Optimization of the Diffusion Time in Graph Diffused-Wasserstein Distances: Application to Domain Adaptation

Amélie Barbe, Paulo Gonçalves, Marc Sebban, Pierre Borgnat, Rémi Gribonval, Titouan Vayer

Optimization of the Diffusion Time in Graph Diffused-Wasserstein Distances: Application to Domain Adaptation

Amélie Barbe
Univ Lyon, ENS de Lyon, Inria CNRS, UCBL
LIP UMR 5668, Lyon, France
amelie.barbe@ens-lyon.fr

Paulo Gonçalves
Univ Lyon, Inria, ENS de Lyon CNRS, UCBL
LIP UMR 5668, Lyon, France
paulo.goncalves@ens-lyon.fr

Marc Sebban
Univ Lyon, UJM-Saint-Etienne, CNRS Institut d'Optique Graduate School Laboratoire Hubert Curien
Saint-Etienne, France
marc.sebban@univ-st-etienne.fr

Pierre Borgnat
Université de Lyon, CNRS ENSL, Laboratoire de Physique
Lyon, France
pierre.borgnat@ens-lyon.fr

Rémi Gribonval
Univ Lyon, Inria, ENS de Lyon CNRS, UCBL
LIP UMR 5668, Lyon, France
remi.gribonval@ens-lyon.fr

Titouan Vayer
Univ Lyon, ENS de Lyon, Inria CNRS, UCBL
LIP UMR 5668, Lyon, France
titouan.vayer@ens-lyon.fr

***Abstract*—The use of the heat kernel on graphs has recently given rise to a family of so-called *Diffusion-Wasserstein distances* which resort to the Optimal Transport theory for comparing attributed graphs. In this paper, we address the open problem of optimizing the diffusion time used in these distances and which plays a key role in several machine learning settings, including graph domain adaptation or graph classification. Inspired from the notion of triplet-based constraints used, e.g., in metric learning, we design a loss function that aims at bringing two graphs closer together while keeping an impostor away, this latter taking the form of a Wasserstein barycenter. After a thorough analysis of the properties of this function, we show on synthetic and real-world data that the resulting Diffusion-Wasserstein distances outperforms the Gromov and Fused-Gromov Wasserstein distances on unsupervised graph domain adaptation tasks. Additionally, we give evidence in such a setting that our method for optimizing the diffusion parameter allows to overcome the limitation of the widely used *circular validation* strategy.**

***Index Terms*—Optimal transport, graphs, domain adaptation, heat kernel.**

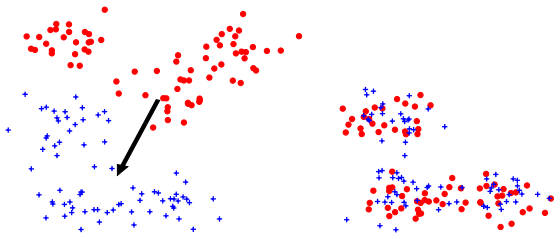
I. Introduction

A *Domain Adaptation* (DA) scenario arises in machine learning when we observe a change of distribution (a.k.a. *domain shift*) between the training data (the *source* distribution) and the samples used at test time with the deployed model (the *target* distribution). To cite a few examples, DA can occur in *image processing*, when changing the lighting or camera lens while acquiring images, in *demography* with social mobility of people or in *fraud detection*, with fraudsters trying to adapt over time to better mimic genuine behaviors. Most of the time, training a new model from the target distribution is not desirable for several reasons: (i) the algorithmic complexity required for optimizing from scratch the parameters of a new model; (ii) the lack of target training examples; (iii) the lack (or absence) of supervision (i.e. no labelled target data available), etc. In such a setting, the domain adaptation theory [1], [2] suggests to reduce the divergence between the source and the target distributions while learning an efficient model from the labelled source data.

One way to solve DA problems is to use *Optimal Transport* [3], [4] (OT). As illustrated in Figure 1, OT provides a natural geometry for comparing and aligning two distributions in the space of probability measures. In the discrete case, when dealing with point clouds, it looks for a coupling matrix (and its corresponding Wasserstein distance) that minimizes the global cost of transporting the individual masses from the source to the target distribution. In [5], the authors introduced OTDA, the first DA algorithm based on OT which moves the source on the top of the target by preventing - using a group-sparse regularization - two source data of different labels from being transported on the same target example. Once the alignment is achieved, a model is learned from the labelled source data and applied on the target distribution.

During the past few years, OT has received a tremendous interest from the machine learning community for comparing structured data, like attributed graphs. To address this task, two OT-based distances have been introduced in the literature. The Gromov-Wasserstein distance [6] (`GW`) has been originally defined for comparing two distributions that do not necessarily lie in the same feature space. Based on intra-distribution pairwise distances/costs, `GW` provides a nice framework for computing a distance between two graphs by encoding some structure, like the shortest path between two vertices. In order

Work supported by the ACADEMICS grant of the IDEXLYON, project of the Université de Lyon, PIA operated by ANR-16-IDEX-0005



(a) Before alignment. (b) After alignment.

Fig. 1: Illustration of OT on a toy example: (a) two source (red circles) and target (blue crosses) distributions are given. (b) reduction of the domain shift after minimization of the global cost of transportation from the source to the target.

to consider both the features associated to the nodes and the structure of the graphs, the *Fused-Gromov-Wasserstein* (FGW) distance has been recently introduced in [7]. FGW acts as a combination of the Wasserstein distance (focusing only on the features) and the GW distance (considering only the structure). Although FGW has been shown to be significantly better than GW in graph classification tasks, it has the shortcoming of not scaling to large graphs due to a complexity in $\mathcal{O}(m^2n^2)$ (with m (resp. n) the size of the source (resp. target) graph).

To overcome this limitation, a new family of graph Diffusion Wasserstein (DW) distances have been defined in [8] exploiting the heat diffusion on attributed graphs. The main advantage of DW is to capture both the feature and structural information in one term based on the Laplacian exponential kernel applied on the features associated to the nodes. This heat kernel is defined as $\exp(-\tau L)$, where L is the graph Laplacian and τ is the time during which the features diffuse along the structure of the graph. Even though DW has been shown to provide promising results when addressing graph DA tasks [8], the quality of this family of Wasserstein distances highly depends on the value of the diffusion time τ. Because of the absence of target labels in unsupervised DA, tuning this parameter remains a tricky and unstable task even with the widely used circular validation strategy [9].

The main contribution of this paper consists in addressing this challenging task by designing a new optimization problem dedicated to automatically learn the optimal diffusion time. Inspired from the triplet-based constraints used in metric learning [10], we design a loss function that aims at bringing the source and target graphs close together while keeping an impostor away. This latter is built from the feature-based Wasserstein barycenter of the two original graphs. We present in this paper a thorough analysis of the properties of this new loss function. We also discuss some strategies to alleviate the additional complexity cost of minimizing it. The experimental results obtained on both synthetic and real datasets give indisputable evidence that our method for optimizing the diffusion parameter allows to outperform the state-of-the-art graph-based Wasserstein distances and overcome the limitation of the circular validation.

The rest of this paper is organized as follows. Section II is dedicated to the preliminary knowledge on OT and the definition of GW, FGW and DW distances. Section III is devoted to the presentation and analysis of our optimization problem aiming at learning the diffusion time. Section IV describes experiments performed on synthetic and real data. We conclude in Section V by opening future promising lines of research.

II. Preliminaries

In this section, we provide the necessary background in OT and recall the definition of the Wasserstein distance (W), as well as those of the Gromov Wasserstein (GW), Fused-Gromov Wasserstein (FGW) and Diffusion Wasserstein (DW) distances.

A. Discrete Optimal Transport and Wasserstein distance

Let us consider two empirical probability measures μ and ν, called *source* and *target* distributions, and supported on two sample sets $X = \{x_i\}_{i=1}^m$ and $Y = \{y_j\}_{j=1}^n$, respectively, lying in some feature space $\mathcal{X}$ and with weights $a = (a_i)_{i=1}^m$, $b = (b_j)_{j=1}^n$ such that $\mu = \sum_{i=1}^m a_i \delta_{x_i}$ and $\nu = \sum_{j=1}^n b_j \delta_{y_j}$ where δ is the Dirac measure. If $\mathcal{X} = \mathbb{R}^r$ for some integer $r \geq 1$, a matrix representation of X (resp. of Y) is the matrix $\mathbf{X} \in \mathbb{R}^{m \times r}$ (resp. $\mathbf{Y} \in \mathbb{R}^{n \times r}$) whose rows are $x_i^\top, 1 \leq i \leq m$ (resp. $y_j^\top, 1 \leq j \leq n$). Let $M = M(X,Y) \in \mathbb{R}_+^{m \times n}$ be a cost matrix, where $M_{ij} \stackrel{def}{=} [d(x_i, y_j)]_{ij}$ is the cost (w.r.t. to some distance d) of transporting x_i on y_j. OT [3], [11] aims at moving μ on the top of ν in an optimal way with respect to M. Let $\Pi(a,b)$ be a transportation polytope defined as the set of admissible coupling matrices γ:

$$\Pi(a,b) = \{\gamma \in \mathbb{R}_+^{m \times n} \text{ s.t. } \gamma 1_n = a, \gamma^T 1_m = b\},$$

where γ_{ij} is the mass transported from x_i to y_j and 1_k is the vector of dimension k with all entries equal to one. The p-Wasserstein distance $\mathtt{W}_p^p(\mu,\nu)$ between the source and target distributions is defined as follows:

$$\mathtt{W}_p^p(\mu,\nu) = \min_{\gamma \in \Pi(a,b)} \langle \gamma, M^p(X,Y) \rangle_F, \tag{1}$$

where $\langle ., . \rangle_F$ is the Frobenius inner product and $M^p(X,Y) := (M_{ij})^p$ is the entrywise p-th power of $M(X,Y)$ with exponent $p > 0$.

Equation (1) is a constrained linear optimisation problem which always has a solution (at least, the naive one of the form $\gamma = ab^T$ which does not take into account the costs of the displacements). Its complexity is polynomial in the sample set sizes (super cubic in practice [12] when using a minimum cost flow solver), and thus becomes quickly prohibitive as the number of samples grows. Moreover, the solution of Equation (1) might be unstable because inherently the linear program looks for vertices in the polytope, and therefore the solution can very easily change from one vertex to another by slightly changing the cost function M. To address these limitations, an entropy-regularized version has been introduced in [3], [13] and is defined as follows:

$$\mathtt{W}_{p,\varepsilon}(\mu,\nu) = \min_{\gamma \in \Pi(a,b)} \langle \gamma, M^p(X,Y) \rangle_F - \varepsilon E(\gamma), \tag{2}$$

where $E(\gamma) = -\sum_{i,j} \gamma_{ij}(\log \gamma_{ij} - 1)$ is the Shannon entropy.

B. Graph-based Wasserstein distances

Let us now consider empirical probability measures supported on graphs, for which we need to extend the notations of the previous section. Following [7], we define an *undirected attributed graph* as the quadruplet $\mathcal{G} = (\mathcal{V}, \mathcal{E}, l_f, l_s)$. $\mathcal{V}$ and $\mathcal{E}$ are the vertices and edges of the graph respectively. $l_f : \mathcal{V} \mapsto \mathcal{X}$ is a function that associates a feature x_i to each vertex v_i. $l_s : \mathcal{V} \mapsto \Omega_s$ is a function that associates a structural representation z_i to each vertex v_i; z_i is further used to define a cost function $C : \Omega_s^2 \mapsto \mathbb{R}_+$ which measures the structural similarity between two nodes. Additionally, if $\mathcal{G}$ is a *labeled* graph, each vertex is assigned a label from some label space $\mathcal{C}$. Finally, when associated with a probability distribution on the nodes, the undirected graph $\mathcal{G}$ can be seen as a probability measure, and thus can be considered in an OT scheme.

In the following, we first recall the definition of the *Gromov-Wasserstein distance* (GW) [6]. Unlike W, the source and target distributions are not required for GW to lie in the same feature space. Instead of using a cost matrix between the source and target samples, it considers two intra-distribution pairwise cost matrices. Therefore, GW can by directly applied on graphs, by using, e.g., the shortest-path matrix (or any structural matrix) for the cost functions C's.

Definition II.1. *Let $\mathcal{G}^s$ and $\mathcal{G}^t$ be two source and target graphs of size m and n respectively with their associated probability distributions μ and ν over the nodes. Let $C^s \in \mathbb{R}_+^{m\times m}$ and $C^t \in \mathbb{R}_+^{n\times n}$ be two pairwise cost matrices associated with each graph. The Gromov-Wasserstein (GW) distance between $\mathcal{G}^s$ and $\mathcal{G}^t$ is defined as follows:*

$$\mathtt{GW}_p^p(C^s, C^t, \mu, \nu) = \min_{\gamma \in \Pi(a,b)} \left\{ \sum_{i,j,k,l} |C^s_{ik} - C^t_{jl}|^p \gamma_{ij}\gamma_{kl} \right\}. \tag{3}$$

Note that for the sake of simplicity, the dependency on C^s and C^t will be omitted in the rest of the paper. It is worth noting that GW only considers the structural information of the graphs. In order to take also into account the feature information at the node level, the Fused Gromov-Wasserstein distance (FGW) has been recently introduced in [7].

Definition II.2. *Let $\mathcal{G}^s$ and $\mathcal{G}^t$ be two source and target graphs of size m and n respectively with their associated probability distributions μ and ν over the nodes. Let $\alpha \in [0, 1]$. The Fused Gromov-Wasserstein (FGW) is defined as follows:*

$$\mathtt{FGW}_p^p(\mu, \nu) = \min_{\gamma \in \Pi(a,b)} \{ \sum_{i,j,k,l} \big((1-\alpha) M_{ij}^p + \alpha |C^s_{ik} - C^t_{jl}|^p\big) \gamma_{ij}\gamma_{kl} \}. \tag{4}$$

The hyper-parameter α allows to control the trade-off between the importance of the structure and the features, and has to be tuned according to the task at hand. FGW covers two special cases. When $\alpha = 0$, we retrieve the Wasserstein distance, that is $\mathtt{FGW}_p^p(\mu, \nu \mid \alpha = 0) = \mathtt{W}_p^p(\mu, \nu)$. When $\alpha = 1$, we recover the Gromov-Wasserstein distance, that is $\mathtt{FGW}_p^p(\mu, \nu \mid \alpha = 1) = \mathtt{GW}_p^p(\mu, \nu)$. Roughly speaking, the optimal coupling matrix $\gamma^\star$ will tend to associate two source and target nodes if both their features and structural representations are similar.

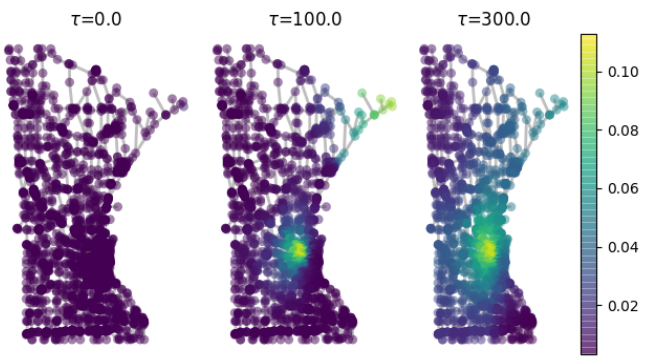


Fig. 2: Illustration of the heat kernel diffusion for three values of τ of a 1D signal ($r = 1$) centred on two nodes in the Minnesota Read Network graph [14].

As implied by the four indices involved in the sum of Equation (4), FGW has the shortcoming of not scaling well to large graphs due to a complexity in $\mathcal{O}(m^2n^2)$. To overcome this drawback, the Diffusion-Wasserstein distance (DW) has been introduced in [8]. It resorts to the so-called *heat kernel* which diffuses the features in the graph before computing the transport. Therefore, DW incorporates both the structural and the feature information. This process depends on a diffusion time τ which allows to capture the graph's structure at different scales (see Figure 2 for an illustration on 1D features).

Definition II.3. *Let $\mathcal{G}^s$ and $\mathcal{G}^t$ be two source and target graphs of size m and n respectively with their associated probability distributions μ and ν over the nodes. Let $L^s \in \mathbb{R}^{m\times m}$ and $L^t \in \mathbb{R}^{n\times n}$ be the Laplacian matrices of $\mathcal{G}^s$ and $\mathcal{G}^t$. Let $\mathbf{X} \in \mathbb{R}^{m\times r}$, $\mathbf{Y} \in \mathbb{R}^{n\times r}$ represent respectively the source and the target sample sets associated to the features ($\in \mathbb{R}^r$) on their vertices.*

Given parameters $0 \leq \tau^s, \tau^t < \infty$, consider the diffused sample sets $\tilde{X}$, $\tilde{Y}$ represented by the matrices $\tilde{\mathbf{X}} = \exp(-\tau^s L^s)\mathbf{X} \in \mathbb{R}^{m\times r}$, $\tilde{\mathbf{Y}} = \exp(-\tau^t L^t)\mathbf{Y} \in \mathbb{R}^{n\times r}$ and define $\tilde{M}(\tau^s, \tau^t) := M(\tilde{X}, \tilde{Y}) \in \mathbb{R}^{m\times n}$, a cost matrix between features that takes into account the structure of the graphs through diffusion operators. We define the Diffusion Wasserstein distance (DW) between μ and ν as:

$$\mathtt{DW}_p^p(\mu, \nu \mid \tau^s, \tau^t) = \min_{\gamma \in \Pi(a,b)} \langle \gamma, \tilde{M}^p \rangle. \tag{5}$$

Here again, $\tilde{M}^p$ is the entrywise p-th power of $\tilde{M}$. The underlying distance is implicit in $M(\cdot, \cdot)$. For the sake of concision, the dependency on τ^s and τ^t will be omitted from the notation $\mathtt{DW}_p^p(\mu, \nu)$ if not specifically required, and simplified as $\mathtt{DW}_p^p(\mu, \nu \mid \tau)$ when the diffusion times are equal. For the same reason, we will omit in the rest of this paper the superscript p used in all the Wasserstein distances.

III. Optimization of the Diffusion Time τ

In this section, we present the main contribution of this paper. Unlike the circular validation [9] that aims at tuning hyper-parameters in unsupervised DA by benefiting from pseudo-target labels, we suggest here to directly optimize the diffusion time τ in a self-supervised way.

A. Circular validation

One peculiarity of unsupervised domain adaptation comes from the absence of target labels. In such a setting, a standard method to tune hyper-parameters is the *circular validation* [9]. The "circular" aspect is due to the fact that the labels go back-and-forth between the source and the target data. Let us detail the underlying principle in the context of an OT-based graph domain adaptation task. Given a transport map γ and a set of labels $l^s \in \mathcal{C}$ for the source graph, one can define pseudo-labels $\hat{l}^t$ for the target graph by choosing, for each node, the label from which the maximum weight comes from:

$$\hat{l}^t_j = \underset{l \in \mathcal{C}}{\operatorname{argmax}} \left\{ \sum_{1 \leq i \leq m} \gamma_{ij} \delta_{l^s_i = l} \right\}. \tag{6}$$

Like-wise, pseudo-labels for the source graph can be re-inferred in a similar fashion:

$$\hat{l}^s_i = \underset{l \in \mathcal{C}}{\operatorname{argmax}} \left\{ \sum_{1 \leq j \leq n} \gamma_{ij} \delta_{\hat{l}^t_j = l} \right\}. \tag{7}$$

It is now possible to define an unsupervised score for ranking the transport maps obtained by different sets of hyper-parameters. This score measures the level of agreement between the original and pseudo source labels:

$$s(\gamma) = \frac{1}{m} \sum_{i=1}^{m} \delta_{l^s_i = \hat{l}^s_i}. \tag{8}$$

It is important to note that while a low score of $s(\gamma)$ is an evidence that the considered hyper-parameter does not lead to a good model, a high score would not allow us to definitely conclude. Indeed, for two graphs of the same size, any permutation matrix γ would produce a perfect score of 1, that can make the circular validation unstable.

B. A triplet-based loss function to learn τ

To address the limitation of the circular validation, we propose in the following to learn the diffusion time by minimizing a loss function that considers an impostor attributed graph. Inspired from the triplet-based constraints used, e.g., in metric learning [10], the impostor facilitates the choice of the diffusion time τ that brings $\mathcal{G}^s$ and $\mathcal{G}^t$ close together without suffering from degenerate phenomena.

Definition III.1. *Let $\mathcal{G}^s$ and $\mathcal{G}^t$ be two source and target graphs of size m and n respectively with their associated probability distributions μ and ν over the nodes. The impostor $\mathcal{G}^0$ with respect to $\mathcal{G}^s$ and $\mathcal{G}^t$ is a graph with $i = \lceil \frac{m+n}{2} \rceil$*

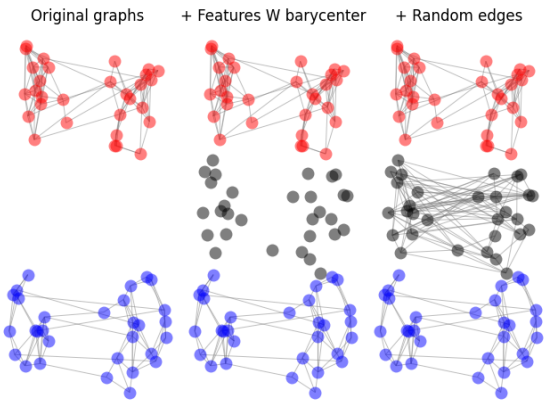


Fig. 3: Illustration of the construction of the impostor (in grey) of two graphs (in red and blue). Features correspond to the 2D coordinates of the nodes. Impostor nodes are supported on the Wasserstein barycenter (eq. 9) of the original graphs' features. Impostor edges are drawn uniformly with probability equal to the average connection probability of the two original graphs.

nodes whose features X^0 are defined as the minimizer of the following Wasserstein barycenter problem:

$$X^0 = \underset{X \in \mathbb{R}^{i \times r}}{\operatorname{argmin}} \left\{ \frac{1}{2} \left(\mathtt{W}(X^s, X) + \mathtt{W}(X^t, X) \right) \right\}. \tag{9}$$

The adjacency matrix A^0 of $\mathcal{G}^0$ is sampled according to an Erdös-Rényi model [15], with connection probability $p^0 = \frac{p^s + p^t}{2}$ the average connection probability of the two graphs.

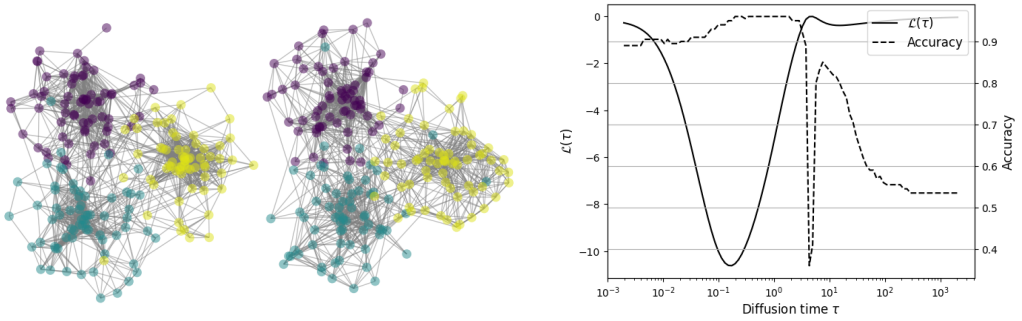
The solution of the Wasserstein barycenter problem is difficult in practice and is the subject of a rich literature (see [16]). Our problem is a free-support barycenter problem where the main obstacle is that we have to optimize on the support X of the barycenter [17], [18]. It has been recently proved that this problem can be solved in polynomial time when the number of points of each measure is fixed [1]. In our case, we chose to rely on the heuristic proposed in [16] which is reasonable as there are only 2 distributions involved and the weights of the barycenter are considered as fixed (uniform in our case). Overall it boils down to iterating over 1) solving two linear OT problems $\mathtt{W}(X^s, X)$ and $\mathtt{W}(X^t, X)$ 2) finding the support X which can be done in closed-form as detailed in [17] (Equation 8). The procedure for generating the impostor is illustrated in Figure 3 for two toy graphs with 2D features.

The diffusion parameter can be now defined as the solution of the following optimization problem:

$$\tau^* = \underset{\tau \geq 0}{\operatorname{argmin}} \left\{ \mathcal{L}(\tau) \right\}, \text{ with} \tag{10}$$

$$\mathcal{L}(\tau) = \mathtt{DW}_p(\mathcal{G}^s, \mathcal{G}^t \mid \tau) - \left(\mathtt{DW}_p(\mathcal{G}^s, \mathcal{G}^0 \mid \tau) + \mathtt{DW}_p(\mathcal{G}^t, \mathcal{G}^0 \mid \tau) \right). \tag{11}$$

Intuitively, like in metric learning, the idea is to learn the parameters of a model (here, a unique diffusion time τ) that (i) constrains $\mathcal{G}^s$ and $\mathcal{G}^t$ to get closer while (ii) preventing a scenario facilitating the bringing together of $\mathcal{G}^s$ and $\mathcal{G}^t$ with a



(a) Source (left) and target (right) community graphs (b) $\mathcal{L}(\tau)$ and DA accuracy as a function of τ.

Fig. 4: Correlation between $\mathcal{L}(\tau)$ and the DA accuracy: (a) source and target community graphs $\mathcal{G}^s$ and $\mathcal{G}^t$; (b) the global minimum of $\mathcal{L}(\tau)$ corresponds to the maximum accuracy reachable in a DA task.

"different" graph. However, unlike supervised metric learning where the impostors of a pair of points of the same class can be defined as the training samples of the opposite label in the close neighborhood, the notion of "different" is here ill-defined because of the absence of target labels in DA tasks. By using $\mathcal{G}^0$ as defined above, we generate an impostor sufficiently close in terms of features (defined as the Wasserstein barycenter) and structure (mean structure of $\mathcal{G}^s$ and $\mathcal{G}^t$ in terms of vertices) forcing the identification of a parameter τ which aligns as well as possible $\mathcal{G}^s$ and $\mathcal{G}^t$ while ensuring a certain "margin" to a different but related graph.

C. Analysis of the triplet-based loss function

In this section, we provide a thorough analysis of our loss function. First, we illustrate the correlation between the minimizer of Eq. (11) and the corresponding accuracy on a DA task. Then, after the derivation of some theoretical properties, we give evidence that $\mathcal{L}(\tau)$ is very robust to some approximations aiming at reducing its algorithmic complexity.

a) Correlation between $\mathcal{L}(\tau)$ and the accuracy in DA: The coupling matrix γ resulting from the calculation of $\mathtt{DW}_p(\mathcal{G}^s, \mathcal{G}^t \mid \tau)$ with τ the solution of Problem (11) can be directly used for addressing a graph DA task. Given γ, each target node receives the mass of some source vertices and then can be assigned the majority class among the transported labels. According to this classification rule, one can compute the DA accuracy measuring the level of agreement between the predicted and the expected labels. In Figure 4, we illustrate the correlation between $\mathcal{L}(\tau)$ and the DA accuracy from two source and target synthetic community graphs, made of 200 nodes and 3 classes. A community graph is built by (i) assigning a random class to each node, (ii) generating 2D features using a Gaussian distribution with a different center for each class, (iii) connecting all points of the same class that are closer than some radius r, and (iv) linking points of different classes at random with a small connection probability. The two resulting source and target graphs are presented in Figure 4a. In Figure 4b, we plot the loss $\mathcal{L}(\tau)$ and the accuracy obtained from a large range of τ. The most interesting point is that the global minimum of the loss (which is not convex in general with potentially several local minima) corresponds to the parameter τ yielding the maximum accuracy. On the other hand, the global maximum of $\mathcal{L}(\tau)$ matches with the worst behavior in DA. Therefore, this correlation between $\mathcal{L}(\tau)$ and the accuracy confirms the interest of our loss for addressing graph-based optimal transport tasks with Diffusion Wasserstein distances.

b) Theoretical properties: Given the definition of the loss function $\mathcal{L}(\tau)$, we can derive the following properties.

Theorem III.2. *Let $\mathcal{G}^s$ and $\mathcal{G}^t$ be two connected attributed graphs. The loss function $\mathcal{L}(\tau)$ of Eq.* (11) *is continuous, negative, and its limits are:*

$$\lim_{\tau \to 0} \mathcal{L}(\tau) = \lim_{\tau \to \infty} \mathcal{L}(\tau) = 0. \tag{12}$$

Proof. Continuity stems from continuity of all the functions involved. Then, writing X^u_τ the distribution of the features of a graph $\mathcal{G}^u$ diffused for a time τ ($u \in \{0, s, t\}$), we have:

$$\begin{aligned}\mathcal{L}(\tau) &= \mathtt{DW}_p(\mathcal{G}^s, \mathcal{G}^t \mid \tau) - \left(\mathtt{DW}_p(\mathcal{G}^s, \mathcal{G}^0 \mid \tau) + \mathtt{DW}_p(\mathcal{G}^t, \mathcal{G}^0 \mid \tau)\right) \\ &= \mathtt{W}(X^s_\tau, X^t_\tau) - (\mathtt{W}(X^s_\tau, X^0_\tau) + \mathtt{W}(X^0_\tau, X^t_\tau)). \end{aligned} \tag{13}$$

W being a distance, negativity follows from triangle inequality. Now, denote X^s and X^t the features of $\mathcal{G}^s$ and $\mathcal{G}^t$, and X^0 the features of the impostor $\mathcal{G}^0$. By definition of X^0 as a minimizer:

$$\begin{aligned}\mathtt{W}(X^s, X^0) + \mathtt{W}(X^t, X^0) &\leq \mathtt{W}(X^s, X^t) + \mathtt{W}(X^t, X^t) \\ &\leq \mathtt{W}(X^s, X^t), \end{aligned} \tag{14}$$

and by triangle inequality we have:

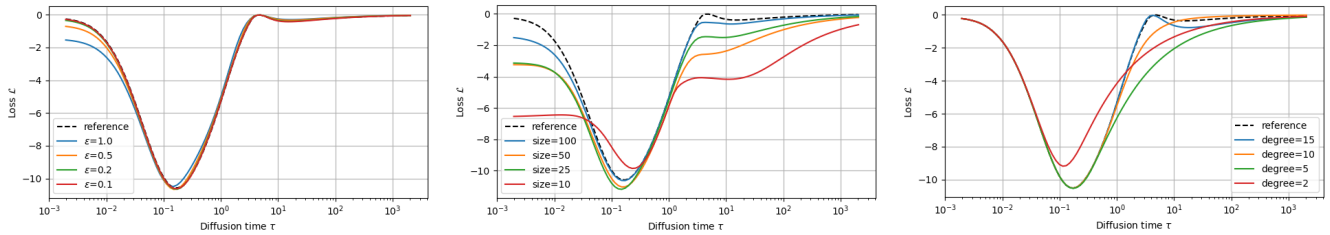
$$\mathtt{W}(X^s, X^0) + \mathtt{W}(X^t, X^0) \geq \mathtt{W}(X^s, X^t). \tag{15}$$

By combining Eq.(14) and (15), we get:

$$\mathcal{L}(0) = \mathtt{W}(X^s, X^t) - (\mathtt{W}(X^s, X^0) + \mathtt{W}(X^t, X^0)) = 0. \tag{16}$$

Finally, from [3], Remark 9.1, we have that the expected value of the barycenter is the barycenter of the expected values:

$$\mathbb{E}[X^0] = \frac{1}{2}\left(\mathbb{E}[X^s] + \mathbb{E}[X^t]\right). \tag{17}$$



(a) Entropy regularization. (b) Impostor downsizing. (c) Chebychev polynomials.

Fig. 5: Behavior of the loss $\mathcal{L}(\tau)$ computed from the two source and target Community Graphs of Fig. 4a in three approximation scenarios: (a) effect of the entropic regularization used in the inner DW distance for various values of the regularization parameter ε; (b) effect of downsizing the number of nodes of the impostor; (c) effect of approximating the diffusion process with different Chebychev polynomials. The dashed line represents the exact $\mathcal{L}(\tau)$.

From [8], Proposition 1, we have the limit of $\mathtt{DW}_2$:

$$\lim_{\tau\to\infty} \mathtt{DW}_2(\mathcal{G}^s, \mathcal{G}^t) = \|\mathbb{E}[X^s] - \mathbb{E}[X^t]\|_2. \tag{18}$$

Therefore, we know that the limit of $\mathcal{L}(\tau)$ exists and is:

$$\begin{aligned}\lim_{\tau\to\infty} \mathcal{L}(\tau) &= \|\mathbb{E}[X^s] - \mathbb{E}[X^t]\|_2 \\ &\quad - \|\mathbb{E}[X^s] - \mathbb{E}[X^0]\|_2 - \|\mathbb{E}[X^t] - \mathbb{E}[X^0]\|_2 \\ &= \|\mathbb{E}[X^s] - \mathbb{E}[X^t]\|_2 \\ &\quad - 1/2\|\mathbb{E}[X^s] - \mathbb{E}[X^t]\|_2 - 1/2\|\mathbb{E}[X^s] - \mathbb{E}[X^t]\|_2 \\ &= 0. \end{aligned} \tag{19}$$

□

Thanks to Theorem. III.2, the proposed triplet loss prevents the trivial values $\tau = 0$ or $\tau \to \infty$ from being chosen.

c) Robustness of $\mathcal{L}(\tau)$ w.r.t. different approximations: The algorithmic complexity of the loss $\mathcal{L}(\tau)$ depends on three main elements: (i) the exponential of the graph Laplacians, (ii) the Wasserstein distance W on the diffused source and target features and (iii) the size of the impostor. Different strategies can be used to simplify the overall complexity, including (but not exhaustively) the entropic regularization [13], the reduction of the size of $\mathcal{G}^0$, or a Chebychev approximation of the diffusion [19]. In the following, we study the capacity of $\mathcal{L}(\tau)$ to resist these approximations and thus to provide a similar global minimum.

Entropic regularization. The entropic regularization of W [13], as defined in Eq. (2), allows to overcome the limitations of the original problem due to the super-cubic complexity, the instability and the non uniqueness of the solution. The entropy-regularized version is several orders of magnitude faster (an η-approximation is computed in $O(n^2 \log(n)\eta^{-3})$ [20]). Using the same graphs as those of Figure 4a, Figure 5a shows the magnitude of the loss $\mathcal{L}(\tau)$ for different values of the regularization parameter ε. We can see that whatever the value of ε the shape of the resulting loss $\mathcal{L}(\tau)$ does not change much, meaning that the global minimum stays very close to the one corresponding to the maximum accuracy (see the dashed-line in Fig. 4b).

Size of the impostor. While Def. III.1 suggests to set the size of $\mathcal{G}^0$ to $\lceil \frac{m+n}{2} \rceil$, we study here the impact of reducing the number of nodes of $\mathcal{G}^0$ before computing the Wasserstein barycenter. The behavior of the loss $\mathcal{L}(\tau)$ is reported in Figure 5b for different reduction ratios (from 2 to 10). Interestingly, we can observe that even though the curves get smoother as the size of the impostor decreases, the global minimum changes very little, pointing to a relatively stable value for τ. Note that the connection probability of $\mathcal{G}^0$ has to adapt to the reduction ratio. For instance, if the size is reduced by a factor of 2, p^0 must be doubled.

Chebychev approximation of the diffusion. Finally, instead of calculating the exact heat kernel applied to the features, one can resort to polynomial approximations [21] of the diffusion. Following [19], we applied a Chebychev approximation of the exponential operator where the truncation order (degree of the final approximating polynomial) offers a trade-off between accuracy and computational speed. An illustration of the effect of various truncation orders on the loss $\mathcal{L}(\tau)$ is given in Figure 5c. Once again, the global minimum of $\mathcal{L}(\tau)$ appears to be very robust to changes in the degree of the approximation.

IV. Experimental Study

In this section, we perform two series of experiments dedicated to compare several optimal transport methods on graph DA tasks. In the first one, we use Contextual Stochastic Bloc Models [22] to generate synthetic data. The second one concerns the `ogbn-arxiv` graph [23] and aims at classifying papers published in a given year from articles published before. All experiments are written in Python and use the libraries `POT` [24] for optimal transport methods, `PyGSP` [25] for graph generation, and `NumPy` and `SciPy` for other computations. They run on a Intel© Core™ i5-5300U CPU @ 2.30GHz×2 processor and 15.5 GiB RAM on a Linux Mint 20 Cinnamon[1].

A. Domain Adaptation on synthetic data

a) Synthetic data generation: The process for generating a synthetic graph of size N is the following. For each of

[1]In case of acceptation, the code and the datasets will be made available.

the N nodes, a label $+1$ or -1 is chosen randomly with equal probability. For each node, a 1D feature is generated randomly by sampling a Gaussian distribution $\mathcal{N}(l, \sigma)$, where l is the node's label and σ is considered as a hyper-parameter. Links between nodes are sampled according to a Bernoulli distribution of probability $p_{l,l'}/N$ where l and l' are the node's labels. Therefore, a graph is described by its size N, its labels $l \in \{+1, -1\}^N$, its attributes $X \in \mathbb{R}^N$ and its adjacency matrix $A \in \{0, 1\}^{N \times N}$. We will add super-scripts s or t to designate either source or target quantities. The hyper-parameters are the sizes m and n of $\mathcal{G}^s$ and $\mathcal{G}^t$, the bandwidths σ^s and σ^t, and the connectivity matrices $\begin{pmatrix} p^s_{+1,+1} & p^s_{-1,+1} \\ p^s_{+1,-1} & p^s_{-1,-1} \end{pmatrix}$ and $\begin{pmatrix} p^t_{+1,+1} & p^t_{-1,+1} \\ p^t_{+1,-1} & p^t_{-1,-1} \end{pmatrix}$.

b) OT methods compared: We compare the following OT methods that can take advantage of the features and structure of both graphs, and only the source labels:

- The Wasserstein distance (`W`) [3] using only the features.
- The Gromov Wasserstein distance (`GW`) [6] taking only into account the structure of the graphs.
- The Fused Gromov-Wasserstein distance (`FGW`) [7] using both the feature and the structural information.
- The DA method based on the Wasserstein distance with Laplacian regularisation `OT_LAPLACE` [26].
- The DA method based on the Wasserstein distance with a group-lasso regularizer for the labels (`L1L2_GL`) [5].
- The Diffusion-Wasserstein distance (`DW_CV`) [8], where τ is tuned using a circular validation criterion.
- The Diffusion-Wasserstein distance, where τ is the minimizer of our loss function $\mathcal{L}(\tau)$, with (`DW`$\mathcal{L}_\varepsilon$) and without (`DW`$\mathcal{L}$) entropic regularization.

c) Experimental setup and results: 50 source/target graphs pairs are generated using the CSBM model described above. A transport map γ is learned for each method. Then the labels are transported according to the rule defined in (6). Using the ground truth labels, the accuracy is computed for each method. Each method is given 20 trials to find its best hyper-parameters, using random search and a circular validation criterion. For `FGW`, α is linearly sampled in $[10^{-6}, 1 - 10^{-6}]$. For `DW`, τ is logarithmically sampled in $[10^{-3}, 10^{-0.5}]$, and for all regularization based methods, the corresponding hyper-parameters range logarithmically in the interval $[10^{-2}, 10^{-0.5}]$. The synthetic data parameters are $n = m = 960$, $\sigma^s = 2$ and $\sigma^t = 4$, source and target connectivity matrices $\begin{pmatrix} 96 & 12 \\ 12 & 96 \end{pmatrix}$ and $\begin{pmatrix} 96 & 12 \\ 12 & 32 \end{pmatrix}$ respectively.

The results are reported in Figure 6. The accuracy scores of each method over the 50 graph pairs are plotted as a boxplot, displaying the median performance, the quartiles and the 10th and 90th percentiles as well as the outliers (i.e. out of the 10th and 90th percentiles). We can make the following comments. First, we can note that the Gromov Wasserstein distance is worse than random guessing. This behavior can probably be explained by a coupling matrix that permutes the classes.

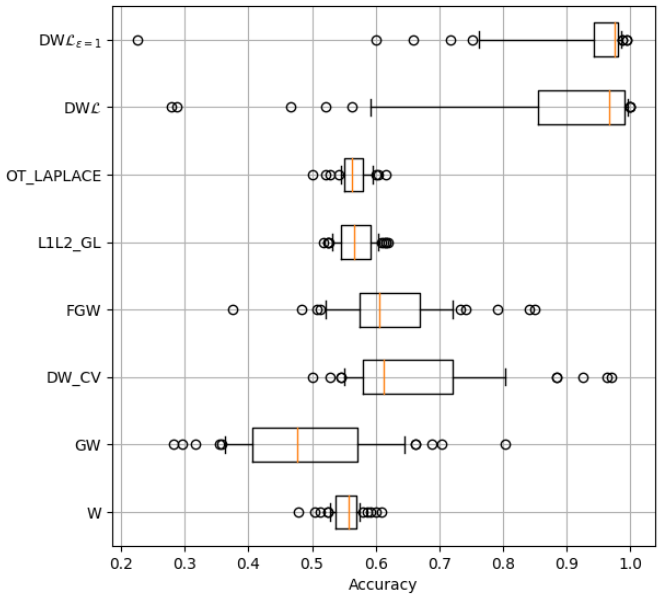


Fig. 6: Median, quartile and decile accuracy of various OT methods on the task of transferring the labels of $\mathcal{G}^s$ to $\mathcal{G}^t$.

Second, the approaches that take into account both the feature and the structural information (i.e. `FGW` and `DW`-based methods) outperform the competitors. Third, `DW`$\mathcal{L}_\varepsilon$ and `DW`$\mathcal{L}$ are better than any other method with a more stable behavior for the regularized version of our loss function. Finally, as expected, learning τ yields a significant improvement compared to `DW_CV` based on the circular validation.

B. Domain Adaptation on real data

This second series of experiments concerns the real `ogbn-arxiv` graph [23]. Although originally designed for node classification, we cast the problem as a Domain Adaptation task and we address it using the same methods as in the previous section. Each node of the graph represents a paper published in Arxiv. A link from one node to another indicates that this later is cited by the former. The feature of a node is an embedding of the paper's title and abstract, and it lies in $\mathbb{R}^{128}$. Nodes are labelled according to their corresponding subject area among 40 possible labels. Finally, each node is associated with a publication year.

In our setting, the source graph corresponds to the papers published before 2004. Its size is $m = 1279$. The target graph contains the articles published before 2005 ($n = 1666$ nodes). This makes the source graph a sub-graph of the target one and therefore, the DA accuracy is measured only on nodes of year 2005:

$$\mathrm{acc}(\gamma) = \frac{1}{387} \sum_{j=1280}^{1666} \delta_{\hat{l}^t_j = l^t_j}. \tag{20}$$

For `GW`, the source and target cost matrices are built from the shortest-distances in the graph. Because the graph is not connected and the solver cannot handle infinite costs between two nodes, infinite values are replaced by twice the

TABLE I: Computation time and test accuracy of various OT-based DA methods on the ogbn-arxiv graph restricted to years ≤ 2004 (for the source) and ≤ 2005 (for the target) with 1279 and 1666 nodes respectively.

Method	Computation time	Test accuracy
DW$\mathcal{L}_{\varepsilon=0.1}$	96s	30%
DW$\mathcal{L}$	319s	28%
L2L1_GL	870s	24%
DW_CV	130s	23%
FGW	2100s	22%
OT_LAPLACE	228s	18%
W	2s	18%
GW	275s	8%

longest length path. For FGW, the same cost matrices are used, along with the pairwise Euclidean distance between the features. For the hyper-parameter α, 10 values are sampled uniformly in $[0, 1]$ and the best one is selected using circular validation. For DW, the hyper-parameter τ is either determined by circular validation among 10 logarithmically spaced values in $[10^{-3}, 10^{1}]$, or chosen by minimizing $\mathcal{L}$. Finally, the size of the impostor graph $\mathcal{G}^0$ is set to 500 nodes.

The test accuracies are reported in Table I as well as the computation times. We can note that DW outperforms the competitors in terms of accuracy and remains much cheaper than FGW from a computational angle. The results also confirm that learning τ by minimizing the triplet loss $\mathcal{L}$ yields much better results than the circular validation.

V. Conclusion

In this paper, we presented an optimization method aiming at learning the diffusion time involved in the so-called Diffusion Wasserstein Distances. The experimental results give evidence that our strategy (i) allows to overcome the limitation of the widely used circular validation and (ii) outperforms the state-of-the-art OT-based domain adaptation methods. Note that we optimized so far only one τ used afterwards for both source and target graphs. In complex scenarios, the optimization of two diffusion times will be probably necessary. This will require to design a new way for building the impostor and derive new theoretical properties on the resulting loss function. We also plan to use our method on graph classification tasks. Finally, a promising perspective would consist in learning the ground metric which plays a key role in the coupling matrix.

References

[1] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, no. 1-2, pp. 151–175, 2010.

[2] I. Redko, E. Morvant, A. Habrard, M. Sebban, and Y. Bennani, *Advances in Domain Adaptation Theory*. Elsevier, ISBN 9781785482366, p. 187, Aug. 2019.

[3] G. Peyré and M. Cuturi, "Computational optimal transport," *Found. Trends Mach. Learn.*, vol. 11, no. 5-6, pp. 355–607, 2019.

[4] Y. Ollivier, H. Pajot, and C. Villani, Eds., *Optimal Transport - Theory and Applications*, ser. London Mathematical Society lecture note series. Cambridge University Press, 2014, vol. 413.

[5] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1853–1865, 2017.

[6] G. Peyré, M. Cuturi, and J. Solomon, "Gromov-wasserstein averaging of kernel and distance matrices," in *Int. Conf. on Machine Learning*, vol. 48, 2016, pp. 2664–2672.

[7] T. Vayer, L. Chapel, R. Flamary, R. Tavenard, and N. Courty, "Fused gromov-wasserstein distance for structured objects," *Algorithms*, vol. 13, no. 9, p. 212, 2020.

[8] A. Barbe, M. Sebban, P. Gonçalves, P. Borgnat, and R. Gribonval, "Graph diffusion wasserstein distances," in *ECML/PKDD (2)*, vol. 12458. Springer, 2020, pp. 577–592.

[9] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A dasvm classification technique and a circular validation strategy." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, 2010.

[10] A. Bellet, A. Habrard, and M. Sebban, *Metric Learning*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2015. [Online]. Available: https://doi.org/10.2200/S00626ED1V01Y201501AIM030

[11] L. V. Kantorovich, "Mathematical methods of organizing and planning production," *Management Science*, vol. 6, no. 4, pp. 366–422, 1939.

[12] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *ICCV*. IEEE Computer Society, 2009, pp. 460–467.

[13] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Conference NeurIPS, December 5-8, Nevada (US)*, C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, Eds., 2013, pp. 2292–2300.

[14] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *AAAI*. AAAI Press, 2015, pp. 4292–4293.

[15] E. N. Gilbert, "Random Graphs," *The Annals of Mathematical Statistics*, vol. 30, no. 4, pp. 1141 – 1144, 1959. [Online]. Available: https://doi.org/10.1214/aoms/1177706098

[16] J. M. Altschuler and E. Boix-Adserà, "Wasserstein barycenters can be computed in polynomial time in fixed dimension," *J. Mach. Learn. Res.*, vol. 22, pp. 44:1–44:19, 2021.

[17] M. Cuturi and A. Doucet, "Fast computation of wasserstein barycenters," in *ICML*, ser. JMLR Workshop and Conference Proceedings, vol. 32. JMLR.org, 2014, pp. 685–693.

[18] G. Luise, S. Salzo, M. Pontil, and C. Ciliberto, "Sinkhorn barycenters with free support via frank-wolfe algorithm," in *NeurIPS*, 2019, pp. 9318–9329.

[19] S. Marcotte, A. Barbe, R. Gribonval, T. Vayer, M. Sebban, P. Borgnat, and P. Gonçalves, "Fast multiscale diffusion on graphs," *arXiv preprint*, vol. abs/2104.14652, 2021.

[20] J. Altschuler, J. Weed, and P. Rigollet, "Near-linear time approximation algorithms for optimal transport via sinkhorn iteration," in *NIPS*, 2017, pp. 1964–1974.

[21] G. Phillips, *Interpolation and Approximation by Polynomials*, ser. CMS Books in Mathematics. Springer, 2003. [Online]. Available: https://books.google.fr/books?id=87vciTxMcF8C

[22] Y. Deshpande, S. Sen, A. Montanari, and E. Mossel, "Contextual stochastic block models," in *NeurIPS 2018, Montréal, Canada.*, 2018, pp. 8590–8602.

[23] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," *CoRR*, vol. abs/2005.00687, 2020.

[24] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, and T. Vayer, "Pot: Python optimal transport," *Journal of Machine Learning Research*, vol. 22, no. 78, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-451.html

[25] M. Defferrard, L. Martin, R. Pena, and N. Perraudin, "Pygsp: Graph signal processing in python," Oct. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1003158

[26] R. Flamary, N. Courty, A. Rakotomamonjy, and D. Tuia, "Optimal transport with Laplacian regularization," in *NIPS 2014, Workshop on Optimal Transport and Machine Learning*, Montréal, Canada, Dec. 2014. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01103076