OPTIMAL TRANSPORT GRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

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, a model that computes graph embeddings using parametric prototypes that highlight key facets of different graph aspects. Towards this goal, we are (to our knowledge) the first to successfully combine optimal transport with parametric graph models. Graph representations are obtained from Wasserstein distances between the set of GNN node embeddings and "prototype" point clouds as free parameters. We theoretically prove that, unlike traditional sum aggregation, our function class on point clouds satisfies a fundamental universal approximation theorem. Empirically, we address an inherent collapse optimization issue by proposing a noise contrastive regularizer to steer the model towards truly exploiting the optimal transport geometry. Finally, we consistently report better generalization performance on several molecular property prediction tasks, while exhibiting smoother graph representations.

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 (Yang et al., 2019; Vamathevan et al., 2019). Historically, graphs were decomposed into features such as molecular fingerprints, or turned into non-parametric graph kernels (Vishwanathan et al., 2010; Shervashidze et al., 2011). More recently, learned representations via graph neural networks (GNNs) have achieved state-of-the-art on graph prediction tasks (Duvenaud et al., 2015; Defferrard et al., 2016; Kipf & Welling, 2017; Yang et al., 2019).

Despite these 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.

Departing from this simple aggregation step, Togninalli et al. (2019) recently proposed a kernel function over graphs by directly comparing non-parametric node embeddings as point clouds through optimal transport (Wasserstein distance). Their *non-parametric* model yields better empirical performance over popular graph kernels, but this idea hasn't been extended to the more challenging parametric case where optimization difficulties have to be reconciled with the combinatorial aspects of optimal transport solvers.

Motivated by these observations and drawing inspiration from prototypical networks (Snell et al., 2017), we introduce a new class of GNNs 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 basis functions; they are 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 any existing GNN architecture. We then compare this resulting embedding point cloud to the prototype embedding sets. Formally, the distance between two point clouds is measured by their optimal transport Wasserstein distances. The prototypes as abstract basis functions can be understood as keys that highlight property values associated with different graph structural features. In contrast to previous kernel methods, the prototypes are learned together with the GNN parameters in an end-to-end manner.

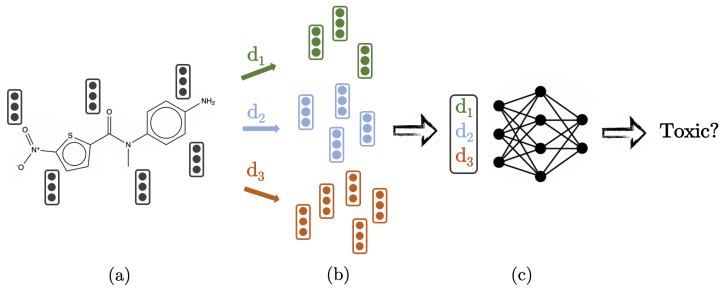


Figure 1: Our OT-GNN prototype model computes graph embeddings from Wasserstein distances between (a) the set of GNN node embeddings and (b) prototype embedding sets. These distances are then used as the molecular representation (c) for supervised tasks, e.g. property prediction. We assume that a few prototypes, e.g. some functional groups, highlight key facets or structural features of graphs relevant to a particular downstream task at hand. We express graphs by relating them to these abstract prototypes represented as free point cloud parameters.

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 trained using combinatorial optimal transport solvers 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 to 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 its merits empirically.

Our contributions. First, we introduce an efficiently trainable class of graph neural networks enhanced with optimal transport (OT) primitives for computing graph representations based on relations with abstract prototypes. Second, we are the first to successfully train parametric graph models together with combinatorial OT distances, despite optimization difficulties. A key element is our noise contrastive regularizer that prevents the model from collapsing back to standard summation, thus fully exploiting the OT geometry. Third, we provide a theoretical justification of the increased representational power compared to the standard GNN aggregation method. 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 (Dai et al., 2016) architecture which was successfully adapted for state-of-the-art molecular property prediction by Yang et al. Yang et al. (2019).

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 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) Peyré et al. (2019) 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.

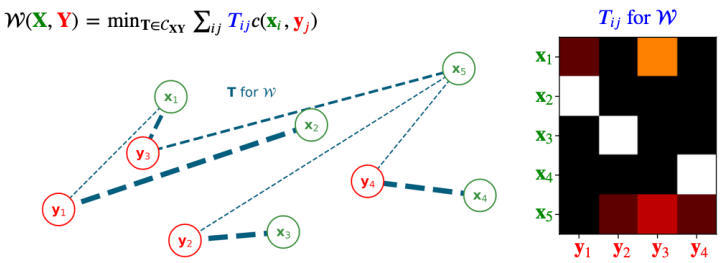


Figure 2: We illustrate, for a given 2D point cloud, the optimal transport plan obtained from minimizing the Wasserstein costs; $c(\cdot, \cdot)$ denotes the Euclidean distance. A higher dotted-line thickness illustrates a greater mass transport.

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 $\mathcal{C}_{\mathbf{XY}}$.

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 $(\mathbb{R}\mathbf{h})_i = \langle \mathbf{r}_i, \mathbf{h} \rangle$. 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.

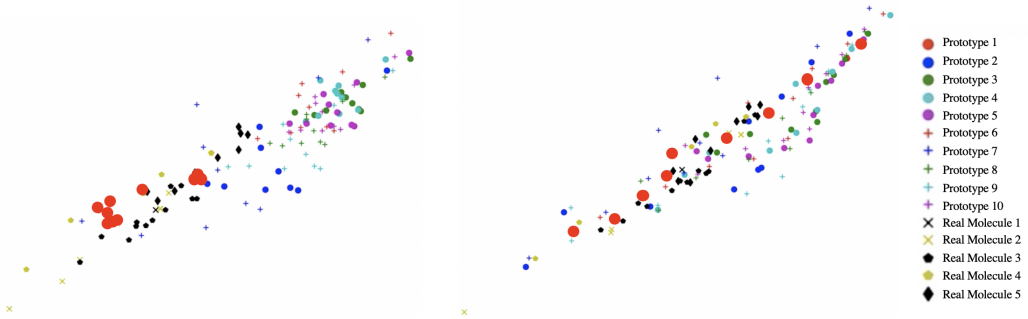
Greater representational power. We formulate mathematically in Section 4 that this kernel has a strictly greater representational power than the kernel corresponding to standard inner-product on top of a sum aggregation, to distinguish between different point clouds. However, there are additional practical concerns that will be 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 $c = -\langle \cdot, \cdot \rangle$:

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

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



(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 same random seed. Points in the prototypes tend to cluster and collapse more when no regularization is used, suggesting that the optimal transport plan no longer remains 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.

A first obstacle and its cause. 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.

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 and Figure 3..

Contrastive regularization. To address this difficulty, we consider adding a regularizer which encourages 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 $N(\mathbf{T}^i) \subset \mathcal{C}_{\mathbf{YQ}_i}$ 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 (2017). Its effect of is shown in Figure 3.

Remarks. The selection of negative examples should reflect the 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 given in Appendix A.2. Note that replacing the set $N(\mathbf{T}^i)$ with a singleton $\{\mathbf{T}\}$ for a contrastive random variable $\mathbf{T}$ lets us rewrite Eq. (6) as[1] $\sum_i \log \sigma(\mathcal{W}_{\mathbf{T}} - \mathcal{W}_{\mathbf{T}^i})$, reminiscent of noise contrastive estimation (Gutmann & Hyvärinen, 2010).

3.3 Optimization & Complexity

Backpropagating gradients through optimal transport (OT) has been the subject of recent research investigations: Genevay et al. (2017) explain how to unroll and differentiate through the Sinkhorn procedure solving OT, which was extended by Schmitz et al. (2018) to Wasserstein barycenters. However, more recently, Xu (2019) proposed to simply invoke the envelop theorem (Afriat, 1971) 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[2] $\sum_{j=1}^m \alpha_j \sigma(k(\cdot, \theta_j) + \beta_j)$ 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 discuss 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.

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

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

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 Boughorbel et al. (2005):

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 stated in different variants before (e.g. Vert (2008)) and as reminded by the below theorem. We here give a novel proof in Appendix B.2.

Theorem 2. *We have that:*

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

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

5 Experiments

5.1 Experimental Setup

We experiment on 4 benchmark molecular property prediction datasets (Yang et al., 2019) including both regression (ESOL, Lipophilicity) and classification (BACE, BBBP) tasks. These datasets cover 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 provide results for our implementation of 4 different widely used graph-based models.

$\mathbf{Fingerprint + MLP}$ is a simple graph baseline that applies a MLP over the input features which are hashed graph structures (called a molecular fingerprint). **GIN** is the Graph Isomorphism Network from (Xu et al., 2019), which is a variant of a GNN. The original GIN does not account for edge features, so we adapt their algorithm to our setting. Next, **GAT** is the Graph Attention Network from (Veličković et al., 2017), which uses multi-head attention layers to propagate information. The original GAT model does not account for edge features, so we adapt their algorithm to our setting. More details about our implementation of the GIN and GAT models can be found in the appendix D.2. Finally, $\mathbf{Chemprop - MPNN}$ (Yang et al., 2019) is a graph network that exhibits state-of-the-art performance for molecular representation learning across multiple classification and regression datasets. Empirically we find that this baseline is indeed the best performing, and so is used as to obtain node embeddings in all our prototype models. Its architecture is described in section 2.1.

Different variants of our OT-GNN prototype model are described below:

$\mathbf{ProtoW\text{-}L2/Dot}$ 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. $\mathbf{ProtoS\text{-}L2}$ is a special case of $\mathbf{ProtoW\text{-}L2}$, in which the point clouds have a *single* point and instead of using Wasserstein distances, we 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 (Flamary & Courty, 2017) to compute Wasserstein distances using the network simplex algorithm (Earth Movers distance), which we find empirically to be faster than the

Sinkhorn algorithm for 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 backpropagate through the cost matrix for computational efficiency. Additionally, to account for the variable size of each input graph, 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 are presented in Appendix D.1.

	Models	ESOL (RMSE) # grphs = 1128	Lipo (RMSE) # grphs= 4199	BACE (AUC) # grphs= 1512	BBBP (AUC) # grphs= 2039
Baselines	Fingerprint + MLP	$.922 \pm .017$	$.885 \pm .017$	$.870 \pm .007$	$.911 \pm .005$
	GIN	$.665 \pm .026$	$.658 \pm .019$	$.861 \pm .013$	$.900 \pm .014$
	GAT	$.654 \pm .028$	$.808 \pm .047$	$.860 \pm .011$	$.888 \pm .015$
	MPNN	$.635 \pm .027$	$.646 \pm .041$	$.865 \pm .013$	$.915 \pm .010$
Ours	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	$\underline{.605 \pm .029}$	$\underline{.604 \pm .014}$	$\mathbf{.873 \pm .015}$	$\mathbf{.920 \pm .010}$

Table 1: Results on the property prediction datasets. **Best** model is in bold, second best is underlined. Lower RMSE is better, while higher AUC is better. Wasserstein models are by default trained with contrastive regularization as described in section 3.2 and outperform those without.

5.2 Experimental Results

5.2.1 Regression and Classification

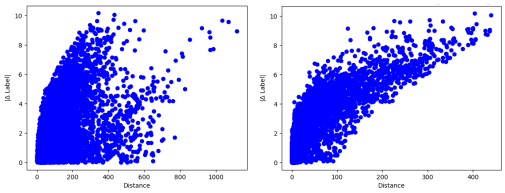
Results are shown in Table 1. Our prototype models outperform state-of-the art GNN/MPNN baselines on all 4 property prediction tasks. Moreover, the prototype models using Wasserstein distance (**ProtoW-L2/Dot**) achieve better performance on 3 out of 4 of the datasets compared to the prototype model using only Euclidean distances (**ProtoS-L2**). This indicates 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. Quantitative results in Table 1 also highlight the benefit of this regularization.

5.2.3 Learned Embedding Space: Qualitative and Quantitative Results

We further examine the learned embedding space of the best baseline (i.e. MPNN) and our best Wasserstein model. We claim that our models learn smoother latent representations. We compute

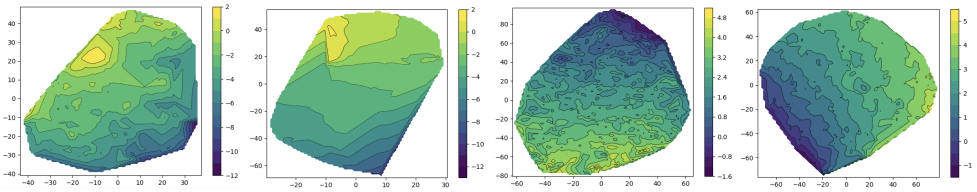


(a) MPNN model (b) ProtoW-L2 model

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
MPNN	$.424 \pm .029$	$.393 \pm .049$
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 MPNN (b) ESOL ProtoW-L2 (c) LIPO MPNN (d) LIPO ProtoW-L2

Figure 5: 2D heatmaps of T-SNE Maaten & Hinton (2008) 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 MPNN baseline (see also main text).

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 is reminiscent of evaluation tasks for the correlation of word embedding similarity with human labels (Luong et al., 2013).

Our $\mathrm{ProtoW\text{-}L2}$ achieves better ρ and r scores compared to the MPNN 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 MPNN model (left) and the $\mathrm{ProtoW\text{-}L2}$ model (right). Our $\mathrm{ProtoW\text{-}L2}$ model, trained to optimize distances in the embedding space, produces more meaningful representations with respect to the label of interest.

Moreover, as qualitatively shown in Figure 5, our model 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 MPNN model. Our Proto-W-L2 model yields smoother heatmaps.

6 Related Work

Graph Neural Networks were introduced by Gori et al. (2005) and Scarselli et al. (2008) as a form of recurrent neural networks. Graph convolutional networks (GCN) made their first appearance later on in various forms. Duvenaud et al. (2015); Atwood & Towsley (2016) 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. Niepert et al. (2016) defined a GCN as a 1D convolution on a chosen node ordering. Kearnes et al. (2016) also used graph convolutions to generate high quality molecular fingerprints. Efficient spectral methods were proposed by Bruna et al. (2013); Defferrard et al. (2016). Kipf & Welling (2017) simplified their propagation rule, motivated from spectral graph theory (Hammond et al., 2011). Different such architectures were later unified into a message passing neural networks framework by Gilmer et al. (2017), and applied to molecular property prediction. A directed variant of message passing from Dai et al. (2016) was later used to improve state-of-the-art in molecular property prediction on a wide variety of datasets by (Yang et al., 2019). Other applications include recommender systems (Ying et al., 2018). Inspired by DeepSets Zaheer et al. (2017), Xu et al. (2019) suggest a simplified architecture called GIN, which theoretically can discriminate between any different local neighborhoods. Other recent approaches modify the sum-aggregation of node embeddings in the GCN architecture with the aim to preserve more information Kondor et al. (2018); Pei et al. (2020). Other recent geometry-inspired GNNs include adaptations to non-Euclidean spaces (Liu et al., 2019; Chami et al., 2019; Bachmann et al., 2019).

7 Conclusion

We propose **OT-GNN**: one of the first parametric graph models that leverages optimal transport to learn graph representations. It learns abstract prototypes as free parametric point clouds that highlight different aspects of the graph. Empirically, we outperform popular baselines in different molecular property prediction tasks, while the learned representations also exhibit stronger correlation with the target labels. Finally, universal approximation theoretical results enhance the merits of our model.

REFERENCES

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

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

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

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*, pp. 113–116. IEEE, 2005.

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

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

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

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

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

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 (eds.), *Advances in Neural Information Processing Systems 29*, pp. 3844–3852. Curran Associates, Inc., 2016.

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*, pp. 2224–2232, 2015.

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.

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

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

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*, pp. 1263–1272. JMLR. org, 2017.

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, pp. 729–734. IEEE, 2005.

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*, pp. 297–304, 2010.

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.

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

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.

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

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.

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

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*, pp. 104–113, 2013.

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

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

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

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.

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

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

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.

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.

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.

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

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

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.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Jean-Philippe Vert. The optimal assignment kernel is not positive definite. *arXiv preprint arXiv:0801.4061*, 2008.

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.

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

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

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.

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*, pp. 974–983, 2018.

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