Optimal Transport Graph Neural Networks

Gary Bécigneul* **Octavian-Eugen Ganea*** **Benson Chen***
Regina Barzilay **Tommi Jaakkola**
Computer Science and Artificial Intelligence Lab, MIT
Correspondence at {garyb,oct,bensonc}@mit.edu

Abstract

Current graph neural network (GNN) architectures naively average or sum node embeddings into an aggregated graph representation—potentially losing structural or semantic information. We here introduce OT-GNN that compute graph embeddings from optimal transport distances between the set of GNN node embeddings and "prototype" point clouds as free parameters. This allows different prototypes to highlight key facets of different graph subparts. We show that our function class on point clouds satisfies a universal approximation theorem, a fundamental property which was lost by sum aggregation. Nevertheless, empirically the model has a natural tendency to collapse back to the standard aggregation during training. We address this optimization issue by proposing an efficient noise contrastive regularizer, steering the model towards truly exploiting the optimal transport geometry. Our model consistently exhibits better generalization performance on several molecular property prediction tasks, yielding also smoother representations.[2]

1 Introduction

Recently, there has been considerable interest in developing learning algorithms for structured data such as graphs. For example, molecular property prediction has many applications in chemistry and drug discovery [39, 35]. Historically, graphs were systematically decomposed into features such as molecular fingerprints, turned into non-parametric graph kernels [36, 32], or, more recently, learned representations via graph neural networks (GNNs) [11, 10, 21].

Despite successes, graph neural networks are often underutilized in whole graph prediction tasks such as molecule property prediction. Specifically, while GNNs produce node embeddings for each atom in the molecule, these are typically aggregated via simple operations such as a sum or average, turning the molecule into a single vector prior to classification or regression. As a result, some of the information naturally extracted by node embeddings may be lost.

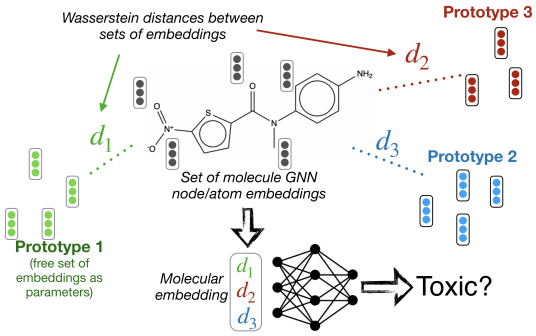


Figure 1: *Intuition for our Wasserstein prototype model. We assume that a few prototypes, e.g. some functional groups, highlight key facets or structural features of graphs in a particular graph classification/regression task at hand. We then express graphs by relating them to these abstract prototypes represented as free point cloud parameters. Note that we do not learn the graph structure of the prototypes.*

*Equal Contribution.
[2]Code available at https://github.com/benatorc/OTGNN.

Preprint. Under review.

Recent work by Togninalli et al. [34] proposed to dispense with the aggregation step altogether and instead derive a kernel function over graphs by directly comparing node embeddings as point clouds through optimal transport (Wasserstein distance). Their *non-parametric* model yields better empirical performance over popular graph kernels, but haven't been so far extended to the more challenging parametric case.

Motivated by this observation and drawing inspiration from prototypical networks [33], we introduce a new class of graph neural networks where the key representational step consists of comparing each input graph to a set of abstract prototypes (fig. 1). These prototypes play the role of dictionary items or basis functions in the comparison; they are also stored as point clouds as if they were encoded from actual real graphs. Each input graph is first encoded into a set of node embeddings using a GNN. We then compare this resulting embedding point cloud to those corresponding to the prototypes. Formally, the distance between two point clouds is measured by appeal to optimal transport Wasserstein distances. The prototypes as abstract basis functions can be understood as keys that highlight property values associated with different structural features. In contrast to kernel methods, the prototypes are learned together with the GNN parameters in an end-to-end manner.

Our model improves upon traditional aggregation by explicitly tapping into the full set of node embeddings without collapsing them first to a single vector. We theoretically prove that, unlike standard GNN aggregation, our model defines a class of set functions that is universal approximator.

Introducing points clouds as free parameters creates a challenging optimization problem. Indeed, as the models are trained end-to-end, the primary signal is initially available only in aggregate form. If trained as is, the prototypes often collapse to single points, reducing the Wasserstein distance between point clouds into Euclidean comparisons of their means. To counter this effect, we introduce a contrastive regularizer which effectively prevents the model from collapsing (Section 3.2). We demonstrate empirically that it both improves model performance and generates richer prototypes.

Our contributions are summarized as follows. First, we introduce an efficiently trainable class of graph neural networks enhanced with optimal transport (OT) primitives for computing graph representations. Second, we devise a principled noise contrastive regularizer to prevent the model from collapsing back to standard aggregation, thus fully exploiting the OT geometry. Third, we provide a mathematical justification of the increased representational power compared to standard aggregation methods used in popular GNNs. Finally, our model shows consistent empirical improvements over previous state-of-the-art on molecular datasets, yielding also smoother graph embedding spaces.

2 Preliminaries

2.1 Directed Message Passing Neural Networks

We briefly remind here of the simplified D-MPNN [9] architecture which was successfully used for molecular property prediction by Yang et al. [39].

This model takes as input a directed graph $G = (V, E)$, with node and edge features denoted by $\mathbf{x}_v$ and $\mathbf{e}_{vw}$ respectively, for v, w in the vertex set V and when $v \to w$ is an edge in E. The parameters of D-MPNN are the below weight matrices $\{\mathbf{W}_i, \mathbf{W}_m, \mathbf{W}_o\}$.

It keeps track of *messages* $\mathbf{m}^t_{vw}$ and *hidden states* $\mathbf{h}^t_{vw}$ for each step t, defined as follows. An initial hidden state is set to $\mathbf{h}^0_{vw} := ReLU(\mathbf{W}_i \mathrm{cat}(\mathbf{x}_v, \mathbf{e}_{vw}))$ where "cat" denotes concatenation. Then, the *message passing* operates as

$$\mathbf{m}^{t+1}_{vw} = \sum_{k \in N(v) \setminus \{w\}} \mathbf{h}^t_{kv}, \quad \mathbf{h}^{t+1}_{vw} = ReLU(\mathbf{h}^0_{vw} + \mathbf{W}_m \mathbf{m}^{t+1}_{vw}), \tag{1}$$

where $N(v) = \{k \in V | (k, v) \in E\}$ denotes v's incoming neighbors. After T steps of message passing, node embeddings are obtained by summing edge embeddings:

$$\mathbf{m}_v = \sum_{w \in N(v)} \mathbf{h}^T_{vw}, \quad \mathbf{h}_v = ReLU(\mathbf{W}_o \mathrm{cat}(\mathbf{x}_v, \mathbf{m}_v)). \tag{2}$$

A final graph embedding is then obtained as $\mathbf{h} = \sum_{v \in V} \mathbf{h}_v$, which is usually fed to a multilayer perceptron (MLP) for classification or regression.

2.2 Optimal Transport Geometry

Optimal Transport (OT) is a mathematical framework that defines distances or similarities between objects such as probability distributions, either discrete or continuous, as the cost of an optimal transport plan from one to the other.

Wasserstein for point clouds. Let a *point cloud* $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ *of size* n be a set of n points $\mathbf{x}_i \in \mathbb{R}^d$. Given point clouds $\mathbf{X}, \mathbf{Y}$ of respective sizes n, m, a ***transport plan*** (or ***coupling***) is a matrix $\mathbf{T}$ of size $n \times m$ with entries in $[0, 1]$, satisfying the two following *marginal constraints*: $\mathbf{T}\mathbf{1}_m = \frac{1}{n}\mathbf{1}_n$ and $\mathbf{T}^T\mathbf{1}_n = \frac{1}{m}\mathbf{1}_m$. Intuitively, the marginal constraints mean that $\mathbf{T}$ preserves the mass from $\mathbf{X}$ to $\mathbf{Y}$. We denote the set of such couplings as $\mathcal{C}_{\mathbf{XY}}$.

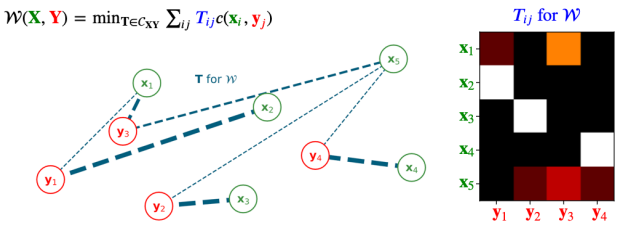


Figure 2: We illustrate, for a given 2D point cloud, the optimal transport plan obtained from minimizing the Wasserstein costs; $c(\cdot, \cdot)$ denotes the Euclidean distance. A higher dotted-line thickness illustrates a greater mass transport.

Given a cost function c on $\mathbb{R}^d$, its associated ***Wasserstein discrepancy*** is defined as

$$\mathcal{W}(\mathbf{X}, \mathbf{Y}) = \min_{\mathbf{T}\in\mathcal{C}_{\mathbf{XY}}} \sum_{ij} T_{ij} c(\mathbf{x}_i, \mathbf{y}_j). \tag{3}$$

We further describe the shape of optimal transports for point clouds of same sizes in Appendix B.3.

3 Model & Practice

3.1 Architecture Enhancement

Reformulating standard architectures. As mentioned at the end of Section 2.1, the final graph embedding $\mathbf{h}$ obtained by aggregating node embeddings is usually fed to a MLP performing a matrix-multiplication $(\mathbf{R}\mathbf{h})_i = \langle \mathbf{r}_i, \mathbf{h} \rangle$. Replacing $\langle \cdot, \cdot \rangle$ by a distance/kernel $k(\cdot, \cdot)$ allows the processing of more general graph representations than just vectors in $\mathbb{R}^d$, such as point clouds or adjacency tensors.

From a single point to a point cloud. We propose to replace the aggregated graph embedding $\mathbf{h} = \sum_{v\in V} \mathbf{h}_v$ by the point cloud (of unaggregated node embeddings) $\mathbf{H} = \{\mathbf{h}_v\}_{v\in V}$, and the inner-products $\langle \mathbf{h}, \mathbf{r}_i \rangle$ by the below written ***Wasserstein discrepancy***:

$$\mathcal{W}(\mathbf{H}, \mathbf{Q}_i) := \min_{\mathbf{T}\in\mathcal{C}_{\mathbf{HQ}_i}} \sum_{vj} T_{vj} c(\mathbf{h}_v, \mathbf{q}_i^j), \tag{4}$$

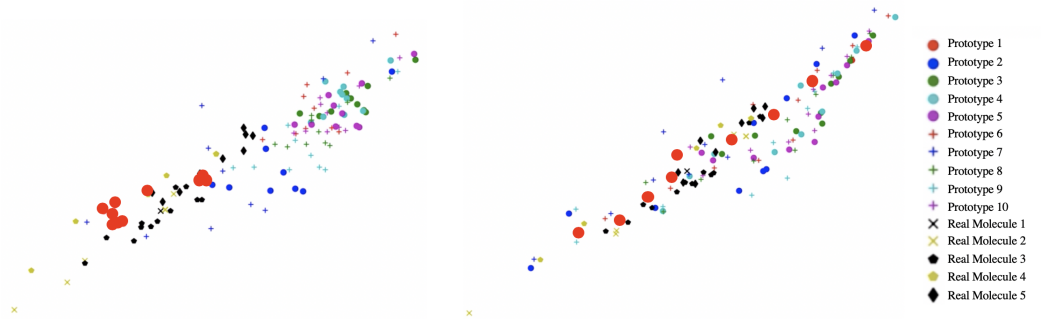
where the $\mathbf{Q}_i = \{\mathbf{q}_i^j\}_j$ are point clouds and free parameters, and the cost is chosen as $c = \| \cdot - \cdot \|_2^2$ or $c = -\langle \cdot, \cdot \rangle$. Note that both options yield identical optimal transport plans.

Greater representational power. We formulate mathematically in Section 4 to what extent this kernel has a strictly greater representational power than the kernel corresponding to standard inner-product on top of a sum aggregation, to distinguish between different point clouds. In practice, we would also like our model to exploit its additional representational power. This practical concern is discussed in the next subsection.

3.2 Contrastive Regularization

What would happen to $\mathcal{W}(\mathbf{H}, \mathbf{Q}_i)$ if all points $\mathbf{q}_i^j$ belonging to point cloud $\mathbf{Q}_i$ would collapse to the same point $\mathbf{q}_i$? All transport plans would yield the same cost, giving for $c = -\langle \cdot, \cdot \rangle$:

$$\mathcal{W}(\mathbf{H}, \mathbf{Q}_i) = -\sum_{vj} T_{vj} \langle \mathbf{h}_v, \mathbf{q}_i^j \rangle = -\langle \mathbf{h}, \mathbf{q}_i / |V| \rangle. \tag{5}$$



(a) No contrastive regularization (b) Using contrastive regularization (weight 0.1)

Figure 3: 2D embeddings of prototypes and of real molecule samples with (right) and without (left) contrastive regularization *for runs using the exact same random seed*. Points in the prototypes tend to cluster and collapse more when no regularization is used, implying that the optimal transport plan is no longer uniquely discriminative. Prototype 1 (red) is enlarged for clarity: without regularization (left), it is clumped together, but with regularization (right), it is distributed across the space.

In this scenario, our proposition would simply over-parametrize the standard Euclidean model.

A first obstacle. Our first empirical trials with OT-enhanced GNNs showed that a model trained with only the Wasserstein component would sometimes perform similarly to the Euclidean baseline, in spite of its greater representational power. Since these two models achieved both similar test *and* train performance, the absence of improvement in generalization was most likely not due to overfitting.

The cause. Further investigation revealed that the Wasserstein model would naturally displace the points in each of its free point clouds in such a way that the optimal transport plan $\mathbf{T}$ obtained by maximizing $\sum_{vj} T_{vj} \langle \mathbf{h}_v, \mathbf{q}_i^j \rangle$ was not *discriminative*, *i.e.* many other transports would yield a similar Wasserstein cost. Indeed, as shown in Eq. (5), if each point cloud collapses to its mean, then the Wasserstein geometry collaspses to Euclidean geometry. In this scenario, any transport plan yields the same Wasserstein cost. Further explanations are provided in Appendix A.1.

Contrastive regularization. This observation has lead us to consider the use of a regularizer which would encourage the model to displace its free point clouds such that the optimal transport plans it computes would be discriminative against chosen *contrastive transport plans*. Namely, consider a point cloud $\mathbf{Y}$ of node embeddings and let $\mathbf{T}^i$ be an optimal transport plan obtained in the computation of $\mathcal{W}(\mathbf{Y}, \mathbf{Q}_i)$; for each $\mathbf{T}^i$, we then build a set $N(\mathbf{T}^i) \subset \mathcal{C}_{\mathbf{Y}\mathbf{Q}_i}$ of *noisy/contrastive* transports. If we denote by $\mathcal{W}_{\mathbf{T}}(\mathbf{X}, \mathbf{Y}) := \sum_{kl} T_{kl} c(\mathbf{x}_k, \mathbf{y}_l)$ the Wasserstein cost obtained for the particular transport $\mathbf{T}$, then our contrastive regularization consists in maximizing the term:

$$\sum_i \log \left(\frac{e^{-\mathcal{W}_{\mathbf{T}^i}(\mathbf{Y}, \mathbf{Q}_i)}}{e^{-\mathcal{W}_{\mathbf{T}^i}(\mathbf{Y}, \mathbf{Q}_i)} + \sum_{\mathbf{T} \in N(\mathbf{T}^i)} e^{-\mathcal{W}_{\mathbf{T}}(\mathbf{Y}, \mathbf{Q}_i)}} \right), \tag{6}$$

which can be interpreted as the log-likelihood that the correct transport $\mathbf{T}_i$ be (as it should) a better minimizer of $\mathcal{W}_{\mathbf{T}}(\mathbf{Y}, \mathbf{Q}_i)$ than its negative samples. This can be considered as an approximation of $\log(\Pr(\mathbf{T}_i \mid \mathbf{Y}, \mathbf{Q}_i))$, where the partition function is approximated by our selection of negative examples, as done e.g. by Nickel & Kiela [26]. Its effect of is shown in Figure 3.

Remarks. The selection of negative examples must reflect the following trade-off: *(i)* to not be too large, for computational efficiency while *(ii)* containing sufficiently meaningful and challenging contrastive samples. Details about choice of contrastive samples are exposed in Appendix A.2. Note that replacing the set $N(\mathbf{T}^i)$ with a singleton $\{\mathbf{T}\}$ for a contrastive random variable $\mathbf{T}$ would let us rewrite Eq. (6) as[3] $\sum_i \log \sigma(\mathcal{W}_{\mathbf{T}} - \mathcal{W}_{\mathbf{T}^i})$, reminiscent of noise contrastive estimation [17].

[3] where $\sigma(\cdot)$ is the sigmoid function.

3.3 Optimization & Complexity

Backpropagating gradients through optimal transport (OT) has been the subject of recent research investigations: Genevay et al. [14] explain how to unroll and differentiate through the Sinkhorn procedure solving OT, which was extended by Schmitz et al. [31] to Wasserstein barycenters. However, more recently, Xu [37] proposed to simply invoke the envelop theorem [1] to support the idea of keeping the optimal transport plan fixed during the back-propagation of gradients through Wasserstein distances. *For the sake of simplicity and training stability, we resort to the latter procedure: keeping* $\mathbf{T}$ *fixed during back-propagation.* We discuss complexity in appendix C.

4 Theoretical Analysis

In this section we show that the standard architecture lacks a fundamental property of *universal approximation* of functions defined on point clouds, and that our proposed architecture recovers this property. We will denote by $\mathcal{X}_d^n$ the set of point clouds $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ of size n in $\mathbb{R}^d$.

4.1 Universality

As seen in Section 3.1, we have replaced the sum aggregation $-$ followed by the Euclidean inner-product $-$ by Wasserstein discrepancies. How does this affect the function class and representations?

A common framework used to analyze the geometry inherited from similarities and discrepancies is that of kernel theory. A kernel k on a set $\mathcal{X}$ is a symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, which can either measure similarities or discrepancies. An important property of a given kernel on a space $\mathcal{X}$ is whether simple functions defined on top of this kernel can approximate any continuous function on the same space. This is called *universality*: a crucial property to regress unknown target functions.

Universal kernels. A kernel k defined on $\mathcal{X}_d^n$ is said to be ***universal*** if the following holds: for any compact subset $\mathcal{X} \subset \mathcal{X}_d^n$, the set of functions in the form[4] $\sum_{j=1}^m \alpha_j \sigma(k(\cdot, \theta_j) + \beta_j)$ is dense in the set $\mathcal{C}(\mathcal{X})$ of continuous functions from $\mathcal{X}$ to $\mathbb{R}$, w.r.t the sup norm $\|\cdot\|_{\infty,\mathcal{X}}$, σ denoting the sigmoid. Although the notion of universality does not indicate how easy it is in practice to learn the correct function, it at least guarantees the absence of a fundamental bottleneck of the model using this kernel.

Theorem 1. *We have that:*

1. *The aggregation kernel* $\mathtt{agg}$ ***is not universal****.*
2. *The Wasserstein kernel* $\mathcal{W}_{\mathrm{L2}}$ *defined in Theorem 2* ***is universal****.*

Proof: See Appendix B.1.

Universality of the Wasserstein kernel $\mathcal{W}_{\mathrm{L2}}$ essentially comes from the fact that its square-root defines a metric, and in particular from the axiom of separation of distances: *if* $d(x, y) = 0$ *then* $x = y$.

4.2 Definiteness

For the sake of simplified mathematical analysis, similarity kernels are often required to be *positive definite* (p.d.), which corresponds to discrepancy kernels being *conditionally negative definite* (c.n.d.). Although such a property has the benefit of yielding the mathematical framework of Reproducing Kernel Hilbert Spaces, it essentially implies *linearity*, *i.e.* the possibility to embed the geometry defined by that kernel in a linear vector space.

We now show that, interestingly, the Wasserstein kernel we used does not satisfy this property, and hence constitutes an interesting instance of a universal, non p.d. kernel. Let us remind these notions.

Kernel definiteness. A kernel k is ***positive definite (p.d.)*** on $\mathcal{X}$ if for $n \in \mathbb{N}^*$, $x_1, ..., x_n \in \mathcal{X}$ and $c_1, ..., c_n \in \mathbb{R}$, we have $\sum_{ij} c_i c_j k(x_i, x_j) \geq 0$. It is ***conditionally negative definite (c.n.d.)*** on $\mathcal{X}$ if for $n \in \mathbb{N}^*$, $x_1, ..., x_n \in \mathcal{X}$ and $c_1, ..., c_n \in \mathbb{R}$ such that $\sum_i c_i = 0$, we have $\sum_{ij} c_i c_j k(x_i, x_j) \leq 0$.

These two notions relate to each other via the below result [4]:

[4] For $m \in \mathbb{N}^*$, $\alpha_j \beta_j \in \mathbb{R}$ and $\theta_j \in \mathcal{X}_d^n$.

Proposition 1. *Let k be a symmetric kernel on $\mathcal{X}$, let $x_0 \in \mathcal{X}$ and define the kernel:*

$$\tilde{k}(x, y) := -\frac{1}{2}[k(x, y) - k(x, x_0) - k(y, x_0) + k(x_0, x_0)]. \tag{7}$$

*Then $\tilde{k}$ **is p.d. if and only if** k **is c.n.d.** Example: $k = \| \cdot - \cdot \|_2^2$ and $x_0 = \mathbf{0}$ yield $\tilde{k} = \langle \cdot, \cdot \rangle$.*

The aggregating kernel against which we wish to compare the Wasserstein kernel is the inner-product over a summation of the points in the point clouds: $\mathtt{agg}(\mathbf{X}, \mathbf{Y}) := \langle \sum_i \mathbf{x}_i, \sum_j \mathbf{y}_j \rangle$.

One can easily show that this also defines a p.d. kernel, and that $\mathtt{agg}(\cdot, \cdot) \leq n^2 \mathcal{W}(\cdot, \cdot)$. However, the Wasserstein kernel is not p.d., as shown by the below theorem which we prove in Appendix B.2.

Theorem 2. *We have that:*

1. *The (similarity) Wasserstein kernel $\mathcal{W}_{\mathrm{dot}}$ **is not positive definite**;*
2. *The (discrepancy) Wasserstein kernel $\mathcal{W}_{\mathrm{L2}}$ **is not conditionally negative definite**, where:*

$$\mathcal{W}_{\mathrm{L2}}(\mathbf{X}, \mathbf{Y}) := \min_{\mathbf{T} \in \mathbf{XY}} \sum_{ij} T_{ij} \|\mathbf{x}_i - \mathbf{y}_j\|_2^2, \; \mathcal{W}_{\mathrm{dot}}(\mathbf{X}, \mathbf{Y}) := \max_{\mathbf{T} \in \mathbf{XY}} \sum_{ij} T_{ij} \langle \mathbf{x}_i, \mathbf{y}_j \rangle. \tag{8}$$

5 Experiments

5.1 Experimental Setup

We test our model on 4 benchmark molecular property prediction datasets [39] including both regression (ESOL, Lipophilicity) and classification (BACE, BBBP) tasks. These datasets cover a variety of different complex chemical properties (e.g. ESOL - water solubility, LIPO - octanol/water distribution coefficient, BACE - inhibition of human β-secretase, BBBP - blood-brain barrier penetration). We show that our models improves over state-of-the-art baselines.

GNN is the state-of-the-art graph neural network that we use as our primary baseline, as well as the underlying graph model for our prototype models. Its architecture is described in section 2.1.

ProtoW-L2/Dot is the model that treats point clouds as point sets, and computes the Wasserstein distances to each point cloud (using either L2 distance or (minus) dot product cost functions) as the molecular embedding.

ProtoS-L2 is a special case of **ProtoW-L2**, in which the point clouds have a *single* point. Instead of using Wasserstein distances, we instead just compute simple Euclidean distances between the aggregated graph embedding and point clouds. Here, we omit using dot product distances, as that model is mathematically equivalent to the GNN model.

We use the the POT library [13] to compute Wasserstein distances using the network simplex algorithm (ot.emd), which we find empirically to be faster than using the Sinkhorn algorithm, due to the small size of the graphs present in our datasets. We define the cost matrix by taking the pairwise L2 or negative dot product distances. As mentioned in Section 3.3, we fix the transport plan, and only backprop through the cost matrix for computational efficiency. Additionally, since the sum aggregation operator easily accounts for the sizes of input graphs, we multiply the OT distance between two point clouds by their respective sizes. To avoid the problem of point clouds collapsing, we employ the contrastive regularizer defined in Section 3.2. More details about experimental setup in Appendix D.1. We also tried extensions to our prototype models using Gromov-Wasserstein geometry. However, we found that these models proved much more difficult to optimize in practice.

5.2 Experimental Results

5.2.1 Regression and Classification

The results on the property prediction datasets are shown in Table 1. We find that the prototype models outperform the GNN on all 4 property prediction tasks, showing that this model paradigm can be more powerful than conventional GNN models. Moreover, the prototype models using Wasserstein distance (**ProtoW-L2/Dot**) achieves better performance on 3 out of 4 of the datasets compared to

Table 1: Results of different models on the property prediction datasets. **Best** in bold, second best underlined. Proto methods are ours. Lower RMSE is better, while higher AUC is better. The prototype-based models generally outperform the GNN, and the Wasserstein models perform better than the model using only simple Euclidean distances, suggesting that the Wasserstein distance provides more powerful representations. Wasserstein models trained with contrastive regularization as described in section 3.2 outperform those without.

# graphs	ESOL (RMSE) $n = 1128$	Lipo (RMSE) $n = 4199$	BACE (AUC) $n = 1512$	BBBP (AUC) $n = 2039$
GNN/Chemprop	$.635 \pm .027$	$.646 \pm .041$	$.865 \pm .013$	$.915 \pm .010$
ProtoS-L2	$.611 \pm .034$	$\mathbf{.580 \pm .016}$	$.865 \pm .010$	$.918 \pm .009$
ProtoW-Dot *(no reg.)*	$.608 \pm .029$	$.637 \pm .018$	$.867 \pm .014$	$.919 \pm .009$
ProtoW-Dot	$\mathbf{.594 \pm .031}$	$.629 \pm .015$	$.871 \pm .014$	$.919 \pm .009$
ProtoW-L2 *(no reg.)*	$.616 \pm .028$	$.615 \pm .025$	$\underline{.870 \pm .012}$	$\underline{.920 \pm .010}$
ProtoW-L2	$\underline{.605 \pm .029}$	$\underline{.604 \pm .014}$	$\mathbf{.873 \pm .015}$	$\mathbf{.920 \pm .010}$

the prototype model using only Euclidean distances (**ProtoS-L2**). This confirms our hypothesis that Wasserstein distance confers greater discriminative power compared to traditional aggregation methods (summation).

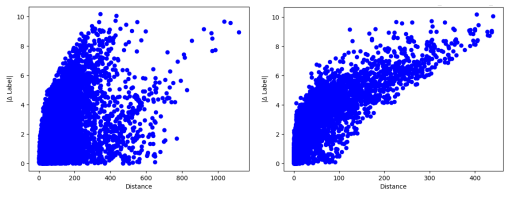
5.2.2 Noise Contrastive Regularizer

Without any constraints, the Wasserstein prototype model will often collapse the set of points in a point cloud into a single point. As mentioned in Section 3.2, we use a contrastive regularizer to force the model to meaningfully distribute point clouds in the embedding space. We show 2D embeddings in Fig. 3, illustrating that without contrastive regularization, prototype point clouds are often displaced close to their mean, while regularization forces them to nicely scatter.

5.2.3 Learned Embedding Space: Qualitative and Quantitative Results

To further support our claim that Wasserstein distance provides more powerful representations, we also examine the embedding space of the GNN baseline and our Wasserstein model. Using the best performing models, we compute the pairwise difference in embedding vectors and the labels for each test data point on the ESOL dataset. Then, we compute two measures of rank correlation, Spearman correlation coefficient (ρ) and Pearson correlation coefficient (r). This procedure is reminiscent of evaluation tasks for word embeddings w.r.t how semantic similarity in embedding space correlates with human labels [24].

Our ProtoW-L2 achieves better ρ and r scores compared to the GNN model (Table 2), that indicating our Wasserstein model constructs more meaningful embeddings with respect to the label distribution. Indeed, Figure 4 plots the pairwise scores for the GNN model (left) and the ProtoW-L2 model (right). Our ProtoW-L2 model, trained to optimize distances in the embedding space, produces more meaningful representations with respect to the label of interest.

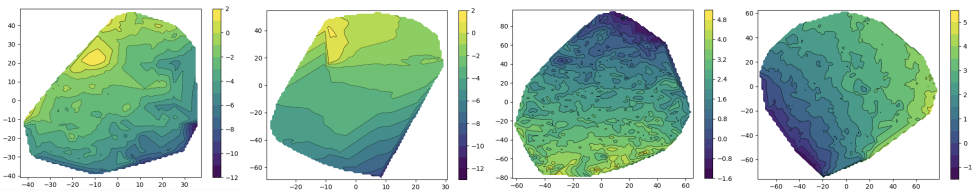


(a) GNN model (b) ProtoW-L2 model

Figure 4: Comparison of the correlation between graph embedding distances (X axis) and label distances (Y axis) on the ESOL dataset.

Table 2: The Spearman and Pearson correlation coefficients on the ESOL dataset for the GNN and ProtoW-L2 model w.r.t. the pairwise difference in embedding vectors and labels.

	Spearman ρ	Pearson r
GNN	$.424 \pm .029$	$.393 \pm .049$
ProtoS-L2	$.561 \pm .087$	$.414 \pm .141$
ProtoW-Dot	$.592 \pm .150$	$.559 \pm .216$
ProtoW-L2	$\mathbf{.815 \pm .026}$	$\mathbf{.828 \pm .020}$



(a) ESOL GNN (b) ESOL ProtoW-L2 (c) LIPO GNN (d) LIPO ProtoW-L2

Figure 5: 2D heatmaps of T-SNE [25] projections of molecular embeddings (before the last linear layer) w.r.t. their associated predicted labels. Heat colors are interpolations based only on the test molecules from each dataset. Comparing (a) vs (b) and (c) vs (d), we can observe a smoother space of our model compared to the GNN baseline as explained in the main text.

Moreover, as can be seen in Figure 5, our model also provides more robust molecular embeddings compared to the baseline, in the following sense: we observe that a small perturbation of a molecular embedding corresponds to a small change in predicted property value – a desirable phenomenon that holds rarely for the baseline GNN model. Qualitatively, this is shown in Figure 5. Our Wasserstein prototype models yields smoother heatmaps, which is desirable for molecular optimization in the latent space via gradient methods.

6 Related Work

Graph Neural Networks were introduced by Gori et al. [16] and Scarselli et al. [30] as a form of recurrent neural network. Graph convolutional networks (GCN) made their first appearance later on in various forms. Duvenaud et al. [11] and Atwood et al. [2] proposed a propagation rule inspired from convolution and diffusion, although these methods do not scale to graphs with either large degree distribution or node cardinality, respectively. Niepert et al. [27] defined a GCN as a 1D convolution on a chosen node ordering. Kearnes et al. [20] also used graph convolutions with great success to generate high quality molecular fingerprints. Efficient spectral methods were also proposed [5, 10]. Kipf & Welling [21] simplified their propagation rule, motivated as well from spectral graph theory [18], achieving impressive empirical results. Most of these different architectures were later unified into a message passing neural networks framework by Gilmer et al. [15], which applies them to molecular property prediction. A directed variant of message passing was motivated by Dai et al. [9], which was later used to improve state-of-the-art in molecular property prediction on a wide variety of datasets by ChemProp [39]. Another notable application includes recommender systems [40]. Ying et al. [41] proposed DiffPool, which performs a pooling operation for GNN in a hierarchical fashion. Inspired by DeepSets [42], Xu et al. [38] suggest both a simplification and generalization of certain GNN architectures, which should theoretically be powerful enough to discriminate between any different local neighborhoods, provided that hidden dimensions grow as much as the input size. Other recent approaches suggest to modify the sum-aggregation of node embeddings in the GCN architecture with the aim to preserve more information [22, 28]. On the other hand, Hongbin et al. [28] propose to preserve more semantic information by performing a bi-level aggregation which depends on the local geometry of the neighborhood of the given node in the graph. Other recent geometry-inspired GNN include adaptations to embeddings lying in hyperbolic spaces [23, 6] or spaces of constant sectional curvature [3].

7 Conclusion

We propose **OT-GNN**: an enhancement of GNN architectures replacing sum-aggregation of node embeddings via Optimal Transport geometry. We introduce an efficient regularizer which prevents the enhanced model from collapsing back to standard aggregation. Empirically, our models show strong performances in different molecular property prediction tasks. The induced geometry of their latent representations also exhibits stronger correlation with target labels.

Acknowledgments and Disclosure of Funding

We thank Louis Abraham for a counter example of positive-definiteness of the Wasserstein Kernel, Andreas Bloch for help with figures and Wengong Jin & Rachel Wu for detailed comments and feedback.

This work was supported by the DARPA AMD program and by the Machine Learning for Pharmaceutical Discovery and Synthesis (MLPDS) consortium.

References

[1] SN Afriat. Theory of maxima and the method of lagrange. *SIAM Journal on Applied Mathematics*, 20(3):343–357, 1971.

[2] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *Advances in neural information processing systems*, pages 1993–2001, 2016.

[3] Gregor Bachmann, Gary Bécigneul, and Octavian-Eugen Ganea. Constant curvature graph convolutional networks. *arXiv preprint arXiv:1911.05076*, 2019.

[4] Sabri Boughorbel, J-P Tarel, and Nozha Boujemaa. Conditionally positive definite kernels for svm based image recognition. In *2005 IEEE International Conference on Multimedia and Expo*, pages 113–116. IEEE, 2005.

[5] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[6] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 4869–4880, 2019.

[7] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013.

[8] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[9] Hanjun Dai, Bo Dai, and Le Song. Discriminative embeddings of latent variable models for structured data. In *International conference on machine learning*, pages 2702–2711, 2016.

[10] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3844–3852. Curran Associates, Inc., 2016.

[11] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[12] Pavel Dvurechensky, Alexander Gasnikov, and Alexey Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by sinkhorn's algorithm. *arXiv preprint arXiv:1802.04367*, 2018.

[13] Rémi Flamary and Nicolas Courty. Pot python optimal transport library. *GitHub: https://github.com/rflamary/POT*, 2017.

[14] Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. *arXiv preprint arXiv:1706.00292*, 2017.

[15] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[16] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005.

[17] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010.

[18] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

[19] Jørgen Hoffmann-Jørgensen. Measures which agree on balls. *Mathematica Scandinavica*, 37(2):319–326, 1976.

[20] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

[21] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations (ICLR)*, 2017.

[22] Risi Kondor, Hy Truong Son, Horace Pan, Brandon Anderson, and Shubhendu Trivedi. Covariant compositional networks for learning graphs. In *International conference on machine learning*, 2018.

[23] Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. In *Advances in Neural Information Processing Systems*, pages 8228–8239, 2019.

[24] Minh-Thang Luong, Richard Socher, and Christopher D Manning. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113, 2013.

[25] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[26] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, pages 6338–6347, 2017.

[27] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023, 2016.

[28] Hongbin Pei, Bingzhe Wei, Chen-Chuan Chang Kevin, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International conference on machine learning*, 2020.

[29] Ofir Pele and Michael Werman. Fast and robust earth mover's distances. In *2009 IEEE 12th International Conference on Computer Vision*, pages 460–467. IEEE, 2009.

[30] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[31] Morgan A Schmitz, Matthieu Heitz, Nicolas Bonneel, Fred Ngole, David Coeurjolly, Marco Cuturi, Gabriel Peyré, and Jean-Luc Starck. Wasserstein dictionary learning: Optimal transport-based unsupervised nonlinear dictionary learning. *SIAM Journal on Imaging Sciences*, 11(1):643–678, 2018.

[32] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(Sep):2539–2561, 2011.

[33] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

[34] Matteo Togninalli, Elisabetta Ghisu, Felipe Llinares-López, Bastian Rieck, and Karsten Borgwardt. Wasserstein weisfeiler-lehman graph kernels. In *Advances in Neural Information Processing Systems*, pages 6436–6446, 2019.

[35] Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, et al. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, 18(6):463–477, 2019.

[36] S Vichy N Vishwanathan, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 11(Apr):1201–1242, 2010.

[37] Hongteng Xu. Gromov-wasserstein factorization models for graph clustering. *arXiv preprint arXiv:1911.08530*, 2019.

[38] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *International Conference on Learning Representations (ICLR)*, 2019.

[39] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

[40] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 974–983, 2018.

[41] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*, pages 4800–4810, 2018.

[42] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pages 3391–3401, 2017.

A Further Details on Contrastive Regularization

A.1 Motivation

One may speculate that it was locally easier for the model to extract valuable information if it would behave like the Euclidean component, preventing it from exploring other roads of the optimization landscape. To better understand this situation, consider the scenario in which a subset of points in a free point cloud "collapses", i.e. become close to each other (see Figure 3), thus sharing similar distances to all the node embeddings of real input graphs. The submatrix of the optimal transport matrix corresponding to these collapsed points can be equally replaced by any other submatrix with the same marginals (*i.e.* same two vectors obtained by summing rows or columns), meaning that the optimal transport matrix is not discriminative. In general, we want to avoid any two rows or columns in the Wasserstein cost matrix being proportional. An additional problem of point collapsing is that it is a non-escaping situation when using gradient-based learning methods. The reason is that gradients of these collapsed points would become and remain identical, thus nothing will encourage them to "separate" in the future.

A.2 On the Choice of Contrastive Samples

Our experiments were conducted with ten negative samples for each correct transport plan. Five of them were obtained by initializing a matrix with uniform *i.i.d* entries from $[0, 10)$ and performing around five Sinkhorn iterations [7] in order to make the matrix satisfy the marginal constraints. The other five were obtained by randomly permuting the columns of the correct transport plan. The latter choice has the desirable effect of penalizing the points of a free point cloud $\mathbf{Q}_i$ to collapse onto the same point. Indeed, the rows of $\mathbf{T}^i \in \mathcal{C}_{\mathbf{HQ}_i}$ index points in $\mathbf{H}$, while its columns index points in $\mathbf{Q}_i$.

B Theoretical Results

B.1 Proof of Theorem 1

1. Let us first justify why $\mathfrak{agg}$ is not universal. Consider a function $f \in \mathcal{C}(\mathcal{X})$ such that there exists $\mathbf{X}, \mathbf{Y} \in \mathcal{X}$ satisfying both $f(\mathbf{X}) \neq f(\mathbf{Y})$ and $\sum_k \mathbf{x}_k = \sum_l \mathbf{y}_l$. Clearly, any function of the form $\sum_i \alpha_i \sigma(\mathfrak{agg}(\mathbf{W}_i, \cdot) + \theta_i)$ would take equal values on $\mathbf{X}$ and $\mathbf{Y}$ and hence would not approximate f arbitrarily well.

2. To justify that $\mathcal{W}$ is universal, we take inspiration from the proof of universality of neural networks [8].

Notation. Denote by $M(\mathcal{X})$ the space of finite, signed regular Borel measures on $\mathcal{X}$.

Definition. We say that σ is discriminatory w.r.t a kernel k if for a measure $\mu \in M(\mathcal{X})$,

$$\int_{\mathcal{X}} \sigma(k(\mathbf{Y}, \mathbf{X}) + \theta) d\mu(\mathbf{X}) = 0$$

for all $\mathbf{Y} \in \mathcal{X}_d^n$ and $\theta \in \mathbb{R}$ implies that $\mu \equiv 0$.

We start by reminding a lemma coming from the original paper on the universality of neural networks by Cybenko [8].

Lemma. If σ is discriminatory w.r.t. k then k is universal.

Proof: Let S be the subset of functions of the form $\sum_{i=1}^m \alpha_i \sigma(k(\cdot, \mathbf{Q}_i) + \theta_i)$ for any $\theta_i \in \mathbb{R}$, $\mathbf{Q}_i \in \mathcal{X}_d^n$ and $m \in \mathbb{N}^*$ and denote by $\bar{S}$ the closure[5] of S in $\mathcal{C}(\mathcal{X})$. Assume by contradiction that $\bar{S} \neq \mathcal{C}(\mathcal{X})$. By the Hahn-Banach theorem, there exists a bounded linear functional L on $\mathcal{C}(\mathcal{X})$ such

[5] W.r.t the topology defined by the sup norm $\|f\|_{\infty,\mathcal{X}} := \sup_{X \in \mathcal{X}} |f(X)|$.

that for all $h \in \bar{S}$, $L(h) = 0$ and such that there exists $h' \in \mathcal{C}(\mathcal{X})$ s.t. $L(h') \neq 0$. By the Riesz representation theorem, this bounded linear functional is of the form:

$$L(h) = \int_{\mathbf{X} \in \mathcal{X}} h(\mathbf{X}) d\mu(\mathbf{X}),$$

for all $h \in \mathcal{C}(\mathcal{X})$, for some $\mu \in M(\mathcal{X})$. Since $\sigma(k(\mathbf{Q}, \cdot) + \theta)$ is in $\bar{S}$, we have

$$\int_{\mathcal{X}} \sigma(k(\mathbf{Q}, \mathbf{X}) + \theta) d\mu(\mathbf{X}) = 0$$

for all $\mathbf{Q} \in \mathcal{X}_d^n$ and $\theta \in \mathbb{R}$. Since σ is discriminatory w.r.t. k, this implies that $\mu = 0$ and hence $L \equiv 0$, which is a contradiction with $L(h') \neq 0$. Hence $\bar{S} = \mathcal{C}(\mathcal{X})$, *i.e.* S is dense in $\mathcal{C}(\mathcal{X})$ and k is universal.

□

Now let us look at the part of the proof that is new.

Lemma. σ is discriminatory w.r.t. $\mathcal{W}_{\mathrm{L2}}$.

Proof: Note that for any $\mathbf{X}, \mathbf{Y}, \theta, \varphi$, when $\lambda \to +\infty$ we have that $\sigma(\lambda(\mathcal{W}_{\mathrm{L2}}(\mathbf{X}, \mathbf{Y}) + \theta) + \varphi)$ goes to 1 if $\mathcal{W}_{\mathrm{L2}}(\mathbf{X}, \mathbf{Y}) + \theta > 0$, to 0 if $\mathcal{W}_{\mathrm{L2}}(\mathbf{X}, \mathbf{Y}) + \theta < 0$ and to $\sigma(\varphi)$ if $\mathcal{W}_{\mathrm{L2}}(\mathbf{X}, \mathbf{Y}) + \theta = 0$.

Denote by $\Pi_{\mathbf{Y},\theta} := \{\mathbf{X} \in \mathcal{X} \mid \mathcal{W}_{\mathrm{L2}}(\mathbf{X}, \mathbf{Y}) - \theta = 0\}$ and $B_{\mathbf{Y},\theta} := \{\mathbf{X} \in \mathcal{X} \mid \sqrt{\mathcal{W}_{\mathrm{L2}}(\mathbf{X}, \mathbf{Y})} < \theta\}$ for $\theta \geq 0$ and $\emptyset$ for $\theta < 0$. By the Lebesgue Bounded Convergence Theorem we have:

$$\begin{aligned} 0 &= \int_{\mathbf{X} \in \mathcal{X}} \lim_{\lambda \to +\infty} \sigma(\lambda(\mathcal{W}_{\mathrm{L2}}(\mathbf{X}, \mathbf{Y}) - \theta) + \varphi) d\mu(\mathbf{X}) \\ &= \sigma(\varphi)\mu(\Pi_{\mathbf{Y},\theta}) + \mu(\mathcal{X} \setminus B_{\mathbf{Y},\sqrt{\theta}}). \end{aligned}$$

Since this is true for any φ, it implies that $\mu(\Pi_{\mathbf{Y},\theta}) = \mu(\mathcal{X} \setminus B_{\mathbf{Y},\sqrt{\theta}}) = 0$. From $\mu(\mathcal{X}) = 0$ (because $B_{\mathbf{Y},\sqrt{\theta}} = \emptyset$ for $\theta < 0$), we also have $\mu(B_{\mathbf{Y},\sqrt{\theta}}) = 0$. Hence μ is zero on all balls defined by the metric $\sqrt{\mathcal{W}_{\mathrm{L2}}}$.

From the Hahn decomposition theorem, there exist disjoint Borel sets P, N such that $\mathcal{X} = P \cup N$ and $\mu = \mu^+ - \mu^-$ where $\mu^+(A) := \mu(A \cap P)$, $\mu^-(A) := \mu(A \cap N)$ for any Borel set A with μ^+, μ^- being positive measures. Since μ^+ and μ^- coincide on all balls on a finite dimensional metric space, they coincide everywhere [19] and hence $\mu \equiv 0$.

□

Combining the previous lemmas with $k = \mathcal{W}_{\mathrm{L2}}$ concludes the proof.

□

B.2 Proof of Theorem 2

1. We build a counter example. We consider 4 point clouds of size $n = 2$ and dimension $d = 2$. First, define $\mathbf{u}_i = (\lfloor i/2 \rfloor, i\%2)$ for $i \in \{0, ..., 3\}$. Then take $\mathbf{X}_1 = \{\mathbf{u}_0, \mathbf{u}_1\}$, $\mathbf{X}_2 = \{\mathbf{u}_0, \mathbf{u}_2\}$, $\mathbf{X}_3 = \{\mathbf{u}_0, \mathbf{u}_3\}$ and $\mathbf{X}_4 = \{\mathbf{u}_1, \mathbf{u}_2\}$. On the one hand, if $\mathcal{W}(\mathbf{X}_i, \mathbf{X}_j) = 0$, then all vectors in the two point clouds are orthogonal, which can only happen for $\{i, j\} = \{1, 2\}$. On the other hand, if $\mathcal{W}(\mathbf{X}_i, \mathbf{X}_j) = 1$, then either $i = j = 3$ or $i = j = 4$. This yields the following Gram matrix

$$(\mathcal{W}(\mathbf{X}_i, \mathbf{X}_j))_{0 \leq i,j \leq 3} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix} \tag{9}$$

whose determinant is $-1/16$, which implies that this matrix has a negative eigenvalue.

2. This comes from proposition 1. Choosing $k = \mathcal{W}_{\mathrm{L2}}$ and $x_0 = \mathbf{0}$ to be the trivial point cloud made of n times the zero vector yields $\tilde{k} = \mathcal{W}_{\mathrm{dot}}$. Since $\tilde{k}$ is not positive definite from the previous point of the theorem, k is not conditionally negative definite from proposition 1.

□

B.3 Shape of the optimal transport plan for point clouds of same size

The below result describes the shape of optimal transport plans for point clouds of same size. For the sake of curiosity, we also illustrate in Figure 2 the optimal transport for point clouds of different sizes. We note that non-square transports seem to remain relatively sparse as well. This is in line with our empirical observations.

Proposition 2. *For $\mathbf{X}, \mathbf{Y} \in \mathcal{X}_{n,d}$ there exists a rescaled permutation matrix $\frac{1}{n}(\delta_{i\sigma(j)})_{1\leq i,j\leq n}$ which is an optimal transport plan, i.e.*

$$\mathcal{W}_{\mathrm{L2}}(\mathbf{X}, \mathbf{Y}) = \frac{1}{n}\sum_{j=1}^{n} \|\mathbf{x}_{\sigma(j)} - \mathbf{y}_j\|_2^2, \quad \mathcal{W}_{\mathrm{dot}}(\mathbf{X}, \mathbf{Y}) = \frac{1}{n}\sum_{j=1}^{n} \langle \mathbf{x}_{\sigma(j)}, \mathbf{y}_j \rangle. \tag{10}$$

Proof. It is well known from Birkhoff's theorem that every squared doubly-stochastic matrix is a convex combination of permutation matrices. Since the Wasserstein cost for a given transport $\mathbf{T}$ is a linear function, it is also a convex/concave function, and hence it is maximized/minimized over the convex compact set of couplings at one of its extremal points, namely one of the rescaled permutations, yielding the desired result. □

C Complexity

C.1 Wasserstein

Computing the Wasserstein optimal transport plan between two point clouds consists in the minimization of a linear function under linear constraints. It can either be performed exactly by using network simplex methods or interior point methods as done by [29] in time $\tilde{\mathcal{O}}(n^3)$, or approximately up to ε via the Sinkhorn algorithm [7] in time $\tilde{\mathcal{O}}(n^2/\varepsilon^3)$. More recently, [12] proposed an algorithm solving OT up to ε with time complexity $\tilde{\mathcal{O}}(\min\{n^{9/4}/\varepsilon, n^2/\varepsilon^2\})$ via a primal-dual method inspired from accelerated gradient descent.

In our experiments, we used the Python Optimal Transport (POT) library [13]. We noticed empirically that the EMD solver yielded faster and more accurate solutions than Sinkhorn for our datasets, because the graphs and point clouds were small enough (< 30 elements). However, Sinkhorn may take the lead for larger graphs.

C.2 General remarks

Significant speed up could potentially be obtained by rewritting the POT library for it to solve OT in batches over GPUs. In our experiments, we ran all jobs on CPUs. A slow-down in speed by a factor 4 was observed from a purely Euclidean to purely Wasserstein models.

D Further Experimental Details

D.1 Setup of Experiments

Each dataset is split randomly 5 times into 80%:10%:10% train, validation and test sets. For each of the 5 splits, we run each model 5 times to reduce the variance in particular data splits (resulting in each model being run 25 times). We search hyperparameters for each split of the data, and then take the average performance over all the splits. The hyperparameters are separately searched for each data split, so that the model performance is based on a completely unseen test set, and that there is no data leakage across data splits. The models are trained for 150 epochs with early stopping if the validation error has not improved in 50 epochs and a batch size of 16. We train the models using the Adam optimizer with a learning rate of 5e-4. For the prototype models, we use different learning rates for the GNN and the point clouds (5e-4 and 5e-3 respectively), because empirically we find that the gradients are much smaller for the point clouds. The molecular datasets used for experiments here are small in size (varying from 1-4k data points), so this is a fair method of comparison, and is indeed what is done in other works on molecular property prediction [39].

Table 3: The parameters for our models (the prototype models all use the same GNN base model), and the values that we used for hyperparameter search. When there is only a single value in the search list, it means we did not search over this value, and used the specified value for all models.

Parameter Name	Search Values	Description
n_epochs	$\{150\}$	Number of epochs trained
batch_size	$\{16\}$	Size of each batch
lr	{5e-4}	Overall learning rate for model
lr_pc	{5e-3}	Learning rate for the free parameter point clouds
n_layers	$\{5\}$	Number of layers in the GNN
n_hidden	$\{50, 200\}$	Size of hidden dimension in GNN
n_ffn_hidden	{1e2, 1e3, 1e4}	Size of the output feed forward layer
dropout_gnn	$\{0.\}$	Dropout probability for GNN
dropout_fnn	$\{0., 0.1, 0.2\}$	Dropout probability for feed forward layer
n_pc	$\{10, 20\}$	Number of free parameter point clouds in prototype models
pc_size	$\{10\}$	Number of points in free parameter point clouds
pc_hidden	$\{5, 10\}$	Size of hidden dimension in point clouds
nc_coef	$\{0., 0.01, 0.1, 1\}$	Coefficient for noise contrastive regularization