Bures-Wasserstein Means of Graphs

Isabel Haasler
EPFL

Pascal Frossard
EPFL

Abstract

Finding the mean of sampled data is a fundamental task in machine learning and statistics. However, in cases where the data samples are graph objects, defining a mean is an inherently difficult task. We propose a novel framework for defining a graph mean via embeddings in the space of smooth graph signal distributions, where graph similarity can be measured using the Wasserstein metric. By finding a mean in this embedding space, we can recover a mean graph that preserves structural information. We establish the existence and uniqueness of the novel graph mean, and provide an iterative algorithm for computing it. To highlight the potential of our framework as a valuable tool for practical applications in machine learning, it is evaluated on various tasks, including k-means clustering of structured aligned graphs, classification of functional brain networks, and semi-supervised node classification in multi-layer graphs. Our experimental results demonstrate that our approach achieves consistent performance, outperforms existing baseline approaches, and improves the performance of state-of-the-art methods.

1 INTRODUCTION

Graphs, or networks, provide a compact representation for large data, describing for example biological, financial or social phenomena through interactions of elements in complex systems (Dong et al., 2019). Here, nodes represent entities, such as genes in a gene co-expression network (Sandhu et al., 2015), areas in the brain (Bullmore and Sporns, 2009), or users in a social network (Majeed and Rauf, 2020), and edges describe correlations or interactions between them. Due to their ability to model highly complex data in a wide range of applications, graphs have naturally become an increasingly important object of machine learning (Dong et al., 2020; Hu et al., 2020).

A key problem when considering graphs as statistical data objects is to define a mean for a set of graphs. Various notions of graph means have been proposed, and they can be cast as versions of the following general optimization problem: Given a set of graphs $G_1, \dots, G_m$ and a set of weights $\lambda_1, \dots, \lambda_m > 0$ with $\sum_{j=0}^{m} \lambda_j = 1$, the weighted mean of the graphs is defined as

$$\bar{G} = \arg\min_{G \in \mathcal{G}} \sum_{j=1}^{m} \lambda_j d(G, G_j)^2, \tag{1}$$

where $\mathcal{G}$ is a suitable space, or set, of graphs, and $d(\cdot, \cdot)$ is some dissimilarity measure for graphs, see, e.g., El Gheche et al. (2020); Kang et al. (2020); Kolaczyk et al. (2020); Lunagómez et al. (2021); Mercado et al. (2019b); Meyer (2022); Peyré et al. (2016); Xu et al. (2019b). For instance, in case $(\mathcal{G}, d)$ is a metric space, (1) is the corresponding sample Fréchet mean (Dubey and Müller, 2019). Thus, finding a meaningful graph mean relies on defining an appropriate distance $d(\cdot, \cdot)$ for graphs, which however, is a challenging task in graph analytics and machine learning (Tantardini et al., 2019; Wills and Meyer, 2020).

In this work we propose a novel graph mean that utilizes a recently introduced optimal transport distance for graphs (Petric Maretic et al., 2019; Maretic et al., 2022b) as the distance $d(\cdot, \cdot)$ in (1). It turns out that using this graph optimal transport distance corresponds to moving the graph mean problem to an embedding space of zero-mean normal distributions equipped with the Bures-Wasserstein distance. We therefore name our novel graph mean the *Bures-Wasserstein mean* of graphs. More precisely, in the embedding space a graph is represented by the probability distribution corresponding to a smooth graph signal that varies slowly over strongly connected nodes. A key idea of our work is that by taking the Wasserstein mean in the embedding space we retrieve mean graphs that may preserve the structural information captured by the smooth graph signals of the input graphs. We

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

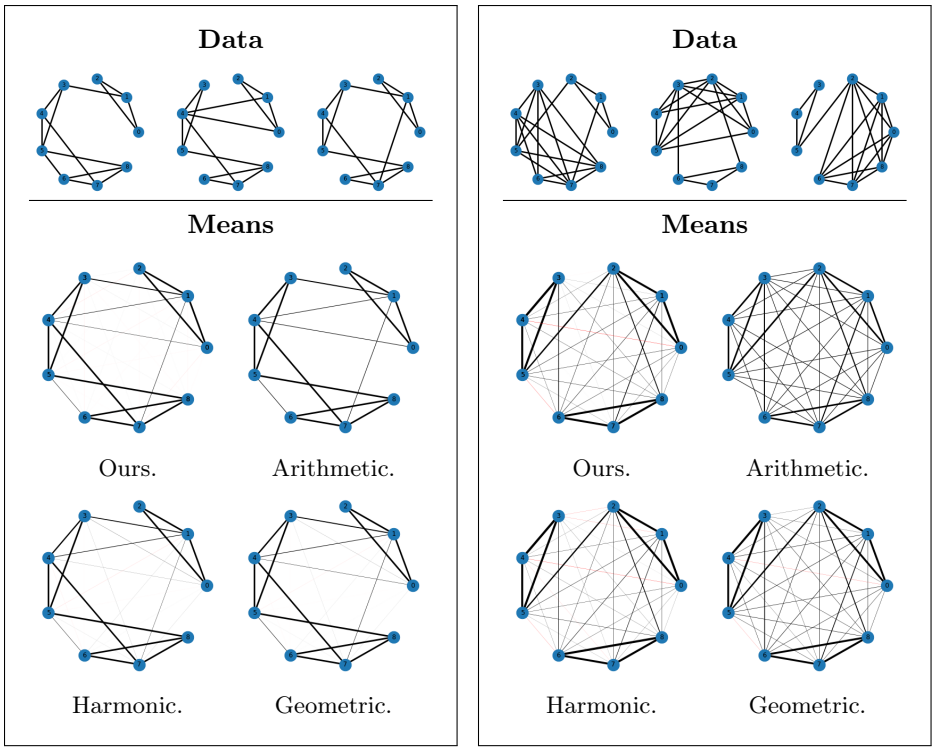

Figure 1: Illustration of different means for two sets of graphs. The edge widths are proportional to the absolute value of edge weights, and red edges correspond to negative edge weights. Our method is compared with the arithmetic mean, harmonic mean, and the geometric (Karcher) mean of the given graph Laplacians.

develop the theoretical foundation for the novel Bures-Wasserstein graph mean by proving the existence and uniqueness of a minimizer of the corresponding graph mean problem (1). Moreover, we provide an iterative algorithm for computing the Bures-Wasserstein graph mean. The potential of our proposed graph mean as a valuable tool for graph machine learning is illustrated through a range of numerical experiments, where it achieves consistent performance, outperforms existing baseline methods, and improves the performance of state-of-the-art methods.

The Bures-Wasserstein mean operates on a class of aligned and undirected graphs with edge weights that may be negative. Such signed weighted graphs are a powerful tool to describe complex systems with positive as well as negative interactions. Therefore, they naturally appear in many applications. For instance, in gene regulatory networks and brain networks negative edge-weights can be used to describe inhibition or repression, and in social networks negative edge-weights often model antagonistic relationships (Chen et al., 2016a,b; Olfati-Saber et al., 2007).

Overall, our main contributions are summarized as follows.

- We introduce a novel mean for a set of aligned graphs, which may have negative edge weights. It operates on the smooth graph signal distributions and can thus preserve structural information of the input graphs.
- The proposed Bures-Wasserstein mean is defined as the solution to an optimization problem of the form (1), and we prove the existence and uniqueness of a solution to this problem.
- We present a new iterative algorithm for computing the Bures-Wasserstein graph mean.
- We illustrate the potential of our framework for graph machine learning problems, like k-means clustering of graphs and functional brain network classification. Moreover, we demonstrate that the performance of a state-of-the-art method for semi-supervised node classification in multi-layer graphs can be improved with the Bures-Wasserstein mean.

Since the Bures-Wasserstein mean acts on signed weighted graphs that are aligned, it is most related to a class of means that act on the graphs' Laplacian matrices, for example arithmetic, harmonic, or various geometric and power means (El Gheche et al., 2020; Mercado et al., 2019a,b). Figure 1 illustrates the advantage of our suggested mean over existing notions of means for graph Laplacian matrices. Our Bures-Wasserstein mean is able to identify communities and captures connections between them in both settings. Other methods do not perform well for both tasks. In particular, the arithmetic mean $\sum_{j=1}^{m} \lambda_j L_j$ of a set of graph Laplacians $L_1, \ldots, L_m$ averages the weight of each edge individually. This local averaging does often not capture global properties of the graphs,

such as communities (Figure 1 right). On the other hand, the harmonic mean $\left(\sum_{j=1}^{m} \lambda_j L_j^\dagger\right)^\dagger$ is based on the arithmetic mean of the pseudo-inverses of the graph Laplacians. The pseudo-inverses describe correlations between all nodes on a global level, but do not describe local graph properties well. In the case of sparse input graphs, the harmonic mean is thus prone to creating spurious edges that are not present in any of the original graphs (Figure 1 left). The geometric mean (Lim and Pálfia, 2012) suffers from the same issue.

Other graph means focus on the case of unweighted graphs. For instance, the graph mean with respect to the Hamming distance is in fact a modification of the arithmetic mean of Laplacians for unweighted graphs (Meyer, 2022). We finally note that another optimal transport based graph mean has been defined using the Gromov-Wasserstein distance (Peyré et al., 2016; Vayer et al., 2019; Xu et al., 2019a,b), which is however fundamentally different from the Bures-Wasserstein mean. In particular, in contrast to Gromov-Wasserstein barycenters, our method considers edge weights and preserves node correspondences, and is thus suitable for multi-layer graphs or graphs with predefined node sets.

To the best of our knowledge, the novel Bures-Wasserstein graph mean is the first graph mean that utilizes the relation between graphs and their smooth graph signal representations, which is a powerful connection, that is well established for graph learning settings (Dong et al., 2016, 2019, 2020; Kalofolias, 2016). We expect that these theoretical and practical results provide a solid foundation for the use of the Bures-Wasserstein graph mean as a novel powerful operator in the graph machine learning toolbox.

2 BACKGROUND

In this section we present some background on graph signal processing and optimal transport theory that will be used in Section 3 to prove our main results.

2.1 Graph signal processing

Graph signal processing aims to generalize classical signal processing concepts, such as Fourier transforms and filters to the graph domain (Dong et al., 2020; Mateos et al., 2019; Ortega et al., 2018). Let $G = (V, E, W)$ be an undirected weighted graph, where V denotes a set of N vertices, E denotes a set of edges, and $W \in \mathbb{R}^{N\times N}$ is the weighted adjacency matrix with element W_{ik} denoting the weight[1] of edge (i, k). Moreover, define the degree matrix $D \in \mathbb{R}^{N\times N}$, which is a diagonal matrix with elements $D_{ii} = \sum_{k=1}^{n} W_{ik}$. Then, the signed weighted graph Laplacian matrix is defined as $L = D - W$.

From a signal processing perspective, features on a graph can be modeled as a graph signal, which is a mapping $x : V \to \mathbb{R}^N$. Consider the spectral decomposition of the corresponding graph Laplacian $L = V\Lambda V^T$, where $\Lambda \in \mathbb{R}^{N\times N}$ is a diagonal matrix containing the eigenvalues of L, and the columns of $V \in \mathbb{R}^{N\times N}$ are the corresponding eigenvectors. Then a graph Fourier transform of a graph signal x can be defined as $\tilde{x} = V^T x$. The graph Fourier transform provides a decomposition of the graph signal into different frequency components. To see this consider the total variation of a graph signal, defined as

$$TV(x) = x^T L x = \sum_{i,j=1}^{N} W_{ij}(x_i - x_j)^2. \tag{2}$$

By plugging in an eigenvector v_k to the corresponding eigenvalue λ_k, one can see that $TV(v_k) = v_k^T L v_k = \lambda_k$. Hence, small eigenvalues are associated with small variation over connected nodes, which can be interpreted as low frequency contents of the signal x.

Moreover, the spectral decomposition of the graph Laplacian gives rise to a factor analysis model (Tipping and Bishop, 1999) for graph signals, see Dong et al. (2016). Namely, assume that the (normalized) observed graph signal x is generated by $x = Vh + \epsilon$, where $h \in \mathbb{R}^N$ represent latent variables in Fourier space, and $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2 I_n)$ for a scalar σ_ϵ. By imposing the Gaussian prior $h \sim \mathcal{N}(0, \Lambda^\dagger)$ on the latent variables, one assumes that the energy of the graph signal lies mostly in the low frequency components. Using this prior in the factor analysis model generating x, the resulting noiseless signal has the distribution $x \sim \mathcal{N}(0, L^\dagger)$. That is, signals sampled from this distribution are smooth in the sense that they have small total variation (2), i.e., they vary slowly over strongly connected nodes.

2.2 Optimal transport

Optimal transport is a classical topic in mathematics that has originally been used in economics and operations research (Villani, 2021), and has now become a popular tool in fields such as machine learning (Arjovsky et al., 2017; Peyré et al., 2019), computer vision (Solomon et al., 2016), and signal processing (Elvander et al., 2020; Kolouri et al., 2017). The optimal transport problem seeks a map that moves the mass from one probability distribution to another one in the most efficient way with respect to an underlying cost, which is often defined as the squared Euclidean

[1]Note that we also allow for negative edge weights in contrast to the previous works (Maretic et al., 2022b; Petric Maretic et al., 2019).

distance. Let μ_0 and μ_1 be two probability measures[2] on the space $\mathbb{R}^n$. We denote the set of probability measures in $\mathbb{R}^n \times \mathbb{R}^n$ with marginals μ_0 and μ_1 as $\Pi(\mu_0, \mu_1)$. The measures $\pi \in \Pi(\mu_0, \mu_1)$ are called transport plans between the given marginals, and $\pi(x, y)$ is associate with the infinitesimal amount of mass transported from x to y. The (Kantorovich) optimal transport problem is to find the transport plan between μ_0 and μ_1 with the smallest associated transportation cost, that is to minimize

$$\left(\mathcal{W}_2(\mu_0, \mu_1)\right)^2 = \inf_{\pi \in \Pi(\mu_0, \mu_1)} \int_{\mathbb{R}^n \times \mathbb{R}^n} \|x - y\|_2^2 \, d\pi(x, y). \tag{3}$$

The optimal objective value of this problem defines the Wasserstein distance $\mathcal{W}_2(\cdot, \cdot)$ (Villani, 2021). This distance defines a metric on the space of absolutely continuous probability measures on $\mathbb{R}^n$ with finite second moment, denoted $\mathcal{P}_2(\mathbb{R}^n)$, and the Fréchet mean on the metric space $(\mathcal{P}_2(\mathbb{R}^n), \mathcal{W}_2)$ is called the Wasserstein barycenter (Agueh and Carlier, 2011). However, note that the Wasserstein barycenter problem can even be defined for measures that are not necessarily absolutely continuous. Given a set of measures $\mu_1, \ldots, \mu_m$ and a set of weights $\lambda_1, \ldots, \lambda_m > 0$ such that $\sum_{j=1}^n \lambda_j = 1$, the weighted Wasserstein barycenter is the minimizer of the optimization problem

$$\bar{\mu} = \arg\min_{\mu} \sum_{j=1}^{m} \lambda_j \left(\mathcal{W}_2(\mu, \mu_j)\right)^2. \tag{4}$$

Although the optimal transport problem (3) is in general challenging to solve, in case the given probability distributions are Gaussian it has an analytical solution (Dowson and Landau, 1982; Olkin and Pukelsheim, 1982; Takatsu, 2010). More precisely, consider two measures $\mu_0 \sim \mathcal{N}(0, \Sigma_0)$ and $\mu_1 \sim \mathcal{N}(0, \Sigma_1)$, describing zero-mean normal distributions with covariance matrices Σ_0, Σ_1. Then the Wasserstein distance between these probability distributions is given by

$$\begin{aligned}\left(\mathcal{W}_2(\mu_0, \mu_1)\right)^2 = \operatorname{trace}(\Sigma_0) + \operatorname{trace}(\Sigma_1) \\ - 2 \cdot \operatorname{trace}\left(\sqrt{\Sigma_0^{1/2} \Sigma_1 \Sigma_0^{1/2}}\right).\end{aligned} \tag{5}$$

Unfortunately, the barycenter (4) of a set of normal distributions given by $\mu_j \sim \mathcal{N}(0, \Sigma_j)$, for $j = 1, \ldots, m$, does not have an analytical solution when $m > 2$ (Agueh and Carlier, 2011). However, in the case that the given distributions are non-degenerate it is known that the barycenter is also a zero mean normal distribution $\bar{\mu} \sim \mathcal{N}(0, \bar{\Sigma})$, and its covariance matrix $\bar{\Sigma}$ is the unique positive definite solution to the matrix equation (Agueh and Carlier, 2011)

$$S = \sum_{j=1}^{m} \lambda_j \left(S^{1/2} \Sigma_j S^{1/2}\right)^{1/2}. \tag{6}$$

Note that zero-mean Gaussians are fully parametrized by their covariance matrices. In fact, there are important connections between the space of normal distributions, and the space of symmetric positive definite matrices. Namely, the Wasserstein distance also defines a metric on the space of symmetric positive-definite matrices, and is in this context also called the Bures-Wasserstein metric (Bhatia et al., 2019b). The barycenter problem (4) can thus be interpreted as taking the mean of a set of symmetric positive-definite matrices $\Sigma_0, \ldots, \Sigma_m$ with respect to the Bures-Wasserstein metric (Bhatia et al., 2019a), and this problem is also solved by the solution to the matrix equation (6). In the next Section we will extend these results in order to define a novel mean for a set of graphs.

3 BURES-WASSERSTEIN GRAPH MEANS

The core idea of the present work is that embedding graphs into a space of distributions equipped with the distance (5) provides an elegant and simple framework for considering graphs as statistical data. Specifically, in this section we define a notion of graph mean (1) through an instance of the Wasserstein barycenter problem (4) and characterize it as the solution of a matrix equation similar to (6). Based on this, we provide an algorithm for computing the graph mean.

3.1 Graph mean

In this work we will use the following rather general assumption.

Assumption 3.1. Let $L \in \mathbb{R}^{N \times N}$ be a signed weighted graph Laplacian that is positive semi-definite and has only one zero eigenvalue.

This assumption is discussed in detail in the supplementary material. Here we note that all connected graphs with non-negative edge weights have Laplacians that satisfy Assumption 3.1, and conversely all signed weighted graphs with Laplacian matrices satisfying Assumption 3.1 are connected (Chen et al., 2016a,b).

Graphs with semi-positive Laplacian matrix can further be represented by their smooth graph signal distribution, as described in Section 2.1. Based on the connection between graphs and their smooth signal representation, we define the Bures-Wasserstein distance for graphs as follows.

[2]Note that a probability measure defines a probability distribution. We often use the terms measure and distribution interchangeably.

Definition 3.2. Let G_0 and G_1 be two aligned graphs with signed weighted graph Laplacian matrices L_0 and L_1 satisfying Assumption 3.1, and consider two measures $\mu_{G_j} \sim \mathcal{N}(0, L_j^\dagger)$, for $j = 0, 1$. The Bures-Wasserstein distance between the graphs G_0 and G_1 is defined as $d_{BW}(G_0, G_1) := \mathcal{W}_2(\mu_{G_0}, \mu_{G_1})$.

Let $\mathcal{G}$ be a set of aligned signed weighted graphs with N nodes and Laplacians that satisfy Assumption 3.1. We define the Bures-Wasserstein mean of the graphs $G_1, \dots, G_m \in \mathcal{G}$ analogously to (1) as the solution to

$$\begin{aligned} \bar{G} &= \arg\min_{G \in \mathcal{G}} \sum_{j=1}^{m} \lambda_j d_{BW}(G, G_j)^2 \\ &= \arg\min_{G \in \mathcal{G}} \sum_{j=1}^{m} \lambda_j \mathcal{W}_2(\mu_G, \mu_{G_j})^2. \end{aligned} \tag{7}$$

We now characterize the solutions to the optimization problem (7). Note that the Bures-Wasserstein graph mean has similarity with the Wasserstein barycenter (4). However, it is important to note that the theory from Section 2.2 cannot be applied directly. Firstly, the feasibility set in (7) is more restrictive than in (4), since the barycenter distribution μ_G must be of the specific form $\mu_G \sim \mathcal{N}(0, L^\dagger)$, where L satisfies Assumption 3.1. Secondly, smooth graph signal distributions are degenerate Gaussians, but the characterization (6) and the uniqueness of the barycenter have only been proved for non-degenerate Gaussians (Agueh and Carlier, 2011; Xia et al., 2014). Thus, extending the theory in Section 2.2 we now prove the existence and uniqueness of a solution $\bar{G}$ to problem (7).

Theorem 3.3. *Let G_j, for $j = 1, \dots, m$, be graphs with signed weighted graph Laplacians L_j that satisfy Assumption 3.1. Then the Bures-Wasserstein mean* (7) *of these graphs is described by the Laplacian matrix $L = S^\dagger$, where S is the unique positive semi-definite symmetric solution to*

$$S = \sum_{j=1}^{m} \lambda_j \left(S^{1/2} L_j^\dagger S^{1/2} \right)^{1/2} \tag{8}$$

that satisfies range$(S) = \mathcal{R} = (\text{span}\{\mathbf{1}_N\})^\perp$.

Proof sketch: We first prove the existence and uniqueness of a solution to the Wasserstein barycenter problem (4) in the special case, where the given distributions are degenerate Gaussians, which are embeddings of graphs in $\mathcal{G}$. We then show that this solution in fact describes a smooth graph signal. An essential observation here is that Assumption 3.1 guarantees that the degeneracy of all given measures $\mu_j \sim \mathcal{N}(0, L_j^\dagger)$ lies in the subspace $\mathcal{R} = (\text{span}\{\mathbf{1}_N\})^\perp \subset \mathbb{R}^N$.

See the supplementary material for a full proof. □

Algorithm 1 Bures-Wasserstein mean of graphs.

Given: $L_1, \dots, L_m \in \mathbb{R}^{N \times N}$ satisfying Assumption 3.1, and initial SPD matrix $S \in \mathbb{R}^{N \times N}$

$\Sigma_j \leftarrow \left(L_j + \frac{1}{N} \mathbf{1}_{N \times N} \right)^{-1}$ for $j = 1, \dots, m$

while Not converged **do**

$$S \leftarrow S^{-1/2} \left(\sum_{j=1}^{m} \lambda_j (S^{1/2} \Sigma_j S^{1/2})^{1/2} \right)^2 S^{-1/2},$$

end while

Return: $L \leftarrow S^{-1} - \frac{1}{N} \mathbf{1}_{N \times N}$

3.2 Algorithm

We now introduce a computational method for finding the Bures-Wasserstein graph mean, see Algorithm 1. This method is based on a fixed point iteration for solving the matrix equation (6) that was first introduced in Álvarez-Esteban et al. (2016, Theorem 4.2). This fixed point iteration can be directly applied to find a solution to the matrix equation (8), which characterizes the Bures-Wasserstein graph mean. In addition, we exploit spectral properties of graph Laplacians as described in the following.

Proposition 3.4. *Let $L_1, \dots, L_m \in \mathbb{R}^{N \times N}$ satisfy Assumption 3.1, and let $\mathbf{1}_{N \times N} \in \mathbb{R}^{N \times N}$ denote a matrix of ones. Then the matrices $\Sigma_j = L_j^\dagger + \frac{1}{N} \mathbf{1}_{N \times N}$, for $j = 1, \dots, m$ are symmetric and strictly positive definite. Moreover, the Bures-Wasserstein mean* (7) *of the graphs with Laplacians $L_1, \dots, L_m$ is the graph with Laplacian $L = \bar{\Sigma} - \frac{1}{N} \mathbf{1}_{N \times N}$, where $\bar{\Sigma}$ is the Bures-Wasserstein barycenter of $\Sigma_1, \dots, \Sigma_m$.*

Proof. See supplementary material. □

Proposition 3.4 permits to transform the singular graph Laplacians into positive definite matrices. This makes the algorithm efficient and stable, as we do not need to take square roots and pseudo-inverses of singular matrices. The fixed-point iteration in Algorithm 1 inherits the convergence guarantee proved in Álvarez-Esteban et al. (2016), and in practice we typically observe convergence within a few iterations.

3.3 Extensions

A special case of the Bures-Wasserstein graph mean problem is the setting where only two graphs are given, i.e., where $m = 2$ in (7). Namely, by defining the weights as $\lambda_1 = t$ and $\lambda_2 = 1 - t$ for $t \in (0, 1)$, the solution to (7) can be understood as the interpolation between two graphs. For the classical Wasserstein problem this is also called displacement interpolation, and the solution for $t \in (0, 1)$ defines a geodesic in

Table 1: Difference of the perturbed graphs' mean to the original graph with respect to several metrics.

	B-W distance	Laplacian	Covariance	Degree centrality	Modularity	Participation coefficient
B-W mean	$\mathbf{0.097 \pm 0.043}$	$\mathbf{2.26 \pm 0.11}$	$\mathbf{0.15 \pm 0.12}$	$\mathbf{0.036 \pm 0.002}$	$\underline{\mathbf{0.011 \pm 0.003}}$	$\underline{\mathbf{0.25 \pm 0.09}}$
Arithmetic	0.170 ± 0.073	$\underline{\mathbf{1.85 \pm 0.10}}$	0.26 ± 0.18	$\underline{\mathbf{0.026 \pm 0.003}}$	0.015 ± 0.003	0.37 ± 0.13
Harmonic	$\underline{\mathbf{0.095 \pm 0.037}}$	2.87 ± 0.19	$\underline{\mathbf{0.13 \pm 0.10}}$	0.047 ± 0.004	$\mathbf{0.012 \pm 0.002}$	$\mathbf{0.26 \pm 0.09}$
Geometric	0.286 ± 0.067	6.42 ± 0.31	0.34 ± 0.14	0.091 ± 0.009	0.015 ± 0.008	0.40 ± 0.13

the space $(\mathcal{P}_2(\mathbb{R}^n), \mathcal{W}_2)$ (McCann, 1997; Villani, 2021). If the two given measures are Gaussian, there is a closed-form solution to the displacement interpolation problem. These results can be extended to problem (7), which gives a closed-form solution of the graph Laplacian interpolation. For completeness, we present the following result for this setting.

Theorem 3.5. *Let G_0 and G_1 be two graphs with signed weighted graph Laplacians L_0 and L_1 that satisfy Assumption 3.1. Then the Bures-Wasserstein mean of these two graphs with weights $1-t$ and t for $t \in (0,1)$ has the signed weighted graph Laplacian $L_t = S_t^\dagger$, where*

$$S_t = L_0^{1/2}\left((1-t)L_0^\dagger + t\left(L_0^{\dagger/2} L_1^\dagger L_0^{\dagger/2}\right)^{1/2}\right)^2 L_0^{1/2}. \quad (9)$$

Proof. See supplementary material. □

It is worth noting the connection between the analytical solution in the interpolation setting from Theorem 3.5 and the iteration in Algorithm 1. In particular, in the case that $m = 2$, and when choosing the starting point $S_0 = L_1^\dagger$, the iteration in Algorithm 1 corresponds to (7.3) and thus converges in one step.

Finally, we note that an extension of the Bures-Wasserstein distance for graphs can take into account filters on the graph signal distributions. This permits to emphasize different types of spectral information in the graph comparison (Maretic et al., 2022a). For instance, employed with a low-pass filter, the Bures-Wasserstein distance predominantly captures differences in the global graph structures, whereas a high-pass filter focuses on local structures. These distances can also be utilized in the graph mean problem (7) when the graph filter is bijective for the set of Laplacian matrices satisfying Assumption 3.1. For more details see the supplementary material.

4 EXPERIMENTS

In this Section we illustrate the behavior of the Bures-Wasserstein graph mean experimentally in several machine learning problems. In particular, we show that in many settings it provides a better representation of a set of aligned graphs than other graph means. In all experiments we set the weights in (7) to $\lambda_j = 1/m$ for $j = 1, \ldots, m$, where m denotes the number of graphs.

4.1 Graph fusion

We first illustrate that the Bures-Wasserstein mean graph preserves graph characteristics of the input graphs well. We create a stochastic block model graph with 50 nodes and 5 communities. The edge probability within the clusters is 0.3, and the edge probability between clusters is 0.1. From this graph we generate 100 graphs by removing and adding 10 random edges, respectively, while enforcing that all graphs are connected. The Bures-Wasserstein barycenter of these 100 graphs is computed using Algorithm 1 and compared to the original graph in terms of several metrics. We also compute the same metrics for the arithmetic, harmonic, and geometric mean of the graph Laplacians. This experiment is repeated 1000 times, and the mean error with respect to the different graph metrics are presented in Table 1. Here, the error in the Laplacian and covariance matrix is measured in Frobenius-norm. The degree centrality (for weighted graphs also called strength), modularity, and participation coefficient are computed as defined in Oehlers and Fabian (2021). The difference in degree centrality and participation coefficient are presented in mean square error over all nodes, and the modularity difference in absolute value.

We see that our proposed method is among the two best-performing methods for all tested graph metrics. Degree centrality describes local properties of the graph, and is thus represented well by the arithmetic mean, which takes the average weight of each edge. Modularity on the other hand describes the global graph structure, and this is well represented by the harmonic mean. The participation coefficient typically captures both local and global phenomena. We see that the Bures-Wasserstein mean preserves all tested graph metrics well, which shows that it successfully fuses both local and global structural information from the input graphs.

We note here that the geometric mean is not preserving any of the graph metrics well. As the methods based on the geometric mean did not perform as well as the other baseline methods in our conducted experiments,

we chose to omit them in the following sections for clarity and brevity.

4.2 K-means clustering of graphs

Next we apply our proposed framework to an unsupervised learning setting. The task in this experiment is to identify graphs that have the same number of communities using k-means clustering. The dataset is constructed as follows. We generate connected graphs with 50 nodes and a varying number of communities, ranging from 1 up to N_C, where $N_C \in \{4, 5, 6\}$. These make up the N_C classes that we aim to identify, and each class contains 20 graphs. The probability for two nodes in a community to be connected is p, where $p \in \{0.2, 0.25, 0.3\}$. The communities are connected on a line, that is, each community is connected to at most two other communities, and the connectivity between the neighboring communities is randomly picked from two possible patterns. Each of these two patterns constructs one edge between the communities, between a different set of nodes[3]. For each class of graphs, the edges within clusters thus have high variability, while the edges between clusters have low variability. Successfully clustering these highly structured types of graphs requires the ability to capture both the community structure as well as the inter-community structure.

We use a k-means clustering method, where the centroids are computed using either the Bures-Wasserstein mean, or the arithmetic or harmonic mean of the graph Laplacians, and the distance between graphs is measured according to the respective distance (Bures-Wasserstein distance, Frobenius norm of Laplacians, or Frobenius norm of pseudo-inverse Laplacians). Moreover, we evaluate also the topological clustering methods in Songdechakraiwut et al. (2021), which compare persistence barcodes of the graphs using the Wasserstein distance. Note that the pure topological mean does not preserve the node alignment, and to address this, the authors propose a hybrid method of topological and arithmetic mean.

The classification performance of the different methods and for different values of N_C (with fixed $p = 0.2$) and respectively different values of p (with fixed $N_C = 5$) are summarized in Figure 2. One can see that for a small number of clusters, or respectively high connectivity within the communities, many of the tested methods perform similarly well. However, the graph classification becomes difficult as the number of clusters increases, and as the connectivity within the communities decreases. In these cases the Bures-Wasserstein method significantly outperforms the methods based on arithmetic and harmonic means. Therefore, we can conclude that the distance in Definition 3.2 captures similarity between the graphs well, and the mean graph (7) provides a useful representation of the cluster centroids.

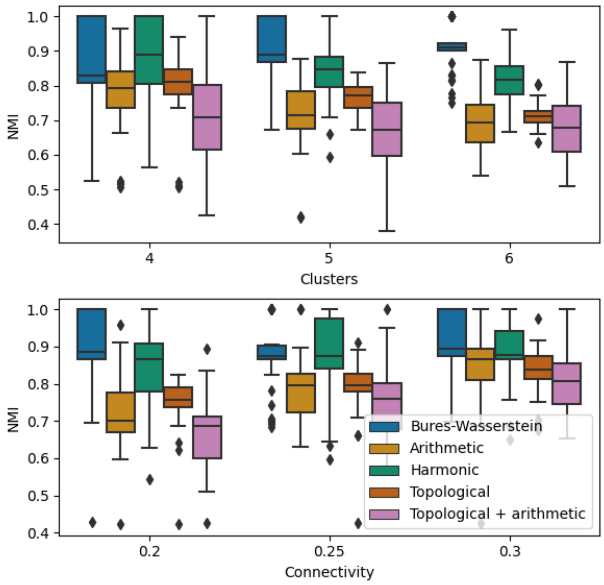

Figure 2: Normalized mutual information of graph classification with k-means clustering for different values of number of classes (clusters), and inter-community edge probability (connectivity).

4.3 Classification of brain states

We now show that the Bures-Wasserstein mean provides a useful centroid for graphs describing real world data. In particular, we utilize our methods to classify brain activity measurements from Schalk et al. (2004); Goldberger et al. (2000)[4]. A subject is performing, or imagining, different types of motor tasks. Following a cue, the subject is contracting, or imagining to contract, the left hand vs. right hand, or the hands vs. feet. Brain signals are measured from 64 sensors placed around the head. For the brain reaction after each cue we construct a graph by assigning a node to each sensor. We then use a 4 second interval after the cue to compute the envelope correlations between each set of sensors and assign it as an edge weight between the corresponding nodes. We remove all edges that have an edge weight smaller than a threshold parameter, which is picked as large as possible while still resulting in a connected graph. Each experiment trial consists of 15 cues, which results in a set of 15 graphs that are split into the two classes (either left vs. right hand, or hands vs. feet). These sets of 15 graphs are constructed for 109 subjects, and 3 experiment repetitions for each subject, resulting in a total of 327 trials. We

[3]Example graphs of this dataset can be found in the supplementary material.

[4]The dataset can be found at `https://mne.tools/0.18/generated/mne.datasets.eegbci.load_data.html`.

Table 2: Functional brain network misclassification rate (%).

	Bures-Wasserstein	Arithmetic	Harmonic	Topological	Topological + arithmetic
Motor execution: left vs. right hand	$\mathbf{13.2 \pm 8.3}$	24.7 ± 11.8	14.8 ± 8.7	33.6 ± 9.6	$\mathbf{8.7 \pm 10.4}$
Motor imagery: left vs. right hand	$\mathbf{12.7 \pm 8.9}$	24.2 ± 11.8	15.3 ± 10.5	33.6 ± 9.0	$\mathbf{7.4 \pm 9.8}$
Motor execution: hands vs. feet	$\mathbf{12.5 \pm 8.5}$	23.7 ± 11.1	15.2 ± 9.6	32.2 ± 9.6	$\mathbf{10.0 \pm 10.7}$
Motor imagery: hands vs. feet	$\mathbf{12.9 \pm 8.9}$	25.5 ± 10.2	15.2 ± 9.8	33.4 ± 9.2	$\mathbf{8.1 \pm 9.9}$

compute the mean of the graphs for each of the two classes in the experiment, using the Bures-Wasserstein mean, the arithmetic and harmonic graph Laplacian mean, and the topological means in Songdechakraiwut et al. (2021). We then compute the misclassification rate as the rate of graphs that are closest to the centroid of the wrong class with respect to the respective graph distance. The resulting misclassification rate for the different sets of experiments, and using different types of graph means, are reported in Table 2. In all experiments the Bures-Wasserstein mean provides centroids that describe the graphs in the two classes better than the centroids resulting from the arithmetic, harmonic, and topological means. The only mean that outperforms the Bures-Wasserstein mean is the hybrid mean combining topological and arithmetic information, which was developed specifically for functional brain networks. The fact that this hybrid mean performs significantly better than the pure arithmetic or topological mean indicates that a hybrid mean that combines the Bures-Wasserstein mean with topological information might drastically improve the clasification performance. Developing a method for computing this mean is a promising direction of future research. Note that the smooth graph signal representation for graphs is motivated by the standard assumption that real data samples are often well described by a normal distribution. It is therefore reasonable to expect that the Bures-Wasserstein mean, which utilizes the smooth graph signal distributions, represents real world data well in many scenarios.

4.4 Learning on multi-layer graphs

Finally, we show that the Bures-Wasserstein mean can serve as a regularizer for semi-supervised learning in a multi-layer graph. A multi-layer graph is a set of graphs with the same set of nodes and different edges, which typically describe different types of interactions in a system. We suggest the following classification method to learn unobserved labels from the structural information of the Bures-Wasserstein mean and some observed node labels, based on the state-of-the-art method introduced in Mercado et al. (2019a). We consider that there are k classes of nodes, and let $Y^{(r)} \in \mathbb{R}^N$ be defined by $Y_i^{(r)} = 1$ if the i-th node belongs to class r, and $Y_i^{(r)} = 0$ otherwise. We then solve

$$f^{(r)} = \arg\min_{f \in \mathbb{R}^N} \|f - Y^{(r)}\|^2 + \rho f^T \bar{\mathcal{L}} f, \qquad (10)$$

where $\bar{\mathcal{L}}$ is some mean of a set of normalized graph Laplacians $\mathcal{L}_1, \ldots, \mathcal{L}_m$ describing the multiple layers, and $\rho > 0$ is a parameter that determines how much to rely on the given observations or the structure of the graphs. Then the i-th node is assigned to the class $\arg\max_r \{f_i^{(r)}; r = 1, \ldots, k\}$. Here, we propose to define $\bar{\mathcal{L}}$ in the optimization problem (10) as the Bures-Wasserstein mean of the given graph Laplacians. The set of normalized graph Laplacians are defined as in Mercado et al. (2019a), i.e., $\mathcal{L} = D^{-1/2} L D^{-1/2}$, where L is the combinatorial graph Laplacian, and D the degree matrix, as defined in Section 3. We use the normalized Laplacians $\mathcal{L}_j$ for each layer j here[5] to define the smooth graph signals $\mu_{G_j} \sim \mathcal{N}(0, \mathcal{L}_j^\dagger)$.

We compare the performance of the node classification task utilizing the Bures-Wasserstein mean with the methods in Mercado et al. (2019a), which utilize different types of power means to define an average graph Laplacian $\bar{\mathcal{L}}$ in (10). More precisely, the family of power means of the Laplacians $\mathcal{L}_j$, for $j = 1, \ldots, m$, is defined as $\bar{\mathcal{L}}^p = (\sum_{j=1}^m \mathcal{L}_j^p)^{1/p}$. In particular, $p = 1$ and $p = -1$ correspond to the arithmetic and harmonic mean, respectively. In addition to these two means, we also compare our method to the $p = -10$ power mean. The regularization parameter is picked as the element from the set $\{0.1, 1, 10\}$ that gives the best results on the tested data. That is, $\rho = 10$ for the arithmetic mean, $\rho = 0.1$ for the harmonic and the power mean with $p = -10$ (as in Mercado et al. (2019a)), and $\rho = 1$ for the Bures-Wasserstein mean.

We consider multi-layer graphs constructed as in Mercado et al. (2019a), where a 10-nearest neighbour graph is constructed for each layer of the real-world datasets *3sources*, *BBC*, *BBCS* and *Wikipedia*. Several metrics for these datasets are presented in the supplementary material. The average classification errors from 50 trials for each dataset and with varying percentage of

[5] In contrast to the previous examples, using normalized graph Laplacians is possible in this application, since we do not need to explicitly recover a graph representation $\bar{G}$ corresponding to the mean graph Laplacian $\bar{\mathcal{L}}$.

Table 3: Test error (%) of node classification for different percentage of observed node labels.

p	5%	10%	15%	20%	25%	30%	40%
3sources							
1	26.8	23.4	21.4	16.7	**14.8**	**13.3**	**11.2**
−1	**22.5**	**21.7**	22.7	19.3	18.8	17.9	17.7
−10	31.2	22.4	**20.6**	**16.2**	**14.9**	**13.5**	12.0
BW	**24.9**	**21.3**	**20.2**	**15.9**	**14.9**	13.6	**11.9**
BBC							
1	22.6	17.1	12.9	10.6	9.3	**8.2**	**7.2**
−1	**16.8**	**11.7**	**10.0**	**9.5**	**9.0**	8.5	8.2
−10	26.7	16.4	12.4	10.5	9.6	8.6	7.7
BW	**20.0**	**13.3**	**10.5**	**9.0**	**8.4**	**7.4**	**6.9**
BBCS							
1	16.1	13.3	11.0	8.8	7.3	6.3	5.2
−1	**12.3**	**8.2**	**6.5**	**5.9**	**5.4**	**5.1**	**4.8**
−10	23.7	13.8	9.7	7.7	6.3	5.6	4.9
BW	**15.8**	**10.4**	**8.0**	**6.5**	**5.6**	**5.2**	**4.7**
Wikipedia							
1	59.6	53.6	48.3	44.5	41.2	38.4	35.3
−1	**49.8**	**40.1**	**36.4**	**34.7**	**33.8**	**33.0**	**32.0**
−10	54.6	43.3	38.4	35.9	34.4	33.2	**31.9**
BW	**49.0**	**39.6**	**36.0**	**34.5**	**33.5**	**32.9**	**32.0**

observed nodes are presented in Table 3. One can see that the method based on the Bures-Wasserstein mean performs well in all tested scenarios, and is among the two best-performing methods in almost all settings. In contrast, the other types of averages are less reliable, and may perform well in some cases, but poorly in others. For instance, note that *3sources* is the smallest dataset, and thus the 10-nearest neighbor graphs are the most strongly connected. This makes the structural information less informative, and none of the power means performs well for every tested percentage of observed node labels. The *Wikipedia* dataset contains the largest number of labels, which makes the node classification task on this dataset the most challenging, and a graph mean that accurately represents the structural information from the different layers becomes crucial. In this case especially our proposed Bures-Wasserstein mean outperforms the power means. Thus, incorporating our proposed framework in this state-of-the-art method for semi-supervised learning, we are able to improve on its performance.

5 CONCLUSION AND OUTLOOK

We introduced a novel framework for defining the mean of a set of graphs, which utilizes the connection between graphs and their smooth graph signal distributions as well as the Wasserstein metric. By combining these two concepts, the recovered mean graph can preserve structural information of the input graphs that is captured by the respective smooth graph signals. By extending previous results for Wasserstein barycenters of non-degenerate Gaussians to a class of degenerate Gaussians, we proved the existence and uniqueness of the mean graph, and provide an effective iterative algorithm for finding it. We evaluated the proposed approach on various machine learning tasks, including k-means clustering of structured graphs, classification of functional brain networks, and semi-supervised node classification in multi-layer graphs. Our experimental results show that our approach achieves consistent performance that is competitive with existing baseline approaches and can improve state-of-the-art methods.

Our results indicate that the proposed Bures-Wasserstein framework for graphs serves as a powerful tool for practical machine learning applications. Since defining means is a key ingredient of statistical analysis our approach may lead to various novel methods for graph machine learning. For example, the Bures-Wasserstein mean together with the corresponding Fréchet variance can be used to build generative models for graphs. More broadly, we see this work as a first step towards a general and flexible statistical framework for graph data. We anticipate that existing methods from optimal transport theory may be extended to admit statistical inference, regression, and high order interpolation methods for populations of graphs (Chen et al., 2018a,b; Karimi et al., 2020; Lambert et al., 2022).

It should be noted that the methods presented in this work are currently limited to the setting of aligned graphs where a matching between the nodes in the given graphs is known. This is the case in various applications, e.g., multi-layer networks and brain networks estimated from EEG data as presented in this work. However, in many other applications graphs are not aligned and may be of different sizes. Another future direction of work is thus to develop computational methods that extend our approach to this different setting.

Acknowledgements

The authors thank the anonymous reviewers for useful comments. This work was supported by the Knut and Alice Wallenberg foundation under grant KAW 2021.0274.

References

Agueh, M. and Carlier, G. (2011). Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924.

Álvarez-Esteban, P. C., Del Barrio, E., Cuesta-Albertos, J., and Matrán, C. (2016). A fixed-

point approach to barycenters in Wasserstein space. *Journal of Mathematical Analysis and Applications*, 441(2):744–762.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. *International Conference on Machine Learning*, pages 214–223.

Bhatia, R., Jain, T., and Lim, Y. (2019a). Inequalities for the Wasserstein mean of positive definite matrices. *Linear Algebra and its Applications*, 576:108–123.

Bhatia, R., Jain, T., and Lim, Y. (2019b). On the Bures-Wasserstein distance between positive definite matrices. *Expositiones Mathematicae*, 37(2):165–191.

Bullmore, E. and Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews Neuroscience*, 10(3):186–198.

Chen, W., Liu, J., Chen, Y., Khong, S. Z., Wang, D., Başar, T., Qiu, L., and Johansson, K. H. (2016a). Characterizing the positive semidefiniteness of signed Laplacians via effective resistances. *IEEE Conference on Decision and Control (CDC)*, pages 985–990.

Chen, Y., Conforti, G., and Georgiou, T. T. (2018a). Measure-valued spline curves: An optimal transport viewpoint. *SIAM Journal on Mathematical Analysis*, 50(6):5947–5968.

Chen, Y., Georgiou, T. T., and Tannenbaum, A. (2018b). Optimal transport for Gaussian mixture models. *IEEE Access*, 7:6269–6278.

Chen, Y., Khong, S. Z., and Georgiou, T. T. (2016b). On the definiteness of graph Laplacians with negative weights: Geometrical and passivity-based approaches. *American Control Conference (ACC)*, pages 2488–2493.

Dong, X., Thanou, D., Frossard, P., and Vandergheynst, P. (2016). Learning Laplacian matrix in smooth graph signal representations. *IEEE Transactions on Signal Processing*, 64(23):6160–6173.

Dong, X., Thanou, D., Rabbat, M., and Frossard, P. (2019). Learning graphs from data: A signal representation perspective. *IEEE Signal Processing Magazine*, 36(3):44–63.

Dong, X., Thanou, D., Toni, L., Bronstein, M., and Frossard, P. (2020). Graph signal processing for machine learning: A review and new perspectives. *IEEE Signal Processing Magazine*, 37(6):117–127.

Dowson, D. and Landau, B. (1982). The Fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12(3):450–455.

Dubey, P. and Müller, H.-G. (2019). Fréchet analysis of variance for random objects. *Biometrika*, 106(4):803–821.

El Gheche, M., Chierchia, G., and Frossard, P. (2020). Orthonet: multilayer network data clustering. *IEEE Transactions on Signal and Information Processing over Networks*, 6:152–162.

Elvander, F., Haasler, I., Jakobsson, A., and Karlsson, J. (2020). Multi-marginal optimal transport using partial information with applications in robust localization and sensor fusion. *Signal Processing*, 171:107474.

Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305.

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). Physiobank, PhysioToolkit, and Physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220.

Gutman, I. and Xiao, W. (2004). Generalized inverse of the Laplacian matrix and some applications. *Bulletin (Académie serbe des sciences et des arts. Classe des sciences mathématiques et naturelles. Sciences mathématiques)*, pages 15–23.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. (2020). Open graph benchmark: Datasets for machine learning on graphs. *Advances in Neural Information Processing Systems*, 33:22118–22133.

Kalofolias, V. (2016). How to learn a graph from smooth signals. *Artificial Intelligence and Statistics*, pages 920–929.

Kang, Z., Shi, G., Huang, S., Chen, W., Pu, X., Zhou, J. T., and Xu, Z. (2020). Multi-graph fusion for multi-view spectral clustering. *Knowledge-Based Systems*, 189:105102.

Karimi, A., Ripani, L., and Georgiou, T. T. (2020). Statistical learning in Wasserstein space. *IEEE Control Systems Letters*, 5(3):899–904.

Kolaczyk, E. D., Lin, L., Rosenberg, S., Walters, J., and Xu, J. (2020). Averages of unlabeled networks: Geometric characterization and asymptotic behavior. *The Annals of Statistics*, 48(1):514 – 538.

Kolouri, S., Park, S. R., Thorpe, M., Slepcev, D., and Rohde, G. K. (2017). Optimal mass transport: Signal processing and machine-learning applications. *IEEE Signal Processing Magazine*, 34(4):43–59.

Lambert, M., Chewi, S., Bach, F., Bonnabel, S., and Rigollet, P. (2022). Variational inference via Wasserstein gradient flows. *arXiv preprint arXiv:2205.15902*.

Lim, Y. and Pálfia, M. (2012). Matrix power means and the karcher mean. *Journal of Functional Analysis*, 262(4):1498–1514.

Lunagómez, S., Olhede, S. C., and Wolfe, P. J. (2021). Modeling network populations via graph distances. *Journal of the American Statistical Association*, 116(536):2023–2040.

Majeed, A. and Rauf, I. (2020). Graph theory: A comprehensive survey about graph theory applications in computer science and social networks. *Inventions*, 5(1):10.

Maretic, H. P., El Gheche, M., Chierchia, G., and Frossard, P. (2022a). FGOT: Graph distances based on filters and optimal transport. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7710–7718.

Maretic, H. P., El Gheche, M., Minder, M., Chierchia, G., and Frossard, P. (2022b). Wasserstein-based graph alignment. *IEEE Transactions on Signal and Information Processing over Networks*, 8:353–363.

Mateos, G., Segarra, S., Marques, A. G., and Ribeiro, A. (2019). Connecting the dots: Identifying network structure via graph signal processing. *IEEE Signal Processing Magazine*, 36(3):16–43.

McCann, R. J. (1997). A convexity principle for interacting gases. *Advances in Mathematics*, 128(1):153–179.

Mercado, P., Tudisco, F., and Hein, M. (2019a). Generalized matrix means for semi-supervised learning with multilayer graphs. *Advances in Neural Information Processing Systems*, 32.

Mercado, P., Tudisco, F., and Hein, M. (2019b). Spectral clustering of signed graphs via matrix power means. *International Conference on Machine Learning*, pages 4526–4536.

Meyer, F. G. (2022). The Fréchet mean of inhomogeneous random graphs. *Complex Networks & Their Applications X*, pages 207–219.

Oehlers, M. and Fabian, B. (2021). Graph metrics for network robustness—a survey. *Mathematics*, 9(8):895.

Olfati-Saber, R., Fax, J. A., and Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233.

Olkin, I. and Pukelsheim, F. (1982). The distance between two random vectors with given dispersion matrices. *Linear Algebra and its Applications*, 48:257–263.

Ortega, A., Frossard, P., Kovačević, J., Moura, J. M., and Vandergheynst, P. (2018). Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828.

Petric Maretic, H., El Gheche, M., Chierchia, G., and Frossard, P. (2019). GOT: an optimal transport framework for graph comparison. *Advances in Neural Information Processing Systems*, 32.

Peyré, G., Cuturi, M., et al. (2019). Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Peyré, G., Cuturi, M., and Solomon, J. (2016). Gromov-Wasserstein averaging of kernel and distance matrices. *International Conference on Machine Learning*, pages 2664–2672.

Sandhu, R., Georgiou, T., Reznik, E., Zhu, L., Kolesov, I., Senbabaoglu, Y., and Tannenbaum, A. (2015). Graph curvature for differentiating cancer networks. *Scientific reports*, 5(1):12323.

Schalk, G., McFarland, D. J., Hinterberger, T., Birbaumer, N., and Wolpaw, J. R. (2004). Bci2000: a general-purpose brain-computer interface (BCI) system. *IEEE Transactions on Biomedical Engineering*, 51(6):1034–1043.

Solomon, J., Peyré, G., Kim, V. G., and Sra, S. (2016). Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics (ToG)*, 35(4):1–13.

Songdechakraiwut, T., Krause, B. M., Banks, M. I., Nourski, K. V., and Van Veen, B. D. (2021). Fast topological clustering with Wasserstein distance. *International Conference on Learning Representations*.

Takatsu, A. (2010). On Wasserstein geometry of Gaussian measures. *Probabilistic Approach to Geometry*, pages 463–472.

Tantardini, M., Ieva, F., Tajoli, L., and Piccardi, C. (2019). Comparing methods for comparing networks. *Scientific Reports*, 9(1):1–19.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622.

Vayer, T., Courty, N., Tavenard, R., and Flamary, R. (2019). Optimal transport for structured data with application on graphs. *International Conference on Machine Learning*, pages 6275–6284.

Villani, C. (2021). *Topics in optimal transportation*, volume 58. American Mathematical Soc.

Wills, P. and Meyer, F. G. (2020). Metrics for graph comparison: A practitioner's guide. *Plos one*, 15(2):e0228728.

Xia, G.-S., Ferradans, S., Peyré, G., and Aujol, J.-F. (2014). Synthesizing and mixing stationary Gaussian texture models. *SIAM Journal on Imaging Sciences*, 7(1):476–508.

Xu, H., Luo, D., and Carin, L. (2019a). Scalable Gromov-Wasserstein learning for graph partitioning and matching. *Advances in Neural Information Processing Systems*, 32:3052–3062.

Xu, H., Luo, D., Zha, H., and Duke, L. C. (2019b). Gromov-Wasserstein learning for graph matching and node embedding. *International Conference on Machine Learning*, pages 6932–6941.

Bures Wasserstein Means of Graphs: Supplementary Materials

In the following we provide supplementary material for several aspects of the proposed Bures-Wasserstein graph mean framework. In Section 6 we discuss Assumption 3.1, in Section 7 we explain how to extend the proposed frameworks to take into account graph filters, and in Section 8 we provide details on the numerical experiments. Finally, Section 9 contains the proofs of the theoretical results in the paper.

6 DISCUSSION OF ASSUMPTION 3.1

In this Section we discuss the class of graphs we consider in this work, namely the class of signed weighted graphs with Laplacian matrix that is positive semi-definite with only one zero eigenvalue, as stated in Assumption 3.1.

Signed graphs are a powerful object to describe complex systems with positive as well as negative interactions. Negative edge weights could for instance model antagonistic relationships in social networks, or model inhibition and repression of expressions in gene regulatory networks. In fact, Assumption 3.1 is ubiquitous for social consensus networks, as it is a sufficient condition for the system to converge to a consensus (Olfati-Saber et al., 2007).

For signed graphs, the spectral properties are more difficult to analyze than for graphs with positive edge weights. For instance, note that for graphs with only positive edge weights the multiplicity of the zero eigenvalue equals the number of disjoint components of the graph (Fiedler, 1973). Thus, if a graph with positive edge weights has a Laplacian with only one zero eigenvalue then it is connected. However, this property does not extend to signed graphs: a connected signed graph may have a Laplacian with multiple zero eigenvalues. On the other hand, it was shown that any signed weighted graph with a Laplacian that has a single zero eigenvalue is connected (Chen et al., 2016a, Lemma 1).

Since we model graphs through their smooth graph signal distribution, a negative edge weight between two nodes is associated with smooth signals that have high dissimilarity between these two nodes. As a result, a negative edge could appear in the mean graph if the corresponding nodes are far away from each other in all input graphs, because in this case there is strong evidence for graph signals to be very dissimilar on the two nodes. As an example, consider the setting in the right panel of Figure 1. Here we find the mean of a set of graphs that are very densely connected: Most pairs of nodes are connected in at least one of the input graphs. However, the node pair $(0, 4)$ is not connected in any of the input graphs, thus relative to all other node pairs these two nodes are very far from each other in all given data samples. From the given data it is thus expected that these two nodes are dissimilar, which is described by a negative edge weight in the mean graph.

Finally, we note that in contrast to our work, the previous works Petric Maretic et al. (2019); Maretic et al. (2022b) defined a similar distance to the Bures-Wasserstein distance, but only for unsigned graphs. However, as observed in the example of the right panel of Figure 1, the Bures-Wasserstein mean may exhibit negative edge weights even if the input graphs have only positive edge weights[6]. In fact, the graphs with positive edge weights do not define a geodesically complete space with respect to the Bures-Wasserstein distance. From this we conclude that Assumption 3.1 defines a more sensible class of graphs for the Bures-Wasserstein distance.

7 DETAILS ON BURES-WASSERSTEIN FILTER GRAPH MEANS

In this Section we review some background on the filtered graph optimal transport distance introduced in Maretic et al. (2022a), and discuss its application to the graph mean problem (1).

[6]It is worth noting that this is the case for many other graph means based on the Laplacian matrix, for instance power means with negative power (e.g., harmonic mean), and geometric means.

7.1 Graph filter distance

Recall that the Bures-Wasserstein distance for graphs is inspired from graph signal processing, a field that aims at generalizing classical signal processing concepts to graphs. Using these tools, we can incorporate signal processing techniques, like filters, into the Bures-Wasserstein distance. A graph signal $x \in \mathbb{R}^N$ filtered by the filter $g : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$, which is acting on the graph Laplacian L, is of the form $g(L)x$. Analogously to classical filters, low-pass graph filters emphasise the low frequencies of the graph signals, which correspond to slow variations of the signal between strongly connected nodes (Ortega et al., 2018). On the other hand, high-pass graph filters emphasise the large frequencies, i.e., higher variations between connected nodes.

Given a white Gaussian noise signal on the graphs nodes, denoted $w \in \mathbb{R}^N$, the filtered signal $g(L)w$ follows the normal distribution $\mathcal{N}(0, g(L)^2)$, see Maretic et al. (2022a). A class of filtered graph distances can be defined by utilizing the filtered graph signals in Definition 3.2 as follows.

Definition 7.1. Let G_0 and G_1 be two graphs with signed weighted Laplacian matrices L_0 and L_1. Given a graph filter $g : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$, the corresponding graph filter distance between the two graphs is defined as $d^g_{BW}(G_0, G_1) := \mathcal{W}_2(\mu_{G^g_0}, \mu_{G^g_1})$, where $\mu_{G^g_j} \sim \mathcal{N}(0, g(L_j)^2)$ for $j = 0, 1$.

Hence, essentially different graph filter distances correspond to different embeddings of the graphs into the space of signal distributions. Utilizing a low-pass filter in Definition 7.1 results in a distance that captures mainly differences in the global graph behavior, and a high-pass filter measures predominantly local differences in the graphs, as observed in Maretic et al. (2022a). Finally, note that the filter $g(L) = L^{\dagger/2}$ gives rise to the smooth graph signal distribution $\mathcal{N}(0, L^\dagger)$. Thus, utilizing this filter in the Bures-Wasserstein filter distance in Definition 7.1 results in the standard Bures-Wasserstein distance for graphs presented in Definition 3.2.

7.2 Bures-Wasserstein filter graph mean

We now consider using these filter graph distances in the Bures-Wasserstein mean problem (7). By emphasising different types of spectral information of the input graphs, we expect to retrieve a mean graph that preserves the corresponding type of spectral properties of the input graphs. We define the Bures-Wasserstein filter graph mean as the solution of

$$\bar{G} = \underset{G\in\mathcal{G}}{\arg\min} \sum_{j=1}^{m} \lambda_j d^g_{BW}(G, G_j)^2 = \underset{G\in\mathcal{G}}{\arg\min} \sum_{j=1}^{m} \lambda_j \mathcal{W}_2(\mu^g_G, \mu_{G^g_j})^2. \tag{11}$$

In order to recover a mean graph G from the Wasserstein barycenter μ^g_G, we require that there is a one-to-one mapping from the set of graphs in $\mathcal{G}$ to the set of signal distributions. In other words, the graph filter is required to be bijective on the set of Laplacian matrices that satisfy Assumption 3.1. With this assumption we can generalize the existence and uniqueness results from Section 3.

Corollary 7.2. *Let G_j, $j = 1, \dots, m$ be graphs with signed weighted Laplacians L_j that satisfy Assumption 3.1, and let $g : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ be a bijective mapping on the set of matrices that satisfy Assumption 3.1. Then the Bures-Wasserstein filter mean graph (11) of these graphs has the signed weighted Laplacian matrix $L = g^{-1}(S^{1/2})$, where S is the unique positive semi-definite symmetric solution to*

$$S = \sum_{j=1}^{m} \lambda_j \left(S^{1/2} g(L_j)^2 S^{1/2}\right)^{1/2}$$

that satisfies range$(S) = \mathcal{R} = (\text{span}\{\mathbf{1}_N\})^\perp$.

Proof. The result follows directly from Theorem 3.3, since the filter g is bijective. □

Corollary 7.3. *Let G_0 and G_1 be two graphs with signed weighted Laplacians L_0 and L_1 that satisfy Assumption 3.1, and let $g : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ be a bijective mapping on the set of matrices that satisfy Assumption 3.1. Then the Bures-Wasserstein filter mean of these two graphs with weights $1 - t$ and t for $t \in (0, 1)$ has the signed weighted graph Laplacian $L_t = g^{-1}(S_t^{1/2})$, where*

$$S_t = g(L_0)^\dagger \left((1-t)g(L_0)^2 + t\left(g(L_0)g(L_1)^2 g(L_0)\right)^{1/2}\right)^2 g(L_0)^\dagger.$$

Proof. The result follows directly from Theorem 3.5, since the filter g is bijective. □

Algorithm 2 Bures-Wasserstein filter mean of graphs.

Given: Matrices $L_1, \dots, L_m \in \mathbb{R}^{N\times N}$ satisfying Assumption 3.1, and initial SPD matrix $S \in \mathbb{R}^{N\times N}$

$\Sigma_j \leftarrow g\left(L_j + \frac{1}{N}\mathbf{1}_{N\times N}\right)^2$ for $j = 1, \dots, m$

while Not converged **do**

$$S \leftarrow S^{-1/2}\left(\sum_{j=1}^{m} \lambda_j (S^{1/2}\Sigma_j S^{1/2})^{1/2}\right)^2 S^{-1/2},$$

end while

Return: $L \leftarrow g^{-1}(S^{1/2}) - \frac{1}{N}\mathbf{1}_{N\times N}$

Table 4: Difference of the perturbed graphs' mean to the original graph with respect to several metrics. Compare Table 1.

	Laplacian	Covariance	degree centrality	modularity	participation coeff.
Arithmetic	1.85 ± 0.10	0.26 ± 0.18	$\mathbf{0.026 \pm 0.003}$	0.015 ± 0.003	0.37 ± 0.13
Harmonic	2.87 ± 0.19	$\mathbf{0.13 \pm 0.10}$	0.047 ± 0.004	0.012 ± 0.002	0.26 ± 0.09
Geometric	7.25 ± 0.36	0.38 ± 0.17	0.103 ± 0.010	0.015 ± 0.009	0.40 ± 0.12
$g(L) = L^\dagger$	2.39 ± 0.13	$\mathbf{0.13 \pm 0.08}$	0.039 ± 0.003	$\mathbf{0.009 \pm 0.002}$	$\mathbf{0.22 \pm 0.08}$
$g(L) = L^{\dagger/2}$	2.26 ± 0.11	0.15 ± 0.12	0.036 ± 0.002	$\mathbf{0.011 \pm 0.003}$	$\mathbf{0.25 \pm 0.09}$
$g(L) = L^{1/2}$	$\mathbf{1.81 \pm 0.10}$	0.21 ± 0.13	$\mathbf{0.026 \pm 0.003}$	0.012 ± 0.002	0.30 ± 0.11
$g(L) = L$	$\mathbf{1.68 \pm 0.10}$	0.23 ± 0.14	$\mathbf{0.025 \pm 0.003}$	0.012 ± 0.002	0.30 ± 0.12

Finally, we can generalize Algorithm 1 to taking into account graph filters. The resulting method is presented in Algorithm 2.

7.3 Experimental results

We now illustrate the behavior of the Bures-Wasserstein filter mean graphs with respect to several filters. In particular, we study the graph filters summarized in the following table.

$g(L) = L^\dagger$	most low-pass
$g(L) = L^{\dagger/2}$	standard Bures-Wasserstein
$g(L) = L^{1/2}$	
$g(L) = L$	most high-pass

We present the results of some of the experiments considered in Section 4 utilizing these filters in the Bures-Wasserstein filter graph mean problem (7.2). The respective optimization problems are solved using Algorithm 2.

7.3.1 Graph fusion

First, we consider the graph fusion experiment that is set up as described in Section 4.1. Table 4 shows how the graph metrics are preserved for different types of Bures-Wasserstein means. As expected, the low-pass filters preserve mostly global graph metrics, such as modularity, and the covariance matrix, whereas the high-pass filters preserve local behavior, such as degree centrality and the Laplacian matrix. These findings highlight the flexibility and versatility of our proposed graph filter means, providing practitioners with an easy-to-use framework that can be readily adapted to different applications, depending on the relative importance of local versus global graph behavior.

7.3.2 Semi-supervised learning in multi-layer graphs

Next, we repeat the node-classification experiments in Section 4.4, and include also the low-pass filter $g(L) = L^\dagger$ and the high-pass filter $g(L) = L^{1/2}$. The results are presented in Figure 3. Overall, we observe that the tested graph filters exhibit strong performance in most experiments. For a detailed comparison of the datasets, we refer to Section 8.4. As a general trend, the high-pass filter mean tends to perform very well, particularly on the

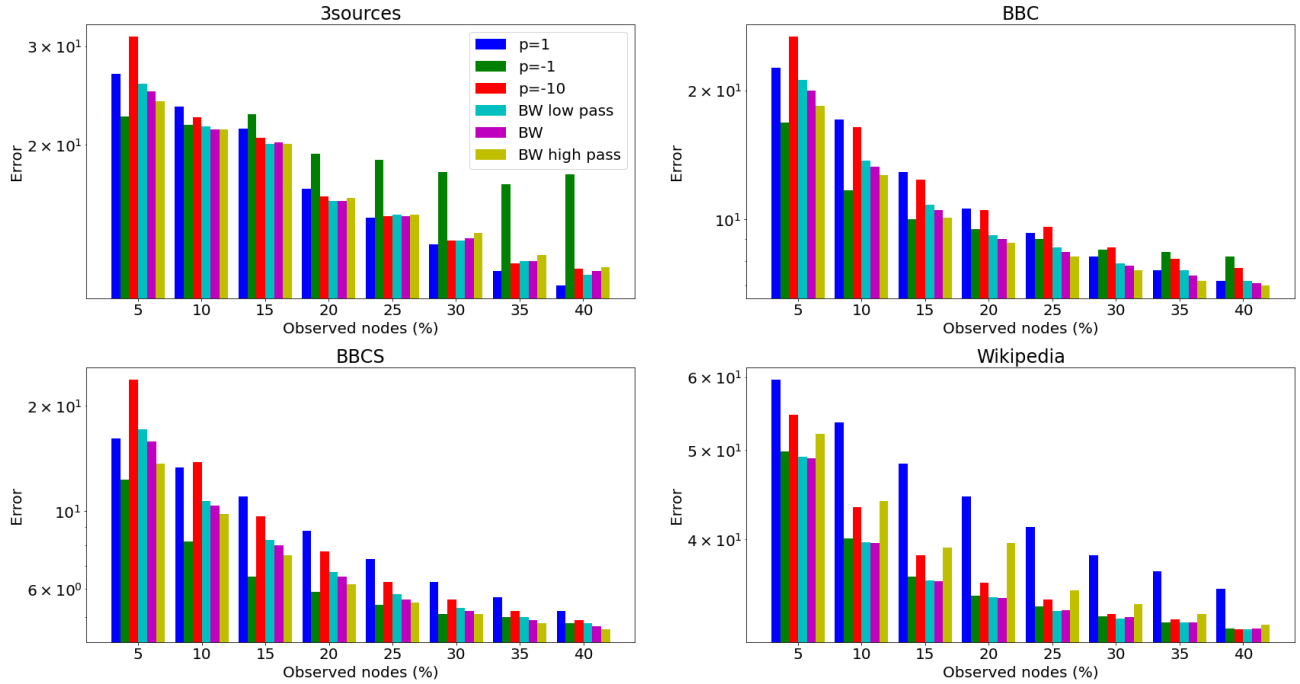

Figure 3: Test error of node classification for different percentage of observed node labels using several types of graph means and distances. Here, *BW low pass* and *BW high pass* correspond to the filter distance with $g(L) = L^{\dagger}$ and $g(L) = L^{1/2}$, respectively.

more homogeneous datasets such as *BBC* and *BBCS*. However, in the more challenging datasets *3sources* and *Wikipedia* the low-pass filter mean may outperform the high-pass filter mean. Note that all tested graph means have the highest error rates for the *Wikipedia* dataset. Especially for this challenging dataset the more low-pass filter means demonstrate significant improvements over the high-pass filter mean.

As already observed in Section 4.4, none of the power means performs well in all scenarios, which is in contrast to the Bures-Wasserstein means. For instance, the harmonic mean (i.e., the power mean with $p = -1$) performs very poorly on the *3sources* dataset when more than 25% of the nodes are observed, and the arithmetic mean (i.e., the power mean with $p = 1$) does not yield satisfactory results on the *Wikipedia* dataset. These observations lead us to conclude that the Bures-Wasserstein filter means offer a reliable family of graph means, capable of delivering consistent performance across various datasets.

8 DETAILS ON EXPERIMENTS

In this Section we present details on the numerical experiments in Section 4. Specifically, in the following subsections we describe the baseline methods, discuss computational aspects of our proposed Algorithm 1, and we provide more information on the data for the k-means clustering example in Section 4.2, and the semi-supervised node-classification problem in 4.4.

8.1 Baseline methods

In this section we briefly describe the baseline methods considered in Figure 1 and Section 4.

Various means for scalars can be generalized to symmetric positive definite matrices (Lim and Pálfia, 2012), and have been utilized for graph Laplacian matrices (Mercado et al., 2019a,b; El Gheche et al., 2020). These means parameterize the class of graphs $\mathcal{G}$ in the graph mean problem (1) by the graph Laplacian matrices of size $\mathbb{R}^{N\times N}$. For instance, when using the Frobenius norm as the distance $d(\cdot,\cdot)$ in (1), the mean of a set of graph Laplacians $L_1,\dots,L_m$ is their arithmetic mean $\sum_{j=1}^m \lambda_j L_j$. Similarly, the harmonic mean is defined as $\left(\sum_{j=1}^m \lambda_j L_j^{\dagger}\right)^{\dagger}$. These two means are special cases of the so-called power means introduced in Section 4.4.

The generalization of the geometric mean from scalars to symmetric positive definite matrices is less straightforward, and cannot be expressed in closed form (Lim and Pálfia, 2012). In this work we compute the geometric mean iteratively as described in El Gheche et al. (2020, Section IV.C).

In Sections 4.2 and 4.3 we evaluate also a topological clustering method that compares persistence barcodes of the graphs using the Wasserstein distance Songdechakraiwut et al. (2021). The retrieved mean thus compares topological features of the input graphs, but does not preserve the node alignment. In order to address this the authors propose a hybrid method of topological and arithmetic mean, which preserves node correspondences. In this work, we utilise a weighting factor of $\lambda = \frac{1}{2}$ for the topological and arithmetic component of this hybrid mean.

8.2 Computational aspects

In this section, we examine the behavior of our proposed computational method outlined in Algorithm 1. As highlighted in Section 3, the fixed-point iteration employed in Algorithm 1 is based on a similar iteration used for the Wasserstein barycenter problem (4) with normal distributions as input (Álvarez-Esteban et al., 2016). Consequently, this fixed-point iteration converges towards the unique solution of the matrix equation (8), which characterizes the Bures-Wasserstein mean graph. Let $S^{(n)} \in \mathbb{R}^{N\times N}$ denote the outcome of the fixed-point iteration after n iterations. In our experiments, we adopt the stopping criterion $|S^{(n)} - S^{(n-1)}| < 10^{-5}$, which is typically achieved within a few iterations ($n \leq 5$). However, it is important to note that to our knowledge there is currently no theoretical analysis available regarding the convergence rate.

We observe that each iteration of the fixed-point method in Algorithm 1 involves computing one matrix inverse and performing $m + 1$ square roots of symmetric positive definite matrices. As a result, the computational complexity of one fixed-point iteration in the algorithm is $\mathcal{O}(mN^3)$. It is worth mentioning that the factor m can be effectively addressed by parallelizing the summation in Algorithm 1. The cubic dependence in N may be addressed by using approximations of the matrix inverse, and matrix roots, or by exploiting graph sparsity. Investigating ways to enhance the computational efficiency of Algorithm 1 is an important direction for future work.

8.3 Details on k-means clustering in Section 4.2

The task in the k-means clustering example in Section 4.2 is to classify structured graphs with different numbers of communities. In Figure 4 we show some graph samples from the dataset. More precisely, we consider the setting with $N_C = 5$ classes, and showcase four samples from each of these classes of graphs. Here, different classes are defined by the number of communities, and communities are arranged on a line. There are two different connectivity patterns between neighbouring communities, which are each picked with probability 1/2. Each of these two patterns constructs one edge between the communities, between a different set of nodes.

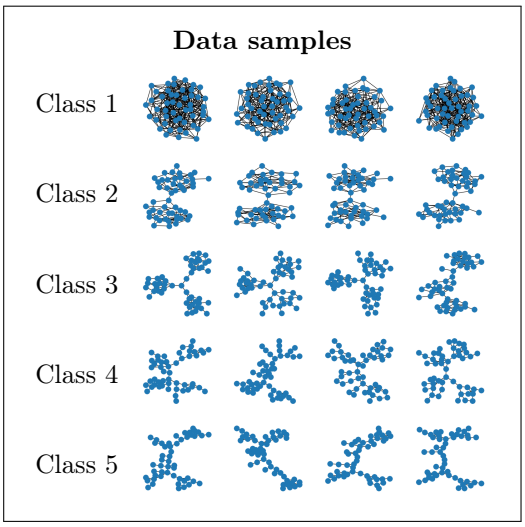

Figure 4: Some data samples of the k-means clustering example in Section 4.2 with $k = N_C = 5$ clusters and inter-community edge probability $p = 0.2$.

Figure 5 illustrates the evolution of centroids computed by the k-means classifier visually. The method is initialized with five graphs picked randomly from the dataset. In this example the k-means clustering method converges in two iterations. We note here that the five final centroids are graphs, where the number of communities ranges from one to five. Moreover, we can see that the two distinct connectivity patterns between neighboring communities are both captured by the centroids. We thus conclude that the final centroids provide meaningful representations of the graphs in each of the five classes.

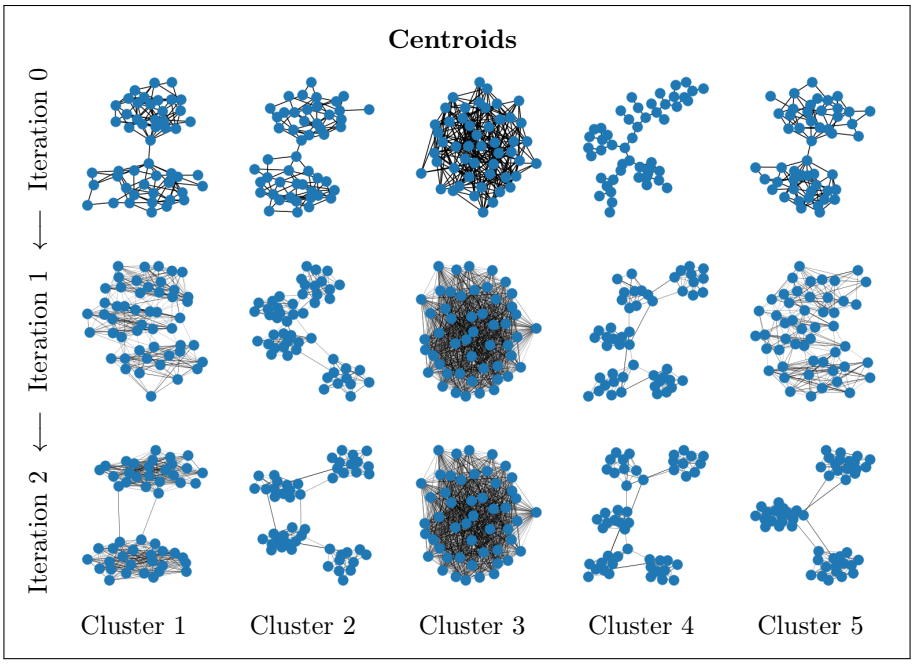

Figure 5: Evolution of the centroids for one trial of the k-means clustering example with data as illustrated in Figure 4.

Table 5: Characteristics of the multi-layer graphs used in the semi-supervised node classification experiments in Section 4.4.

	Layers	Nodes	Labels	Edges layers	Mean degree centrality average	Mean degree centrality layers
3sources	3	169	6	1084, 1139, 1188	13.4	12.8, 13.5, 14.1
BBC	4	685	5	4806, 4775, 4816, 4874	14.1	14.0, 14.0, 14.1, 14.2
BBCS	2	544	5	3803, 3855	14.1	14.0, 14.17
Wikipedia	2	693	10	5260, 5038	14.9	15.2, 14.5

8.4 Details on semi-supervised learning in multi-layer graphs in Section 4.4

For the semi-supervised node-classification problem in Section 4.4, we use four commonly used datasets: *3sources*, *BBC*, *BBCS*, and *Wikipedia*. Here, layers correspond to different features of the data, and for each layer a k-nearest neighbour graph is constructed based on the Pearson linear correlation between nodes, as described in Mercado et al. (2019a). Thus, nodes with high correlation are close to each other in the resulting graphs. Table 5 summarizes some characteristics of the resulting multi-layer graphs. We observe that the datasets *BBC* and *BBCS* exhibit a higher level of homogeneity between the layers in terms of the number of edges and average degree centrality. This finding explains the consistent and relatively similar performance of all tested methods on these datasets, as discussed in Section 4.4. However, the more complex datasets, such as *3sources* and *Wikipedia*, exhibit greater heterogeneity between the layers, leading to more pronounced differences in the performance of the tested methods. Moreover, these datasets present additional challenges. For instance, in the *3sources* dataset, the graphs have the smallest number of nodes, but have similar degree centrality, and thus their nodes are most strongly connected, resulting in less informative structural information. On the other hand, the *Wikipedia* dataset involves 10 different types of node labels, making the classification task particularly challenging. Remarkably, as demonstrated in Sections 4.4 and 7.3.2, our proposed Bures-Wasserstein mean consistently outperforms state-of-the-art methods, particularly on these challenging datasets.

9 PROOFS

In this Section we present the proofs of the main results in Section 3.

9.1 Proof of Theorem 3.3

We now present a proof of the the main result of the paper, which asserts the existence and uniqueness of a solution to the Bures-Wasserstein mean problem for graphs. To prove the result, we first show that the involved optimal transport problems on $\mathbb{R}^N$ can be projected into the space $\mathcal{R} = (\text{span}\{\mathbf{1}_n\})^\perp$ without changing the optimal solution.

Lemma 9.1. *Let μ_0, μ_1 be two probability measures with support on a d-dimensional subspace $\mathcal{R} \subset \mathbb{R}^N$, and let $P : \mathbb{R}^N \to \mathbb{R}^d$ be a mapping that is distance preserving between $\mathcal{R}$ and $\mathbb{R}^d$. Denote $\mathcal{W}_2^{\mathcal{X}}(\cdot,\cdot)$ the Wasserstein distance in space $\mathcal{X}$. Then it holds that*

$$\mathcal{W}_2^{\mathbb{R}^N}(\mu_0, \mu_1) = \mathcal{W}_2^{\mathbb{R}^d}(P_\# \mu_0, P_\# \mu_1). \tag{12}$$

Proof. Since the measures μ_0 and μ_1 have support in $\mathcal{R}$ it follows that any feasible transport plan in $\Pi(\mu_0, \mu_1)$ has support in $\mathcal{R} \times \mathcal{R}$. Thus, the isometry between $\mathcal{R}$ and $\mathbb{R}^d$ given by the mapping P infers a bijection $P \times P$ between the feasibility sets $\Pi(\mu_0, \mu_1)$ and $\Pi(P_\#\mu_0, P_\#\mu_1)$ of the optimization problems on the left hand side and right hand side of equation (12), respectively. More precisely, let $P^{-1} : \mathbb{R}^d \to \mathcal{R}$ be the inverse of P restricted to $\mathcal{R}$. Then, for any measurable set $A \subset \mathcal{R}$ it holds that

$$\pi(A, \mathbb{R}^N) = \mu_0(A) \implies \Big((P \times P)_\# \pi\Big)\big(P(A) \times \mathbb{R}^d\big) = (P_\#\mu_0)\,(P(A)),$$

and for any measurable set $\hat{A} \subset \mathbb{R}^d$ and measure $\hat{\pi} \in \mathbb{R}^d \times \mathbb{R}^d$ it holds that

$$\hat{\pi}(\hat{A}, \mathbb{R}^d) = (P_\#\mu_0)(\hat{A}) \implies \Big(\big(P^{-1} \times P^{-1}\big)_\# \hat{\pi}\Big)\Big(P^{-1}(\hat{A}) \times \mathbb{R}^N\Big) = \mu_0(P^{-1}(A)).$$

Similar relationships hold for the second marginal of the product measures and μ_1. Hence it follows that

$$\pi \in \Pi(\mu_0, \mu_1) \iff (P \times P)_\# \pi \in \Pi(P_\#\mu_0, P_\#\mu_1).$$

Finally, note that for any $\pi \in \Pi(\mu_0, \mu_1)$, where μ_0 and μ_1 have support in $\mathcal{R}$, since π has support in $\mathcal{R} \times \mathcal{R}$, it holds that

$$\begin{aligned}\int_{\mathbb{R}^N \times \mathbb{R}^N} \|x - y\|_2^2 \, d\pi(x,y) &= \int_{\mathcal{R} \times \mathcal{R}} \|x - y\|_2^2 \, d\pi(x,y) \\ &= \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|_2^2 \, d\left((P \times P)_\# \pi\right)(x,y).\end{aligned}$$

Thus, the objective values of the optimization problems in equation (12) coincide for π on the left hand side of (12) and $(P \times P)_\#\pi$ on the right hand side of (12). This concludes the equality of the Wasserstein distances in (12). □

We can now prove Theorem 3.3.

Proof of Theorem 3.3. We prove the result by considering the Wasserstein barycenter problem (4) of the measures $\mu_{G_j} \sim \mathcal{N}(0, L_j^\dagger)$ for $j = 1, \dots, m$. It turns out that its solution in fact defines a unique solution to the Bures-Wasserstein mean graph problem (7).

Since the Laplacians L_j satisfy Assumption 3.1, they all have the same nullspace $\text{span}\{\mathbf{1}_N\}$ and thus, the support of all the probability measures $\mu_{G_j} \sim \mathcal{N}(0, L_j^\dagger)$ for $j = 1, \dots, m$ is the subspace $\mathcal{R} = (\text{span}\{\mathbf{1}_N\})^\perp$ of $\mathbb{R}^N$. Consider the orthogonal projection $P^\perp : \mathbb{R}^N \to \mathcal{R}$, and note that it holds $\mathcal{W}_2^{\mathbb{R}^N}(\mu, \mu_{G_j}) \geq \mathcal{W}_2^{\mathbb{R}^N}(P^\perp_{\#\mu}, \mu_{G_j})$ for any probability measure μ on $\mathbb{R}^N$. Thus, the Wasserstein barycenter μ of the measures μ_{G_j}, for $j = 1, \dots, m$, also has support $\mathcal{R}$.

Consider a matrix $U \in \mathbb{R}^{N\times N-1}$ whose columns form an orthonormal basis of the subspace $\mathcal{R}$ in $\mathbb{R}^N$. Then, the mapping $P : \mathbb{R}^N \to \mathbb{R}^{N-1}$ defined as $v \mapsto U^T v$ is distance preserving between $\mathcal{R}$ and $\mathbb{R}^{N-1}$. Hence, applying Lemma 9.1 to every term in the sum of the Wasserstein barycenter problem (4) yields that

$$\underset{\mu}{\text{minimize}} \sum_{j=1}^{m} \lambda_j \mathcal{W}_2^{\mathbb{R}^N}(\mu, \mu_{G_j})^2 = \underset{\bar{\mu}}{\text{minimize}} \sum_{j=1}^{m} \lambda_j \mathcal{W}_2^{\mathbb{R}^{N-1}}(\bar{\mu}, P_{\#}\mu_{G_j})^2, \tag{13}$$

where μ is a measure on $\mathbb{R}^N$ and $\bar{\mu}$ is a measure on $\mathbb{R}^{N-1}$. Moreover, the minimizers μ and $\bar{\mu}$ of the left hand side and ride hand side of (13), respectively, satisfy $P_{\#}\mu = \bar{\mu}$, since μ has support in $\mathcal{R}$.

Note that the projected probability measures on $\mathbb{R}^{N-1}$ are given by

$$P_{\#}\mu_{G_j} \sim \mathcal{N}\left(0, U^T L_j^{\dagger} U\right),$$

where the projected covariance matrices $U^T L_j^{\dagger} U \in \mathbb{R}^{(N-1)\times(N-1)}$ are strictly positive definite and symmetric. Thus, by (Agueh and Carlier, 2011, Theorem 6.1) the barycenter problem in the right hand side of (13) has a unique solution. Moreover, by Agueh and Carlier (2011, Theorem 6.1) the unique solution to the barycenter problem in the right hand side of (13) is of the form $\bar{\mu} \sim \mathcal{N}(0, \bar{\Sigma})$, and the covariance matrix $\bar{\Sigma}$ is the unique positive definite solution to (6), where $\Sigma_j = U^T L_j^{\dagger} U$ for $j = 1, \dots, m$. It follows that

$$\begin{aligned}
\sum_{j=1}^{m} \lambda_j \left((U\bar{\Sigma}U^T)^{1/2} L_j^{\dagger} (U\bar{\Sigma}U^T)^{1/2}\right)^{1/2} &= \sum_{j=1}^{m} \lambda_j \left(U\bar{\Sigma}^{1/2} U^T L_j^{\dagger} U \bar{\Sigma}^{1/2} U^T\right)^{1/2} \\
&= U \left(\sum_{j=1}^{m} \lambda_j \left(\bar{\Sigma}^{1/2} \Sigma_j \bar{\Sigma}^{1/2}\right)^{1/2}\right) U^T \\
&= U\bar{\Sigma}U^T.
\end{aligned}$$

Thus, the matrix $S = U\bar{\Sigma}U^T \in \mathbb{R}^{N\times N}$ is a solution to (8). Moreover, this matrix is symmetric positive semi-definite and has the range $\mathcal{R}$. Thus, $\mu \sim \mathcal{N}(0, U\bar{\Sigma}U^T)$ has support $\mathcal{R}$ and is a minimizer of the left hand side in (13). This constitutes the existence of a positive semi-definite solution to (8) with range $\mathcal{R}$ that characterizes the barycenter of μ_{G_j} for $j = 1, \dots, m$.

Now assume that there is another positive semi-definite matrix $\hat{S} \in \mathbb{R}^{N\times N}$ with $\hat{S} \neq S$ that solves (8) and satisfies $\text{range}(\hat{S}) = \mathcal{R}$. Then $U^T \hat{S} U$ must be a solution to (6), where $\Sigma_j = U^T L_j^{\dagger} U$ for $j = 0, \dots, m$, and since this solution is unique it must hold $U^T \hat{S} U = \bar{\Sigma}$. However, since $\text{range}(\hat{S}) = \mathcal{R}$ the mapping $\hat{S} \to U^T \hat{S} U$ is a bijection on $\mathcal{R}$, and thus it holds $\hat{S} = U\bar{\Sigma}U^T$, which contradicts the assumption $\hat{S} \neq S$. This asserts the uniqueness of the fixed point.

Finally, we note that since $U\bar{\Sigma}U^T$ is symmetric positive semi-definite and has the range $\mathcal{R}$, the same holds for the matrix $L = (U\bar{\Sigma}U^T)^{\dagger}$. Thus, L is in fact a graph Laplacian matrix that satisfies Assumption 3.1.

□

9.2 Proof of Proposition 3.4

In the following we prove Proposition 3.4, which is used in Algorithm 1 for efficient and stable computation.

Proof. For ease of notation denote $\mathbf{1} = \mathbf{1}_{N\times N}$ in this proof. By Gutman and Xiao (2004, Theorem 4), for $j = 1, \dots, m$, the matrix $\Sigma_j = L_j^{\dagger} + \frac{1}{N}\mathbf{1}$ has the same eigenvectors as the matrix $L_j^{\dagger}$, but the zero-eigenvalue is exchanged for an eigenvalue one. Thus, the matrices Σ_j are symmetric strictly positive-definite.

Moreover, note that for any matrix $A \in \mathbb{R}^{N\times N}$ satisfying Assumption 3.1 it holds that

$$\left(A + \frac{1}{N}\mathbf{1}\right)^{\dagger} = A^{\dagger} + \frac{1}{N}\mathbf{1}, \qquad \left(A + \frac{1}{N}\mathbf{1}\right)^{1/2} = A^{1/2} + \frac{1}{N}\mathbf{1}.$$

Thus, if S satisfies the matrix equation (8) with $L_0, \dots, L_m$, and Assumption 3.1, it follows that

$$\begin{aligned}\sum_{j=1}^{m} \lambda_j \left(\left(S + \frac{1}{N}\mathbf{1}\right)^{1/2} \left(L_j + \frac{1}{N}\mathbf{1}\right)^{\dagger} \left(S + \frac{1}{N}\mathbf{1}\right)^{1/2} \right)^{1/2} &= \sum_{j=1}^{m} \lambda_j \left(S^{1/2} L_j^{\dagger} S^{1/2} + \frac{1}{N}\mathbf{1} \right)^{1/2} \\ &= \sum_{j=1}^{m} \lambda_j \left(S^{1/2} L_j^{\dagger} S^{1/2} \right)^{1/2} + \frac{1}{N}\mathbf{1}.\end{aligned}$$

Thus, $\bar{\Sigma} = L^{\dagger} + \frac{1}{N}\mathbf{1}$ is the unique positive definite solution of the matrix equation (8) with inputs $L_j + \frac{1}{N}\mathbf{1}$, for $j = 1, \dots, m$. □

9.3 Proof of Theorem 3.5

Finally, we present a proof for the analytical expression of the Bures-Wasserstein mean problem for graphs in case only two graphs are given, which can also be interpreted as an interpolation problem. We note that Theorem 3.5 can be proved using Lemma 9.1 and similar arguments as in the proof of Theorem 3.3. Here we present a direct and independent proof utilizing similar techniques as the proof of Xia et al. (2014, Proposition 6.1).

Proof. We use the fact that the optimal transport geodesic between $\mu_{G_0} \sim \mathcal{N}(0, L_0^{\dagger})$ and $\mu_{G_1} \sim \mathcal{N}(0, L_1^{\dagger})$ is defined by $\mu_t = ((1-t)I + tT)_{\#}\, \mu_{G_0}$, where T is a symmetric matrix that satisfies $T L_0^{\dagger} T = L_1^{\dagger}$ (Takatsu, 2010). It is easy to check that the matrix

$$T = L_0^{1/2} \left(L_0^{\dagger/2} L_1^{\dagger} L_0^{\dagger/2} \right)^{1/2} L_0^{1/2} \tag{14}$$

satisfies this identity, since all the matrices in the expression (14) have the same range $\mathcal{R} = (\mathrm{span}\{\mathbf{1}_n\})^{\perp}$. Thus, the geodesic is of the form $\mu_t \sim \mathcal{N}(0, \Sigma_t)$, and the covariance matrix can be expressed as

$$\begin{aligned}\Sigma_t =& \left((1-t)I + tT\right) L_0^{\dagger} \left((1-t)I + tT\right) \\ =& L_0^{1/2} \left((1-t)L_0^{\dagger} + t \left(L_0^{\dagger/2} L_1^{\dagger} L_0^{\dagger/2} \right)^{1/2} \right) L_0^{1/2} L_0^{\dagger} L_0^{1/2} \left((1-t)L_0^{\dagger} + t \left(L_0^{\dagger/2} L_1^{\dagger} L_0^{\dagger/2} \right)^{1/2} \right) L_0^{1/2} \\ =& L_0^{1/2} \left((1-t)L_0^{\dagger} + t \left(L_0^{\dagger/2} L_1^{\dagger} L_0^{\dagger/2} \right)^{1/2} \right)^2 L_0^{1/2}\end{aligned}$$

Since $L_0^{\dagger}$ is symmetric and positive semi-definite, from the first line it follows that Σ_t is also symmetric and positive semi-definite. Moreover, the range of Σ_t is $\mathcal{R}$, as it is composed of matrices with this range. The pseudo-inverse $\Sigma_t^{\dagger}$ inherits these properties and thus satisfies Assumption 3.1.

Now assume that there is another mapping $\hat{T}$ that satisfies $\hat{T} L_0^{\dagger} T = L_1^{\dagger}$. However, note that then

$$\left(L_0^{\dagger/2} T L_0^{\dagger/2} \right)^2 = L_0^{\dagger/2} L_1^{\dagger} L_0^{\dagger/2} = \left(L_0^{\dagger/2} \hat{T} L_0^{\dagger/2} \right)^2,$$

and thus by taking the square root we see that $\hat{T}$ and T are equal on $\mathcal{R}$. Hence, T in (14) is the unique mapping with range in $\mathcal{R}$ that satisfies $T L_0^{\dagger} T = L_1^{\dagger}$. This constitutes the uniqueness of the interpolant $L_t = \Sigma_t^{\dagger}$ satisfying Assumption 3.1. □