LEARNING DISCRETE WASSERSTEIN EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite their prevalence, Euclidean embeddings of data are fundamentally limited in their ability to capture latent semantic structures, which need not conform to Euclidean spatial assumptions. Here we consider an alternative, which embeds data as discrete probability distributions in a Wasserstein space, endowed with an optimal transport metric. Wasserstein spaces are much larger and more flexible than Euclidean spaces, in that they can successfully embed a wider variety of metric structures. We propose to exploit this flexibility by learning an embedding that captures the semantic information in the Wasserstein distance between embedded distributions. We examine empirically the representational capacity of such learned Wasserstein embeddings, showing that they can embed a wide variety of complex metric structures with smaller distortion than an equivalent Euclidean embedding. We also investigate an application to word embedding, demonstrating a unique advantage of Wasserstein embeddings: we can directly visualize the high-dimensional embedding, as it is a probability distribution on a low-dimensional space. This obviates the need for dimensionality reduction techniques such as t-SNE for visualization.

1 INTRODUCTION

Pre-trained embeddings form the basis for many state-of-the-art learning systems. Word embeddings like word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), fastText (Bojanowski et al., 2017), and ELMo (Peters et al., 2018) are ubiquitous in natural language processing, where they are used for tasks like machine translation (Neubig et al., 2018). Graph embeddings (Nickel et al., 2016) like node2vec (Grover & Leskovec, 2016) are used to represent knowledge graphs, and pre-trained image models are used in a wide variety of computer vision tasks.

A good embedding should represent the semantic structure of the data with high fidelity, in a way that is accessible to downstream tasks. This makes the choice of a target space for the embedding important, since different spaces can represent different types of semantic structure. The most common choice is to embed data into Euclidean space, where distances and angles between vectors encode their levels of association (Mikolov et al., 2013; Weston et al., 2011; Kiros et al., 2014; Mirzazadeh et al., 2014). Yet Euclidean spaces are limited in their ability to represent complex relationships between inputs, as they make restrictive assumptions about neighborhood sizes and connectivity. This has been documented recently in the case of tree-structured data, for example, where spaces of negative curvature are required due to exponential scaling of neighborhood sizes (Nickel & Kiela, 2017; 2018).

In this paper, we embed inputs as probability distributions in a Wasserstein space. Wasserstein spaces endow probability distributions with an *optimal transport* metric, which measures (in the simplest case) the distance traveled in transporting the mass in one distribution to match another. Recent theory has shown that Wasserstein spaces are quite flexible – more so than Euclidean spaces – allowing a variety of other metric spaces to be embedded within them while preserving their original distance metrics. As such, they make an attractive target for learning embeddings in machine learning, where this flexibility might capture complex relationships between objects when other embeddings fail to do so.

Unlike prior work on Wasserstein embeddings, which has focused on embedding into Gaussian distributions (Muzellec & Cuturi, 2018; Zhu et al., 2018), we propose to embed input data as discrete distributions, supported at a fixed number of points. In doing so we attempt to access the full flexi-

bility of Wasserstein spaces to represent a wide variety of semantic structures. This approach is enabled by recent work that approximates Wasserstein distances with the Sinkhorn divergence (Cuturi, 2013), in which the underlying optimal transport problem is regularized to render it more tractable. While exact Wasserstein distances are generally too computationally costly to be used in a learning setting, the Sinkhorn divergence is computed efficiently by a fixed point iteration. This iteration is suitable for gradient-based optimization, via automatic differentiation. This has been shown in the context of generative modeling (Genevay et al., 2018), for example.

We investigate empirically two settings for Wasserstein embeddings. First, we show their representational capacity by embedding a variety of complex networks, on which Wasserstein embeddings achieve higher fidelity than both Euclidean and hyperbolic embeddings. And second, we examine Wasserstein word embeddings, which show retrieval performance comparable to existing methods. One major benefit of embedding into probability distributions is that the distributions can be visualized directly, unlike most embeddings, which require a dimensionality reduction step (such as t-SNE) before visualization. We demonstrate the power of this approach by visualizing the learned word embeddings.

2 Preliminaries

2.1 Optimal transport and Wasserstein distance

The p**-Wasserstein distance** between probability distributions μ and ν over a metric space $\mathcal{X}$ is

$$\mathcal{W}_p(\mu, \nu) = \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X} \times \mathcal{X}} d(x_1, x_2)^p \, d\pi(x_1, x_2) \right)^{\frac{1}{p}}, \tag{1}$$

where the infimum is taken over **transport plans** π that distribute the mass in μ to match that in ν, with the p-th power of the **ground metric** $d(x_1, x_2)$ on $\mathcal{X}$ giving the cost of moving a unit of mass from support point $x_1 \in \mathcal{X}$ underlying distribution μ to point $x_2 \in \mathcal{X}$ underlying ν. The Wasserstein distance is the cost of the optimal transport plan matching μ and ν (Villani, 2003).

In this paper, we are concerned with **discrete distributions** supported on finite sets of points in $\mathbb{R}^n$:

$$\mu = \sum_{i=1}^{M} \mathbf{u}_i \delta_{\mathbf{x}^{(i)}} \qquad \text{and} \qquad \nu = \sum_{i=1}^{N} \mathbf{v}_i \delta_{\mathbf{y}^{(i)}}. \tag{2}$$

Here, $\mathbf{u}$ and $\mathbf{v}$ are vectors of nonnegative weights summing to 1, and $\{\mathbf{x}^{(i)}\}_{i=1}^{M}, \{\mathbf{y}^{(i)}\}_{i=1}^{N} \subset \mathbb{R}^n$ are the **support points**. In this case, the optimal transport plan π matching μ and ν in equation 1 becomes discrete as well, supported on the product of the supports. Define $D \in \mathbb{R}_+^{M \times N}$ to be the matrix of pairwise ground metric distances, with $D_{ij} = \|\mathbf{x}^{(i)} - \mathbf{y}^{(j)}\|_p$. Then, for our discrete distributions, equation 1 is equivalent to solving the following:

$$\mathcal{W}_p(\mu, \nu)^p = \min_{T \geq 0} \operatorname{tr}(D^p T^\top) \quad \text{subject to} \quad T\mathbf{1} = \boldsymbol{u}, \quad T^\top \mathbf{1} = \boldsymbol{v}, \tag{3}$$

with T_{ij} giving the transported mass between $\mathbf{x}_i$ and $\mathbf{y}_j$. The power D^p is taken elementwise.

2.2 Sinkhorn divergence

Equation 3 is a linear program that can be challenging to solve in practice. To improve efficiency of numerical optimal transport, recent learning algorithms use an entropic regularizer proposed by Cuturi (2013). The resulting **Sinkhorn divergence** solves a slightly modified version of equation 3:

$$\mathcal{W}_p^\lambda(\mu, \nu)^p = \min_{T \geq 0} \operatorname{tr}(D^p T^\top) + \lambda \mathbf{1}^\top \left(T \circ (\log(T) - \mathbf{1}\mathbf{1}^\top) \right) \mathbf{1} \;\text{ s.t. }\; T\mathbf{1} = \boldsymbol{u},\; T^\top \mathbf{1} = \boldsymbol{v}, \tag{4}$$

where $\log(\cdot)$ is applied elementwise, $\circ$ is a Hadamard (elementwise) product, and $\lambda \geq 0$ is a regularization parameter. For $\lambda > 0$, the optimal solution takes the form $T^* = \Delta(\boldsymbol{r}) \exp\left(-D^p/\lambda\right) \Delta(\boldsymbol{c})$, for new vector variables $\boldsymbol{r}$ and $\boldsymbol{c}$, where $\Delta(\cdot)$ puts a vector along the diagonal of a square matrix. Hence, rather than optimizing over matrices T, one can optimize over vectors $\boldsymbol{r}$ and $\boldsymbol{c}$, reducing the size of

the problem to $M + N$. This can be solved via *matrix rebalancing* starting from an initial matrix $K := \exp(\frac{-D^p}{\lambda})$ and alternatively projecting onto the marginal constraints until convergence:

$$\boldsymbol{r} \leftarrow \boldsymbol{u}./K\boldsymbol{c} \qquad \boldsymbol{c} \leftarrow \boldsymbol{v}./K^\top \boldsymbol{r}. \tag{5}$$

Here, ./ denotes elementwise division for vectors.

Beyond simplicity of implementation, equation 5 has an additional advantage for machine learning applications: The steps of this algorithm are easily differentiable. With this observation in mind, Genevay et al. (2018) incorporate entropic transport into learning pipelines by applying automatic differentiation (back propagation) to a fixed number of Sinkhorn iterations.

2.3 What can we embed in theory?

Given two metric spaces $\mathcal{A}$ and $\mathcal{B}$, an embedding of $\mathcal{A}$ into $\mathcal{B}$ is a map $\phi : \mathcal{A} \to \mathcal{B}$ that preserves distances, in the sense that the distortion is small:

$$Ld_{\mathcal{A}}(u, v) \leq d_{\mathcal{B}}(\phi(u), \phi(v)) \leq CLd_{\mathcal{A}}(u, v), \quad \forall u, v \in \mathcal{A}, \tag{6}$$

for some uniform constants $L > 0$ and $C \geq 1$. The distortion of the embedding ϕ is the smallest C such that equation 6 holds.

One can characterize how "large" a space is (its *representational capacity*) by the spaces that successfully embed into it with low distortion. In practical terms, this capacity determines the types of data (and relationships between them) that can be well-represented in the embedding space. $\mathbb{R}^n$ with the Euclidean metric, for example, embeds into the L^1 metric with low distortion, while the reverse is not true (Deza & Laurent, 2009). Conversely, we do not expect Manhattan-structured data to be well-represented in Euclidean space, no matter how clever the mapping.

Wasserstein spaces are very large: Many spaces can embed into Wasserstein spaces with low distortion, even when the converse is not true. $\mathcal{W}_p(\mathcal{A})$, for $\mathcal{A}$ an arbitrary metric space, embeds any product space $\mathcal{A}^n$, for example (Kloeckner, 2010). Similarly, every finite metric space embeds certain powers of its metric into $\mathcal{W}_p(\mathbb{R}^3)$, $p > 1$, with vanishing distortion (Andoni et al., 2015). This last result suggests a potential universality for the ground space $\mathbb{R}^3$, which would mean that $\mathcal{W}_p(\mathbb{R}^3)$ **is a plausible target space for arbitrary metrics on symbolic data, with a finite set of symbols.** The authors are not aware of any similar universality results for L^p or hyperbolic spaces, for example.

The reverse direction, embedding Wasserstein spaces into others, is particularly well-studied in the case of discrete distributions. Theoretical results in this domain are motivated by interest in efficient algorithms for approximating Wasserstein distances by embedding them into spaces with more easily-computed metrics. In this direction, low-distortion embeddings are not easy to find. $\mathcal{W}_2(\mathbb{R}^3)$, for example, is known not to embed into L^1 space (Andoni et al., 2016). Some positive results exist, nevertheless. For a Euclidean ground metric, for example, the 1-Wasserstein distance can be approximated in a wavelet domain (Shirdhonkar & Jacobs, 2008) or by high-dimensional embedding into L^1 (Indyk & Thaper, 2003).

In §4, we investigate empirically the embedding capacity of Wasserstein spaces, by attempting to learn low-distortion embeddings for a variety of input spaces.

2.4 Related work

While learning to embed into a vector space has a long history, there is a recent tendency in the representation learning community to generalize from individual points to more complex structures and spaces like probability distributions (Vilnis & McCallum, 2015; Athiwaratkun & Wilson, 2018), Euclidean norm balls (Mirzazadeh et al., 2015; Mirzazadeh, 2017), Poincaré balls (Nickel & Kiela, 2017), and Lorentz models (Nickel & Kiela, 2018). From a modeling perspective, generalization to other spaces helps models carry information like uncertainty (Vilnis & McCallum, 2015; Bojchevski & Günnemann, 2018) to the objects being embedded (Vilnis & McCallum, 2015). More importantly, alternative embedding spaces have the capacity to encode relations like inclusion, exclusion, hierarchy, and ordering (Mirzazadeh et al., 2015; Vendrov et al., 2015; Athiwaratkun & Wilson, 2018). Our work share the same themes, since we take point clouds as the embedding targets.

The distance or discrepancy measure between points in an embedding space is another major defining factor for a representation learning model. For points, the most typical choice is to use an L_p norm. When embedding into the set of histograms, discrepancy typically is evaluated using KL divergence (Kullback & Leibler, 1951). KL divergence between two distributions can be problematic, however. In particular, the behavior of KL divergence relies heavily on distributions having common non-empty support; once the support sets are disjoint, regardless of how close or far the support points lie on the underlying space, the KL divergence remains the same. As an alternative discrepancy measure, transport distances (Villani, 2008; Peyré & Cuturi, 2017; Solomon, 2018) do not suffer from the mentioned problems. Hence, models based on optimal transport are gaining popularity in machine learning; see (Rubner et al., 1998; Courty et al., 2014; Frogner et al., 2015; Kusner et al., 2015; Arjovsky et al., 2017; Genevay et al., 2018; Claici et al., 2018) for some examples.

Learned embeddings into Wasserstein spaces do not have long precedence. A recent line of research proposes embedding into parameterized sets of distributions (Muzellec & Cuturi, 2018; Zhu et al., 2018). While restricting to smaller sets facilitates use of closed-form expressions for transport distances, restriction to Gaussian distributions leads to a less expressive representation space. It is noteworthy that Courty et al. (2018) study embedding from Wasserstein into Euclidean space. In contrast, we learn to embed into the space of discrete probability distributions endowed with the Wasserstein distance, well-known to be dense in $\mathcal{W}_2$ as the number of support points increases (Kloeckner, 2012; Brancolini et al., 2009).

3 Learning Wasserstein embeddings

3.1 The learning problem

The learning task we consider is that of recovering a pairwise distance or similarity relationship that is partially or fully observed. We are given a collection of objects $\mathcal{C}$—these can be words, symbols, images, or any other data—as well as samples $\left\{\left(u^{(i)}, v^{(i)}, r(u^{(i)}, v^{(i)})\right)\right\}$ of a target relationship $r : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$ that tells us the degree to which pairs of objects are related.

Our objective is to find a map $\phi : \mathcal{C} \to \mathcal{W}_p(\mathcal{X})$ such that the relationship $r(u, v)$ can be recovered from the Wasserstein distance between $\phi(u)$ and $\phi(v)$, for any $u, v \in \mathcal{C}$. Examples include:

1. Metric embedding: r is a distance metric and we want $\mathcal{W}_p(\phi(u), \phi(v)) \approx r(u, v)$ for all $u, v \in \mathcal{C}$.
2. Graph embedding: $\mathcal{C}$ contains the vertices of a graph and $r : \mathcal{C} \times \mathcal{C} \to \{0, 1\}$ is the adjacency relation; we would like the neighborhood of each $\phi(u)$ in $\mathcal{W}_p$ to coincide with graph adjacency.
3. Word embedding: $\mathcal{C}$ contains individual words and r is a semantic similarity between words. We want distances in $\mathcal{W}_p$ to predict this semantic similarity.

Although the details of each task require some adjustment to the learning architecture, our basic representation and training procedure detailed below applies to all three examples.

3.2 Optimization

Given a set of training samples $\mathcal{S} = \left\{\left(u^{(i)}, v^{(i)}, r^{(i)}\right)\right\}_{i=1}^{N} \subset \mathcal{C} \times \mathcal{C} \times \mathbb{R}$, we want to learn a map $\phi : \mathcal{C} \to \mathcal{W}_p(\mathcal{X})$. We must address two issues.

First we must define the range of our map ϕ. The whole of $\mathcal{W}_p(\mathcal{X})$ is infinite-dimensional, and for a tractable problem we need a finite-dimensional output. We restrict ourselves to discrete distributions with an *a priori* fixed number of support points M, reducing to the linear program in equation 3. Such a distribution is parameterized by the locations of its support points $\{\mathbf{x}^{(j)}\}_{j=1}^{M}$, forming a point cloud in the ground metric space $\mathcal{X}$. For simplicity, we restrict to uniform weights $\mathbf{u}, \mathbf{v} \propto \mathbf{1}$, although it is certainly possible to optimize simultaneously over weights and locations; as noted in (Brancolini et al., 2009; Claici et al., 2018), however, weighting does not asymptotically improve approximation quality as a function of M. As we discuss in §2.3, the discrete distributions are sufficiently expressive to embed a wide variety of input spaces.

The second issue is that, as noted in §2.2, exact computation of $\mathcal{W}_p$ in general is costly, requiring the solution of a linear program. As in (Genevay et al., 2018), we instead replace $\mathcal{W}_p$ with the Sinkhorn divergence $\mathcal{W}_p^\lambda$, which is solvable by a simple fixed-point iteration introduced in §2.2, equation 5.

Learning then takes the form of empirical loss minimization,

$$\phi_* = \arg\min_{\phi \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(\mathcal{W}_p^\lambda\left(\phi(u^{(i)}), \phi(v^{(i)})\right), r^{(i)}\right), \tag{7}$$

over a hypothesis space of maps $\mathcal{H}$. The loss $\mathcal{L}$ is problem-specific and scores the coincidence between the Wasserstein distance $\mathcal{W}_p$ and the target relationship r, evaluated at the pair $\left(u^{(i)}, v^{(i)}\right)$. As mentioned in §2.2, gradients are available from automatic differentiation of the Sinkhorn procedure, and hence with a suitable loss function the learning objective 7 can be optimized by standard gradient-based methods. In the experiments that follow, we use the Adam optimizer.

4 Empirical study

4.1 Representational capacity: embedding complex networks

We first demonstrate the representational power of learned Wasserstein embeddings. As discussed in §2.3, theory suggests that Wasserstein spaces are quite flexible, in that they can embed a wide variety of metrics with low distortion. We show that this is true in practice as well.

To generate a variety of metrics to embed, we take networks with various patterns of connectivity and compute the shortest path distances between vertices. The collection of vertices for each network serves as the input space $\mathcal{C}$ for our embedding, and our goal is to learn a map $\phi : \mathcal{C} \to \mathcal{W}_p(\mathbb{R}^k)$ such that the Wasserstein distance $\mathcal{W}_p(\phi(u), \phi(v))$ matches as closely as possible the shortest path distance between vertices u and v, for all pairs of vertices. We learn a **minimum distortion embedding**: given a fully observed distance metric $d_{\mathcal{C}} : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$ in the input space, we minimize the mean distortion:

$$\mathcal{L}(\phi, d_{\mathcal{C}}) = \frac{1}{\binom{n}{2}} \sum_{j>i} \frac{|\mathcal{W}_1^\lambda(\phi(v_i), \phi(v_j)) - d_{\mathcal{C}}(v_i, v_j)|}{d_{\mathcal{C}}(v_i, v_j)}. \tag{8}$$

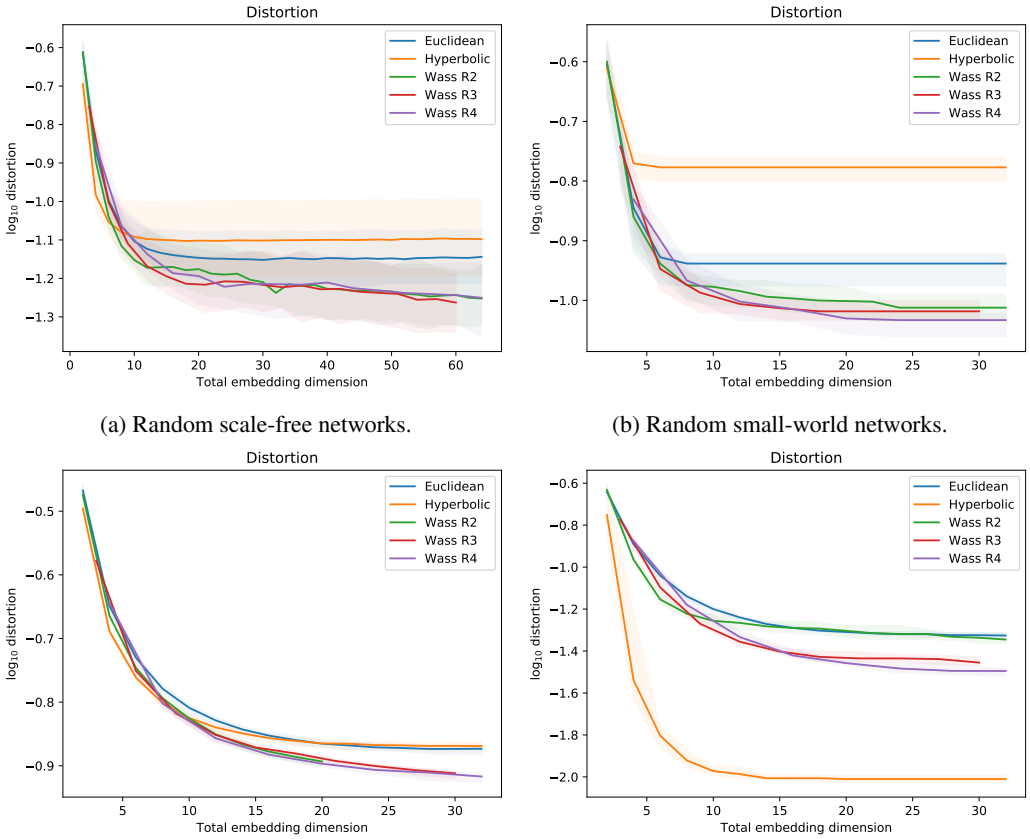
ϕ is parameterized as in §3.2.

We examine the performance of Wasserstein embedding using both random networks and real networks. The random networks in particular allow us systematically to test robustness of the Wasserstein embedding to particular properties of the metric we are attempting to embed.

For random networks, we use three standard generative models: Barabási–Albert (Albert & Barabási, 2002), Watts–Strogatz (Watts & Strogatz, 1998), and the stochastic block model (Holland et al., 1983). **Random scale-free networks** are generated from the Barabási–Albert model, and possess the property that distances are on average much shorter than in a Euclidean spatial graph, scaling like the log of the number of vertices. **Random small-world networks** are generated from the Watts–Strogatz model; in addition to log-scaling of the average path length, Watts–Strogatz graphs also show clustering of vertices into distinct neighborhoods. **Random community-structured networks** are generated from the stochastic block model, which places vertices within densely-connected communities, with only sparse connections between the different communities. We additionally generate **random trees** by choosing a random number of children [1] for each node, progressing in breadth-first order until a specified total number of nodes is reached. In all cases, we generate networks with 128 vertices.

We compare against two baselines, trained using the same distortion criterion: **Euclidean embeddings**, and **hyperbolic embeddings**. Euclidean embeddings we expect to struggle with all of the chosen graph types, as they are limited to representing spatial relationships with zero curvature. Hyperbolic embeddings model tree-structured metrics, as they capture the exponential scaling of the graph neighborhoods, and have been suggested for a variety of other graph families as well (Zhao et al., 2011).

[1] Each non-leaf node has a number of children drawn uniformly from $\{2, 3, 4\}$.



(a) Random scale-free networks.

(b) Random small-world networks.

(c) Random community-structured networks.

(d) Random trees.

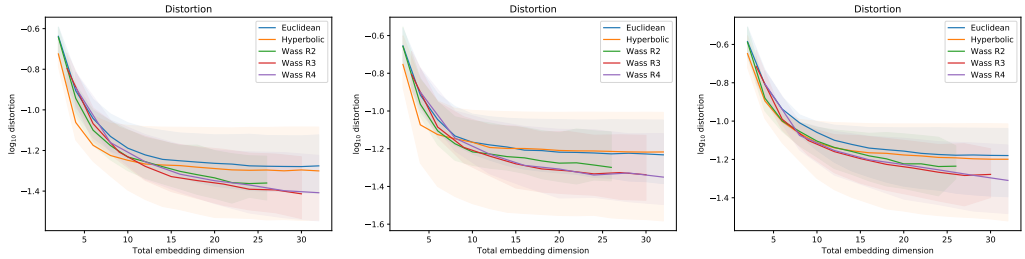
Figure 1: Random networks: Learned Wasserstein embeddings achieve lower distortion than Euclidean and hyperbolic embeddings. Hyperbolic outperform specifically on random trees.

Figure 1 shows the result of embedding random networks [2]. As the total embedding dimension increases, the distortion decreases for all of the embedding methods. Importantly, the Wasserstein embeddings achieve lower distortion than both the Euclidean and hyperbolic embeddings, establishing their flexibility under the varying conditions represented by the different network models. In some cases, the Wasserstein distortion continues to decrease long after the other embeddings have saturated their capacity. Note that, as expected, the hyperbolic space significantly outperforms both Euclidean and Wasserstein on tree-structured metrics.

Note also that we test the $\mathbb{R}^2$, $\mathbb{R}^3$, and $\mathbb{R}^4$ ground metric spaces. For all of the random networks we examined, the performance between $\mathbb{R}^3$ and $\mathbb{R}^4$ ground metrics is indistinguishable. This is consistent with theoretical results (§2.3) that suggest that $\mathbb{R}^3$ is sufficient to embed a wide variety of metrics.

We also examine fragments of **real networks**: an ArXiv co-authorship network, an Amazon product co-purchasing network, and a Google web graph (Leskovec & Krevl, 2014). For each graph fragment, we choose uniformly at random a starting vertex, then extract the subgraph on 128 vertices taken in breadth-first order from that starting vertex. The results of this experiment are shown in Figure 2. Again we see that the Wasserstein embeddings achieve lower distortion than either Euclidean or hyperbolic embeddings.

[2]The solid line is the median over 20 randomly-generated inputs, while shaded is the middle 95%.



(a) arXiv co-authorship. (b) Amazon product co-purchases. (c) Google web graph.

Figure 2: Real networks: Learned Wasserstein embeddings achieve lower distortion than Euclidean and hyperbolic embeddings of real network fragments.

4.2 WORD2CLOUD: WASSERSTEIN WORD EMBEDDINGS

In this section, we focus on embedding of words to point clouds. In a sentence $s = (\mathbf{x}_0, \ldots, \mathbf{x}_n)$, a word $\mathbf{x}_i$ is associated with word $\mathbf{x}_j$ if $\mathbf{x}_j$ is in the context of $\mathbf{x}_i$. Here, *context* is defined as a symmetric window around $\mathbf{x}_i$ and the association is shown with labels r, that is, the label $r_{\mathbf{x}_i,\mathbf{x}_j} = 1$ if and only if $|i - j| \leq l$ where l is the window size. For word embedding, we used a contrastive loss function (Hadsell et al., 2006):

$$\mathcal{L}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{\mathbf{x}_i,\mathbf{x}_j} (1 - r_{\mathbf{x}_i,\mathbf{x}_j}) \mathcal{W}_1\big(\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\big) + r_{\mathbf{x}_i,\mathbf{x}_j} \Big(\Big[m - \mathcal{W}_1\big(\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\big) \Big]_+ \Big)^2, \quad (9)$$

which tries to embed two words $\mathbf{x}_i, \mathbf{x}_j$ close to each other in terms of Wasserstein distance (here $\mathcal{W}_1$) if they are in the context of each other and at least margin m away otherwise; this approach is similar to that suggested by Mikolov et al. (2013), up to the loss and distance functions.

We used a Siamese architecture (Bromley et al., 1993) for our model, with negative sampling (again like Mikolov et al. (2013)) for selecting words out of the context. The network architecture in each branch consists of a linear layer with 64 nodes followed by our point cloud embedding layer. The Wasserstein distance layer connects the two branches of the Siamese network. Training dataset is Text8[3] which is of moderate size but commonly used as a language modeling benchmark and consists of a text corpus with $17M$ tokens from Wikipedia. We used a vocabulary subset of size 8000 and a window size of $l = 2$ (i.e., 2 words on each side), $\lambda = 0.05$, number of epochs of 3, negative sampling rate of 1 and Adam (Kingma & Ba, 2014) for optimization.

Given a fixed number of parameters, first we study the effect of dimensionality of point clouds in the semantic neighborhood captured by the embeddings. Since the total number of parameters is fixed, increasing dimensionality means losing some points. Table 1 shows the 5 nearest neighbors using the distance measure trained shown in the first block. Here, the total number of parameters used is in $\{63, 64\}$. Interestingly, it is more effective to use a budget of 64 parameters in 16-point, 4-dimensional cloud than in a 32-point, 2-dimensional cloud. It is notable that changing the structure of embedding target in this way, can directly improve the quality of the representation learned.

Next we evaluate these models on a number of benchmark similarity tasks from (Faruqui & Dyer, 2014) to compare the quality of the embeddings in a retrieval task when we change the dimensionality of the point clouds with fixed total number of parameters. Results are shown in Table 2. Here, for consistency, we have picked the latent layer before our cloud layer and considered it as the embedding to be compared. The results of our method appear in the middle block of Table 2. The gradual improvement with increasing the dimensionality of the point clouds. The right block in Table 2 shows baselines: Respectively RNN(80D) (Kombrink et al., 2011), Metaoptimize (50D) (Turian et al., 2010), SENNA (50D) (Collobert, 2011) Global Context (50D) (Huang et al., 2012) and word2vec (80D) (Mikolov et al., 2013). The reported performance measure is based on (Faruqui & Dyer, 2014) computing the correlation with ground-truth of the rankings. All correlations are based on (Faruqui & Dyer, 2014). Note that there are many ways to improve the performance: increasing the vocabulary/window size/number of epochs/negative sampling rate, using larger texts, and accelerating performance. We defer this tuning to future work focused specifically on NLP.

[3]From `http://mattmahoney.net/dc/text8.zip`

$\mathcal{W}_1(\mathbb{R}^2)$	one:	f, two, i, after, four
	united:	series, professional, team, east, central
	algebra:	skin, specified, equation, hilbert, reducing
$\mathcal{W}_1(\mathbb{R}^3)$	one:	two, three, s, four, after
	united:	kingdom, australia, official, justice, officially
	algebra:	binary, distributions, reviews, ear, combination
$\mathcal{W}_1(\mathbb{R}^4)$	one:	six, eight, zero, two, three
	united:	army, union, era, treaty, federal
	algebra:	tables, transform, equations, infinite, differential

Table 1: Change in the 5-nearest neighbors when increasing dimensional of each point cloud with fixed total length of representation.

Task Name	# Pairs	# Found	$\mathcal{W}_1(\mathbb{R}^2)$ 17M	$\mathcal{W}_1(\mathbb{R}^3)$ 17M	$\mathcal{W}_1(\mathbb{R}^4)$ 17M	R —	M 63M	S 631M	G 900M	W 100B
RG-65	65	64	0.18	0.56	0.69	0.27	-0.02	0.50	0.66	0.54
Verb-143	143	144	0.12	0.14	0.29	0.29	0.06	0.36	0.44	0.27
WS-353	353	351	0.14	0.22	0.37	0.24	0.10	0.49	0.62	0.64
WS-353-S	203	201	0.19	0.35	0.47	0.36	0.15	0.61	0.70	0.70
WS-353-R	252	252	0.05	0.12	0.24	0.18	0.09	0.40	0.56	0.61
MC-30	30	30	-0.04	0.43	0.48	0.47	-0.14	0.57	0.66	0.63
Rare-Word	2034	1159	0.08	0.27	0.11	0.29	0.11	0.39	0.06	0.39
MEN	3000	2915	0.20	0.26	0.31	0.24	0.09	0.57	0.31	0.65
MTurk-287	287	284	0.30	0.30	0.43	0.33	0.09	0.59	0.36	0.67
MTurk-771	771	770	0.10	0.24	0.27	0.26	0.10	0.50	0.32	0.57
SimLex-999	999	998	0.06	0.09	0.13	0.23	0.01	0.27	0.10	0.31

Table 2: Performance on a number of similarity benchmarks when dimensionality of point clouds increase given a fixed total number of parameters. The middle block shows the performance of the proposed models. The training corpus size when known appears below each model name.

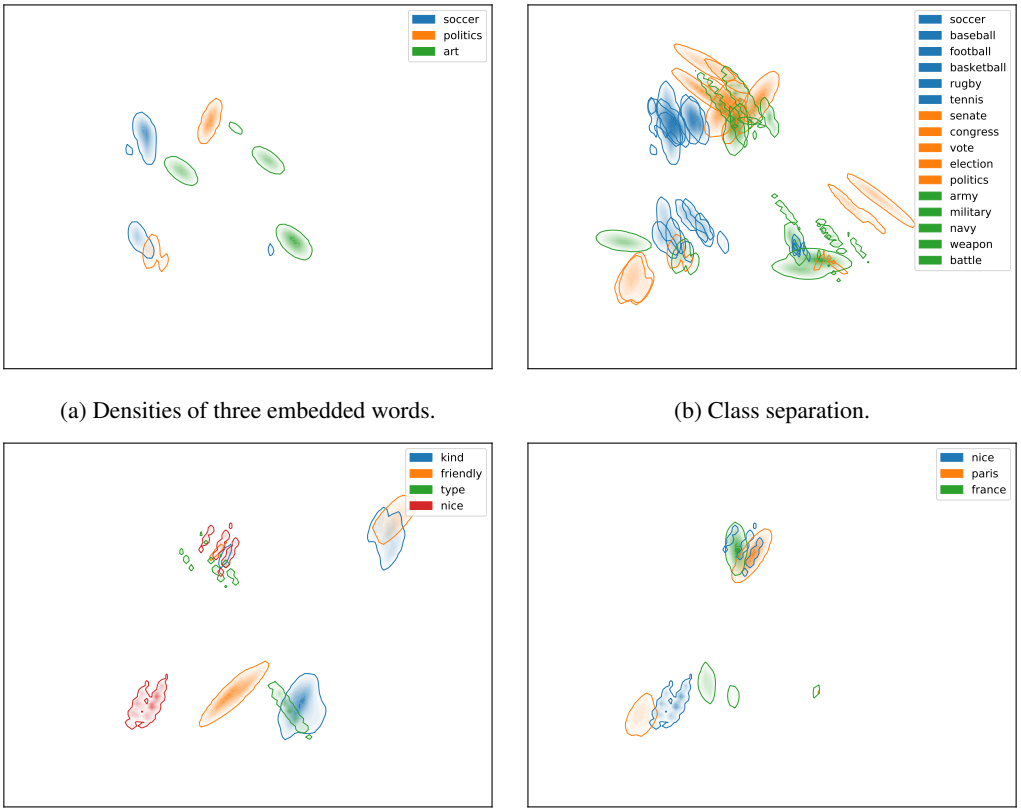
4.2.1 DIRECT, INTERPRETABLE VISUALIZATION OF HIGH-DIMENSIONAL EMBEDDINGS

Wasserstein embeddings over low-dimensional ground metric spaces have a unique property: We can **directly visualize the embedding**, which is a point cloud in the low-dimensional ground space. This is not true for most existing embedding methods, which rely on dimensionality reduction techniques such as t-SNE for visualizing the embedded data. Whereas dimensionality reduction only approximately captures closeness of points in the embedding space, with Wasserstein embeddings we can see the **exact embedding** of each input, by visualizing the point cloud.

We demonstrate this capability by visualizing the learned word representations. Importantly, the point clouds strongly cluster, which leads to apparent, distinct modes in their density. We therefore use kernel density estimation to visualize their densities. In Figure 3a, we visualize three distinct words, thresholding each density at a low value and visualizing its upper level set, which reveals the modes. These level sets are overlaid, with each color in the the figure corresponding to a distinct embedded word. The density for each word is depicted by the opacity of the color within each level set.

We first note that it is easy to visualize in aggregate multiple sets of words, by assigning all words within a set a single color. This immediately reveals how well-separated are the sets. This is shown in Figure 3b: as expected, there is some overlap between military and political terms, while names of sports are separated from the rest.

Looking at the embeddings in more detail, we can start to dissect relationships (and confusion) between different sets of words. We observe that each word tends to concentrate its mass in two or more distinct regions. This multimodal nature of the density allows for multifaceted relationships between words, as a word can partially overlap with many distinct groups of words simultaneously. Figure 3c shows the embedding for a word that has multiple distinct meanings (`kind`), alongside synonyms for both senses of the word (`nice, friendly, type`). We see that `kind` has two primary modes, which overlap separately with `friendly` and `type`. `nice` is included to show a failure of the embedding to capture the full semantics – Figure 3d shows that the network has in fact



(a) Densities of three embedded words.

(b) Class separation.

(c) Word with multiple meanings: `kind`.

(d) Debugging a failed association: `nice`.

Figure 3: Directly visualizing high-dimensional word embeddings.

learned that `nice` is a city in France, while ignoring its second interpretation. This demonstrates the potential of this visualization for debugging our network.

5 Discussion and Conclusion

Several desiderata determine the value and effectiveness of an embedding space for representation learning. The space must be large enough to embed a variety of metrics, while admitting a mathematical description compatible with learning algorithms; additional features including direct interpretability make it easier to understand, analyze, and potentially debug the output of a representation learning procedure. Based on its theoretical properties and performance across a variety of modalities, we find that Wasserstein space is a strong candidate worthy of consideration when the capacity of Euclidean space does not suffice.

Several extensions and directions for additional research are suggested by our work. Beyond simple extensions like weighting points in the point cloud, one observation is that we can lift nearly *any* representation space $\mathcal{X}$ to distributions over that space $\mathcal{W}(\mathcal{X})$ represented as point clouds; in this paper we focused on the case $\mathcal{X} = \mathbb{R}^n$. Since $\mathcal{X}$ embeds within $\mathcal{W}(\mathcal{X})$ using δ-functions, this might be viewed as a general "lifting" procedure increasing the capacity of a representation. We can also consider other tasks, such as co-embedding of different modalities into the same transport space. Additional theoretical study may also reveal which properties of a metric space make it amenable to embedding into Wasserstein space, or into peers such as unbalanced transport (Chizat et al., 2015) or versions involving quadratic terms (Peyré et al., 2016; Ezuz et al., 2017).

REFERENCES

Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47, 2002.

Alexandr Andoni, Assaf Naor, and Ofer Neiman. Snowflake universality of wasserstein spaces. *arXiv preprint arXiv:1509.08677*, 2015.

Alexandr Andoni, Assaf Naor, and Ofer Neiman. Impossibility of sketching of the 3d transportation metric with quadratic cost. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 55. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pp. 214–223, 2017.

Ben Athiwaratkun and Andrew Gordon Wilson. On modeling hierarchical data via probabilistic order embeddings. In *ICLR*, 2018.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *TACL*, 5:135–146, 2017.

Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *ICLR*, 2018.

Alessio Brancolini, Giuseppe Buttazzo, Filippo Santambrogio, and Eugene Stepanov. Long-term planning versus short-term planning in the asymptotical location problem. *ESAIM: Control, Optimisation and Calculus of Variations*, 15(3):509–524, 2009.

Jane Bromley, James W. Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. Signature verification using A "Siamese" time delay neural network. *IJPRAI*, 7(4):669–688, 1993.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: geometry and Kantorovich formulation. *arXiv:1508.05216*, 2015.

Sebastian Claici, Edward Chien, and Justin Solomon. Stochastic Wasserstein barycenters. In *ICML*, 2018.

Ronan Collobert. Deep learning for efficient discriminative parsing. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, 2011.

Nicolas Courty, Rémi Flamary, and Devis Tuia. Domain adaptation with regularized optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 274–289. Springer, 2014.

Nicolas Courty, Rémi Flamary, and Mélanie Ducoffe. Learning Wasserstein embeddings. In *ICLR*, 2018.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, 2013.

Michel Marie Deza and Monique Laurent. *Geometry of Cuts and Metrics*. 2009.

Danielle Ezuz, Justin Solomon, Vladimir G Kim, and Mirela Ben-Chen. GWCNN: A metric alignment layer for deep shape analysis. In *CGF*, volume 36, pp. 49–57, 2017.

Manaal Faruqui and Chris Dyer. Community evaluation and exchange of word vectors at wordvectors.org. In *Association for Computational Linguistics: System Demonstrations*, June 2014.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso A. Poggio. Learning with a Wasserstein loss. In *NIPS*, 2015.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with Sinkhorn divergences. In *AISTATS*, 2018.

Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *KDD*, 2016.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pp. 873–882. Association for Computational Linguistics, 2012.

Piotr Indyk and Nitin Thaper. Fast Image Retrieval via Embeddings. In *3rd International Workshop on Statistical and Computational Theories of Vision*. ICCV, 2003.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Ryan Kiros, Richard S. Zemel, and Ruslan Salakhutdinov. A multiplicative model for learning distributed text-based attribute representations. In *NIPS*, 2014.

Benoit Kloeckner. Approximation by finitely supported measures. *ESAIM: Control, Optimisation and Calculus of Variations*, 18(2):343–359, 2012.

Benoˆıt Kloeckner. A geometric study of wasserstein spaces: embedding powers. 2010.

Stefan Kombrink, Tomas Mikolov, Martin Karafiát, and Lukás Burget. Recurrent neural network based language modeling in meeting recognition. In *Annual Conference of the International Speech Communication Association INTERSPEECH*, 2011.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *ICML*, pp. 957–966, 2015.

Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*. 2013.

Farzaneh Mirzazadeh. *Solving Association Problems with Convex Co-embedding*. PhD thesis, University of Alberta, 2017.

Farzaneh Mirzazadeh, Yuhong Guo, and Dale Schuurmans. Convex co-embedding. In *AAAI*, 2014.

Farzaneh Mirzazadeh, Siamak Ravanbakhsh, Nan Ding, and Dale Schuurmans. Embedding inference for structured multilabel prediction. In *NIPS*, 2015.

Boris Muzellec and Marco Cuturi. Generalizing point embeddings using the Wasserstein space of elliptical distributions. In *NIPS*, 2018.

Graham Neubig, Matthias Sperber, Xinyi Wang, Matthieu Felix, Austin Matthews, Sarguna Padmanabhan, Ye Qi, Devendra Singh Sachan, Philip Arthur, Pierre Godard, John Hewitt, Rachid Riad, and Liming Wang. XNMT: the extensible neural machine translation toolkit. In *AMTA (1)*, pp. 185–192. Association for Machine Translation in the Americas, 2018.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *NIPS*, 2017.

Maximilian Nickel and Douwe Kiela. Learning continuous hierarchies in the Lorentz model of hyperbolic geometry. In *ICML*, 2018.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proc. IEEE*, 104(1):11–33, 2016.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, pp. 1532–1543, 2014.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL-HLT*, 2018.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. Technical report, 2017.

Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-Wasserstein averaging of kernel and distance matrices. In *ICML*, pp. 2664–2672, 2016.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. A metric for distributions with applications to image databases. In *ICCV*, 1998.

Sameer Shirdhonkar and David W Jacobs. Approximate earth mover's distance in linear time. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8. IEEE, 2008.

Justin Solomon. Optimal transport on discrete domains. *CoRR*, abs/1801.07745, 2018.

Joseph P. Turian, Lev-Arie Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association*, 2010.

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. In *ICLR*, 2015.

Cédric Villani. *Topics in Optimal Transportation*. Number 58. American Mathematical Soc., 2003.

Cédric Villani. *Optimal Transport: Old and New*, volume 338. 2008.

Luke Vilnis and Andrew McCallum. Word representations via Gaussian embedding. In *ICLR*, 2015.

Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440, 1998.

Jason Weston, Samy Bengio, and Nicolas Usunier. WSABIE: scaling up to large vocabulary image annotation. In *IJCAI*, 2011.

Xiaohan Zhao, Alessandra Sala, Haitao Zheng, and Ben Y Zhao. Fast and scalable analysis of massive social graphs. *arXiv preprint arXiv:1107.5114*, 2011.

Dingyuan Zhu, Peng Cui, Daixin Wang, and Wenwu Zhu. Deep variational network embedding in Wasserstein space. In *KDD*, 2018.