D. FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation, 2018.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R., and Smola, A. Deep Sets, 2017.

Zemel, Y. and Panaretos, V. M. Fréchet Means and Procrustes Analysis in Wasserstein Space, 2017.

Zhang, L. and Zhu, Z. Unsupervised Feature Learning for Point Cloud by Contrasting and Clustering with Graph Convolutional Neural Network, 2019.